Lehrstuhl für Informatik 1
Friedrich-Alexander-Universität
Erlangen-Nürnberg

MASTERARBEIT

Maßnahmen gegen Nutzerverfolgung mittels Browserfingerprinting

Tim Grocki

Erlangen, 23. März 2015

Examiner: Prof. Dr. Felix Freiling
Advisor: Dr. Zinaida Benenson

Eidesstattliche Erklärung / Statutory Declaration

Hiermit versichere ich eidesstattlich, dass die vorliegende Arbeit von mir selbständig, ohne Hilfe Dritter und ausschließlich unter Verwendung der angegebenen Quellen angefertigt wurde. Alle Stellen, die wörtlich oder sinngemäß aus den Quellen entnommen sind, habe ich als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

I hereby declare formally that I have developed and written the enclosed thesis entirely by myself and have not used sources or means without declaration in the text. Any thoughts or quotations which were inferred from the sources are marked as such. This thesis was not submitted in the same or a substantially similar version to any other authority to achieve an academic grading.

Der Friedrich-Alexander-Universität, vertreten durch den Lehrstuhl für Informatik 1, wird für Zwecke der Forschung und Lehre ein einfaches, kostenloses, zeitlich und örtlich unbeschränktes Nutzungsrecht an den Arbeitsergebnissen der Arbeit einschließlich etwaiger Schutz- und Urheberrechte eingeräumt.

Nürnberg, den _____

Tim Grocki

Zusammenfassung

In dieser Arbeit wurde untersucht, ob Maßnahmen existieren, die Nutzer im Internet vor der Wiederidentifizierung durch Browserfingerprinting schützen. Dazu wurde eine Reihe von Maßnahmen mit Analyse, praktischer Überprüfung und Simulation daraufhin untersucht, ob sie vor Browserfingerprinting schützen.

Dabei wurden bei allen Gegenmaßnahmen Schwächen gefunden, die dafür sorgen, dass ein Schutz vor Browserfingerprinting nur in Spezialfällen garantiert werden kann. Die Möglichkeit, mit spezialisierten Fingerprintingskripten trotz deaktiviertem Javascript umfangreich Informationen über die Nutzer zu erheben, und die Möglichkeit, Fingerprints trotz Randomisierung zur Nutzerverfolgung zu nutzen, waren dabei besonders bemerkenswert, da sie bisher unbekannt waren.

Da keine zuverlässig wirksamen Maßnahmen gegen Browserfingerprinting gefunden werden konnten, wurde gefolgert, dass mehrere Maßnahmen kombiniert werden sollten, um einen möglichst guten Schutz vor Browserfingerprinting zu erlangen und hierfür wurde ein Vorgehen vorgeschlagen.

Abstract

This thesis researched measures to protect users against re-identification by browser-fingerprinting. To check whether reliable measures exist a number of measures were examinined using analysis, practical testing and simulation.

All countermeasures had weaknesses reducing guaranteed protection to special cases. Two results were especially notable since they were unknown previously. The first one is the possibility to obtain extensive information about users, despite deactivated Javascript, using specialised fingerprintingscripts. The second one is the ability to use the randomisation of fingerprints to track users.

Since no reliable measures against browser-fingerprinting were found it was concluded that multiple measures should be combined to achieve the best possible protection against browser-fingerprinting. An approach for this was proposed.

Inhaltsverzeichnis

1 EINLEITUNG

Die Privatsphäre von Nutzern im Internet ist ein Thema, zu dem viel Forschung betrieben wird, da das Internet einerseits das Potenzial hat, Nutzer anonym in Verbindung zu bringen, andererseits aber auch als Instrument einer umfassenden und automatisierten Überwachung dienen kann. Ein Aspekt dieser Überwachung ist das Erstellen von Nutzerprofilen, indem das Browsingverhalten beobachtet und analysiert wird. Dies kann beispielsweise durch dem Einsatz von Cookies oder die Beobachtung der IPs der Nutzer erreicht werden. Es gibt andererseits aber zahlreiche Werkzeuge wie TOR, Proxys oder Privatebrowsing-Modi, die das Beobachten des Browsingverhaltens der Nutzer verhindern oder einschränken sollen.

Eine Technik, die genutzt werden kann, um die Verfolgung eines Nutzer während seiner Browsingsession zu vollziehen, ist das Browserfingerprinting. Die Eignung von Browserfingerprinting zur Nutzerverfolgung wurde bereits 2010 von Eckersley [22] nachgewiesen und in weiteren Arbeiten bestätigt [54]. Dass Browserfingerprinting auch tatsächlich zur Nutzerverfolgung eingesetzt wird, untersuchten Nikiforakis und andere bereits 2013 [12, 46]. Maßnahmen gegen Browserfingerprinting sind im Gegensatz zum Browserfingerprinting selbst nur wenig erforscht worden. Da Browserfingerprinting nicht lediglich eine Variante einer bereits erforschten Technik zur Nutzerverfolgung ist, sondern einen grundlegendend neuen Ansatz darstellt, müssen auch neue Ansätze gefunden werden, um einen Schutz aufzubauen. Dementsprechend schützen bekannte Maßnahmen gegen Nutzerverfolgung wie TOR, Proxys oder Privatebrowsing-Modi nicht vor Browserfingerprinting.

Da die Maßnahmen gegen das Browserfingerprinting bisher nicht ausführlich erforscht wurden, untersucht diese Arbeit, ob Maßnahmen existieren, die vor der Nutzerverfolgung mittels Browserfingerprinting schützen können. Exisierende Ansätze sollen zusätzlich beschrieben und auf ihre Wirksamkeit überprüft werden. Erkennbare Lücken in der Erforschung der Abwehr von Browserfingerprinting sollen durch neue Ansätze gefüllt werden.

Die eingehende Untersuchung der Maßnahmen gegen Browserfingerprinting nutzt einerseits der Forschung, da die wissenschaftlich abgesicherten Erkenntnisse diesbezüglich lückenhaft sind und das Verständnis von Browserfingerprinting verbessert wird, indem es aus einer neuen Perspektive betrachtet wird. Aufgrund der Schlüsselstellung des Browserfingerprintings als Technik zur Nutzerverfolgung unterstützt diese Arbeit andererseits auch den Schutz der Privatsphäre von Nutzern im Internet, da sie auf Stärken, Schwächen und

Gefahren von Schutzmaßnahmen hinweist. Da bereits Implementierungen von Maßnahmen gegen Browserfingerprinting existieren und Veröffentlichungen zu Browserfingerprinting regelmäßig von Presseberichten begleitet werden, ist der Bedarf an einer solchen Forschung offensichtlich.

Nicht Teil dieser Arbeit ist das tatsächliche Implementieren von Gegenmaßnahmen, da dies den Umfang einer Masterarbeit überschreiten würde. Ebenfalls nicht Teil der Arbeit sind Maßnahmen gegen Browserfingerprinting speziell zu forensischen, polizeilichen oder geheimdienstlichen Zwecken, obwohl vermutlich viele Ergebnisse auf diese Bereiche übertragen werden könnten.

Um festzustellen, ob es einen wirksamen Schutz vor Browserfingerprinting zur Nutzerverfolgung gibt, wird ein Spektrum an Schutzmaßnahmen auf ihre Effektivität hin untersucht. Es kann so aber leider nicht garantiert werden, dass alle möglichen Maßnahmen gegen Browserfingerprinting betrachtet werden.

Als Basis für eine Untersuchung der Gegenmaßnahmen wird zunächst in Kapitel 2 ein Überblick über das Thema „Browserfingerprinting“ erstellt, indem der aktuelle Wissensstand zu diesem Thema zusammengefasst wird.

Anschließend wird in Kapitel 3 das theoretische Modell vorgestellt, dass von Eckersley genutzt wurde, um das Browserfingerprinting zu untersuchen. Dieses Modell wird erweitert, um das Browserfingerprinting in den folgenden Kapiteln detaillierter modellieren und analysieren zu können.

In Kapitel 4 wird festgelegt, welche Maßnahmen in der weiteren Arbeit untersucht werden. Vor der Auswahl der Maßnahmen werden zunächst die Stärken und Schwächen des Browserfingerprintings eingeschätzt und als Orientierung genutzt, um die Auswahl besser durchführen zu können. Die Auswahl selbst soll möglichst aus allen Maßnahmen gegen Browserfingerprinting erfolgen und die nicht weiter untersuchten Maßnahmen nur kurz betrachtet werden.

Mit Kapitel 5 werden die Maßnahmen gegen Browserfingerprinting analysiert. Diese Analyse geschieht hier zunächst auf einer abstrakten Ebene und soll die Wirksamkeit der untersuchten Maßnahmen unabhängig von den momentanen technischen Verhältnissen untersuchen.

Anschließend werden in Kapitel 6 die in Kapitel 5 gefundenen Ansätze auf ihre Anwendbarkeit untersucht. Als Bezug soll das existierende Ökosystem von Browsern, Browser-Plugins und Fingerprintingskripten dienen und für demonstrierte Techniken soll Beispielcode erstellt werden.

Für weitere Analysen wird in Kapitel 7 eine Simulation der Maßnahmen gegen das Browserfingerprinting durchgeführt. Auf diese Weise kann auch überprüft werden, ob unvorhergesehene Effekte auftreten oder Analysefehler in den vorherigen Schritten begangen wurden.

Abschließend werden in Kapitel 8 die Ergebnisse zusammengefasst und eine Schussfolgerung aus den Analyseergebnissen gezogen. Diese soll eine Empfehlung für den Aufbau eines Schutzes gegen die Nutzerverfolgung mittels Browserfingerprinting enthalten und andeuten, auf welche Weise weitere Forschung stattfinden könnte.

In Anhang A sind detailiertere Tabellen gegeben und in Anhang B ist eine CD mit den in der Arbeit erstellen Quelltexten und zusätzlichen Daten zu finden. Der Inhalt der angehängten CD soll zur bequemeren Verfügbarkeit auch unter `http://thesisgrocki.cybernetic-solutions.de` zu finden sein.

Bei dieser Arbeit ist zu beachten, dass in den Formulierungen Stellung gegen die erfolgreiche Nutzung des Browserfingerprintings und für eine große Anonymität der Nutzer bezogen wird. Soweit nicht anders erwähnt sind „Gegenmaßnahmen“ Maßnahmen, die eine Verfolgung der Nutzer mittels Browserfingerprinting erschweren, ein „negativer“ Effekt ist ein Effekt, der die Anonymität der Nutzer verringert. Dies stellt allerdings keine inhaltliche Positionierung dar, sondern dient lediglich einer verbesserten Lesbareit.

2 Verwandte Arbeiten

Browserfingerprinting kann eingesetzt werden, um Nutzer anhand der Merkmale ihres Browsers voneinander zu unterscheiden oder zu identifizieren. Dies wurde bereits in mehreren Arbeiten untersucht. Hauptsächlich diese Einsatzmöglichkeiten werden in diesem Kapitel dargestellt, da der Fokus dieser Arbeit auf der Nutzerverfolgung liegt.

Um einen Überblick über den Wissensstand zum Browserfingerprinting zu geben, wird zunächst in Abschnitt 2.1 seine Funktionsweise zusammengefasst. In Abschnitt 2.2 werden verwandte Techniken dargestellt und in Abschnitt 2.3 wird die Nutzung des Browserfingerprintings vorgestellt. Anschließend wird in Abschnitt 2.4 das aktive und passive Fingerprinting beschrieben und auf Merkmale eingegangen, die für das Browserfingerprinting genutzt werden. In Abschnitt 2.5 wird kurz die Rechtslage angesprochen und in Abschnitt 2.6 bekannte Maßnahmen gegen Fingerprinting dargestellt. Abschließend wird das Kapitel in Abschnitt 2.7 zusammengefasst.

2.1. Funktionsweise des Browserfingerprintings

Browserfingerprinting ist eine Spezialisierung des Prinzips des Fingerprintings von Programmen. Es hat im Gegensatz zu diesem aber nicht das Ziel, Programmtypen, sondern Browserinstallationen voneinander zu unterscheiden [22].

Das Fingerprinting von Programmen im Allgemeinen ist eine schon länger bekannte Technik, die genutzt wird, um gezielte Angriffe auf Computersysteme durchzuführen [29]. Dabei werden scheinbar unwichtige Eigenheiten von Programmen gemessen und zu einem Fingerprint zusammengefasst. Dieser Fingerprint kann genutzt werden, um in Datensammlungen Informationen über die eingesetzten Programme nachschlagen zu können. Informationen, die auf diese Art und Weise gewonnen werden können, betreffen zum Beispiel den Webservertyp und die genaue Programmversion [37]. Werden möglichst unveränderliche und nicht unterdrückbare Eigenheiten genutzt und kombiniert, ist der Fingerprint wie der namensgebende menschliche Fingerabdruck schwer fälschbar und wird auch unfreiwillig abgegeben.

Beispielsweise kann der Fingerprint des TCP/IP-Stacks eines Servers erhoben werden, um sein Betriebssystem zu bestimmen. Eigenheiten, die dazu genutzt werden können, sind unter anderem die Sortierung von Optionen, Muster in den Sequenznummern und Maximalgrößen von Paketen. Diese Eigenheiten werden vom Analysewerkzeug Nmap abgefragt, indem dem Server Analysepakete gesendet werden und dessen Antwort analysiert wird [51].

Wird der Fingerprint eines Browsers erhoben, werden ähnliche Techniken genutzt wie beim Erheben des Fingerprints eines SSH-Servers. Browserfingerprinting kann daher auch genutzt werden, um gezielte Angriffe auf Browser durchzuführen [25, 35, 57]. Ein Browser unterscheidet sich allerdings von Programmen wie SSH-Servern in drei wichtigen Punkten.

- Ein Browser wird auf einem Clientsystem ausgeführt und ist somit zur Verfolgung von Nutzern brauchbar.
- Ein Browser ist ein komplexeres System als ein SSH-Server, da der Browser wesentlich mehr Eigenheiten, Einstellungs-, Erweiterungs- und Interaktionsmöglichkeiten hat.
- Ein Browser interpretiert komplexe Seiten oder führt Skripte aus.

Deshalb lassen sich beim Fingerprinting eines Browsers besonders viele Informationen gewinnen.

Stehen genug Informationen zur Verfügung, um eine Browserinstallation von allen anderen Browserinstallationen zu unterscheiden, ist ihr Fingerprint einzigartig und die Browserinstallation kann wiedererkannt werden. Kann angenommen werden, dass die Fingerprints aller Browserinstallationen einzigartig sind, kann der Fingerprint von Browserinstallationen als Identifizierer für diese genutzt werden. Kann zusätzlich angenommen werden, dass jeder Nutzer exakt eine Browserinstallationen nutzt, kann deren Fingerprint sogar als Identifizierer für die Nutzer dienen.

Da diese Informationen nur genutzt werden, um Browserinstallationen zu unterscheiden, sind sie auch dann nützlich, wenn ihr Inhalt ansonsten nicht relevant oder nutzlos ist. Beispielsweise ist eine nutzerdefinierte Schriftfarbe für die Vorbereitung eines Angriffes irrelevant, kann aber zur Unterscheidung von Installationen genutzt werden.

Die Möglichkeit, Browserfingerprinting zur Identifizierung von Browserinstallationen zu nutzen, wurde 2009 von Mayer vorgestellt [38] und 2010 in einem Paper von Eckersley weiter untersucht [22]. Eckersley hat ein Modell des Browserfingerprintings formuliert, das in Kapitel 3 vorgestellt wird, und eine Studie durchgeführt, um den Informationsgehalt von Browserfingerprints abschätzen zu können. In dieser Studie wurden 470 161 Browserfingerprints gesammelt, von denen 83,6 % einzigartig waren, obwohl die genutzte Fingerprintingmethode nicht optimiert war. Die in der Studie aufgebaute Zufallsverteilung hatte eine Entropie von 18,6 Bit, wodurch zu erwarten ist, dass ein zufälliger Fingerprint mit nur einem von 286 777 Fingerprints übereinstimmt. In der Diplomarbeit von Tillmann wurde 2013 bestätigt, dass die meisten Browserinstallationen mittels Browserfingerprinting identifizierbar sind [54].

In den Arbeiten von Eckersley und Tillmann wurde darauf hingewiesen, dass die Stabilität der Browserfingerprints wichtig ist, sollen Browserinstallationen nicht nur unterschieden, sondern auch wiedererkannt werden. Instabil sind Browserfingerprints, wenn sich die Browserinstallation zum Beispiel durch Updates oder Neukonfigurationen ändert. Um trotz dieser Änderungen Browserinstallationen identifizieren zu können, wurden von Eckersley eine Methode demonstriert, die Änderungen des Fingerprints entdecken und ausgleichen kann. Als einer dieser komplexeren Methoden, Fingerprints zu unterscheiden, wurde von Flood sogar maschinelles Lernen untersucht [26].

2.2. Mit dem Browserfingerprinting verwandte Techniken

Es gibt Techniken, die mit dem Browserfingerprinting verwandt sind, aber dem Browserfingerprinting nicht exakt entsprechen. Um Missverständnisse und Unklarheiten zu vermeiden, werden diese Techniken dargestellt, ohne dass in der weiteren Arbeit näher auf sie eingegangen wird.

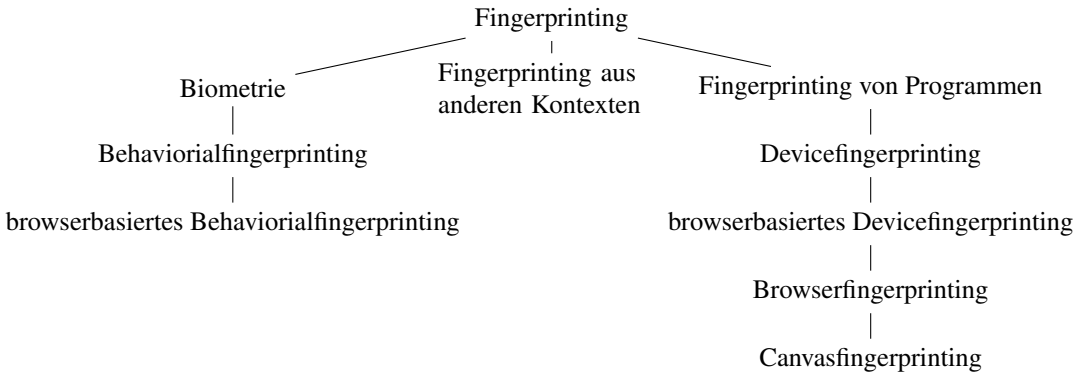


Abbildung 2.1.: Hierarchische Darstellung der Fingerprintingvarianten

Die Biometrie ist für das Fingerprinting namensgebend. Bei dieser werden Merkmale eines Menschen erhoben, um sie später erneut zu messen und den Menschen wiederzuerkennen [31]. Dafür genutzte Merkmale sind beispielsweise Muster der Haut in den Fingerspitzen oder in der Iris von Menschen. Diese Merkmale verändern sich nicht selbstständig und sind nur schwer zu manipulieren.

Das Behavioralfingerprinting hat nicht die Analyse einer Browserinstallationen zum Ziel, sondern versucht Menschen zu analysieren. Dies geschieht, indem Merkmale des Nutzers selbst und nicht von dessen technischen Systemen erhoben werden. Ein Beispiel dafür ist das Fingerprinten seines Tippmusters oder seiner Mausbewegungen [14, 15]. Um diesen Fingerprint des Nutzers zu messen, können wie im **browserbasierten Behavioralfingerprinting** im Browser ablaufende Analyseskripte genutzt werden. Das Behavioralfingerprinting ist, obwohl es Browser zum Erheben der Fingerprints nutzen kann, der Biometrie und nicht dem Devicefingerprinting zuzurechnen.

Das Fingerprinting von Programmen im Allgemeinen nutzt die Messung von Verhaltenweisen von Programmen, um auf Informationen über diese zu schließen. Diese Informationen reichen im Normalfall nur für die Unterscheidung von Programmen, der Programmversion oder bestimmten Konfigurationen aus. Einsatzzwecke sind das Vorbereiten von gezielten Angriffen [25, 35, 57] und die Erkennung von bösartiger Software [17].

Das Browserfingerprinting ist ein Spezialfall des Fingerprintings von Programmen. Dabei können sehr viele Informationen über den Nutzer erhoben werden, was es erlaubt, nicht nur Browsertypen, sondern auch konkrete Browserinstallationen zu unterscheiden und zu identifizieren. Über die Browserinstallation sollen letztendlich die Nutzer des Browsers unterschieden und identifiziert werden. Diese Form des Fingerprintings wird in dieser Arbeit betrachtet.

Das Devicefingerprinting ist eine direkte Erweiterung des Browserfingerprintings. Dabei wird nicht nur der Fingerprint des Browsers erhoben, sondern möglichst viele Informationen über den eingesetzten Computer, dessen Betriebssystem und dessen Programme gesammelt. Dazu werden zusätzliche Quellen wie der TCP/IP-Stack in den Fingerprint miteinbezogen [54].

Das Devicefingerprinting ist sehr nahe mit dem Browserfingerprinting verwandt. So enthalten Informationen, die beim Browserfingerprinting gesammelt werden, üblicherweise Informationen über das genutzte Betriebssystem oder die Hardware des Nutzers. Dadurch wird Grenze zum Devicefingerprinting bereits überschritten, obwohl nur Informationen aus dem Browser ausgelesen werden. Für diesen Spezialfall gibt es den Begriff des „browserbasierten Devicefingerprintings“. Da das browserbasierte Devicefingerprinting in den genutzten Quellen mit dem Browserfingerprinting begrifflich vermischt war, kann eine klare Trennung auch in dieser Arbeit nicht gewährleistet werden.

Es gibt mit dem **Cross-Browserfingerprinting** ein reines Devicefingerprinting, dass zwar über einen Browser geschehen soll, aber keine browserspezifischen Merkmale in den Fingerprint einbezieht [18].

Das Canvasfingerprinting ist ein Spezialfall des Browserfingerprintings. Dabei werden Merkmale über das Zeichnen und Auslesen von Schriften und Objekten in HTML5-Canvas erhoben [41]. Durch diese Methode lassen sich viele Informationen gewinnen und das Canvasfingerprinting ist eine oft eingesetzte Fingerprintingtechnik [46].

Das Fingerprinting aus anderen Kontexten wie der Geologie [48], Biologie [45], Medizin [21], Forensik [28] oder auch der Klimaforschung [58] funktioniert größtenteils ebenfalls nach dem Prinzip des Wiedererkennens von Merkmalen. Die dabei genutzten Methoden und verfolgten Ziele sind dabei allerdings sehr unterschiedlich und werden in dieser Übersicht nicht weiter behandelt, da dies den Umfang dieser Arbeit überschreiten würde.

2.3. Nutzung des Browserfingerprintings

Die tatsächliche Nutzung von Browserfingerprinting durch die Industrie wurde in der von Nikiforakis und anderen durchgeführten Studie „Cookieless Monster: Exploring the Ecosystem of Web-based Device Fingerprinting“ untersucht [46]. Bei dieser Studie wurde ein Crawler genutzt, um Webseiten zu finden, die Fingerprinting einsetzen. Diese Studie wurde im Jahr 2013 durchgeführt und es wurden auf 40 der 10 000 nach dem Alexarating [1] am meisten besuchten Webseiten Fingerprintingskripte gefunden. Mit derselben Technik wurden ebenfalls 2013 in 404 der 1 000 000 nach dem Alexarating am meisten besuchten Webseiten auf Javascript basierendes Fingerprinting und in 97 der 10 000 nach dem Alexarating am meisten besuchten Webseiten flashbasiertes Fingerprinting festgestellt [12]. Diese Zahl ist aber eine untere Grenze, da mit dem eingesetzte Crawler nur bestimmte Browserfingerprintingskripte erkannt werden konnten und nicht alle Unterseiten überprüft wurden. In einer neueren Studie [13] wurde festgestellt, dass über 5 % der 100 000 nach dem Alexarating am meisten besuchten Webseiten Canvasfingerprinting, ein Spezialfall des Browserfingerprintings, einsetzen.

Die Frage, wofür die gefundenen Browserfingerprintingskripte genutzt werden, konnte allerdings nicht abschließend geklärt werden. Untersucht wurde aber, auf welcher Art Webseite Browserfingerprintingskripte gefunden wurden. Dabei wurden Spam, Pornografie und bösartige Seiten als stärkste Kategorien eingeschätzt.

Bekannte Nutzungsmöglichkeiten dieser Form der Nutzerwiederidentifizierung sind

- der Ersatz von Cookies,
- das zusätzliche Absichern von Sessions,
- die Bindung von Nutzern an Services,
- das Identifizieren von bösartigen Nutzern und
- die Polizei- oder Geheimdienstarbeit.

Diese Nutzungsmöglichkeiten und ihre prinzipiellen Funktionsweisen werden in den nächsten Abschnitten vorgestellt.

2.3.1. Ersatz von Cookies

Es gibt zwar verschiedene Arten, Daten in einem Browser abzulegen, die Cookies direkt ersetzen können [40, 50] und sogar wie bei Evercookie gesammelt ansprechbar sind [33], aber Browserfingerprinting wird trotzdem genutzt, um Cookies zu ersetzen [46]. Im Gegensatz zu Cookies wird dabei das Ablegen von

Daten im Browser vermieden. Deshalb hat der Nutzer bei Fingerprints keine Möglichkeit, diese wie Cookies zu löschen, und die Wiederidentifizierung über Browserfingerprints hinterlässt nicht zwingend Spuren im Browser. Wenn Privatmodi von Browsern lediglich verhindern, dass Daten im Browser abgelegt werden, schützen sie nicht vor Browserfingerprinting.

Diese Möglichkeit, Cookies zu ersetzten, kann wie Cookies selbst für verschiedene Zwecke eingesetzt werden. Von großer Bedeutung ist die Erstellung von Nutzerprofilen zu Verkaufs- oder Werbezwecken.

Browserfingerprinting kann auch in Kombination mit Cookies genutzt werden, um Nutzer trotz des Löschens aller Cookies wiederzuerkennen und die Cookies wiederherzustellen. Eckersley gibt an [22], dass für eine solche Regeneration von Cookies 15-20 Bit Informationen ausreichen, wenn weitere Informationen wie IP-Adressen hinzugezogen werden. Es wird eine Zunahme der Nutzung dieser Technik erwartet, da Tracking-Unternehmen Cookies nutzen, Cookies aber zunehmend als Problem behandelt und vermieden werden. Ein ähnliches Regenerationssystem wird bereits für die Regeneration von Cookies über Flashcookies genutzt [13].

Eine zur Cookieregeneration ähnliche Technik wurde 2012 untersucht [59]. Bei dieser wurden lediglich der Useragent und die IP gemeinsam zum Identifizieren der Nutzer ausgewertet. Kombiniert wurden diese beiden Eigenschaften auf 20,29 Bit Entropie geschätzt, was höher ist als die ursprüngliche Schätzung von Eckerlsey für das Browserfingerprinting insgesamt [22]. Diese Studie betraf die Nutzer von Bing und Hotmail, wurde über den Zeitraum von mehreren Monaten durchgeführt und hat somit einen Umfang von mehreren Hundert Millionen Nutzern.

2.3.2. Nutzeraktionen beschränken

Auf Datingseiten oder Umfrageseiten existiert ein Interesse, Mehrfachregistrierungen komplett zu verhindern, da im Fall von mehreren Accounts für einen Nutzer Missbrauch vermutet werden kann [12, 46]. Auch Zeitungen und Magazine könnten Nutzeraktionen beschränken, um Nutzer eine gewisse Anzahl von Artikeln gratis lesen zu lassen und erst danach eine Gebühr zu verlangen [12].

Sollen bestimmte Aktionen für jeden Nutzer auf eine oder wenige Nutzungen eingeschränkt werden, kann versucht werden, dies über den Fingerprint seines Browsers durchzusetzen. Dazu können einfach die Fingerprints der Nutzer gespeichert werden und der Fingerprint nach Beenden der Aktion als verbraucht markiert oder ein dem Fingerprint zugeordneter Zähler inkrementiert werden. Hat ein Nutzer einen bereits verbrauchten Fingerprint, kann die angeforderte Aktion verweigert werden.

2.3.3. Zusätzliches Absichern von Sessions

Sollen eine Reihe von Aktionen ausgeführt werden, die durch ein Login geschützt sind, wird oft eine Session genutzt, um das ursprüngliche Passwort nur einmal zu übertragen. Um den Nutzer wiederzuerkennen, wird zu Beginn der Session ein Geheimnis generiert, dass sich Client und Server teilen. Durch Vorlage dieses Geheimnisses kann sich der Client gegenüber dem Server ausweisen und auf die mit der Session verknüpften Daten zugreifen.

Beim Sessionhijacking erfährt oder errät ein Angreifer dieses Geheimnis und nutzt es, um sich gegenüber dem Server auszuweisen. Um Nutzer gegen einen solchen Angriff zu schützen, kann der Server zusätzlich den Fingerprint des Clients speichern und überprüfen [55]. Ändert sich der Fingerprint auffällig stark, kann angenommen werden, dass eine unzulässige Übertragung der Session stattgefunden hat und die Session kann geschlossen werden.

Die Sicherheit eines solchen Systems beruht auf den Annahmen, dass der Fingerprint dem Angreifer unbekannt ist oder der Angreifer den Fingerprint nicht reproduzieren kann und dass der Angreifer nicht zufällig den Fingerprint des Opfers hat. Da nicht erwiesen ist, dass diese Annahmen immer zutreffen, kann Sessionhijacking nicht ausgeschlossen werden, aber immerhin wird die Komplexität eines solchen Angriffs durch das Überprüfen der Fingerprints erhöht.

Gelingt es einem Angreifer, den legitimen Fingerprint zu fälschen, versagt dieser Schutz und der Angreifer kann mit seinem Angriff fortfahren. Ändert sich der Fingerprint eines legitimen Nutzers zu stark, wird die Session unberechtigt geschlossen. Dadurch ist der Nutzer allerdings lediglich gezwungen sich erneut einzuloggen, wenn er die Webseite weiter nutzen will.

2.3.4. Accounts an Nutzer binden

Soll die Nutzung eines Accounts auf einen oder wenige Nutzer beschränkt werden, kann versucht werden, dies über deren Fingerprint ihres Browsers durchzusetzen [46]. Dazu speichert der Server einen oder mehrere Fingerprints, die mit den Accounts verknüpft sind, und kann nun bei Login-Versuchen überprüfen, ob der Fingerprint des Browsers mit den verknüpften Fingerprints übereinstimmt. Um Fingerprints mit einem Account zu verknüpfen, kann zum Beispiel der Fingerprint des ersten Logins gespeichert oder zusätzliche Authentifikationsmethoden genutzt werden. Ist der Fingerprint nicht mit dem Account verknüpft, kann das Login verweigert, zusätzliche Authentifikationsmethoden gefordert oder ein Einbrucherkennungssystem benachrichtigt werden.

Dies kann beispielsweise im Hochsicherheitsbereich genutzt werden, bei dem angenommen werden kann, dass der Nutzer seinen Account nur von einem einzigen Gerät nutzt. Existieren zusätzliche Authentifikationsmethoden, kann auch auf Änderungen des Fingerprints von legitimen Nutzern eingegangen werden. Ein Beispiel für einen solchen Hochsicherheitsbereich sind Onlinebezahlsysteme [46] oder Banken. Aber auch in Bereichen mit niedrigen Sicherheitsanforderungen, wie dem bezahlten Mediastreaming, kann eine Bindung von Accounts an Browser angewendet werden. Dort haben die Nutzer ein Interesse an der Mehrfachnutzung eines Accounts, die Anbieter hingegen daran, dass möglichst viele Accounts erstellt werden. Um zu verhindern, dass sich eine große Menge von Nutzern unter einem Account einloggt, und um sicherzustellen, dass sich ein Nutzer aber trotzdem von mehreren Geräten einloggen kann, wird einer kleine Menge von Fingerprints an den Account gebunden.

Genau wie beim zusätzlichen Absichern von Sessions ist die vollständige Sicherheit dieses Mechanismus nicht erwiesen, da hier dieselben Annahmen zugrunde liegen. Eine fehlerhafte Erkennung eines legitimen Nutzers als Angreifer kann hier allerdings schwerwiegendere Konsequenzen für diesen Nutzer haben, wenn diesem zum Beispiel das Login dauerhaft verweigert wird.

2.3.5. Das Wiedererkennen von bösartigen Nutzern

Bösartige Nutzer wie Hacker, Scammer, Spammer oder Trolle hinterlassen normalerweise Spuren wie IP-Adressen auf dem Server. Benutzt der Angreifer einen Browser, kann auch der Browserfingerprint eine der hinterlassenen Spuren sein.

Diese Browserfingerprints können gesammelt, in Datenbanken zusammengefasst und eventuell genutzt werden, um bösartige Nutzer anhand dieser Datenbank zu erkennen [12]. Solchen Nutzern kann zum Beispiel die Nutzung der Seite verweigert werden oder es kann ein Einbruchserkennungssystem über deren Besuch benachrichtigt werden. Die Datenbanken könnten auch zwischen mehreren Parteien geteilt und ausgetauscht werden, wenn der zum Erstellen des Fingerprints genutzte Fingerprintingalgorithmus allen Seiten bekannt ist.

Dieses System versagt, wenn der Angreifer seinen Fingerprint verändert und somit nicht mehr erkennbar ist. Teilt sich ein Angreifer einen Fingerprint mit einem Unbeteiligten, wird dieser fälschlicherweise ebenfalls in die Datenbank von Angreifern aufgenommen und entsprechend behandelt. Sollen diese Fehler möglichst vermieden werden, benötigt dieses System eine möglichst große Entropie, besonders wenn die Datenbank gemeinsam genutzt und die Menge der betroffenen Nutzer so vergrößert wird.

2.3.6. Polizei- und Geheimdienstarbeit

Browserinstallationen identifizieren oder auch nur unterscheiden zu können, wäre für Polizei- und Geheimdienstarbeit nützlich. Dadurch könnten zum Beispiel Webseitenbesuche mit im Nachhinein beschlagnahm-

ten Betriebssysteminstallationen in Verbindung gebracht werden. Mehrfachtäter könnten über Browserfingerprints erkannt und Taten in Zusammenhang gestellt werden. Die wohl mächtigste Nutzungsmöglichkeit ist das Deanonymisieren der Nutzer von Anonymisierungsnetzwerken wie TOR. Dies kann zum Beispiel geschehen, indem der Browserfingerprint nach Verlassen des Anonymisierungsnetzwerkes oder in einer Situation gemessen wird, bei der der Zugriff dem Nutzer zugeordnet werden kann.

Beispielsweise gibt es Medienberichte, nachdem im Gerichtsverfahren um die Drogenhandelsseite „Silkroad 2.0“, ein Moderator, der TOR zur Anonymisierung nutzte, mittels Browserfingerprinting deanonymisiert wurde. Konkret wurden durch die Kombination aus einer Betaversion des Browsers Chrome und einer veralteten Version des Betriebssystems „OS X“ zwei Zugriffe auf unterschiedliche Seiten in Verbindung gebracht. Diese Verküpfung erlaubte im weiteren eine Emailadresse und somit eine bestimmte Person mit Administrationszugriffen auf die Infrastruktur von „Silkroad 2.0“ in Verbindung zu bringen [44].

Da angenommen wird, dass Polizei- oder Geheimdienste Browserfingerprinting nutzen oder zumindest die Möglichkeit dazu haben, bietet das Anonymisierungswerkzeug TOR mit dem TOR-Browser einen gegen Browserfingerprinting abgehärteten Browser. Die für die Polizeiarbeit relevante Frage der Beweiskraft von Browserfingerprints ist dabei noch ungeklärt.

2.4. Passives und aktives Fingerprinting

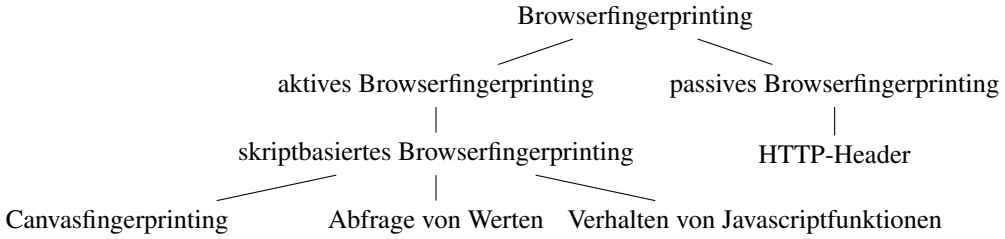


Abbildung 2.2.: Hierarchische Darstellung der Formen des Browserfingerprintings

Browserfingerprinting kann anhand der genutzten Methoden zum Erheben von Fingerprints in aktives und passives Browserfingerprinting unterschieden werden. Für das Fingerprinting von Programmen existiert bereits eine Definition für aktives und passives Browserfingerprinting [52]. In dieser Definition ist Fingerprinting passiv, wenn die Kommunikation zwischen Client und Server nicht verändert wird, und ansonsten aktiv. Für das Browserfingerprinting wird eine analoge Unterscheidung genutzt [24, 54].

Um diese beiden Browserfingerprintingtypen zu darzustellen, wird zunächst in Abschnitt 2.4.1 das passive und in Abschnitt 2.4.2 das aktive Browserfingerprinting beschrieben. Anschließend werden sie in Abschnitt 2.4.3 gegenübergestellt.

2.4.1. Passives Fingerprinting

Beim passiven Fingerprinting werden Informationen in den Fingerprint einbezogen, die während der normalen Nutzung eines Dienstes wie einer Webseite auftreten. Typische Werte, die mittels passiven Fingerprintings ausgelesen werden, sind in Tabelle 2.1 aufgelistet.

Alle diese Merkmale können ebenfalls durch aktives Browserfingerprinting ausgelesen werden, weswegen bei einer passiven Analyse prinzipiell weniger Informationen anfallen als bei einer aktiven. Die Menge der Entropie des passiven Fingerprintings kann nur geschätzt werden, aber als Untergrenze für die Entropie des Useragents liegen Schätzungen von 6,31 Bit bei Broenink [20], 8,10 Bit bei Boda [18], 10 Bit bei Eckersley [22] und 11,59 Bit bei Yen und anderen vor [56].

Passives Browserfingerprinting findet per Definition rein serverseitig statt und hinterlässt keine Spuren im anaysierten Browser oder Traffic, weshalb passives Browserfingerprinting vom Client nicht ohne Weiteres

Accept	Vom Browser unterstützte und erwünschte Datenformate
Accept-Language	Sprache, in der die Webseite generiert werden soll
Accept-Encoding	Vom Browser unterstützte Codecs
Accept-Charset	Unterstützte und erwünschte Zeichenkodierungen
Connection	Gewünschte Art der weiteren Verbindung
Useragent	Kurze Selbstbeschreibung oder Name des Browsers

Tabelle 2.1.: Fingerprintingmerkmale, die beim passiven Browserfingerprinting erhoben werden [20].

erkannt werden kann. Aus diesem Grund konnte in Nikiforakis Studie auch nur das aktive Browserfingerprinting erkannt und untersucht werden [46].

Weiterhin muss das passive Browserfingerprinting bedeuten, dass nicht nur der Server oder Client, sondern auch dritte Parteien Fingerprints von Browsern erheben können. Dies ist möglich, wenn Drittparteien Kenntnis über die unverschlüsselte Kommunikation zwischen Client und Server erlangen, da diese für passives Browserfingerprinting den Traffic nicht verändern müssen. Zum Browserfingerprinting durch Drittparteien konnten leider keine Untersuchungen gefunden werden, im Hinblick auf einen Geheimdienst als Drittpartei ist diese Möglichkeit aber relevant.

2.4.2. Aktives Fingerprinting

Beim aktiven Fingerprinting wird die Kommunikation zwischen Server und Client so gestaltet, dass der Client mehr Informationen als üblich über sich preisgibt. Beispielsweise kann die Kommunikation so gestaltet werden, dass das zu fingerprintende System einfach nach weiteren Informationen wie etwa unterstützten Aktionen gefragt wird [32]. Weitere Eigenheiten können ausgeforscht werden, indem Obergrenzen von beispielsweise Farbwerten getestet werden [19] oder Syntax verwendet wird, die undefiniertes oder nicht einheitliches Verhalten aufweist [10].

Moderne Browser bieten normalerweise die Möglichkeit, Skripte im Browser ausführen zu lassen. Diese bieten sehr weitreichende Möglichkeiten, Informationen über die Browserinstallation zu gewinnen. Einfache Methoden fragen diese Informationen einfach bei dem Browser an. Komplexere Methoden sind zum Beispiel das pixelgenaue Analysieren von Schriftdarstellungen [41] und das Benchmarken von Javascriptfunktionen, um auf die genutzte Javascriptengine zu schließen [42, 43]. Für das Zurücksenden der gesammelten Daten kann auf dafür vorgesehenen Funktionen wie AJAX [22] zurückgegriffen werden, aber auch andere verstecktere Kanäle könnten genutzt werden [46].

Einer der bekanntesten von Browsern unterstützten Skriptsprachen ist Javascript. Alleine diese war im Jahr 2010 von 99% der Nutzer aktiviert [12] und es gibt weitere Sprachen wie Flash, Silverlight oder Java, die zum Erstellen von Fingerprints geeignet sind. Beim Browserfingerprinting kann also davon ausgegangen werden, dass es möglich ist, solche Skripte einzusetzen, ohne allzu viele Nutzer auszulassen.

Statt Skriptsprachen wie Javascript oder Flash können auch andere Methoden zum Browserfingerprinting genutzt werden. Beispielweise können spezielle Browsererweiterungen für umfangreichere Analysen genutzt werden, wenn der Nutzer dazu gebracht werden kann, diese zu installieren [46]. Ein aktives Vorgehen zur Bestimmung von Browsertypen und Browserversionen wird genutzt, um Webseiten auf Browser angepasst darzustellen. Dabei werden Unterschiede beim Interpretieren von CSS und HTML genutzt, um spezielle Designs zu aktivieren [2].

Browsermerkmale, von denen bekannt ist, dass sie für aktives Browserfingerprinting verwendet werden können, sind in Tabelle 2.2 aufgelistet.

Browserfingerprinting mittels Skripten steht, wohl aufgrund der Menge an erhebbaren Informationen und der breiten Verfügbarkeit von Skripten, im Fokus von Industrie und Forschung. Das Interesse der Industrie ist daran erkennbar, dass diese in vielen Browserfingerprintern sehr stark Javascript, Flash oder Silverlight nutzen [46]. Das Interesse der Forschung am skriptbasierten Browserfingerprinting ist ebenfalls groß. Beispielsweise wurde in der Diplomarbeit von Tillmann ein Browserfingerprint eingesetzt, dass ohne aktivierte

Appname [54]	Eine kurze Selbstbeschreibung oder Name des Browsers
Javascriptversion [20]	Die Javascriptversion
Hardwareinformationen [20]	z.B. Prozessorarchitektur oder Bildschirmauflösung
Charset [20]	Vom Browser unterstützte Zeichencodierung
Sprache [20]	Die Spracheinstellung des Browsers
Cookieunterstützung [20]	Die Unterstützung von Cookies und Supercookies
Unterstützung von Skripten [20]	Die Unterstützung von Sprachen wie Java, Flash oder Silverlight
Zeitzone [20]	Die Zeitzoneneinstellung des Browsers
Farben [54]	Die eingestellten Farben für Buttons, aktivierte Links oder Ähnliches
Plugins [22, 54]	Die installierten Plugins
Mimetypes [54]	Dem Browser bekannte Mimetypes
Schriften [22, 54]	Die dem Browser zur Verfügung stehenden Schriften
Schriftdarstellungen [6, 41]	Analyse der Darstellungsweisen von Schriften durch den Browser
Benchmarks [42, 43]	Die Dauer von bestimmten Aktionen
Privatsphäreeinstellungen [46]	Einstellungen wie der Do-Not-Track-Header
Konstanten [46]	Mathematische Konstanten des Browsers
Spezialfähigkeiten [46]	Fähigkeiten und Funktionen, die nur manche Browser haben

Tabelle 2.2.: Fingerprintingmerkmale, die beim aktiven Browserfingerprinting erhoben werden.

Skripte noch nicht einmal lauffähig war und somit nicht skriptbasiertes Browserfingerprinting vernachlässigt wurde [54]. Ebenso konnte der Crawler, mit dem in der Studie „Cookieless Monster: Exploring the Ecosystem of Web-based Device Fingerprinting“ nach Browserfingerprinting gesucht wurde, nur auf Skripten basierendes Browserfingerprinting erkennen [46].

2.4.3. Gegenüberstellung von aktiven und passiven Browserfingerprinting

Das aktive Browserfingerprinting, das in Abschnitt 2.4.2 beschrieben wurde, unterscheidet sich nicht nur in der genutzten Technik vom passiven Browserfingerprinting, das in Abschnitt 2.4.1 beschrieben wurde.

Aktives Browserfingerprinting erlaubt es einerseits, wesentlich detailiertere Analysemethoden anzuwenden und somit mehr Informationen über die Browser zu erheben. Die Analysen des passiven Browserfingerprintings können dabei immer vom aktiven Browserfingerprinting genutzt werden. Javascript und Flash werden beim aktiven Browserfingerprinting oft eingesetzt, es gibt aber auch andere Techniken für eine aktive Analyse von Browserfingerprinting.

Andererseits ist das aktive Browserfingerprinting eine Technik, die Spuren im Traffic hinterlässt. Dies ermöglicht, aktives Browserfingerprinting nachzuweisen und gezielt nach Browserfingerprintingskripten zu suchen. Passives Browserfingerprinting ist im Gegensatz dazu nicht direkt nachzuweisen und kann auch durch Drittparteien geschehen, wenn diese den unverschlüsseltenTraffic erbeuten können.

2.5. Rechtliches

Browserfingerprinting zur Nutzerverfolgung wird über verschiedene eingetragene Firmen und Werbenetzwerke kommerzialisiert. Daran ist erkennbar, dass Browserfingerprinting nicht in einer komplett illegalen Umgebung stattfindet, und somit ist auch die detaillierte Rechtslage interessant. Bei den verwandten Cookies bieten Regulierungen wie der Do-Not-Track-Header allerdings nur eine geringe Schutzwirkung [13].

Ein zentraler Faktor in Tillmanns rechtlicher Bewertung [54] ist die Frage, ob beim Browserfingerprinting personenbeziehbare oder pseudonymisierte Daten vorliegen. Liegen personenbezogene oder personenbeziehbare Daten, also Daten, die mit Personen in Zusammenhang gebracht werden können, vor, fallen diese in Deutschland unter das Bundesdatenschutzgesetz. Dieser Schutz kann die explizite Einwilligung des Nutzers vor der Datensammlung oder das Verbot, Datenbestände zusammenzuführen, beinhalten. Die Wichtig-

keit dieses Punktes wird dadurch bestärkt, dass auch Firmen unabhängig von deutschem Recht beteuern, keine personenbezogene Informationen zu speichern, obwohl sie Browserfingerprinting einsetzen [12]. Es wird auch befürchtet, dass diese Argumentation ausgeweitet werden könnte, um Datenschutzgesetze zu umgehen. Dazu könnten Daten mit einem Browserfingerprint statt mit einem Namen oder einer Adresse verknüpft gespeichert werden.

In einer Untersuchung zum Browserfingerprinting wurde festgestellt, dass die Nutzer üblicherweise nicht über die beim Browserfingerprinting vorgenommene Datensammlung informiert werden [12]. Beim Browserfingerprinting, das durch Drittparteien ausgeführt wird, bereitet auch die Frage Probleme, wer informieren sollte. Der Nutzer kann grundsätzlich auf der einbindenden oder eingebundenen Seite informiert werden. Da das Einbinden von Drittparteien für den normalen Nutzer oft nicht erkennbar ist, werden die Nutzer einen solchen Hinweis auf der einbindenden Seite suchen. Die einbindende Seite hat aber nicht notwendigerweise das Wissen darüber, dass Browserfingerprinting eingesetzt wird.

Für Cookies gibt es in der EU Vorschriften, die die Zustimmung des Nutzers zum Speichern von Daten in Browsern erfordern [39]. Eine solche Zustimmung ist allerdings beim Browserfingerprinting nicht vorgeschrieben. Wird Browserfingerprinting verwendet, um Cookies zu regenerieren, muss davon ausgegangen werden, dass der Browser des Nutzers keine Cookies erlaubt oder Cookies gelöscht werden. Der Nutzer drückt somit seinen Willen aus, nicht wiedererkannt zu werden, und eine Nutzung zu diesem Zweck kann illegal sein. Das Wiederherstellen von Cookies hat auch bereits zu Gerichtsverfahren mit einer Einigung auf eine Zahlung in der Höhe von 500 000 $ geführt [13].

Für die rechtliche Betrachtung ist auch interessant, dass das Regenerieren von Cookies mittels Browserfingerprinting seit 2011 in den USA zum Patent angemeldet ist [30]. Würde dieses Patent anerkannt und verteidigt werden, könnte dies den Einsatz von Browserfingerprinting erheblich verkomplizieren, auch wenn der Nutzer informiert und einverstanden wäre. Welchen Effekt ein solches Patent hätte, kann an dieser Stelle aber nicht eingeschätzt werden.

2.6. Bekannte Gegenmaßnahmen

Zur Abwehr von Browserfingerprinting und dem Fingerprinting von Programmen existieren mehrere Vorschläge zur Vorgehensweise. Diese Gegenmaßnahmen wurden, um die Übersichtlichkeit und Referenzierbarkeit zu erhöhen, in Kategorien unterteilt. Es besteht dabei allerdings kein Anspruch auf Vollständigkeit. Als Quellen für diese Gegenmaßnahmen wurden Veröffentlichungen, Diskussionen, Privatsphären-Plugins und das Tor-Browser-Bundle herangezogen.

2.6.1. Selbstbeschreibungen einschränken

Eine intuitive Vorgehensweise, um die vom Browser preisgegebene Selbstinformation zu reduzieren, ist, die Menge der vom Browser herausgegebenen Daten klein zu halten. Die Geheimhaltung von Informationen wie Sprachpräferenzen oder installierten Schriften verhindert allerdings die Nutzung von Features, die auf diesen Informationen basieren.

Solche Einschränkungen führen allerdings nicht zwingend zum Erfolg, sondern können auch den eigentlichen Zielen entgegenwirken [22]. Dies lässt sich darauf zurückführen, dass das Geheimhalten einer Information selbst wieder eine Information ist, die zur Unterscheidung von Browserinstallationen verwendet werden kann. Impliziert das Geheimhalten von Information mehr Information als die geheim gehaltene Information, gibt der Browser also insgesamt mehr Information über sich preis. Es kann sogar zur Identifizierbarkeit einer Browserinstallation ausreichen, wenn eine Information nur von dieser geheim gehalten wird.

Es existieren verschiedene Ansätze, um die vom Browser preisgegebene Daten zu verringern. Zwei dieser Wege werden hier als Beispiel dargestellt.

Eckersley hat vorgeschlagen, dass Browser und Add-ons statt Mikroversionsnummern nur grobe Versionsangaben preisgeben könnten [22]. Diese Versionsnummern werden teilweise mit spezifischen Builds

oder Revisions angegeben. Da Versionsangaben von Browser- oder Plugin-Herstellern festgelegt werden, würden grobe Versionsangaben auch automatisch von allen Nutzern übernommen. Eckersley weist dabei auch darauf hin, dass es ein dem widersprechendes Interesse gibt, genaue Versionsnummern für Debugging nutzen.

Eckersley hat zudem vorgeschlagen, Informationen wie die installierten Schriftarten nicht als Liste, sondern nur noch auf Anfrage für spezifische Werte herauszugeben [22]. Das hätte zur Folge, dass obskure Schriftarten, die eine große Menge von Informationen tragen, nicht oder nur durch Prüfung einer großen Liste von Schriftarten in das Fingerprinting einbezogen werden könnten. Auch die Sortierung solcher Listen könnte nicht mehr als Information verwendet werden, um Nutzer zu unterscheiden.

2.6.2. Standardisierung von Systemen

Identische Browserinstallationen können durch eine Messung eines Browserfingerprints nicht unterschieden werden. Je ähnlicher sich Browserinstallationen sind, desto genauer muss die Messung sein, um die Unterschiede zu erkennen. Findet eine breite Standardisierung von Browserinstallationen statt, indem zum Beispiel der Installationsvorgang standardisiert wird, geben die Nutzer zwar noch Informationen über sich preis, können anhand dieser aber nicht mehr unterschieden werden.

Mit diesem Effekt erklärt Eckersley, dass manche Browser vergleichsweise wenig Information preisgeben. Konkret nennt Eckersley Browser von Smartphones wie dem iPhone, die bereits vorinstalliert sind und kaum konfigurierbar sind. Ein anderes Beispiel sind geklonte Installationen, bei denen eine Referenzinstallation eines Betriebssystems auf viele Computer kopiert wird [22].

Das TOR-Browser Bundle nutzt diesen Effekt, um möglichst wenig Informationen über Browserinstallationen preiszugeben. Es versucht nicht, vorhandene Browserinstallationen anzupassen und wird nicht über die üblichen Installationswege installiert. Stattdessen ist das TOR-Browser Bundle ein stark in sich abgeschlossenes Paket und benötigt auf Linux keine weitere Installation oder Konfiguration. Durch die Empfehlung, aus verschiedenen Gründen keine anderen Browser als das TOR-Browser Bundle und keine Browserplugins im Tor-Netzwerk zu nutzen, wird versucht einen Standard zu setzen, der von einer großen Gruppe von Personen genutzt wird [20].

2.6.3. Skriptsprachen einschränken

Skriptsprachen wie Javascript erlauben, wie in Sektion 2.4.2 dargestellt, viele Informationen aus Seitenkanälen und API-Abfragen zu gewinnen. Dabei ist nicht nur das Erheben von Daten möglich, sondern diese können auch an einen Server gesendet werden.

Ein radikaler Ansatz, dies zu verhindern, ist das Deaktivieren von Skriptsprachen [46]. Dies kann über den Browser oder spezialisierte Plugins wie NoScript geschehen. Dadurch gehen alle auf Skripten basierenden Nutzungsmöglichkeiten verloren und Webseiten können sogar unbenutzbar werden. Die Analysemöglichkeiten werden dadurch allerdings so stark eingeschränkt, dass dies eine effektive Gegenmaßnahme darstellt und komplette Fingerprintingbibliotheken nicht ohne Javascript lauffähig sind.

Ein anderer Ansatz, der auch im TOR-Browser verwendet wird, ist die Reduzierung der Mächtigkeit der Skriptsprachen [9, 11]. Dort wurde eine Javascriptfunktion so abgeändert, dass sie einen festen Wert zurückgibt, wodurch Canvasfingerprinting verhindert werden soll.

Im TOR-Browser-Bundle wird zusätzlich eine Zwischenlösung zwischen der kompletten Geheimhaltung und der kompletten Preisgabe von Informationen angewendet. Dabei darf nur eine feste Menge von Sprachen über die dafür vorgesehene Javascript-API abgefragt werden [12].

2.6.4. Traffic normalisieren

Eine Möglichkeit, die vorgeschlagen wurde, um TCP/IP-Fingerprinting von Servern zu verhindern, ist das Normalisieren des Traffics [51]. Bei dieser Methode wird der von den Servern generierte Traffic abgefangen,

in eine abstrahierte Form übertragen und der Traffic auf Basis der abstrahierten Form wiederhergestellt. Bei dieser für TCP/IP semantisch unbedeutenden Transformation sollen Informationen, die auf die in den Servern eingesetzte Software schließen lassen, verloren gehen.

Wollte man dieses Konzept auf Browserfingerprinting übertragen, müsste der HTTP-Traffic abgefangen werden und eine Abstrahierung des HTTP-Traffics möglich sein. Da die Verschlüsselung von HTTP-Traffic serverseitig erzwungen werden kann, kann eine Manipulation des Traffics verhindert werden und somit ist dieses Konzept nicht direkt auf Browserfingerprinting übertragbar.

2.6.5. Fingerprints ändern

Soll verhindert werde, dass ein Browser wiederidentifiziert wird, kann versucht werden die Browserinstallation so stark zu verändern, dass sie nicht wiedererkannt wird. Wenn das gelingt, ist die Menge der preisgegebenen Informationen und die Einzigartigkeit der Browserinstallation unwichtig. Kann die Veränderung der Browserinstallation vom Server erkannt werden, kann dies jedoch genutzt werden, um die veränderte Information zu ignorieren und die Browserinstallation wiederzuerkennen.

Mit dem Browser-Plugin Firegloves gab es einen Ansatz, dieses Prinzip zu nutzen, um ein Plugin gegen Browserfingerprinting zu erstellen. Dabei wurden Browserattribute randomisiert und so der Fingerprint geändert. Zu diesem Projekt sind aber leider nur noch Referenzen und Beschreibungen zu finden [16, 47], weswegen auf eine detaillierte Beschreibung verzichtet werden muss.

Mit PriVaricator [47] existiert ein weiter Ansatz, der auf der Randomisierung von Browsermerkmalen beruht. Bei diesem Plugin wird die über Javascript ermittelte Liste der installierten Schriftarten und Plugins randomisiert, um einen neuen Fingerprint zu erhalten. Ein Orakel für ein in der Industrie eingesetztes Fingerprintingskript konnte auf diese Weise getäuscht werden und hat die randomisierten Fingerprints als verschiedene Ergebnisse deklariert.

Auf dem Ansatz von PriVaricator aufbauend gibt es zwei weitere Umsetzungen des Randomisierens, bei der einer die Randomisierung mit einer Erkennung von Browserfingerprintingskripten kombiniert [24] und ein zweiter die Form der Randomisierung verbessert [36].

Es existiert auch eine Arbeit, die sich mit der optimalen Form der Randomisierung beschäftigt [16].

Es gibt auch einen Bericht [34] über Empfehlungen, einen Useragent-Switcher zu nutzen, um Einbruchserkennungssysteme zu umgehen. Dabei wird der Fingerprint verändert und kann auch Inkonsistenzen haben, ein naiver Fingerprintingalgorithmus wird den Fingerprint als neu und einzigartig erkennen. Aus diesem Grund wird auch vorgeschlagen, bei Einbruchserkennungs- und Sicherheitssystemen nur schlecht fälschbare Merkmale in den Fingerprint einzubeziehen.

2.6.6. Entzug der Kommunikation

Da Browserfingerprinting auf der Analyse von Kommunikation basiert, benötigt es diese für eine Analyse. Wird der analysierenden Partei die Kommunikation entzogen, kann das Browseringerfingerprinting nicht mehr stattfinden. Dies bedeutet allerdings auch, dass ein eventueller Mehrwert, der durch diese Kommunikation entsteht, verloren geht. Ist die analysierende Partei eine Drittpartei, kann der Zugriff auf die Kommunikation über eine verschlüsselte Verbindungen verhindert werden.

Ein anderer Fall, bei dem die Kommunikation entzogenen werden kann, sind Trackingdienstleister, deren Analysecode als verstecktes Element in Webseiten von Drittparteien eingebunden wird [39]. Um die Analyse durch diese Dienstleister komplett zu unterbinden, wurden Plugins wie Ghostery [4] entwickelt, die diese eingebundenen Elemente erkennen und entfernen. Dadurch wird nicht mit dem Trackingdienstleister kommuniziert und dieser kann keine Analyse durchführen. Die populären Plugins AdBlock Plus und Ghostery haben angekündigt, neben ihrem eigentlichen Einsatzzweck auch Browserfingerprintingskripte zu filtern und zu blockieren. Entzug der Kommunikation wurde aber bisher nicht wissenschaftlich untersucht.

2.6.7. Fälschung von Fingerprints

Werden Fingerprints genutzt, um Sessions abzusichern oder Accounts an Nutzer zu binden, muss ein Angreifer bestimmte Fingerprints nachstellen, wozu er allerdings Kenntnis über den zu fälschenden Fingerprint erlangen muss. Da Browserfingerprinting auch in kritischen Bereichen eingesetzt werden könnte, wurde die Fälschung von Fingerprints in der Forschung erwähnt, waren aber nicht selbst Thema einer Forschungsarbeit.

Ein Mittel, einen Fingerprint nachzuahmen, ist das Verändern von Browsereinstellungen. Normalerweise nicht veränderbare Einstellung, wie der Useragent, können mit Plugins wie dem Useragent-Switcher geändert werden [34]. Falschinformationen können allerdings über Inkonsistenzen in den Angaben [46] oder über kaum veränderbare Eigenheiten im Syntaxparsing des Browsers [10] erkannt werden.

2.6.8. Kombination von Maßnahmen

Eine Kombination aus Erkennen, Filtern und Randomisieren von Fingerprints wurde in der Masterarbeit von Khademi untersucht und implementiert [24]. Diese Implementierung wurde bei 4 Fingerprintingskripten aus Forschung und Industrie getestet und konnte diese täuschen, ohne die Geschwindigkeit des Browsers zu beeinträchtigen.

2.6.9. Übersicht über die Maßnahmen

In der wissenschaftlichen Diskussion wird folgende Reihe von Maßnahmen erwähnt, die Fingerprinting beoder verhindern sollen.

- Das Einschränken von Selbstbeschreibungen, Standardisierung von Systemen und Einschränken von Skriptsprachen wurden bereits von Eckersley erwähnt, aber nicht erforscht. Trotzdem werden diese Maßnahmen von Eckersley und anderen als gute Kandidaten für Maßnahmen gegen Browserfingerprinting empfohen.
- Das Normalisieren von Traffic wurde in einem anderen Kontext erforscht. Dieser Ansatz lässt sich aber nicht auf das Browserfingerprinting übertragen.
- Das Ändern von Fingerprints wird von verschiedenen Stellen empfohlen und wurde in Form der Randomisierung in zwei Arbeiten erfogreich getestet.
- Das Entziehen der Kommunikation wird bereits von Browserplugins wie NoScript ermöglicht, wurde aber nicht erforscht.
- Das Fälschen von Fingerprints hingegen wurde in Zusammenhang mit Browserfingerprinting, aber nicht direkt als Schutz vor Nutzerverfolgung erwähnt.
- Auch das Kombinieren von Maßnahmen, um eine höhere Effektivität zu erreichen, wurde für der Kombination von Erkennen, Filtern von Fingerprintingskripten und Randomisieren von Fingerprints untersucht.

Der größte Teil dieser Schutzmaßnahmen ist also schlecht erforscht und lediglich das Randomisieren von Fingerprints wurde in mehreren Veröffentlichungen untersucht und als wirksam bestätigt.

2.7. Zusammenfassung

In diesem Kapitel wurde der Forschungsstand zum Thema Browserfingerprinting dargestellt.

Wie in Abschnitt 2.1 beschrieben, wurde diese Methode theoretisch modelliert und es wurde mehrfach empirisch bewiesen, dass viele Nutzer mit dieser Methode identifizierbar sind. Ebenfalls wurde bereits gezeigt, dass Browserfingerprinting außerhalb der Forschung eingesetzt wird.

Gegenmaßnahme	Vorkommen
Traffic normalisieren	In anderem Kontext von Smart genutzt [51]
Deaktivieren von Javascript	empfohlen von Eckerley [22] empfohlen von Boda [18]
Einschränken von Selbstbeschreibungen	empfohlen von Eckerley [22] genutzt vom TOR-Browser-Bundle [8]
Standardisierung von Systemen	Von Boda als Hinderniss für Identifizierbarkeit bezeichnet [18] Von Eckerley als Hinderniss für Identifizierbarkeit bezeichnet [22] genutzt vom TOR-Browser-Bundle [8]
Ändern von Fingerprints	FireGloves [16] Fluxfonts [54] geplant für das TOR-Browser-Bundle [8] erforscht mit PriVaricator [47] erwähnt von Laperdix [36] behandelt von Besson und anderen [16]
Einschränken von Skriptsprachen	FireGloves [54] empfohlen von Eckerley [22] genutzt vom TOR-Browser-Bundle [8]
Deaktivieren von clientseitigen Skripten	empfohlen von Tillman [54] vom TOR-Browser-Bundle genutzt [8]
Das Entziehen der Kommunikation	von AdBlock genutzt [36] von Ghostery genutzt [36]

Tabelle 2.3.: Die bisherige Nennung verschiedener Maßnahmen gegen Browserfingerprinting

Wie in Abschnitt 2.2 festgestellt, existieren auch eine Reihe dem Browserfingerprinting verwandte Techniken und Typen des Browserfingerprings. Das Browserfingerprinting und das browserbasierte Devicefingerprinting werden allerdings in der Praxis nicht klar unterschieden. Es gibt auch ferne Verwandte des Browserfingerprintings, wie die Biometrie.

Die bekannten Einsatzmöglichkeiten des Browserfingerprintings wurden in Abschnitt 2.3 beschrieben und gehen über die Nutzerverfolgung hinaus. Beispielsweise kann das Browserfingerprinting genutzt werden, um das Übernehmen von Sessions zu erschweren und Online-Bezahlsysteme abzusichern. Die Methoden, die dem Nutzerinteresse eher widersprechen, da sie ihn gegen seinen Willen identifizieren oder beschränken, benötigen ein sehr hohes Maß an Information. Nutzungsmöglichkeiten, wie das zusätzliche Absichern von Sessions, funktionieren mit wenig Informationen über die Browser, da nur auf Änderungen geprüft wird und nicht versucht wird Nutzer wiederzuerkennen.

Eine wichtige Unterscheidung ist das Trennen des Browserfingerprintings in aktives und passive Browserfingerprinting, das in Abschnitt 2.4 untersucht wurde. Das aktive Browserfingerprinting ist dabei wesentlich auffälliger als das passive, kann aber auch deutlich mehr Informationen erheben.

Auf die rechtliche Situation wurde Abschnitt in 2.5 eingegangen, indem die EU Datenschutzrichtline und ihre Bedeutung für das Browseringerprinting vorgestellt wurde.

Es gibt eine Reihe bekannter Ansätze, um Browserfingerprinting zu verhindern, über die in Abschnitt 2.6 ein Überbick gegeben wurde. Diese sind bei Weitem nicht so gut erforscht wie das Browserfingerprinting selbst. Viele vorgeschlagene Gegenmaßnahmen schränken den Nutzer teilweise stark ein oder sind nicht umgesetzt. Am besten erforscht und getestet ist das Randomisieren von Fingerprints, dadurch ist es der momentane Favorit unter den Gegenmaßnahmen. Es mangelt aber an erforschten Alternativen.

Insgesamt hat sich gezeigt, dass das Browserfingerprinting eine gut erforschte Methode ist, Browserinstallationen und ihre Nutzer wiederzuerkennen. Maßnahmen gegen Browserfingerprinting sind allerdings größtenteils noch unerforscht, obwohl einige Ansaätze dafür exestieren.

3

Modell des Browser-fingerprintings

In der Studie „panopticlick“ von Eckersley [22] wurde ein mathematisches Modell des Fingerprintingprozesses genutzt, um Aussagen über die Einsetzbarkeit des Browserfingerprintings zu treffen. Auf diesem Modell basiert das in dieser Arbeit genutzte Modell des Browserfingerprintings. Da Eckersley allerdings sein Modell nur knapp vorstellt, wird sein Modell in dieser Arbeit erweitert und mit einer von Eckersley abweichenden Notation versehen.

Um das Modell detailliert vorzustellen, wird zunächst das Grundmodell in Abschnitt 3.1 vorgestellt. Dieses Modell wird anschließend in Abschnitt 3.2 erweitert, um Merkmale getrennt voneinander betrachten zu können, und in Abschnitt 3.3 werden Metriken für Anonymität vorgestellt. Anschließend wird in Abschnitt 3.4 auf die Übertragung von Erkenntnissen vom globalen auf den lokalen Kontext eingegangen und in Abschnitt 3.5 das Grundmodell um die Definition einiger Merkmalseigenschaften erweitert. Abschließend wird in Abschnitt 3.6 die Verwendung des Grundmodells in einigen Studien vorgestellt.

3.1. Grundmodell

Um Aussagen über das Browserfingerprinting treffen zu können, kann es über ein Zufallsprozess modelliert werden. Der Zufallsprozess und der Browserfingerprintingvorgang stehen auf folgende Weise in Verbindung:

Ein Zufallsexperiment entspricht dabei der Installation, Konfiguration und weiteren Änderungen eines Browsers durch einen Nutzer.

Bezeichnung	Bedeutung
X	Die Menge aller möglichen Browser
X_i	Ein Browser
F	Die Menge aller möglichen Fingerprintingalgorithmen
F_s	Ein Fingerprintingalgorithmus
E	Die Menge aller möglichen Fingerprints
f_k	Ein Fingerprint
$f_k^{F_s}$	Ein mit F_s gemessener Fingerprint
B_k	Die Menge der Browser mit dem Fingerprint f_k
$M^{Merkmalsname}$	Ein Merkmal
$M_i^{Merkmalsname}$	Eine Merkmalsausprägung
$H(E)$	Die Entropie einer Menge von Fingerprints
$P(f_k)$	Die Auftrittswahrscheinlichkeit eines Fingerprints
$I(f_k)$	Der Informationsgehalt des Fingerprints f_k
$I(\{f_i^{F_s}, f_k^{F_t}\})$	Der Informationsgehalt der Kombination aus den Fingerprints $f_i^{F_s}$ und $f_k^{F_t}$
$I(f_i^{F_s} \mid f_k^{F_t})$	Der Informationsgehalt des Fingerprints $f_i^{F_s}$ in Kenntnis von $f_k^{F_t}$

Tabelle 3.1.: Die genutzten Bezeichnungen und ihre Bedeutung

Die Zufallsverteilung dieses Zufallsexperimentes wird von den Interessen und Verhaltensweisen aller Nutzer bestimmt und wird deswegen im Normalfall als unbekannt angenommen.

Ein Browser ist in dieser Arbeit das Programm, das der Nutzer ausführt, um Webseiten zu betrachten. Der Browser entspricht also einer Browserinstanz, die installiert, konfiguriert und genutzt wurde. Das Programm an sich wird als Browsertyp oder Browserversion bezeichnet. Der Browser wird mit $X_i \in X$ bezeichnet, wobei X die Menge aller möglichen Browser ist.

Ein Fingerprintingalgorithmus ist eine Methode, Merkmale von Browsern aus X zu messen. Der Fingerprintingalgorithmus wird mit der Funktion F_j bezeichnet, die einen Browser als Parameter akzeptiert. Die Definition der Funktion F_j ist durch den eingesetzten Fingerprintingalgorithmus gegeben. Die Menge aller möglichen Fingerprintingalgorithmen wird mit F bezeichnet.

Der Fingerprint ist das Ergebnis der Messung der Merkmale eines Browsers X_i mit einem Fingerprintingalgorithmus F und entspricht somit $f_k = F(X_i)$. Ein Fingerprint wird mit $f_k \in E$ bezeichnet, wobei E die Menge aller möglichen Fingerprints ist. Die Menge E und die Mächtigkeit der Menge E wird dabei als unbekannt angenommen. Für einen konkreten Fingerprintingalgorithmus F_i wird der Fingerprint mit $f_k^{F_i}$ bezeichnet.

Die Browser mit einem Fingerprint f_k werden als B_k bezeichnet.

Der Informationsgehalt oder Überraschungswert eines Fingerprints gibt an, wie viel zur Unterscheidung von Fingerprints nutzbarer Information dieser enthält. Er wird mit der Funktion I berechnet und in Bit angegeben und ist eine andere Darstellung der Auftrittswahrscheinlichkeit eines Fingerprints. Seine Definition ist:

$$I(f_k) = -log_2(P(f_k)) = -log_2(\frac{|B_k|}{|\text{Browser}|})$$

Ist das Auftreten eines Fingerprints sehr unwahrscheinlich, ist der Informationsgehalt des Fingerprints sehr groß, da:

$$\lim_{P(f_k)\to 0} I(f_k) = \lim_{P(f_k)\to 0} -log_2(P(f_k)) = \infty$$

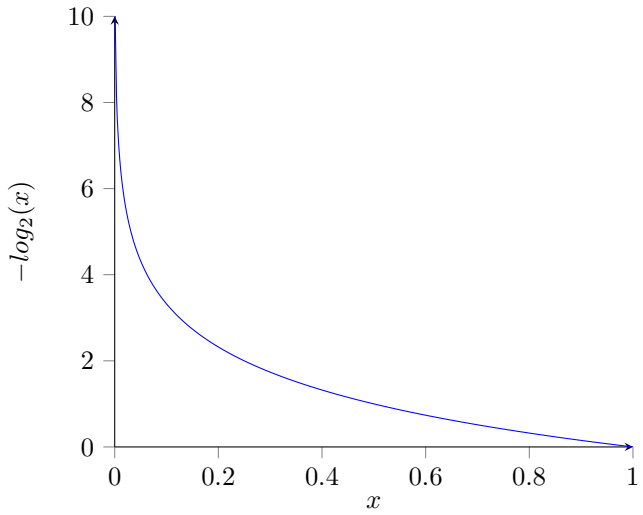


Abbildung 3.1.: Der Graph der in I genutzten Funktion $-log_2(x)$

Ist das Auftreten eines Fingerprints sehr wahrscheinlich, ist der Informationsgehalt des Fingerprints sehr klein, da:

$$\lim_{P(f_k)\to 1} I(f_k) = \lim_{P(f_k)\to 1} -log_2(P(f_k)) = 0$$

Das detailliertere Verhalten dieser Funktion kann dem Graphen 3.1 entnommen werden.

1 Bit Informationsgehalt erlaubt es mit anderen Worten, eine beobachte Menge von Browsern mit einer Information in zwei disjunkte Teilmengen mit gleicher Mächtigkeit zu unterteilen. Jedes weitere Bit erlaubt die sich ergebenden Mengen erneut zu halbieren.

Als Beispiel für die Berechnung des Informationsgehalts soll eine Menge von Browsern angenommen werden, in der

- $\frac{1}{3}$ der Browser den Fingerprint f_1 haben,
- $\frac{1}{2}$ der Browser den Fingerprint f_2 haben,
- $\frac{1}{6}$ der Browser den Fingerprint f_3 haben.

Der Informationsgehalt

- des Fingerprints f_1 ist $-log_2(\frac{1}{3}) \approx 1,58$ Bit,
- der des Fingerprints f_2 ist $-log_2(\frac{1}{2}) \approx 1$ Bit,
- der des Fingerprints f_3 ist $-log_2(\frac{1}{6}) \approx 2,58$ Bit.

Mit dem Informationsgehalt von 1 Bit des Fingerprints f_2 lässt sich die Menge der Browser also in die Teile B_{f_2} und $B_{f_1} \cup B_{f_3}$ teilen, die gleich mächtig sind.

Die Entropie der Zufallsverteilung entspricht dem Erwartungswert des Informationsgehalts für ein zufälliges f_k. Sie wird mit H bezeichnet, in Bit angegeben und hat die Formel:

$$H(E) = \sum_{k=0}^{|E|} P(f_k)I(f_k) = -\sum_{k=0}^{|E|} P(f_k)log_2(P(f_k))$$

Ist der Informationsgehalt eines Fingerprints f_k groß genug, können so viele Teilmengen aus der Menge

der Browser ausgeschlossen werden, bis nur noch ein Browser übrig bleibt. Das ist der Fall, wenn gilt:

$$I(f_k) \geq log_2(|\text{Browser}|)$$

Dies gilt, da mit jedem Bit Informationsgehalt die Hälfte der Browser ausgeschlossen werden kann.

Da so von allen Browsern nur noch ein Browser übrig bleibt, ist nur dieser mit dem Fingerprint f_k verknüpft. Dadurch kann ein Browser mit diesem Fingerprint eindeutig identifiziert werden. Wird angenommen, dass $|\text{Browser}| \leq |\text{Menschen}|$ gilt, reichen bereits 33 Bit Informationsgehalt eines Fingerprints aus, um den dazugehörigen Browser eindeutig zu identifizieren:

$$log_2(|\text{Browser}|) \leq log_2(|\text{Menschen}|) \leq log_2(8 \cdot 10^9) \leq 33$$

Soll nicht nur ein Fingerprint f_k betrachtet werden, sondern die Fingerprints aller Browser betrachtet werden, ist die Entropie ihrer Zufallsverteilung relevant. Da diese Entropie dem Erwartungswert des Informationsgehalt eines zufälligen Fingerprints entspricht, bietet sie eine Abschätzung für die Identifizierbarkeit von Browsern im Allgemeinen. So kann erwartet werden, dass ein zufälliger Fingerprint den dazugehörigen Browser identifiziert, wenn die Entropie 33 Bit beträgt [54].

3.2. Kombination von mehreren Fingerprintingalgorithmen

Um die Messung eines Fingerprints detaillierter analysieren zu können, ist es wünschenswert, Teilmessungen und nicht nur die gesamte Messung des Fingerprints betrachten zu können. Dazu können die Teilmessungen als Messungen von Fingerprints mit verschiedenen Algorithmen aus F interpretiert werden. Messen diese Algorithmen jeweils nur ein Merkmal wie den Useragent oder den Browsertyp, können Merkmal und Merkmalkombinationen so einzeln untersucht werden.

Da die Verteilung der einzelnen Merkmale wie des Useragents und des Browsertyps allerdings voneinander abhängig sein können, können diese nicht einfach kombiniert werden. Sind die Abhängigkeiten zwischen Messungen mit den Fingerprintingalgorithmen bekannt, kann die Menge der insgesamt preisgegeben Information mit der folgenden Formel berechnet werden, da [1]:

$$\begin{gathered}
I(\{F_s(X_i), F_t(X_i), F_u(X_i), ...\}) = I(\{f_j^{F_s}, f_k^{F_t}, f_l^{F_u}, ...\}) = \\
-log_2(P(f_j^{F_s} \cap f_k^{F_t} \cap f_l^{F_u} \cap ...)) = \\
-log_2(P(f_j^{F_s}) \cdot P(f_k^{F_t}|f_j^{F_s}) \cdot P(f_l^{F_u}|f_k^{F_t}|f_j^{F_s}) \cdot ...) = \\
-log_2(P(f_j^{F_s})) - log_2(P(f_k^{F_t}|f_j^{F_s})) - log_2(P(f_l^{F_u}|f_k^{F_t}|f_j^{F_s})) - ... = \\
I(f_j^{F_s}) + I(f_k^{F_t}|f_j^{F_s}) + I(f_j^{F_s}|f_k^{F_t}|f_j^{F_s})...
\end{gathered}$$

Dabei ist hier $I(f_k^{F_t}|f_j^{F_s})$ als $-log_2(P(f_k^{F_t}|f_j^{F_s}))$ definiert. Da nicht allgemein angenommen werden kann, dass Abhängigkeiten zwischen den Messungen bekannt sind, ist $I(f_k^{F_t}|f_j^{F_s})$ allerdings nur in Spezialfällen berechenbar.

Um Merkmale einfacher und genauer angeben zu können, werden an dieser Stelle Notationen für Merkmale und Merkmalsausprägungen festgelegt. Ein Merkmal wird mit $M^{Merkmalsname}$ bezeichnet. Beispielsweise ist der Useragent ein Merkmal und kann mit $M^{Useragent}$ bezeichnet werden. Definiert sind Merkmale als die Menge von Merkmalsausprägungen, die mit einem bestimmten Algorithmus aus F gemessen werden können. So ist $M^{Useragent}$ die Menge aller möglichen Merkmalsausprägungen, die das Useragent-Feld haben kann. Die Merkmalsausprägungen werden über einen Index in Bezug auf das Merkmal referenziert. Beispielsweise wird ein konkreter Useragent mit $M_i^{Useragent}$ bezeichnet.

[1] Eckersleys Formel (5) wurde hier nicht verwendet, da sie nicht nachvollzogen werden konnte.

3.3. Anonymitätsmaße

Eine Betrachtungsweise der Anonymität, die aus der Perspektive eines einzelnen Nutzers hilfreich ist, ist die der Anonymity-Sets [49]. Ein Anonymity-Set ist durch die Menge der Browser gegeben, die die gleichen Merkmalsausprägungen oder denselben Fingerprint wie der betrachtete Browser haben. Beispielsweise hat ein Nutzer mit einem Anonymity-Set der Größe 1 einen einzigartigen Fingerprint und ein Browser zu dem es 2 andere Browser mit identischen Merkmalsausprägungen gibt, hat ein Anonymity-Set der Größe 3. Hat ein Browser ein Anonymity-Set, das größer als 1 ist, gibt es mindestens einen anderen Browser mit identischen Merkmalen. Er kann also nicht von diesem unterschieden, also auch nicht eindeutig identifiziert werden. Wird trotzdem eine Identifizierung versucht, gelingt diese nach einfacher Rechnung mit der Wahrscheinlichkeit $\frac{1}{|\text{Anonymity-Set}|}$.

Die Größe des Anonymity-Sets eines Fingerprints kann aus dessen Informationsgehalt und der Anzahl der Browser berechnet werden, da:

$$|\text{Anonymity-Set}| = |B_i| = \frac{|B_i|}{|\text{Browser}|} \cdot |\text{Browser}| = P(f_i) \cdot |\text{Browser}| = 2^{(-I(f_i))} \cdot |\text{Browser}|$$

Eine Verallgemeinerung des Anonymity-Sets ist die k-Anonymität [53], die genutzt wird, um den Grad der Anonymisierung von Datenbanken zu messen. Im Kontext vom Browserfingerprinting ist eine k-Anonymität gegeben, wenn jeder Fingerprint ein Anonymity-Set von mindestens der Mächtigeit k hat. Ist eine Menge von Browsern k-anonym, ist garantiert, dass ein Identifikationsversuch eines beliebigen Browsers höchstens mit der Wahrscheinlichkeit $\frac{1}{k}$ gelingt.

3.4. Fingerprinting spezifischer Nutzergruppen

Es kann nicht immer davon ausgegangen werden, dass die Menge der zu unterscheidenden Browser der Menge der Menschen auf diesem Planeten entspricht. Von globaler Identifizierbarkeit wird gesprochen, wenn ein Browser aus der Menge aller eingesetzten Browser identifiziert werden muss, und von lokaler Identifizierbarkeit, wenn spezifische Nutzergruppen betroffen sind. Verfolgt beispielsweise eine einzelne Webseite oder ein Verbund von Seiten ihre Nutzer, müssen nur die Browser dieser Nutzer unterschieden werden. Sind dies 1 000 Browser, reichen im Bezug auf diese Nutzergruppe bereits 10 Bit Informationsgehalt aus, um einen Browser zu identifizieren, da $2^{10} = 1024 > 1000$.

Wurde der Informationsgehalt für die Gesamtheit der Browser oder eine andere Gruppe von Nutzern berechnet, kann dieser nicht auf eine andere Gruppe von Browsern übertragen werden und nur in Spezialfällen für eingegrenzte Nutzergruppen übernommen werden.

Hat diese Nutzergruppe dieselbe prozentuale Verteilung von Fingerprints wie die Menge, mit deren Hilfe der Informationsgehalt bestimmt wurde, kann der Informationsgehalt unverändert übernommen werden.

Ein anderer Fall, in dem Aussagen über den Informationsgehalt getroffen werden können, liegt vor, wenn die Nutzergruppe eine repräsentative Stichprobe der Gesamtheit der Nutzer darstellt. Das neue Anonymity-Set der Fingerprints und somit auch die neue Entropie ist dabei das Ergebnis eines Zufallsexperiments. Die Wahrscheinlichkeit, dass l Nutzer aus Anonymity-Set^{global} auch im lokalen Anonymity-Set^{lokal} auftreten, kann mit folgender Formel berechnet werden:

$$P(|\text{Anonymity-Set}^{lokal}| = l) = \frac{\binom{|\text{Anonymity-Set}^{global}|}{l}\binom{|\text{Browser}^{global}|-|\text{Anonymity-Set}^{global}|}{|\text{Browser}^{lokal}|-l}}{\binom{|\text{Browser}^{global}|}{|\text{Browser}^{lokal}|}}$$

Diese Formel beschreibt das Dividieren der Anzahl der günstigen Fälle durch die Anzahl der möglichen Fälle. Die möglichen Fälle sind dabei durch Binomialkoeffizienten $\binom{|\text{Browser}^{global}|}{|\text{Browser}^{lokal}|}$ gegeben und die günstigen Fälle durch das Produkt aus der Anzahl $\binom{|\text{Anonymity-Set}^{global}|}{l}$ der Möglichkeiten, l Nutzer aus dem

Anonymity-Set^{global} auszuwählen, und der Anzahl $\binom{|\text{Browser}^{global}|-|\text{Anonymity-Set}^{global}|}{|\text{Browser}^{lokal}|-l}$ der Auswahlmöglichkeiten für die restlichen Nutzer. Aus dieser Formel folgt die Wahrscheinlichkeit, dass die mindestens l aus Nutzer aus Anonymity-Set^{global} auch in Anonymity-Set^{lokal} auftreten:

$$P(|\text{Anonymity-Set}^{lokal}| \geq l) = 1 - \sum_{i=0}^{l-1} \frac{\binom{|\text{Anonymity-Set}^{global}|}{i}\binom{|\text{Browser}^{global}|-|\text{Anonymity-Set}^{global}|}{|\text{Browser}^{lokal}|-i}}{\binom{|\text{Browser}^{global}|}{|\text{Browser}^{lokal}|}}$$

Sowohl der Erwartungswert der Entropie als auch der Informationsgehalt der Fingerprints für die eingegrenzte Nutzergruppe muss den Werten der Gesamtheit der Nutzer entsprechen.

$\lvert\text{Browser}^{global}\rvert$	$\lvert\text{Browser}^{lokal}\rvert$	$\lvert\text{Anonymity-Set}^{global}\rvert$	l	$P(\lvert\text{Anonymity-Set}^{lokal}\rvert \geq l)$
470161	10000	1000	10	99,8%
470161	10000	1000	20	64,0%
470161	10000	1000	30	4,1%
470161	10000	100	2	63,0%
470161	10000	100	5	6,3%
470161	1000	1000	2	62,8%
470161	1000	1000	5	6,4%
470161	100	1000	2	1,9%
470161	10000	100	2	99,8%
470161	10000	100	5	6,3%
470161	10000	10	2	1,8%

Tabelle 3.2.: Die Übertragbarkeit der Größen von Anonymity-Sets unter verschiedenen Bedingungen

Der Fall, dass die Nutzer keine repräsentative Stichprobe der Gesamtheit sind, sondern dass die Auswahl durch Gemeinsamkeiten wie sozialer Schichtzugehörigkeit oder Herkunft beeinflusst wird, ist allerdings realistischer. In diesem Fall ist es schwer, korrekte Aussagen zu treffen. Eine Aussage kann allerdings getroffen werden, wenn auch die Fingerprints der Browser dieser Nutzer ebenfalls stärkere Gemeinsamkeiten aufweisen als die Fingerprints der Gesamtheit der Nutzer. Dadurch muss der Informationsgehalt der Fingerprints, die eine solche Gemeinsamkeit aufweisen, sinken oder gleich bleiben. Für Fingerprints, die diese Gemeinsamkeiten nicht aufweisen, muss der gegenteilige Effekt eintreten.

3.5. Merkmalseigenschaften

In dieser Arbeit wird auf die Merkmale und nicht nur auf ganze Fingerprints eingegangen. Gruppen von Merkmalen können wie in Abschnitt 3.2 beschrieben zu neuen Merkmalen zusammengefasst und bezeichnet werden. Um detaillierter über diese Merkmale sprechen zu können, werden an dieser Stelle einige Bezeichnungen für Merkmalseigenschaften festgelegt.

Veränderliche, teilweise veränderliche und unveränderliche Merkmale werden dadurch unterschieden, ob sie vom Nutzer änderbar sind oder nicht. Ein Merkmal M ist genau dann unveränderlich, wenn alle Merkmalsausprägungen aus M vom Nutzer nicht plausibel durch andere Merkmalsausprägungen aus M ersetzt werden können. Ein Merkmal M genau dann veränderlich, wenn alle Merkmalsausprägungen aus M vom Nutzer plausibel durch alle anderen Merkmalsausprägungen aus M ersetzt werden können. Ansonsten ist das Merkmal teilweise veränderlich. Ob ein Nutzer Merkmale plausibel verändern kann, kann dabei nicht nur an der technischen Möglichkeit ausgemacht werden, sondern muss auch dessen technisches Wissen und dessen Toleranz für Verluste von Zeit und Komfort einbeziehen. Dadurch hängt die Abgrenzung vom Nutzer ab und muss im konkreten Kontext festgelegt werden.

Beispielsweise ist der Useragent des Browser veränderlich, da dieser leicht durch jegliche Werte ersetzt werden kann. Die bereitgestellten Javascriptfunktionen sind nur sehr schwer zu ändern und können deswegen

als unveränderlich gelten.

Bekannte, teilweise bekannte und unbekannte Merkmale werden dadurch unterschieden, ob sie allgemein bekannt sind. Ein Merkmal M ist bekannt, wenn die Existenz eines Algorithmus aus F bekannt ist, der alle Merkmalsausprägungen aus M korrekt ermitteln kann. Ein Merkmal M ist teilweise bekannt, wenn ein Teilmerkmal von M, aber nicht M bekannt ist. In allen anderen Fällen ist das Merkmal unbekannt. Ist ein Algorithmus nur bestimmten Personengruppen bekannt, kann ein Merkmal dieser Persongruppe bekannt und gleichzeitig für die Allgemeinheit unbekannt sein.

Beispielsweise waren Merkmale, die nur über das Canvasfingerprinting erhoben werden können, zum Zeitpunkt von Eckersleys unbekannt und wurde nach der Veröffentlichung von Mowery zu einem bekannten Merkmal.

Stabile und instabile Merkmale unterscheiden sich dadurch, ob sie sich verändern. Ein Merkmal M ist stabil, wenn es über die Zeit bei allen Browsern gleichbleibt. Ein Merkmal M ist instabil, wenn dieses Merkmal über die Zeit bei mindestens einem Browser andere Merkmalsausprägungen annimmt. Da es sein kann, dass sich instabile Merkmale selten oder nur bei wenigen Browsern ändern, wird ein Merkmal als stabiler als ein anderes bezeichnet, wenn dieses sich seltener oder bei weniger Browsern ändert.

Ob Flash installiert ist oder nicht, ändert sich beispielsweise selten und ist somit ein stabiles Merkmal, während sich die exakte Version des Flashplugins öfter ändert und somit ein instabiles Merkmal ist.

3.6. Anwendungen des Modells

Das von Eckersley aufgestellte Modell wurde in mehreren Studien genutzt, um auf die globale Zusammensetzung von Fingerabdrücken zu schließen.

Das übliche Ziel dieser Studien war es, die Vermutung zu untermauern, dass ein großer Teil der Browser über ihren Fingerabdruck eindeutig zu identifizieren ist. Dazu wurden konkrete Fingerprintingalgorithmen erstellt und die Fingerprints einer Stichprobe von Nutzern gemessen, die über Medienkampagnen zur Studie eingeladen wurden. Um die Nutzer trotz sich ändernden Fingerprints wiedererkennen zu können und so Fehlerquellen wie instabile Merkmale auszuschließen, wurden Cookies und Supercookies verwendet.

Mit Hilfe dieser möglichst großen Stichprobe von Fingerprints wurde eine Abschätzung der Wahrscheinlichkeitsverteilung für die Browser X erstellt. Dazu wurden trotz nicht neutraler Stichprobe die Wahrscheinlichkeiten der gemessenen Fingerprints als Annäherung der Zufallsverteilung angenommen. Die so gewonnen Werte für die Entropie und Informationsgehalt der Zufallsverteilung der Fingerprints und bestimmter Merkmale geben eine untere Grenze für die globale Entropie und den globalen Informationsgehalt von Merkmalen an. Auf diese Weise konnte zu der gemessenen Probe eine Untergrenze der Entropie angegeben werden und Abschätzungen über die Menge an Informationen getroffen werden, die ein Browser oder ein Merkmal preisgibt. In Tabelle 3.3 sind Studien zusammengstellt, die mit dieser Technik arbeiteten.

Studie	Jahr	Teilnehmer	Entropie	einzigartige Nutzer
Eckerleys [22]	2010	460 171	>18,1 Bit	83,6%
Tillman [54]	2012	17 937	-	92,57%
Yen und andere [59]	2012	1 771 907	>20,09 Bit	80,62%
Broenink [20]	2012	1 124	>6,31 Bit	-
Boda und andere [18]	2012	989	>8,57 Bit	-

Tabelle 3.3.: Die Ergebnisse verschiedener Studien zu Browserfingerprinting

3.7. Zusamenfassung

In Abschnitt 3.1 wurde zunächst das Grundmodell des Browserfingerprintings von Eckersley dargestellt, indem die Komponenten und Kernbegriffe des Browserfingerprintings beschrieben wurden. Diese Grundbegriffe wurden in Abschnitt 3.2 um eine Formel für die Kombination von mit verschiedenen Fingerprintingalgorithmen gemessenen Fingerprints erweitert. Mit dieser Formel können einzene Merkmale von Fingerprints detaillierterer betrachten werden. Zusätzlich dazu wurden in Abschnitt 3.3 verschiedene Maße der Anonymität von Nutzern vorgestellt.

Anschließend wurden in Abschnitt 3.4 bestimmte Nutzergruppen betrachtet. In Abschnitt 3.5 wurden verschiedene Typen von Merkmalen definiert.

Abschließend wurde in Abschnitt 3.6 gezeigt, wie das Modell in Studien verwendet wurde.

4 Auswahl von Gegenmassnahmen

Um Nutzerverfolgung mittels Browserfingerprinting zu verhindern oder zu behindern, gibt es bereits einige Ansätze, die in Kapitel 2.6 vorgestellt wurden. Diese wurden aber teilweise nicht auf ihre Effektivität hin überprüft oder sind noch komplett unerforscht.

Um auch neue Methoden entwickeln und besser Gegenmaßnahmen für die weitere Untersuchung auswählen zu können, werden in Kapitel 4.1 einige Vorüberlegungen zu den Gegenmaßnahmen gegen Browserfingerprinting getätigt. Anschließend werden die Gegenmaßnahmen und Strategien in Kapitel 4.2 genauer dargestellt, die in dieser Arbeit untersucht werden, und in Kapitel 4.3 Gegenmaßnahmen und Strategien dargestellt, die nicht weiter betrachtet werden.

4.1. Vorüberlegungen

Browserfingerprinting hat prinzipielle Stärken und Schwächen, die vom eingesetzten Fingerprintingalgorithmus unabhängig sind oder für ganze Gruppen von Fingerprintingalgorithmen gelten, die hier möglichst vollständig aufgelistet werden. Die Stärken und Schwächen werden unter dem Fokus der Nutzerverfolgung betrachtet und als Orientierung genutzt, um die zu überprüfenden Gegenmethoden zu bewerten.

4.1.1. Schwächen des Browserfingerprintings

Falsch negative Ergebnisse stellen einen der Fehler dar, die beim Browserfingerprinting auftreten können und bezeichnen das Versagen, Browser wiederzuerkennen. Dieser Fehler kann auftreten, wenn 2 Nutzer denselben Fingerprint besitzen, und kann bedeuten, dass die Verfolgung eines Nutzers unterbrochen wird. Die zu dem alten und neuen Fingerprint angesammelten Daten könnten also nicht mehr in Verbindung gebracht werden und der nutzbare Datensatz wäre geringer.

Tritt dieser Fehler nur bei bestimmten Nutzern oder bei seltenen Ereignissen auf, ist der Schaden auf diese beschränkt. Solche Fehler müssten also häufig bei einer großen Masse von Nutzern auftreten, um die allge-

meine Nutzerverfolgung über Browserfingerprinting ernsthaft zu behindern. Trozdem sind falsch negative Ergebnisse eine Schwäche des Browserfingerprintings.

Falsch positive Ergebnisse repräsentieren die zweite grundsätzliche Art, auf die Browserfingerprinting versagen kann, und bezeichnen die Erkennung unterschiedlicher Browser als ein einzigen. Dieser Fehler kann auftreten, wenn sich der Fingerprint eines Browsers verändert, und wenn dieser Fehler nicht abgefangen wird, werden die Aktivitäten mehrerer Nutzer zusammengefasst und zum Zwecke der Nutzerverfolgung als eine Gesamtheit bewertet.

Falsch positive Ergebnisse sind für das Browserfingerprinting also wesentlich problematischer als falsch negative Ergebnisse, da nicht nur weniger Daten erhoben werden können, sondern fehlerhafte Verbindungen zwischen den Daten der beteiligten Nutzer und Falschinformationen erzeugt werden. Aus diesem Grund sind falsch positive Ergebnisse eine große Schwäche des Browerfingerprintings.

Die Notwendigkeit der Übermittlung von Daten vom Browser zum Analyseserver ist eine Schwäche des Browserfingerprintings. So werden beim auf clientseitigen Skripten basierenden Fingerprinting die Merkmale des Browsers im Browser selbst bestimmt. Die ermittelten Daten müssen nun dem Analyseserver aber auf irgendeine Art und Weise mitgeteilt werden. Scheitert diese Übermittlung der Daten, können die ermittelten Merkmale nicht verwendet werden.

Auch andere Arten des Browserfingerprintings wie passives Browserfingerprinting benötigen Informationen, die vom Browser preisgegeben werden und einem Analyseserver mitgeteilt werden müssen. Durch das Abfangen der Übermittlung können diese Informationen der Verwertung entzogen werden.

Das Browserfingerprinting ist also prinzipiell dafür anfällig, dass die Kommunikation des Browsers manipuliert wird.

Clientseitige Codeausführung ist die Basis für aktives Browserfingerprinting. Dazu muss dem Browser der Teil des Codes zur Verfügung gestellt werden, mit dem der Fingerprint erhoben wird. Dadurch ist eine Codeanalyse möglich, die es theoretisch erlaubt, aktives Browserfingerprinting zu erkennen, weiter zu analysieren und zu bekämpfen. Eine solche Analyse kann allerdings sehr aufwendig sein, wenn die Fingerprintingskripte, wie bereits teilweise praktiziert [12], ihrerseits Analysen behindern.

Auch besteht für clientseitige Codeausführung keine Garantie für die korrekte Ausführung des Codes. Der Code kann fehlerhaft, manipuliert oder auch gar nicht ausgeführt werden. So können zum Beispiel manipulierte Browser genutzt werden, um Browserfingerprintingskripte zu suchen [12].

Dieser Nachteil für Browserfingerprinting wird dadurch abgemildert, dass eine automatische Analyse von Code nicht immer möglich ist. Einem automatischen Erkennen aller Browserfingerprintingskripte widerspricht die Unmöglichkeit, das Halteproblem zu lösen, aus dem sich folgern lässt, dass das die Funktionsweise von Programmen nicht allgemein bestimmbar ist. Daraus muss letztendlich eine Situation folgen, wie sie aus dem Antivirus-Bereich und vom Erkennen bösartiger Skripte bekannt ist.

Dass Browserfingerprintingskripte sich auf ein unmanipuliertes Ausführen ihres Codes durch die Nutzer verlassen müssen, ist also eine Schwäche des Browserfingerprintings, auch wenn eine Manipulation erschwert werden kann.

Das Nutzen der erhobenen Fingerprints, um beispielsweise Persönlicheitsprofile zu erstellen, muss als Motivation für den professionellen Einsatz von Browserfingerprinting angenommen werden. Dies birgt zwei Nachteile für das Browserfingerprinting.

Erstens kann versucht werden, von der Nutzung der Fingerprints auf nicht sichtbare Teile eines Fingerprintingalgorithmus zu schließen. Dies kann beispielsweise in Form von Orakeln passieren, die durch ihre Reaktionen auf Anfragen verraten, ob ein Fingerprintingalgorithmus getäuscht werden konnte oder nicht [47].

Zweitens ist anzunehmen, dass die Nutzung der Fingerprints Ansprüche an die Güte der Fingerprints hat. Treten zu viele Fehler auf, wäre es vorstellbar, dass die Gütekriterien für eine weitere Verwendung unterschritten werden und so eine weitere Verwendung verhindert wird.

Instabile Merkmale wurden schon früh als Problem für das Browserfingerprinting erkannt. So kann ein naiver Fingerprintingalgorithmus die Messung von Fingerprints eines Browsers nicht verknüpfen, wenn sich dessen Fingerprint verändert. Dies kann zum Beispiel durch Updates oder Neukonfigurationen geschehen.

In den Arbeiten von Eckersley [22] und Tillmann [54] zum Browserfingerprinting wurden Algorithmen demonstriert, die die Veränderungen der meisten Fingerprints ausgleichen konnten. Eckersley sieht aber in instabilen Fingerprints eine Möglichkeit, gegen Browserfingerprinting vorzugehen. Die Beweise und Formeln der Kernteile der Arbeiten von Eckersley und Tillmann beschäftigen sich aber nicht weiter mit dieser Erweiterung des Browserfingerprintings, wodurch die Auswirkungen solcher Algorithmen auf die dort entwickelten Beweise und Formeln unbekannt ist.

Sind einzelne Merkmale instabil, ist nicht klar, ob die Nutzerverfolgung durch ihre Einbeziehung profitieren würde. Werden instabile Merkmale nicht in den Fingerprint einbezogen, stehen diese nicht zur Unterscheidung von Nutzern zur Verfügung und der Informationsgehalt der Fingerprints sinkt. Andererseits verringert sich dadurch auch die Instabilität des Fingerprints und es müssen weniger Änderungen ausgeglichen werden.

Dass also zwischen gut unterscheidbaren und über längere Zeit verfolgbaren Fingerprints abgewogen werden muss, ist eine Schwäche des Browserfingerprintings.

Der schwache Beweis der Effizienz des Browserfingerprintings ist ein Nachteil des Browserfingerprintings. So ist das Browserfingerprinting zwar gut erforscht und es ist bekannt, dass viele Nutzer mit dieser Technik identifizierbar sind, dies ist allerdings kein Beweis dafür, dass Browserfingerprinting für alle Nutzer und auch in der Zukunft funktioniert. Studien können nur Aussagen über den Zeitraum der Studie und Prognosen über die Zukunft treffen. Für exakte Aussagen über die Zukunft des Browserfingerprintings ist dieses zu sehr von konkreten Fingerprintingmethoden, eingesetzten Browsern und der Gefahrenwahrnehmung der Nutzer abhängig.

Es ist auch nicht bewiesen, dass nicht mehrere Nutzer durch Zufall denselben Fingerprint haben können, und in den Studien zu Browserfingerprinting sind Nutzer, die sich einen Fingerprint teilen, nicht unüblich. Dies sorgt dafür, dass bei Erkennen eines identischen Fingerprints nicht ohne Restzweifel gefolgert werden kann, dass derselbe Browser vorliegt. Aufgrund dessen muss davon ausgegangen werden, das ein Browserfingerprintingalgorithmus ein gewisses Maß an Fehlern produziert, die in irgendeiner Weise toleriert werden müssen.

Dass das korrekte Funktionieren des Browserfingerprintings nur für einen Teil der Nutzer und nicht für die Zukunft belegt ist, ist eine Schwäche des Browserfingerprintings, da dies die Möglichkeit offen hält, den Anteil der nicht verfolgbaren Nutzer auszuweiten und das Browserfingerprinting in der Zukunft zu verunmöglichen.

Die statistischen Abhängigkeiten zwischen den einzelnen gemessenen Merkmalen haben sowohl Nachteile als auch Vorteile für das Browserfingerprinting. Der Nachteil ergibt sich dadurch, dass eine fälschungssichere Analyse der Browserversion nur wenig neue Information liefert, wenn der Useragent bereits die Information über die Browserversion in sich trägt. Als Beispiel sollen die vom Browser bereitgestellten Funktionen $M^{Funktionen}$ und der Useragent $M^{Useragent}$ gemessen werden, wobei

$$P(M_{Firefox}^{Funktionen} | M_{Firefox}^{Useragent}) \approx 1$$

.

Die Abhängigkeiten wirken einerseits gegen das Browserfingerprinting, da sich die Gesamtmenge der preisgegebenen Information durch das Erkennen von $M_{Firefox}^{Funktionen}$ im Wissen von $M_{Firefox}^{Useragent}$ nicht oder nur

kaum vergrößert. Es würde lediglich die Information gewonnen, dass der Useragent nicht auf bestimmte Arten gefälscht wurde. Von dieser Information ist aber kein großer Inforamtionsgehalt zu erwarten, da:

$$I(M_{Firefox}^{Funktionen}|M_{Firefox}^{Useragent}) = -log_2(P(M_{Firefox}^{Funktionen}|M_{Firefox}^{Useragent})) \approx -log_2(1) = 0$$

4.1.2. Stärken des Browserfingerprintings

Beim Aufbau von Webseiten genutzte Angaben sind eine Stärke des Browserfingerprintings, denn diese Angaben können nicht beliebig verändert werden, ohne die Benutzbarkeit des Browsers einzuschränken. Beispiele dafür sind die gewünschte Sprache, die unterstützten Komprimierungsarten oder der Useragent, die genutzt werden, um für Nutzergruppen optimierte Webseiten zu generieren [3]. Werden nicht wahrheitsgemäße Angaben zum Aufbau einer Webseite genutzt, wird sie auf falsche Weise optimiert und kann eventuell nicht oder nur falsch dargestellt werden. Dadurch können die betroffenen Angaben als unveränderlich gelten.

Aufgrund der großen Menge von unterschiedlichen Webseiten ist es schwer einzuschätzen, welche Merkmale eines Browsers zur Optimierung genutzt werden. Dies bedeutet, dass Manipulationen von Merkmalen aufgrund zentral gegebener Regeln das Risiko bergen, die Benutzbarkeit des Browsers einzuschränken.

Die mögliche Nutzung von zusätzlichen Informationen, um die Nutzerverfolgung durch Browserfingerprinting zu verbessern, ist eine Stärke des Browserfingerprintings. So kann die Menge der unbekannten Nutzer reduziert oder Fingerprints verknüpft werden. Dadurch wird die Verfolgung der Nutzer sehr erleichtert, weil so der Informationsgehalt des Fingerprints sinkt, der notwendig ist, um einen Nutzer wiederzuerkennen. Solche Zusatzinformation kann wie beim Devicefingerprinting das Einbeziehen des Fingerprints des gesammten TCP/IP-Stacks [54] oder auch nur die IP-Adresse sein [59].

Seitenkanäle sind bei mancher Software ein Problem, da sie unerwünscht Informationen freigeben. Dieses Problem kann zwar auf verschiedene Weisen angegangen, aber nicht komplett behoben werden. Die Seitenkanäle erlauben, selbst bei kritischen Anwendungen wie beispielsweise Verschlüsselung [27], geheime Informationen auszulesen.

Seitenkanäle werden auch beim Browserfingerprinting genutzt, um Informationen zu gewinnen. In Anbetracht der Probleme bei anderen Anwendungen und der Komplexität von Browsern ist es schwer vorstellbar, dass diese Seitenkanäle in Browsern eliminiert werden können.

Serverseitiger Code wird auf den für Browserfingerprinting eingesetzten Analyseservern genutzt. Dies betrifft das Erheben mancher Browsermerkmale, das Speichern der Fingerprintdatenbank und die Nutzung dieser Datenbank. Solange das Analysesystem selbst nicht untersucht werden kann, können diese Vorgänge nicht direkt analysiert werden und nur Vermutungen über sie angestellt werden. Dies ist ein großer Vorteil für das Browserfingerprinting, da manche Typen des Browserfingerprintings wie passives Browserfingerprinting von den Nutzern nicht erkannt werden können. Dass die Art der Speicherung und die genaue Weise des Vergleichs von Fingerprints nicht direkt beobachtbar ist, erschwert zudem eine Untersuchung von eingesetzten Fingerprintingalgorithmen.

Die Möglichkeit, wichtige Teile der Logik des Browserfingerprinting einer Analyse zu entziehen, ist eine Stärke des Browseringerprintings.

Der Vorgangscharakter des Browserfingerprintings wird in existierenden Arbeiten übergangen und das Fingerprinting eines Browsers wird als einmalige Messung behandelt. Würden aber mehrere Messungen des Fingerprints zum Beispiel über Cookies oder Login-Vorgänge in Verbindung gebracht, könnten Veränderungen am Browserfingerprint erkannt und analysiert werden. Bei einem stabilen Fingerprint muss das Ergebnis einer solchen Analyse lediglich sein, dass sich der Fingerprint nicht geändert hat.

Verändert sich ein Fingerprint aber, kann diese Veränderung auf Muster hin analysiert und in eine abstraktere Form des Fingerprints miteinbezogen werden. Der Fingerprintingalgorithmus könnte beispielsweise versuchen Plugins zu erkennen, die die Merkmale von Browsern randomisieren, und darauf reagieren. Zusätzlich könnte es möglich sein, Merkmale des Browsers in den Fingerprint einzubeziehen, die über mehrere Zugriffe hinweg gemessen werden müssen. Als ein solches Merkmal könnte das Verhalten von 3,85% der Bing Nutzer dienen, deren Cookies bei jeder einzelnen Anfrage erneut gelöscht werden [59]. Alleine diese Form des Cookielöschens trägt also $-log_2(3,85\%) \approx 4,7$ Bit Informationsgehalt.

Die Möglichkeit, diese Daten in den Fingerprint einzubeziehen ist eine Stärke des Fingerprintings.

Die statistischen Abhängigkeiten zwischen den erhobenen Merkmalen sind auch eine Stärke des Browserfingerprintings. So bieten diese Abhängigkeiten die Möglichkeit, Fälschungen zu erkennen. Bei einer Fälschung ist beispielsweise $P(M^{Funktionen}_{InternetExplorer}|M^{Useragent}_{Firefox})$ und nicht $P(M^{Funktionen}_{Firefox}|M^{Useragent}_{Firefox})$ für die Anonymität der Fälscher relevant. Dabei muss

$$P(M^{Funktionen}_{InternetExplorer}|M^{Useragent}_{Firefox}) \leq 1 - P(M^{Funktionen}_{Firefox}|M^{Useragent}_{Firefox})$$

sein. Der Informationsgehalt des Fingerprints muss in diesem Fall sehr groß sein, da:

$$I(\{M^{Funktionen}_{InternetExplorer}, M^{Useragent}_{Firefox}\}) = I(M^{Useragent}_{Firefox}) + I(M^{Funktionen}_{InternetExplorer}|M^{Useragent}_{Firefox}) \geq$$

$$I(M^{Funktionen}_{InternetExplorer}|M^{Useragent}_{Firefox}) = -log_2(P(M^{Funktionen}_{InternetExplorer}|M^{Useragent}_{Firefox})) \geq$$

$$-log_2(1 - P(M^{Funktionen}_{Firefox}|M^{Useragent}_{Firefox})) \approx$$

$$-log_2(0) = \infty$$

Widersprechen beispielsweise detailliertere Angaben dem Useragent, wird klar, dass er gefälscht ist. Davon ist ein großer Informationsgehalt zu erwarten. Zusätzlich dazu ist auch die Art und Weise der Fälschung bekannt, wodurch sich die Fälscher noch einmal untereinander differenzieren.

Diese statistischen Effekte treten selbst bei einfachen Auswertungslogiken ohne explizite Fälschungserkennung ein, so lange $M^{Funktionen}_{Firefox}$, $M^{Funktionen}_{InternetExplorer}$ und $M^{Useragent}_{Firefox}$ gemessen werden. Eine fehlgeschlagene Fälschung muss also nicht tatsächlich als eine solche erkannt werden, um Nutzer durch einen seltenen Fingerprint auffällig zu machen.

Veränderungen des Zufallsprozesses sind im Grundmodell des Browserfingerprintings nicht vorgesehen. Da aber beständig neue Browser unter neuen Bedingungen hinzugefügt werden und alte Browser verschwinden, ist dieser Zufallsprozess nicht statisch. Auch bestehen Abhängigkeiten von Messungen des Zufallsprozesses und der Neukonfiguration oder Abänderungen von Browsern. Dies ist zum Beispiel der Fall, wenn Merkmale verändert oder versteckt werden, weil sich in einer Messung herausgestellt hat, dass sie einen hohen Inormationsgehalt haben.

Durch Änderungen an Browsern kann also der Zufallsprozess selbst verändert werden. Werden koordinierte Änderungen an vielen Browsern vorgenommen, sind auch bewusste Manipulationen des Zufallsprozesses denkbar. Dadurch könnte die Verlässlichkeit des Browserfingerprints verringert werden, indem die Stabilität der Browserfingerprints oder die Entropie der Zufallsverteilung reduziert wird.

Die Abschätzung der Verteilung der Fingerprints wird durch die Veränderungen des Zufallsprozesses erschwert, da die dafür genutzten Fingerprints nicht über lange Zeiträume hinweg gesammelt werden dürften. Werden Browserfingerprints wie beim seit 2010 zugänglichen „panopticlick“ über lange Zeit gesammelt, wird die aufgebaute Zufallsverteilung dadurch verzerrt, dass alte und neue Fingerprints vermischt werden. Selbst unter der Annahme, dass sich die Browser von lediglich 75% der Teilnehmer an der „panopticlick“ Studie stark verändert haben und somit die gemessenen Fingerprints nicht mehr exisitieren, wird heutzutage

der Informationsgehalt von Fingerprints bei „panopticlick“ bereits um 2 Bit überschätzt, da:

$$I(f_k) = -log_2(\frac{|B_k|}{\frac{|Browser|}{4}}) = -log_2(\frac{|B_k|}{|Browser|} \cdot 4) =$$

$$-log_2(\frac{|B_k|}{|Browser|}) - log_2(4) = -log_2(\frac{|B_k|}{|Browser|}) - 2$$

Das Schutzparadox wurde bereits von Eckersley als wichtiger Faktor für Maßnahmen gegen Browserfingerprinting erkannt [22] und wurde unter dem Namen „The Paradox of fingerprintable Privacy Enhancing Technologies“ beschrieben. Aufgrund der Länge dieses Namen wird dieses Paradox im weiteren als „Schutzparadox“ bezeichnet.

Das Schutzparadox beschriebt das Problem, dass Techniken, die dazu gedacht sind, die Privatsphäre und Anonymität der Nutzer zu verbessern, selbst Teil des Fingerprints sein können. Nutzen nur wenige Nutzer diese Techniken, können diese also paradoxerweise die Anonymität ihrer Nutzer verschlechtern.

Dies ist eine Stärke des Browserfingerprintings, da dies dafür sorgt, dass Maßnahmen gegen dieses leicht ihrem eigentlichen Ziel entgegengesetzt wirken können.

4.2. Zu untersuchende Gegenmaßnahmen und Strategien

In diesem Abschnitt werden die Gegenmaßnahmen und Strategien, die in der weiteren Arbeit untersucht werden, ausgewählt. Die theoretische Untersuchung findet in Kapitel 5, die experimentelle Überprüfung Kapitel 6 und schließlich die Simulation in Kapitel 7 statt.

Das Standardisieren von Browsern ist eine der Gegenmaßnahmen, die beim TOR-Browser getroffen werden. Eckersley erwähnt das Standardisieren von Browsern auch mehrfach, da er Maschinenklone und iPhones als auffällig schlecht zu identifizieren bezeichnet und das Reduzieren der Versionsangaben vorschlägt. Auch könnte versucht werden, mit Betriebssystemen von Live-CDs auf existierende Standards zurückzugreifen.

Da das Standardisieren der Browser eine vorgeschlagene und umgesetzte Maßnahme ist, wird dies in dieser Arbeit untersucht.

Das Randomisieren von Fingerprints ist ein intuitiver Ansatz, Browserfingerprinting zu verhindern. Dadurch soll der Browserfingerprint so instabil werden, dass er nach einer Änderung nicht mehr wiedererkannt werden kann. Funktioniert dies, kann der Nutzer nicht wiedererkannt werden, auch wenn sein Fingerprint einzigartig ist.

Dies wurde mit dem Projekt PriVaricator [47] bereits untersucht, welches das Implementieren einer Cromiumvariante beinhaltet, die gängige Fingerprintingalgorithmen auf diese Weise täuschen konnte. Aufgrund von Bedenken bezüglich der Benutzbarkeit des Browsers wurden dabei aber nur wenige Eigenschaften des Browsers randomisiert.

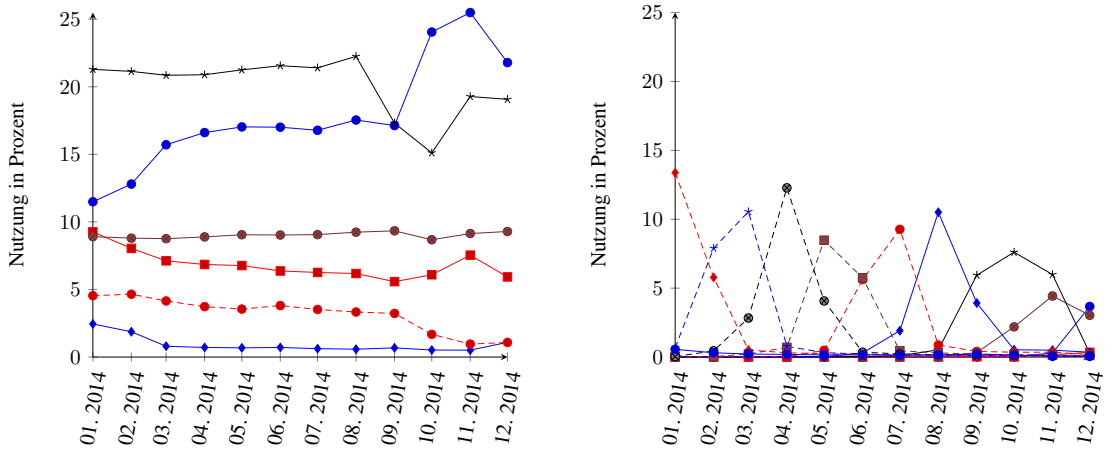
Trotz des guten Forschungsstands wird diese Methode noch einmal untersucht, damit Effekte wie der Vorgangscharakter oder die Reaktionen auf die Randomisierung seitens der Browserfingerprintingskripte einbezogen werden können.

Automatische Browserupdates sind eine Gegenmaßnahme, die von Browserherstellern getroffen werden könnte.

Der Browsertyp und die Browserversion sind Merkmale, die große Abhängigkeiten zu anderen Merkmalen haben. Diese sind zum Beispiel die Javascriptengine oder die Menge der ohne Plugins zur Verfügung

gestellten Funktionen. Wäre die Browserversion im Bezug auf den Browsertyp standardisiert, wären also auch die von der Browserversion abhängigen Merkmale standardisiert.

Über automatische Browserupdates könnte versucht werden, diese Standardisierung von Browserversionen zu erzwingen. Da dies auch noch andere Sicherheitsvorteile bietet, haben manche Browser bereits automatische oder sogar erzwungene Updates. Der Vorteil, den ein solches System bieten könnte, ist aus Abbildung 4.1 erkennbar. Hier wird deutlich, dass die eingesetzten Firefoxversionen konsistenter als die Internetexplorerversionen sind, obwohl sich die Firefoxversionen häufiger ändern.



(a) Der Marktanteil der Internet Explorer Versionen 6 bis 11 im Jahr 2014

(b) Der Marktanteil der Firefox Versionen 25 bis 35 im Jahr 2014

Abbildung 4.1.: Visualisierung der Versionsadaption für Firefox und Internetexplorer im Jahr 2014 [5]

Da der Effekt von automatischen Browserupdates auf das Browserfingerprinting nicht untersucht wurde und die in Abbildung 4.1 genutzten Daten nicht wissenschaftlich abgesichert sind, soll dies in dieser Arbeit weiter untersucht werden.

Das Verheimlichen von Merkmalen ist ein Ansatz, das Browserfingerprinting zu behindern, indem Informationen nicht zur Analyse zur Verfügung gestellt werden.

Das Verheimlichen von Merkmalen in der Form des Deaktivierens von clientseitigen Skriptsprachen über Browserplugins wie NoScript ist die wohl am meisten empfohlene Maßnahme gegen Browserfingerprinting. Da Browserfingerprinting viel Information durch die Nutzung von Skripten gewinnen kann, besteht die Vermutung, dass es durch das Deaktivieren von Javascript so stark behindert wird, dass ein Identifizieren des Browsers nicht mehr möglich ist. Um im TOR-Browser das Canvasfingerprinting zu verhindern, werden Informationen verheimlicht, indem bei Lesezugriffen auf die Farbwerte eines Pixels immer eine feste Farbe zurückgegeben wird.

Das Deaktivieren von Javascript mit solchen Plugins wird zwar oft empfohlen, ist aber kaum wissenschaftlich untersucht. Zusätzlich ist es einer der radikalsten Ansätze des Verheimlichens von Merkmalen, weshalb es in dieser Arbeit stellvertretend für das Verheimlichen von Merkmalen untersucht wird.

Das Fälschen von Fingerprints kann notwendig sein, wenn mit Browserfingerprinting abgesicherte Sessions übernommen werden sollen. Es könnte aber auch versucht werden, dies als Maßnahme gegen Browserfingerprinting zu nutzen, indem ein Browserfingerprint gefälscht wird, der möglichst oft vorkommt. Als Methoden einen Fingerprint mit einem geringen Informationsgehalt zu finden, sollen bekannte Fingerprints und Messungen von Fingerprints genutzt werden.

Dies wird in dieser Arbeit mit Hinblick auf die technische Komplexität einer Fälschung und dem Informationsgehalt einer solchen Fälschung untersucht.

Durch das eingeschränkte Fälschen von Fingerprints kann versucht werden, die zu erwarteten Probleme beim uneingeschränkten Fälschen von Fingerprints abzumildern, indem nicht beliebige, sondern Fingerprints gefälscht werden, die bereits teilweise mit dem vorhandenen Fingerprint übereinstimmen. Zu diesem Zweck sollen Merkmale nicht verändert werden, von denen eine große Menge von Merkmalen abhängt, sondern vorzugsweise solche, von denen keine anderen Merkmale abhängen.

Da dies einen bisher unerforschten Ansatz darstellt, wird er in dieser Arbeit untersucht.

Das Blockieren von Kommunikation könnte als Maßnahme gegen Browserfingerprinting dienen, da das Browserfingerprinting teilweise von Drittparteien als Service angeboten wird [39]. Dabei werden Analyseserver zur Verfügung gestellt, die den Analysecode beinhalten und in Webseiten eingebunden werden können. Arbeitet die einbindende Seite und der Analyseserver bei der Verfolgung der Nutzer durch Vergleich von Zugriffen zusammen, werden beide Server als ein Analyseserver gewertet. Wird die Kommunikation des Browsers zu den Analyseservern, die das Browserfingerprinting durchführen, komplett blockiert, kann der Browser nicht analysiert und kein Fingerprint erhoben werden. Wird die Kommunikation nur teilweise blockiert, entspricht dies einem Filtern von Kommunikation.

Dies verspricht eine sehr effektive Maßnahme gegen Browsefingerprinting zu sein und die populären Plugins AdBlock und Ghostery haben angekündigt, neben ihrem eigentlichen Einsatzzweck auch Browserfingerprintingskripte zu blockieren. Aus diesen Gründen wird dies in dieser Arbeit untersucht.

Das Filtern von Kommunikation unterscheidet sich vom Blockieren der Kommunikation dadurch, dass nicht die komplette Kommunikation zu einem Server blockiert wird, sondern nur bestimmte Teile der Kommunikation herausgefiltert werden. Wird die komplette Kommunikation zu den Analyservern herausgefiltert, entspricht dies der Gegenmaßnahme des Blockierens der Kommunikation.

Browserfingerprinting bleibt so zwar möglich, es kann aber versucht werden, Browserfingerprinting und insbesondere aktives Browserfingerprinting einzuschränken. Herausgefiltert werden kann der Analysecode, der vom Server zum Browser gesendet wird, oder die im Browser erhobenen Daten, die zur Analyse einem Analyseserver geschickt werden müssen. Wegen der Ankündigung von AdBlock und Ghostery, die Kommunikation von Browserfingerprintingskripten zu filtern, wird es in dieser Arbeit untersucht.

Kontextspezifische Fingerprints könnten das Browserfingerprinting behindern, da das Browserfingerprinting zur Nutzerverfolgung darauf abzielt, Nutzer über Kontextwechsel verfolgen zu können. Um dies zu unterbinden, kann versucht werden, in verschiedenen Kontexten einen unterschiedlichen Fingerprint zu präsentieren.

Da diese Maßnahme vielversprechend erscheint und auch als Seiteneffekt unbeabsichtigt auftreten kann, wird dies in dieser Arbeit untersucht.

Mit einer Kombination verschiedener Maßnahmen gegen Browserfingerprinting könnte versucht werden, die Schwächen von einzelnen Maßnahmen auszugleichen. Durch Kombination könnten so Angriffe vermieden und zusätzliche Ebenen des Schutzes geboten werden, um bei Versagen einer Schutzmaßnahme trotzdem eine Wiederidentifizierung verhindern zu können.

Dieser Ansatz wurde in der Forschung bereits mit der Kombination von Filtern von Browserfingerprintingskripten, dem Erkennen von Browserfingerprintingskripten und dem Randomisieren von Fingerprints untersucht [24]. Im praktischen Einsatz kombiniert der TOR-Browser mehrere Maßnahmen gegen Browserfingerprinting.

Dies sind allerdings nur wenige Beispiele für einzelne Kombinationen von Maßnahmen und die möglichen Kombinationen aus verschiedenen Maßnahmen wurden nicht strukturiert untersucht. Aufgrund dieser Unklarheit wird auch die Kombination aus den vorgestellten Maßnahmen untersucht. Die Formulierung einer solchen Kombination findet erst nach der Untersuchung der einzelnen Maßnahmen statt, um sinnvolle Kombination von Maßnahmen bestimmen zu können.

4.3. Nicht zu untersuchende Gegenmaßnahmen und Strategien

Die Gegenmaßnahmen, die in dieser Arbeit bewusst nicht untersucht werden, werden hier der Vollständigkeit halber trotzdem kurz dargestellt.

Die Illegalisierung von Browserfingerprinting ist eine Möglichkeit, das Browserfingerprinting zu behindern. Um eine anderweitige Nutzung von Browserfingerprinting wie die Absicherung von Sessions nicht einzuschränken, sollte ein Verbot nur die Nutzerverfolgung betreffen oder andere Nutzungen mit Ausnahmen versehen.

Als eine Basis für eine solche rechtliche Konstruktion könnte die Regelung der EU zu Cookies dienen, die Cookies und Pseudocookies nur mit Zustimmung des Nutzers erlaubt. Das Verbot dürfte sich allerdings nicht nur auf das Speichern von Daten im Browser beziehen, sondern einen Kernvorgang des Browserfingerprintings betreffen. Dies kann das Erheben, Speichern oder Übermitteln von Daten über die von Privatpersonen eingesetzten Computern und Programmen sein.

Als eine andere Basis für eine Illegalisierung von Browserfingerprinting können Datenschutzgesetze dienen. In diesen wird das Speichern und die Weitergabe von personenbezogenen oder personenbeziehbaren Daten bereits reguliert. Das könnte dahin gehend erweitert werden, dass Daten, die in Bezug zu Merkmalen von Personen oder ihren technischen Systemen gespeichert werden, nicht als pseudonym, sondern als personenbeziehbar gewertet werden, was juristisch einen entscheidenden Unterschied macht.

Die EU-Datenschutzrichtlinie 95/46/EG könnte auch einen Ansatz bieten, dass Nutzen von Fingerprints als Indentifizierer zum Ablegen von zu Daten verbieten. So verbietet diese unter gewissen Umständen Daten mit einem Identifikator zu verknüpfen, der aus „einem oder mehreren spezifischen Elementen, die Ausdruck [der] physischen, physiologischen, psychischen, wirtschaftlichen oder sozialen Identität sind," besteht [23]. Dies entspricht dem Verbot, Daten mit dem Fingerprint verschiedener Identitäten einer Person zu verknüpfen, und müsste lediglich um eine Art technische Identität erweitert werden.

Eine Formulierung und Bewertung eines solchen Vorschlages fällt allerdings in den Fachbereich von Juristen und wird in dieser Arbeit nicht untersucht. Absehbar ist allerdings, dass politische und juristische Probleme überwunden werden müssen, um den rechtlichen Rahmen zu verändern. Aber auch wenn dieser rechtliche Rahmen verändert wird, sind Durchsetzungsschwierigkeiten zu erwarten.

Die Aufklärung der Nutzer ist für das Browserfingerprinting relevant, da eine breite Masse von Nutzern betroffen ist. Diese Nutzer sind sich der Existenz und der Funktionsweise des Browserfingerprintings nicht unbedingt bewusst. Durch Aufklärung der Nutzer kann versucht werden, politischen Druck aufzubauen und eine Nutzerbasis zu schaffen, die an Gegenmaßnahmen teilnimmt. Eine solche Nutzerbasis kann zum Beispiel helfen Browserfingerprinting weiter zu erforschen oder das Privacy-Paradox zu überwinden.

Um eine solche Aufklärung zu erreichen, könnte das Browserfingerprinting, begleitet von Medienarbeit, weiter erforscht werden. So würde nicht nur die Bevölkerung aufgeklärt, sondern auch das Browserfingerprinting angreifbarer werden.

Wissenschaftliche Veröffentlichung zum Thema Browserfingerprinting haben mit mehreren Artikeln in großen Zeitungen bereits eine gute Medienresonanz gehabt. So konnten zu Eckersleys Studie 11 Berichte, zu Tillmanns Diplomarbeit 10 Berichte und zu der neuesten Studie zu Canvasfingerprinting 21 Berichte gefunden werden. Die tatsächliche Zahl der Berichte ist vermutlich größer, aber auch bereits diese Zahlen zeigen ein großes Interesse der Medien an diesem Thema. Ein weiterer Faktor ist, dass ein Großteil der Bevölkerung mit der Nutzerverfolgung im Internet nicht einverstanden ist [54] und somit für solche Artikel ansprechbar ist.

So ist dieser Ansatz ist zwar vielversprechend, wird aber in dieser technisch ausgerichteten Arbeit nicht weiter behandelt.

Das Vertrauen in Fingerprintstabilität zu schwächen, könnte die Profiteure von Browserfingerprinting verunsichern und dafür sorgen, dass mit dem Vertrauen in die Fingerprintstabilität auch das Vertrauen in die Fähigkeit des Browserfingerprintings sinkt, Browser tatsächlich wiederzuerkennen.

Gelänge dies, müsste die Stabilität der gemessenen Fingerprints wiederum erhöht werden, um dieses Vertrauen wiederzuerlangen. Dies könnte geschehen, indem die Menge der im Fingerprint verwendeten Merkmale reduziert wird. Als Nebeneffekt kann sich allerdings auch der Informationsgehalt der Fingerprints verringern.

Da auf diese Weise der verwertbare Informationsgehalt der Fingerprints gesenkt werden könnte, ohne dass die technischen Gegebenheiten geändert werden müssten, ist dies ein interessanter Ansatz. Weil eine ojektive Untersuchung eines solchen Vorgehens für den Autor nicht greifbar ist, wird dies nicht näher untersucht.

Opt-Out Lösungen werden nicht untersucht, da dieses nicht auf technischer Ebene durchgesetzt werden kann. Untersucht werden könnte lediglich, ob die Fingerprinter den Nutzerwunsch, nicht verfolgt zu werden, respektieren. Nutzbare Opt-Out Lösungen wären der DO-NOT-TRACK-Header oder spezialisierte Opt-Out Lösugen, wie sie verschiedene Dienstleister für Browserfingerprinting anbieten [24].

Die Erforschung des Browserfingerprintings selbst hat zwar keinen direkten Einfluss auf das Browserfingerprinting an sich, kann aber dazu dienen, bessere Aussagen über das Browserfingerprinting zu treffen und Maßnahmen gegen Browserfingerprinting zu verbessern. Dies kann die Erforschung der Verteilung der Fingerprints und die Erforschung der eingesetzten Browserfingerprintingskripte betreffen.

Bei der Erforschung der Verteilung können die Techniken von Eckersley und Tillmann genutzt und verfeinert werden. Eine nützliche Erweiterung wären beispielsweise das Bereitstellen von Rohdaten oder das Einschränken der betrachteten Fingerprints auf bestimmte Zeiträume. Dadurch kann eingeschätzt werden, welche Merkmale und Merkmalsausprägungen eine hohe Entropie beziehungsweise einen hohen Informationsgehalt aufweisen, und ob Maßnahmen sinnvoll durchzuführen sind.

Die Erforschung von vorhandenen Fingerprintingalgorithmen würde helfen, die Gefahr durch Browserfingerprinting nicht nur theoretisch betrachten zu können. Dadurch könnten Gegenmaßnahmen zu konkreten Analysemethoden entwickelt werden, durch Codeanalysen neue Analysemethoden entdeckt werden und Möglichkeiten gefunden werden, den Erfolg von Gegenmaßnahmen zu testen. Dies könnte mit Werkzeugen wie FPDetective geschehen und wurde bereits mehrfach angewendet [24, 46]

4.4. Zusammenfassung

In diesem Kapitel wurde bestimmt, welche Maßnahmen gegen Browserfingerprinting in der weiteren Arbeit untersucht werden. Dazu wurden zunächst in Abschnitt 4.1 Stärken und Schwächen des Browserfingerprintings gesammelt und kurz betrachtet. In den darauf folgenden Abschnitten 4.2 und 4.3 wurde bestimmt welche Maßnahmen weiter untersucht beziehungsweise nicht weiter untersucht werden.

Zur weiteren Untersuchung wurden folgende Maßnahmen bestimmt:

- Das Standardisieren von Browsern
- Das Randomisieren von Fingerprints
- Automatische Browserupdates
- Das uneingeschränkte Fälschen Fingerprints
- Das eingeschränkte Fälschen von Fingerprints
- Das Blockieren von Kommunikation
- Das Filtern von Kommunikation
- Kontextspezifische Fingerprints
- Kombinationen verschiedener Maßnahmen

5 Analyse der Gegenmassnahmen

Für eine Analyse der Maßnahmen gegen Browserfingerprinting wird auf Eckersleys Modell zurückgegriffen, das in Kapitel 3 dargestellt wurde. Die Gegenmaßnahmen, die in Kapitel 4 ausgewählt wurden, werden in diesem Kapitel untersucht. Die dabei getroffenen Aussagen sollen möglichst allgemeingültig sein und nicht von Trends in der Browserwahl abhängig sein. Die Maßnahmen werden von Abschnitt 5.1 bis Abschnitt 5.10 untersucht und in 5.11 zusammengefasst.

5.1. Standardisieren von Browsern

Das Standardisieren von Browsern hat zum Ziel, alle Browser oder Gruppen von Browsern ähnlicher zu machen. Dadurch soll die Wahrscheinlichkeit gesteigert werden, dass sich die standardisierten Browser in einem großen Anonymity-Set befinden oder sogar alle Browser nicht voneinander unterschieden werden können. Dies kann dadurch modelliert werden, dass ein Anteil x der Browser die von ihnen standardisierten Merkmale $M^{standard}$ auf die gleiche Merkmalausprägung $M_i^{standard}$ setzen. Die nicht standardisierten Merkmale M^{fest} bleiben unverändert und bestimmen den minimalen Informationsgehalt der Fingerprints und die minimale Entropie deren Verteilung. Ein Fingerprint mit den Merkmalen $M_j^{standard}, M_k^{fest}$ würde also durch das Standardisieren die Merkmale $M_i^{standard}, M_k^{fest}$ annehmen.

Wird ein Merkmal für alle Browser standardisiert, würde $M^{standard}$ nur noch $M_i^{standard}$ enthalten. $M_i^{standard}$ hätte somit einen Informationsgehalt von 0 Bit und $M^{standard}$ hätte eine Entropie von 0 Bit. Die Entropie von M^{fest} und der Informationsgehalt der Merkmalsausprägungen aus M^{fest} bleiben dabei unverändert und der Informationsgehalt des standardisierten Fingerprints ist durch M_k^{fest} gegeben.

Wird ein Merkmal nur von einem Anteil von x Browsern standardisiert, hat $M_i^{standard}$ einen Informationsgehalt von höchstens $-log_2(x)$ Bit. Die Entropie von $M^{standard}$ kann allerdings steigen, wenn eine Standardisierung ungünstig verläuft. Dies kann dann geschehen, wenn durch die Standardisierung vorhandene Anonymity-Sets getrennt werden. Als Beispiel dafür sollen 4 Browser mit den Merkmalsausprägungen $M_1^{standard}$, und 3 Browsern mit der Merkmalsausprägung $M_2^{standard}$ angenommen werden. Die Entropie

dieser Verteilung beträgt also:

$$-\sum_{k=0}^{|E|} P(f_k)log_2(P(f_k)) = -[(\frac{1}{4}(log_2(\frac{1}{4})) + (\frac{3}{4}(log_2(\frac{3}{4})))] \approx 0,81\text{Bit}$$

Wird nun auf $M_1^{standard}$ standardisiert und nimmt nur ein $M_2^{standard}$ an der Standardisierung teil, haben 2 Browser die Merkmalsausprägung $M_1^{standard}$ und 2 Browser die Merkmalsausprägung $M_2^{standard}$. Die Entropie der Verteilung steigt und beträgt somit:

$$-[\frac{1}{2}(log_2(\frac{1}{2})) + \frac{1}{2}(log_2(\frac{1}{2}))] = 1\text{Bit}$$

Dieses konstruierte Beispiel ist allerdings kaum noch als eine Standardisierung erkennbar, da auf eine Merkmalsausprägung standardisiert wird, die nur ein Browser nutzt, obwohl 3 Browser eine andere identische Merkmalsausprägung besitzen und nur ein Browser an der Standardisierung teilnimmt. Damit die Maßnahme den gewünschten Effekt hat, müssen also die Merkmale vieler Browser standardisiert oder auf Merkmalsausprägungen mit großem Anonymity-Set standarisiert werden, damit der gesetzte Standard ein großes Anonyity-Set entwickelt. Zusätzlich dazu muss darauf geachtet werden, nur wenige Anonymity-Sets zu trennen, damit sich die Anonymität anderer Nutzer nicht verschlechtert.

Die Unterschiede in den Browsern sind zudem nicht nur auf Zufallsprozesse, sondern auch auf unterschiedliche Interessen der Nutzer zurückzuführen. Das führt dazu, dass manche Merkmale als unveränderlich gelten müssen, um die Benutzbarkeit des Browsers nicht einzuschränken. Dies können Merkmale mit hohem Informationsgehalt wie die Wahl des Browsers, des Farbschemas oder der genutzten Plugins sein. Diese Nutzer könnten also nicht an einer Standardisierung teilnehmen und es könnte sogar sein, dass sich ihr Anonymity-Set durch Reduzieren der darin enthaltenen Fingerprints verschlechtert. Um sicherzustellen, dass eine Standardisierung von vielen Nutzern mitvollzogen werden kann, müssen Merkmale zur Standardisierung gewählt werden, bei denen nur wenig widersprechende Nutzerinteressen vorliegen, und für viele Nutzer akzeptable Merkmalsausprägungen als Standard gewählt werden.

Eine Standardisierung bietet also eine Möglichkeit, gegen Browserfingerprinting vorzugehen und bei richtiger Vorgehensweise eine Verschlechterung der Anonymität der Nutzer auszuschließen. Wird ein Merkmal für alle vorhandenen Browser standardisiert, muss der Informationsgehalt der Fingerprints und damit die Entropie sinken oder gleich bleiben.

Dies könnten Browserhersteller für ihren jeweiligen Browsertyp erreichen. Da es unplausibel ist, dass sich verschiedene Browsertypen ein Anonymity-Set teilen, ist anzunehmen, dass hierbei kaum negative Effekte auftreten. Die Studien von Eckersley und Tillmann legen zusätzlich nahe, dass die meisten Fingerprints einzigartig sind und so durch ein Standardisieren mehr Anonymity-Sets erzeugt als zerstört würden. Dadurch würde der positive Effekt überwiegen und die Entropie der Fingerprints sinken.

Da M^{fest} den minimalen Informationsgehalt bestimmt, wird die Effektivität der Maßnahme dadurch begrenzt. Es müsste also eine umfangreiche Standardisierung stattfinden, damit die Entropie ausreichend reduziert wird.

5.2. Randomisieren von Fingerprints

Das Randomisieren von Merkmalen eines Browsers kann dadurch modelliert werden, dass der Fingerprint eines Browsers in die Merkmale M^{stabil} und M^{random} aufgeteilt wird. M^{stabil} bezeichnet dabei die Merkmale, die nicht randomisiert werden und in dieser Betrachtung als stabil angenommen werden. Die randomisierten Merkmale sind hingegen instabil und werden mit M^{random} bezeichnet. Beim Randomisieren wird die Merkmalsausprägung von M^{random} auf eine zufällige Merkmalsausprägung aus M^{random} gesetzt. Ein gemessener Fingerprint ist also beispielsweise vor einer Randomisierung $\{M_i^{stabil}, M_j^{random}\}$ und nach dieser $\{M_i^{stabil}, M_k^{random}\}$.

Wird nur mit einer kleinen Menge an möglichen Ergebnissen randomisiert, ist es vorstellbar, dass sich die gemessenen Fingerprints wiederholen und der Nutzer so trotzdem unter mehreren Fingerprints verfolgt werden kann. Dies wird in Abschnitt 5.2.1 untersucht. Kann der Browserfingerprint in mehreren Zuständen der Randomisierung gemessen werden, kann, wie in Abschnitt 5.2.2 beschrieben, versucht werden die Randomisierung zu erkennen und den Nutzer trotzdem zu identifizieren. Aber auch ohne Kenntnis, dass ein spezieller Fingerprint randomisiert ist, kann, wie in Abschnitt 5.2.3 beschrieben, versucht werden eine Randomisierung auszugleichen. In Abschnitt 5.2.4 werden die Ergebnisse zusammengefasst und bewertet.

5.2.1. Notwendiges Maß der Randomisierung

Damit ein Nutzer aufgrund nicht ausreichender Randomisierung wiedererkannt werden kann, muss der Fingerprint zu verschiedenen Gelegenheiten gemessen werden. Dabei kann sich M_j^{random} verändern und zu einer anderen Merkmalsausprägung aus M^{random} werden. Das Merkmal M_i^{stabil} bleibt dagegen konstant und muss deshalb in diesem Abschnitt nicht weiter betrachtet werden.

Eine obere Grenze für die Gelegenheiten, bei denen ein neuer Fingerprint präsentiert werden kann, kann leicht gegeben werden. Da M^{random} höchstens $|M^{random}|$ verschiedene Fingerprints enthält, kann dies selbst bei optimaler Ausnutzung der Merkmalsausprägungen bei höchstens $|M^{random}|$ Messungen geschehen.

Um eine Schätzung des notwendigen Maßes zu gewinnen, die sich näher an vorhandenen Ansätzen orientiert, wird ein Spezialfall einer solchen Randomisierung betrachtet. Bei diesem wird die Wahrscheinlichkeit für die Ereignisse als gleich verteilt angenommen und bei jeder Messung ein neues Ereignis ausgewürfelt. Bei n Messungen eines Fingerprints wiederholen sich dabei keine Ereignisse mit Wahrscheinlichkeit $P_{anon} = \frac{\frac{|M^{random}|!}{(|M^{random}|-n)!}}{|M^{random}|^n}$.

Diese Formel ergibt sich dadurch, dass die Anzahl der günstigen Fälle durch die Anzahl der möglichen Fälle dividiert wird. Die Zahl der günstigen Fälle ist durch eine Variation ohne Wiederholung mit der Formel $\frac{|M^{random}|!}{(|M^{random}|-n)!}$ gegeben und die Zahl der möglichen Fälle ist durch eine Variation mit Wiederholung mit der Formel $|M^{random}|^n$ gegeben.

	$\|M^{random}\| = 10^3$	$\|M^{random}\| = 10^6$	$\|M^{random}\| = 10^9$	$\|M^{random}\| = 10^{12}$
$n = 10^1$	$95,5861\%$	$99,9955\%$	$100,0000\%$	$100,0000\%$
$n = 10^2$	$0,5959\%$	$99,5062\%$	$99,9995\%$	$100,0000\%$
$n = 10^3$	$0,0000\%$	$60,6733\%$	$99,9501\%$	$100,0000\%$
$n = 10^4$	–	$0,0000\%$	$95,1234\%$	$99,9950\%$
$n = 10^5$	–	$0,0000\%$	$0,6737\%$	$99,5013\%$

Tabelle 5.1.: P_{anon} für verschiedene n und $|M^{random}|$ in Prozent auf 4 Nachkommastellen gerundet

Es kann schon bei kleinem $|M^{random}|$ und großem n mit großer Sicherheit ausgeschlossen werden, dass Fingerprints mehrfach präsentiert werden. Dies ist aus Tabelle 5.1 ersichtlich, bei kombinatorischen Berechnungen kleine Werte wie 10^9 für $|M^{random}|$ und teilweise große Werte für n wie 10^3 genutzt wurden. Dabei werden selbst einzelne Wiederholungen als kritisches Versagen betrachtet und somit nur das vollständige Funktionieren der Randomisierung betrachtet. Der Browser PriVaricator [47] kann jeden Zugriff auf bestimmte Javascriptfunktionen randomisieren und jeweils 100 verschiedene Ergebnisse zurückgeben. Greift ein Browserfingerprintingskript beim Messen eines Fingerprints m-mal auf diese Funktionen zu, sind 10^{2m} Ergebnisse möglich, wodurch aufgrund der großen Menge von möglichen Ergebnissen zu erwarten ist, dass PriVaricator auch bei vielen Messungen kein Fingerprint doppelt generiert.

5.2.2. Erkennung des Randomisierers

Mit Hilfe von Cookies oder Logins können Browser auch ohne Browserfingerprinting wiedererkannt werden. Dadurch kann der Fingerprints des Browser mehrfach gemessen werden. Werden dadurch verschiedene

Merkmalsausprägungen aus M^{random} gemessen, kann eine Änderung des dazugehörigen Fingerprints erkannt werden. Sind diese Änderungen häufig oder weisen Muster auf, kann ein Randomisierer vermutet werden.

Ein Fingerprintingskript kann mit dieser Erkenntnis versuchen die Randomisierung auszugleichen. Die erste Möglichkeit ist, die als randomisiert erkannten Merkmale, also M^{random}, zu ignorieren. Der Haupteffekt des Randomisierers wäre ausgehebelt und die Frage nach dem Informationsgehalt des Fingerprints wäre wieder relevant. Zur Unterscheidung des Nutzers stünde also zunächst nur noch $\{M_i^{stabil}\}$ zur Verfügung.

M^{random} muss aber nicht einfach ignoriert werden, sondern das Verändern der Werte kann als bisher unbekannte Merkmalsausprägung M_*^{random} interpretiert werden. Dadurch stünden nun $\{M_i^{stabil}, M_*^{random}\}$ zur Unterscheidung des Nutzers zur Verfügung. Die Art der Änderung von M^{random} kann zusätzlich als Merkmal M^{Muster} dienen und es stünde nun $\{M_i^{stabil}, M_*^{random}, M_k^{Muster}\}$ zur Unterscheidung von Nutzern zur Verfügung.

Da M_*^{random} und M_k^{Muster} die Messung eines Zufallsprozesses darstellt, kann der zugrundeliegende Zufallsprozess nur abgeschätzt werden. Daher kann beim Einsatz eines Randomisierers nicht mit absoluter Sicherheit gesagt werden, welche Merkmale sich verändern und in welchem Ausmaß dies geschieht. Kann M_*^{random} und M_k^{Muster} allerdings korrekt erkannt werden und reicht $\{M_j^{stabil}, M_*^{random}, M_k^{Muster}\}$ zur Identifikation des Nutzers aus, bietet der Randomisierer keinen sicheren Schutz gegen Browserfingerprinting. Damit ein Randomisierer einen prinzipiell wirksamen Schutz gegen Fingerprinting bieten kann, muss dieser also versuchen den Informationsgehalt von M_j^{stabil} gering zu halten, um die Analyse seiner Funktion durch einen Fingerprinter zu verhindern. Dies könnte erreicht werden, indem viele Merkmale selten verändert werden und die Änderungen nicht vom Fingerprinter beobachtet werden können.

Übertragung auf vorhandene Randomisierungsimplementierungen

Im Falle des PriVaricators [47] wird nicht versucht die echten Werte gegen alternative Erhebungswege zu schützen. So geht durch das Ignorieren von M^{random} also kaum Information verloren, wenn alternative Erhebungswege genutzt werden, um die ignorierten Werte zu ersetzen, und M^{stabil} besitzt weiterhin eine hohe Entropie.

Da keine großen Kampagnen zur Randomisierung von Browsermerkmalen bekannt sind, ist zudem anzunehmen, dass M_*^{random} einen großen Informationsgehalt besitzt. Als weitere Informationsquelle bietet sich die Möglichkeit des PriVaricators an, die Parameter für die Randomisierung über Konfigurationsfiles zu ändern. Können diese Parameter durch ein Browserfingerprintingskript erkannt werden, kann dies als Merkmal M^{Muster} gemessen und die Randomisierer untereinander weiter differenziert werden.

Das größte Problem beim PriVaricator ist allerdings, dass keinerlei Schutz gegen das Beobachten der Randomisierung besteht. Ein Analyseskript kann sogar innerhalb eines Durchlaufs einen Fingerprint mehrfach mit derselben Methode und unterschiedlichen Ergebnissen messen. Dies könnte so oft geschehen, bis die Anzahl der Wiederholungen ausreichen, um die Form der Randomisierung mit ausreichender Zuverlässigkeit zu erkennen.

Bei der Veröffentlichung zum Randomisier FPGuard wird lediglich angenommen, dass ein Randomisierer zwischen jedem Besuch den Fingerprint ändern und Werte nicht zu stark verändert soll, um die Darstellungen von Webseiten nicht zu stark zu verzerren. Die Beobachtung des Randomisierens oder seine Umgehung, wird in jener Arbeit nicht erwähnt [24]. Dadurch ist anzunehmen, dass sich auch die Randomisierung von FPGuard gut beobachten lässt.

Der Randomisierer Blink [36] nutzt virtuelle Maschinen, die tatsächlich umkonfiguriert werden, um zu verhindern, dass inkonsistente Fingerprints erzeugt werden. Dadurch ist anzunehmen, dass M^{stabil} sehr klein ist. Ebenfalls randomisiert diese Implementierung nicht bei jeder Anfrage, sondern nur zu Beginn von Sessions, um die Benutzbarkeit des Browsers mögichst wenig zu beinträchtigen. Dadurch ist M_*^{random} schwerer zu bestimmen als bei dem PriVaricator.

Durch die Kombination aus kleinem M^{stabil} und schwer erkennbaren M_*^{random} ist anzunehmen, dass diese Implementierung einen starken Schutz gegen die Erkennung des Randomisierers ist, auch wenn dies kein Designziel dieser Implementierung war.

5.2.3. Ignorieren von Merkmalen

Ist der Einsatz und die Funktionsweise von Randomisierern zum Beispiel durch Projektseiten bekannt, können die Browserfingerprintingalgorithmen darauf reagieren, ohne zu wissen, welche Fingerprints randomisiert wurden. Dazu können die als randomisiert bekannten Merkmale allgemein ignoriert werden. Das muss aber nicht zwangsweise für alle Nutzer passieren, wenn Randomisierer nur für bestimmte Browser verfügbar sind oder die Randomisierung nur in Kombination mit anderen Merkmalen auftritt. Im Falle des PriVaricators [47] wäre dies zum Beispiel das Erkennen des Chromium-Browsers, auf dem PriVaricator basiert. Für FPGuard wird in der dazugehörigen Arbeit erwähnt, dass es eine Schwäche des Plugins ist, dass ein Fingerprintingskript direkt überprüfen kann ob, FPGuard installiert ist. Dadurch müssen Merkmale nur bei Nutzern ignoriert werden, die FPGuard auch tasächlich einsetzen.

In diesem Fall stehen die randomisierten Merkmale M^{random} nicht mehr zur Verfügung, wenn die ansonsten gemessenen Merkmale M^{stabil} implizieren, dass ein Randomisierer eingesetzt wird, indem dieser beispielsweise in der Liste der Plugins aufgelistet wird. Es stehen also nur noch die Merkmale M^{stabil} zur Identizierung der Nutzer Verfügung.

Bei PriVaricator und FPGuard werden unter anderem die Abfrage von Plugins und die Erkennung von Schriften randomisiert. Die Entropie der installierten Plugins wurde von Eckersley auf mindestens 15,4 Bit und die Entropie der gemessenen Schriften auf mindestens 13,9 Bit geschätzt [22]. So ist zu erwarten, dass die Entropie der Fingerprints relevant sinken würde, wenn die Schriften nicht mehr ausgewertet werden würden. PriVaricator versucht als Forschungsimplementierung allerdings nicht, detaillierte Prüfungen und alternative Plugin- oder Schriftbestimmungsmöglichkeiten zu verhindern. Dadurch ist anzunehmen, dass die Informationen auf anderen Wegen, wie der Schriftenerkennung über Flash, beschafft werden können und die im Fingerprint enthaltenen Informationen kaum reduziert werden würden.

Das Ignorieren von Merkmalen betrifft alle Fingerprints, deren Merkmale andeuten, dass ein Randomisierer eingesetzt wird, egal ob diese Browser einen Randomisierer tatsächlich nutzen oder nicht. In einer allgemeineren Betrachtung ist also interessant, wie sich das Ignorieren von Merkmalen auf die Entropie der Fingerprints auswirkt. Würden bei einem Anteil von x der Browser der Informationsgehalt durchschnittlich um y Bit sinken, würde die Entropie der gemessenen Werte

$$H(M^{reduziert}) =$$

$$(1-x) \cdot H(M^{gesamt}) + x \cdot (H(M^{gesamt}) - y)$$

$$1 \cdot H(M^{gesamt}) - x \cdot H(M^{gesamt}) + x \cdot H(M^{gesamt}) - x \cdot y) =$$

$$H(M^{gesamt}) - x \cdot y$$

betragen und somit um $x \cdot y$ Bit fallen.

Also Beispiel wird eine Situation angenommen, in der bei 30% der Browser die Pluginauflistung ignoriert werden würde, und dadurch durchschnittlich 10 Bit weniger an Informationen preisgegeben würde. Dadurch würde die Entropie der Zufallsverteilung aller Fingerprints um 30% von 10 Bit also 3 Bit sinken.

Zu beachten ist hierbei, dass der Verlust der Information nicht $H(M^{random})$, sondern $H(M^{random}|M^{stabil})$ entspricht und deshalb ohne Kenntnis der Abhängigkeiten nicht bestimmt werden kann. Ist allerdings bekannt, dass starke Abhängigkeiten von M^{stabil} zu M^{random} bestehen, kann M^{random} ignoriert werden, ohne dass preisgegebene Information relevant reduziert wird.

Zur Verfolgung von Nutzern ist diese Methode also zwar prinzipiell geeignet, bedeutet aber die Verschlechterung der allgemeinen Messergebnisse, wenn die verlorene Information nicht ersetzt werden kann. Daher kann davon ausgegangen werden, dass Merkmale nicht ignoriert werden, wenn nur ein kleiner Teil der Nutzer Randomisierer einsetzt und diese nicht gut erkannt werden können. In diesem Fall würde ein Erkennen von wenigen randomisierenden Nutzern bedeuten, dass durch die sinkende Entropie viele andere Nutzer nicht verfolgt werden könnten und so im Endeffekt weniger Nutzer verfolgt werden könnten.

5.2.4. Zusammenfassung

Das in Abschnitt 5.2.1 untersuchte Maß der Randomisierung ist ein zu vernachlässigendes Problem, da sehr kleine Mengen möglicher vom Randomisierer generierter Ergebnisse ausreichen, um zu verhindern, dass ein Fingerprint zwei Mal ausgegeben wird.

Ein Fingerprintingalgorithmus, der nicht auf die Nutzung von Randomisierern eingestellt ist, kann von Randomisierern getäuscht werden. Kann aber dieser einen Randomisierer erkennen und auf diesen eingehen, ist es wie in Abschnitt 5.2.2 möglich, den Nutzer trotzdem zu verfolgen, wenn $\{M_i^{stabil}, M_*^{stabil}, M_k^{Muster}\}$ genug Information trägt.

Das in Abschnitt 5.2.3 untersuchte Ignorieren von Merkmalen funktioniert zwar prinzipiell, um Nutzer trotz Randomisierung ihrer Browsermerkmale zu identifizieren, aber die Gesamtqualität der Nutzerverfolgung würde darunter leiden. Würde dies viele Nutzer und aussagekräftige Merkmale betreffen, würde das Browserfingerprinting relevant behindert werden.

Der Ansatz der Randomisierung ist also vielversprechend und funktioniert für von der Industrie genutzte Fingerprintingskripte [47]. Das Beobachten, Ignorieren und Fingerprinten von Änderungen im Fingerprint bietet allerdings die Möglichkeit, das Randomisieren von Fingerprints auszugleichen, weswegen ein besonderer Augenmerk darauf gelegt werden sollte, dass die vorgenommenen Änderungen nicht in einen Fingerprint einbezogen werden können oder nur einen kleinen Informationsgehalt haben.

5.3. Automatische Browserupdates

Im Idealfall sorgen automatische Browserupdates dafür, dass alle Browser eines Typs derselben Version entsprechen. Merkmale wie die Javascriptengine würden also keinen zusätzlichen Informationsgehalt mehr beitragen. Weiterhin auswertbar wären allerdings Merkmale, die nicht von der Browserversion bestimmt werden wie

- der Browsertyp,
- die Farbeinstellungen,
- die unterstützten MIME-Types,
- die installierten Schriften,
- die installierten Plugins,
- die Bildschirmgröße und Auflösung,
- die Sprache und
- das Betriebssystem und die Hardware.

Da zwar kein Schutzparadox durch automatische Browserupdates auftritt, aber weiterhin ein hoher Informationsgehalt zu erwarten ist, schützen sie nur in einem geringen Maß gegen Browserfingerprinting. Führen nur einzelner Nutzer automatische Browserupdates durch, ist es vorstellbar, dass sich der Informationsgehalt seines Fingerprints durch Updates sogar erhöht. Da hier aber von einem durch Browserhersteller durchgesetztes Vorgehen ausgegangen wird, ist dieser Effekt nicht zu erwarten.

5.4. Deaktivieren von clientseitigen Skriptsprachen

Das Deaktivieren von clientseitigen Skripten in Browsern verhindert das skriptbasierte Browserfingerprinting komplett. Andere Formen des Browserfingerprintings sind allerdings dadurch nicht betroffen.

Weiterhin von der Browserinstallation preisgegeben werden die Merkmale $M^{passiv \setminus skript}$, die mit passivem Browserfingerprinting erhoben werden können, ohne das clientseitige Skripte ausgeführt wurden, und die

Merkmale $M^{aktiv\backslash skript}$, die aktiv, aber ohne Nutzung von Skripten erhoben werden können. Zusätzlich ist noch das Merkmal $M^{noscript}$, das das Deaktivieren von Javascript anzeigt, erhebbar und es könnten Merkmale M^{plugin} gefunden werden, die anzeigen, auf welche Art und Weise Skripte deaktiviert werden.

Skripte zu deaktivieren, schützt also nur vor Browserfingerprinting, wenn die Merkmale $M^{passiv\backslash skript}$, $M^{aktiv\backslash skript}$, $M^{noscript}$ und M^{plugin} nicht genug Informationsgehalt aufweisen, um die Nutzer zu identifizieren. Um den Informationsgehalt dieser Merkmale abschätzen zu können, wird hier die Entropie der einzelnen Merkmale betrachtet.

Die Entropie von $M^{passiv\backslash skript}$ kann durch bisherige Schätzungen von Merkmalen aus M^{passiv} nach unten abgeschätzt werden. Für die in M^{passiv} enthaltenen Merkmale $M^{Useragent}$ und $M^{HTTP-Header}$ existieren bereits Schätzungen für die Entropie von 10 Bit beziehungsweise 6,09 Bit [56]. Diese Werte können zwar aufgrund der statistischen Abhängigkeiten nicht einfach addiert werden, wird aber eine kombinierte Entropie von 15 Bit angenommmen, wäre zu erwarten, dass ein zufälliger Nutzer nur mit einer Wahrscheinlichkeit von $\frac{1}{2^{15}} = 0,003\%$ vorkommt.

Über die Entropie von $M^{aktiv\backslash skript}$ sind keine Informationen bekannt, weswegen in Abschnitt 6.1.2 versucht wird, diese experimentell näher zu bestimmen.

Der Informationsgehalt des Deaktivierens von clientseitigen Skripten und Entropie des Merkmals $M^{noscript}$ kann aus dem Anteil der Nutzer berechnet werden, die Javascript deaktivieren. Im Jahr 2010 hatten ca. 99% der Nutzer Javascript aktiviert [12], wodurch der Infomationsgehalt des Deaktivieren von Skripten

$$-log_2(0,01) \approx 6,64 \text{ Bit}$$

und die Entropie von $M^{noscript}$

$$-(0,01 \cdot log_2(0,01) + 0,99 \cdot log_2(0,99)) \approx 0,08 \text{ Bit}$$

beträgt[1].

Die Entropie von M^{plugin} hängt davon ab, wieviele unterschiedliche Möglichkeiten es gibt, clientseitige Skripte zu deaktivieren und wie verbreitet diese sind. Dies wird in Abschnitt 6.1.2 genauer untersucht,

Selbst wenn das Deaktivieren von clientseitigen Skriptsprachen die Identifikation eines Nutzers verhindern kann, muss das Deaktiverens von Skripten auch sinnvoll möglich sein. Nutzt eine Webseite Skriptsprachen zu ihrer Darstellung oder zur Interaktion mit dem Nutzer, wird durch das Deaktivieren von Skripten die Benutzbarkeit der Webseite eingeschränkt oder die Webseite sogar unbenutzbar. Dieser Effekt kann leicht proviziert werden, indem Schlüsselkomponenten der Webseite über Skriptsprachen eingebunden werden. In diesem Fall steht der Nutzer vor der Wahl, Skriptsprachen zu aktivieren oder die Webseite nicht zu nutzen.

Ob ein einzelner Nutzer Skripte deaktivieren kann, hängt also davon ab, welche Seiten er besucht und wie groß seine Motivation ist, nicht verfolgt zu werden. Um dieses Problem zu verringern, unterstützen Plugins wie NoScript die Verwendung von Ausnahmeregeln, die die Ausführung von bestimmten Skripten erlauben. Kann ein Fingerprintingskript von einen solchen Ausnahme profitieren und wird ausgeführt, kann es ungehindert den Browserfingerprint erheben. Der Nutzer hat so gleichzeitig unterschiedliche Fingerprints abhängig davon, welche Webseite den Fingerprint misst. Es handelt sich also um kontextspezifischen Fingerprints, die in Abschnitt 4.2 behandelt werden.

Insgesamt kann das Deaktivieren von clientseitigen Skripten Schutz vor der Identifikation eines Browsers über Browserfingerprinting bieten, wenn der Informationgehalt von $\{M^{passiv\backslash skript}, M^{aktiv\backslash skript}, M^{noscript}, M^{plugin}\}$ gering genug ist. Die Entropie von $\{M^{passiv\backslash skript}, M^{aktiv\backslash skript}, M^{noscript}, M^{plugin}\}$ hängt dabei von der Entropie von $M^{aktiv\backslash skript}$ und M^{plugin} ab, die in Abschnitt 6.1.2 genauer untersucht wird. Die entstehenden Einschränkungen in der Benutzbarkeit des Browsers legen aber nahe, dass diese Maßnahme, selbst wenn sie technisch funktioniert, wegen mangelnder Usability nicht für eine

[1] Für den Zeitraum bis 2014 wurden leider keine neuen Zahlen gefunden, weswegen die Zahl von 2010 als Schätzung verwendet wird.

breite Nutzerbasis geeignet ist. Dass Plugins wie NoScript das Definieren von Ausnahmen für bestimmter Skripte anbieten, kann als Beleg für diese Einschätzung angesehen werden.

5.5. Uneingeschränkte Fälschung von Fingerprints

Das Fälschen eines Fingerprints gelingt für einen Fingerprintingalgorithmus, wenn dieser beim Browser des Fälschers dieselben Merkmale misst wie beim Original. Das Fälschen eines Fingerprints im Allgemeinen gelingt, wenn das Fälschen für alle Fingerprintingalgorithmen gelingt. Bei erfolgreichem Fälschen ist der Browser des Fälschers nicht von dem Original zu unterscheiden und eine globale Identifizierung des Fälschers über seinen Fingerprint ist unmöglich.

Durch das Fälschen von Fingerprints wird nur garantiert, dass sich der Fälscher in einem Anonymity-Set mit einem einzigen weiteren Browser befindet. Beinhaltet dieses nur Fälscher und ein weiteren Nutzer, wird lediglich 1 Bit zusätzlicher Informationsgehalt benötigt, um diese zu unterscheiden.

Da ein konkretes Fingerprintingskript zudem nur einen Teil aller Browser beobachtet, ist nicht sicher, ob Fälscher und Original von diesem beobachtet werden und das Fälschen auch in diesem Kontext vor einer Identifizierung schützt. Um dies wahrscheinlicher zu machen, sollte der Fingerprint von mehreren Nutzern gleichzeitig gefälscht werden.

Die größtmögliche Menge von Nutzern, deren Fingerprint gleichzeitig gefälscht werden kann, ist dabei durch das größte auftretende Anonymity-Set gegeben. Dieses hatte bei Eckersleys Studie eine Größe von 1 186, was $\frac{1186}{470161} \approx 0,25\%$ der gemessenen Browser entspricht. Der Informationsgehalt jedes von Eckersley gemessenen Fingerprints war also größer als $-log_2(\frac{1186}{470161}) \approx 8,631$ Bit und ein Fälscher hätte keinen Fingerprint mit weniger als $8,63$ Bit Informationsgehalt fälschen können.

Durch die Fälschung des Browserfingerprints wird aber die Zufallsverteilung, auf deren Basis der Informationsgehalt des Fingerprints berechnet wurde, verändert. Das Anonymity-Set hat sich um den Fälscher vergrößert und der neue Informationsgehalt des Fingerprints ist:

$$-log_2(\frac{|\text{Anonymity set}^{neu}|}{|Browser|}) = -log_2(\frac{|\text{Anonymity set}^{alt}| + 1}{|Browser|})\text{Bit}$$

Ein einzelner Fälscher hat zwar nur einen geringen Einfluss auf den Informationsgehalt, aber selbst ein geringer Anteil von Fälschern kann ein Anonymity-Set von beispielsweise 0,25 % der Browser deutlich vergrößern. Wählen viele Fälscher ihre Originale nach geringem Informationsgehalt aus, findet also zwangsweise eine Standardisierung statt, da diese Fälscher nicht nur Originale auf ähnliche Weise auswählen, sondern ein Original dadurch attraktiver wird, dass es oft gefälscht wird.

Damit ein Fälscher aber überhaupt eine solche Wahl treffen kann, benötigt er Informationen über zu fälschende Fingerprints und ihren Informationsgehalt. Erkenntnisse aus Eckersleys Studie wie das Wissen, welcher der Fingerprints den geringsten Informationsgehalt hat, sind dabei eher als Orientierung nützlich, da dieser Wert nicht mehr aktuell ist. Durch das Fälschen von Fingerprints und andere Faktoren ändert sich die Verteilung der Fingerprints ständig und muss permanent neu ermittelt werden. Die Fälscher müssen zudem Zugriff auf diese Informationen haben.

Die technische Durchführbarkeit einer solchen Fälschung ist natürlich Voraussetzung für das Fälschen von Browserfingerprints. Dabei muss zwischen der Fälschung bezüglich bekannter Fingerprintingalgorithmen und einer Fälschung im Allgemeinen unterschieden werden. Bei bekannten Fingerprintingalgorithmen hat der Fälscher den Vorteil, dass er nicht nur weiß, welche Merkmale er fälschen muss, sondern den Erfolg der Fälschung vor der Anwendung überprüfen kann.

Damit eine Fälschung erfolgreich ist, müssen alle erhobenen Merkmale des Browsers des Fälschers den Merkmalen des Originals entsprechen. Um die Merkmale des Originals zu duplizieren, kann der Zustand des Browsers verändert werden oder das Messen des Merkmals manipuliert werden. Die beiden Vorgehensweisen können kombiniert werden, indem zum Fälschen mancher Merkmale der Zustand des Browsers verändert und für andere die Messung manipuliert wird.

Die Möglichkeit, den Zustand des Originals nachzubilden, hängt dabei hauptsächlich von der Motivation des Nutzers ab, der je nach Fingerprintingalgorithmus spezielle Browser, Plugins, Schriften oder Betriebssysteme installieren oder beim Devicefingerprinting sogar spezielle Hardware nutzen müsste, um den Fingerprint des Originals nachzubilden. Dies ist aber von einem großen Teil der Nutzer nicht zu erwarten.

Werden Analysemethoden manipuliert, müssen alle genutzten beziehungsweise möglichen Analysemethoden getäuscht werden. Betrifft dieses Täuschen Merkmale, die von der Webseite genutzt werden, wie die Sprache, müssen diese ebenfalls gefälscht werden. Der Nutzer kann dadurch Nachteile in der Benutzbarkeit des Browsers erleiden, obwohl der Zustand des Browsers nicht verändert wird. Ob eine bekannte Analysemethode getäuscht werden kann, kann leicht überprüft werden. Das Täuschen von unbekannten Analysemethoden ist aber schwer zu garantieren.

Werden nicht alle Merkmale gefälscht, versagt nicht nur das Fälschen des Fingerprints, es entsteht auch ein neuer Fingerprint mit neuem Informationsgehalt. Wird der Zustand eines Browsers oder Analysemethoden manipuliert, kann ein inkonsistenter Fingerprint entstehen, der nicht durch die normale Nutzung eines Browsers entstehen kann. Von einem solchen Fingerprint ist ein sehr hoher Informationsgehalt zu erwarten.

Die Fälschung eines Fingerprints muss scheitern, wenn er Merkmale enthält, die aufgrund der individuellen Ansprüche des Nutzers oder des technischen Aufwand als unveränderlich gelten müssen. Dies gilt in ähnlicher Form, wenn unbekannte Merkmale und Analysemethoden in den Fingerprint einbezogen werden. Würde in diesem Kontext eine Fälschung versucht, würden solche Merkmale nicht gefälscht und die Analysemethoden nicht getäuscht. Die Fälschung würde also nicht gelingen, wenn die nicht gefälschten Merkmale nicht zufällig zum Original identisch sind.

Insgesamt erscheint das Fälschen von beliebigen Fingerprints als aufwendige, aber effektive Maßnahme gegen Browserfingerprinting mit bekannten Algorithmen. Gelingt das Fälschen, kann der Fingerprint mit dem geringsten Informationsgehalt angenommen werden, was bei wenigen Nutzern einen relativ schwachen Schutz gegen Browserfingerprinting bietet, der aber mit zunehmender Zahl von Fälschern effektiver wird. Der Fälscher muss dazu aber wissen, welche Fingerprints einen geringen Informationsgehalt haben und somit gute Kandidaten für Fälschungen sind. Ist ein Fingerprintingalgorithmus noch unbekannt oder verändert er sich, kann das Fälschen von Fingerprints fehlschlagen und einen Fingerprint mit hohem Informationsgehalt produzieren.

5.6. Eingeschränkte Fälschung von Fingerprints

Beim eingeschränkten Fälschen von Fingerprints wird nur ein Teil der Merkmale als veränderlich angenommen und es können deshalb nur bestimmte Fingerprints gefälscht. Dadurch ist dies ein Spezialfall der Fälschung von Fingerprints, die in Abschnitt 5.5 untersucht wurden. Die Grundannahmen gelten also weiterhin und zur Fälschung eines Fingerprints sollten alle Merkmale bekannt und veränderbar oder mit dem Original identisch sein. Weiterhin muss ein Fingerprint mit geringem Informationsgehalt gefälscht werden, damit dies verlässlich vor einer Identifizierung schützt.

Durch die nicht fälschbaren Merkmale M_i^{fest} ist der Raum der fälschbaren Fingerprints eingeschränkt. Innerhalb dieses Raumes eignet sich jeder Fingerprint für einen Fälschungsversuch. Versagt dieser, hat der Browser einen inkonsistent gefälschten Fingerprint mit vermutlich hohem Informationsgehalt.

Die Auswahl des zu fälschenden Fingerprint findet also anhand von M_i^{fest} statt und erlaubt nur Fingerprints, die sich aus M_i^{fest} und einer beliebigen Kombination aus fälschbaren Merkmalsausprägungen zusammensetzen. Der Informationsgehalt von M_i^{fest} bestimmt dabei die Menge der möglichen Originale und ein Browser ist mit der Wahrscheinlichkeit $2^{(-I(M_i^{fest}))}$ ein fälschbares Original. Damit erwartet werden kann ein fälschbares Originale zu einem M_i^{fest} zu finden, müssen also $2^{I(M_i^{fest})}$ zufällige Fingerprints bekannt sein. Mit anderen Worten, es ist schwieriger, einen fälschbaren Fingerprint zu finden, je mehr Merkmale unveränderlich und je seltener sie sind.

Wenn mehrere geeignete Fingerprints gefunden werden, kann eine Auswahl aus diesen erfolgen, wobei

der Fingerprint mit dem größten Anonymity-Set den geringsten Informationsgehalt produziert. Bei der freien Auswahl aus allen Fingerprints konnte in Eckersleys Studie, wie in Abschnitt 5.5 beschrieben, im Idealfall ein Anonymity-Set von 1 186 Browsern und damit einen Informationsgehalt von 8,63 Bit erreicht werden. Kann nur der Fingerprint mit dem zehntgrößten Anonymity-Set gefälscht werden, verringert sich das Anonymity-Set bei Eckersley bereits auf 618 Browser. Ein Fingerprint, der nicht vorgibt Javascript zu deaktivieren und nicht vorgibt der Fingerprint eines Mobiltelefons zu sein, kommt auf ein Anonymity-Set von nicht mehr als 97 Browsern. Fällt der ausgewählte Fingerprint nicht unter die tausend Fingerprints mit dem größten Anonymity-Set, hat dieser ein Anonymity-Set von unter 10 Nutern und hätte damit in Eckersleys Studie einen Informationsgehalt von mindestens $-log_2(\frac{10}{|\text{Browser}|}) = -log_2(\frac{10}{470161}) = 15.5$ Bit. Diese Werte sind aber nur beispielhaft, da die von Eckersley gemessenen Figerprints nicht mehr aktuell sind.

Das Finden eines Fingerprints mit niedrigem Informationsgehalt stellt also selbst dann ein Problem dar, wenn alle Browserfingerprints bekannt sind. Wie beim uneingeschränkten Fälschen von Fingerprints findet aber eine Standardisierung statt, indem oft gefälschte Werte an Attraktivität gewinnen.

Ein Vermittlungsserver, der als Informationsquelle für Fälschungen dient, kann dabei kein Wissen über die tatsächliche Verteilung aller Fingerprints haben und muss sich auf eine Abschätzung dieser Verteilung auf Basis der ihm bekannten Fingerprints stützen. Die Datenbank von Fingerprints, aus denen der Vermittlungsserver Fingerprints auswählt und Abschätzungen vornimmt, kann auf verschiedene Weisen aufgebaut werden:

- Die Nutzer könnten ihren Browserfingerprint bewusst zur Fälschung vorschlagen. Dies würde einer Weitergabe des Fingerprints zur Verschleierung der eigenen Handlungen oder der Spende des Fingerprints entsprechen. Dabei kann eine Einwilligung zur Verarbeitung und Fälschung angenommen werden, es müssten aber sehr viele Spender gewonnen werden, um eine befriedigende Auswahl an Fingerprints zu erhalten.
- Der Vermittlungsserver könnte zwangsweise die Fingerprints aller Browser, die von ihm Informationen abfragen, messen und diese Fingerprints zu Fälschung anbieten. Dies könnte als Bedingung zur Nutzung des Vermittlungsservers an die Nutzer kommuniziert und deren Zustimmung auf diese Weise eingeholt werden. Auf diese Weise könnten so viele Fingerprints wie Nutzer erhoben werden. Der hohe Anteil an Fälschungen, der dabei zu erwarten ist, verzerrt aber die Analyse des Informationsgehalts der Fingerprints, da die gemessene Stichprobe nicht repräsentativ ist.
- Der Vermittlungsserver könnte Fingerprints aus externen Quellen beziehen, indem er zum Beispiel auf Webseiten eingebunden wird und die Fingerprints der Nutzer dieser Webseiten misst. Dieses Vorgehen ähnelt aber stark dem Browserfingerprinting zur Verfolgung von Nutzern, obwohl es eine Gegenmaßnahme ist. Der Nutzer kann hierbei nicht nachprüfen, wie die erhobenen Daten verwendet werden, und ein Fälschen eines Fingerprints kann Einfluss auf den Nutzer des Originals haben, da die Identitäten dieser vermischt werden. Aus diesem Grund kann wie beim Browserfingerprinting zur Nutzerverfolgung keine implizite Erlaubnis zum Nutzen dieser Fingerprints angenommen werden. Würde eine Möglichkeit gefunden werden, dies mit Zustimmung der Nutzer durchzuführen, könnten so leicht große Mengen an Fingerprints erhoben werden und Schätzungen zu dem Informationsgehalt von Fingerprints durchgeführt werden.

Bei allen Möglichkeiten, diese Datenbank aufzubauen, muss darauf geachtet werden, dass die Datenbank der Fingerprints sich nicht einfach erweitern darf, sondern aktuell gehalten werden muss, damit keine veralteten Fingerprints die Vorschläge verzerren. Dies kann erreicht werden, indem Fingerprints, die eine gewisse Lebenszeit überschritten haben, aus der Datenbank gelöscht werden.

Da M_i^{fest} sich durch die Fälschung nicht verändert, muss das Original auch M_i^{fest} enthalten und einen Informationsgehalt von mindestens $I(M_i^{fest})$ haben. M_i^{fest} begrenzt also nicht nur durch das Reduzieren der Auswahl den Informationsgehalt des Originals, sondern gibt auch seinen kleinstmöglichen Informationsgehalt an, da er in diesem enthalten ist.

Insgesamt erscheint das eingeschränkte Fälschen von Fingerprints als eine geeignete Möglichkeit, die globale Identifizierbarkeit eines Browsers zu verhindern. Die Sicherheit gegen eine Identifizierung inner-

halb bestimmter Nutzergruppen oder gegen Fingerprinting unter Nutzung von zusätzlichen Informationen ist aber bei einer geringen Anzahl von Fälschern nur schwach, da zu erwarten ist, dass der gefälschte Fingerprint nur ein kleines Anonymity-Set hat. Damit ein zur Fälschung geeigneter Fingerprint oder sogar ein Fingerprint mit kleinem Informationsgehalt gefunden werden kann, müssen die unveränderlichen Merkmale einen möglichst geringen Informationsgehalt aufweisen. Der Aufbau der Datenbank der Fingerprints des Vermittlungsservers ist ebenfalls entscheidend, um fälschbare Fingerprints zu finden und zu bewerten. Dabei sind allerdings nicht nur praktische, sondern auch ethische Probleme zu bedenken, da das Fälschen eines Fingerprints andere Nutzer betreffen kann, wenn ihnen fremde Aktionen zugeordnet werden.

5.7. Blockieren von Kommunikation

Wird die Kommunikation eines Browsers zum Analyseserver blockiert, weiß dieser nicht einmal, dass er einen Browserfingerprint erstellen sollte, und eine Nutzerverfolgung ist unmöglich. Scheitert das Blockieren kann die Analyse ungehindert stattfinden.

Um zu entscheiden, ob die Kommunikation zu einem Server blockiert werden sollte, ist es einerseits notwendig zu wissen, ob er Fingerprinting betreibt, und andererseits, ob der Nutzer auf diesen Server zugreifen will. Wird ein Analyseserver nicht blockiert, da er beispielsweise rein passives Fingerprinting betreibt oder noch nicht in die Blockadelisten eingetragen wurde, bleibt der Nutzer ungeschützt. Wird ein Server blockiert, der für den Nutzer nützlich ist, kann er nicht mehr in Anspruch genommen werden und der Nutzer erleidet dadurch Nachteile. Der Aufbau und Pflege der Blockadelisten sind also Schlüsselkomponenten für die Wirkung und Akzeptanz dieser Gegenmaßnahme.

Ein Blockieren der Kommunikation kann dadurch behindert werden, dass keine dedizierten Analyseserver genutzt werden, sondern der Analysecode und nützliche Inhalte gemeinsam auf einem Server gehostet werden. So müssen entweder Einbußen in der Benutzbarkeit des Browsers hingenommen werden oder das Fingerprinting zugelassen werden.

Insgesamt ist das Blockieren von Kommunikation also eine sehr effektive, aber sehr unzuverlässige Maßnahme gegen Browserfingerprinting, da bekannt sein muss, welche Kommunikation blockiert werden muss, und die Listen, nach denen blockiert wird, stets aktuell gehalten werden müssen.

5.8. Filtern von Kommunikation

Wird die Kommunikation zu einem Analyseserver gefiltert, wird trotzdem mit dem Analyseserver kommuniziert und es kann ein Fingerprint erhoben werden. Dieser Fingerprint setzt sich aus den nicht herausgefilterten Merkmalen M_i^{Rest} und der Art und Weise des Filterns M_j^{Filter} zusammen.

Damit das Herausfiltern eines Browserfingerprintingskriptes überhaupt erfolgreich sein kann, müssen die Regeln, nach denen gefiltert wird, das Fingerprintingskript oder dessen Kommunikation beschreiben können. Auch müssen die Filterlisten aktuell sowie umfassend sein. Über diese Anforderungen kann versucht werden das Pflegen dieser Regeln zu erschweren, indem die Namen der Skripte oft geändert werden oder andere Verschleierungsmethoden benutzt werden.

Damit eine Filterregel die Benutzbarkeit der Browser nicht beeinträchtigt, darf diese keine Skripte oder Kommunikation der Webseite filtern, die vom Nutzer erwünscht sind. Es kann versucht werden, diese Einbußen bei der Benutzbarkeit des Browsers zu provozieren, indem die nützlichen und die zum Fingerprinting genutzten Teile der Webseiten miteinander vermischt werden. Gelingt dies, muss entweder das Fingerprinting zugelassen oder die Benutzbarkeit des Browsers eingeschränkt werden, was die Nutzer dazu bringen könnte, das Filtern von Kommunikation zu deaktivieren.

Das erfolgreiche Filtern von Kommunikation ist aber nicht ausreichend, um die Identifizierung eines Browsers durch Browserfingerprinting zu verhindern, denn der Informationsgehalt des Fingerprints, der trotz

Filtern erhoben werden kann, könnte ausreichen, um Browser zu Identifizieren. Hat der Browser beispielsweise einen einzigartigen Useragent oder filtert der Browser als einziger die Kommunikation, ist dieser dadurch identifizierbar. Um dies einzuschätzen, wird die Entropie seiner Komponenten M^{Rest} und M^{Filter} berachtet.

Der zwangsläufig erhebbare Fingerprint M^{Rest} enthält das passive Fingerprinting M^{passiv} und muss also mindestens dessen Entropie besitzen. M^{passiv} enthält $M^{HTTP-Header}$ und $M^{Useragent}$, deren Entropie von Eckersley auf mindestens 6,09 Bit und 10 Bit geschätzt wurde. Ist trotz Filtern eine aktive Analysemethode möglich, können wesentlich mehr Merkmale erhoben werden und die Entropie von M^{Rest} kann stark steigen.

Die Entropie von M^{Filter} ist von der Anzahl der Nutzer, die ihre Kommunikation filtern und den Unterschieden in der Form des Filtern der Kommunikation abhängig. Wird das Filtern in Werbeblocker integriert oder ist es nicht vom Deaktivieren von clientseitigen Skripten zu unterscheiden, kann deren Anonymity-Set übernommen werden und der Informationsgehalt der Fingerprints würde sinken. Kann so erreicht werden, dass 5 % der Browser den Traffic auf dieselbe Weise filtern, hätte diese Merkmalsauprägung ein Informationsgehalt von $-log_2(5\%) = 4,322$ Bit.

Ein Nachteil des Filtern der Kommunikation ist, dass es nicht explezit als solcher Erkannt werden muss. Selbst mit einfachen Auswertungslogiken wird das Fehlen von Werten registriert und $I(M^{Filter})$ kann selbst von naiven Fingerprintingskripten gemessen werden.

Es ist also zu erwarten, dass trotz Filterns der Informationsgehalt der Fingerprints bei einem großen Teil der Browser ausreicht, um sie zu identifizieren. Allerdings ist es auch zu erwarten, dass der durchschnittliche Informationsgehalt der Fingerprints der betroffenen Browser durch das Filtern der Kommunikation sinkt, wenn Merkmale, wie die installierten Schriften und Plugins, die von Eckersley auf 13,9 Bit und 15.4 Bit geschätzt werden, nicht mehr gemessen werden können, Kann der ganze Fingerprint gemessen werden, da das Filtern des Browserfingerprintingskriptes erkannt und umgangen wird, erhöht sich der Informationsgehalt der Fingerprints sogar zwangsweise. Ob das Filtern von Kommunikation gegen Browserfingerprinting schützt, hängt also von der Qualität und der Vollständigkeit der Filterregeln ab und eine permanente Betreuung dieser Filterregeln ist nötig.

5.9. Kontextspezifische Fingerprints

Soll die Nutzerverfolgung durch Browserfingerprinting verhindert werden, indem in jedem Kontext ein anderer Fingerprint präsentiert wird, darf der Nutzer nicht über diese Kontexte hinweg verfolgbar sein. Innerhalb eines Kontextes wird die Nutzerverfolgung dadurch nicht eingeschränkt.

Relevante Kontexte, in denen es Gelegenheiten gibt, den Fingerprint zu messen, sind

- Seitenaufrufe auf verschiedenen Domains,
- Login-Vorgänge,
- Seitenaufrufe auf einer Domain in mehreren Browsingsessions,
- Seitenaufrufe auf einer Domain innerhalb einer Browsingsession und
- mit privaten Daten verknüpfte Seitenaufrufe.

Da Datensammlungen in Bezug auf den Fingerprint als Pseudonym angelegt werden können, sind für den Datenschutz mit privaten Daten verknüpfte Seitenaufrufe besonders kritisch. Erlauben diese zum Beispiel eine Verknüpfung des Fingerprints mit dem Realnamen, kann dieser auch mit dem Seitenaufruf verknüpft werden. Diese Fingerprints können zu einem Fingerprint kombiniert werden oder ein neuer Fingerprint aus diesen zu berechnet werden. Dadurch entsteht ein neuer Identifizierer, der dazu dienen könnte, bisher nicht verknüpfbare Kontexte zu verknüpfen oder der Nutzer wiederzuerkennen. Damit der Fingerprint dazu genutzt werden kann, muss er stabil sein, also immer wieder gemessen werden können und sich nicht ver-

ändern, und einen hohen Informationsgehalt haben. Ist der Fingerprint stabil und dessen Informationsgehalt groß genug, kann der Nutzer global identifiziert werden.

Der Nutzer kann immer noch über verschiedene Kontexte hinweg verfolgt werden, wenn dies über zusätzliche Informationen geschieht oder die in verschiedenen Kontexten präsentierten Fingerprints verknüpfbar sind. Ansonsten ist die Maßnahme erfolgreich. Das in Abschnitt 5.2 untersuchte Randomisieren ist ein Spezialfall dieses Prozesses.

Kontexte können über Namen der Nutzer, Email-Adressen, Nutzeraccounts, IP-Adresse oder andere Identifizierer verknüpft werden. Beispielsweise bietet ein Vorgehen, bei dem für jede Browsingsession oder für Zugriffe auf bestimmte Domains ein anderer Fingerprint präsentiert wird, keinen sicheren Schutz, wenn diese Kontexte mit einem bestimmten Nutzeraccount in Verbindung gebracht werden können. Durch die zusätzlichen Informationen ist zwar bereits die Verfolgung des Kontextwechsels möglich, es können nun aber mehrere unterschiedliche Fingerprints des Browsers erhoben werden.

In verschiedenen Kontexten verschiedene Fingerprints zu präsentieren, ist eine effiziente Methode, die Nutzerverfolgung über Kontextwechsel zu verhindern, die aber bei Versagen den Schutz gegen Browserfingerprinting schwächt. Innerhalb des Kontextes wird die Nutzerverfolgung allerdings nicht behindert und eine Verknüpfung von Kontexten durch zusätzliche Informationen kann einen neuen, besseren Pseudoidentifizierer erzeugen. Für einen guten Schutz muss also darauf geachtet werden, dass Kontextwechsel vom Fingerprinter nicht beobachtbar sind.

5.10. Kombination verschiedener Maßnahmen

Die Kombination verschiedener Maßnahmen kann mit verschiedenen Zielen verfolgt werden. Erstens kann versucht werden Maßnahmen so zu kombinieren, dass eine Identifizierung durch Browserfingerprinting beweisbar ausgeschlossen wird. Das zweite mögliche Ziel ist, Maßnahmen so zu kombinieren, dass sie das Browserfingerprinting behindern ohne es sicher zu verhindern. Bei der Kombination von Maßnahmen muss die Möglichkeit betrachtet werden, dass der Schutz vor Browserfingerprinting durch eine Kombination von Maßnahmen auch abgeschwächt werden könnte.

Um dies zu betrachten, wird in den folgenden Abschnitten jeweils eine Gegenmaßnahme herausgegriffen und die Kombination dieser mit anderen Maßnahmen untersucht.

5.10.1. Standardisieren von Browsern

Das Standardisieren ist ein guter Weg, die Entropie der Fingerprints zu senken, dessen Effektivität allerdings durch die unveränderlichen Merkmale begrenzt wird. Dieser Effekt tritt auch in der Kombination mit allen anderen Gegenmaßnahmen auf.

Wird auf die Nutzung von weiteren Gegenmaßnahmen standardisiert, wird das Schutzparadox für diese Maßnamen gemildert. Es wird jedoch verstärkt, wenn auf den Verzicht vom Gegenmaßnahmen standardisiert wird.

Dies gilt auch für den Spezialfall der automatische Browserupdates, die eine schwache Standardisierung darstellen. Der dabei gesetzte Standard betrifft allerdings nur die Browserversion, von der anzunehmen ist, dass sie nur selten andere Gegenmaßnahmen ausschließt und sich somit gefahrlos einsetzen lässt.

Das Standardisieren von Browsersern lässt sich also mit allen anderen untersuchten Maßnahmnen kombinieren.

5.10.2. Fälschen von Fingerprints

Das Fälschen von Fingerprints ist eine aufwendige Technik, von der aber bei geringer Teilnahme nur ein kleines Anonymity-Set erwartet werden kann. Selbst wenn Fingerprints uneingeschränkt gefälscht werden

können oder nur wenige Merkmale unveränderbar sind, kann der Browser immer noch von einem Großteil der Nutzer unterschieden werden. Der Effekt des Fälschens von Fingerprints kann aber durch die Kombination mit anderen Maßnahmen verstärkt und das geringe Anonymity-Set der Fälschung ausgeglichen werden.

Das Randomisieren von Fingerprints kann mit dem Fälschen von Fingerprints kombiniert werden, indem die Auswahl des zu fälschenden Fingerprints nicht nach einem festem Schema, sondern zufällig geschieht.

Die Anzahl von Fingerprints, die dabei zur Verfügung stehen, ist dabei wichtig, um bei jeder Gelegenheit einen neuen Fingerprint präsentieren zu können. Sind nur wenige Fingerprints verfügbar, kann der Browser unter diesen Fingerprints verfolgt werden, da sie immer wieder benutzt werden müssen. Haben diese Fingerprints ein großes Anonymity-Set, fällt die Verfolgung mehrerer Nutzer zusammen und sie wird durch falsche Verknüpfungen erschwert. Das Verhalten des Browsers wird also mit den vereinigten Anonymity-Sets aller gefälschten Fingerprints in Verbindung gebracht. Je geringer der Informationsgehalt der unveränderlichen Merkmale ist, desto wahrscheinlicher wird es, viele fälschbare Fingerprints mit großem Anonymity-Set zu finden.

Ein auf diese Weise randomisierter Fingerprint wäre auch schwer als solcher zu erkennen, da die Auswahl aus echten Fingerprints verspricht, keine unglaubwürdigen Merkmalskombinationen zu erzeugen. Würde diese Randomisierung trotzdem erkannt, müssten zur Bestimmung eines nützlichen Fingerprints alle fälschbaren Merkmale erkannt werden und es bliebe lediglich die Menge der unveränderlichen Merkmale zur Identifizierung über. Der Informationsgehalt dieser Merkmale sollte aber zum Zweck der Fälschung allerdings sowieso gering sein.

Das Fälschen aller Fingerprints, also das Fälschen und Nutzen aller genutzten Fingerprints könnte theoretisch sogar k-Anonymität garantieren. Läuft ein Fälscher alle Fingerprints einmal ab und lässt sich mit jedem Fingerprint einmal messen, hat jeder Fingerprint den Fälscher in seinem Anonymity-Set und somit eine k-Anonymität von 1. Dadur wäre der Nutzer nicht mehr global indentifizierbar. Würden mehere Fälscher dies tun, würde jedes Anonymity-Set diese Fälscher enthalten und die k-Anonymität entsprechend steigen. Dass ein Fälscher alleine alle Fingerprints einmal fälschen kann, ist aber nicht anzunehmen. Einer großen Gruppe von Fälschern könnte es allerdings auf diese Weise gelingen, eine k-Anonymität für einen Teil der Fingerprints zu garantieren.

Das Filtern, Blockieren von Kommunikation oder Deaktivieren von Javascript verhindert das Ausführen von Fingerprintingskripten teilweise oder komplett. Dies reduziert die erhebbaren Merkmale, fügt aber das Fehlen von Werten als Merkmale neu hinzu. Eine Kombination dieser Maßnahmen mit einer Fälschung würde bedeuten, dass gefälschte Werte zurückgegeben werden, wo sonst Werte fehlen würden.

Dazu kann die Kommunikation zusätzlich gefälscht und ein gefälschtes Analyseergebnis zurückgegeben werden, wenn die verhinderte Analysemethode bekannt ist. Arbeiten Original und Fälscher zusammen das Analyseergebnis des Originals sogar vom Original selbst an den Fälscher oder die Fälscher weitergegeben werden. Das Fälschen und Weitergeben von Analyseergebnissen kann erschwert werden, der Analysecode kann aber immer beobachtet werden und zumindest theoretisch können auch Fälschungsstrategien gefunden werden. Da alle gefilterten Merkmale so fälschbar sind, würde die Wahrscheinlichkeit gesteigert, viele fälschbare Fingerprints zu finden.

Ist die verhinderte Analysemethode allerdings nicht bekannt, kann das Analyseergebnis nicht gefälscht werden. Das Blockieren von Kommunikation und das Fälschen von Fingerprints ergänzen sich trotzdem gut, da hier im Einzelfall sogar die Wahl zwischen den beiden Strategien besteht. Das Filtern von Kommunikation und Deaktivieren von Javascript ist allerdings selbst ein Merkmal des Fingerprints und kann die Wahrscheinlichkeit senken, viele fälschbare Fingerprints zu finden. Arbeiten bekannte und unbekannte Fingerprintingskripte zusammen kann dies sogar zusätzliche Analysen erlauben.

Ein Ansatz für das Fälschen von Analyseergebnissen wären Surrogate Scripts in NoScript, da sie erlauben bestimmte Skripte nicht nur zu blockieren, sondern auch durch selbst definierte Skripte zu ersetzen. Es

müssten dazu allerdings nicht nur die Filter- und Blockadelisten betreut werden, sondern die Analyseskripte untersucht und Fälschungsmethoden implementiert werden. Sollen Analyseergebnisse über den Vermittlungsserver ausgetauscht werden, müsste dies dem Koordinationsprotokoll hinzugefügt werden. Gegenüber unbekannten oder sich ändernden Fingerprintingalgorithmen kann diese Kombination allerdings einer Identifizierung zuträglich sein.

Eine Standardisierung der Fingerprints geht mit der massenhaften Fälschung von Fingerprints einher. Eine zusätzliche Standardisierung der Browser vergrößert aber die Anonymity-Sets der Fingerprints und damit auch die Chance, einen fälschbaren Fingerprint mit großem Anonymity-Set zu finden. Auch der Informationsgehalt der unveränderlichen Merkmale kann durch eine Standardisierung verringert werden, was die Wahrscheinlichkeit erhöht, fälschbare Fingerprints zu finden.

5.10.3. Deaktivieren von clientseitigen Skripten

Die Effektivität des Deaktivierens von clientseitigen Skripten beruht darauf, dass nur wenige Informationen über den Browser preisgegeben werden und viele Browser die clientseitigen Skripte deaktivieren. Würde die Nutzerbasis vergrößert und der Informationsgehalt der preisgegebenen Information verringert, würde dies die Effektivität dieser Maßnahme weiter steigern. Dies könnte durch Kombination mit weiteren Maßnahmen erreicht werden.

Die Benutzbarkeit der Browser mit deaktivierten Skripten zu verbessern, hätte das Potenzial, die Nutzerbasis zu vergrößern, die clientseitige Skripte deaktiviert. Eine Möglichkeit, dies zu erreichen, wäre das Erlauben von harmlosen Skripten, ohne das eine Nutzerinteraktion notwendig ist. Das Klassifizieren eines Skriptes als harmlos könnte durch Auditoren oder eine automatisierte Codeanalyse geschehen und ein Wiedererkennen eines solchen Skriptes könnte über einen Hash-Abgleich geschehen.

Alternativ kann auch wie bei dem Plugin FPGuard eine automatische Erkennung von Browserfingerptintingskripten genutzt werden, um nur diese zu blockieren [24] oder wie bei den Surrogate Scripts in NoScript können Skripte ersetzt statt bockiert werden.

Das passive Browserfingerprinting kann trotz des Deaktivierens von clientseitigen Skripten durchgeführt werden und bestimmt somit den minimalen Informationsgehalt. Diese Merkmale enthalten, wie in Abschnitt 5.4 untersucht, beispielweise den Useragent, der eine Entropie von mindestens 10 Bit hat. Durch Fälschungen dieser Merkmale könnten Merkmalsausprägungen mit einem geringen Informationsgehalt gewählt oder durch Standardisieren dieses Merkmals die Entropie des passiven Browserfingerprintings reduziert werden.

5.10.4. Automatisiertes Kombinieren von Maßnahmen

Welche Maßnahmen sinnvoll miteinander kombinierbar sind, kann von dem jeweiligen Kontext abhängen und schwer abschätzbar sein. Bei der breiten Masse der Nutzer kann also nicht davon ausgegangen werden, dass sie diese Abschätzung durchführen. Dies könnte den Nutzern abgenommen werden, indem ein Browserplugin eingestzt wird, das den Browser analysiert und dem Nutzer sinnvolle Maßnahmen gegen das Browserfingerprinting vorschlägt und auf Wunsch umsetzt.

Ob es sinnvoll ist, Gegenmaßnahmen gegen Browserfingerprinting einzusetzen, ist oft von Kontexten abhängig und es ist unklar, ob das Kombinieren von Maßnahmen die Wahrscheinlichkeit einer Identifikation verringert oder erhöht. Dies hängt dabei unter anderem von der aktuellen Verteilung der Fingerprints und dem Browser selbst ab. Für die Entscheidung, eine Maßname einzusetzen, muss das Plugin nicht nur die Fähigkeit haben, Maßnahmen zu aktivieren oder zu deaktivieren, sondern muss auch entscheiden können, ob der Einsatz von Maßnahmen sinnvoll ist und ob die Maßnahmen für den Nutzer akzeptabel sind.

Die Effektivität vieler Maßnahmen hängt von der aktuellen Verteilung der Browser ab, da das Schutzparadox verhindern kann, dass Maßnahmen gegen Browserfingerprinting schützen. Eine Schätzung über die Verteilung der Browser wird also benötigt und könnte über Messungen der Fingerprints von Webseitenbesuchern erhoben werden. Diese könnte zentral betreut und vom Plugin abgefragt werden. Das Erheben von Eigenschaften des Browsers, die nicht über Analysescripte erhoben werden können, könnte über das Plugin selbst geschehen.

Informationen über Merkmale, die aus technischen Gründen nicht veränderbar sind, oder Maßnamen, die aus technischen Gründen nicht anwendbar sind, können zusammen mit dem Plugin ausgeliefert werden. Welche Merkmale aus Benutzbarkeitsgründen nicht veränderbar sind, könnte vom Nutzer erfragt werden.

Mit diesem Informationen könnte das Plugin abschätzen, welche Kombination von Maßnahmen umsetzbar und sinnvoll sind, und könnte diese durchführen.

5.11. Zusammenfassung

In diesem Kapitel wurden verschiedene Maßnahmen gegen das Browserfingerprinting analysiert. Dabei wurden möglichst allgemeingültige, also von Trends auf dem Browsermarkt unabhängige Aussagen getroffen.

Als Erstes wurde in Abschnitt 5.1 **das Standardisieren von Browsern** untersucht. Dieses könnte das Browserfingerprinting komplett verhindern, wenn alle Browser alle ihre Merkmale standardisieren würden. Standardisieren alle Browser nur ein Merkmal reduziert sich der Informationsgehalt und Entropie der Fingerprints garantiert.

Dass alle Browser an einer Standardisierung teilnehmen, kann aber nicht erwartet werden, wodurch beim Standardisieren darauf zu achten ist, dem nahezukommen und keine Anonymity-Sets zu trennen. Dies wäre beispielsweise gewährleistet, wenn ein Browserhersteller die Standardisierung in allen Browsern eines Typs erzwingt.

Das Standardisieren ist nicht nur durch technische Probleme, sondern auch dadurch beschränkt, dass Nutzer verschiedene Interessen haben und deshalb vorausichtlich nur ein begrenztes Maß an Standardisierung akzeptieren.

Das Randomisieren von Fingerprints wurde in Abschnitt 5.2 untersucht. Obwohl bestätigt werden konnte, dass das Grundprinzip bei naiven Fingerprintern funktioniert, wurden Methoden gefunden, es anzugreifen.

Die erste Möglichkeit dazu ist gegeben, wenn der Fingerprint nicht nur einmal, sondern mehrfach gemessen wird und so die Randomisierung beobachtet werden kann. Dieser Fall wurde in der Arbeit von Nikiforakis [47] nicht betrachtet, so ist ein solcher Angriff bisher auch nicht bekannt geworden. Kann von einem Fingerprinter erkannt werden, auf welche Weise ein Fingerprint randomisiert wird, ist dies ein Merkmal mit vermutlich hohem Informationsgehalt. In Kombination mit dem nichtrandomisierten Anteil des Fingerprints kann dies randomisierendee Browser leicht identifizierbar machen. Ein Randomisierer sollte also darauf achten, dass das Randomisieren nicht beobachtbar ist und viele Merkmale, aber diese nur selten, randomisiert werden.

Die zweite Möglichkeit, trotz Randomisierung verlässliche Fingerprints zu erheben, ist das Ignorieren der randomisierten Merkmale. Das kann für alle Nutzer geschehen oder auch auf Randomisierer angepasst, falls sich diese beispielsweise über die Liste der installierten Plugins verraten. Die randomisierten Merkmale können so gar nicht verwertet werden, was das Browserfingerprinting in jedem Fall stark behindert. Je mehr Merkmale randomisiert werden, desto stärker wird ein solches Browserfingerprintingskript behindert. Auch diese Methode wurde bisher nicht wissenschaftich untersucht.

Als in den meisten Fällen unbegründet hat sich die Befürchtung herausgestellt, dass ein Nutzer unter mehreren Nutzerprofilen verfolgt wird, da nur zwischen wenigen möglichen Fingerprints gewechselt wird. Bei der Kombination aus Randomisierung und Fälschung wird dieser Fall aber wieder relevant, da nur wenige Fingerprints zur Verfügung stehen.

Korrekt eingesetzt oder bei naivem Browserfingerprinting funktioniert das Randomisieren von Fingerprints also. Dass für einen sicheren Schutz viele Merkmale nur selten randomisiert werden sollten, ist nicht naheliegend und wurde in bisherigen Arbeiten auch nicht beachtet. Zusätzlich werden durch das Randomisieren von Fingerprints nur die Nutzer von Randomisierern geschützt und die Situation der restlichen Nutzer könnte sich verschlechtern, da die Randomisierer nicht plausible Fingerprints annehmen.

Die in Abschnitt 5.3 untersuchten **automatischen Browserupdates**, haben sich als ein Spezialfall des Standardisierens von Browserfingerprints erwiesen. Da dieses fast alle Browser eines Types standardisiert, reduziert dies verlässlich die Entropie der entsprechenden Fingerprints. Der Effekt ist allerdings dadurch begrenzt, dass viele wichtige Merkmale des Browserfingerprintings nicht durch die Browserversion bestimmt werden.

Das Deaktivieren von clientseitigen Skriptsprachen wurde in Abschnitt 5.4 stellvertretend für das Verheimlichen von Merkmalen untersucht. Bei dieser Maßnahme gegen Browserfingerprinting wurden zwei Probleme gefunden.

Erstens hängt von der Menge der teilnehmenden Nutzer ab, wie gut es vor Browserfingerprinting schützt. Haben die weiterhin erhebbaren Merkmale und das Verheimlichen von Merkmalen einen geringeren Informationsgehalt als der ursprüngliche Fingerprint, schützt dieses vor Browserfingerprinting, ansonsten ist der Nutzer durch diese Gegenmaßnahme sogar besser zu identifizieren.

Die bisherigen Studien legen nahe, dass das Deaktivieren von Skriptsprachen den Informationsgehalt der Fingerprints reduziert, er aber weiterhin groß bleibt. An dieser Stelle kann keine abschließende Bewertung vorgenommen werden, da in diesen Studien kein aktives Browserfingerprinting ohne Einsatz von Skriptsprachen untersucht wurde. Die dafür gegebenen Möglichkeiten werden in Abschnitt 6.1.2 untersucht.

Der Schutz dieser Maßnahme kann noch verbessert werden, indem mehr Nutzer an ihr teilnehmen. Dies führt zum zweiten Problem des Deaktivierens von Skriptsprachen. Die starken Einschränkungen in der Benutzbarkeit der Browser sorgen dafür, dass vermutlich ein Großteil der Nutzer nicht davon überzeugt werden kann, Javasript zu deaktivieren. Auch das Nutzen von Ausnahmeregelungen für bestimmte Skripte kann den Schutz vor Browserfingerprinting verschlechtern, wenn dadurch Browserfingerprintingskripte ausgeführt werden.

Das Deaktivieren von Skriptsprachen ist also keine geeignete Maßnahme gegen die massenhafte Nutzerverfogung durch Browserfingerprinting.

In Abschnitt 5.5 wurde untersucht, ob **das uneingeschränkte Fälschen von Fingerprints** genutzt werden kann, um vor Browserfingerprinting zu schützen. Dies hat unabhängig von der technischen Umsetzbarkeit nur einen begrenzten Effekt, da im besten Falle lediglich der Fingerprint mit dem geringsten Informationsgehalt dupliziert werden kann. Als Orientierung wurde Eckersleys Studie herangezogen, in welcher der beste Fingerprint 6,64 Bit Informationsgehalt hatte und 0,25% der Fingerprints stellte. Dies verhindert zwar eine eindeutige Identifizierung, erlaubt aber weiterhin den Fälscher von vielen Nutzern zu unterscheiden. Fälschen allerdings viele Teilnehmer ihren Fingerprint, verbessert sich dieser Wert.

Das eingeschränkte Fälschen von Fingerprints wurde in 5.6 untersucht. Bei diesem ist neben der technischen Umsetzbarkeit das Hauptproblem, dass es sehr schwierig ist, gute fälschbare Fingerprints zu finden. Können wichtige Teile des Fingerprints nicht gefälscht werden, wird eine große Datenbank von Fingerprints benötigt, um überhaupt fälschbare Fingerprints zu finden. Zum Aufbau einer solchen Datenbank wurden mehrere Vorschläge gemacht, wobei die effektivsten Vorschläge allerdings moralisch fragwürdig sind.

Das Blockieren von Kommunikation ist wie in Abschnitt 5.7 untersucht ein sehr effektives Mittel gegen Browserfingerprinting, wenn es erfolgreich ist. Wird die Kommunikation zu einem Analyseserver blockiert, ist die Nutzerverfolgung nicht möglich, ansonsten kann sie ungehindert stattfinden. Diese Lösung benötig aber Blockadelisten, die betreut werden müssen. Auch können Benutzbarkeitsprobleme durch Blockieren von Kommunikation provoziert werden.

Das in Abschnitt 5.8 untersuchte **Filtern von Kommunikation** ist ein vielversprechender Ansatz. Im Gegensatz zum Blockieren von Kommunikation wird aber immer ein Fingerprint erhoben und das Filtern selbst kann Teil des Fingerprints sein.

Dadurch ist zu erwarten, dass trotz Filterns ein großer Anteil der Nutzer gut identifizierbar bleibt, der Informationsgehalt ihrer Fingerprints aber sinkt. Das Filtern von Kommunikation basiert zudem auf Filterlisten, die gewartet werden müssen. Dies kann von den Fingerprintern verkompliziert werden und es können Benutzbarkeitsprobleme durch Filtern von Kommunikation provoziert werden.

Eine sicherer Schutz vor jeder Form des Browserfingerprintings ist von dieser Technik also nicht zu erwarten.

Als letztes wurden in Abschnitt 5.10 **Kombinationen aus verschiedenen Maßnahmen** jeweils kurz betrachtet. Dabei wurde keine Kombination aus Maßnahmen gefunden, die sicher vor Browserfingerprinting schützt. In einigen Fällen können sich die Maßnahmen jedoch gegenseitig verstärken.

Das Standardisieren kann mit allen anderen Maßnahmen kombiniert werden und verbessert deren Effektivität.

Das Fälschen von Fingerprints kann mit einer Reihe von Maßnahmen sinnvoll kombiniert werden. Interessant ist dabei einerseits die Kombination mit einer Standardisierung oder dem Blockieren und Filtern von Kommunikation. Dies ermöglicht es, mehr und bessere Fingerprints zu fälschen. Andererseits kann das Fälschen auch mit einer Randomisierung kombiniert ober anderen Formen der Fingerprintauswahl kombiniert werden, um so nicht auf ein kleines Anonymity-Set beschränkt zu sein.

Die Wirkung des Deaktivieren von clientseitigen Skriptsprachen kann verstärkt werden, wenn dieses mit einer Standardisierung von beispielsweise des Useragents einhergeht oder die Benutzbarkeit dieser Maßnahme verbessert wird.

Die Entscheidung darüber, welche Maßnahmen sinnvoll miteinander zu kombinieren sind, könnte dem Nutzer auch durch ein Plugin abgenommen werden. Durch Interaktion mit dem Nutzer könnte dieses herausfinden, welche Maßnahmen akzeptabel sind und mithilfe einer zentralen Informationquelle entscheiden, welche Maßnahmen auch sinnvoll sind.

Diese Betrachtung der Kombination aus Gegenmaßnahmen konnte aufgrund der vielen möglichen Kombinationen nur knapp geschehen.

Insgesamt bietet also keine der untersuchten Maßnahmen oder untersuchten Kombinationen aus Maßnahmen einen sicheren Schutz für die Allgemeinheit der Nutzer vor jeder Form des Browserfingerprintings. Es gibt zwar Gegenmaßnahmen, diese bieten allerdings nur mit Einschränkungen Schutz.

So schützt das Randomisieren auch bei richtiger Anwendung nur Nutzer, die randomisieren, und das Filtern und Blockieren schützt nur gegen bekannte Fingerprintingskripte. Das Deaktivieren von Skriptsprachen, Standardisieren und Fälschen von Fingerprints hingegen verspricht zwar auch Nutzer zu schützen, die sich nicht selbst schützen, kann aber nicht in allen Fällen Schutz garantieren.

6

Praktische Überprüfung der Massnahmen

In diesem Kapitel werden die in Kapitel 5 untersuchten Methoden, so weit sie sinnvoll sind, praktisch überprüft. Dabei wird nicht mehr auf abstrakter Ebene gearbeitet, sondern es werden Versuche mit den gängigen Browsern und existierenden Browserfingerprintingskripten durchgeführt.

In Abschnitt 6.1 werden die offengebliebenen Fragen über das Deaktivieren von clientseitigen Skriptsprachen untersucht. In Abschnitt 6.2 wird versucht, Fingerprints zu fälschen und in Abschnitt 6.3 das Filtern und Blockieren von Fingerprintingskripten genauer untersucht. Anschließend wird in Abschnitt 6.4 das automatische Ersetzen von Analyseskripten und in Abschnitt 6.5 darauf aufbauend die Weitergabe der Ergebniss von Analyseskripten zwischen Nutzern betrachtet. Zuletzt werden die Ergebnisse in Abschnitt 6.6 zusammengefasst.

Der in diesem Kapitel erstellte Beispielcode und detailliertere Codebeispiele sind in Anhang B zu finden.

6.1. Deaktivieren von clientseitigen Skriptsprachen

Das Deaktivieren von clientseitigen Skriptsprachen hat, wie in Abschnitt 5.4 festgestellt, zwei prinzipielle Schwächen. Die erste ist die Nutzung von Skriptsprachen zum Aufbau von Webseiten und die Nachteile, die dem Nutzer durch das Deaktivieren entstehen können. Dies wird in Abschnitt 6.1.1 untersucht. Die zweite ist die Identifizierung des Nutzers anhand der weiterhin erhebbaren Merkmale des Browsers, welche in Abschnitt 6.1.2 untersucht wird. In Abschnitt 6.1.3 werden die Ergebnisse zusammengefasst.

6.1.1. Nutzung von Skriptsprachen

Um den Einsatz von Scriptsprachen abzuschätzen, wurden beliebte Domains mit einer einfachen Methode auf die Verwendung von clientseitigen Skripten gestestet.

Die Menge der untersuchten Domains wurde anhand der Liste der 1 000 000 nach des Alexarating [1] am meisten besuchten Webseiten bestimmt. Diese Liste wurde von `http://s3.amazonaws.com/alexa-static/top-1m.csv.zip` heruntergeladen und in ein nutzbares Format konvertiert. Die Startseite der so erhaltenen Domains wurde heruntergeladen und überprüft, ob sie die Zeichenkette „<script“ enthält. Nur wenn dies der Fall war, wurde gefolgert, dass die Domain Skriptsprachen nutzt. Diese Methode ist allerdings fehleranfällig, da die so gefundenen Zeichenketten nicht zwangsläufig ein für die Benutzbarkeit des Browsers relevantes Script anzeigen, und Skripte, die nicht über den Script-Tag geladen werden, auf Unterseiten geladen werden oder hinter Weiterleitungen versteckt sind, nicht erkannt werden konnten. Das genutzte Skript und die Analyseergebnisse sind in Anhang B in dem Ordner „JavascriptCrawl“ abgelegt.

Auf diese Weise wurden, wie aus Tabelle 6.1 ersichtlich, auf mehr als 90 % der erreichbaren Webseiten eingebettete Skripte gefunden. Dies zeigt, dass der Einsatz von Javaskript bei der Entwicklung von Webseiten als selbstverständlich wahrgenommen wird. Beispielsweise verwendet selbst die Webseite des Skriptblockers NoScript Javascript als Teil des Installationsbuttons für NoScript und als Teil des Spendenbuttons.

Obwohl nicht jedes eingebundene Skript für die Benutzung einer Webseite wichtig ist, deutet die starke Nutzung darauf hin, dass Webseiten ohne Javascript teilweise nicht voll funktionsfähig sind. Das komplette Deaktivieren von Skriptsprachen kann daher für den Großteil der Nutzer als inakzeptabel gelten und es wird angekommen, dass die meisten Nutzer Ausnahmeregelungen für bestimmte Skripte nutzen.

		Skripte	keine Skripte	nicht erreicht	Anteil Skripte ohne nicht erreichte Seiten
Top	100	96	3	1	97 %
Top	1000	940	46	14	95 %
Top	10000	9373	496	131	95 %
Top	100000	90603	7251	2146	93 %
Top	1000000	875295	96818	27887	90 %

Tabelle 6.1.: Die Anzahl der Seiten, auf denen Skripte gefunden wurden beziehungsweise keine Skripte gefunden wurden, und der Seiten, die nicht erreicht wurden.

Will eine Seite ihre Nutzer vor die Wahl stellen, ein eingesetztes Analyseskript zu erlauben oder die Webseite nicht zu nutzen, kann dies geschehen, indem das Laden des Webseiteninhalts mit der Analyse verwoben wird.

Soll beispielsweise die Datei `Inhalt.html` auf diese Weise versteckt werden, kann diese in `versteckt.html` umbenannt und stattdessen folgender Inhalt in der Datei `Inhalt.html` abgelegt werden:

```
<html>
 <head>
  <script>
   function replaceContent() {
      var request = new XMLHttpRequest();
      request.onreadystatechange = replace;
      analyse();
      request.open('GET','./versteckt.html');
      request.send();

      function analyse() {
          //put browserfingerprintingscript here
      }

      function replace() {
          if (request.readyState == 4 ) {
               var content = request.responseText;
               document.write(content);
               document.close();
          }
```

```
      }
    }
  </script>
 </head>
 <body onload="replaceContent();">
  you need javascript to view this page
 </body>
</html>
```

Dieses Beispielskript kann zwar keine Parameter wie Header, Post- und Get-Parameter weiterleiten, kann aber in diese Richtung erweitert werden. Würde dieses Skript eingesetzt, könnte ein Nutzer, der lediglich Javascript deaktiviert, die versteckte Seite nicht ansehen. Trifft der Nutzer allerdings weitergehende Maßnahmen, kann dies im Allgemeinen umgangen werden. Dies ist beispielsweise mit den Surrogate Scripts in NoScript möglich, die erlauben, statt des blockierten Skripts ein um den Analysecode bereinigtes Skript auszuführen.

Javascript stellt beim Deaktivieren ein Problem dar, weil es zum Generieren von Webseiten genutzt wird. Mit Maßnahmen, die trotz Deaktivierens von Skriptsprachen ein Darstellen der Webseiten erlauben, könnte dieses Problem allerdings gelöst werden.

6.1.2. Nicht-skriptbasiertes Fingerprinting

Damit die nach dem Deaktivieren von Skriptsprachen verbleibenden Merkmale zur Identifikation ausreichen, muss, wie in Abschnitt 5.4 vorgestellt, der Informationsgehalt von $\{M_i^{passiv\backslash skript}, M_j^{aktiv\backslash skript}, M_k^{noscript}, M_l^{plugin}\}$ groß genug sein. Die Entropie von $M^{aktiv\backslash skript}$ und M^{plugin} konnte aber nicht mit theoretischen Mitteln untersucht werden und wird an dieser Stelle praktisch untersucht. Dabei wurde nicht nur der im Text der Arbeit erwähnte Beispielcode erstellt, sondern auch eine Forschungsimplementierung eines solchen Fingerprinters erstellt, die in Ordner „CssHtmlFingerprinter“ des Anhangs B zu finden ist. Die Tests wurden, soweit nicht anders erwähnt, mit den Browsern

- Chromium 39.0,
- Firefox 35.0,
- Opera 12.16,
- Midori 0.4.3 und
- Internet Explorer 11.0

durchgeführt.

Um die Entropie von $M^{aktiv\backslash skript}$ einschätzen zu können, wird versucht wichtige Merkmale aus M^{aktiv} zu messen, ohne Skriptsprachen zu nutzen. Um dabei ein aktives Vorgehen zu erlauben, muss nicht nur ein unterschiedliches Verhalten zwischen den Browsern und Browserinstallationen provoziert werden, sondern dieses unterschiedliche Verhalten auch an einen Analyseserver kommuniziert werden. Diese Kommunikation wird über das Nachladen von Dateien realisiert. Die angefragten Dateien müssen dabei keine Bild- oder Nutzdateien sein, sondern können Analyseskripte sein.

Das für die Testimplementierung genutzte Analyseskript ist:

```
<?php

$con   = new mysqli("localhost","fingerprint","xxx","fingerprint");
$query = "INSERT INTO `fingerprint` (`id`,`value`) VALUES (
                '".$con->real_escape_string($_GET["id"])."',
                '".$con->real_escape_string($_GET["value"])."')";
$con->query($query);
$con->close();

?>
```

Dieses Skript legt Merkmalsausprägungen, die im Value-Parameter übergeben werden, einer Id zugeordnet in einer SQL-Datenbank ab. Die Id wird als Id-Parameter übergeben und wird für jedes Analysefenster einmal zufällig generiert.

Ein Fingerprint kann mit folgender SQL-Abfrage ermittelt werden:

```
SELECT `value` FROM  `fingerprint` WHERE `id` = '123321'
```

Das Erheben eines Merkmals funktioniert nun so, dass das Analyseskript so in der Webseite eingebunden wird, dass es geladen wird, wenn eine Merkmalsausprägung zutrifft, und die Merkmalsausprägung als Parameter übergeben wird. Eine solches Einbinden kann beispielsweise auf folgende Weise geschehen:

```
<html>
 <head>
  <link rel="stylesheet" type="text/css"
        href="addToFingerprint.php?id=123321&value=Merkmalsausprägung"/>
 </head>
 <body></body>
</html>
```

Es wurden verschiedene Möglichkeiten gefunden, Informationen über den Browser zu ermitteln und an einen Analyseserver zu kommunizieren, die im Folgenden vorgestellt werden:

Der Browsertyp kann anhand des Satzes der unterstützten HTML-Tags, der CSS-Ausdrücke und deren Attribute unterschieden werden. Beispielsweise unterstützt der Internet Explorer bis zur Version 8 den Comment-Tag, der einen nicht standardisierten HTML-Kommentar darstellt [3]. Kann das unterschiedliche Verhalten an ein Analyseskript kommuniziert werden, erlaubt dies eine grobe Analyse der HTML- oder beziehungsweise der CSS-Engine.

Die Version des Internet Explorers kann über die Conditional Comments ausgelesen werden. Die Conditional Comments werden nur vom Internet Explorer und nur in älteren Versionen unterstützt und erlauben in HTML-Kommentaren Code zu definieren, der nur von bestimmten Versionen des Internet Explorers interpretiert und von anderen Browsern ignoriert wird. In diesem Code kann nun das Analyseskript als CSS-Stylesheet eingebunden werden und die Version des Internet Explorers als Parameter übergeben werden.

Als Beispiel ein HTML-Dokument, das misst, ob es mit Internet Explorer 9 aufgerufen wurde:

```
<html>
 <head>
  <!--[if IE 9]>
   <link rel="stylesheet" type="text/css"
         href="addToFingerprint.php?id=123321&value=condComm-Ie9"/>
  <![endif]-->
 </head>
 <body></body>
</html>
```

Das korrekte Funktionieren der Conditional Comments wurde mit den Browsern Chromium, Firefox, Opera, Midori und Internet Explorer überprüft. Es ist von dieser Technik allerdings nur wenig neue Information zu erwarten, da die Browserversion gewöhnlicherweise bereits im Useragent enthalten ist.

Eine detaillierte Analyse der CSS-Engine kann über die Support-Abfrage in CSS geschehen. Diese erlaubt zu überprüfen, ob eine Zeichenkette eine gültige Zuweisung eines Wertes an eine CSS-Eigenschaft darstellt. Ist diese gültig, wird ein Block von CSS-Ausdrücken interpretiert. In diesem Block kann das Analyseskript als Hintergrundbild eines HTML-Elements eingebunden werden und die Eigenschaften des Mediums als Parameter übergeben werden.

Als Beispiel ein HTML-Dokument, das misst, ob der Browser „cyan“ als gültige Hintergrundfarbe ansieht:

```
<html>
 <head>
  <style type="text/css">
@supports (background-color:cyan) {
  #analyse1 {
    background-image:
       url("addToFingerprint.php?id=123321&value=Supports-BackgroundColor-Cyan");
  }
}
  </style>
 </head>
 <body>
  <div id="analyse1"></div>
 </body>
</html>
```

Diese Abfragen werden aber momentan nur von Browsern mit der Webkit- und der Gecko-Engine unterstützt. Dazu gehören die Browser Chromium, Firefox, Opera und Midori, aber nicht die Browser Internet Explorer und Safari. Sind diese Abfragen möglich, können umfangreiche Informationen über die Implementierung und Konfiguration der Browser erhoben werden. In einem Test wurden 200 Mozilla spezifische Erweiterungen mit Firefox Versionen 21.0 bis 35.0 gemessen und auf Unterscheidbarkeit geprüft.

- In der Version 21.0 war die Support-Anfrage noch nicht aktiviert.
- Von Version 22.0 bis 29.0 konnten Werte erhoben werden, die sich allerdings nicht änderten.
- Die Versionen 30.0 bis 33.0 hatte jeweils unterschiedliche Messergebnisse.
- Ab Version 33.0 konnten keine weiteren Änderungen festgestellt werden.

In diesem Test konnten also nur wenig Veränderungen in Erweiterungen durch die Browserversion gemessen werden. Das heißt allerdings, dass der Test auf weniger signifikante Merkmale reduziert und somit beschleunigt werden könnte. Der Browsertyp und teilweise die Browserversion lässt sich aber gut erkennen und es gibt in Firefox zusätzlich die Möglichkeit, das Interpretieren solcher Merkmale zu konfigurieren, was zum Fingerprinting genutzt werden könnte.

Informationen über die installierten Schriften können in manchen Browsern über das Nachladen von Schriften erhoben werden. Dazu kann in einer Webseite eine spezielle Schriftklasse definiert werden. Wird diese benutzt, versucht sie zunächst die zu überprüfende Schrift zu laden und lädt, wenn diese nicht verfügbar ist, eine Schriftdatei nach. Statt einer Schriftdatei wird allerdings ein Analyseskript mit einem Zugriffsidentifizierer und der Angabe, dass diese Schrift nicht installiert ist, als Parameter geladen. Wird diese neu definierte Schriftklasse genutzt, wird also entweder die installierte Schrift oder andernfalls das Analyseskript geladen. Diese Technik wurde bereits 2013 in dem Paper zu FPDetective erwähnt [12].

Als Beispiel ein HTML-Dokument, das prüft, ob die Schrift „Arial Black“ installiert ist:

```
<html>
 <head>
  <style type="text/css">
@font-face {
  font-family: "test1";
  src: local("Arial Black"),
       url("addToFingerprint.php?id=123321&value=NoFont-ArialBlack");
}
#analyse1 a {
  font-family: "test1";
}
 </head>
 <body>
  <div id="analyse1"><a></a></div>
 </body>
</html>
```

Dadurch kann zwar nicht geprüft werden, welche Schriften installiert sind, aber zu einer gegebenen Liste von Schriften kann festgestellt werden, ob diese installiert sind. Diese Technik funktioniert bei den Browsern Firefox, Opera und Midori. Beim Browser Chromium werden alle Schriften als fehlend erkannt und im Browser Internet Explorer funktioniert diese Technik erst ab Version 10. Beim Browser Firefox blockiert das oft zum Deaktivieren von Skriptsprachen genutzte NoScript auch das Nachladen von Schriften und alle Schriften werden fäschlicherweise als fehlend erkannt.

Die Entropie der Schriften wird von Eckersley mit 13,9 Bit Entropie angegeben. Da die hier vorgestellte Technik nicht von allen Browsern unterstützt wird und Schriften abgefragt werden müssen, anstatt aufgelistet zu werden, wird die hier gemessene Entropie diesen Wert unterschreiten, aber annähern.

Informationen über die Hardware können über Media-Abfragen ermittelt werden, welche es erlauben, für Ausgabemedien mit bestimmten Eigenschaften zusätzliche CSS-Ausdrücke zu laden. Dass diese Möglichkeit bestehen könnte, wurde bereits erwähnt [7], aber nicht konkretisiert.

Je nach Eigenschaften des Ausgabemediums können so für HTML-Elemente bestimmte Hintergrundbilder geladen werden. Wird statt eines Hintergrundbilds das Analyseskript mit einem Zugriffsidentifizierer und der Eigenschaft des Ausgabemediums als Parameter geladen, können Informationen über das Ausgabemedium an den Server kommuniziert werden.

Als Beispiel ein HTML-Dokument, das überprüft, ob die Bildschirmbreite auf 400 Pixel gestellt ist:

```
<html>
 <head>
  <style type="text/css">
@media (device-width:400px) {
  #analyse1 {
    background-image:
    url("addToFingerprint.php?id=123321&value=deviceWidth-400px");
  }
}
  </style>
 </head>
 <body>
  <div id="analyse1"></div>
 </body>
</html>
```

Diese Technik erlaubt, Informationen über das Betriebssystem und die Hardware des Nutzers zu sammeln. Bei den Browsern Firefox, Opera, Chromium, Midori und Internet Explorer ist es möglich, auf diese Weise die Bildschirmgröße und beim Browser Firefox teilweise sogar Betriebssystemversionen oder Farbschemata auszulesen.

Allein die Entropie von Bildschirmauflösung und Farbtiefe wird von Eckersley auf 4,83 Bit geschätzt [1]. Andere Werte, die erhoben werden können, können nicht aus allen Browsertypen ausgelesen werden und die Entropie dieser Werte wird hier nicht geschätzt.

Die unterstützten Mimetypes können über den Object-Tag ausgelesen werden. Der Object-Tag ist eigentlich zum Darstellen von abstrakten Inhalten gedacht. Um dies zur Analyse zu verwenden, wird erstens genutzt, dass ein Object-Tag Alternativinhalt darstellt, wenn er den eigentlichen Inhalt nicht laden kann. Zweitens wird genutzt, dass der Object-Tag versucht jeden Dateityp darzustellen. Um zu testen, ob der Browser bestimmte Dateitypen darstellt, wird versucht diese zu laden, und bei Versagen wird eine Rückmeldung an das Analyseskript gegeben.

Als Beispiel ein HTML-Dokument, das testet, ob Dateien vom Typ „Png“ angezeigt werden:

```
<html>
 <head></head>
```

[1] Das Merkmal der Bildschirmauflösung und Farbtiefe wird in Eckersleys Arbeit mit „screen resolution“ und davon abweichend mit „video“ bezeichnet, obwohl dies nicht explizit erwähnt wird

```
<body>
 <object type="image/png" data="fakeFile.php?mime=image/png" width="0" height="0" >
  <div style="background-image: url(addToFingerprint.php?id=123321&
      value=missingMime-image-png"></div>
 </object>
</body>
</html>
```

Um nicht von jedem Mimetype eine Datei vorrätig haben zu müssen, wurde statt einer Datei ein Skript geladen, dass sich als eine Datei mit dem als Parameter gegebenen Typ ausgibt. Der Inhalt dieser Datei ist dabei immer „ich bin ein File“.

Mit dieser Methode lassen sich in den Browsern Chromium, Firefox und Opera auslesen, welche Dateitypen interpretiert werden. Auf allen Browsern ist diese Analyse langsam. So brauchte das Überprüfen von 150 Dateitypen in einem Test mehrere Sekunden. Zudem provoziert das Einbinden von Dateitypen teilweise Fragen an den Nutzer oder Informationspopups im Browser. Der Browser Midori stürzte sogar so oft mit einer Segmentierungsverletzung ab, dass dieser nicht getestet werden konnte. Bei dem Browser Internet Explorer wurden alle MIME-Types als fehlend erkannt.

Diese Methode ist also auffällig und unzuverlässig, funktioniert aber prinzipiell in den Browsern Chromium, Firefox und Opera. Eine optimierte Implementierung könnte diese Probleme allerdings lösen, indem sie über die Zeit verteilt arbeitet, eine bessere Auswahl von Dateitypen trifft und Dateitypen besser imitiert.

Ob SSL Zertifikate vom Browser als gültig erkannt werden, lässt sich ebenfalls über den Object-Tag auslesen. Dazu wird versucht ein Objekt mit SSL abgesichert zu laden. Scheitert dies, wird davon ausgegangen, dass das SSL-Zertifikat des Servers oder der genutzte Verschlüsselungstyp nicht akzeptiert wird. Dadurch lassen sich die akzeptierten selbst signierten Zertifikate, die akzeptierten Zertifikatstellen und die unterstützten Verschlüsselungstypen auslesen.

Als Beispiel ein HTML-Dokument, das überprüft, ob das selbst signierte Zertifikat von `events.ccc.de` akzeptiert wird:

```
<html>
 <head></head>
 <body>
  <object data="https://events.ccc.de" width="0" height="0" >
   <div style="background-image: url(addToFingerprint.php?id=123321&
       value=noCert-events-ccc-de"></div>
  </object>
 </body>
</html>
```

In der Testimplementierung für diese Arbeit wurden Domains mit besonderen SSL-Verbindungen als Orakel geladen. Da diese Technik die gesamte angegebene Seite lädt und versucht sie unsichtbar darzustellen, ergeben sich mehrere Schwächen dieser Methode. Zunächst ist sie aufgrund der Netzwerkzugriffe langsam und Javascript und Umleitungen werden teilweise ausgeführt. Zudem könnte die als Orakel genutzte Seite diese Zugriffe bemerken und Maßnahmen gegen das Einbinden tätigen. Das Finden von geeigneten Orakeln hat sich beim Erstellen der Testimplementierung als Problem herausgestellt. Werden diese Orakel aber gezielt erstellt und sind Teil des Fingerprintings, entfällt der Großteil dieser Probleme. Alternativ könnten nur einzelne Elemente der Seiten wie Bilder geladen werden, um die Ladezeiten zu reduzieren und kein HTML zu interpretieren.

Im Browser Opera werden Anfragen erzeut, in denen der Nutzer gefragt wird, ob er die getesteten Zertifikate laden soll. Der Browser Midori stürzte während dieses Tests wieder mit einem Segmentierungsverletzung ab. In den Browsern Chromium und Firefox funktioniert diese Technik. Beim Browser Internet Explorer wurden alle Zertifikate als nicht akzeptiert erkannt.

Das Erkennen der akzeptierten SSL Zertifikate funktioniert prinzipiell in den Browsern Chromium und Firefox, es ist aber auffällig, langsam und unzuverlässig, kann jedoch noch verfeinert werden.

Informationen über das Netzwerk können über Iframes oder Object-Tags ermittelt werden. Ein Object-Tag kann eine Webseite innerhalb eines HTML-Dokuments darstellen. Gelingt dies nicht, wird ein Alternativinhalt angezeigt. Dies kann genutzt werden, um zu testen, ob eine Seite geladen wird. Dadurch getestet werden kann das interne Netzwerk, die Ports des Betriebssystems des Nutzers oder allgemeine Seiten.

Als Beispiel ein HTML-Dokument, das testet, ob der Computer und die IP 192.168.1.1 auf Port 80 HTTP-Antworten liefern:

```
<html>
 <head></head>
 <body>
  <object data="http://localhost:80" width="0" height="0" >
   <div style="background-image: url(addToFingerprint.php?id=123321&
       value=noHttp-localhost-80"></div>
  </object>
  <object data="http://192.168.1.1:80" width="0" height="0" >
   <div style="background-image: url(addToFingerprint.php?id=123321&
       value=noHttp-192-168-1-1-80"></div>
  </object>
 </body>
</html>
```

Wird das interne Netzwerk gescannt, können Router, Drucker und Server im Netzwerk gefunden werden, wenn diese HTTP-Services anbieten. Dies ist aber aufgrund der Instabilität dieses Merkmals und der Fähigkeit, Firewalls zu umgehen, aber eher ein Sicherheitsproblem und so für Browserfingerprinting weniger geeignet. Für das Browserfingerprinting interessanter sind die Ports des Computers, die eine HTTP-Antwort liefern. Auch interessant ist, ob Seiten geladen werden können, die beispielsweise wegen pornografischen Inhalts oder aus anderen Gründen auf Filterlisten stehen, denn so kann geprüft werden, ob der Nutzer von solche Filtern betroffen ist.

Auch diese Technik ist sehr langsam und auffällig, da sie die gesamte angegebene Seite lädt und versucht diese unsichtbar darzustellen. Dies trifft noch stärker zu als bei der Methode zur Prüfung der akzeptierten SSL-Zertifikate. Da oft kein Server hinter einer angefragten IP steht, muss auf eine Zeitüberschreitung gewartet werden. Dadurch dauerte eine solche Analyse für 255 Urls in einem Test bereits mehrere Minuten. Zusätzlich dazu wurde der Nutzer nach Nutzername und Passwort gefragt, wenn eine mit Htaccess geschützte Seite angefragt wurde, was auffällig ist.

Prinzipiell funktioniert diese Methode aber mit den Browsern Chromium und Firefox. Der Browser Midori stürzte auch bei diesem Test mit einer Segmentierungsverletzung ab. Der Browser Opera und das Plugin NoScript blockierten Anfragen in das lokale Netzwerk und auf den Computer. Beim Browser Internet Explorer wurden alle Zertifikate als nicht akzeptiert erkannt.

Diese Technik funktioniert also prinzipiell, ist in dieser Form allerdings nicht für praktischen Einsatz geeignet.

Informationen über installierte Plugins konnten nicht direkt erhoben werden. In einigen Fällen haben oder verursachen die Plugins trotzdem messbare Eigenheiten, die als Merkmal dienen können.

Das Plugin NoScript kann beispielsweise von einem Browser mit komplett deaktiviertem Javascript unterschieden werden. Dazu kann die Eigenart von NoScript genutzt werden, auch das Nachladen von Schriften zu blockieren.

Dies kann mit folgendem HTML-Dokument überprüft werden:

```
<html>
 <head>
  <style type="text/css">
@font-face {
        font-family: "fontFaceTest";
  src: url("addToFingerprint.php?id=123321&value=fontFaceLoading");
}
  </style>
```

```
 </head>
 <body>
  <a></a>
 </body>
</html>
```

Die Firefox Plugins AdBlock Plus und Ghostery blockieren das Laden bestimmter Urls, was mit einem Object-Tag gemessen werden kann. Dadurch lässt sich nicht nur testen, ob AdBlock Plus oder Ghostery aktiviert sind, sondern auch, welche Filter-Listen diese benutzen.

Als Beispiel ein HTML-Dokument, das testet, ob Urls geladen werden können, die die Zeichenkette „&pageReferrer=“ enthalten:

```
<html>
 <head>
  <link rel="stylesheet" type="text/css"
        href="addToFingerprint.php?id=123321&value=notBlocking-Referer&pageReferrer="/>
 </head>
 <body></body>
</html>
```

Es wurde kein Weg gefunden, Konstanten oder Standardwerte aus CSS auszulesen. Wie viel Entropie diese Konstanten zum Fingerprint beitragen würden, ist unbekannt.

Die Entropie von M^{plugin} wird aber an dieser Stelle nicht detailliert untersucht, da die Untersuchung von $M^{aktiv \setminus skript}$ bereits so ergiebig war.

Insgesamt ist es möglich, auch ohne Nutzung clientseitiger Skripte umfangreiche Analysen durchzuführen. So können je nach Browser Informationen gesammelt werden, die

- die CSS-Engine,
- die Bildschirmauflösung,
- die vom Browser akzeptierten SSL-Zertifikate,
- die von Browser interpretierten Dateitypen,
- das Betriebssystem und die Betriebssystemversion,
- den Anzeigestil des Betriebssystems und
- die installierten Plugins

betreffen. Die Möglichkeiten dieser Informationssammlung sind allerdings im Vergleich einer Analyse mit Javascript oder Flash sehr begrenzt und unzuverlässig. So wurde beispielsweise kein Weg gefunden, Informationen über

- Konstanten des Browsers wie Standardfarbeinstellungen und
- detaillierte Hardwareinformationen

zu ermitteln, und manche Analysemethoden funktionieren nur in bestimmten Browsern. Die Menge und geschätzte Entropie der mit den gefundenen Analysemethoden erhebbaren Werten legt nahe, dass eine Analyse des Browsers ohne clientseitige Skripte in einem ähnlichen Umfang Informationen erheben kann wie eine Analyse des Browsers mit clientseitigen Skripten. Zusätzlich dazu hat das Blockieren von clientseitigen Skripten einen Informationsgehalt von 6,64 Bit.

Der Verdacht, dass das Identifizieren der Nutzer durch Browserfingerprinting auch trotz des Deaktivierens von clientseitigen Skripten möglich ist, ist also gut untermauert, aber nicht bewiesen. Auch sind die meisten der demonstrierten Techniken bisher neu und es ist unklar, ob sie für Browserfingerprinting in der Praxis genutzt werden. Einige von ihnen sind zudem sehr auffällig, aufwendig zu implementieren, unzuverlässig,

skalieren nicht gut oder funktionieren nur bei manchen Browsern. Beispielsweise benötigen mehrere Tests einen Netzwerkzugriff für jedes Merkmal, das nicht zutrifft. Müssen deswegen beispielsweise 30 000 Anfragen pro Nutzer vom Server verarbeitet werden statt einer einzigen, wird dies den Analyseserver stark belasten. Trotz dieser Unklarheiten sollten Empfehlungen zum Deaktivieren von clientseitigen Skripten als Schutz vor Browserfingerprinting mit einer Warnung zum fraglichen Schutz versehen werden, bis Studien die Entropie und den Umfang des nicht skriptbasierten Fingerprintings geschätzt haben.

6.1.3. Zusammenfassung

Wie in Abschnitt 6.1.1 untersucht, können vom Deaktivieren von clientseitigen Skripten Einbußen in der Benutzbarkeit erwartet werden. Zusätzlich bestehen, wie in 6.1.2 untersucht, auch ohne clientseitige Skripte weiterhin weitreichende Möglichkeiten, Informationen über den Browser zu sammeln.

Diese Kombination von Nachteilen in der Benutzbarkeit und einer geringfügigen Verringerung des Informationsgehalts legt nahe, dass das Deaktivieren von clientseitigen Skripten für die breite Masse der Nutzer keine geeignete Maßnahme gegen Nutzerverfolgung durch Browserfingerprinting ist. Für konkrete Fingerprintingalgorithmen kann das Deaktivieren von clientseitigen Skriptsprachen allerdings weiterhin eine wirksame Gegenmaßnahme sein. Auch kann aufgrund der teilweise sehr unpraktischen Analysemethoden angenommen werden, dass die CSS und HTML basierten Techniken nicht zum Einsatz kommen, so lange nur wenige Nutzer clientseitige Skripte deaktivieren.

6.2. Fälschen von Fingerprints

Um das Fälschen von Fingerprints zu testen, wird in diesem Abschnitt versucht konkrete Fingerprintingmechanismen zu täuschen. Dazu wird zunächst in Abschnitt 6.2.1 versucht den bestmöglichen Fingerprint der Panopticlick Studie dem Fingerprintingskript aus dieser Studie zu präsentieren. Zusätzlich dazu wird in Abschnitt 6.2.2 eine Methode vorgestellt, auf Javascript basierende Fingerprintingmechanismen zu täuschen.

6.2.1. Uneingeschränkte Fälschung von Fingerprints

Die uneingeschränkte Fälschung wird getestet, indem versucht wird, den besten Fingerprint aus der Panopticlick Studie zu reproduzieren. Dazu müssen alle von Panopticlick erhobenen Merkmalsausprägungen reproduziert werden. Als Basis für diese Fälschung wurde ein Firefox genutzt und der Erfolg der Fälschung mit der Panopticlick Seite geprüft.

Der Useragent wurde dazu mit dem Plugin User Agent Switcher auf den Wert „Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.7) Gecko/20091221 Firefox/3.5.7“ gesetzt.

Das Speichern von Cookies wurde im Browser aktiviert.

Die Fehlen der auf Javascript basierenden Werte wurden gefälscht, indem wie beim Original Javascript deaktiviert wurde. Dies wurde mit dem Plugin NoScript durchgeführt.

Die HTTP-Header konnten leider nicht gefälscht werden, da Eckersley die Merkmalsausprägung dieses Merkmals nicht wie die anderen Merkmalsausprägungen angegeben hat. Versuche, die Merkmalsausprägung zu erraten, schlugen fehl. Das Fälschen der HTTP-Header hätte über das Plugin Modify Headers stattfinden können.

Selbst eine solch simple Fälschung konnte nicht durchgeführt werden, da nicht bekannt war, welcher Wert gefälscht werden muss. Nicht nur in Eckersleys Arbeit wurde keine vollständige Angabe eines Fingerprints mit niedrigem Informationsgehalt gefunden, auch die Suche nach einer alternativen Quelle blieb erfolglos. Erschwerend kommt hinzu, dass für eine sinnvolle Fälschung aktuelle Daten und Informationen über mehr Merkmale benötigt würden. Das Problem der unzureichenden Information müsste also systematisch gelöst werden, damit dieser Ansatz umgesetzt werden könnte.

6.2.2. Täuschung von javascriptbasiertem Fingerprinting

Aktives auf Javascript basierendes Fingerprinting lässt sich prinzipiell fälschen, indem eine Codeanalyse durchgeführt wird und der Code wie gewünscht angepasst wird. Skripte, die zunächst eine Analyse betreiben, anschließend die Analyseergebnisse in eine Nachricht verpacken und diese an einen Analyseserver schicken, sind besonders fälschungsanfällig. Für eine solche Täuschung kann der Analyseteil komplett ignoriert werden und das gewünschte Analyseergebnis unter Nutzung der Originalcodes an den Server geschickt werden. Wird immer dasselbe Skript eingesetzt, kann dies automatisiert ersetzt werden.

Als Beispiel wird eine solche Täuschung am Panopticlick Skript durchgeführt. Dieses benutzt folgenden Code als Grundstruktur:

```
// fetch client-side vars
var whorls = new Object();

try {
  whorls['plugins'] = identify_plugins();
} catch(ex) {
  whorls['plugins'] = "permission denied";
}

...

whorls['supercookies'] = test_dom_storage() + test_ie_userdata();

// send to server for logging / calculating
// and fetch results

var callback = function(results){
  success = 1;
  $('#content').html(results);
};

$.post("index.php?action=ajax_log_clientvars", whorls, callback, "html" );
```

In diesem Kernteil des Skriptes kann die Analyse einfach durch feste Werte ersetzt werden. Der Code der Panopticlick Studie würde nach diesem Prozess beispielsweise so aussehen:

```
// fetch client-side vars
var whorls = new Object();

//fake values
whorls['plugins'] = "Plugin Angaben";
...
whorls['supercookies'] = "Cookie Angaben";

// send to server for logging / calculating
// and fetch results

var callback = function(results){
  success = 1;
  $('#content').html(results);
};

$.post("index.php?action=ajax_log_clientvars", whorls, callback, "html" );
```

Die vollständigen Codebeispiele sind im Ordner „AjaxFake“ des Anhangs B zu finden.

Wird das Skript mit den festen Werten statt des Skripts mit den Analysecode ausgeführt, werden die gewünschten Daten an den Server geschickt. Dadurch ermittelte Panopticlick in einem Test mit seinem Javascriptanteil Werte wie „Plugin Angaben“ oder „Cookie Angaben“.

Als zweites Beispiel für die Täuschung eines solchen Skriptes, wurde das Fingerprintingskript von Tillmann herangezogen. Dieses hat folgenden Kerncode:

```
/* ... convert String into an Array. */
fontlist = fonts.split(',');
fp['fonts'] = fontlist;

/* ... create XML and send it to the server. */
var t = createXml(fp);
sendAjax(t, 1);

...

/** Start off by getting all System colors */
fp = getSystemColors();

/** Collect some more data... */
fp['png_id']                                    = getPngId();
...
fp['plugins']                                   = getPlugins();

/** Transform object into XML and send it to the server. */
var xml = createXml(fp);
sendAjax(xml, 0);
```

Dieser Kerncode wurde durch folgenden Code ersetzt:

```
/* ... convert String into an Array. */
fp['fonts'] = new Array("Schrift1","Schrift2");

/* ... create XML and send it to the server. */
var t = createXml(fp);
sendAjax(t, 1);

...

/** Start off by getting all System colors */
fp['color_activeborder'] = "#000000";
...

/** Collect some more data... */
fp['png_id']             = "Png-Id Angaben";
...
fp['plugins']            = "Plugin Angaben";

/** Transform object into XML and send it to the server. */
var xml = createXml(fp);
sendAjax(xml, 0);
```

An diesen Beispielen wird deutlich, dass Browsingskripte, die nach diesem Muster erstellt wurden, sich schnell und einfach täuschen lassen.

6.3. Filtern und Blockieren von Kommunikation

Um das Filtern und Blockieren zu untersuchen, wird in Abschnitt 6.3.1 demonstriert, wie Filterlisten für das Plugin AdBlock Plus erstellt werden. In Abschnitt 6.3.2 wird untersucht, wie schwer es ist, einen Filter durch Umbenennung zu umgehen.

6.3.1. Filterregeln für Browserfingerprintingskripte

Mit dem Browser-Plugin Adblock Plus kann das Nachladen von Ressourcen blockiert werden. Die Regeln, nach denen entschieden wird, ob das Nachladen einer Ressource blockiert wird, werden aus zentral betreuten Listen extrahiert, die regelmäßig aktualisiert werden. Die Liste zum Filtern von Trackingskripten von Adblock Plus muss allerdings erst vom Nutzer aktiviert werden.

Das Hinzufügen von neuen Filterregeln für Adblock Plus ist sehr einfach. Als Beispiel werden Filterregeln für Panopticlick, das Browserfingerprintingscript von Tillmann und die in dieser Arbeit genutzten Testimplementierung gegeben:

```
*bfp.henning-tillmann.de/ajax.php*
*panopticlick.eff.org/index.php?action=ajax_log_clientvars*
*addToFingerprint.php?*
```

Die Fingerprintingskripte von Tillmann und Eckersley enthalten aber auch passive Anteile, die nicht blockiert werden.

6.3.2. Automatisierte Namensveränderungen

Namensveränderungen können das Filtern und Blockieren von Kommunikation verhindern, wenn diese nicht nachvollzogen werden können. Dies ist Fall, wenn sich der Pfad eines Browserfingerprintingskriptes verändert, dies aber nicht in Filterlisten nachvollzogen wird. Dazu können verschiedene Ansätze gewählt werden.

Dateinamen, Ordnerstrukturen und Skriptparameter können serverseitig automatisch verändert werden, ohne dass die Dateien oder die Namen der Dateien tatsächlich verändert werden müssen. Dazu können üblicherweise in Webservern interne Weiterleitungen eingerichtet werden.

Als Beispiel ein Auszug aus einer Konfiguration des Webservers Apache, die die Zugriffe von der Datei „neu.php“ zur Datei „original.php“ umleitet:

```
RewriteEngine on
RewriteRule   ^(.*)/neu\.php(.*)$ $1/original.php$2 [PT]
```

Domainbasiertes Filtern und Blockieren von Kommunikation kann durch einen Alias in DNS umgangen werden. Dieser erlaubt mit einem CNAME-Eintrag, Zugriffe von einer Domain oder Subdomain auf eine andere Domain weiterleiten zu lassen. Zum Umgehen von Filtern kann eine Domain erstellt werden, die auf eine blockierte Domain verweist, und auf Webseiten kann statt der blockierten Domain die neue Domain eingebunden werden. Die Plugins Ghostery und Disconnect wurden überprüft und konnten durch einen solchen DNS-Alias umgangen werden.

Auf dieses Umbenennen kann unterschiedlich reagiert werden. Eine Möglichkeit ist das manuelle Einpflegen der neuen Namen in die Filterdatenbanken. Dass dies aber bei automatisierten Umbenennen handhabbar wäre, erscheint unwahrscheinlich. Würde das Nachtragen der Umbenennungen aber automatisiert, kann das Browserfingerprinting auf Massenbasis kaum geschehen, ohne dass dieses in den Filterlisten beschrieben wird.

Allen diesen Möglichkeiten gemein ist, dass die Umbenennung nicht nur serverseitig passieren kann. Das Einbinden der Skripte oder das Aufrufen von Funktionen muss diese Umbenennungen auch nachvollziehen, damit diese weiterhin funktionieren. Dadurch können die Namensänderungen beobachtet werden, indem eine Seite beobachtet wird, von der bekannt ist, dass sie das Namen wechselnde Browserfingerprintingskript nutzt.

6.4. Ersetzen von Fingerprintingskripten

In Abschnitt 5.10 wurde angemerkt, dass es möglich sein könnte, das Filtern und Blockieren von Fingerprintingskripten mit einer Fälschung zu kombinieren, indem die blockierten Skripte ersetzt werden. Dies wird an dieser Stelle getestet, indem die Browserfingerprintingskripte von Eckersley und Tillmann durch Ersetzen der Fingerprintingskripte getäuscht werden.

Der zunächst angedachte Ansatz, dazu Surrogate Skripts von Noscript zu nutzen, scheiterte daran, dass eine ausreichend detaillierte Kontrolle der Skripte nicht erreicht werden konnte. Stattdessen wurde eine Kombination aus den Plugins AdBlock Plus und Ghostery genutzt. Da die betrachteten Browserfingerprintingskripte Funktionen nutzen, die außerhalb des Analyseskripts definiert werden, darf Javascript für diese Seiten nicht deaktiviert werden.

Die Browserfingerprintingskripte wurden wie in Abschnitt 6.3.1 beschrieben blockiert und dadurch das Analyseskript deaktiviert. Um die Analyseskripte auch zu ersetzen, wurde das Firefox-Plugin Greasemonkey genutzt, welches erlaubt nutzerdefinierten Code nach dem Aufruf bestimmter URLs auszuführen. Auf den URLs, die die Analyseskripte einbinden, wurde als Ersatz für das urspüngliche Analyseskript ein Skript geladen, dass die gewünschten Analyseergebnisse zurückgab. Als leichte Variationen der im Abschnitt 6.2.2 genutzten Skripte, bei denen lediglich zusätzlich auf Eigenheiten von Greasemonkey eingegangen wurde, sind diese Greasemonkeyskripte ebenfalls im Ordner „AjaxFake“ des Anhangs B zu finden.

Die gefundene Lösung zeigt, dass es möglich ist, Analyseskripte automatisiert zu ersetzen. Der ursprüngliche Ansatz, Surrogate Scripts zu nutzen, scheiterte allerdings und das Angeben der URLs, die die Browserfingerprintingskripte einbinden, ist nicht gut geeignet, um Browserfingerprintingskripte von Drittparteien zu behindern, da diese von vielen verschiedenen URLs einbunden werden. Die Schwachstelle könnte aber vermieden werden, wenn das geladene Browserfingerprintingskript direkt ersetzt werden kann.

6.5. Weitergabe von Analyseergebnissen

In Abschnitt 5.10 wurde vermutet, dass beim in Abschnitt 6.4 behandelten Ersetzen von Fingerprintingskripten auch Ergebnisse der Analyseskripte direkt vom Original an den Fälscher weitergegeben werden könnten. Dies soll an den Fingerprintingskripten von Eckersley und Tillmann demonstriert werden.

Dazu wurden jeweils zwei Varianten der in Abschnitt 6.2.2 erstellten Skripte erstellt, von denen ein Exportskript den Analysecode ausführt und dem Nutzer das Analyseergebnis zur Verfügung stellt, aber nicht an den Analyseserver versendet. Dieses Analyseergebnis wird als Zeichenkette ausgegeben und kann an Fälscher weitergegeben werden. Das Importskript fragt Nutzer nach einem solchen Analyseergebnis und fälscht anhand dessen ein Teil des Fingerprints. Die Skripte sind im Ordner „AjaxFake“ des Anhangs B zu finden.

Als Beispiel der Kerncode des Exportskriptes für Eckerleys Analyseskript

```
fp['mimetypes']                                    = getMimeTypes();
fp['plugins']                                      = getPlugins();

/** Transform object into XML and send it to the server. */
alert("main");
alert(JSON.stringify(fp));
```

und der Kerncode des Importskriptes für Eckersleys Analyseskript:

```
/** fp includes all data retrievied within this file. */
var main = prompt("main");
var fp = JSON.parse(main);

/** Transform object into XML and send it to the server. */
var xml = createXml(fp);
sendAjax(xml, 0);
```

Beim ersten Versuch, die Skriptvarianten bei Tillmann zu erstellen, wurden die Daten zu spät abgegriffen und waren deshalb bereits mit zwei Ids verbunden. Dadurch konnten die Daten nicht direkt zur Fälschung genutzt werden, es wurde aber festgestellt, dass mit diesen Ids der Fingerprint des ursprünglichen Nutzers im Nachhinein vollständig verändert werden kann. Solche direkten Angriffe auf die Datenbanken der Browserfingerprinter sind ein Ansatz, Browserfingerprinting zu behindern, der in dieser Arbeit bisher nicht angedacht wurde, aber als Computersabotage auch nicht weiter verfolgt wird.

Das Weitergeben von Analysedaten zwischen Browsern, um die Fälschung von Fingerprints zu unterstützen, konnte bei Eckersleys und Tillmanns Browserfingerprintingskripten demonstriert werden. Das Weitergeben der Analysedaten selbst musste hier manuell geschehen, es spricht allerdings nichts dagegen, auch dies vollständig zu automatisieren.

6.6. Zusammenfassung

In diesem Kapitel wurden einige Maßnahmen gegen Browserfingerprinting praktisch überprüft.

Zunächst wurde in Abschnitt 6.1 das Deaktivieren von clientseitigen Skripten untersucht. Dabei wurde festgestellt, dass das alleinige Deaktivieren von clientseitigen Skripten aufgrund der Menge an Informationen, die trotz Deaktivierens von clientseitigen Skripten gewonnen werden können, nur einen schwachen Schutz gegen Browserfingerprinting darstellt. Die Benutzbarkeitsprobleme, die zu erwarten sind und auch provoziert werden können, machen diese Maßnahme zudem unpraktisch. Das Deaktivieren von clientseitigen Skriptsprachen allein ist deshalb keine Maßnahme, die Nutzerverfolgung durch Browserfingerprinting ernsthaft gefährden kann.

In Abschnitt 6.2 wurde das Fälschen von Fingerprints getestet. Das Fälschen des Fingerprints mit dem niedrigsten Informationsgehalt aus Eckersleys Studie scheiterte an einem Mangel an Informationen über diesen Fingerprint, obwohl das Täuschen von Eckersleys und Tillmanns Browserfingerprintingskripten erfolgreich war. Bessere Daten über die Verteilung der Fingerprints sind also notwendig, wenn Fingerprints mit niedrigem Informationsgehalt gefälscht werden sollen.

Die in Abschnitt 6.3 durchgeführten Experimente bestärken die Aussage, dass Filtern und Blockieren ein effektiver Weg ist, um gegen Browserfingerprinting vorzugehen, da das Anlegen neuer Filterregeln einfach ist. Das Umbenennen von Browserfingerprintingskripten ist zwar ebenfalls einfach, kann aber automatisiert nachgetragen werden und muss in den ausgelieferten Skripten ebenfalls nachvollzugen werden.

Die Kombination aus Filtern und Fälschen von Fingerprints wurde in Abschnitt 6.4 untersucht, indem demonstriert wurde, dass bestimmte Analyseskripte durch Nutzung vorhandener Plugins durch Varianten dieser Skripte ersetzt werden können, die beliebige Werte an den Analyseserver senden.

Als letztes wurde in Abschnitt 6.5 gezeigt, dass es möglich ist, die bestimmte Analyseskripte so zu verändern, dass diese die ermittelten Ergebnisse an den Nutzer ausgeben, statt sie an den Analyseserver zu schicken. Ebenso konnten Varianten erstellt werden, die ermöglichen Nutzereingaben anzunehmen statt Ergebnisse zu ermitteln und diese Nutzereingaben als Ergebnisse an den Analyseserver zu schicken.

Insgesamt bestätigte die praktische Überprüfung die im Kapitel 5 getätigen Aussagen. Lediglich das Deaktivieren von clientseitigen Skripten musste neu bewertet werden, da es unerwartet schlechten Schutz vor Browserfingerprinting bietet. Soll dieses auch vor den in diesem Kapitel demostrierten Analysemethoden schützen, müsste es mit weiteren Maßnahmen kombiniert werden.

Dazu könnten die CSS-Anweisungen oder HTML-Tags, die Informationen über den Browser preisgeben, zusätzlich deaktiviert werden. Kandidaten hierfür wären die Media- und Support-Anfragen in CSS, die Fähigkeit, über CSS Schriften nachzuladen, und der Object- und Iframe-Tag in HTML. Dadurch würde der Großteil der vorgestellten Analysemethoden fehlschlagen und dieser Ansatz erscheint Erfolg versprechend. Beispielsweise konnte aus dem sehr primitiven Browser Links, der diese CSS-Anweisungen und HTML-Tags nicht unterstützt, keine Informationen über den Browser extrahiert werden.

Auch das Nachladen von Ressourcen kann manipuliert werden, um die Analyse des Browsers zu ver- oder behindern. Könnten keine Ressourcen nachgeladen werden, könnte keine Kommunikation stattfinden, es würden aber auch starke Einbußen in der Benutzbarkeit des Browsers folgen. Würden alle möglichen Ressourcen unabhängig von Ihrer Nutzung nachgeladen, kann keine sinnvolle Analyse stattfinden und die Webseite würde bis auf ein höheres zu übertragendes Datenvolumen normal nutzbar sein. Es gäbe auch Zwischenlösungen wie das Blockieren des Nachladens von bestimmten Ressourcen mit AdBlock Plus. Diese Maßnahmen können allerdings wiederum durch andere Formen der Kommunikation wie das Ein- und Ausblenden von Links umgangen werden.

Das Einschränken von CSS und HTML sowie das Kontrollieren des Nachladens von Ressourcen könnte vermutlich das aktive nicht skriptbasierte Browserfingerprinting so stark behindern, dass dieses nicht mehr praktikabel ist. Allerdings könnte es in den Fingerprint einbezogen werden, weswegen solche Maßnahmen in vorhandene Browser oder Browserplugins wie NoScript integriert werden sollten, um das Privacy-Paradox abzumildern.

7 SIMULATION DER MASSNAHMEN

Um die Auswirkungen von Gegenmaßnahmen gegen Browserfingerprints untersuchen zu können, ohne die Gegenmaßnahmen direkt zu implementieren, soll eine nicht-deterministische diskrete Simulation genutzt werden.

In Abschnitt 7.1 wird der grundsätzliche Aufbau der Simulation beschrieben. Anschließend werden in Abschnitt 7.2 Durchläufe der Simulation zunächst ohne Gegenmaßnahmen durchgeführt und die genutzten Parameter angepasst.

Um Abweichungen von Eckerleys Studie einschätzen zu können, wurde zunächst in Abschnitt 7.3 die Anzahl der Nutzer variiert und in Abschnitt 7.4 die Anzahl der genutzten Attribute verändert. Darauf folgt in Abschnitt 7.5 die Simulation des Effekts des Schutzparadoxes und von Abschnitt 7.6 bis 7.8 werden die in den vorherigen Kapiteln untersuchten Maßnahmen simuliert, so weit dies sinnvoll ist. Abschließend werden in Abschnitt 7.9 die Ergebnisse zusammengefasst.

Der Sourcecode und die genauen Ergebnisse der Simulationen sind im Ordner „Simulation“ des Anhangs B abgelegt. Tabellen mit detailierterenen Ergebnissen und deren Variationskoeffizenten sind in Anhang A zu finden.

7.1. Aufbau der Simulation

Die Simulation soll auf Aktivitäten des Nutzers basieren und von diesen ausgehend die Trackingdatenbanken der Analyseserver aufbauen. Dazu werden zunächst in der Basissimulation nur naive Nutzer und naive Fingerprintingalgorithmen genutzt. Für spätere Nutzung werden spezialisierte Nutzer und Fingerprintingalgorithmen erstellt, wobei die Grundfunktion der Simulation beibehalten und Änderungen explizit beschrieben werden.

Zeitscheiben stellen das Fortschreiten der Zeit in der Simulation dar. Zu Beginn jeder Zeitscheibe wird jedes User-Objekt mittels eines „Ticks“ benachrichtigt und vollzieht eine Aktion wie das Besuchen eines

Fingerprinters. Die User-Objekte werden dabei nach der Reihenfolge ihres Indexes abgearbeitet und nach dem Beenden ihrer Aktionen wird der nächste Nutzer über den „Tick“ informiert. Um Verzerrungen aufgrund dieser Ausführungsreihenfolge zu vermeiden, werden pro „Tick“ nur wenige Aktionen von einem User-Objekt ausgeführt.

Nach Beendigung jeder Zeitscheibe werden die Statistiken für diese Zeitscheibe angefertigt und der nächste „Tick“ angestoßen. Ist eine feste Menge an Zeitscheiben durchlaufen, wird die Simulation beendet und mit der Nachverarbeitung der Ergebnisse begonnen.

Soll kein zeitliches Verhalten simuliert werden, kann lediglich ein einziger „Tick“ ausgeführt werden.

Ein Nutzer beziehungsweise dessen Browser werden als User-Objekt repräsentiert. Dass ein Nutzer mehrere Browserinstallationen nutzt oder dass mehrere Nutzer dieselbe Browserinstallation nutzen, wird in dieser Simulation vernachlässigt.

Die Merkmale werden in dieser Simulation abstrakt dargestellt. Das User-Objekt speichert dementsprechend die Merkmalsausprägungen als Zahl. Im einfachsten Fall ist dies lediglich ein Merkmal, das direkt dem Fingerprint entspricht.

Bei jedem "´Tick“ wird das User-Objekt benachrichtigt und es hat die Möglichkeit, eine Aktion auszuführen. In der Basissimulation ist dies der Besuch eines Fingerprintingskriptes, das User-Objekt kann aber um beliebige Fähigkeiten erweitert werden.

Das User-Objekt speichert zusätzlich eine eindeutige Nutzer-ID, die bei jedem Besuch eines Fingerprintingskriptes mitgesendet wird, aber nicht direkt in einem Fingerprint verwertet werden darf.

Vor Beginn der Simulation müssen die User-Objekte initialisiert werden. Dazu werden die Merkmale des Nutzers nach einem als Simulationsparameter gegebenen Schema zufällig generiert.

Ein Fingerprinter fragt beim Besuch eines User-Objektes eine bestimmte Menge von dessen Merkmalen und seine Nutzer-ID ab. Die Anfrage der Daten entspricht dem Erheben eines Fingerprints und die zurückgegebenen Daten entsprechen dem Fingerprint des Nutzers. Die erhobenen Fingerprints werden für die Dauer eines „Ticks“ gespeichert, nach Ablauf des „Ticks“ analysiert und gelöscht.

Die Statistiken über die gemessenen Fingerprints werden nach jedem „Tick“ berechnet und beinhalten

- die Entropie der Fingerprints,
- die Größe der zehn größten Anonymity-Sets und
- der Anteil der Nutzer, die ein Anonymity-Set der Größe
  - eins,
  - zwei,
  - zwei bis neun,
  - zehn oder mehr

  haben.

Das Erstellen der Statistiken geschieht aus Sicht der Fingerprinter und für jeden Fingerprinter individuell. Um belastbarere Ergebnisse zu erhalten, werden mehrere Simulationsdurchläufe durchgeführt und als Qualitätsindikator der Variationskoeffizient angegeben.

Eine Auswertung dieser Statistiken muss in Hinblick auf das jeweilige Simulationsziel geschehen.

Um die Ergebnisse möglichst reproduzierbar zu machen, wird der in der Simulation genutzte Zufallsgeneratoren mit einem festen Wert geseedet. Bei Versuchen, die eigenen Experimente zu reproduzieren, wurde allerdings festgestellt, dass in wenigen Fällen trotz Seedens der Zufallsgeneratoren aus unbekannten Gründen unterschiedliche Ergebnisse in eigentlich identischen Simluationsdurchläufen ermittellt wurden. Der feste Seed wurde trotzdem beibehalten und angegeben.

7.2. Durchlauf ohne Gegenmaßnahmen

Die technische Implementierung einer Simulation kann fehlerhaft oder die Ergebnisse können stark verzerrt sein. Um diese Fehler abzuschätzen und einzuschätzen, wie gut die Simulation die Realität nachbildet, werden zunächst Simulationen ohne Gegenmaßnahmen durchgeführt. Damit die Ergebnisse mit der Realität in Verbindung gebracht und überprüft werden können, wird versucht die Studie von Eckersley nachzubilden. Die hierbei ermittelten Parameter werden als Standardparameter für die weiteren Simulationen genutzt. Da sich die Fingerprints in dieser Simulation nicht sebstständig ändern, wird das zeitliche Verhalten dabei ignoriert und lediglich ein „Tick“ simuliert.

In Eckersleys Studie wurden bei 470 161 freiwilligen Teilnehmern Fingerprints

- mit einer Entropie von 18,1 Bit,
- einem größten Anonymity-Set von 1 186 Nutzern und
- 83,6% einzigartigen Nutzern

gemessen. Detaillierte Informationen können Tabelle 7.1 entnommen werden. Ein Simulationsversuch soll als erfolgreich gelten, wenn alle gemessenen Werte in der selben Größenordnung wie die Werte aus Eckersleys Studie liegen, was erreicht wird, wenn sie mindestens 10% und höchstens 1000% der von Eckersley gemessenen Werten betragen.

	1. Versuch	2. Versuch	3.Versuch	4.Versuch	Eckersleys Studie [22]
Nutzer	470 161	470 161	470 161	470 161	470 161
Entropie	17,60 Bit	17,43 Bit	17,91 Bit	18,34 Bit	18,1 Bit
Anon1	10,38	14,62	1 185,11	1 164,58	1 186
Anon2	9,88	13,83	1 148,07	396,89	1 100
Anon3	9,59	13,44	1 118,36	385,58	1 017
Anon4	9,36	13,20	1 088,58	378,96	940
Anon5	9,18	13,03	1 061,50	373,72	886
Anon6	9,08	12,90	1 034,84	368,51	788
Anon7	9,03	12,81	1 008,65	364,24	755
Anon8	9,00	12,68	983,39	359,33	746
Anon9	8,99	12,52	958,75	353,97	702
Anon10	8,98	12,37	936,76	347,33	618
Nutzer1	18,8%	22,1%	84,5%	84,2%	83,6%
Nutzer2	31,4%	21,0%	5,3%	5,1%	5,3%
Nutzer2-9	81,2%	77,3%	5,5%	9,7%	8,2%
Nutzer10+	0,0%	0,6%	9,9%	6,2%	8,1%

Tabelle 7.1.: Die Simulationsergebnisse in Vergleich zu Eckersleys Studie. Anon$x$ bezeichnet dabei, das x. größte Anonymity-Set und Nutzer$x$ die Nutzer mit einem Anonymity-Set der Größe x.

Der erste Versuch zum Erstellen der Simulationsparameter war das Nachahmen der von Eckersley gemessenen Entropie und der Menge der an der Studie teilnehmenden Nutzer. In der Simulation wurde lediglich ein Merkmal zur Darstellung des Fingerprints verwendet und dieses mit einer Gleichverteilung ausgewürfelt. Die Entropie der Gleichverteilung ist $log_2(n)$, wobei n die Anzahl der Elemente ist, aus denen die Auswahl stattfindet. Dadurch hat eine Auswahl aus $n = 2^{18,1} \approx 280\,959$ Elementen die Entropie der Fingerprints aus Eckerleys Studie. Die Anzahl der Teilnehmer aus Eckersleys Studie wurde für die Simulation übernommen.

In 1 000 Simulationsdurchläufen wurde mit diesen Parametern durchschnittlich

- eine Entropie von 17,60 Bit,
- ein größtes Anonymity-Set mit durchschnittlich 10,38 Nutzern und

- 18,8% Nutzer mit einzigartigem Fingerprint

gemessen. Der Variationskoeffizient der Entropie und der Anteile der Nutzer mit Anonymity-Sets waren dabei bis auf den Anteil der Nutzer mit einem Anonymity-Set von 10 oder mehr kleiner als 0,35% und der Variationskoeffizient der Größen der größten Anonymit-Sets war kleiner als 7%. Der Variationskoeffizient des sehr kleinen Anteiles der Nutzer mit einem Anonymity-Set von 10 oder mehr war etwa 56%.

Die Entropie der in der Simulation gemessenen Fingerprints ist mit 17,60 Bit geringer als die Entropie der genutzen Zufallsverteilung von 18,1 Bit. Dies kann dadurch erklärt werden, dass in der Stichprobe zu jedem Fingerprint ein Informationsgehalt von maximal $-log_2(\frac{1}{470\,161}) \approx 18,84$ Bit gemessen werden kann, auch wenn dieser Wert theoretisch höher wäre. Dies weist darauf hin, dass Studien besonders mit einer kleinen Menge von Teilnehmern die Entropie systematisch unterschätzen.

Die Abweichungen bei der Menge der Nutzern mit einzigartigem Fingerprint und bei der Größe des größten Anonymity-Sets zeigen aber, dass die genutzten Simulationsparameter nicht das gewünschte Ergebnis erzeugen und somit die Realität nur schlecht modellieren. Daraus kann geschlossen werden, dass nicht nur die Entropie der genutzten Verteilung das Ergebniss beinflusst, sondern auch ihre Zufallsdichte. Die Auswirkungen der Gleichverteilung auf die Anonymität der Nutzer sind interessant, da die Anzahl der einzigartigen Nutzern gering ist, aber fast kein Nutzer hat ein Anonymity-Set von zehn oder mehr.

Um eine bessere Modellierung von Eckersleys Experiment zu erreichen, wurde im zweiten Simulationsversuch nicht nur die Entropie der von Eckersley gemessenen Verteilung nachgebildet, sondern auch ihre Form. Dazu wurde eine geometrische Verteilung statt einer Gleichverteilung gewählt, da diese wie die von Eckersley gemessene Verteilung eine hohe Wahrscheinlichkeit für wenige Werte und für die restlichen Werte eine geringe Wahrscheinlichkeit hat. Die Entropie der geometrischen Verteilung ist durch die Formel $H = \frac{-(1-p)\cdot log_2(1-p) - p\cdot log_2(p)}{p}$ gegeben, wobei p ein Parameter der geometrischen Verteilung ist. Durch Annäherung wurde der Wert $p = 0,0000095$ bestimmt, der eine Entropie von etwa 18,13 Bit produziert.

In 1 000 Simulationsdurchläufen wurde mit dieser Zufallsverteilung durchschnittlich

- eine Entropie von 17,43 Bit,
- ein größtes Anonymity-Set mit durchschnittlich 14,56 Nutzern und
- 22,2% Nutzer mit einzigartigem Fingerprint

gemessen.

Der Variationskoeffizient der Entropie und der Anteile der Nutzer mit Anonymity-Sets waren dabei bis auf den Anteil der Nutzer mit einem Anonymity-Set von 10 oder mehr kleiner als 0,4% und der Variationskoeffizient der Größen der größten Anonymit-Sets war kleiner als 7%. Der Variationskoeffizient des weiterhin kleinen Anteiles der Nutzer mit einem Anonymity-Set von 10 oder mehr war etwa 6,4%.

Auch die Ergebnisse dieser Simulation weichen noch zu stark von den Ergebnissen aus Eckersleys Studie ab, als dass die Simulation eine erfolgreiche Nachbildung von Eckersleys Studie sein könnte.

Aufrund dieser Probleme wurde im dritten Versuch ein anderer Ansatz zum Erstellen der Zufallsverteilung gewählt. Es wurden zwei Zufallsverteilungen genutzt, von denen die eine große Anonymity-Sets und die andere eine große Menge von einzigartigen Fingerprints hervorbringt. Diese Zufallsverteilungen wurden kombiniert, indem beim Generieren der Merkmale mit 10% Wahrscheinlichkeit die Zufallsverteilung für große Anonymity-Sets und mit 90% Wahrscheinlichkeit die Zufallsverteilung für viele einzigartige Fingerprints genutzt wurde. Als Zufallsverteilungen wurden geometrische Verteilungen gewählt. Die Zufallsverteilung für große Anonymity-Sets hat den Parameter $p = 0,025$ und die Zufallsverteilung für die große Menge von einzigartigen Fingerprints hat den Parameter $p = 0,0000003$. Die Parameter wurden ermittelt, indem sie angenähert wurden, bis ein Anonymity-Set von etwa 1 200 Nutzern und etwa 84,9% einzigartige Nutzer gemessen wurden.

In 1 000 Simulationsdurchläufen wurde mit dieser Kombination aus Zufallsverteilungen durchschnittlich

- eine Entropie von 17.91 Bit,
- ein größtes Anonymity-Set mit durchschnittlich 1 182,75 Nutzern und
- 84,5% Nutzer mit einzigartigem Fingerprint

gemessen.

Der Variationskoeffizient der Entropie und der Anteile der Nutzer mit Anonymity-Sets waren dabei kleiner als 1% und der Variationskoeffizient der Mächtigkeit der größten Anonymit-Sets war kleiner als 2,4%.

Wie aus Tabelle 7.1 ersichtlich ist, liegen alle gemessenen Werte nun in der selben Größenordnung wie die von Eckersley gemessenen Werte und weichen sogar um maximal 52% von diesen ab, obwohl nur das größte Anonymity-Set und die Anzahl der einzigartigen Nutzer zur Auswahl der Parameter genutzt wurde. Damit spiegelt eine Simulation mit diesen Parametern Eckersleys Studie also gut wieder.

Da für eine Simulation eine größere Anzahl von Merkmalen wünschenswert ist, wird im vierten Simulationsversuch die Verteilung für große Anonymity-Sets und die Verteilung für einzigartige Fingerprints in mehrere Merkmale mit einzelnen Zufallsverteilungen differenziert. Dazu wurde der Fingerprint durch 10 Merkmale dargestellt, die alle entweder mit der Verteilung für große Anonymity-Sets oder mit der Verteilung für einzigartige Fingerprints initialisiert werden. Um diese Änderung auszugleichen, wurde für die Verteilungen für große Anonymity-Sets der Parameter $p = 0,69$ und für die Verteilungen für große Anonymity-Sets der Parameter $p = 0,36$ genutzt.

In 1 000 Simulationsdurchläufen wurde mit diesen Parametern durchschnittlich

- eine Entropie von 18,49 Bit,
- ein größtes Anonymity-Set mit durchschnittlich 1 149,80 Nutzern und
- 92,04% Nutzer mit einzigartigem Fingerprint

gemessen.

Der Variationskoeffizient der Entropie und der Anteile der Nutzer mit Anonymity-Sets waren dabei kleiner als 0,9% und der Variationskoeffizient der Mächtigkeit der größten Anonymit-Sets war kleiner als 3,1%.

Wie aus diesen Ergebnissen und Tabelle 7.1 ersichtlich ist, stellt die Simulation mit diesen Parametern Eckerslseys Studie gut dar.

Die plausiblen Ergebnisse der durchgeführten Simulationen bestätigen ihre technisch korrekte Implementierung. Dass eine solche Kombination aber erst nach mehreren Versuchen und einer nicht exakt begründbaren Parameterwahl möglich war, zeigt, dass auch die innere Struktur und nicht nur die Entropie der genutzten Zufallsverteilungen relevant ist. Soll die Simulation die Realität wiederspiegeln, darf nicht von den hier entwickelten Parametern abgewichen werden. Die ermittelten Parameter werden soweit möglich in den weiteren Simulationen weiterverwendet.

7.3. Variation in den Nutzerzahlen

Die Studie von Eckersley hatte 470 161 Teilnehmer, weswegen diese Zahl von Nutzern auch für die meisten Simulationen in dieser Arbeit übernommen wurde. Dass allerdings andere Fingerprintingskripte ebenfalls diese Anzahl von Browsern messen, ist nicht anzunehmen. Setzt beispielsweise eine kleine Seite Browserfingerprinting ein oder können Nutzer durch andere Faktoren ausgeschlossen werden, sind einige Tausend Nutzer durchaus zu erwarten. Für globales Fingerprinting und große Seiten sind Millionen bis Milliarden Browser eine realistischere Annahme. Um die Anonymität der Nutzer in diesen Situationen besser einschätzen zu können, werden Variationen in den Nutzerzahlen simuliert.

Nutzer	Entropie	Nutzer1	Nutzer2	Nutzer2-9	Nutzer10+	anon1	anon2	anon10
47	5,55	100,0%	0,0%	0,0%	0,0%	1,0	1,0	1,0
470	8,87	99,13%	0,81%	0,87%	0,0%	1,9	1,7	1,0
4701	12,15	96,88%	1,46%	2,89%	0,23%	11,6	7,8	3,2
47016	15,33	92,37%	2,57%	5,46%	2,17%	113,5	46,3	30,4
470161	18,34	84,17%	5,1%	9,68%	6,16%	1169,9	394,2	348,5
4701610	20,99	67,06%	9,75%	20,54%	12,4%	11657,9	3769,4	3597,0

Tabelle 7.2.: Die durchschnittlichen Ergebnisse der Simulation mit verschieden großen Nutzerzahlen bei 10 Simulationsdurchläufen. anon$x$ entspricht dabei dem x. größten Anonymity-Set und Nutzer$x$ gibt die Anzahl der Nutzer mit einem Anonymity-Set der Größe x an.

Als Erstes wurden verschiedene Größenordnungen von Nutzern simuliert. Um den Bezug zu Eckersley Studie zu wahren, wurden dabei 470 161 Nutzer als Basiswert genutzt. Die Ergebnisse sind in Tabelle 7.2 einsehbar. Für 47 016 100 und mehr Nutzer wurden keine Simulationen durchgeführt, da dies zu viel Ressourcen benötigt hätte.

Für die genutzte Verteilung ist zu erkennen, dass die Anzahl der betrachteten Nutzer einen starken Einfluss auf die Anonymität der Nutzer hat. Bei wenigen Tausend Nutzern sind über 95% der Nutzer identifizierbar und die restlichen Nutzer haben nur ein sehr kleines Anonymity-Set. Bei 4 701 610 Nutzern sind hingegen nur noch etwa 67% der Nutzer eindeutig identifizierbar und es treten sehr große Anonymity-Sets auf. Es haben aber immer noch etwa 92% der Nutzer ein Anonymity-Set, das weniger als 10 Nutzer umfasst.

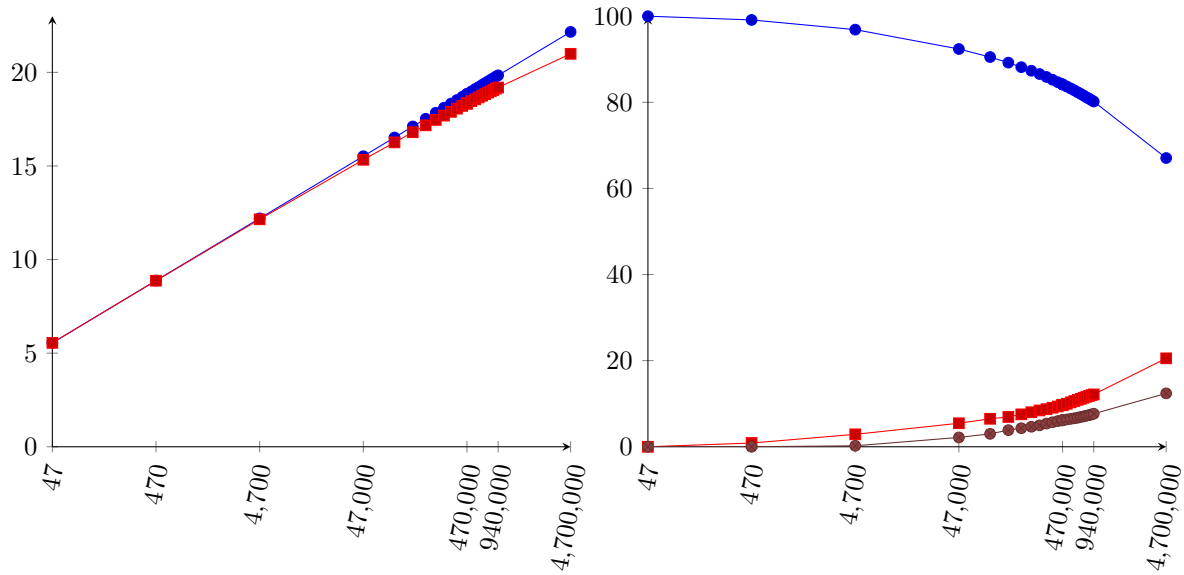
Dies heißt, dass eine globale Identifizierbarkeit mit der in der Simulation genutzten Verteilung nicht gegeben ist, die Nutzer teilen sich aber nur mit wenigen anderen Nutzern einen Fingerprint. Kleine Stichproben hingegen machen die meisten Nutzer eindeutig identifizierbar.

Nutzer	Entropie	Nutzer1	Nutzer2	Nutzer2-9	Nutzer10+	anon1	anon2	anon5	anon10
10%	15,33	92,38%	2,55%	5,39%	2,24%	116,5	45,2	39,3	30,9
20%	16,26	90,51%	3,07%	6,5%	3,0%	236,2	88,0	74,5	65,2
30%	16,8	89,21%	3,43%	6,93%	3,86%	349,6	126,5	115,5	100,4
40%	17,17	88,16%	3,77%	7,53%	4,31%	473,3	168,5	152,1	136,0
50%	17,46	87,33%	4,02%	8,02%	4,65%	576,5	203,8	189,6	170,5
60%	17,69	86,57%	4,29%	8,44%	4,98%	715,7	242,0	224,6	200,4
70%	17,89	85,89%	4,49%	8,73%	5,38%	827,6	282,3	258,6	241,2
80%	18,06	85,25%	4,72%	9,06%	5,69%	934,5	328,3	303,0	276,5
90%	18,21	84,67%	4,91%	9,36%	5,97%	1061,0	360,8	337,0	316,3
100%	18,34	84,14%	5,11%	9,7%	6,16%	1182,7	395,1	371,7	348,2
110%	18,46	83,67%	5,28%	9,99%	6,34%	1301,5	436,2	413,2	386,0
120%	18,56	83,21%	5,43%	10,32%	6,47%	1408,6	469,3	451,0	420,9
130%	18,66	82,78%	5,58%	10,61%	6,61%	1490,5	511,5	485,0	452,7
140%	18,75	82,34%	5,73%	10,9%	6,75%	1612,9	546,9	527,4	492,3
150%	18,84	81,94%	5,86%	11,13%	6,93%	1744,1	587,3	557,4	528,4
160%	18,92	81,56%	5,99%	11,39%	7,05%	1880,6	627,8	594,4	554,1
170%	18,99	81,17%	6,13%	11,62%	7,21%	1971,1	660,9	630,0	590,1
180%	19,06	80,85%	6,24%	11,8%	7,35%	2086,4	698,1	663,6	636,6
190%	19,13	80,51%	6,33%	11,99%	7,5%	2240,9	735,9	703,5	670,1
200%	19,19	80,18%	6,45%	12,16%	7,67%	2358,9	780,4	745,7	709,3

Tabelle 7.3.: Die durchschnittlichen Ergebnisse der Simulation mit verschieden großen Nutzerzahlen bei 10 Simulationsdurchläufen. Die Zahl der Nutzer ist in Prozent auf 470 161 bezogen angegeben. anon$x$ entspricht dabei dem x. größten Anonymity-Set und Nutzer$x$ gibt die Anzahl der Nutzer mit einem Anonymity-Set der Größe x an.

Als Nächstes wurden kleinere Abweichungen der Nutzerzahlen simuliert. Dazu wurde der Bereich von 10%

bis 200% Nutzern bezüglich den 470 161 Nutzern aus Eckerleys Studie in 10% Schritten abgedeckt. Die Ergebnisse sind in Tabelle 7.3 einsehbar.



(a) Die Entwicklung der Entropie (rot) und der maximal möglichen Entropie (blau) bei steigenden Stichprobengrößen.

(b) Die Entwicklung des Anteiles der einzigartigen Nutzer (blau), der Nutzer mit einem Anonymity-Set der Größe 2-9 (rot) und der Nutzer mit einem Anonymity-Set von 10 oder mehr (braun) in Prozent bei steigenden Stichprobengrößen.

Abbildung 7.1.: Die Entwicklung der Simulationsergebnisse bei variierenden Nutzerzahlen. Die Achse der Nutzerzahlen wurde logarithmisch gewählt.

Zumindest bei der in der Simulation genutzten Zufallsverteilung verändert sich die Zahl der einzigartigen Nutzer und die gemessene Entropie schon bei kleinen Abweichungen von den originalen Nutzerzahlen stark. Dadurch wird ersichtlich, dass alle Angaben mit ihrer Stichprobengröße gewichtet werden müssen oder nur als grobe Schätzung betrachtet werden können. Eine Entwicklung der Werte über verschiedenen Stichprobengrößen, wie in Abbildung 7.1 gezeigt, könnte eine exaktere Abschätzung erlauben.

Für die genutzte Verteilung kann beispielsweise erkannt werden, dass die Werte sich noch nicht stabilisiert haben. Bei der Entropie ist deutlich zu erkennen, dass diese zunächst die maximal möglichen Werte annimmt und ihre Kurve dann anfängt abzuflachen, da sie gegen ihren idealen Wert konvergieren muss. Es ist allerdings ebenfalls deutlich zu erkennen, dass der Konvergenzprozess noch nicht abgeschlossen ist und die Entropie weiter steigen wird. Bei den Kurven für die prozentualen Nutzerzahlen ist klar, dass diese gegen 0% bzw. 100% konvergieren. Anhand der Kurve wird deutlich, dass sich diese Zahlen im Bereich von mehreren Millionen Nutzern stark ändern.

Nutzer	Entropie	Nutzer1	Nutzer2	Nutzer2-9	Nutzer10+	anon1	anon2	anon10
470161-47016	18,21	84,69%	4,94%	9,36%	5,95%	1042,5	358,0	311,5
470161-4701	18,33	84,2%	5,09%	9,65%	6,14%	1161,1	392,0	344,2
470161-470	18,34	84,15%	5,1%	9,66%	6,19%	1178,9	394,9	347,1
470161-47	18,34	84,17%	5,1%	9,69%	6,14%	1172,8	393,0	347,6
470161	18,34	84,17%	5,1%	9,68%	6,16%	1169,9	394,2	348,5

Tabelle 7.4.: Die durchschnittlichen Ergebnisse der Simulation mit verschieden großen Nutzerzahlen bei 10 Simulationsdurchläufen. anon$x$ entspricht dabei dem x. größten Anonymity-Set und Nutzer$x$ gibt die Anzahl der Nutzer mit einem Anonymity-Set der Größe x an.

Als Letztes wurde der Effekt simuliert, der eintritt, wenn eine Gruppe von Nutzern sich auf irgendeine Weise dem Browserfingerprinting entzieht. Dazu wurden die 470 161 Nutzer als Basiswert für die Anzahl der simulierten Nutzer genommen und Bruchteile von 0,01% bis 10% davon abgezogen. Die Ergebnisse sind in Tabelle 7.4 einsehbar.

Wenn sich nur bis zu 1% Nutzer dem Browserfingerprinting entziehen, hat dies keinen klar erkennbaren Einfluss auf die Anonymität der restlichen Nutzer. Entziehen sich 10% der Nutzer dem Fingerprinting, konnte eine leichte Verschlechterung der Anonymität der restlichen Nutzer festgestellt werden. Solange sich also nicht massenhaft Nutzer dem Browserfingerprinting entziehen, kann dies ohne Rücksicht auf die restlichen Nutzer erfolgen.

Insgesamt zeigte sich, dass zumindest für die konkrete Verteilung die Nutzeranzahl eine große Rolle spielt, wobei bei wenigen Tausend Nutzern fast alle Nutzer eindeutig identifizierbar wären und bei mehreren Millionen Nutzern ein großer Teil der Nutzer nicht mehr eindeutig sind. Entziehen sich kleine Mengen von Nutzern dem Browserfingerprinting, hatte dies keinen eindeutigen Effekt.

Ob diese Ergebnisse allerdings auf Eckersleys Studie und die Realität übertragbar sind, ist fraglich, kann aber nicht mit den von Eckersley veröffentlichten Daten überprüft werden. Dies betont die Wichtigkeit, bei solchen Studien Daten über die Entwicklung der Werte oder den gesamten Datensatz zu veröffentlichen.

7.4. Variation in der Anzahl der Merkmale

Merkmale	Entropie	Nutzer1	Nutzer2-9	Nutzer10+	anon1	anon2	anon10
1	2,53	0,0%	0,0%	100,0%	185049,0	107641,8	2737,1
2	5,06	0,01%	0,09%	99,9%	77163,0	42215,0	14953,8
3	7,57	0,15%	0,79%	99,06%	35145,8	17513,6	9462,9
4	10,06	1,03%	4,16%	94,81%	17724,7	7954,9	3930,9
5	12,45	4,63%	13,26%	82,11%	9938,8	3976,2	1776,8
6	14,57	14,31%	26,96%	58,74%	6032,8	2197,7	903,9
7	16,25	32,03%	34,97%	33,0%	3820,0	1350,2	508,2
8	17,37	54,08%	29,56%	16,36%	2517,8	861,3	315,5
9	18,01	72,77%	17,97%	9,26%	1719,9	581,0	509,8
10	18,34	84,18%	9,68%	6,14%	1155,0	394,1	348,4
11	18,51	89,86%	5,97%	4,17%	815,9	270,8	236,4
12	18,62	92,76%	4,35%	2,89%	548,1	193,9	165,0
13	18,68	94,55%	3,42%	2,02%	382,0	135,4	114,7
14	18,73	95,83%	2,96%	1,21%	265,1	95,6	77,4
15	18,77	96,83%	2,44%	0,72%	182,8	68,1	55,1
16	18,79	97,67%	1,88%	0,45%	126,9	47,1	38,3
17	18,81	98,33%	1,44%	0,23%	84,4	36,2	25,9
18	18,82	98,83%	1,07%	0,11%	56,1	25,2	18,5
19	18,83	99,22%	0,73%	0,06%	41,2	19,2	12,7
20	18,84	99,49%	0,48%	0,03%	28,6	14,6	9,2

Tabelle 7.5.: Die durchschnittlichen Ergebnisse der Simulation mit unterschiedlichen Mengen von Merkmalen bei 10 Simulationsdurchläufen. anon$x$ entspricht dabei dem x. größten Anonymity-Set und Nutzer$x$ gibt die Anzahl der Nutzer mit einem Anonymity-Set der Größe x an.

In diesem Abschnitt soll die Anzahl der gemessenen oder messbaren Merkmale beim Browserfingerprinting und ihr Einfluss auf die sich ergebende Zufallsverteilung bestimmt werden. Für die Simulationen in den anderen Abschnitten wurden fest 10 Merkmale genutzt, um den Fingerprint darstellen. Jedes dieser Merkmale ist statistisch von den anderen Merkmalen unabhängig, also für ein Zehntel der gemessenen Entropie verantwortlich.

Das vollständige Standardisieren einiger Merkmale und daraus folgende Wegfallen dieser Merkmale kann durch das Verringern der gemessenen Merkmale simuliert werden Unbekannte Merkmale, die in der Zukunft zum Browserfingerprinting verwendet werden können oder bereits für das Browserfingerprinting verwendet werden, können durch das Vergrößern der Merkmale simuliert werden.

Wie aus Tabelle 7.5 zu entnehmen ist, hat die Anzahl der Merkmale einen großen Einfluss auf die Anonymität der Nutzer. Das Verringern der gemessenen Merkmale reduziert die Entropie der Fingerprints und die Anzahl der einzigartigen Nutzer stark und bereits das Weglassen eines einzigen Merkmales verbessert die Anonymität der Nutzer deutlich. Werden die Hälfte der Merkmale weggelassen, haben über 80% der Nutzer ein Anonymity-Set der Größe 10+. Das Hinzukommen von neuen Merkmalen verschlechtert die Anonymität der Nutzer deutlich. So sorgt das Hinzukommen von 2 neuen Merkmalen bereits dafür, dass sich die Anzahl der Nutzer mit einem Anonymity-Set der Größe 2-9 oder 10+ jeweils halbiert, und ein Verdoppeln der messbaren Merkmale macht praktisch alle Nutzer eindeutig identifizierbar.

Insgesamt zeigte sich, dass zumindest in der Simulation jedes zusätzliche oder weggelassene Merkmal einen großen Einfluss auf die Anonymität der Nutzer hat. Wie stark dies auf Eckersleys Studie übertragbar ist, ist allerdings unklar. Da seit Eckersleys Studie einige neue Fingerprintingmethoden gefunden wurden, deutet dies aber darauf hin, dass in Eckersleys Studie die Zahl der einzigartigen Nutzer stark unterschätzt wurde. Das Standardisieren der Browser durch Browserhersteller kann allerdings die Anonymität der Nutzer stark verbessern, auch wenn die Entropie selbst sich nur in geringem Maß verändert.

7.5. Schutzparadox

Das Schutzparadox spielt beim Browserfingerprinting eine sehr wichtige Rolle. Beispielsweise beim Verheimlichen von Merkmalen wie dem Deaktivieren von clientseitigen Skriptsprachen und beim Randomisieren von Fingerprints ist es entscheidend für die Frage, ob Nutzer identifizierbar sind.

Nutzer	Merkmale	Entropie	Nutzer1	Nutzer2-9	Nutzer10+	anon1	anon2	anon10
47	1	2,25	3,6%	36,4%	60,0%	20,4	9,8	-
47	3	4,96	62,3%	37,7%	0,0%	4,5	3,4	1,0
47	5	5,49	95,1%	4,9%	0,0%	2,1	1,2	1,0
47	7	5,55	99,6%	0,4%	0,0%	1,1	1,0	1,0
47	9	5,55	100,0%	0,0%	0,0%	1,0	1,0	1,0
470	1	2,48	0,5%	3,1%	96,4%	188,0	108,0	2,0
470	3	6,73	23,3%	46,5%	30,2%	34,3	19,6	8,7
470	5	8,48	76,4%	22,1%	1,4%	10,6	6,1	3,1
470	7	8,82	95,4%	4,6%	0,0%	4,0	3,0	1,3
470	9	8,86	98,7%	1,3%	0,0%	2,6	1,8	1,0
4701	1	2,53	0,0%	0,4%	99,6%	1849,3	1069,3	29,8
4701	3	7,37	5,2%	19,4%	75,4%	353,8	187,4	82,7
4701	5	10,76	43,9%	40,1%	16,0%	95,9	46,1	21,1
4701	7	11,88	84,2%	13,2%	2,6%	40,1	17,8	8,2
4701	9	12,11	95,0%	4,5%	0,5%	15,2	9,6	4,5
47016	1	2,53	0,0%	0,0%	100,0%	18507,9	10764,5	278,2
47016	3	7,54	1,0%	4,5%	94,6%	3530,4	1774,5	917,9
47016	5	11,98	16,5%	32,4%	51,0%	988,3	416,2	183,5
47016	7	14,48	61,0%	27,7%	11,2%	381,2	150,5	58,6
47016	9	15,22	87,9%	8,7%	3,4%	169,5	63,9	42,7

Tabelle 7.6.: Die durchschnittlichen Ergebnisse der Simulation unterschiedlichen Mengen von Merkmalen und Nutzerzahlen bei 10 Simulationsdurchläufen. anon$x$ entspricht dabei dem x. größten Anonymity-Set und Nutzer$x$ gibt die Anzahl der Nutzer mit einem Anonymity-Set der Größe x an.

Das Schutzparadox wird dadurch modelliert, dass die Anzahl der simulierten Nutzer so reduziert wird, dass sie den Teilnehmern einer Maßnahme entspricht. Dazu wurden 47, 470, 4701 und 47016 Nutzer simuliert, was 0,01%, 0,1%, 1% beziehungsweise 10% der Nutzer aus Eckersleys Studie entspricht. Die Schutzwirkung der Maßnahme wird durch das Weglassen von Merkmalen simuliert. Dabei wurde der Bereich von 1 bis 9 Merkmalen komplett abgedeckt, aber nicht komplett in Tabelle 7.6 dargestellt.

Aus der Simulation wurde ersichtlich, dass die Simulation des Schutzparadox nur bei großen Teilnehmerzahlen und stark reduzierten Merkmalsmengen Anonymity-Sets in solcher Größe erreicht, wie sie in der Simulation ohne Gegenmaßnahmen vorkommt. Der Anteil der einzigartigen Nutzer sinkt aber bereits bei 1% Teilnehmern und 5 Merkmalen auf unter 50%. Verbleibt nur ein einziges Merkmal, sind die Nutzer kaum voneinander zu unterscheiden und die Größe des Anonymity-Sets bestimmt sich hauptsächlich durch die Anzahl der Nutzer, da die beiden größten Anonymity-Sets mehr als 50% der Nutzer umfassen. Fällt nur 1 Merkmal weg und werden also 9 Merkmale genutzt, überwiegt das Schutzparadox selbst bei 10% Teilnehmer und verschlechtert die Anonymität der Nutzer.

In der Simulation hat sich also gezeigt, dass das Schutzparadox eine starke Wirkung hat. Mit hohen Nutzerzahlen und weniger Merkmalen konnte die Anonymität deutlich verbessert werden, die Anonymitätsgarantien bleiben aber in den meisten Fällen schwach. Selbst die schwache Anonymitätsforderung, dass keiner der Teilnehmer einen einzigartigen Fingerprint hat, wurde mit keinem genutzten Parametern erfüllt. Das technisch lösbare starke Reduzieren der messbaren Merkmale hatte einen großen Einfluss auf die Anonymität der Nutzer. Dies deutet darauf hin, dass sich zur Umgehung des Schutzparadoxes dieser Ansatz dem Vergrößern der Nutzerzahlen vorzuziehen ist, aber trotzdem eine große Nutzerbasis angestrebt werden sollte.

7.6. Fälschung von Fingerprints

Beim Fälschen von Fingerprints wurde in den vorhergehenden Kapiteln davon ausgegangen, dass ein oft gefälschter Fingerprints für weitere Fälschungen attraktiver wird. Um dieses Verhalten zu simulieren und beobachten zu können, wird zunächst die Simulation um das Fälschen von Fingerprints erweitert. Anschließend wird in Abschnitt 7.6.1 das uneingeschränkte Fälschen von Fingerprints und in Abschnitt 7.6.2 das eingeschränkete Fälschen von Fingerprints in einigen Varianten simuliert und die Ergebnisse werden vorgestellt. Dass ausschließlich die Fälscher Informationen austauschen, wird in Abschnitt 7.6.2 untersucht.

Das Fälschen der Fingerprints an sich lässt sich einfach dadurch darstellen, dass die User-Objekte um die Fähigkeit erweitert werden, ihre Merkmalsausprägungen zu verändern. Das unvollständige Fälschen von Fingerprints wird simuliert, indem Teile der Merkmale als unveränderlich markiert werden können. Damit die User-Objekte eine Fälschung durchführen können, müssen ihnen aber fälschbare Fingerprints bekannt sein und eine Strategie zur Auswahl eines fälschbaren Fingerprints festgelegt werden. Als Quelle für fälschbare Fingerprints wird den User-Objekten Zugriff auf die Ergebnisse eines Fingerprinters gegeben und der am häufigsten gemessene fälschbare Fingerprint wird zur Fälschung ausgewählt. Steht kein geeigneter Fingerprint zur Verfügung, bleibt der Fingerprint unverändert.

Da dies eine Änderung der Fingerprints über einen Zeitraum impliziert, werden mehrere „Ticks“ in der Simulation ausgeführt. In jedem „Tick“ besuchen alle User-Objekte die ihnen zugeordneten Fingerprinter und die Fälscher fälschen nach Abschluss jedes Ticks den für sie optimalen Fingerprint. Nach Beenden jedes „Ticks“ wird die Entropie, die Größe des größten Anonymity-Sets und die Anzahl der einzigartigen Fingerprints ausgewertet.

Für p_{fake} und p_{visit} werden die Werte 100% , 1% und 0,01% genutzt, was allen 470 171 Nutzern, etwa 4 702 beziehungsweise etwa 47 Nutzern entspricht. Dadurch wird der mögliche Wertebereich grob abgedeckt und selbst 0,01% wäre auf die Gesamtheit der Browser bezogen bereits eine große Menge.

7.6.1. Uneingeschränkte Fälschung von Fingerprints

Können Fälscher beliebige Fingerprints fälschen, ist, wie in Abschnitt 5.5 dargestellt, zu erwarten, dass die Fälscher sich dem größten Anonymity-Set zuordnen und dieses erweitern. Dies soll mittels Simulation überprüft werden.

Um das Fälschen von beliebigen Fingerprints zu simulieren, werden User-Objekte mit der Wahrscheinlichkeit p_{fake} als Fälscher markiert. Alle Merkmale des Fälschers werden als veränderlich markiert. Jedes der User-Objekte besucht den Fingerprinter, der dazu genutzt wird, Statistiken zu erstellen. Außerdem wird eine Wahrscheinlichkeit p_{visit} festgelegt, mit der ein User-Objekt seinen Fingingerprint zur Fälschung anbietet. Dies erfolgt durch das Besuchen eines zusätzlich Fingerprinters, von dem alle Fälscher ihre Information erhalten. Da zu erwarten ist, dass sich die Verteilung der Fingerprints schnell stabilisieren, werden nur 5 „Ticks“ ausgeführt.

Die Ergebnisse dieser Simulation sind in Tabelle 7.8 einsehbar. Bei den Werten aus dieser Tabelle kam ein hoher Variationkoeffizient von 20% und 50% nur bei 5 Werten vor, die das zweitgrößte Anonymity-Set mit $p_{fake} = 1\%$ und $p_{visit} = 1\%$ beziehungsweise $p_{visit} = 0,01\%$ angeben. Der Variationkoeffizient der restlichen Werte war bis auf 6 Ausnahmen unter 4% und bei diesen unter 8%.

Tick	Entropie	anon1	anon2	Nutzer1
1	18,338 04	1 130	375	395 185
2	18,337 41	1 130	375	395 134
3	18,336 74	1 185	375	395 135
4	18,337 41	1 130	375	395 134
5	18,336 93	1 130	412	395 135
6	18,336 92	1 130	426	395 135
7	18,336 92	1 130	426	395 135
8	18,336 92	1 130	426	395 135
9	18,336 92	1 130	430	395 135
10	18,336 93	1 130	407	395 135

(a) Die Ergebnisse eines Simulationsdurchlaufs mit 55 Fälschern und 0,1% beobachteten Nutzern

Tick	Entropie	anon1	anon2	Nutzer1
1	18,336 26	1 141	386	395 447
2	18,335 27	1 183	386	395 415
3	18,335 27	1 183	386	395 415
4	18,335 27	1 183	386	395 415
5	18,335 81	1 141	386	395 414
6	18,335 54	1 141	386	395 415
7	18,335 27	1 183	386	395 415
8	18,335 27	1 183	386	395 415
9	18,335 27	1 183	386	395 415
10	18,335 54	1 141	386	395 415

(b) Die Ergebnisse eines Simulationsdurchlaufs mit 42 Fälschern und 0,1% beobachteten Nutzern

Tick	Entropie	anon1	anon2	Nutzer1
1	18,340 99	1 126	389	395 879
2	18,220 28	5 118	1 117	392 066
3	18,220 28	5 118	1 117	392 066
4	18,220 28	5 118	1 117	392 066
5	18,220 28	5 118	1 117	392 066
6	18,220 28	5 118	1 117	392 066
7	18,220 28	5 118	1 117	392 066
8	18,220 28	5 118	1 117	392 066
9	18,220 28	5 118	1 117	392 066
10	18,220 28	5 118	1 117	392 066

(c) Die Ergebnisse eines Simulationsdurchlaufs mit 4 742 Fälschern und 0,1% beobachteten Nutzern

Tick	Entropie	anon1	anon2	Nutzer1
1	18,332 70	1 193	412	395 382
2	18,331 99	1 192	412	395 355
3	18,331 84	1 192	412	395 355
4	18,331 98	1 192	412	395 355
5	18,331 83	1 192	412	395 355
6	18,331 67	1 239	412	395 355
7	18,331 97	1 192	412	395 355
8	18,332 27	1 192	412	395 354
9	18,332 83	1 192	412	395 355
10	18,332 10	1 192	412	395 355

(d) Die Ergebnisse eines Simulationsdurchlaufs mit 47 Fälschern und 0,1% beobachteten Nutzern

Tabelle 7.7.: Detaillierte Ergebnisse der uneingeschränkten Fälschung. anon$x$ entspricht dabei dem x. größten Anonymity-Set und Nutzer1 gibt die Anzahl der einzigartigen Nutzer an.

p_{fake}	p_{visit}	|Fälscher|		1. Tick	2. Tick	3. Tick	4. Tick	5. Tick
100%	100%	470 161	Entropie:	18,34 Bit	0,0 Bit	0,0 Bit	0,0 Bit	0,0 Bit
			anon1:	1 161,2	470 161	470 161	470 161	470 161
			anon2	398,7	0,0	0,0	0,0	0,0
			Nutzer1:	395 728,4	0,0	0,0	0,0	0,0
			Nutzer2-9:	45 361,3	0,0	0,0	0,0	0,0
			Nutzer10+	29 071,3	460 171	460 171	460 171	460 171
100%	1%	470 161	Entropie:	18,34 Bit	0,0 Bit	0,0 Bit	0,0 Bit	0,0 Bit
			anon1:	1 163,1	470 161	470 161	470 161	470 161
			anon2	394,4	0,0	0,0	0,0	0,0
			Nutzer1:	395 696.4	0,0	0,0	0,0	0,0
			Nutzer2-9:	45 465.0	0,0	0,0	0,0	0,0
			Nutzer10+	28 999,6	460 171	460 171	460 171	460 171
100%	0,01%	470 161	Entropie:	18,34 Bit	0,0 Bit	0,0 Bit	0,0 Bit	0,0 Bit
			anon1:	1 152,2	470 161	470 161	470 161	470 161
			anon2	401,1	0,0	0,0	0,0	0,0
			Nutzer1:	395 581,3	0,0	0,0	0,0	0,0
			Nutzer2-9:	45 650,1	0,0	0,0	0,0	0,0
			Nutzer10+	28 929,6	460 171	460 171	460 171	460 171
1%	100%	4 730,9	Entropie:	18,34 Bit	18.21 Bit	18.21 Bit	18.21 Bit	18.21 Bit
			anon1:	1 174,0	5 892,2	5 892,2	5 892,2	5 892,2
			anon2	390,4	386,5	386,5	386,5	386,5
			Nutzer1:	395 665,8	391 912,6	391 912,6	391 912,6	391 912,6
			Nutzer2-9:	45 466,7	44 842,3	44 842,3	44 842,3	44 842,3
			Nutzer10+	29 028,5	33 406,1	33 406,1	33 406,1	33 406,1
1%	1%	4 685,8	Entropie:	18,34 Bit	18,22 Bit	18,22 Bit	18,22 Bit	18,22 Bit
			anon1:	1 155,2	5 681,4	5 681,4	5 681,4	5 681,4
			anon2	395,0	539,2	539,2	539,2	539,2
			Nutzer1:	395 699,1	392 004,2	392 004,2	392 004,2	392 004,2
			Nutzer2-9:	45 573,5	44 939,4	44 939,4	44 939,4	44 939,4
			Nutzer10+	28 888,4	33 217,4	33 217,4	33 217,4	33 217,4
1%	0,01%	4 701,9	Entropie:	18,34 Bit	18,22 Bit	18,22 Bit	18,22 Bit	18,22 Bit
			anon1:	1 148,0	4 702,9	4 739,8	4 855,2	4 703,3
			anon2	399,1	1 136,3	1 136,3	1 058,4	1 136,3
			Nutzer1:	395 715,0	391 995,4	391 995,6	391 995,7	391 995,5
			Nutzer2-9:	45 505,9	44 897,1	44 897,1	44 897,1	44 896,6
			Nutzer10+	28 940,1	33 268,5	33 268,3	33 268,2	33 268,9
0,01%	100%	49,6	Entropie:	18,34 Bit	18,34 Bit	18.34 Bit	18.34 Bit	18.34 Bit
			anon1:	1 160,1	1 209,7	1 209,7	1 209,7	1 209,7
			anon2	394,8	394,7	394,7	394,7	394,7
			Nutzer1:	395 745,0	395 705,9	395 705,9	395 705,9	395 705,9
			Nutzer2-9:	45 556,9	45 550,8	45 550,8	45 550,8	45 550,8
			Nutzer10+	28 859,1	28 904,3	28 904,3	28 904,3	28 904,3
0,01%	1%	43,5	Entropie:	18.34 Bit	18.34 Bit	18.34 Bit	18.34 Bit	18.34 Bit
			anon1:	1 164,1	1 214,4	1 214,4	1 214,4	1 205,4
			anon2	392,9	392,9	392,9	392,9	399,3
			Nutzer1:	395 612,9	395 571,2	395 571,2	395 571,2	395 571,2
			Nutzer2-9:	45 520,0	45 514,5	45 514,5	45 514,5	45 514,5
			Nutzer10+	29 028,1	29 075,3	29 075,3	29 075,3	29 075,3
0,01%	0,01%	45,3	Entropie:	18.34 Bit	18.34 Bit	18.34 Bit	18.34 Bit	18.34 Bit
			anon1:	1 145,1	1 144,9	1 144,9	1 148,9	1 144,9
			anon2	395,4	395,4	395,4	395,4	395,4
			Nutzer1:	395 712,4	395 679,6	395 679,5	395 679,8	395 679,6
			Nutzer2-9:	45 539,6	45 532,3	45 532,5	45 531,9	45 532,3
			Nutzer10+	28 909,0	28 949,1	28 949,0	28 949,3	28 949,1

Tabelle 7.8.: Die durchschnittlichen Ergebnisse der Simulation der uneingeschränkten Fälschung von Fingerprints bei 10 Simulationsdurchläufen. anon$x$ entspricht dabei dem x. größten Anonymity-Set und Nutzer$x$ gibt die Anzahl der Nutzer mit einem Anonymity-Set der Größe x an.

Das Fälschen von beliebigen Fingerprints verhält sich wie erwartet:

Die Fälscher ordnen sich dem als größten wahrgenommenen Anonymity-Set zu. Dadurch wird dieses um die Fälscher vergrößert und die Anzahl der Nutzer mit einem Anonymity-Set der Größe von zehn oder mehr steigt. Die Entropie der sich ergebenden Fingerprints verringerte sich dabei nur bei einer großen Anzahl von Fälschern in einem relevanten Maß. Fälschen alle Nutzer den besten Fingerprint, fällt die Entropie sogar auf 0 Bit.

Die Abwanderung der Fälscher aus ihren ursprünglichen Anonymity-Sets kann daran beobachtet werden, dass sich die Größe der anderen Anonymity-Sets, die Menge der einzigartigen Nutzer und anon2-9 reduziert.

Wird allerdings nur ein kleiner Teil der Browser gemessen, kann es passieren, dass das größte Anonymity-Set nicht erkannt wird und die Fälscher sich einem anderen Anonymity-Set zuordnen. In der Simulation ist dies regelmäßig vorgekommen, wenn nur 0,01% der Nutzer beobachtet werden konnten, was etwa 47 Nutzern entspricht.

Für den Fall, dass das größte Anonymity-Set nicht korrekt erkannt wurde, konnte der Effekt der Fälschungen detaillierter beobachtet werden, indem einzelne Simulationsdurchläufe betrachtet wurden. Dies ist beispielsweise in Tabelle 7.7 dargestellt.

So wird der zentral zur Fälschung ausgewählte Fingerprint zum Fingerprint mit dem größten Anonymity-Set, wenn die Menge der Fälscher größer als das bisher größte Anonymity-Set ist. Ansonsten ordnen sich die Fälscher nicht dem größten Anonymity-Set zu, wodurch es nicht vergrößert oder sogar verkleinert wird.

Insgesamt hat die Simulation die in Abschnitt 5.5 getroffenen Aussagen bestätigt. Damit allerdings das größte Anonymity-Set zuverlässig identifiziert werden kann, musste die Informationsquelle mehr als 0,1% der Fingerprints messen. Fälschten 1% der Nutzer mithilfe derselben Informationquelle ihre Fingerprints, erzeugte dies ein neues größtes Anonymity-Set, das daraufhin leichter aufzufinden war. Dazu war es allerdings nicht notwendig, dass das die Informationquelle den besten Fingerprint korrekt bestimmt hat. Fälschten nur 0,01% der Nutzer mit Hilfe einer einzigen Informationsquelle, die nur 0,01% der Browser messen kann, ihren Fingerprint, wurde der beste Fingerprint nur selten gefunden wird und die Fälscher hatten nicht ausreichen Gewicht, um dies zu ändern.

Die konkreten Zahlen können nicht direkt auf die Realwelt übertragen werden, aber es wurde deutlich, dass entweder eine große Masse von Nutzern ihren Fingerprint fälschen muss oder die Fälscher dies gut koordiniert tun müssen, damit selbsverstärkende Effekte auftreten. Solange nur eine Minderheit ihre Fingerprints fälscht, bleibt der Effekt einer solchen Maßnahme für die Gesamtheit der Nutzer gering.

7.6.2. Eingeschränkte Fälschung von Fingerprints

Von der eingeschränkten Fälschung von Fingerprints, bei der nur bestimmte Fingerprints gefälscht werden können, ist nach Abschnitt 5.6 zu erwarten, dass diese sich ebenso wie beim uneingeschränkten Fälschen von Fingerprints zwar größeren, aber häufig nicht dem größten Anonymity-Sets zuordnen.

Um das Fälschen von geeigneten Fingerprints zu simulieren, wird ein User-Objekt mit der Wahrscheinlichkeit p_{fake} als Fälscher markiert. Die Merkmale des Fälschers werden jeweils mit der Wahrscheinlichkeit p_{fixed} als unveränderlich und ansonsten als veränderlich markiert. Die restliche Simulation wird von der Simulation aus Abschnitt 7.6.1 übernommen.

Das Überprüfen von 100% Anteilen wurde unterlassen, da es sehr aufwändig gewesen wäre, für jeden Fälscher den besten Fingerprint zu bestimmen. Dies macht klar, dass bei einer eventuellen Anwendung dieser Gegenmaßnahme die Berechnung der besten Fingerprints zu den Nutzern verlagert werden sollte und auch die Suche nach Fingerprints optimiert werden sollte.

Die Ergebnisse dieser Simulation sind in Tabelle 7.9 einsehbar. Bei den Werten aus dieser Tabelle kam ein Variationkoeffizient von über 4% nur bei der Anzahl der Fälscher und mit $p_{\text{fake}} = 1\%, p_{\text{visit}} = 1\%, p_{\text{fixed}} = 10\%$ vor, die Entropie und Nutzer$x$ hatten sogar einen Variationkoeffizienten von unter 1,3%.

p_{fake}	p_{visit}	p_{fixed}	|Fälscher|		1. Tick	2. Tick	10. Tick	50. Tick	100. Tick
1%	1%	90%	4 717,0	Entropie:	18.34 Bit	18.34 Bit	18.34 Bit	18.34 Bit	18.34 Bit
				anon1:	1 158,1	1 164,7	1 163,0	1 164,6	1 165,4
				anon2	388,4	388,7	389,9	389,6	388,9
				Nutzer1:	395 682,3	395 598,7	395 322,4	394 867,6	394 647,
				Nutzer2-9:	45 658,9	45 745,4	46 010,0	46 469,5	46 679,6
				Nutzer10+	28 819,8	28 816,9	28 828,6	28 823,9	28 833,6
1%	1%	10%	4 688,2	Entropie:	18,34 Bit	18,25 Bit	18,35 Bit	18,25 Bit	18,25 Bit
				anon1:	1 178,8	3 517,0	3 582,0	3 667,6	3 667,6
				anon2	394,0	602,6	535,0	451,0	453,3
				Nutzer1:	395 516,0	391 294,5	391 236,2	391 225,2	391 234,2
				Nutzer2-9:	45 611,7	46 320,1	46 382,7	46 387,5	46 353,6
				Nutzer10+	29 033,3	32 546,4	32 542,1	32 548,3	32 573,2
1%	0,01%	90%	4 656,9	Entropie:	18,34 Bit	18,34 Bit	18,34 Bit	18,34 Bit	18,34 Bit
				anon1:	1 181,0	1 180,1	1 179,3	1 185,0	1 181,7
				anon2	403,6	403,2	404,7	403,1	402,8
				Nutzer1:	395 668,4	395 667,1	395 659,6	395 623,7	395 591,9
				Nutzer2-9:	45 540,5	45 542,9	45 550,6	45 577,7	45 608,9
				Nutzer10+	28 952,1	28 951,0	28 950,8	28 959,6	28 960,2
1%	0,01%	10%	4 690,6	Entropie:	18,34 Bit	18,27 Bit	18,27 Bit	18,27 Bit	18,27 Bit
				anon1:	1 154,8	1 671,7	1 654,1	1 810,1	1 698,4
				anon2	394,4	1 142,0	1 142,0	1 142,0	1 144,0
				Nutzer1:	395 711,4	392 493,7	391 960,4	391 756,2	391 723,0
				Nutzer2-9:	45 605,2	45 021,0	45 487,3	45 701,0	45 725,2
				Nutzer10+	28 844,4	32 646,3	32 713,3	32 703,8	32 712,8
0,01%	1%	90%	47,2	Entropie:	18,34 Bit	18,34 Bit	18,34 Bit	18,34 Bit	18,34 Bit
				anon1:	1 182,3	1 182,5	1 182,5	1 182,5	1 182,5
				anon2	394,7	394,7	394,7	394,7	394,7
				Nutzer1:	395 650,9	395 649,6	395 647,8	395 643,0	395 641,4
				Nutzer2-9:	45 522,9	45 524,0	45 526,3	45 530,8	45 532,7
				Nutzer10+	28 987,2	28 987,4	28 986,9	28 987,2	28 986,9
0,01%	1%	10%	51,0	Entropie:	18,34 Bit	18,34 Bit	18,34 Bit	18,34 Bit	18,34 Bit
				anon1:	1 151,5	1 179,0	1 177,3	1 176,9	1 179,3
				anon2	399,6	400	399,7	400,3	399,9
				Nutzer1:	395 679,8	395 629,9	395 628,6	395 629,5	395 628,2
				Nutzer2-9:	45 481,6	45 496,1	45 498,7	45 495,7	45 500,3
				Nutzer10+	28 999,6	29 035,0	29 033,7	29 035,8	29 032,5
0,01%	0,01%	90%	47,3	Entropie:	18,34 Bit	18,34 Bit	18,34 Bit	18,34 Bit	18,34 Bit
				anon1:	1 167,2	1 167,2	1 167,2	1 167,2	1 167,2
				anon2	405,6	405,6	405,6	405,5	405,6
				Nutzer1:	395 800,4	395 800,4	395 800,4	395 799,9	395 799,9
				Nutzer2-9:	45 498,6	45 498,6	45 498,5	45 498,8	45 499,0
				Nutzer10+	28 862,0	28 862,0	28 862,1	28 862,3	28 862,1
0,01%	0,01%	10%	48,2	Entropie:	18,34 Bit	18,34 Bit	18,34 Bit	18,34 Bit	18,34 Bit
				anon1:	1,177,6	1,177,7	1,178,1	1,177,4	1,178,3
				anon2	393,4	393,3	393,3	393,3	393,3
				Nutzer1:	395 702,9	395 653,4	395 647,8	395 645,6	395 645,4
				Nutzer2-9:	45 516,5	45 552,0	45 556,4	45 560,1	45 555,5
				Nutzer10+	28 941,6	28 955,6	28 956,8	28 955,3	28 960,1

Tabelle 7.9.: Die durchschnittlichen Ergebnisse der Simulation der eingeschränkten Fälschung von Fingerprints bei 10 Simulationsdurchläufen. anon$x$ entspricht dabei dem x. größten Anonymity-Set und Nutzer$x$ gibt die Anzahl der Nutzer mit einem Anonymity-Set der Größe x an.

Das Fälschen von geeigneten Fingerprints verhält sich also wie erwartet. Das heißt, es senkt die Entropie und die Anzahl der einzigartigen Nutzer deutlich schlechter als die uneingeschränkte Fälschung von Fingerprints und nur wenige Fälscher ordnen sich dem größten Anonymity-Set zu. Schon geringe Mengen unveränderlicher Merkmale schränken die Wirksamkeit dieser Maßnahme ein, und können nur wenige Merkmale gefälscht werden, hat diese Maßnahme kaum Wirkung.

Der deutlichste Unterschied zur uneingeschränten Fälschung von Fingerprints ist, dass sich Verteilung der Fingerprints nicht in einem Schritt stabilisiert, sondern auch nach vielen „Ticks“ nicht stabil ist.

Ein zweiter Unterschied ist, dass beim eingeschränkten Fälschen die Anzahl der einzigartigen Nutzer teilweise um mehr als die Anzahl der Fälscher reduziert wurde. Dies kann dadurch erklärt werden, dass die Fälscher in Ermangelung eines besseren Fingerprints einzigartige Fingerprints fälschten und diese dadurch nicht mehr einzigartig waren.

Insgesamt konnten die Annahmen aus Abschnitt 5.6 bestätigt werden, wobei die Simulation die Notwendigkeit der guten Informationssammlung und der umfangreichen Fälschung unterstreicht.

7.6.3. Fälschung mit Hilfe der Fingerprints anderer Fälscher

p_{fake}	p_{fixed}	|Fälscher|		1.Tick	2.Tick	3.Tick	4.Tick	5.Tick
1%	10%	4673,5	Entropie:	18,338	18,257	18,253	18,253	18,253
			Nutzer1:	84,1786%	83,4879%	83,4372%	83,433%	83,4314%
			Nutzer2-9:	9,686%	9,6454%	9,6279%	9,6243%	9,6229%
			Nutzer10+:	6,1354%	6,8667%	6,9349%	6,9428%	6,9457%
			anon1:	1175,2	3526,0	3571,7	3571,7	3571,7
			anon10:	351,4	364,3	361,0	355,3	353,0
1%	90%	4695,0	Entropie:	18,338	18,338	18,338	18,338	18,338
			Nutzer1:	84,1552%	84,1456%	84,1455%	84,1455%	84,1455%
			Nutzer2-9:	9,6957%	9,7025%	9,7024%	9,7024%	9,7024%
			Nutzer10+:	6,1492%	6,1519%	6,1521%	6,1521%	6,1521%
			anon1:	1165,4	1172,6	1172,6	1172,6	1172,6
			anon10:	348,3	349,4	349,4	349,4	349,4
0,01%	10%	47,8	Entropie:	18,338	18,338	18,338	18,338	18,338
			Nutzer1:	84,1656%	84,1598%	84,1591%	84,159%	84,159%
			Nutzer2-9:	9,6746%	9,6773%	9,6771%	9,6772%	9,6772%
			Nutzer10+:	6,1598%	6,1629%	6,1638%	6,1638%	6,1638%
			anon1:	1163,3	1163,3	1163,3	1163,3	1163,3
			anon10:	345,6	345,6	345,6	345,6	345,6
0,01%	90%	45,0	Entropie:	18,339	18,339	18,339	18,339	18,339
			Nutzer1:	84,1978%	84,1978%	84,1978%	84,1978%	84,1978%
			Nutzer2-9:	9,6563%	9,6563%	9,6563%	9,6563%	9,6563%
			Nutzer10+:	6,1459%	6,1459%	6,1459%	6,1459%	6,1459%
			anon1:	1169,4	1169,4	1169,4	1169,4	1169,4
			anon10:	346,3	346,3	346,3	346,3	346,3

Tabelle 7.10.: Die durchschnittlichen Ergebnisse der Simulation der Fälschung mit Hilfe der Fingerprints anderer Fälscher. anon$x$ entspricht dabei dem x. größten Anonymity-Set und Nutzer$x$ gibt die Anzahl der Nutzer mit einem Anonymity-Set der Größe x an.

In Abschnitt 5.6 wurden die Fälscher selbst als eine der Informationsquellen der Fälschung genannt. Ob dies ausreicht, um die zu fälschenden Fingerprints zu bestimmen, soll in diesem Abschnitt untersucht werden.

In der Simulation zeigte sich, wie in Tabelle 7.10 einsehbar, dass bei 1% Fälschern das größte Anonymity-Set erkannt wird und sich die Fälscher diesem je nach Möglichkeit zuordnen. Bei 0,01% Fälschern konnte

das größte Anonymity-Set allerdings nicht erkannt werden und die Fälscher ordneten sich diesem auch nicht zu. Der Einfluss von p_{fixed} entspricht dem eingeschränkten Fäschen von Fingerprints. In allen Fällen hatte das Fälschen nur einen geringen Einfluss auf die gesamte Verteilung der Fingerprints.

Daran zeigt sich, dass es ausreicht, die Fingerprints der Fälscher zu nutzen, um das größte Anonymity-Set zuverlässig zu bestimmen, solange es nicht nur eine sehr kleine Anzahl von Fälschern existiert.

7.7. Randomisieren von Fingerprints

p_{random}	$p_{\text{randomTick}}$		1. Tick	2. Tick	l0. Tick	20. Tick	50. Tick	100. Tick
100%	100%	AnzKorr	0,0	994,2	1 000,0	1 000,0	1 000,0	1 000,0
		AnzFing	1,0	2,0	10,0	20,0	50,0	100,0
100%	10%	AnzKorr	0,0	0,0	7,9	239,8	943,8	999,7
		AnzFing	1,0	1,65	6,9	13,4	32,9	65,5
100%	1%	AnzKorr	0,0	0,0	0,0	0,0	0,1	10,5
		AnzFing	1,0	1,1	1,9	2,8	5,8	10,5
10%	100%	AnzKorr	349,2	999,8	1 000,0	1 000,0	1 000,0	1 000,0
		AnzFing	1,0	1,7	6,8	13,3	32,7	64,5
10%	10%	AnzKorr	348,2	388,3	667,0	871,6	995,3	1 000
		AnzFing	1,0	1,1	1,8	2,8	5,5	10,4
10%	1%	AnzKorr	347,6	350,9	383,1	421,4	530,4	679,2
		AnzFing	1,0	1,0	1,1	1,2	1,5	2,0
1%	100%	AnzKorr	903,3	1 000,0	1 000,0	1 000,0	1 000,0	1 000,0
		AnzFing	1,0	1,1	1,9	2,8	5,7	10,4
1%	10%	AnzKorr	901,7	911,9	963,1	987,4	999,7	1 000,0
		AnzFing	1,0	1,0	1,1	1,2	1,5	2,0
1%	1%	AnzKorr	905,4	905,5	911,5	919,3	939,0	961,9
		AnzFing	1,0	1,0	1,0	1,0	1,0	1,1

Tabelle 7.11.: Die durchschnittlichen Ergebnisse der Simulation der Randomisierung von Fingerprints bei 10 Simulationsdurchläufen und 1 000 Nutzern.„AnzKorr“ entspricht dabei der Anzahl von korrekt gemessenen Fingerprints und „AnzFing“ gibt die Anzahl der verschiedenen Fingerprints pro Nutzer an.

Das Randomisieren von Fingerprints kann, wie in Abschnitt 5.2 dargestellt, angegriffen werden, indem die Form der Randomisierung gemessen und in den Fingerprint einbezogen wird. Um diese Angriffsmöglichkeit zu untermauern und die Randomisierung von Fingerprints auf weitere Schwachstellen zu überprüfen, wird die Randomisierung von Fingerprints und der vorgestellte Angriff simuliert. Dazu muss zunächst die Grundsimulation an mehreren Stellen angepasst werden.

Die Anonymitätsmaße der dadurch erkannten Fingerprints kann dem Abschnitt 7.5 entnommen werden, indem die randomisierten Merkmale vernachlässigt werden und nur die randomisierenden Nutzer betrachtet werden.

Das Messen der Randomisierung wird dadurch modelliert, dass der Fingerprinter um diese Fähigkeit erweitert wird. Dazu wird der Fingerprinter, der in der Basissimulation für jeden Tick isoliert einen Fingerprint misst, durch einen Fingerprinter ersetzt, der die in jedem Tick gemessenen Fingerprints über die User-Id miteinander in Verbindung bringt und diese zu einem neuen Fingerprint zusammenführt. Wenn die Merkmalsausprägung eines Merkmals in allen Fingerprints übereinstimmt, wird sie beim Zusammenführen der Fingerprints beibehalten. Ansonsten wird dem Merkmal der Spezialwert „None“ zugewiesen, der eine Änderung des Fingerprints anzeigt.

p_{fake}	p_{visit}	p_{fixed}	|Fälscher|		1.Tick	2.Tick	10.Tick	50.Tick	100.Tick
1%	1%	0%	4699,0	Entropie:	18,338	18,317	18,317	18,317	18,317
				Nutzer1:	84,1515%	82,8371%	82,827%	82,8291%	82,8277%
				Nutzer2-9:	9,6852%	10,9945%	11,0079%	11,0046%	11,0048%
				Nutzer10+:	6,1633%	6,1684%	6,1651%	6,1663%	6,1675%
				anon1:	1163,2	1161,9	1165,4	1164,4	1163,3
				anon10:	347,7	348,1	348,9	346,8	348,3
1%	1%	90%	4693,4	Entropie:	18,339	18,339	18,338	18,337	18,337
				Nutzer1:	84,1905%	84,173%	84,1179%	84,0219%	83,978%
				Nutzer2-9:	9,6726%	9,6888%	9,7441%	9,8382%	9,8824%
				Nutzer10+:	6,137%	6,1382%	6,138%	6,1399%	6,1396%
				anon1:	1160,1	1161,9	1160,6	1161,8	1160,5
				anon10:	346,5	345,6	346,2	345,8	346,1
1%	0,01%	0%	4731,1	Entropie:	18,338	18,276	18,275	18,276	18,276
				Nutzer1:	84,1716%	83,3684%	83,3686%	83,3684%	83,3681%
				Nutzer2-9:	9,67%	9,5322%	9,5338%	9,5334%	9,5327%
				Nutzer10+:	6,1583%	7,0994%	7,0975%	7,0982%	7,0991%
				anon1:	1172,8	1180,3	1181,5	1193,0	1171,2
				anon10:	349,3	346,7	345,3	345,5	345,3
1%	0,01%	90%	4701,3	Entropie:	18,339	18,339	18,339	18,339	18,339
				Nutzer1:	84,1879%	84,1876%	84,1857%	84,1785%	84,1715%
				Nutzer2-9:	9,6447%	9,6451%	9,6472%	9,6552%	9,6602%
				Nutzer10+:	6,1673%	6,1673%	6,1671%	6,1663%	6,1682%
				anon1:	1158,3	1157,1	1159,6	1156,2	1156,4
				anon10:	349,4	349,7	348,8	349,4	350,3
0,01%	1%	0%	49,2	Entropie:	18,338	18,338	18,338	18,338	18,338
				Nutzer1:	84,1563%	84,1393%	84,1395%	84,1394%	84,1394%
				Nutzer2-9:	9,6982%	9,7153%	9,7148%	9,7146%	9,7151%
				Nutzer10+:	6,1455%	6,1454%	6,1457%	6,146%	6,1455%
				anon1:	1167,6	1167,9	1167,6	1167,8	1167,7
				anon10:	348,9	348,9	349,0	348,9	348,9
0,01%	1%	90%	44,8	Entropie:	18,339	18,339	18,339	18,339	18,339
				Nutzer1:	84,1811%	84,181%	84,1802%	84,1794%	84,1789%
				Nutzer2-9:	9,6645%	9,6647%	9,6654%	9,6658%	9,6667%
				Nutzer10+:	6,1543%	6,1543%	6,1543%	6,1548%	6,1544%
				anon1:	1181,7	1181,6	1181,7	1181,6	1181,7
				anon10:	346,0	346,0	346,0	346,0	346,0
0,01%	0,01%	0%	46,8	Entropie:	18,338	18,338	18,338	18,338	18,338
				Nutzer1:	84,1812%	84,168%	84,1686%	84,1681%	84,1686%
				Nutzer2-9:	9,6644%	9,678%	9,6773%	9,6769%	9,6767%
				Nutzer10+:	6,1544%	6,154%	6,1541%	6,155%	6,1547%
				anon1:	1190,0	1189,9	1189,9	1189,9	1189,9
				anon10:	342,0	342,0	342,0	342,1	342,0
0,01%	0,01%	90%	51,4	Entropie:	18,338	18,338	18,338	18,338	18,338
				Nutzer1:	84,1801%	84,1801%	84,18%	84,18%	84,1799%
				Nutzer2-9:	9,661%	9,661%	9,6611%	9,6611%	9,6613%
				Nutzer10+:	6,1589%	6,1589%	6,1589%	6,1589%	6,1588%
				anon1:	1172,6	1172,6	1172,6	1172,5	1172,5
				anon10:	345,4	345,4	345,4	345,4	345,5

Tabelle 7.12.: Die durchschnittlichen Ergebnisse der Simulation der Fälschung von zufälligen Fingerprints bei 10 Simulationsdurchläufen. anon$x$ entspricht dabei dem x. größten Anonymity-Set und Nutzer$x$ gibt die Anzahl der Nutzer mit einem Anonymity-Set der Größe x an.

Um das Randomisieren der Browser darzustellen, wird das Nutzerobjekt um die Fähigkeit erweitert, Merkmale zu randomisieren. Dazu werden Merkmale mit der Wahrscheinlichkeit p_{random} als randomisierbar markiert. Am Ende jedes Ticks werden Merkmale, die entsprechend markiert sind, mit der Wahrscheinlichkeit $p_{\text{randomTick}}$ auf eine zufällige Merkmalsausprägung gesetzt. Die Aussagekräftigkeit dieser Simulation wird allerdings dadurch eingeschränkt, dass nur 10 Merkmale genutzt werden, um den Fingerprint darzustellen.

Das Erstellen der Statistiken wird anders vorgenommen als in den restlichen Simulationen. Betrachtet wird das korrekte Messen der Randomisierung und die Anzahl der unterschiedlichen Fingerprints der Nutzer. Die Entropie und Einzigartigkeit der Fingerprints wird nach dem Messen der Randomisierung berechnet und es sollen nur Nutzer betrachtet werden, die ihren Fingerprint randomisieren können. Kann der Fingerprinter festellen, welche Merkmale randomisiert sind, hat er den korrekten Fingerprint gemessen.

Um den Fingerprinter zur Erkennung der Randomisierung in der Simulation zu testen, wurden 100 Ticks mit 1 000 Nutzern durchgeführt. Die Simulation wurde 10-mal durchgeführt und die Ergebnisse aller Durchläufe werden gemittelt.

Die Ergebnisse der Simulation können in Tabelle 7.11 eingesehen werden. Der Variationkoeffizient der gemessenen Werte lag bis auf 5 Ausnahmen unter 6%, davon hatten 4 Werte einen Variationskoeffizienten von unter 31% und ein Ausreißer einen von 300%, der sich allerdings auf einen Wert von nur 0,2 bezog.

Werden alle Merkmalsausprägungen bei jedem Tick neu ausgewürfelt, konnte die Randomisierung für fast alle Nutzer mit wenigen Messungen ermittelt werden und mit 10 Messungen sogar für alle Nutzer gemessen werden. Dass dafür nicht bereits 2 Messungen ausreichen, kann dadurch erklärt werden, dass beim Randomisieren einer Merkmalausprägung zufällig eine Merkmalsausprägung ausgewürfelt wurde, die bereits vorhanden war, und sich der Fingerprint nicht tatsächlich verändert.

Wird nur selten randomisiert, hat der Fingerprinter deutliche Probleme, die Randomisierung des Fingerprints korrekt zu messen. So konnte bei einem $p_{\text{randomTick}}$ von 10% auch nach 20 Ticks und bei einem $p_{\text{randomTick}}$ von 1% selbst nach 100 Ticks die Randomisierung aller Nutzer nicht korrekt gemessen werden. Bei einem kleinen $p_{\text{randomTick}}$ wurden allerdings auch für jeden Nutzer nur wenige verschiedene Fingerprints gemessen, wodurch die Kernfunktion der Randomisierung nicht tatsächlich zum Tragen kommt.

Dass bereits beim ersten Tick Fingerprints korrekt gemessen worden sind, kann dadurch erklärt werden, dass bei manchen Randomisierern kein Merkmal als randomisierbar markiert wurde und die Einstufung als Nichtrandomisierer, die bei allen Nutzern zu Beginn angenommen wird, in diesem Fall korrekt war. Dies ist ein unerwünschter Effekt, da so bei bei einem p_{random} von 1% und 10% über 90% bzw 33% nicht tatsächlich randomisierten, obwohl sie als randomisierende Nutzer markiert waren. Dieser Effekt muss auch bei der Fälschung von Fingerprints auftreten, was die ermittelten Ergebnisse verfälscht.

Die Methode, trotz Randomisierung einen Fingerprint messen zu können, funktioniert also wie erwartet. Das heißt, sie kann eine umfangreiche und häufige Randomisierung ausgleichen, hat aber Probleme, seltene Randomisierung von vielen Merkmalen zu erkennen.

7.8. Fälschung zufälliger Fingerprints

Die Fälschung von zufälligen Fingerprints wurde in Abschnitt 5.10 erwähnt, als Kombination von Randomisierung und Fälschung. Dies wurde mit einer Simulation detaillierter betrachtet.

Die Simulation dafür ist ähnlich zur in Abschnitt 7.6.2 genutzen Simulation zur eingeschränkten Fälschung von Fingerprints. Die Nutzer sind mit der Wahrscheinlichkeit p_{fake} Fälscher, ihre Merkmale werden jeweils mit der Wahrscheinlichkeit p_{fixed} als unveränderbar markiert und die Nutzer bieten mit der Wahrscheinlichkeit p_{visit} ihren Fingerprint zur Fälschung an. Diese Simulation hat eine hohe Komplexität und konnte deshalb aufgrund hohem Ressourcenbedarfs weniger Parameter abdecken als erwünscht. Für p_{fake} und p_{visit} wurden die Werte 1% und 0,01% simuliert, p_{fixed} hatte 0%,10%,50% und 90% als mögliche Werte. Simulationen wurden mit allen Permutation dieser Werte durchgeführt und zusätzlich teilweise 100% für p_{fake} und p_{visit} genutzt.

Da es eine große Menge von möglichen Parametern, erhebbaren Statistiken und Kombinationen aus diesen

gibt, werden die Ergebnisse an dieser Stelle nicht komplett angegeben, sondern auf besonders interessante Aspekte konzentriert. Die vollständigen Tabellen mit Variationskoeffizienten sind im Anhang A.9 und die vollständigen Daten sind in dem Ordner „Simulation/random_fake“ von Anhang B zu finden.

Als erster Aspekt wird eine globale Sicht auf die Simulation vorgestellt. Dazu wird die Entropie und die Größe der Anonymity-Sets der Nutzer betrachtet. Wie in Tabelle 7.12 einsehbar haben wenig Fälscher auch nur wenig direkten Einfluss auf die Verteilung der Fingerprints.

p_{fake}	p_{visit}	p_{fixed}		1.Tick	2.Tick	10.Tick	50.Tick	100.Tick
1%	1%	90%	$\lvert Fingerprints\rvert$:	1,0	1,0	1,2	1,9	2,3
			AnzWechsel :	0,0	0,0	0,3	1,8	3,5
1%	1%	50%	$\lvert Fingerprints\rvert$:	1,0	1,6	5,8	25,4	46,8
			AnzWechsel :	0,0	0,6	5,0	26,9	54,3
1%	1%	10%	$\lvert Fingerprints\rvert$:	1,0	2,0	9,9	49,4	98,3
			AnzWechsel :	0,0	1,0	8,9	48,6	98,2
1%	1%	0%	$\lvert Fingerprints\rvert$:	1,0	2,0	10,0	50,0	99,9
			AnzWechsel :	0,0	1,0	9,0	49,0	99,0
1%	0,01%	90%	$\lvert Fingerprints\rvert$:	1,0	1,0	1,0	1,1	1,1
			AnzWechsel :	0,0	0,0	0,0	0,1	0,1
1%	0,01%	50%	$\lvert Fingerprints\rvert$:	1,0	1,1	2,2	7,2	13,3
			AnzWechsel :	0,0	0,1	1,2	6,4	12,9
1%	0,01%	10%	$\lvert Fingerprints\rvert$:	1,0	1,9	8,9	43,8	87,4
			AnzWechsel :	0,0	0,9	7,9	42,9	86,7
1%	0,01%	0%	$\lvert Fingerprints\rvert$:	1,0	2,0	10,0	50,0	99,9
			AnzWechsel :	0,0	1,0	9,0	49,0	99,0
0,01%	1%	90%	$\lvert Fingerprints\rvert$:	1,0	1,0	1,3	1,9	2,5
			AnzWechsel :	0,0	0,0	0,3	1,6	3,1
0,01%	1%	50%	$\lvert Fingerprints\rvert$:	1,0	1,6	6,1	26,4	48,6
			AnzWechsel :	0,0	0,6	5,3	28,3	57,2
0,01%	1%	10%	$\lvert Fingerprints\rvert$:	1,0	2,0	9,9	49,1	97,8
			AnzWechsel :	0,0	1,0	8,9	48,4	97,8
0,01%	1%	0%	$\lvert Fingerprints\rvert$:	1,0	2,0	10,0	50,0	99,9
			AnzWechsel :	0,0	1,0	9,0	49,0	99,0
0,01%	0,01%	90%	$\lvert Fingerprints\rvert$:	1,0	1,0	1,0	1,1	1,1
			AnzWechsel :	0,0	0,0	0,0	0,1	0,1
0,01%	0,01%	50%	$\lvert Fingerprints\rvert$:	1,0	1,1	2,1	7,0	12,8
			AnzWechsel :	0,0	0,1	1,1	6,1	12,3
0,01%	0,01%	10%	$\lvert Fingerprints\rvert$:	1,0	1,9	9,0	44,3	88,1
			AnzWechsel :	0,0	0,9	8,0	43,4	87,6
0,01%	0,01%	0%	$\lvert Fingerprints\rvert$:	1,0	2,0	10,0	50,0	99,9
			AnzWechsel :	0,0	1,0	9,0	49,0	99,0

Tabelle 7.13.: Die durchschnittlichen Ergebnisse der Simulation der Fälschung von zufälligen Fingerprints bei 10 Simulationsdurchläufen. $|Fingerprints|$ gibt an, wie viele unterschiedlich Fingerprint durchschnittlich von den Fälschern angenommen wurden. AnzWechsel gibt an, wie häufig die Fälscher durchschnittlich ihren Fingerprint änderten.

Beobachtet werden konnte, dass sich die Verteilung der Fingerprints bei einem p_{fixed} von 90% kaum veränderte. Dies kann dadurch erklärt werden, dass wie in Tabelle 7.13 einsehbar in diesem Fall für die meisten Fälscher keine fälschbaren Fingerprints für die Fälscher gefunden werden konnten und keine Fälschung stattfand.

p_{fake}	p_{visit}	p_{fixed}		1.Tick	2.Tick	10.Tick	50.Tick	100.Tick
1%	1%	90%	$\lvert Nutzer_{tick} \rvert$:	9,47	9,91	9,8	9,92	9,73
			$\lvert Nutzer_{gesamt} \rvert$:	9,47	12,36	18,51	22,67	23,99
			$\lvert Fälscher_{gesamt} \rvert$:	1,08	1,14	1,31	1,47	1,52
			AntGefälscht :	3,9463%	4,2735%	5,4372%	7,0702%	7,8268%
1%	1%	50%	$\lvert Nutzer_{tick} \rvert$:	9,52	9,85	10,28	10,52	10,29
			$\lvert Nutzer_{gesamt} \rvert$:	9,52	17,46	69,35	208,75	312,82
			$\lvert Fälscher_{gesamt} \rvert$:	1,09	1,5	4,79	18,89	33,24
			AntGefälscht :	3,9067%	5,0344%	10,5852%	26,1839%	39,1943%
1%	1%	0%	$\lvert Nutzer_{tick} \rvert$:	9,43	11,37	12,26	11,49	11,45
			$\lvert Nutzer_{gesamt} \rvert$:	9,43	19,8	103,76	512,6	1005,88
			$\lvert Fälscher_{gesamt} \rvert$:	1,08	2,16	10,84	54,04	107,41
			AntGefälscht :	3,9333%	5,3295%	12,5813%	35,0381%	53,7219%
1%	0,01%	90%	$\lvert Nutzer_{tick} \rvert$:	9,19	9,0	9,35	8,79	8,9
			$\lvert Nutzer_{gesamt} \rvert$:	9,19	9,44	11,33	14,82	16,56
			$\lvert Fälscher_{gesamt} \rvert$:	1,08	1,09	1,14	1,23	1,28
			AntGefälscht :	2,9677%	2,9802%	3,0673%	3,4013%	3,7196%
1%	0,01%	50%	$\lvert Nutzer_{tick} \rvert$:	9,46	14,03	14,74	13,29	14,49
			$\lvert Nutzer_{gesamt} \rvert$:	9,46	18,31	62,08	199,52	301,32
			$\lvert Fälscher_{gesamt} \rvert$:	1,08	3,64	20,18	68,07	100,15
			AntGefälscht :	3,9813%	4,0027%	4,1903%	5,015%	5,9722%
1%	0,01%	0%	$\lvert Nutzer_{tick} \rvert$:	9,17	112,23	122,44	118,43	109,63
			$\lvert Nutzer_{gesamt} \rvert$:	9,17	120,4	978,92	3606,06	5140,98
			$\lvert Fälscher_{gesamt} \rvert$:	1,08	99,99	870,98	3136,33	4199,73
			AntGefälscht :	3,9009%	3,9255%	4,0999%	5,0059%	5,9813%
0,01%	1%	90%	$\lvert Nutzer_{tick} \rvert$:	9,14	6,28	9,28	7,45	9,78
			$\lvert Nutzer_{gesamt} \rvert$:	9,14	10,04	12,08	15,75	17,02
			$\lvert Fälscher_{gesamt} \rvert$:	1,0	1,0	1,0	1,0	1,0
			AntGefälscht :	0,0729%	0,0806%	0,0995%	0,1315%	0,1425%
0,01%	1%	50%	$\lvert Nutzer_{tick} \rvert$:	20,36	19,52	20,23	19,72	10,03
			$\lvert Nutzer_{gesamt} \rvert$:	20,36	37,24	96,95	253,95	356,23
			$\lvert Fälscher_{gesamt} \rvert$:	1,01	1,01	1,04	1,2	1,35
			AntGefälscht :	0,1405%	0,2833%	0,7564%	1,8982%	2,614%
0,01%	1%	0%	$\lvert Nutzer_{tick} \rvert$:	10,21	14,46	9,42	16,5	14,1
			$\lvert Nutzer_{gesamt} \rvert$:	10,21	23,67	96,15	465,13	941,53
			$\lvert Fälscher_{gesamt} \rvert$:	1,0	1,02	1,08	1,56	2,12
			AntGefälscht :	0,0851%	0,2076%	0,7924%	2,606%	3,983%
0,01%	0,01%	90%	$\lvert Nutzer_{tick} \rvert$:	7,28	7,28	7,34	5,47	6,53
			$\lvert Nutzer_{gesamt} \rvert$:	7,28	7,28	7,61	11,41	12,68
			$\lvert Fälscher_{gesamt} \rvert$:	1,0	1,0	1,0	1,0	1,0
			AntGefälscht :	0,0724%	0,0724%	0,0764%	0,1181%	0,1314%
0,01%	0,01%	50%	$\lvert Nutzer_{tick} \rvert$:	8,21	9,39	11,38	6,79	8,17
			$\lvert Nutzer_{gesamt} \rvert$:	8,21	13,44	40,33	108,53	169,2
			$\lvert Fälscher_{gesamt} \rvert$:	1,0	1,04	1,15	1,53	1,86
			AntGefälscht :	0,0657%	0,1168%	0,3353%	0,8694%	1,3701%
0,01%	0,01%	0%	$\lvert Nutzer_{tick} \rvert$:	10,02	4,03	6,65	10,15	10,35
			$\lvert Nutzer_{gesamt} \rvert$:	10,02	13,05	85,33	504,7	968,75
			$\lvert Fälscher_{gesamt} \rvert$:	1,0	1,85	9,26	31,64	41,71
			AntGefälscht :	0,0929%	0,1145%	0,4891%	1,8281%	2,9092%

Tabelle 7.14.: Die durchschnittlichen Ergebnisse der Simulation der Fälschung von zufälligen Fingerprints bei 10 Simulationsdurchläufen. $\lvert Nutzer_{tick} \rvert$ ist die durchnittliche Anzahl von Nutzern, mit denen sich ein Fälscher in diesem Tick in einem Anonymity-Set befunden hat. $\lvert Nutzer_{gesamt} \rvert$ ist die durchnittliche Anzahl von Nutzern, mit denen sich ein Fälscher in einem der Ticks in einem Anonymity-Set befunden hat. $\lvert Fälscher_{gesamt} \rvert$ ist die durchnittliche Anzahl von Fälschern, mit denen sich ein Fälscher in einem der Ticks in einem Anonymity-Set befunden hat. AntGefälscht ist der Anteil der Nutzer, die sich in mindestens einem der Ticks mit einem Fälscher in einem Anonymity-Set befunden haben.

Für ein p_{fixed} von 0% wurde in den meisten Fällen beobachtet, dass sich der Anteil der einzigartigen Nutzer um das ein bis zweifache von p_{fake} reduziert. Gleichzeitig erhöhte sich der Anteil der Nutzer mit einem Anonymityset der Größe 2-9 um etwa diese Zahl. Dies kann dadurch erklärt werden, dass ein Großteil der Fälscher zu Beginn einzigartig war und einen einzigartigen Fingerprint gefälscht hat. Dadurch reduziert sich die Anzahl der Nutzer mit einzigartigem Fingerprint doppelt und es entsteht ein neues Anonymity-Set mit 2 oder mehr Nutzern.

Der Ausnahmefall zeigte sich, als eine große Anzahl von Fälschern eine kleine Menge von Fingerprints fälschte. Hier reduzierte sich der Anteil der einzigartigen Nutzer um weniger als p_{fake} und der Anteil der Nutzer mit einem Anonymity-Set der Größe 10+ um etwa diese Zahl. Dies kann dadurch erklärt werden, dass viele Fälscher dieselben Fingerprints gefälscht hatten und so zwar die ursprünglich einzigartigen Fälscher nicht mehr einzigartig waren, aber statt vieler kleiner Anonymity-Sets wenige große gebildet wurden.

Insgesamt bleibt der Einfluss der zufälligen Fälschung aus dieser Perspektive in allen Fällen gering.

Der zweite Aspekt ist die die direkte Vermischung der Identität der Fälscher und normalen Nutzer. Diese kann für einen Tick oder über den gesamten Zeitraum hinweg gemessen werden. Inherhalb eines Ticks besaßen die Fälscher, wie in Tabelle 7.14 sichtbar, in fast allen Fällen nur ein kleines durchschnittliches Anonymity-Set von weniger als 25 Nutzern.

Der Ausnahmefall zeigte sich, als 1% Fälscher die Fingerprints von 0,01% der Nutzer fälschte. Dabei wurden große durchnittliche Anonymity-Sets gemessen von teilweise über 100 Nutzern gemessen. Dies ist dieselbe Konstellation von Parametern, die auch Anonymity-Sets in der Größe von 10+ statt 2-9 erzeugt hat. Daher ist anzunehmen, dass die Fälscher selbst den Großteil des gemessenen Anonymity-Sets bildeten.

Wird $|\text{Nutzer}_{\text{gesamt}}|$ betrachtet, also mit wievielen Nutzern jeder Fälscher über alle Ticks hinweg durschnittlich einen Fingerprint teilte, stieg diese Anzahl mit zunehmenden Ticks deutlich, erreichte nach 100 Ticks bei mehreren Parametern über 900 Nutzer und blieb bis auf 2 Ausnahmen unter 1 200 Nutzern. Je weniger Merkmale unveränderlich waren, desto größer war diese Zahl.

Das heißt, dass die Fälscher sich bei ausreichend geringem Anteil von unveränderlichen Merkmalen zwar mit einer steigenden Anzahl von Nutzern in Verbindung bringen, dieser Prozess aber langsam ist.

Der Anteil der Nutzer, deren Fingerprint mindestens einmal von einem Fälscher übernommen wurde, war bereits beim ersten Tick ein Vielfaches von p_{fake} und stieg über die Zeit. Ein geringe Menge an unveränderlichen Merkmalen und viele Nutzer, die ihren Fingerprint zur Fälschung zur Verfügung stellten, beschleunigte dieses Wachstum. Ein kleiner Anteil von Fälschern konnte so bereits nach 100 Ticks die Fingerprints eines relativ großen Anteiles der Nutzer übernehmen. Bei einem p_{fake} von 0,01%, p_{visit} von 1% und p_{pfixed} von 50% wurde so 2,6% der Nutzer mit einem Fälscher in Verbindung gebracht, was mehr als das 250-Fache von p_{fake} ist. Mit einem p_{fake} von 1% und einem p_{visit} von 1% wurden sogar mehr als 50% der Nutzer mit einem Fälscher in Verbindung gebracht.

Der dritte Aspekt ist die die indirekte Vermischung der Indentität der Fälscher und normalen Nutzer. Dies tritt auf, wenn ein Browserfingerprintingskript den Nutzer auf anderem Weg verfolgen kann und alle Nutzer, die sich zu einem Zeitpunkt einen Fingerprint teilten zu einer Identät zusammenfasst. Vorstellbar wäre dies für ein System das Browserfingerprinting für Cookieregeneration nutzt.

Ein solches System würde in einer Simulation ohne Gegenmaßnahmen maximal das Größte Anonymity-Set als ein Nutzer erkennen und nicht mehr als 1 500 Nutzer unter einer Identität zusammenfassen. Für etwa 83% Prozent der Nutzer würde dies fehlerfrei funktionieren.

In einer Simulation, in der allerdings zufällige Fingerprints gefälscht werden, kann diese Zahl erheblich steigen. Als Maß für diese indirekte Verknüpfung wurde berechnet, wieviele Nutzer und Fälscher durchnittlich ein bis drei Zwischenschritte von jedem Fälscher entfernt sind. Beispielsweise ist ein Nutzer einen Zwischenschritt, im Englischen „hop“, von einem Fälscher entfernt, wenn dieser Fälscher mit einem weitern Fälscher in Verbindung steht und dieser Fälscher wiederum mit dem Nutzer in Verbindung steht. Da die Berechnung dieser Statistik sehr aufwändig ist, konnte sie nicht für alle Simulationsdurchläufe erhoben werden.

p_{fake}	p_{visit}	p_{fixed}		1.Tick	2.Tick	10.Tick	50.Tick	100.Tick
0,01%	100%	0%	touched1hop:	6,4852	12,8499	101,8123	500,2464	972,7629
			forgedTouched1hop:	0,0	0,0	0,0144	0,0379	0,0833
			touched2hop:	6,4852	12,8499	101,8123	500,2464	972,7629
			forgedTouched2hop:	0,0	0,0	0,0144	0,0379	0,0833
			touched3hop:	6,4852	12,8499	101,8123	500,2464	972,7629
			forgedTouched3hop:	0,0	0,0	0,0144	0,0379	0,0833
0,01%	100%	50%	touched1hop:	8,5532	13,3703	55,1945	183,1118	318,0557
			forgedTouched1hop:	0,0	0,0039	0,0039	0,0285	0,0721
			touched2hop:	8,5532	13,3703	55,1945	194,9628	363,6869
			forgedTouched2hop:	0,0	0,0039	0,0039	0,0285	0,0917
			touched3hop:	8,5532	13,3703	55,1945	194,9628	368,0046
			forgedTouched3hop:	0,0	0,0039	0,0039	0,0285	0,0917
0,01%	1%	0%	touched1hop:	10,2108	23,6791	102,2235	697,023	1828,8755
			forgedTouched1hop:	0,0042	0,0157	0,0915	0,8927	2,3739
			touched2hop:	10,2108	23,6791	102,9062	884,0906	3304,1861
			forgedTouched2hop:	0,0042	0,0157	0,0915	1,2039	4,7118
			touched3hop:	10,2108	23,6791	102,9062	932,6957	4690,1823
			forgedTouched3hop:	0,0042	0,0157	0,0915	1,2826	7,3014
0,01%	1%	50%	touched1hop:	20,3553	40,4132	105,9125	359,3449	626,7492
			forgedTouched1hop:	0,009	0,009	0,0372	0,2603	0,5727
			touched2hop:	20,3553	40,4132	105,9125	427,6205	893,8096
			forgedTouched2hop:	0,009	0,009	0,0372	0,291	0,7484
			touched3hop:	20,3553	40,4132	105,9125	427,6205	915,9014
			forgedTouched3hop:	0,009	0,009	0,0372	0,291	0,7666
0,01%	0,01%	0%	touched1hop:	10,0191	20,5346	609,8271	7105,6579	13062,3299
			forgedTouched1hop:	0,0	0,8547	35,9263	45,8	45,8
			touched2hop:	10,0191	20,5346	2259,094	8595,0	13678,0
			forgedTouched2hop:	0,0	0,8547	45,795	45,8	45,8
			touched3hop:	10,0191	20,5346	2299,5	8595,0	13678,0
			forgedTouched3hop:	0,0	0,8547	45,8	45,8	45,8
0,01%	0,01%	50%	touched1hop:	8,211	18,4377	71,5865	388,6301	801,9411
			forgedTouched1hop:	0,0043	0,0413	0,2365	1,3264	2,4767
			touched2hop:	8,211	18,4377	80,3932	741,5687	1746,7344
			forgedTouched2hop:	0,0043	0,0413	0,268	2,2698	4,1302
			touched3hop:	8,211	18,4377	80,408	772,8321	1838,4465
			forgedTouched3hop:	0,0043	0,0413	0,268	2,3232	4,3088

Tabelle 7.15.: Die durchschnittlichen Ergebnisse der Simulation der Fälschung von zufälligen Fingerprints bei 10 Simulationsdurchläufen. touched$x$hop entspricht dabei der Anzahl der Nutzer, mit denen sich ein Fälscher durchnittlich über x Zwischenschritte in Verbindung brachte. forgedTouched$x$hop entspricht dabei der Anzahl der Fälscher, mit denen sich ein Fälscher durchnittlich über x Zwischenschritte in Verbindung brachte.

Bringen sich die Fälscher mit fast keinen anderen Fälschern in Verbindung, werden die Fälscher auch durch Zwischenschritte nur mit keinen oder fast keinen zusätzlichen Nutzern in Verbindung gebracht. Bringen sich die Fälscher allerdings mit vielen anderen Fälschern in Verbindung, vervielfacht sich die durchschnittliche Anzahl von Nutzern, die mit dem Fälscher in Verbindung gebracht werden.

Wie in Tabelle 7.14 einsehbar brachten sich beispielsweise die Fälscher bei den Parametern $p_{\text{fake}} = 0,01\%$, $p_{\text{visit}} = 1\%$ und $p_{\text{fixed}} = 0\%$ durchschnittlich direkt mit 942 Nutzern in Verbindung, von denen 2,1 Fälscher waren. Tabelle 7.15 zeigt nun, dass die Anzahl der Nutzer, mit denen sich die Fälscher in Verbindung brachten, nach dem ersten Zwischenschritt über diese Fälscher auf 1 828 stieg und sich also fast veroppelte. Bei dem zweiten Zwischenschritt wurden über 3,4 Fälscher 3 304 Nutzer erreicht und mit dem dritten über 5,7 Fälscher sogar 4 690 Nutzer erreicht.

Ein Sonderfall wurde bei den Parametern $p_{\text{fake}} = 0,01\%$, $p_{\text{visit}} = 0,01\%$ und $p_{\text{fixed}} = 0\%$ festgestellt. Hier brachten sich schnell fast alle Fälscher direkt miteinander in Verbindung, wodurch sich die Fälscher bereits bei der ersten Zwischenschritt mit 13 062 Nutzern in Verbindung brachten. Dieser Wert stieg mit weiteren Zwischenschritten kaum an und entsprach etwa der Menge der gesamten Nutzer, die mit Fälschern in Verbindung stehen. Dies kann dadurch erklärt werden, dass die Fälscher und alle Nutzer, die mit diesen in Verbindung stehen, zu einer einzigen Identität verschmelzen.

Wird dieses Verhalten auf Parameter übertragen die nicht Simuliert werden konnten, wird klar dass auch ein stark abgeschwächter Effekt dafür sorgt, dass sehr viele Nutzer über einen einzigen Zwischenschritte mit den Fälschern verbunden sind. Mit den Parametern $p_{\text{fake}} = 1\%$, $p_{\text{visit}} = 1\%$ und $p_{\text{fixed}} = 0\%$ würden beispielsweise bereits beim ersten Zwischenschritt etwa 100 Fälscher mit jeweils durchnittlich 1 000 Nutzern einbezogen. Die Anzahl der sich ergebenden Nutzer wird sicher nicht 100 000 betragen, aber selbst ein Zehntel dieser Zahl wäre über 2% der gesamten Nutzerzahl. Dass es aufgrund hoher Rechenzeiten nicht möglich war, diese Werte zu bestimmen, sorgt zwar einerseits dafür, dass dies Vermutung nicht bestätigt werden konnte, bestärkt sie aber.

Der letzte Aspekt, der betrachet wird, ist die Anonymität der Fälscher, unter dem Aspekt des Randomisierens. Dazu wird gemessen, für welchen Anteil der Fälscher die unveränderlichen Merkmale also M_{fest} korrekt erkannt werden konnten.

p_{fake}	p_{visit}	p_{fixed}		1.Tick	2.Tick	5.Tick	10.Tick	50.Tick	100.Tick
1%	1%	0%	Erkannt:	0,0%	6,9%	90,22%	99,93%	100,0%	100,0%
1%	1%	50%	Erkannt:	0,1%	11,65%	54,86%	67,38%	79,77%	82,81%
1%	1%	90%	Erkannt:	34,93%	37,05%	40,83%	43,75%	50,88%	53,62%
1%	0,01%	0%	Erkannt:	0,0%	6,96%	90,58%	99,91%	100,0%	100,0%
1%	0,01%	50%	Erkannt:	0,09%	1,94%	12,15%	20,26%	38,38%	46,01%
1%	0,01%	90%	Erkannt:	34,89%	34,96%	35,13%	35,42%	36,87%	38,16%
0,01%	100%	0%	Erkannt:	0,0%	8,06%	91,8%	100,0%	100,0%	100,0%
0,01%	100%	50%	Erkannt:	0,22%	21,66%	77,41%	83,7%	83,91%	83,91%
0,01%	100%	90%	Erkannt:	36,69%	44,76%	53,41%	55,12%	55,12%	55,12%
0,01%	1%	0%	Erkannt:	0,0%	8,67%	92,93%	100,0%	100,0%	100,0%
0,01%	1%	50%	Erkannt:	0,0%	11,79%	56,26%	68,52%	82,91%	86,12%
0,01%	1%	90%	Erkannt:	34,72%	37,24%	39,95%	43,1%	50,35%	54,22%
0,01%	0,01%	0%	Erkannt:	0,0%	7,35%	90,27%	100,0%	100,0%	100,0%
0,01%	0,01%	50%	Erkannt:	0,0%	2,01%	10,47%	18,39%	39,31%	46,42%
0,01%	0,01%	90%	Erkannt:	36,38%	36,38%	36,55%	36,55%	38,28%	39,59%

Tabelle 7.16.: Die durchschnittlichen Ergebnisse der Simulation der Fälschung von zufälligen Fingerprints bei 10 Simulationsdurchläufen. Erkannt bezeichnet dabei den Anteil der Fälscher, deren M_{fest} korrekt erkannt werden konnte.

Dabei wurde, wie in Tabelle 7.16 sichtbar, deutlich, dass mit allen Parametern für einen großen Anteil von Fälschern M_{fest} erkannt werden konnte. Je größer p_{fixed} dabei war, desto besser konnte M_{fest} erkannt

werden. Allerdings änderten Fälscher, wie in Tabelle 7.13 einsehbar, bei großen M_{fest} auch ihren Fingerprint seltener.

Dadurch wird deutlich, dass das Erkennen der Randomisierung auch bei dieser Technik ein Faktor ist, der einbezogen werden muss, soll Anonymität für die Nutzer gesichert werden. Dies kann dadurch geschehen, dass die Randomisierung so gestaltet wird, dass M_{fest} schwerer zu erkennen ist, oder, wie in Kapitel 7.5 dargestellt, eine ausreichende Anonymität trotz Erkennen von M_{fest} gegeben ist.

Insgesamt erreicht eine gleichverteilte Auswahl der zu fälschenden Fingerprints in den meisten Betrachtungsweisen nur einen geringen Effekt. So verringert sich die Anzahl der einzigartigen Nutzer um maximal das Doppelte der Anzahl der Fäscher und über die Zeit brachten sich die Fälscher bis auf einen Spezialfall nur mit etwa 1000 andern Nutzern in Verbindung. Ein im Verhältnis zum Anteil der Fälscher große Menge von Nutzern wurde, für kurze Zeit mit mindestens einem Fälscher in Verbindung gebracht. Dieser geringe Effekt tritt aber trotz der Verbindung des Fälschens mit dem Randomisieren auf.

Mit einem kleinen Anteil von veränderlichen Merkmalen konnten die Fälscher ihren Fingeprint häufig ändern, was dafür sorgt, dass eine Verfolgung dieser Nutzer über Bowsefingerprinting kaum möglich ist. Mit zusätzlichen Verfolgungsmechanismen konnte der Anteil der unveränderlichen Merkmale von vielen Fälschern wenigen Messungen erkannt werden, was zeigt, dass der Fingerprint der Nutzer trotz dieser Maßnmahme zur Identifizierung von Nutzern dienen könnte. Soll diese Technik eingesetzt werden, muss also sichergestellt werden, dass die unveränderlichen Merkmale nicht erkannt werden können oder dies die Anonymität der Nutzer nicht gefährdet.

Wurde ein sehr umfangreiches Vereinigen der Identitäten durch Browserfingerprintingskripte von Nutzern mit identischen Fingerprints angenommen, verschmolzen die Identität der Fälscher mit einer großen Menge von Nutzern, wenn ihre Identität mit der anderer Fälscher vereinigt wurde. Es konnte sogar beobachtet werden, dass die Identität fast aller Fälscher und somit fast aller Nutzer, die mit einem Fälscher in Verbindung gekommen sind, verschmolz.

Je nachdem welche Fingerprintingalgorithmen und zusätzliche Verfolgungsmöglichkeiten angenommen werden, kann das Fälschen von zufälligen Fingerprints auch bei geringen Nutzerzahlen die Anonymität der Nutzer in verschiedenem Maß verbessern, solange nur ein geringer Teil der Merkmale unveränderlich ist.

7.9. Zusammenfassung

In diesem Kapitel wurden einige Maßnahmen gegen Browserfingerprinting mithilfe einer Simulation untersucht.

Dazu wurde zuerst in Abschnitt 7.1 eine **Basissimulation** aufgebaut und in Abschnitt 7.2 Simulationen zunächst **ohne Maßnahmen gegen Browserfingerprinting** durchgeführt. Die Ergebnisse aus Eckersleys Studie konnten dabei nachvollzogen werden, obwohl die Parameterwahl dafür unerwartet problematisch und nicht exakt begründet war.

Dabei wurde klar, dass die Angabe der Entropie nicht ausreicht, um die Verteilung der Fingerprints zu beschreiben. So spielt die Form der Zufallsdichte eine wichtige Rolle für beispielsweise den Anteil der einzigartigen Nutzer.

Anschließend wurde in Abschnitt 7.3 und 7.4 **Variationen in der Anzahl der Nutzer und der Attribute** simuliert. Dabei hat sich gezeigt, dass die Anzahl der Nutzer und Merkmale für die Anonymität der Nutzer bestimmend ist. So wurde zumindest für die genutzte Verteilung klar, dass keine globale Einzigartigkeit der Fingerprints gegeben ist, aber fast jeder Nutzer einzigartig ist, wenn nur kleine Nutzerzahlen betrachtet werden. Obwohl schon kleine Änderungen der Nutzerzahlen die Anonymität deutlich beeinflussen, können

sich kleine Gruppen von Nutzern ohne Rücksicht auf die anderen Nutzer dem Browserfingerprinting entziehen. Kann die Anzahl der Merkmale verringert werden, die im Fingerprint verwertet werden, verbessert sich die Anonymität der Fingerprints deutlich und verschlechtert sich andererseits erheblich, wenn sich die Anzahl der Merkmale vergrößert.

Dabei wurde deutlich, dass die Anonymität der Nutzer zu stark von der Stichprobengröße einer Studie abhängt, als dass so ermittelte Werte ohne Angabe der Stichprobengröße verwendet werden sollten. Die Entwicklung dieser Werte über die wachsende Größe der Stichprobe könnte eine bessere Angabe der Anonymität erlauben. Diese Erkenntnis stellt auch durch die Simulation erworbenen Erkenntnisse weiter infrage, da bei festen Anteilen von Nutzern je nach Stichprobengröße andere Effekte auftreten.

Der Effekt des Schutzparadoxes wurde in Abschnitt 7.5 simuliert. Dabei wurde bestätigt, dass das Schutzparadox eine starke Wirkung hat, und durch hohe Nutzerzahlen und wenig verbleibenden Merkmalen abgeschwächt wird. Die verbleibenden Merkmale zu reduzieren hatte zumindest bei der genutzten Zufallsverteilung eine stärkere Wirkung als die Nutzerzahlen zu steigern.

Bei der Simulation des **uneingeschränkten Fälschens von Fingerprints** in Abschnitt 7.6.1 ordneten sich die Fälscher wie erwartet dem größten Anonymity-Set zu. Damit die selbstverstärkende Wirkung auftritt, müssen allerdings viele Fälscher teilnehmen oder das größte Anonymity-Set muss korrekt erkannt werden. Dies kann sichergestellt werden, indem viele Fingerprints gemessen werden. Das Fälschen senkt die Entropie der Verteilung der Fingerprints zudem nur geringfügig, solange nur wenig Fälscher teilnehmen.

Das eingeschränte Fälschen von Fingerprints, das in Abschnitt 7.6.2 simuliert wurde, hatte einen noch geringeren Einfluss auf die Entropie der sich ergebenden Fingerprints. Die Fälscher verteilen sich aber auf viele verschiedene Fingerprints und schützen so Nutzer, die nicht selbst fälschen. Damit diese Maßnahme funktioniert, müssen viele Fingerprints zum Fälschen gesammelt und es dürfen nur wenig Merkmale unveränderlich sein.

Die Simulation **der Fälschung mit Hilfe der Fingerprints anderer Fälscher** in Abschnitt 7.6.3 brachte weitgehend dieselben Ergebnisse wie das eingeschränkte Fälschen von Fingerprints. Bei einem sehr geringen Anteil von Fälschern konnte der optimale Fingerprint allerdings nicht gefunden werden und die Fälscher ordneten sich einem nicht optimalen Fingerprint zu.

Die Simulation **des Randomisierens von Fingerprints** in Abschnitt 7.7 konnte zeigen, dass ein Nachvollziehen der Randomisierung von Fingerprints mit Merkmalen, die bei jedem Tick neu ausgewürfelt wurden, mit wenigen Messungen möglich war. Bei Fingerprints mit nur selten randomisierten Merkmalen war das Messen der Randomisierung trotz vieler Messungen nur teilweise erfolgreich. In diesen Fällen hat der Nutzer allerdings auch nur selten seinen Fingerprint gewechselt. In dieser Simulation wurden lediglich nur 10 Merkmale randomisiert. Würden mehr Merkmale genutzt, könnte die einzelnen Merkmale selten geändert und trotzdem eine häufige Änderung des Fingerprints erzielt werden. Dadurch würde das Erkennen der Randomisierung stark behindert.

Als letztes wurde **das Fälschen von zufälligen Fingerprints** in Abschnitt 7.8 simuliert. Dabei wurde festgestellt, dass sich die Verteilung der Fingerprints mit realistischen Parametern kaum veränderte. Trotz der Kombination von Fälschung und Randomisierung, nahmen die Fälscher keine einzigartigen Fingerprints an. Die unveränderlichen Merkmale konnten allerdings teilweise mit wenigen Messungen erkannt werden, weswegen nur wenig Merkmale unveränderlich sein dürfen, um eine Anonymität der Nutzer zu gewährleisten. Wenn Identitäten von Nutzern mit identischem Fingerprint umfangreich verknüpft wurden und die Fälscher untereinander stark verknüpft waren, konnten sich die Fälscher mit einem großen Teil der Nutzer

in Verbindung bringen.

In der Simulation selbst konnten keine groben Fehler gefunden werden, da die Ergebnisse in sich schlüssig sind. Ein unbeabsichtigter Effekt sorgte aber für eine Verzerrung der Ergebnisse, da nicht alle als Randomisierer oder Fälscher markierten Nutzer auch tatsächlich randomisierten beziehungsweise fälschten. Aufgrund der Verzerrungen durch die gewählte Nutzeranzahl und Zufallsverteilung sind die Ergebnisse der Simulation allerdings sowieso als Tendenzen zu verstehen.

Deutlich wurde auch, dass eine detailliertere Modellierung der Realität die Ergebnisse verbessern würde. Beispielsweise würde ein Einbeziehen der Veränderungen der Fingerprints durch Updates, eine bessere Nachbildung der Verteilung der Fingerprints, größere Nutzerzahlen und eine Modellierung von Nutzerinteressen die Simulation deutlich realistischer machen.

Trotz dieser Mängel konnten die in Kapitel 5 getätigten Aussagen, soweit sie getestet wurden, mit dem genutzten Modell bestätigt werden.

8 SCHLUSS

Die Hauptfragestellung dieser Arbeit war, ob es Maßnahmen gibt, die gegen die Nutzerverfolgung mittels Browserfingerprinting schützen. Solche Maßnahmen sollten genauer beschrieben werden.

In diesem Schlusskapitel werden in Abschnitt 8.1 die Ergebnisse dieser Arbeit zusammengefasst und in Abschnitt 8.2 Schlussfolgerungen aus dieser Arbeit gezogen. In Abschnitt 8.3 werden mögliche Ansätze für weitere Forschung dargestellt. Abschließend folgt in Abschnitt 8.4 wie vorgeschrieben ein Lebenslauf des Autors.

8.1. Zusammenfassung

In der Einleitung wurde in Kapitel 1 dargestellt, dass das Browserfingerprinting dazu dienen kann, Nutzer zu verfolgen und somit ein Problem für die Privatsphäre der Nutzer im Internet darstellt. Nachdem die Fragestellung der Arbeit geklärt wurde, wurde die Vorgehensweise erläutert und die Struktur der Arbeit vorgestellt.

In der Zusammenfassung des Forschungsstands wurde im Kapitel 2 die Funktionsweise des Browserfingerprintings, mit dem Browserfingerprinting verwandte Techniken und Typen des Browserfingerprintings beschrieben. Außerdem wurden verschiedene Nutzungmöglichkeiten beschrieben, die sich nicht nur auf Nutzerverfolgung beschränken. Zwar wurden einige Ansätze für Maßnahmen gegen Browserfingerprinting gefunden, die bereits angedacht, erforscht oder umgesetzt wurden, insgesamt sind sie allerdings nicht oder nur schlecht erforscht. Eine Ausnahme ist das Randomisieren von Fingerprints, das in zwei Arbeiten behandelt und bereits mit Browserfingerprintingskripten aus der Industrie getestet wurde.

Das von Eckersley in seiner Arbeit aus dem Jahr 2010 nur kurz beschriebene Modell des Browserfingerprintings wurde in Kapitel 3 mit einer neuen Notation versehen und so erweitert, dass es auch in komplexeren Fällen angewendet werden kann.

Nach der Beschreibung der Stärken und Schwächen des Browserfingerprintings wurden in Kapitel 4 Maßnahmen zur weiteren Untersuchung ausgewählt:

- Das Standardisieren von Browsern
- Automatische Browserupdates
- Das Randomisieren von Fingerprints
- Das Verheimlichen von Merkmalen
- Das uneingeschränkte Fälschen von Fingerprints
- Das eingeschränkte Fälschen von Fingerprints
- Das Blockieren von Kommunikation
- Das Filtern von Kommunikation
- Kontextspezifische Fingerprints
- Kombination verschiedener Maßnahmen

Mit der Analyse der zur Untersuchung festgelegten Maßnahmen wurde in Kapitel 5 die Untersuchung der Gegenmaßnahmen begonnen.

Dabei wurde festgestellt, dass das Standardisieren von Browsern nur unter unrealistischen Bedingungen einen vollständigen Schutz aller Nutzer bieten könnte. Mit realistichen Annahmen garantiert es keinen Erfolg und kann in Einzelfällen sogar Nutzer deanonymisieren. Je mehr Merkmale standardisiert werden und je mehr Nutzer an der Standardisierung teilnehmen, desto besser funktioniert diese allerdings. Der Effekt des Standardisierens ist prinzipiell durch Faktoren wie unterschiedliche Nutzerinteressen begrenzt, es bleibt aber trotzdem eine sinnvolle Maßnahme.

Die automatischen Browserupdates sind ein Spezialfall der Standardisierung von Browsern, wobei lediglich die Browserversion standardisiert wird und somit ein geringer positiver Effekt mit hoher Sicherheit auftritt.

Bei der Untersuchung des Randomisieren von Fingerprints wurden zwei Möglichkeiten gefunden, Nutzer trotz Randomisierung zu erkennen. Für beide Möglichkeiten müssten die Browserfingerprintingskripte allerdings auf die Randomisierung eingehen, wodurch das Randomisieren den Nutzer besser identifizierbar macht. Die Effektivität des Schutzes steigt, wenn viele Merkmale randomisiert werden, die einzelnen Merkmale selten randomisiert werden und mehr Nutzer randomisieren.

Das Verheimlichen von Merkmalen wurde anhand des Deaktivierens von clientseitigen Skripten untersucht. Ob dies den Nutzer schützen kann, hängt dabei stark von der Verteilung der Fingerprints ab. Das Verheimlichen von wenigen Merkmalen bei wenigen Nutzern führt dazu, dass deren Identifzierung erleichter wird.

Bei beiden Fälschungsvarianten trat das Problem auf, dass es schwierig ist, zur Fälschung nutzbare Fingerprints zu finden. Dadurch kann eine Fälschung zwar garantieren, dass Nutzer nicht einzigartig sind, aber nicht, dass sie auch ein großes Anonymity-Set haben. Je mehr Nutzer fälschen, je mehr Merkmale gefälscht werden können und je mehr Fingerprints bekannt sind, desto bessere Fingerprints können zur Fälschung gefunden werden. Damit eine Fälschung überhaupt möglich ist, müssten Datenbanken erstellt werden, in der Fälscher nach geeigneten Fingerprints suchen könnten.

Das erfolgreiche Blockieren von Fingerprintingskripten kann Anwender vollständig vor Browserfingerprinting schützen, kann aber erschwert werden und ist von der Zuverlässigkeit von Blockadelisten abhängig.

Das Filtern von Fingerprintingskripten ist ein Verheimlichen von Merkmalen und unterscheidet sich im besten Fall nicht vom Deaktivieren von Skriptsprachen. Bei Misserfolg wird Nutzer identifizierbarer.

Es wurde auch keine Kombination aus Maßnahmen gefunden, die einen Schutz aller Nutzer vor Browserfingerprinting bietet.

In Kapitel 6 wurden einige der Ansätze praktisch untersucht.

Das Deaktivieren von Skriptsprachen hat sich dabei als erstaunlich schwache Maßnahme gegen Browserfingerprinting herausgestellt. So kann ein auf CSS und HTML basierendes Fingerprintingskript umfangreich Informationen über den Browser ermitteln. Diese zwar neue und umständliche Technik stellt die Schutzwirkung des Deaktivierens von Skripten in Frage.

Das Täuschen von Fingerprintingskripten wurde getestet und war technisch überraschend leicht durchzuführen. Allerdings scheiterte das Fälschen des besten Fingerprints aus Eckerleys Studie daran, dass dieser nicht vollständig bekannt war.

Beim Untersuchen des Filterns und Blockierens von Kommunikation wurde festgestellt, dass dies einfach möglich ist, aber auch leicht behindert werden kann.

Bestimmte auf Javascript basierende Analyseskripte konnten automatisch durch abgeänderte Versionen ersetzt werden, die dem Nutzer die Analysergebnisse zur Weitergabe präsentieren konnten und statt echten Analyseergebnissen vordefinierte oder vom Nutzer vorgegebenen Werte an den Analyseserver schickte.

Abschließend wurde in Kapitel 7 eine Simulation des Browserfingerprintingvorgangs erstellt und das Fälschen und Randomisieren von Fingerprints simuliert. Dabei traten die erwarteten Effekte auf und die in den vorherigen Kapiteln getätigten Aussagen wurden bestätigt.

Zusätzlich wurde festgestellt, dass die Entropie der Fingerprints nur ein schlechtes Maß für die Anonymität der Nutzer ist, da die gemessene Entropie stark von der Stichprobengröße abhängt und die Anonymität der Nutzer stark von der Form der Zufallsverteilung abhängt.

8.2. Schlussfolgerung

Keine der untersuchten Maßnahmen schützt unter realistischen Annahmen alle Nutzer gegen alle Formen des Browserfingerprintings. Daraus kann aber nicht geschlossen werden. dass solche Maßnahmen nicht existieren. Werden die Ansprüche eingeschränkt, indem beispielsweise nur bestimmte Formen des Browserfingerprintings oder nur bekannte Fingerprintingskripte betrachtet werden, gibt es Maßnahmen, die Nutzer gut vor Browserfingerprinting schützen.

In Kombinationen konnten sich die Maßnahmen allerdings wirkungsvoll ergänzen und sind teilweise nur mit hohem Aufwand zu umgehen. Aber auch das birgt das Risiko, dass Nutzer nicht ausreichend vor einer Identifizierung geschützt werden. Die Ergebnisse von Eckersley und Tillmann legen allerdings nahe, dass Inaktivität auf jeden Fall dazu führt, dass Browserfingerprintig auf Massenbasis geschieht. Deshalb sollten Gegenmaßnahmen auch ohne Erfolgsgarantie angewendet werden.

Kann das Browserfingerprinting nicht prinzipell, sondern nur bei bestimmten in der Industrie eingesetzten Fingerprintingskripten verhindert werden, ist es von den vorhanden Ressourcen abhängig, ob Browserfingerprinting auf einer Massenbasis möglich ist oder nicht. Kann gezielt eine Situation geschaffen werden, in der es unmöglich ist, diese Ressourcen aufzubringen, kann der massenhaften Einsatz von Browserfingerprinting verhindert werden. Deshalb wird ein Vorgehen vorgeschlagen, das leicht umzusetzende Gegenmaßnahmen so anordnet, dass sie jeweils eine möglichst große Wirkung entfalten.

Die Basis für dieses Phasenmodell soll eine Browsererweiterung sein, die automatisiert die Anwendung von Maßnahmen vorschlägt und in mehreren Stufen ausgebaut werden kann.

In der ersten Phase soll die Erweiterung vorhandene und populäre Gegenmaßnahmen vorschlagen und so eine Basis für weiter Maßnahmen aufbauen, während die Nutzer einen grundlegenden Schutz haben. Solche wären beispielsweise das Deaktivieren von clientseitigen Skripten, das Filtern und Blockieren von Kommunikation und automatische Browserupdates.

Beim Browser Firefox können diese Maßnahmen durch ein einfaches Installieren und Konfigurieren von vorhandenen Browserplugins durchgeführt werden und somit schnell implementiert werden. Dadurch haben diese bereits eine recht große Nutzerbasis und das Privay-Paradox wird sowohl für die Nutzer dieser Browsererweiterung als auch für die bisherigen Nutzer abgemildert. Dies bietet die Möglichkeit, sie gegen bestimmte Fingerprintingtypen zu schützen oder verbreite Fingerprintingskripte zu behindern. Schlagen diese Maßnahmen fehl, kann dies allerdings für die Nutzer nachteilig sein. Sie schützen aber voraussichtlich die Anwender in erheblichen Umfang vor der Verfolgung mittels Browserfingerprinting.

Ein individueller Schutz gegen Browserfingerprinting stellt aber aufgrund erwartungsgemäß kleiner Nutzerzahlen selbst bei Erfolg keine allgemeine Bedrohung für das Browserfingerprinting dar, kann allerdings die

Verbreitung der hier vorgeschlagenen Erweiterung unterstützen. Nutzer, die bereits diese Kombination von Maßnahmen gegen Browserfingerprinting anwenden, werden durch das Abmildern des Schutzparadox noch zusätzlich geschützt. Diese Phase dient aber nicht nur dazu, Schutz vor Browserfingerprintring zu bieten, sondern auch dazu, eine große Nutzerbasis aufzubauen.

Zusätzlich dazu kann in dieser Phase versucht werden, die Erweiterung zu nutzen, um Fingerprintingskripte zu finden und diese zu blockieren. Das Crawlen der Webseiten würde dabei von den Nutzer erledigt und würde auch passwortgeschützte Bereiche untersuchen. Durch die Gegenmaßnahmen wie den Einsatz von NoScript wären Zugriffe auf beispielsweise Javascriptfunktionen allerdings schwerer zu analysieren als bei dem Vorgehen von FPDetective oder FPGuard. Würde auffälliges Verhalten registriert, könnte ein Bericht an eine zentrale Mailingliste versendet werden. So könnten die Filterregeln der Plugins aktuell gehalten und die Erforschung von Browserfingerprinting unterstützt werden. Diese Funktionalität könnte als zusätzliches Plugin erstellt und von der Haupterweiterung vorgeschlagen werden, damit Nutzer mit Privatsphärebedenken diese nicht nutzen müssten.

Durch die Nutzung von vorhandenen und verbreiteten Browserplugins soll der Effekt des Schutzparadox verringert werden. Deswegen sollten Listenerweiterungen in dieser Phase nur in den eingebundenen Plugins eingebracht und am besten in der Standartinstallation der Plugins aktiviert sein. Soll also beispielsweise auch der Schutz von NoScript um das Einschränken von CSS und HTML erweitert werden, muss dies in dieser Phase zunächst über das Plugin NoScript geschehen.

Die zweite Phase soll beginnen, wenn das Plugin es geschafft hat, genug Nutzer zu gewinnen, um das Schutzparadox ausreichend abmildern zu können. In dieser Phase können neue Maßnahmen gegen das Browserfingerprinting eingesetzt werden oder vorhandene Maßnahmen verändert werden. Zusätzlich dient diese Phase dazu, den Nutzern Kontrolle über den Fingerprints seines Browsers zu geben.

Für jede Gegenmaßnahme muss einzeln entschieden werden, ob die Nutzerbasis der Erweiterungung ausreicht, um das Schutzparadox soweit abzumildern, dass es die Schutzwirkung einer Gegenmaßnahme nicht aufhebt. Die Möglicheit, diese Maßnahmen für alle oder nach einer vorhergehenden Anfrage für bestimmte Teile der Nutzerbasis innerhalb kurzer Zeit zu aktivieren, erlaubt es allerdings, eine solche Abschätzung zu treffen und das Schutzparadox zu überwinden.

Soweit noch nicht geschehen, könnte in dieser Phase CSS und HTML nun auch direkt eingeschränkt werden, um das darauf basierende Fingerprinting zu behindern.

Eines der Probleme in der ersten Phase ist aber, dass Merkmale verheimlicht werden und damit das Fehlen von Merkmalen die Anwender trotz teilweisen Ausgleichs des Schutzparadox von vielen anderen Nutzern unterscheidet. Um diese Lücken zu füllen, können gefälschte oder zufällige Werte genutzt werden. Da in dieser Phase die Infrastruktur zum Füllen dieser Lücken erst geschaffen werden muss, sollen zunächst zufällige Werte zurückgegeben werden.

Dies könnte bei Firefox über die Plugins AdBlock und Greasemonkey geschehen, indem statt eines Analyseskripts ein Skript ausgeführt wird, dass ein randomisiertes Ergebnis zurückliefert. Dieser Ansatz hätte den Vorteil, dass so eine große Menge von Merkmalen zur Randomisierung zur Verfügung stände. Diese könnten somit jeweils selten randomisiert werden und es gäbe nur wenig feste Merkmale, die den Nutzer indentifizierbar machen könnnten.

Auch Merkmale wie der Useragent könnten randomisiert werden, um die festen Merkmale weiter zu reduzieren. Dabei müsste allerdings vorsichtig vorgegangen werden, da dies Fehler in der Darstellung von Webseiten provozieren kann. Um die Einschränkungen für den Nutzer möglichst gering zu halten, sollte mit dem Randomisieren von zusätzlichen Merkmalen auch ein System für die Meldung und Behebung solcher Fehler mitgeliefert werden.

Um ein Beobachten der Randomisierung zu erschweren, sollte dabei aber nicht bei jeder Anfrage ein zufälliger Fingerprint erzeugt werden, sondern der Fingerprint nur bei Start und Ende einer privaten Browsingsession oder nach dem Löschen aller Cookies neu generiert werden. Um zu verhindern, dass Einbruchserkennungssysteme auf Änderungen des Fingerprints reagieren, sollten bestimmte Fingerprintingskripte und Webseiten von der Randomisierung ausgenommen werden können.

Der Schutz würde sich weiterhin nur auf individuelle Nutzer beschränken und nur gegen bestimmte Browserfingerprintingskripte schützen. Die Analyse mit Fingerprintingskripten, die nicht auf eine Randomisierung eingehen, kann dadurch allerdings komplett verhindert werden. Geht ein Browserfingerprintingskript aber auf Randomisierung ein, muss durch eine ausreichend große Nutzerbasis und ein somit ausreichend abgemildertes Schutzparadox eine Identifizierung verhindert werden.

Eine Randomisierung, die von einem Teil der Nutzer angewendet wird, verschlechtert allerdings die Situation der restlichen Nutzer, da die Randomisierer aus deren Anonymity-Sets abwandern und einen zufälligen und somit nicht plausiblen Fingerprint annehmen.

In der dritten Phase soll die in der zweiten Phase vorgenommene Randomisierung erweitert werden, um nicht nur individuelle Nutzer zu schützen. Dies soll geschehen, indem die Nutzer trotz Randomisierung ausschließlich vorhandene Fingerprints fälschen. Für Fingerprinter, die nicht auf Randomisierung eingehen, werden so die Identitäten der Nutzer mit diesem Fingerprint vermischt. Die Qualität der über Browserfingerprinting aufgebauten Nutzerprofile selbst wird also durch Fehleinträge bedroht.

Als Basis für eine Fälschung muss, wie in Abschnitt 5.6 beschrieben, eine von Plugins durchsuchbare Bibliothek von Browserfingerprints geschaffen werden. Werden erreichbare Fingerprints gefunden, können diese angenommen werden. Wenn auch die Fingerprints von unbeteiligten Nutzern gesammelt werden, können sogar diese geschützt werden, obwohl sie sich nicht sebstständig vor Browserfingerprinting schützen.

Das Fälschen von Fingerprints würde das Risiko verringern, dass das Randomisieren als solches erkannt wird, und der Schutz für den individuellen Nutzer wird gestärkt. Durch das Provozieren von Fehleinträgen auch bei unbeteiligten Nutzern ist der Schutz nicht nur auf die Nutzer des Plugins beschränkt und bereits eine relativ kleine Zahl von Nutzern kann den Wert des Browserfingerprintings zur Nutzerverfolgung im Allgemeinen stark reduzieren. Durch die bevorzugte Auswahl häufiger Fingerprints kann dies auch zu einer Standardisierung der Fingerprints genutzt werden.

Je mehr Merkmale gefälscht werden können, desto besser ist der Schutz vor Browserfingerprinting. Schlägt die Fälschung fehl, hat der Nutzer zwar einen einzigartigen Fingerprint, zur Identifizierung muss aber zusätzlich die Randomisierung erkannt werden, um den Nutzer mit den verbleibenden Merkmalen von den restlichen Anwendern der Erweiterung unterscheiden zu können.

In der vierten Phase soll der Nutzer auch gegen unbekannte Browserfingerprintingskripte geschützt werden. Dazu soll der tatsächliche Zustand der Browser durch Standardisieren von Merkmalen mit hoher Entropie, also die Schriftarten, Farbeinstellungen und installierten Plugins angepasst werden. Auch wenn das Plugin diese Konfigurationen vollständig standardisieren könnte, müsste es Nutzerwünsche respektieren, um nicht an Akzeptanz zu verlieren.

Um den Standard nicht manuell bestimmen zu müssen und den für die Nutzer besten Standard zu finden, soll diese Standardisierung durch Fälschung stattfinden. Damit eine Fälschung von Fingerprints eine Standardisierung als Nebeneffekt hat, muss ein Rückkopplungseffekt zwischen Fälschung und Auswahl der Fingerprints eintreten. Damit diese Rückkopplungseffekte möglichst stark sind, soll eine Datenbank aufgebaut werden, die speziell diesem Zweck dient.

Dazu soll die vorgeschlagene Erweiterung die Browserkonfigurationen an einen zentralen Server schicken, wo sie gespeichert und die Datenbank zur Auswahl der Konfigurationen bilden sollen. Jeder Browser soll diejenige Konfiguration annehmen, die nicht mit Nutzerwünschen im Widerspruch steht und den geringstmöglichen Informationsgehalt hat. Damit die Rückkopplungseffekte schnell eintreten, sollte diese Standardisierung regelmäßig und in kurzen Abständen vorgenommen werden.

Eine solche Standardisierung verbessert auch die Effizienz der in der ersten bis dritten Phase getroffen Maßnahmen und bietet auch dann Schutz, wenn die vorhergehenden Phasen umgangen werden.

In der fünften Phase soll der sich herausgebildete Standard den Browserherstellern und Standardisierungsstellen vorgeschlagen werden. Durch die in der vierten Phase genutzten Rückkopplungseffekte haben

die so vorgeschlagenen Werte einen geringen Informationsgehalt, wenig Benutzbarkeitsprobleme und bereits eine Nutzerbasis, die für einen geringen Informationsgehalt sorgt.

Würde diese Standardisierung von den Browserherstellern akzeptiert, wäre ein Großteil der Nutzer von dieser Standardisierung betroffen. Dies würde die Effizienz einer Standardisierung und der in der ersten bis vierten Phase getroffenen Maßnahmen erheblich steigern und das Browserfingerprinting allgemein stark behindern.

Dieses Phasenmodell wird als Handlungsvorschlag gegeben, um den Schutz der Nutzer stufenweise zu verstärken. Da jede der vorgestellten Phasen eine neue Form des Schutzes bietet, muss sie jeweils mit einer anderen Technik umgangen werden, wenn die Nutzer weiterhin mit Browserfingerprinting verfolgt werden sollen. Ist der Schutz einer Phase ausreichend, kann auf die Umsetzung der folgenden Phase verzichtet werden oder diese sogar vorsorglich vorgenommen werden.

Da in den ersten beiden Phasen nur individuelle Nutzer geschützt werden, ist nicht zu erwarten, dass diese Phasen gestört werden, solange die Nutzerbasis der Erweiterung klein ist. Ab der dritten Phase wird zwar auch die Verfolgung von Nutzern behindert, aber gleichzeitig der Einsatz einer solchen Erweiterung getarnt. Der Schaden, der in der vierten Phase an den durch Browserfingerprinting erstellten Datenbanken in Form von Fehleinträgen auftritt, ist auf viele Nutzer verteilt und ist somit ebenfalls getarnt. Deshalb ist zu erwarten, dass diese Maßnahme erst umgangen wird, wenn bereits starker Schaden an den Datenbanken entstanden ist. Die vierte Phase ist für Browserfingerprintingskripte nur wahrnehmbar, wenn die Maßnahmen der ersten bis dritten Phase umgangen werden konnten.

Durch die zu erwartende späte Reaktion und dem daraus folgenden technischen Vorsprung der Gegenmaßnahmen könnte sich die Durchsetzungsfähigkeit von Browserfingerprintingtechniken auf der einen Seite und Gegenmaßnahmen auf der anderen Seite zugunsten der Maßnahmen gegen Browserfingerprinting entwickeln.

8.3. Ansätze für weitere Forschung

Die Gegenmaßnamen gegen Browserfingerprinting waren vor dieser Arbeit kaum erforscht. Obwohl diese Arbeit die Forschung weitergeführt hat, betrachtet sie einige Themenfelder nur kurz und lässt einige Ansätze zur weiteren Forschung offen.

Für viele Abschätzungen wären weitere Einblicke in die konkrete Verteilung der Browserfingerprints wichtig. So war es in dieser Arbeit ein Problem, dass aktuelle Zahlen fehlten und die bisherigen Studien die ermittelten Daten nur auszugsweise oder zusammengefasst veröffentlichten. Eine neue Studie zum Browserfingerprinting, die mit aktuellen Techniken die aktuelle Verteilung der Browser misst und die Ergebnisse veröffentlicht, würde eine solche Betrachtung erleichtern.

Dass es möglich ist, aktives Browserfingerprinting mit CSS und HMTL durchzuführen, konnte zwar mit Beispielimplementierungen gezeigt werden, aber die Entropie dieser Merkmale konnte nicht bestimmt werden. Eine Studie, die diese Techniken bei einer großen Gruppe von Browsern testet und den sich ergebenden Informationsgehalt misst, könnte Klarheit darüber bringen, welche Gefahr diese tatsächlich darstellen.

Gegenmaßnahmen, die nicht in dieser Arbeit oder anderen Arbeiten untersucht wurden, sind weiterhin für die Forschung interessant, da sich die untersuchten Maßnahmen als unzufriedenstellend erwiesen haben. Eine neue bisher unbekannte Maßnahme hätte also das Potenzial, die einzige sicher wirkende Gegenmaßnahme gegen Browserfingerprinting zu sein. Dies gilt auch für die Kombination aus vorhandenen Maßnahmen, die aufgrund der vielen Möglicheiten, sie zu kombinieren, nicht umfassend untersucht werden konnte.

Die in dieser Arbeit gefundenen Mögichkeiten, trotz Randomisierer Fingerprints zu erheben und Nutzer durch diese zu identifizieren, stellen neue Anforderungen an das Randomisieren von Fingerprints. Ob diese erfüllt werden können und ob die gefundenen Möglichkeiten, die Randomisiereung zu umgehen, von der Industrie genutzt werden, könnte in weiterer Forschung untersucht werden.

Die in dieser Arbeit durchgeführte Simulation könnte um verschiedene Faktoren, wie der Änderung von Fingerprints über die Zeit oder einer besseren Nachbildung der Verteilung der Fingerprints, erweitert werden. Dies würde erlauben, detailliertere Simulationen durchzuführen und so unbekannte Effekte zu finden oder auszuschließen.

8.4. Lebenslauf des Autors

Der Verfasser dieser Arbeit, Tim Grocki, wurde 1988 in Essen, Deutschland, geboren und hat 2008 die allgemeine Hochschulreife am Paul-Pfinzing-Gymnasium Hersbruck erworben. Im Jahr 2012 hat er den Abschluss Bachelor of Science im Studienfach Informatik an der Friedrich-Alexander Universität Erlangen-Nürnberg erlangt. Die Bachelorarbeit trägt den Titel „Ablaufplanung von Diagnose-Funktionen beim Start von Automobilen“.

Literaturverzeichnis

[1] URL https://alexa.com. Zuletzt besucht am 17. Februar 2015.

[2] Browserhacks. URL https://browserhacks.com. Zuletzt besucht am 17. Februar 2015.

[3] Conditional-CSS. URL https://conditional-css.com. Zuletzt besucht am 17. Februar 2015.

[4] URL http://www.ghostery.com. Zuletzt besucht am 17. Februar 2015.

[5] URL http://netmarketshare.com. Zuletzt besucht am 17. Februar 2015.

[6] JavaScript/CSS Font Detector. 2007. URL https://www.lalit.org/lab/javascriporganizationt-css-font-detect/. Zuletzt besucht am 15. Februar 2015.

[7] Fingerprinting. 2011. URL http://trac.webkit.org/wiki/Fingerprinting#a4.CSS. Zuletzt besucht am 17. Februar 2015.

[8] Fingerprinting defenses in the Tor Browser. 2014. URL http://www.torproject.org/projects/torbrowser/design/#fingerprinting-linkability. Zuletzt besucht am 21. März 2015.

[9] Private Script Context. 2014. URL https://www.w3.org/community/pua/wiki/Draft.

[10] Erwan Abgrall, Yves Le Traon, Martin Monperrus, Sylvain Gombault, Mario Heiderich, and Alain Ribault. XSS-FP: Browser Fingerprinting using HTML Parser Quirks. *arXiv preprint arXiv:1211.4812*, 2012.

[11] Gunes Acar. Obfuscation for and against device fingerprinting. Position Paper for Symposium on Obfuscation New York University, February 15, 2014.

[12] Gunes Acar, Marc Juarez, Nick Nikiforakis, Claudia Diaz, Seda Gürses, Frank Piessens, and Bart Preneel. FPDetective: Dusting the web for fingerprinters. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 1129–1140. ACM, 2013.

[13] Gunes Acar, Christian Eubank, Steven Englehardt, Marc Juarez, Arvind Narayanan, and Claudia Diaz. The web never forgets: Persistent tracking mechanisms in the wild. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 674–689. ACM, 2014.

[14] Awad E. Ahmed and Issa Traore. Detecting Computer Intrusions Using Behavioral Biometrics. In *PST*, 2005.

[15] Francesco Bergadano, Daniele Gunetti, and Claudia Picardi. User authentication through keystroke dynamics. *ACM Transactions on Information and System Security (TISSEC)*, 5(4):367–397, 2002.

[16] Frédéric Besson, Nataliia Bielova, and Thomas Jensen. Enforcing Browser Anonymity with Quantitative Information Flow. Technical Report RR-8532, INRIA, 2014. URL http://hal.inria.fr/hal-00984654.

[17] Daniel Bilar. Statistical structures: Fingerprinting malware for classification and analysis. *Proceedings of Black Hat Federal 2006*, 2006.

[18] Károly Boda, Ádám Máté Földes, Gábor György Gulyás, and Sándor Imre. User tracking on the web via cross-browser fingerprinting. In *Information Security Technology for Applications*, pages 31–46. Springer, 2012.

[19] Sergey Bratus, Cory Cornelius, David Kotz, and Daniel Peebles. Active behavioral fingerprinting of wireless devices. In *Proceedings of the first ACM conference on Wireless network security*, pages 56–61. ACM, 2008.

[20] Ralph Broenink. Using Browser Properties for Fingerprinting Purposes. In *16th biannual Twente Student Conference on IT*, pages 169–176, 2012.

[21] Timothy G Buchman, Bernard Roizman, Garrett Adams, and Beth Hewitt Stover. Restriction endonuclease fingerprinting of herpes simplex virus DNA: a novel epidemiological tool applied to a nosocomial outbreak. *Journal of Infectious Diseases*, 138(4):488–498, 1978.

[22] Peter Eckersley. How unique is your web browser? In *Privacy Enhancing Technologies*, pages 1–18. Springer, 2010.

[23] Rat Europäisches Parlament. Richtlinie 95/46/eg des europäischen parlaments und des rates vom 24. oktober 1995 zum schutz natürlicher personen bei der verarbeitung personenbezogener daten und zum freien datenverkehr. *Amtsblatt Nr. L*, 281(23):11, 1995.

[24] Amin Faiz Khademi. Browser fingerprinting: Analysis, detection, and prevention at runtime. 2014. Queen's University, Masterarbeit.

[25] Mark Fioravanti. Client fingerprinting via analysis of browser scripting environment. *SANS Information Security Reading Room*, 2010.

[26] Erik Flood and Joel Karlsson. Browser fingerprinting. 2012.

[27] Daniel Genkin, Adi Shamir, and Eran Tromer. RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis. *IACR Cryptology ePrint Archive*, 2013:857, 2013.

[28] Miroslav Goljan, Jessica Fridrich, and Tomáš Filler. Large scale test of sensor fingerprint camera identification. In *IS&T/SPIE Electronic Imaging*, pages 72540I–72540I. International Society for Optics and Photonics, 2009.

[29] Katerina Goseva-Popstojanova, Brandon Miller, Risto Pantev, and Ana Dimitrijevikj. Empirical analysis of attackers activity on multi-tier Web systems. In *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*, pages 781–788. IEEE, 2010.

[30] Peter H Horadan, Matthew R Shanahan, and Mark B Upson. Method and Apparatus for Correlating Multiple Cookies as Having Originated from the Same Device Using Device Fingerprinting, May 24 2011. US Patent App. 13/114,780.

[31] Anil K Jain and Sharath Pankanti. Beyond fingerprinting. *Scientific American*, 299(3):78–81, 2008.

[32] Hong Kaing, Michael Risher, and Brian Schulte. User Tracking: Persistent Cookies and Browser Fingerprinting. 2013. URL `http://cs.gmu.edu/~yhwang1/INFS612/2013_Spring/Projects/Final/2013_Spring_PGN_5_final_report.pdf`.

[33] Samy Kamkar. Evercookie. 2010. URL `http://samy.pl/evercookie`. Zuletzt besucht am 15. Februar 2015.

[34] Amit Klein. How Fraudsters Are Disguising PCs to Fool Device Fingerprinting, 2012. URL `http://securityintelligence.com/how-fraudsters-are-disguising-pcs-to-fool-device-fingerprinting/`. Zuletzt besucht am 17. Februar 2015.

[35] Clemens Kolbitsch, Benjamin Livshits, Benjamin Zorn, and Christian Seifert. Rozzle: De-cloaking internet malware. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 443–457. IEEE, 2012.

[36] Pierre Laperdrix, Walter Rudametkin, and Benoit Baudry. Mitigating browser fingerprint tracking: multi-level reconfiguration and diversification. 2015.

[37] Dustin Lee, Jeff Rowe, Calvin Ko, and Karl Levitt. Detecting and defending against Web-server fingerprinting. In *Computer Security Applications Conference, 2002. Proceedings. 18th Annual*, pages 321–330. IEEE, 2002.

[38] Jonathan R Mayer. Any person... a pamphleteer": Internet Anonymity in the Age of Web 2.0. *Undergraduate Senior Thesis, Princeton University*, 2009.

[39] Jonathan R Mayer and John C Mitchell. Third-party web tracking: Policy and technology. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 413–427. IEEE, 2012.

[40] Katherine McKinley. Cleaning up after cookies. Technical report, Technical report, iSEC PARTNERS, 2010. URL `https://www.isecpartners.com/research/white-papers/cleaning-up-after-cookies.aspx`. Zuletzt besucht am 15. Februar 2015.

[41] Keaton Mowery and Hovav Shacham. Pixel perfect: Fingerprinting canvas in HTML5. *Proceedings of W2SP*, 2012.

[42] Keaton Mowery, Dillon Bogenreif, Scott Yilek, and Hovav Shacham. Fingerprinting information in JavaScript implementations. *Proceedings of W2SP*, 2011.

[43] Martin Mulazzani, Philipp Reschl, Markus Huber, Manuel Leithner, Sebastian Schrittwieser, and Edgar Weippl. Fast and reliable browser identification with javascript engine fingerprinting. In *Web 2.0 Workshop on Security and Privacy (W2SP)*, volume 5. FH Campus Wien, 2013.

[44] Joe Mullin. Silk Road 2.0, infiltrated from the start, sold $8M per month in drugs. URL `http://arstechnica.com/tech-policy/2014/11/silk-road-2-0-infiltrated-from-the-start-sold-8m-per-month-in-drugs/`. Zuletzt besucht am 17. Februar 2015.

[45] D Neuhaus, H Kühl, J-G Kohl, P Dörfel, and T Börner. Investigation on the genetic diversity of Phragmites stands using genomic fingerprinting. *Aquatic Botany*, 45(4):357–364, 1993.

[46] Nick Nikiforakis, Alexandros Kapravelos, Wouter Joosen, Christopher Kruegel, Frank Piessens, and Giovanni Vigna. Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. In *Security and Privacy (SP), 2013 IEEE Symposium on*, pages 541–555. IEEE, 2013.

[47] Nick Nikiforakis, Wouter Joosen, and Benjamin Livshits. PriVaricator: Deceiving Fingerprinters with Little White Lies. 2014. Microsoft Corporation, Tech. Rep.

[48] Athanasios N Papanicolaou, Jimmy F Fox, and John Marshall. Soil fingerprinting in the Palouse Basin, USA, using stable carbon and nitrogen isotopes. *International Journal of Sediment Research*, 18(2):278–284, 2003.

[49] Andreas Pfitzmann and Marit Köhntopp. Anonymity, unobservability, and pseudonymity—a proposal for terminology. In *Designing privacy enhancing technologies*, pages 1–9. Springer, 2001.

[50] Martin Pool. Meantime: non-consensual HTTP user tracking using caches. 2000. URL `https://sourcefrog.net/projects/meantime/`. zuletzt besucht am: 15. Februar 2015.

[51] Matthew Smart, G Robert Malan, and Farnam Jahanian. Defeating TCP/IP Stack Fingerprinting. In *Usenix Security Symposium*, 2000.

[52] Lance Spitzner. Passive fingerprinting. *FOCUS on Intrusion Detection: Passive Fingerprinting (May 3, 2000)*, pages 1–4, 2000.

[53] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[54] Henning Tillmann. Browserfingerprinting: Tracking ohne Spuren zu hinterlassen. 2013. Humbold Universität zu Berlin.

[55] Thomas Unger, Martin Mulazzani, Dominik Fruhwirt, Markus Huber, Sebastian Schrittwieser, and Edgar Weippl. SHPF: Enhancing HTTP (S) Session Security with Browser Fingerprinting. In *Availability, Reliability and Security (ARES), 2013 Eighth International Conference on*, pages 255–261. IEEE, 2013.

[56] Mick Vaites. The effectiveness of a browser fingerprint as a tool for tracking, 2013. The open university.

[57] Gérard Wagener, Alexandre Dulaunoy, and Radu State. Torinj: Automated Exploitation Malware Targeting Tor Users. *arXiv preprint arXiv:1208.2877*, 2012.

[58] Zhihong Xu. Fingerprinting global climate change and forest management within rhizosphere carbon and nutrient cycling processes. *Environmental Science and Pollution Research*, 13(5):293–298, 2006.

[59] Ting-Fang Yen, Yinglian Xie, Fang Yu, Roger Peng Yu, and Martin Abadi. Host Fingerprinting and Tracking on the Web: Privacy and Security Implications. In *NDSS*, 2012.

A

ANHANG

A.1. Detaillierte Tabellen zur Basissimulation

Werte - Basissimulation

Versuch	Entropie	Nutzer1	Nutzer2	Nutzer2-9	Nutzer10+	anon1	anon2	anon3	anon4	anon5	anon6	anon7	anon8	anon9	anon10
1. Versuch	17,6028	18,7589%	31,3917%	81,2346%	0,0065%	10,375	9,875	9,589	9,355	9,182	9,076	9,031	9,004	8,994	8,978
2. Versuch	17,4334	22,1294%	20,9857%	77,288%	0,5827%	14,624	13,834	13,444	13,195	13,031	12,902	12,807	12,679	12,515	12,371
3. Versuch	17,9087	84,5304%	5,2602%	5,546%	9,9237%	1185,111	1148,074	1118,356	1088,583	1061,501	1034,844	1008,652	983,387	958,75	936,758
4. Versuch	18,3386	84,1697%	5,1008%	9,6792%	6,1511%	1164,581	396,892	385,579	378,956	373,718	368,703	364,238	359,329	353,965	347,327

Varationskoeffizienten - Basissimulation

Versuch	Entropie	Nutzer1	Nutzer2	Nutzer2-9	Nutzer10+	anon1	anon2	anon3	anon4	anon5	anon6	anon7	anon8	anon9	anon10
1. Versuch	0,01%	0,26%	0,32%	0,06%	55,69%	6,91%	5,1%	5,28%	5,12%	4,2%	2,92%	1,92%	1,11%	0,99%	1,63%
2. Versuch	0,01%	0,27%	0,39%	0,09%	6,4%	6,47%	4,86%	4,15%	3,51%	2,97%	3,2%	3,42%	3,73%	4,01%	3,92%
3. Versuch	0,03%	0,07%	0,85%	0,83%	0,45%	2,39%	2,0%	1,98%	1,99%	1,98%	1,97%	2,01%	2,01%	1,96%	1,97%
4. Versuch	0,01%	0,07%	0,83%	0,59%	0,76%	2,95%	3,01%	2,37%	2,2%	2,1%	2,07%	2,03%	2,09%	2,25%	2,54%

A.2. Detaillierte Tabellen zu den Variationen der Nutzeranzahl

Werte - Variationen in den Attributen

Nutzer	Entropie	Nutzer1	Nutzer2	Nutzer2-9	Nutzer10+	anon1	anon2	anon3	anon4	anon5	anon6	anon7	anon8	anon9	anon10
47	5,5546	100,0%	0,0%	0,0%	0,0%	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
470	8,8674	99,1277%	0,8085%	0,8723%	0,0%	1,9	1,7	1,3	1,2	1,0	1,0	1,0	1,0	1,0	1,0
4701	12,1462	96,8836%	1,455%	2,8866%	0,2297%	11,6	7,8	6,4	5,6	4,5	4,1	4,0	3,5	3,2	3,2
47016	15,3323	92,3681%	2,5693%	5,4611%	2,1708%	113,5	46,3	42,3	40,0	38,4	37,2	35,8	33,7	31,9	30,4
470161	18,3388	84,1686%	5,1006%	9,6763%	6,1551%	1169,9	394,2	387,4	379,9	375,8	370,5	363,2	359,0	353,7	348,5
4701610	20,9931	67,0589%	9,7487%	20,5448%	12,3963%	11657,9	3769,4	3741,0	3712,9	3685,7	3663,2	3653,2	3631,7	3616,6	3597,0
10%	15,3313	92,3756%	2,5549%	5,3867%	2,2377%	116,5	45,2	42,9	41,1	39,3	37,4	35,6	33,9	33,1	30,9
20%	16,2631	90,5051%	3,0679%	6,4986%	2,9962%	236,2	88,0	81,0	77,8	74,5	72,9	70,1	68,6	67,1	65,2
30%	16,797	89,2113%	3,4312%	6,9318%	3,8569%	349,6	126,5	121,1	117,8	115,5	112,5	108,8	106,6	103,7	100,4
40%	17,171	88,159%	3,7703%	7,5334%	4,3076%	473,3	168,5	160,9	156,3	152,1	149,2	147,6	143,9	139,7	136,0
50%	17,4596	87,327%	4,0237%	8,0211%	4,652%	576,5	203,8	198,7	193,4	189,6	186,3	182,7	179,6	176,4	170,5
60%	17,6946	86,5744%	4,2903%	8,4434%	4,9822%	715,7	242,0	234,5	232,8	224,6	219,8	213,8	210,9	207,6	200,4
70%	17,8888	85,8923%	4,4926%	8,7251%	5,3826%	827,6	282,3	269,9	263,3	258,6	255,3	253,4	249,8	247,4	241,2
80%	18,0573	85,2473%	4,7159%	9,0602%	5,6925%	934,5	328,3	315,4	307,3	303,0	297,7	292,4	287,8	283,3	276,5
90%	18,2052	84,6679%	4,9137%	9,3627%	5,9693%	1061,0	360,8	347,8	342,0	337,0	332,1	326,1	323,6	320,8	316,3
100%	18,3371	84,1385%	5,11%	9,6972%	6,1643%	1182,7	395,1	384,0	378,0	371,7	368,9	365,3	360,6	352,9	348,2
110%	18,4566	83,6685%	5,283%	9,9915%	6,34%	1301,5	436,2	424,6	419,3	413,2	409,0	404,7	398,6	390,9	386,0
120%	18,564	83,2066%	5,4277%	10,3191%	6,4743%	1408,6	469,3	463,6	455,1	451,0	442,8	436,2	432,0	428,0	420,9
130%	18,664	82,7796%	5,5767%	10,6133%	6,6071%	1490,5	511,5	498,0	492,9	485,0	480,1	475,2	469,4	462,5	452,7
140%	18,7544	82,3431%	5,7313%	10,9032%	6,7537%	1612,9	546,9	536,5	530,9	527,4	522,2	515,6	506,9	501,3	492,3
150%	18,8378	81,943%	5,8563%	11,1272%	6,9298%	1744,1	587,3	573,1	563,6	557,4	554,3	549,7	544,1	537,7	528,4
160%	18,9173	81,5637%	5,989%	11,3859%	7,0504%	1880,6	627,8	611,2	602,7	594,4	586,9	582,6	574,6	570,3	554,1
170%	18,9907	81,174%	6,1319%	11,6181%	7,2079%	1971,1	660,9	648,3	638,5	630,0	624,4	617,8	614,8	603,5	590,1
180%	19,0601	80,8492%	6,24%	11,8008%	7,35%	2086,4	698,1	688,6	673,8	663,6	659,3	655,9	647,1	642,0	636,6
190%	19,1254	80,5091%	6,3293%	11,9926%	7,4983%	2240,9	735,9	720,4	710,8	703,5	694,6	691,0	686,4	679,5	670,1
200%	19,1858	80,1775%	6,4463%	12,157%	7,6654%	2358,9	780,4	760,7	753,8	745,7	739,1	734,4	724,5	719,2	709,3
470161-47016	18,2068	84,6926%	4,9358%	9,3555%	5,9519%	1042,5	358,0	346,5	341,7	336,4	332,5	328,3	323,3	316,3	311,5
470161-4701	18,3252	84,2041%	5,0906%	9,6515%	6,1444%	1161,1	392,0	381,0	374,0	369,8	364,0	357,6	354,7	348,1	344,2
470161-470	18,3353	84,1452%	5,0984%	9,6614%	6,1934%	1178,9	394,9	389,4	383,7	375,4	368,0	361,7	356,8	353,2	347,1
470161-47	18,3387	84,1721%	5,0972%	9,6904%	6,1375%	1172,8	393,0	386,5	380,4	376,5	373,1	364,9	361,0	354,6	347,6

Varationskoeffizienten - Variationen in den Attributen

Nutzer	Entropie	Nutzer1	Nutzer2	Nutzer2-9	Nutzer10+	anon1	anon2	anon3	anon4	anon5	anon6	anon7	anon8	anon9	anon10
47	0,0%	0,0%	nan	nan	nan	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
470	0,08%	0,61%	64,25%	69,37%	nan	28,34%	26,96%	35,25%	33,33%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
4701	0,05%	0,37%	19,56%	12,26%	56,75%	16,89%	17,95%	15,93%	21,43%	11,11%	7,32%	0,0%	14,29%	12,5%	12,5%
47016	0,05%	0,2%	2,77%	2,67%	7,53%	12,63%	10,09%	5,99%	6,02%	5,73%	5,99%	6,09%	6,37%	6,02%	5,74%
470161	0,02%	0,07%	0,77%	0,37%	0,56%	2,44%	1,95%	1,6%	1,69%	1,85%	1,94%	2,84%	3,43%	3,64%	3,97%
4701610	0,01%	0,03%	0,15%	0,15%	0,21%	1,01%	0,78%	0,52%	0,73%	0,8%	0,7%	0,64%	0,64%	0,68%	0,35%
10%	0,02%	0,14%	4,57%	3,26%	5,46%	7,38%	5,31%	4,48%	3,68%	4,7%	5,37%	5,92%	7,28%	7,08%	6,38%
20%	0,01%	0,1%	2,68%	1,83%	3,29%	6,7%	5,57%	2,92%	3,76%	2,96%	2,41%	3,74%	3,98%	4,69%	5,21%
30%	0,02%	0,09%	1,65%	1,05%	1,66%	6,45%	1,98%	2,89%	2,65%	2,33%	3,67%	2,32%	2,11%	3,48%	4,35%
40%	0,02%	0,09%	1,38%	0,89%	1,62%	3,46%	4,82%	3,58%	3,6%	2,83%	2,16%	2,21%	2,87%	3,65%	3,73%
50%	0,01%	0,11%	1,48%	1,09%	0,73%	4,81%	3,09%	3,59%	2,95%	1,5%	1,52%	2,41%	2,31%	2,73%	3,2%
60%	0,02%	0,08%	1,0%	0,54%	1,31%	2,57%	4,81%	3,53%	3,81%	2,76%	2,82%	2,13%	2,33%	2,99%	3,58%
70%	0,02%	0,07%	1,08%	0,77%	1,1%	2,56%	2,64%	3,08%	2,73%	1,3%	1,29%	1,45%	1,69%	1,32%	1,67%
80%	0,01%	0,05%	0,59%	0,39%	0,49%	3,78%	2,5%	2,83%	2,38%	1,54%	1,19%	1,47%	2,49%	2,24%	2,4%
90%	0,01%	0,06%	0,92%	0,58%	0,76%	2,72%	2,7%	1,63%	1,94%	2,4%	2,71%	2,7%	2,76%	2,46%	1,84%
100%	0,01%	0,06%	1,1%	0,64%	0,79%	3,07%	2,26%	0,95%	1,43%	2,16%	2,36%	1,8%	2,01%	2,05%	2,56%
110%	0,02%	0,08%	0,55%	0,38%	0,88%	3,08%	3,14%	2,44%	1,89%	1,83%	1,78%	1,62%	2,18%	2,31%	2,18%
120%	0,01%	0,08%	0,55%	0,59%	0,47%	2,14%	1,36%	1,91%	2,33%	1,95%	2,29%	1,95%	2,01%	2,27%	2,13%
130%	0,01%	0,07%	0,83%	0,44%	0,54%	2,32%	3,6%	2,36%	1,96%	1,54%	1,69%	1,23%	1,26%	1,56%	1,86%
140%	0,01%	0,06%	0,72%	0,39%	0,63%	2,15%	2,52%	1,33%	1,41%	1,53%	1,11%	0,66%	1,38%	1,18%	1,52%
150%	0,01%	0,04%	0,59%	0,48%	0,64%	2,04%	1,6%	1,62%	1,71%	1,46%	1,15%	1,32%	1,31%	1,13%	1,68%
160%	0,01%	0,05%	0,58%	0,44%	0,56%	2,79%	2,22%	1,07%	1,75%	2,1%	2,08%	2,26%	2,06%	2,21%	3,63%
170%	0,01%	0,06%	0,6%	0,49%	0,42%	2,29%	1,52%	1,16%	1,35%	1,66%	1,32%	1,11%	1,3%	1,72%	1,94%
180%	0,01%	0,06%	0,78%	0,55%	0,47%	1,17%	1,8%	1,8%	1,79%	1,67%	1,74%	1,39%	1,64%	1,72%	1,6%
190%	0,01%	0,04%	0,49%	0,27%	0,4%	2,62%	2,02%	1,06%	1,17%	1,24%	1,39%	1,29%	1,19%	1,34%	2,22%
200%	0,01%	0,06%	0,55%	0,35%	0,33%	2,3%	2,35%	1,67%	1,62%	1,22%	1,51%	1,73%	1,83%	2,06%	1,15%
470161-47016	0,02%	0,1%	0,83%	0,57%	0,83%	1,61%	2,94%	1,5%	1,89%	1,87%	1,31%	2,04%	1,85%	2,5%	2,57%
470161-4701	0,01%	0,08%	0,82%	0,63%	0,7%	2,66%	2,77%	2,57%	1,73%	1,73%	1,73%	2,08%	1,98%	2,32%	2,27%
470161-470	0,01%	0,06%	0,64%	0,65%	0,45%	2,65%	1,82%	1,89%	1,17%	1,76%	1,77%	2,11%	1,68%	2,06%	2,64%
470161-47	0,01%	0,08%	0,73%	0,72%	0,71%	2,68%	1,59%	2,15%	2,34%	2,2%	2,14%	1,31%	1,45%	1,59%	2,07%

A.3. Detaillierte Tabellen zu den Variationen in den Attributen

Werte - Variationen in den Attributen

Merkmale	Entropie	Nutzer1	Nutzer2	Nutzer2-9	Nutzer10+	anon1	anon2	anon3	anon4	anon5	anon6	anon7	anon8	anon9	anon10
1	2,527	0,0003%	0,0004%	0,0033%	99,9964%	185049,0	107641,8	65410,6	40707,7	25863,5	16413,4	10489,8	6720,7	4271,5	2737,1
2	5,0556	0,0124%	0,012%	0,0852%	99,9025%	77163,0	42215,0	41894,6	24773,6	24605,6	24517,7	15185,1	15065,7	15006,3	14953,8
3	7,5733	0,1511%	0,1246%	0,7909%	99,058%	35145,8	17513,6	17421,5	17312,5	9736,7	9634,1	9607,1	9545,9	9519,5	9462,9
4	10,0637	1,0322%	0,7361%	4,1628%	94,805%	17724,7	7954,9	7879,8	7845,8	7763,8	4034,0	3990,1	3971,7	3938,9	3930,9
5	12,4452	4,6287%	2,8005%	13,261%	82,1103%	9938,8	3976,2	3940,8	3915,0	3887,9	3836,9	1824,1	1800,1	1786,6	1776,8
6	14,57	14,3065%	6,9044%	26,956%	58,7375%	6032,8	2197,7	2177,2	2166,0	2151,7	2127,9	2091,9	920,9	912,8	903,9
7	16,2462	32,0303%	11,257%	34,9707%	32,9989%	3820,0	1350,2	1327,9	1316,0	1305,4	1296,6	1282,8	1266,0	516,8	508,2
8	17,3722	54,077%	12,2812%	29,5596%	16,3635%	2517,8	861,3	850,2	844,9	825,9	820,6	812,7	801,7	781,4	315,5
9	18,0089	72,7688%	9,0684%	17,9703%	9,2609%	1719,9	581,0	565,7	557,8	552,3	543,1	537,0	531,1	524,5	509,8
10	18,3387	84,1754%	5,0915%	9,6829%	6,1417%	1155,0	394,1	390,0	382,5	378,1	372,4	368,8	360,9	353,8	348,4
11	18,512	89,8603%	2,7987%	5,9652%	4,1745%	815,9	270,8	264,7	258,3	255,5	252,9	249,6	245,9	241,7	236,4
12	18,6156	92,7619%	1,8299%	4,3521%	2,886%	548,1	193,9	186,5	180,5	178,2	176,9	174,4	172,2	168,4	165,0
13	18,6847	94,5503%	1,4042%	3,4249%	2,0248%	382,0	135,4	131,5	127,3	124,2	121,5	119,4	118,5	116,4	114,7
14	18,7336	95,8279%	1,2187%	2,9606%	1,2115%	265,1	95,6	90,5	88,5	86,1	84,5	82,6	80,7	79,2	77,4
15	18,7685	96,8337%	1,0633%	2,4419%	0,7244%	182,8	68,1	63,9	61,9	61,0	59,7	58,3	57,1	55,9	55,1
16	18,7932	97,6718%	0,8901%	1,8763%	0,4519%	126,9	47,1	45,3	43,7	42,4	41,1	40,1	39,6	38,8	38,3
17	18,8108	98,3306%	0,7255%	1,439%	0,2304%	84,4	36,2	33,3	31,2	30,5	29,0	27,5	27,1	26,4	25,9
18	18,8222	98,8261%	0,5706%	1,0684%	0,1055%	56,1	25,2	24,2	23,3	21,9	21,4	20,4	19,7	19,2	18,5
19	18,8301	99,215%	0,4268%	0,7288%	0,0562%	41,2	19,2	17,4	16,5	15,7	15,3	14,7	14,0	13,3	12,7
20	18,8352	99,4919%	0,3075%	0,4827%	0,0255%	28,6	14,6	13,5	12,3	11,6	10,9	10,5	10,1	9,6	9,2

Varationskoeffizienten - Variationen in den Attributen

Merkmale	Entropie	Nutzer1	Nutzer2	Nutzer2-9	Nutzer10+	anon1	anon2	anon3	anon4	anon5	anon6	anon7	anon8	anon9	anon10
1	0,05%	65,47%	104,82%	24,18%	0,0%	0,13%	0,2%	0,17%	0,49%	0,4%	0,76%	0,78%	1,15%	1,53%	1,54%
2	0,04%	7,44%	10,95%	7,08%	0,01%	0,29%	0,44%	0,35%	0,48%	0,28%	0,3%	0,52%	0,42%	0,48%	0,44%
3	0,04%	4,2%	5,29%	1,01%	0,01%	0,53%	0,52%	0,34%	0,47%	0,4%	0,37%	0,34%	0,39%	0,55%	0,44%
4	0,04%	1,52%	1,8%	1,48%	0,06%	0,66%	0,67%	0,55%	0,54%	0,92%	1,14%	0,44%	0,47%	0,48%	0,37%
5	0,04%	0,58%	1,4%	0,74%	0,11%	0,64%	1,53%	0,7%	0,77%	1,06%	1,09%	0,99%	0,63%	0,63%	0,68%
6	0,03%	0,46%	0,64%	0,3%	0,18%	1,44%	1,07%	1,07%	0,99%	0,88%	1,22%	0,91%	1,44%	1,38%	1,42%
7	0,03%	0,23%	0,47%	0,23%	0,32%	1,61%	1,97%	1,31%	1,63%	1,99%	1,84%	1,75%	2,0%	2,32%	2,1%
8	0,02%	0,17%	0,41%	0,28%	0,39%	2,5%	1,25%	1,42%	1,53%	1,01%	0,89%	1,35%	1,38%	2,08%	2,92%
9	0,01%	0,11%	0,87%	0,59%	0,74%	2,46%	2,4%	2,19%	1,8%	2,26%	1,51%	2,27%	2,41%	2,12%	3,04%
10	0,02%	0,07%	0,87%	0,61%	0,41%	2,12%	2,54%	2,62%	2,18%	2,92%	2,49%	2,9%	2,41%	2,4%	3,22%
11	0,01%	0,04%	1,28%	0,58%	0,72%	2,92%	3,26%	2,4%	1,36%	1,73%	1,82%	1,47%	1,99%	2,84%	3,69%
12	0,01%	0,04%	1,01%	0,82%	1,21%	2,85%	5,12%	3,51%	2,64%	2,59%	2,39%	2,15%	2,43%	3,41%	2,92%
13	0,0%	0,03%	1,97%	0,88%	0,88%	4,91%	3,42%	2,53%	2,79%	3,2%	3,63%	3,27%	3,03%	2,64%	2,28%
14	0,01%	0,04%	1,77%	0,78%	2,24%	5,36%	6,5%	4,62%	4,27%	4,0%	4,11%	3,84%	3,33%	3,28%	3,33%
15	0,0%	0,03%	2,08%	1,33%	2,61%	7,98%	6,07%	2,75%	3,85%	4,02%	3,82%	3,26%	3,45%	3,53%	3,29%
16	0,0%	0,02%	1,47%	0,95%	3,71%	6,09%	4,6%	3,13%	3,7%	4,25%	3,84%	4,52%	5,2%	4,58%	4,68%
17	0,0%	0,02%	1,1%	1,3%	5,03%	10,89%	7,08%	7,48%	5,88%	5,91%	6,17%	7,5%	7,64%	8,16%	8,71%
18	0,0%	0,02%	2,07%	1,53%	7,34%	12,79%	7,05%	7,11%	4,72%	5,19%	6,34%	7,66%	8,51%	9,55%	10,88%
19	0,0%	0,01%	2,55%	1,6%	14,72%	17,09%	12,5%	8,98%	9,09%	4,97%	7,76%	10,11%	9,58%	8,27%	9,35%
20	0,0%	0,01%	2,75%	2,11%	14,33%	18,58%	10,25%	11,11%	10,94%	5,72%	8,66%	7,68%	8,22%	11,6%	8,13%

A.4. Detaillierte Tabellen zum Schutzparadox

Werte - Schutzparadox

Nutzer	Merkmale	Entropie	Nutzer1	Nutzer2	Nutzer2-9	Nutzer10+	anon1	anon2	anon3	anon4	anon5	anon6	anon7	anon8	anon9	anon10
47	1	2,2506	3,617%	6,383%	36,383%	60,0%	20,4	9,8	7,3	3,6	2,6	1,7	-	-	-	-
47	2	4,1381	30,6383%	17,0213%	62,1277%	7,234%	8,5	5,0	4,0	3,0	2,9	2,5	2,2	2,1	1,8	1,4
47	3	4,961	62,3404%	14,0426%	37,6596%	0,0%	4,5	3,4	2,8	2,4	1,8	1,6	1,5	1,3	1,1	1,0
47	4	5,395	85,5319%	11,9149%	14,4681%	0,0%	2,4	1,9	1,6	1,3	1,2	1,2	1,0	1,0	1,0	1,0
47	5	5,4945	95,1064%	2,9787%	4,8936%	0,0%	2,1	1,2	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
47	6	5,5376	98,2979%	1,7021%	1,7021%	0,0%	1,3	1,1	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
47	7	5,5503	99,5745%	0,4255%	0,4255%	0,0%	1,1	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
47	8	5,5546	100,0%	0,0%	0,0%	0,0%	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
47	9	5,5546	100,0%	0,0%	0,0%	0,0%	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
470	1	2,4769	0,4681%	0,4255%	3,1277%	96,4043%	188,0	108,0	64,6	39,7	25,6	16,2	10,3	7,7	3,9	2,0
470	2	4,8882	6,234%	3,8723%	25,4894%	68,2766%	76,6	47,0	38,7	28,1	23,8	20,3	18,4	15,4	13,9	13,1
470	3	6,7301	23,2979%	12,4255%	46,4681%	30,234%	34,3	19,6	17,7	15,8	13,0	12,2	11,0	10,3	9,3	8,7
470	4	7,9627	53,3617%	17,1489%	39,6809%	6,9574%	17,9	10,4	8,9	7,2	6,8	6,3	5,7	5,0	4,8	4,4
470	5	8,4801	76,4255%	10,8085%	22,1277%	1,4468%	10,6	6,1	5,3	4,6	4,3	4,0	3,6	3,4	3,2	3,1
470	6	8,7314	89,8085%	5,8298%	10,1915%	0,0%	5,6	4,2	3,3	3,1	2,9	2,5	2,1	2,0	2,0	2,0
470	7	8,8168	95,4043%	3,0638%	4,5957%	0,0%	4,0	3,0	2,2	2,0	2,0	2,0	1,9	1,8	1,7	1,3
470	8	8,8455	97,5106%	1,7021%	2,4894%	0,0%	3,1	2,3	2,1	1,9	1,6	1,3	1,2	1,1	1,0	1,0
470	9	8,861	98,6809%	0,9787%	1,3191%	0,0%	2,6	1,8	1,6	1,3	1,1	1,0	1,0	1,0	1,0	1,0
4701	1	2,5271	0,0298%	0,0596%	0,4127%	99,5575%	1849,3	1069,3	659,2	412,0	254,8	161,4	107,5	66,3	42,9	29,8
4701	2	5,0119	0,736%	0,6297%	4,6479%	94,616%	784,0	434,8	408,4	260,9	243,9	236,1	161,8	153,1	143,5	137,3
4701	3	7,3733	5,1776%	3,7907%	19,4384%	75,384%	353,8	187,4	176,5	165,6	110,4	100,3	96,2	91,9	87,7	82,7
4701	4	9,3743	19,4171%	10,2914%	39,6235%	40,9594%	184,1	88,2	81,2	76,4	69,5	48,6	45,5	42,2	40,1	38,3
4701	5	10,7563	43,8843%	14,733%	40,083%	16,0328%	95,9	46,1	42,8	39,2	35,5	33,0	24,3	23,1	22,1	21,1
4701	6	11,5201	68,311%	12,308%	25,4542%	6,2348%	58,9	28,1	25,6	22,3	20,6	18,3	16,6	14,5	13,1	12,3
4701	7	11,8846	84,1502%	6,956%	13,2185%	2,6314%	40,1	17,8	15,0	14,1	12,8	11,6	9,7	8,9	8,4	8,2
4701	8	12,0349	91,1508%	4,0162%	7,6431%	1,2061%	23,3	13,1	10,9	9,6	8,6	7,9	7,3	6,7	6,3	5,8
4701	9	12,1078	94,95%	2,1825%	4,548%	0,502%	15,2	9,6	8,3	7,7	6,5	5,8	5,4	5,0	4,8	4,5
47016	1	2,5266	0,0057%	0,003%	0,034%	99,9602%	18507,9	10764,5	6553,6	4051,2	2579,2	1660,6	1045,5	647,7	439,4	278,2
47016	2	5,0495	0,1015%	0,0961%	0,6823%	99,2162%	7755,5	4238,5	4185,9	2491,4	2453,6	2417,3	1535,6	1510,9	1488,1	1457,7
47016	3	7,5386	0,9512%	0,7521%	4,4834%	94,5655%	3530,4	1774,5	1746,3	1726,1	1002,7	976,9	962,5	949,8	931,9	917,9
47016	4	9,9149	4,9915%	3,1764%	15,8733%	79,1352%	1784,4	798,9	782,0	773,6	749,2	433,7	417,6	409,9	400,6	394,9
47016	5	11,9763	16,541%	8,3193%	32,4275%	51,0316%	988,3	416,2	404,6	393,1	388,5	377,2	199,8	192,2	187,7	183,5
47016	6	13,5302	37,4934%	13,0177%	37,8675%	24,6391%	606,7	237,4	227,0	216,8	211,0	204,9	195,6	105,8	101,1	98,6
47016	7	14,4812	61,0482%	12,6319%	27,7133%	11,2385%	381,2	150,5	140,9	136,3	132,8	128,6	124,1	117,0	60,5	58,6
47016	8	14,9746	78,3748%	8,1683%	15,5919%	6,0333%	260,1	93,5	89,5	84,8	80,8	79,3	76,3	74,3	71,3	39,5
47016	9	15,2156	87,8967%	4,3811%	8,7357%	3,3676%	169,5	63,9	58,7	55,7	53,5	52,0	50,4	48,4	44,9	42,7

Varationskoeffizienten - Schutzparadox

Nutzer	Merkmale	Entropie	Nutzer1	Nutzer2	Nutzer2-9	Nutzer10+	anon1	anon2	anon3	anon4	anon5	anon6	anon7	anon8	anon9	anon10
47	1	8,75%	52,94%	80,28%	38,48%	22,19%	18,76%	20,81%	23,77%	30,93%	25,51%	26,96%	nan	nan	nan	nan
47	2	4,58%	19,44%	40,31%	20,54%	154,07%	26,96%	15,49%	19,36%	21,08%	24,14%	26,83%	18,18%	14,29%	22,22%	34,99%
47	3	2,1%	11,43%	60,68%	18,92%	nan%	14,91%	19,51%	21,43%	20,41%	22,22%	30,62%	33,33%	35,25%	27,27%	0,0%
47	4	1,44%	8,24%	57,14%	48,69%	nan%	20,41%	15,79%	30,62%	35,25%	33,33%	33,33%	0,0%	0,0%	0,0%	0,0%
47	5	0,8%	3,33%	91,47%	64,64%	nan%	33,33%	33,33%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
47	6	0,51%	2,87%	165,83%	165,83%	nan%	35,25%	27,27%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
47	7	0,23%	1,28%	300,0%	300,0%	nan%	27,27%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
47	8	0,0%	0,0%	nan%	nan%	nan%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
47	9	0,0%	0,0%	nan%	nan%	nan%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
470	1	1,6%	53,01%	100,0%	53,14%	1,64%	5,7%	11,05%	8,68%	15,45%	6,81%	25,42%	17,39%	29,07%	21,3%	38,73%
470	2	1,76%	14,16%	21,67%	8,5%	3,59%	9,22%	13,66%	11,09%	13,55%	13,91%	9,62%	11,19%	14,29%	10,89%	12,05%
470	3	1,36%	4,5%	16,27%	7,09%	10,23%	17,97%	12,33%	10,12%	13,22%	9,1%	6,13%	7,04%	7,58%	11,83%	11,55%
470	4	0,91%	4,14%	12,31%	6,49%	29,96%	18,44%	13,73%	17,73%	17,35%	17,15%	14,29%	13,7%	8,94%	12,5%	11,13%
470	5	0,65%	2,44%	13,32%	9,96%	102,69%	33,01%	18,62%	20,75%	19,92%	14,89%	19,36%	18,43%	14,41%	12,5%	9,68%
470	6	0,28%	1,42%	16,34%	12,47%	nan%	26,73%	17,82%	13,89%	9,68%	18,57%	20,0%	14,29%	0,0%	0,0%	0,0%
470	7	0,19%	1,07%	25,46%	22,22%	nan%	31,62%	14,91%	18,18%	0,0%	0,0%	0,0%	15,79%	22,22%	26,96%	35,25%
470	8	0,14%	0,79%	35,36%	31,06%	nan%	30,43%	19,92%	14,29%	15,79%	30,62%	35,25%	33,33%	27,27%	0,0%	0,0%
470	9	0,06%	0,51%	61,64%	38,03%	nan%	25,51%	22,22%	30,62%	35,25%	27,27%	0,0%	0,0%	0,0%	0,0%	0,0%
4701	1	0,82%	65,47%	47,38%	20,77%	0,08%	2,32%	2,44%	3,13%	2,7%	4,06%	5,51%	11,49%	14,26%	13,61%	12,08%
4701	2	0,56%	9,94%	24,7%	8,89%	0,46%	3,25%	2,69%	4,95%	3,02%	5,42%	5,56%	5,7%	5,58%	3,01%	3,94%
4701	3	0,46%	3,77%	6,2%	5,15%	1,26%	4,75%	5,05%	4,53%	4,78%	6,48%	5,93%	5,42%	6,16%	5,77%	6,65%
4701	4	0,36%	2,54%	6,68%	1,91%	2,06%	3,79%	7,98%	6,35%	7,15%	6,63%	7,76%	6,08%	5,49%	4,09%	3,88%
4701	5	0,27%	1,95%	4,09%	2,55%	4,32%	10,14%	8,43%	6,92%	10,0%	12,74%	11,89%	6,65%	7,85%	5,88%	8,6%
4701	6	0,15%	1,34%	5,68%	4,21%	10,18%	11,87%	14,63%	9,76%	11,7%	9,76%	12,23%	12,4%	11,64%	10,49%	8,17%
4701	7	0,11%	0,66%	8,19%	4,01%	12,26%	12,39%	12,51%	11,55%	11,19%	10,94%	11,69%	9,28%	12,76%	12,14%	10,63%
4701	8	0,09%	0,51%	7,89%	5,49%	21,96%	21,46%	16,18%	10,42%	13,34%	17,4%	13,22%	15,07%	15,0%	14,29%	15,03%
4701	9	0,05%	0,26%	10,43%	6,92%	35,41%	18,79%	16,27%	14,31%	10,14%	14,18%	15,03%	14,81%	15,49%	12,5%	11,11%
47016	1	0,2%	43,98%	128,57%	27,95%	0,01%	0,45%	0,94%	1,12%	1,1%	1,97%	2,54%	3,63%	2,14%	3,26%	4,03%
47016	2	0,24%	13,72%	20,77%	5,93%	0,04%	1,28%	1,24%	1,21%	0,71%	0,96%	1,5%	1,7%	1,09%	1,11%	1,38%
47016	3	0,14%	4,42%	6,84%	1,87%	0,12%	1,7%	1,11%	1,28%	1,03%	1,09%	0,93%	1,38%	1,28%	1,26%	1,35%
47016	4	0,17%	1,72%	2,98%	1,99%	0,38%	2,27%	1,63%	0,9%	1,51%	2,17%	1,88%	1,49%	1,9%	1,84%	1,82%
47016	5	0,09%	0,86%	1,87%	1,18%	0,74%	3,13%	2,95%	2,46%	1,86%	1,62%	3,84%	2,78%	2,43%	1,56%	1,66%
47016	6	0,07%	0,43%	1,21%	0,69%	0,97%	3,05%	2,92%	2,91%	3,62%	3,82%	3,53%	4,97%	4,94%	3,2%	3,12%
47016	7	0,06%	0,43%	1,76%	1,15%	1,72%	3,08%	5,68%	4,2%	3,15%	2,52%	3,84%	5,56%	4,51%	5,79%	5,77%
47016	8	0,04%	0,21%	2,06%	1,31%	2,94%	8,39%	7,85%	8,11%	7,44%	5,24%	4,95%	3,57%	3,07%	5,43%	6,02%
47016	9	0,03%	0,21%	3,62%	2,28%	3,07%	9,39%	9,38%	4,57%	3,5%	4,44%	4,21%	4,09%	5,71%	6,41%	10,48%

A.5. Detaillierte Tabellen zur uneingeschränkten Fälschung von Fingerprints

Werte - uneingeschränkte Fälschung von Fingerprints

P_{fake}	P_{visit}		1.Tick	2.Tick	3.Tick	4.Tick	5.Tick
		Entropie:	18,3376	0,0	0,0	0,0	0,0
		Nutzer1:	84,1687%	0,0%	0,0%	0,0%	0,0%
		Nutzer2:	5,0758%	0,0%	0,0%	0,0%	0,0%
		Nutzer2-9:	9,648%	0,0%	0,0%	0,0%	0,0%
		Nutzer10+:	6,1833%	100,0%	100,0%	100,0%	100,0%
		anon1:	1161,2	470161,0	470161,0	470161,0	470161,0
		anon2:	398,7	nan	nan	nan	nan
100%	100%	anon3:	386,0	nan	nan	nan	nan
		anon4:	380,4	nan	nan	nan	nan
		anon5:	373,6	nan	nan	nan	nan
		anon6:	368,0	nan	nan	nan	nan
		anon7:	362,9	nan	nan	nan	nan
		anon8:	357,8	nan	nan	nan	nan
		anon9:	351,4	nan	nan	nan	nan
		anon10:	346,6	nan	nan	nan	nan
		Entropie:	18,3383	0,0	0,0	0,0	0,0
		Nutzer1:	84,1619%	0,0%	0,0%	0,0%	0,0%
		Nutzer2:	5,0971%	0,0%	0,0%	0,0%	0,0%
		Nutzer2-9:	9,6701%	0,0%	0,0%	0,0%	0,0%
		Nutzer10+:	6,168%	100,0%	100,0%	100,0%	100,0%
		anon1:	1163,1	470161,0	470161,0	470161,0	470161,0
		anon2:	394,4	nan	nan	nan	nan
100%	1%	anon3:	384,9	nan	nan	nan	nan
		anon4:	377,8	nan	nan	nan	nan
		anon5:	374,0	nan	nan	nan	nan
		anon6:	369,3	nan	nan	nan	nan
		anon7:	361,2	nan	nan	nan	nan
		anon8:	354,3	nan	nan	nan	nan
		anon9:	350,1	nan	nan	nan	nan
		anon10:	342,1	nan	nan	nan	nan
		Entropie:	18,3375	0,0	0,0	0,0	0,0
		Nutzer1:	84,1374%	0,0%	0,0%	0,0%	0,0%
		Nutzer2:	5,1153%	0,0%	0,0%	0,0%	0,0%
		Nutzer2-9:	9,7095%	0,0%	0,0%	0,0%	0,0%
		Nutzer10+:	6,1531%	100,0%	100,0%	100,0%	100,0%
		anon1:	1152,2	470161,0	470161,0	470161,0	470161,0
		anon2:	401,1	nan	nan	nan	nan
100%	0,01%	anon3:	385,7	nan	nan	nan	nan
		anon4:	376,8	nan	nan	nan	nan
		anon5:	373,1	nan	nan	nan	nan
		anon6:	368,2	nan	nan	nan	nan
		anon7:	363,3	nan	nan	nan	nan
		anon8:	357,5	nan	nan	nan	nan
		anon9:	352,4	nan	nan	nan	nan
		anon10:	348,7	nan	nan	nan	nan
		Entropie:	18,3378	18,2128	18,2128	18,2128	18,2128
		Nutzer1:	84,1554%	83,3571%	83,3571%	83,3571%	83,3571%
		Nutzer2:	5,0996%	5,0297%	5,0297%	5,0297%	5,0297%
		Nutzer2-9:	9,6705%	9,5376%	9,5376%	9,5376%	9,5376%
		Nutzer10+:	6,1742%	7,1052%	7,1052%	7,1052%	7,1052%
		anon1:	1174,0	5892,2	5892,2	5892,2	5892,2
		anon2:	390,4	386,5	386,5	386,5	386,5
1%	100%	anon3:	382,6	380,4	380,4	380,4	380,4
		anon4:	378,0	374,7	374,7	374,7	374,7
		anon5:	372,6	370,2	370,2	370,2	370,2
		anon6:	367,6	364,3	364,3	364,3	364,3
		anon7:	363,5	359,0	359,0	359,0	359,0
		anon8:	357,7	354,4	354,4	354,4	354,4
		anon9:	351,3	347,1	347,1	347,1	347,1
		anon10:	345,6	342,2	342,2	342,2	342,2
		Entropie:	18,3392	18,2167	18,2167	18,2167	18,2167
		Nutzer1:	84,1625%	83,3766%	83,3766%	83,3766%	83,3766%
		Nutzer2:	5,0982%	5,03%	5,03%	5,03%	5,03%
		Nutzer2-9:	9,6932%	9,5583%	9,5583%	9,5583%	9,5583%
		Nutzer10+:	6,1444%	7,0651%	7,0651%	7,0651%	7,0651%
		anon1:	1155,2	5681,4	5681,4	5681,4	5681,4
		anon2:	395,0	539,2	539,2	539,2	539,2
1%	1%	anon3:	382,0	379,5	379,5	379,5	379,5
		anon4:	374,9	371,6	371,6	371,6	371,6
		anon5:	368,0	365,4	365,4	365,4	365,4
		anon6:	363,3	360,0	360,0	360,0	360,0
		anon7:	361,1	357,9	357,9	357,9	357,9
		anon8:	354,7	351,7	351,7	351,7	351,7
		anon9:	347,8	344,5	344,5	344,5	344,5
		anon10:	341,4	338,0	338,0	338,0	338,0
		Entropie:	18,3392	18,2242	18,2238	18,2229	18,2242
		Nutzer1:	84,1658%	83,3747%	83,3748%	83,3748%	83,3747%
		Nutzer2:	5,1002%	5,0313%	5,0313%	5,0313%	5,0313%
		Nutzer2-9:	9,6788%	9,5493%	9,5493%	9,5493%	9,5492%
		Nutzer10+:	6,1554%	7,076%	7,0759%	7,0759%	7,0761%
		anon1:	1148,0	4702,9	4739,8	4855,2	4703,3
		anon2:	399,1	1136,3	1136,3	1058,4	1136,3
1%	0,01%	anon3:	387,9	395,1	395,1	394,4	395,1
		anon4:	382,9	383,9	383,9	383,5	383,9
		anon5:	374,3	378,7	378,7	377,8	378,7
		anon6:	368,6	370,2	370,2	370,0	370,2
		anon7:	363,8	365,1	365,1	364,4	365,1
		anon8:	354,0	359,5	359,5	357,4	359,5
		anon9:	349,1	350,1	350,1	349,8	350,1
		anon10:	342,1	346,2	346,2	344,9	346,2

p_{fake}	p_{visit}		1.Tick	2.Tick	3.Tick	4.Tick	5.Tick
		Entropie:	18,3392	18,3381	18,3381	18,3381	18,3381
		Nutzer1:	84,1722%	84,1639%	84,1639%	84,1639%	84,1639%
		Nutzer2:	5,0976%	5,0973%	5,0973%	5,0973%	5,0973%
		Nutzer2-9:	9,6896%	9,6883%	9,6883%	9,6883%	9,6883%
		Nutzer10+:	6,1381%	6,1477%	6,1477%	6,1477%	6,1477%
		anon1:	1160,1	1209,7	1209,7	1209,7	1209,7
		anon2:	394,8	394,7	394,7	394,7	394,7
0,01%	100%	anon3:	382,7	382,6	382,6	382,6	382,6
		anon4:	378,0	377,8	377,8	377,8	377,8
		anon5:	369,2	369,2	369,2	369,2	369,2
		anon6:	366,1	366,1	366,1	366,1	366,1
		anon7:	360,1	360,1	360,1	360,1	360,1
		anon8:	355,9	355,9	355,9	355,9	355,9
		anon9:	351,7	351,6	351,6	351,6	351,6
		anon10:	345,9	345,9	345,9	345,9	345,9
		Entropie:	18,3371	18,3359	18,3359	18,3359	18,3359
		Nutzer1:	84,1441%	84,1353%	84,1353%	84,1353%	84,1353%
		Nutzer2:	5,1114%	5,1108%	5,1108%	5,1108%	5,1108%
		Nutzer2-9:	9,6818%	9,6806%	9,6806%	9,6806%	9,6806%
		Nutzer10+:	6,1741%	6,1841%	6,1841%	6,1841%	6,1841%
		anon1:	1164,1	1214,4	1214,4	1214,4	1205,4
		anon2:	392,9	392,9	392,9	392,9	399,3
0,01%	1%	anon3:	383,8	383,8	383,8	383,8	385,8
		anon4:	376,6	376,6	376,6	376,6	377,2
		anon5:	373,1	373,0	373,0	373,0	373,0
		anon6:	370,2	370,2	370,2	370,2	370,2
		anon7:	367,7	367,7	367,7	367,7	367,7
		anon8:	358,5	358,4	358,4	358,4	358,4
		anon9:	353,6	353,6	353,6	353,6	353,6
		anon10:	347,3	347,3	347,3	347,3	347,3
		Entropie:	18,3391	18,3387	18,3387	18,3386	18,3387
		Nutzer1:	84,1653%	84,1583%	84,1583%	84,1584%	84,1583%
		Nutzer2:	5,1142%	5,1134%	5,1134%	5,1134%	5,1134%
		Nutzer2-9:	9,686%	9,6844%	9,6844%	9,6843%	9,6844%
		Nutzer10+:	6,1487%	6,1573%	6,1573%	6,1573%	6,1573%
		anon1:	1145,1	1144,9	1144,9	1148,9	1144,9
		anon2:	395,4	395,4	395,4	395,4	395,4
0,01%	0,01%	anon3:	385,1	385,1	385,1	385,1	385,1
		anon4:	378,0	377,9	377,9	377,9	377,9
		anon5:	373,7	373,6	373,6	373,6	373,6
		anon6:	367,6	367,4	367,4	367,4	367,4
		anon7:	365,2	365,2	365,2	365,2	365,2
		anon8:	360,1	360,1	360,1	360,1	360,1
		anon9:	354,9	354,9	354,9	354,9	354,9
		anon10:	348,6	348,6	348,6	348,6	348,6

Varationskoeffizienten - uneingeschränkte Fälschung von Fingerprints

p_{fake}	p_{visit}		1.Tick	2.Tick	3.Tick	4.Tick	5.Tick
		Entropie:	0,011%	-	-	-	-
		Nutzer1:	0,0499%	-	-	-	-
		Nutzer2:	0,7978%	-	-	-	-
		Nutzer2-9:	0,5667%	-	-	-	-
		Nutzer10+:	0,6142%	0,0%	0,0%	0,0%	0,0%
		anon1:	1,6389%	0,0%	0,0%	0,0%	0,0%
		anon2:	3,6261%	-	-	-	-
100%	100%	anon3:	1,4609%	-	-	-	-
		anon4:	1,5202%	-	-	-	-
		anon5:	1,168%	-	-	-	-
		anon6:	1,1335%	-	-	-	-
		anon7:	1,1122%	-	-	-	-
		anon8:	1,413%	-	-	-	-
		anon9:	1,6797%	-	-	-	-
		anon10:	2,3161%	-	-	-	-
		Entropie:	0,0091%	-	-	-	-
		Nutzer1:	0,0824%	-	-	-	-
		Nutzer2:	1,0017%	-	-	-	-
		Nutzer2-9:	0,6639%	-	-	-	-
		Nutzer10+:	0,6481%	0,0%	0,0%	0,0%	0,0%
		anon1:	2,5182%	0,0%	0,0%	0,0%	0,0%
		anon2:	1,7129%	-	-	-	-
100%	1%	anon3:	1,5823%	-	-	-	-
		anon4:	1,7905%	-	-	-	-
		anon5:	1,8982%	-	-	-	-
		anon6:	1,503%	-	-	-	-
		anon7:	1,416%	-	-	-	-
		anon8:	2,0858%	-	-	-	-
		anon9:	2,4553%	-	-	-	-
		anon10:	3,4499%	-	-	-	-
		Entropie:	0,0134%	-	-	-	-
		Nutzer1:	0,0847%	-	-	-	-
		Nutzer2:	0,5807%	-	-	-	-
		Nutzer2-9:	0,5633%	-	-	-	-
		Nutzer10+:	0,7242%	0,0%	0,0%	0,0%	0,0%
		anon1:	1,776%	0,0%	0,0%	0,0%	0,0%
		anon2:	3,1248%	-	-	-	-
100%	0,01%	anon3:	2,5798%	-	-	-	-
		anon4:	1,9924%	-	-	-	-
		anon5:	1,6191%	-	-	-	-
		anon6:	1,6241%	-	-	-	-
		anon7:	1,1616%	-	-	-	-
		anon8:	1,3603%	-	-	-	-
		anon9:	1,3855%	-	-	-	-
		anon10:	1,5339%	-	-	-	-
		Entropie:	0,0105%	0,0126%	0,0126%	0,0126%	0,0126%
		Nutzer1:	0,0754%	0,0791%	0,0791%	0,0791%	0,0791%
		Nutzer2:	1,0069%	1,0489%	1,0489%	1,0489%	1,0489%
		Nutzer2-9:	0,743%	0,7482%	0,7482%	0,7482%	0,7482%
		Nutzer10+:	0,6746%	0,5971%	0,5971%	0,5971%	0,5971%
		anon1:	2,6315%	1,0336%	1,0336%	1,0336%	1,0336%
		anon2:	2,3342%	2,6265%	2,6265%	2,6265%	2,6265%
1%	100%	anon3:	1,6744%	1,8744%	1,8744%	1,8744%	1,8744%
		anon4:	2,6162%	2,6955%	2,6955%	2,6955%	2,6955%
		anon5:	2,4808%	2,6044%	2,6044%	2,6044%	2,6044%
		anon6:	2,1114%	2,4151%	2,4151%	2,4151%	2,4151%
		anon7:	2,256%	2,1504%	2,1504%	2,1504%	2,1504%
		anon8:	2,5131%	2,5713%	2,5713%	2,5713%	2,5713%
		anon9:	1,8274%	1,7115%	1,7115%	1,7115%	1,7115%
		anon10:	1,9159%	1,8929%	1,8929%	1,8929%	1,8929%
		Entropie:	0,0131%	0,0158%	0,0158%	0,0158%	0,0158%
		Nutzer1:	0,0703%	0,0738%	0,0738%	0,0738%	0,0738%
		Nutzer2:	0,89%	0,9046%	0,9046%	0,9046%	0,9046%
		Nutzer2-9:	0,573%	0,5847%	0,5847%	0,5847%	0,5847%
		Nutzer10+:	0,6284%	0,5617%	0,5617%	0,5617%	0,5617%
		anon1:	3,0339%	5,3936%	5,3936%	5,3936%	5,3936%
		anon2:	2,5062%	54,5743%	54,5743%	54,5743%	54,5743%
1%	1%	anon3:	1,9237%	1,9763%	1,9763%	1,9763%	1,9763%
		anon4:	1,8109%	1,798%	1,798%	1,798%	1,798%
		anon5:	1,2861%	1,3858%	1,3858%	1,3858%	1,3858%
		anon6:	1,767%	1,4907%	1,4907%	1,4907%	1,4907%
		anon7:	1,7091%	1,3714%	1,3714%	1,3714%	1,3714%
		anon8:	1,4974%	1,4557%	1,4557%	1,4557%	1,4557%
		anon9:	3,2803%	3,3281%	3,3281%	3,3281%	3,3281%
		anon10:	3,3041%	3,1617%	3,1617%	3,1617%	3,1617%
		Entropie:	0,0118%	0,0168%	0,0152%	0,0251%	0,0168%
		Nutzer1:	0,0276%	0,036%	0,036%	0,036%	0,0359%
		Nutzer2:	0,8952%	0,9143%	0,9143%	0,9143%	0,9143%
		Nutzer2-9:	0,3069%	0,3396%	0,3396%	0,3396%	0,3406%
		Nutzer10+:	0,7164%	0,6452%	0,6456%	0,645%	0,6464%
		anon1:	3,0445%	1,6183%	1,7792%	7,6343%	1,6257%
		anon2:	2,923%	3,0504%	3,0504%	21,2462%	3,0504%
1%	0,01%	anon3:	1,62%	3,2024%	3,2024%	3,3082%	3,2024%
		anon4:	1,7341%	1,5735%	1,5735%	1,6042%	1,5735%
		anon5:	1,7234%	1,822%	1,822%	1,919%	1,822%
		anon6:	1,2265%	1,6017%	1,6017%	1,6035%	1,6017%
		anon7:	1,0344%	1,3883%	1,3883%	1,3622%	1,3883%
		anon8:	1,5212%	1,135%	1,135%	1,8944%	1,135%
		anon9:	1,8317%	1,4645%	1,4645%	1,6313%	1,4645%
		anon10:	2,7529%	1,3961%	1,3961%	2,2438%	1,3961%

p_{fake}	p_{visit}		1.Tick	2.Tick	3.Tick	4.Tick	5.Tick
		Entropie:	0,011%	0,0107%	0,0107%	0,0107%	0,0107%
		Nutzer1:	0,0799%	0,0796%	0,0796%	0,0796%	0,0796%
		Nutzer2:	0,8236%	0,8221%	0,8221%	0,8221%	0,8221%
		Nutzer2-9:	0,7444%	0,7405%	0,7405%	0,7405%	0,7405%
		Nutzer10+:	0,5235%	0,5112%	0,5112%	0,5112%	0,5112%
		anon1:	1,4008%	1,4704%	1,4704%	1,4704%	1,4704%
		anon2:	1,3005%	1,3069%	1,3069%	1,3069%	1,3069%
0,01%	100%	anon3:	2,2358%	2,246%	2,246%	2,246%	2,246%
		anon4:	1,6351%	1,6521%	1,6521%	1,6521%	1,6521%
		anon5:	1,725%	1,725%	1,725%	1,725%	1,725%
		anon6:	1,7679%	1,7679%	1,7679%	1,7679%	1,7679%
		anon7:	1,8016%	1,8016%	1,8016%	1,8016%	1,8016%
		anon8:	1,6926%	1,6926%	1,6926%	1,6926%	1,6926%
		anon9:	2,0584%	2,0985%	2,0985%	2,0985%	2,0985%
		anon10:	1,471%	1,471%	1,471%	1,471%	1,471%
		Entropie:	0,013%	0,0134%	0,0134%	0,0134%	0,0134%
		Nutzer1:	0,0666%	0,0672%	0,0672%	0,0672%	0,0672%
		Nutzer2:	0,7771%	0,7778%	0,7778%	0,7778%	0,7778%
		Nutzer2-9:	0,6514%	0,6456%	0,6456%	0,6456%	0,6456%
		Nutzer10+:	0,7044%	0,7031%	0,7031%	0,7031%	0,7031%
		anon1:	2,3916%	2,2234%	2,2234%	2,2234%	2,1925%
		anon2:	3,3447%	3,3447%	3,3447%	3,3447%	3,836%
0,01%	1%	anon3:	3,0825%	3,0825%	3,0825%	3,0825%	2,9753%
		anon4:	2,9214%	2,9214%	2,9214%	2,9214%	2,7315%
		anon5:	2,8351%	2,8575%	2,8575%	2,8575%	2,8575%
		anon6:	2,5392%	2,5392%	2,5392%	2,5392%	2,5392%
		anon7:	2,4868%	2,4868%	2,4868%	2,4868%	2,4868%
		anon8:	2,3413%	2,3015%	2,3015%	2,3015%	2,3015%
		anon9:	2,0981%	2,0981%	2,0981%	2,0981%	2,0981%
		anon10:	1,9573%	1,9573%	1,9573%	1,9573%	1,9573%
		Entropie:	0,0189%	0,019%	0,019%	0,019%	0,019%
		Nutzer1:	0,0834%	0,0837%	0,0838%	0,0838%	0,0837%
		Nutzer2:	0,7958%	0,7883%	0,7893%	0,7906%	0,7901%
		Nutzer2-9:	0,5067%	0,5064%	0,5066%	0,5069%	0,5067%
		Nutzer10+:	0,8273%	0,8301%	0,8302%	0,828%	0,8307%
		anon1:	3,5004%	3,5093%	3,5093%	4,0891%	3,5093%
		anon2:	2,9822%	2,9822%	2,9822%	2,9822%	2,9822%
0,01%	0,01%	anon3:	1,9343%	1,9343%	1,9343%	1,9343%	1,9343%
		anon4:	1,473%	1,4943%	1,4943%	1,4943%	1,4943%
		anon5:	1,8772%	1,916%	1,916%	1,916%	1,916%
		anon6:	1,7299%	1,7732%	1,7732%	1,7732%	1,7732%
		anon7:	1,7003%	1,7003%	1,7003%	1,7003%	1,7003%
		anon8:	2,0348%	2,0348%	2,0348%	2,0348%	2,0348%
		anon9:	2,359%	2,359%	2,359%	2,359%	2,359%
		anon10:	2,6014%	2,6014%	2,6014%	2,6014%	2,6014%

A.6. Detaillierte Tabellen zur eingeschränkten Fälschung von Fingerprints

Werte - eingeschränkte Fälschung von Fingerprints

p_{fake}	p_{visit}	p_{fixed}		1.Tick	2.Tick	10.Tick	20.Tick	30.Tick	50.Tick	70.Tick	90.Tick	100.Tick
			Entropie:	18,3397	18,3393	18,3385	18,338	18,3378	18,3374	18,3372	18,337	18,337
			Nutzer1:	84,1589%	84,1411%	84,0823%	84,0447%	84,0189%	83,9856%	83,9631%	83,9456%	83,9389%
			Nutzer2:	5,1153%	5,1269%	5,1712%	5,1995%	5,222%	5,2494%	5,2711%	5,2904%	5,2957%
			Nutzer2-9:	9,7113%	9,7297%	9,786%	9,8264%	9,851%	9,8837%	9,9058%	9,9235%	9,9284%
			Nutzer10+:	6,1298%	6,1292%	6,1316%	6,1289%	6,1301%	6,1306%	6,1311%	6,1309%	6,1327%
			anon1:	1158,1	1164,7	1163,0	1165,0	1165,3	1164,6	1164,0	1165,4	1165,4
			anon2:	388,4	388,7	389,9	389,6	389,2	389,6	389,1	388,7	388,9
1%	1%	90%	anon3:	377,6	379,1	379,4	378,9	379,0	379,7	379,1	379,1	378,6
			anon4:	373,6	374,9	374,1	374,8	374,7	375,3	375,6	374,9	374,5
			anon5:	369,0	369,8	370,3	369,3	369,4	369,8	370,1	369,8	369,5
			anon6:	364,1	365,2	365,2	364,6	365,3	365,2	365,2	364,4	364,8
			anon7:	360,5	361,8	361,7	361,4	362,0	361,1	361,6	361,4	361,9
			anon8:	356,5	357,5	357,0	357,6	357,0	357,2	357,2	356,7	357,5
			anon9:	353,1	353,5	353,6	353,7	353,7	353,9	353,4	353,3	353,6
			anon10:	348,3	348,1	348,6	348,3	347,7	348,2	348,0	347,7	348,6
			Entropie:	18,3367	18,2516	18,2514	18,2511	18,2509	18,2511	18,2507	18,251	18,2508
			Nutzer1:	84,1235%	83,2256%	83,2132%	83,2138%	83,2132%	83,2109%	83,2138%	83,2124%	83,2128%
			Nutzer2:	5,1076%	5,2134%	5,2213%	5,2204%	5,2241%	5,2246%	5,2186%	5,2228%	5,221%
			Nutzer2-9:	9,7013%	9,852%	9,8653%	9,8601%	9,8591%	9,8663%	9,8595%	9,861%	9,8591%
			Nutzer10+:	6,1752%	6,9224%	6,9215%	6,9261%	6,9277%	6,9228%	6,9267%	6,9266%	6,9281%
			anon1:	1178,8	3517,0	3582,0	3582,0	3582,0	3667,6	3667,6	3667,6	3667,6
			anon2:	394,0	602,6	535,0	525,7	533,8	451,4	454,4	446,4	453,3
1%	1%	10%	anon3:	385,9	438,8	434,5	444,4	443,9	435,8	442,2	438,5	440,0
			anon4:	379,3	426,7	421,8	432,8	425,7	419,9	432,9	426,4	431,6
			anon5:	374,5	422,5	416,5	425,0	415,6	414,2	425,1	416,6	421,2
			anon6:	372,0	410,5	407,9	415,7	408,1	403,9	416,7	406,1	411,8
			anon7:	365,4	399,2	399,1	407,0	400,9	393,6	398,8	394,9	399,8
			anon8:	360,5	380,0	388,5	389,8	390,7	385,0	380,2	384,5	389,6
			anon9:	355,7	367,3	375,0	371,8	376,2	373,2	368,9	370,4	375,7
			anon10:	346,7	355,3	364,1	362,0	362,8	356,5	357,9	354,4	361,1
			Entropie:	18,3382	18,3382	18,3381	18,3381	18,3381	18,338	18,338	18,3379	18,3379
			Nutzer1:	84,1559%	84,1557%	84,1541%	84,1514%	84,1498%	84,1464%	84,1438%	84,141%	84,1397%
			Nutzer2:	5,1089%	5,109%	5,1098%	5,1114%	5,1127%	5,1148%	5,1163%	5,1184%	5,1199%
			Nutzer2-9:	9,6862%	9,6867%	9,6883%	9,6894%	9,6912%	9,6941%	9,6968%	9,6985%	9,7007%
			Nutzer10+:	6,1579%	6,1577%	6,1576%	6,1592%	6,159%	6,1595%	6,1595%	6,1604%	6,1596%
			anon1:	1181,0	1180,1	1179,3	1183,2	1180,5	1185,0	1182,0	1183,4	1181,7
			anon2:	403,6	403,2	404,7	404,0	403,4	403,1	403,5	402,5	402,8
1%	0,01%	90%	anon3:	390,6	390,8	391,5	390,4	390,7	390,6	391,6	391,0	390,2
			anon4:	381,3	381,1	381,5	381,3	380,8	381,4	381,1	381,1	381,0
			anon5:	374,2	374,6	374,8	374,5	373,0	373,5	373,8	375,0	375,0
			anon6:	368,7	369,0	368,4	368,1	367,4	368,1	368,0	369,0	368,6
			anon7:	361,5	362,4	361,1	362,2	361,2	360,2	359,5	360,3	362,1
			anon8:	356,8	356,7	356,7	357,8	357,5	356,1	356,3	356,8	357,1
			anon9:	350,9	351,1	351,4	350,7	350,5	350,8	350,4	350,7	350,3
			anon10:	343,8	343,7	343,9	343,7	345,2	343,7	343,9	342,5	343,5
			Entropie:	18,3391	18,2742	18,2719	18,2713	18,2718	18,2703	18,2714	18,2712	18,2707
			Nutzer1:	84,1651%	83,4807%	83,3673%	83,3453%	83,3351%	83,3238%	83,3199%	83,3165%	83,3168%
			Nutzer2:	5,1037%	5,0367%	5,0645%	5,0788%	5,0883%	5,0982%	5,1031%	5,108%	5,1068%
			Nutzer2-9:	9,6999%	9,5757%	9,6748%	9,6974%	9,7056%	9,7203%	9,7232%	9,7271%	9,7254%
			Nutzer10+:	6,135%	6,9436%	6,9579%	6,9573%	6,9592%	6,9559%	6,957%	6,9564%	6,9578%
			anon1:	1154,8	1671,7	1654,1	1668,7	1658,1	1810,1	1690,4	1653,0	1698,4
			anon2:	394,4	1142,0	1142,1	1141,9	1142,0	1142,1	1142,6	1141,9	1144,0
1%	0,01%	10%	anon3:	383,4	391,3	390,2	391,4	390,2	391,5	397,5	390,5	398,8
			anon4:	378,0	380,6	380,2	383,0	380,2	384,3	383,8	380,4	384,0
			anon5:	373,7	375,2	373,9	374,4	373,9	374,3	376,3	374,1	377,0
			anon6:	365,5	371,6	369,4	370,7	369,7	369,6	370,3	369,3	370,1
			anon7:	361,3	362,9	362,2	364,4	362,6	364,1	364,3	361,9	362,8
			anon8:	356,3	357,4	357,5	358,2	357,0	357,7	358,9	357,0	358,5
			anon9:	352,2	353,3	353,1	353,2	353,0	353,9	354,3	353,0	353,7
			anon10:	345,4	349,5	349,6	349,9	349,4	349,8	349,6	349,4	349,8
			Entropie:	18,338	18,338	18,3379	18,3379	18,3379	18,3379	18,3379	18,3379	18,3379
			Nutzer1:	84,1522%	84,1519%	84,1516%	84,1511%	84,151%	84,1505%	84,1503%	84,1503%	84,1502%
			Nutzer2:	5,1135%	5,1138%	5,1142%	5,1144%	5,1144%	5,1148%	5,1153%	5,115%	5,1153%
			Nutzer2-9:	9,6824%	9,6826%	9,6831%	9,6833%	9,6834%	9,6841%	9,684%	9,6842%	9,6845%
			Nutzer10+:	6,1654%	6,1654%	6,1653%	6,1656%	6,1656%	6,1654%	6,1657%	6,1655%	6,1653%
			anon1:	1182,3	1182,5	1182,5	1182,5	1182,5	1182,5	1182,5	1182,5	1182,5
			anon2:	394,7	394,7	394,7	394,7	394,7	394,7	394,7	394,7	394,7
0,01%	1%	90%	anon3:	384,6	384,5	384,5	384,4	384,5	384,4	384,5	384,5	384,5
			anon4:	376,8	376,7	376,7	376,7	376,7	376,7	376,7	376,7	376,7
			anon5:	372,7	372,7	372,7	372,7	372,7	372,7	372,7	372,7	372,7
			anon6:	365,5	365,5	365,5	365,5	365,5	365,5	365,5	365,5	365,5
			anon7:	361,0	361,0	361,0	361,0	361,0	361,0	361,0	361,0	361,0
			anon8:	357,0	357,0	357,0	357,0	357,0	357,0	357,0	357,0	357,0
			anon9:	351,3	351,3	351,3	351,3	351,3	351,3	351,3	351,3	351,3
			anon10:	344,1	344,1	344,1	344,1	344,1	344,1	344,1	344,1	344,1
			Entropie:	18,3377	18,3369	18,3369	18,3369	18,3369	18,3369	18,3369	18,3369	18,3369
			Nutzer1:	84,1584%	84,1477%	84,1475%	84,1475%	84,1475%	84,1477%	84,1475%	84,1476%	84,1474%
			Nutzer2:	5,1041%	5,1074%	5,1077%	5,1077%	5,1078%	5,1074%	5,1077%	5,1075%	5,108%
			Nutzer2-9:	9,6736%	9,6767%	9,6773%	9,6772%	9,6773%	9,6766%	9,6773%	9,6772%	9,6776%
			Nutzer10+:	6,168%	6,1755%	6,1753%	6,1753%	6,1752%	6,1757%	6,1752%	6,1752%	6,175%
			anon1:	1151,5	1179,0	1177,3	1172,7	1173,0	1176,9	1179,4	1174,0	1179,3
			anon2:	399,6	400,0	399,7	401,1	404,8	400,3	400,0	404,2	399,9
0,01%	1%	10%	anon3:	390,7	390,8	390,6	391,2	391,3	390,8	390,6	390,9	390,7
			anon4:	381,7	382,4	382,0	382,3	382,6	382,3	382,2	382,5	382,2
			anon5:	375,6	376,1	375,8	377,3	376,1	376,5	376,0	376,0	376,3
			anon6:	372,0	372,3	372,2	373,1	372,5	372,5	372,3	372,4	372,5
			anon7:	367,5	367,8	368,0	369,3	368,3	368,2	368,0	368,5	368,0
			anon8:	361,1	361,5	361,4	361,9	361,6	361,5	361,6	361,4	361,4
			anon9:	356,6	356,8	356,7	357,7	356,6	356,9	356,7	357,0	356,7
			anon10:	349,6	350,0	350,2	350,2	350,1	349,9	349,9	350,1	349,9

p_{fake}	p_{visit}	p_{fixed}		1.Tick	2.Tick	10.Tick	20.Tick	30.Tick	50.Tick	70.Tick	90.Tick	100.Tick
			Entropie:	18,3389	18,3389	18,3389	18,3389	18,3389	18,3389	18,3389	18,3389	18,3389
			Nutzer1:	84,184%	84,184%	84,184%	84,184%	84,1839%	84,1839%	84,1839%	84,1839%	84,1839%
			Nutzer2:	5,1138%	5,1138%	5,1138%	5,1139%	5,1138%	5,1139%	5,114%	5,1139%	5,1139%
			Nutzer2-9:	9,6772%	9,6772%	9,6772%	9,6773%	9,6773%	9,6773%	9,6774%	9,6773%	9,6773%
			Nutzer10+:	6,1387%	6,1387%	6,1388%	6,1387%	6,1388%	6,1388%	6,1387%	6,1388%	6,1388%
			anon1:	1167,2	1167,2	1167,2	1167,1	1167,1	1167,2	1167,1	1167,2	1167,2
			anon2:	405,6	405,6	405,6	405,5	405,6	405,5	405,5	405,6	405,6
0,01%	0,01%	90%	anon3:	392,6	392,6	392,6	392,6	392,6	392,7	392,7	392,6	392,6
			anon4:	383,4	383,4	383,4	383,4	383,4	383,4	383,4	383,4	383,4
			anon5:	375,4	375,4	375,4	375,5	375,5	375,4	375,5	375,4	375,4
			anon6:	370,0	370,0	369,9	369,9	370,1	369,9	370,0	370,1	370,0
			anon7:	366,4	366,4	366,4	366,4	366,4	366,4	366,4	366,4	366,4
			anon8:	363,4	363,4	363,4	363,4	363,4	363,4	363,4	363,4	363,4
			anon9:	358,8	358,8	358,8	358,8	358,8	358,8	358,8	358,8	358,8
			anon10:	350,4	350,4	350,4	350,4	350,4	350,4	350,4	350,4	350,4
			Entropie:	18,3383	18,338	18,338	18,338	18,338	18,338	18,338	18,338	18,338
			Nutzer1:	84,1633%	84,1527%	84,1516%	84,1516%	84,1514%	84,1511%	84,1515%	84,1513%	84,151%
			Nutzer2:	5,0918%	5,0949%	5,0953%	5,0954%	5,0958%	5,096%	5,0957%	5,096%	5,0966%
			Nutzer2-9:	9,681%	9,6886%	9,6895%	9,6886%	9,6893%	9,6903%	9,689%	9,6893%	9,6893%
			Nutzer10+:	6,1557%	6,1587%	6,1589%	6,1598%	6,1594%	6,1586%	6,1595%	6,1594%	6,1596%
			anon1:	1177,6	1177,7	1178,1	1180,1	1178,5	1177,4	1177,4	1178,1	1178,3
			anon2:	393,4	393,3	393,3	393,3	393,3	393,3	393,8	393,3	393,3
0,01%	0,01%	10%	anon3:	385,5	385,5	385,5	385,5	385,5	385,5	385,5	385,5	385,6
			anon4:	379,6	379,6	379,6	379,6	379,6	379,7	380,0	379,6	379,6
			anon5:	373,3	373,3	373,3	373,5	373,3	373,3	373,5	373,3	373,3
			anon6:	368,4	368,5	368,3	368,3	368,3	368,3	368,3	368,3	368,3
			anon7:	362,0	361,9	361,9	361,9	361,9	361,9	362,0	361,9	361,9
			anon8:	356,5	356,4	356,4	356,4	356,4	356,5	356,4	356,5	356,5
			anon9:	352,0	352,0	352,2	352,0	352,0	352,0	352,0	352,0	352,0
			anon10:	343,9	343,9	343,9	343,9	344,0	343,9	343,9	343,9	343,9

Varationskoeffizienten - eingeschränkte Fälschung von Fingerprints

p_{fake}	p_{visit}	p_{fixed}		1.Tick	2.Tick	10.Tick	20.Tick	30.Tick	50.Tick	70.Tick	90.Tick	100.Tick
			Entropie:	0,01%	0,02%	0,02%	0,02%	0,02%	0,02%	0,02%	0,02%	0,02%
			Nutzer1:	0,07%	0,08%	0,08%	0,08%	0,08%	0,08%	0,08%	0,08%	0,08%
			Nutzer2:	0,81%	0,77%	0,85%	0,83%	0,84%	0,74%	0,79%	0,8%	0,82%
			Nutzer2-9:	0,71%	0,62%	0,67%	0,67%	0,68%	0,65%	0,67%	0,68%	0,65%
			Nutzer10+:	0,8%	0,67%	0,68%	0,72%	0,72%	0,75%	0,76%	0,75%	0,75%
			anon1:	3,26%	12,38%	6,86%	6,86%	6,86%	2,14%	2,14%	2,14%	2,14%
			anon2:	2,64%	74,99%	48,25%	42,62%	48,37%	1,78%	1,83%	3,31%	2,88%
1%	1%	10%	anon3:	2,33%	2,39%	3,48%	6,67%	2,75%	3,82%	2,61%	2,82%	2,83%
			anon4:	2,44%	3,18%	4,02%	4,28%	2,61%	5,2%	2,93%	3,65%	2,82%
			anon5:	2,14%	3,34%	4,65%	3,54%	1,73%	5,33%	3,25%	3,82%	3,2%
			anon6:	2,41%	3,37%	4,38%	4,31%	2,34%	5,62%	2,71%	4,58%	4,15%
			anon7:	1,74%	4,31%	4,47%	3,75%	3,15%	5,12%	5,28%	6,4%	5,05%
			anon8:	1,99%	4,23%	4,55%	3,33%	3,37%	5,2%	4,23%	5,55%	4,44%
			anon9:	1,96%	2,14%	3,06%	2,16%	4,1%	3,93%	3,73%	3,31%	4,14%
			anon10:	1,74%	1,27%	3,75%	2,83%	3,5%	2,81%	3,88%	2,93%	4,25%
			Entropie:	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%
			Nutzer1:	0,05%	0,05%	0,05%	0,04%	0,05%	0,04%	0,04%	0,04%	0,04%
			Nutzer2:	0,49%	0,48%	0,44%	0,44%	0,43%	0,37%	0,42%	0,44%	0,37%
			Nutzer2-9:	0,41%	0,42%	0,37%	0,38%	0,38%	0,32%	0,34%	0,32%	0,33%
			Nutzer10+:	0,64%	0,66%	0,66%	0,65%	0,65%	0,67%	0,65%	0,63%	0,65%
			anon1:	2,98%	3,03%	3,18%	3,1%	3,08%	3,07%	3,21%	3,09%	3,09%
			anon2:	2,21%	2,25%	2,27%	2,13%	2,22%	2,25%	2,33%	2,23%	2,22%
1%	1%	90%	anon3:	1,61%	1,62%	1,76%	1,62%	1,66%	1,62%	1,74%	1,64%	1,78%
			anon4:	2,0%	2,09%	2,06%	1,93%	1,96%	2,0%	2,03%	2,12%	1,97%
			anon5:	2,14%	1,99%	1,74%	2,01%	2,09%	2,0%	1,76%	1,95%	1,96%
			anon6:	2,36%	2,34%	2,3%	2,47%	2,34%	2,36%	2,44%	2,39%	2,32%
			anon7:	2,6%	2,52%	2,59%	2,53%	2,56%	2,57%	2,59%	2,52%	2,48%
			anon8:	2,1%	2,38%	2,02%	2,3%	2,02%	2,31%	2,4%	2,32%	2,31%
			anon9:	2,38%	2,24%	2,18%	2,22%	2,01%	2,17%	2,27%	2,09%	2,27%
			anon10:	2,1%	2,25%	2,19%	2,24%	1,96%	2,15%	2,14%	2,3%	1,95%
			Entropie:	0,02%	0,02%	0,02%	0,02%	0,02%	0,02%	0,01%	0,02%	0,02%
			Nutzer1:	0,07%	0,07%	0,07%	0,07%	0,07%	0,07%	0,07%	0,07%	0,07%
			Nutzer2:	0,63%	0,67%	0,69%	0,66%	0,69%	0,64%	0,68%	0,65%	0,63%
			Nutzer2-9:	0,51%	0,54%	0,54%	0,55%	0,56%	0,53%	0,59%	0,54%	0,56%
			Nutzer10+:	0,79%	0,7%	0,75%	0,75%	0,76%	0,71%	0,73%	0,71%	0,72%
			anon1:	2,95%	2,91%	2,46%	2,55%	2,64%	19,53%	5,58%	1,98%	4,44%
			anon2:	3,4%	2,8%	2,78%	2,81%	2,79%	2,81%	2,89%	2,81%	3,02%
1%	0,01%	10%	anon3:	2,38%	3,56%	3,53%	4,12%	3,53%	3,99%	5,49%	3,53%	6,23%
			anon4:	2,41%	2,5%	2,59%	3,69%	2,59%	3,91%	3,11%	2,62%	3,33%
			anon5:	2,06%	2,51%	2,43%	2,53%	2,43%	2,45%	2,58%	2,51%	2,7%
			anon6:	2,15%	2,61%	1,81%	2,44%	1,88%	2,08%	1,83%	1,84%	1,8%
			anon7:	2,05%	2,31%	1,81%	2,32%	1,82%	1,72%	1,95%	1,88%	1,98%
			anon8:	2,24%	1,93%	1,69%	1,92%	1,8%	1,53%	1,9%	1,8%	1,86%
			anon9:	2,27%	2,33%	2,19%	2,2%	2,27%	1,35%	2,13%	2,21%	2,07%
			anon10:	2,48%	2,44%	2,39%	2,42%	2,41%	2,1%	2,4%	2,41%	2,46%
			Entropie:	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%
			Nutzer1:	0,07%	0,07%	0,07%	0,07%	0,07%	0,07%	0,07%	0,07%	0,07%
			Nutzer2:	0,6%	0,6%	0,6%	0,59%	0,58%	0,59%	0,6%	0,61%	0,61%
			Nutzer2-9:	0,63%	0,63%	0,65%	0,64%	0,64%	0,66%	0,65%	0,64%	0,64%
			Nutzer10+:	0,47%	0,46%	0,48%	0,48%	0,48%	0,49%	0,5%	0,5%	0,5%
			anon1:	1,8%	1,8%	2,06%	1,95%	2,27%	1,82%	1,96%	1,79%	1,94%
			anon2:	2,61%	2,59%	2,68%	2,81%	2,82%	2,84%	2,9%	2,79%	2,59%
1%	0,01%	90%	anon3:	3,17%	3,14%	3,03%	3,19%	2,93%	2,8%	2,78%	2,7%	2,76%
			anon4:	2,39%	2,36%	2,25%	2,69%	2,32%	2,09%	2,61%	2,4%	2,22%
			anon5:	2,22%	2,08%	2,56%	2,0%	2,09%	2,24%	2,27%	2,08%	2,33%
			anon6:	1,48%	1,41%	1,7%	1,36%	1,57%	1,68%	1,55%	1,42%	1,49%
			anon7:	2,29%	2,23%	2,23%	2,36%	2,31%	2,34%	2,27%	2,29%	1,67%
			anon8:	2,08%	2,09%	2,05%	2,06%	2,39%	2,28%	2,2%	2,27%	2,31%
			anon9:	1,98%	2,06%	1,91%	2,11%	2,12%	2,15%	2,15%	1,93%	2,1%
			anon10:	2,49%	2,47%	2,58%	2,78%	2,25%	2,74%	2,63%	2,75%	2,9%
			Entropie:	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%
			Nutzer1:	0,07%	0,07%	0,07%	0,07%	0,07%	0,07%	0,07%	0,07%	0,07%
			Nutzer2:	1,13%	1,12%	1,12%	1,12%	1,12%	1,12%	1,13%	1,13%	1,12%
			Nutzer2-9:	0,62%	0,63%	0,62%	0,62%	0,62%	0,63%	0,62%	0,62%	0,62%
			Nutzer10+:	0,53%	0,55%	0,54%	0,54%	0,54%	0,54%	0,54%	0,54%	0,54%
			anon1:	3,72%	3,59%	3,46%	3,92%	3,51%	3,87%	3,6%	3,47%	3,6%
			anon2:	2,55%	2,62%	2,52%	2,41%	3,72%	2,51%	2,63%	4,2%	2,58%
0,01%	1%	10%	anon3:	2,46%	2,52%	2,5%	2,54%	2,6%	2,45%	2,5%	2,54%	2,51%
			anon4:	1,59%	1,64%	1,56%	1,72%	1,82%	1,6%	1,58%	1,82%	1,59%
			anon5:	1,17%	1,2%	1,15%	1,65%	1,19%	1,24%	1,25%	1,21%	1,16%
			anon6:	1,13%	1,19%	1,09%	1,5%	1,1%	1,02%	1,19%	1,09%	1,1%
			anon7:	1,51%	1,48%	1,39%	1,3%	1,24%	1,46%	1,41%	1,23%	1,36%
			anon8:	2,03%	1,85%	1,85%	2,07%	1,88%	2,01%	1,86%	1,85%	1,87%
			anon9:	2,04%	2,05%	2,05%	2,42%	2,04%	2,09%	2,05%	2,08%	2,07%
			anon10:	2,52%	2,59%	2,57%	2,62%	2,55%	2,57%	2,57%	2,6%	2,53%
			Entropie:	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%
			Nutzer1:	0,07%	0,07%	0,07%	0,07%	0,07%	0,07%	0,07%	0,07%	0,07%
			Nutzer2:	0,73%	0,74%	0,74%	0,74%	0,74%	0,74%	0,74%	0,74%	0,75%
			Nutzer2-9:	0,57%	0,57%	0,57%	0,57%	0,57%	0,57%	0,57%	0,57%	0,57%
			Nutzer10+:	0,42%	0,42%	0,42%	0,42%	0,42%	0,42%	0,42%	0,42%	0,42%
			anon1:	2,52%	2,5%	2,5%	2,5%	2,5%	2,5%	2,5%	2,5%	2,5%
			anon2:	2,82%	2,82%	2,82%	2,82%	2,82%	2,82%	2,82%	2,82%	2,82%
0,01%	1%	90%	anon3:	2,15%	2,17%	2,17%	2,15%	2,17%	2,15%	2,17%	2,17%	2,17%
			anon4:	3,43%	3,44%	3,44%	3,44%	3,44%	3,44%	3,44%	3,44%	3,44%
			anon5:	3,3%	3,3%	3,3%	3,3%	3,3%	3,3%	3,3%	3,3%	3,3%
			anon6:	2,86%	2,86%	2,86%	2,86%	2,86%	2,86%	2,86%	2,86%	2,86%
			anon7:	2,77%	2,77%	2,77%	2,77%	2,77%	2,77%	2,77%	2,77%	2,77%
			anon8:	2,89%	2,89%	2,89%	2,89%	2,89%	2,89%	2,89%	2,89%	2,89%
			anon9:	2,29%	2,29%	2,29%	2,29%	2,29%	2,29%	2,29%	2,29%	2,29%
			anon10:	2,23%	2,23%	2,23%	2,23%	2,23%	2,23%	2,23%	2,23%	2,23%

p_{fake}	p_{visit}	p_{fixed}		1.Tick	2.Tick	10.Tick	20.Tick	30.Tick	50.Tick	70.Tick	90.Tick	100.Tick
0,01%	0,01%	10%	Entropie:	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%
			Nutzer1:	0,1%	0,1%	0,1%	0,1%	0,1%	0,1%	0,1%	0,1%	0,1%
			Nutzer2:	1,12%	1,11%	1,1%	1,11%	1,12%	1,12%	1,11%	1,11%	1,12%
			Nutzer2-9:	0,87%	0,86%	0,86%	0,86%	0,86%	0,86%	0,86%	0,86%	0,86%
			Nutzer10+:	0,46%	0,47%	0,46%	0,48%	0,47%	0,47%	0,47%	0,47%	0,46%
			anon1:	2,46%	2,49%	2,61%	2,33%	2,4%	2,47%	2,47%	2,41%	2,41%
			anon2:	2,2%	2,21%	2,21%	2,21%	2,21%	2,21%	2,07%	2,21%	2,21%
			anon3:	2,31%	2,31%	2,31%	2,31%	2,31%	2,31%	2,31%	2,31%	2,32%
			anon4:	2,35%	2,35%	2,35%	2,35%	2,35%	2,39%	2,34%	2,35%	2,35%
			anon5:	2,41%	2,41%	2,41%	2,4%	2,41%	2,41%	2,44%	2,41%	2,41%
			anon6:	2,21%	2,19%	2,24%	2,24%	2,24%	2,24%	2,24%	2,24%	2,24%
			anon7:	0,91%	0,94%	0,94%	0,94%	0,94%	0,94%	0,94%	0,94%	0,94%
			anon8:	1,78%	1,74%	1,74%	1,74%	1,74%	1,75%	1,74%	1,77%	1,78%
			anon9:	1,57%	1,57%	1,47%	1,57%	1,57%	1,57%	1,57%	1,57%	1,57%
			anon10:	1,96%	1,96%	1,96%	1,96%	2,0%	1,96%	1,96%	1,96%	1,96%
0,01%	0,01%	90%	Entropie:	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%
			Nutzer1:	0,08%	0,08%	0,08%	0,08%	0,08%	0,08%	0,08%	0,08%	0,08%
			Nutzer2:	1,06%	1,06%	1,06%	1,06%	1,06%	1,06%	1,06%	1,06%	1,06%
			Nutzer2-9:	0,83%	0,83%	0,83%	0,83%	0,83%	0,83%	0,83%	0,83%	0,83%
			Nutzer10+:	0,83%	0,83%	0,83%	0,83%	0,83%	0,83%	0,83%	0,83%	0,83%
			anon1:	3,11%	3,11%	3,11%	3,11%	3,11%	3,11%	3,11%	3,11%	3,11%
			anon2:	3,1%	3,1%	3,1%	3,11%	3,1%	3,11%	3,11%	3,1%	3,1%
			anon3:	2,33%	2,33%	2,33%	2,33%	2,33%	2,34%	2,34%	2,33%	2,33%
			anon4:	0,98%	0,98%	0,98%	0,98%	0,98%	0,98%	0,98%	0,98%	0,98%
			anon5:	1,59%	1,59%	1,59%	1,6%	1,6%	1,59%	1,6%	1,59%	1,59%
			anon6:	1,34%	1,34%	1,38%	1,38%	1,34%	1,38%	1,34%	1,34%	1,34%
			anon7:	1,43%	1,43%	1,43%	1,43%	1,43%	1,43%	1,43%	1,43%	1,43%
			anon8:	2,1%	2,1%	2,1%	2,1%	2,1%	2,1%	2,1%	2,1%	2,1%
			anon9:	2,04%	2,04%	2,04%	2,04%	2,04%	2,04%	2,04%	2,04%	2,04%
			anon10:	2,28%	2,28%	2,28%	2,28%	2,28%	2,28%	2,28%	2,28%	2,28%

A.7. Detaillierte Tabellen zur Fälschung mit Hilfe der Fingerprints anderer Fälscher

Werte - Fälschung mit Hilfe der Fingerprints anderer Fälscher

p_{random}	p_{fixed}		1.Tick	2.Tick	3.Tick	4.Tick	5.Tick
		Entropie:	18,3381	18,3378	18,3378	18,3378	18,3378
		Nutzer1:	84,1552%	84,1456%	84,1455%	84,1455%	84,1455%
		Nutzer2:	5,1235%	5,1291%	5,1287%	5,1287%	5,1287%
		Nutzer2-9:	9,6957%	9,7025%	9,7024%	9,7024%	9,7024%
		Nutzer10+:	6,1492%	6,1519%	6,1521%	6,1521%	6,1521%
		anon1:	1165,4	1172,6	1172,6	1172,6	1172,6
		anon2:	393,8	395,2	395,3	395,3	395,3
1%	90%	anon3:	384,4	385,0	384,9	384,9	384,9
		anon4:	377,8	379,1	379,1	379,0	379,0
		anon5:	374,9	375,0	375,1	375,1	375,1
		anon6:	367,7	368,4	368,2	368,2	368,2
		anon7:	364,7	365,2	364,9	364,7	364,6
		anon8:	361,2	361,8	361,6	361,5	361,3
		anon9:	357,0	358,1	358,1	358,1	358,0
		anon10:	348,3	349,4	349,4	349,4	349,4
		Entropie:	18,3385	18,2565	18,2529	18,253	18,2531
		Nutzer1:	84,1786%	83,4879%	83,4372%	83,433%	83,4314%
		Nutzer2:	5,0897%	5,0578%	5,0361%	5,0319%	5,031%
		Nutzer2-9:	9,686%	9,6454%	9,6279%	9,6243%	9,6229%
		Nutzer10+:	6,1354%	6,8667%	6,9349%	6,9428%	6,9457%
		anon1:	1175,2	3526,0	3571,7	3571,7	3571,7
		anon2:	401,7	586,2	538,3	524,5	524,0
1%	10%	anon3:	386,4	428,5	427,7	415,2	404,0
		anon4:	379,2	423,0	416,8	404,7	392,3
		anon5:	374,3	413,2	407,1	387,7	381,6
		anon6:	367,9	403,7	395,1	382,3	374,6
		anon7:	364,4	399,8	385,1	371,4	368,6
		anon8:	361,6	382,8	379,4	366,4	364,1
		anon9:	355,8	371,2	370,1	360,6	359,2
		anon10:	351,4	364,3	361,0	355,3	353,0
		Entropie:	18,3386	18,3262	18,3235	18,3232	18,3232
		Nutzer1:	84,1956%	83,8312%	83,7925%	83,7993%	83,7997%
		Nutzer2:	5,0809%	5,2813%	5,1676%	5,1492%	5,1489%
		Nutzer2-9:	9,6463%	9,9197%	9,9294%	9,9022%	9,8951%
		Nutzer10+:	6,1581%	6,2491%	6,278%	6,2985%	6,3053%
		anon1:	1164,4	1313,1	1313,1	1313,1	1313,1
		anon2:	394,2	424,9	422,5	422,2	421,9
1%	50%	anon3:	383,4	403,2	401,9	401,1	400,6
		anon4:	379,6	394,3	392,3	391,7	391,1
		anon5:	374,0	385,5	384,5	382,8	381,8
		anon6:	367,7	380,6	379,5	378,6	378,3
		anon7:	363,2	374,9	372,6	371,8	370,8
		anon8:	358,9	370,3	367,8	366,1	364,6
		anon9:	356,2	364,6	362,7	362,5	362,2
		anon10:	349,5	357,6	356,7	356,6	356,3
		Entropie:	18,3388	18,2143	18,2143	18,2143	18,2143
		Nutzer1:	84,1777%	83,3839%	83,3839%	83,3839%	83,3839%
		Nutzer2:	5,0987%	5,0268%	5,0268%	5,0268%	5,0268%
		Nutzer2-9:	9,6567%	9,5272%	9,5272%	9,5272%	9,5272%
		Nutzer10+:	6,1656%	7,0889%	7,0889%	7,0889%	7,0889%
		anon1:	1159,3	5869,3	5869,3	5869,3	5869,3
		anon2:	396,7	393,6	393,6	393,6	393,6
1%	0%	anon3:	384,7	379,7	379,7	379,7	379,7
		anon4:	378,4	374,8	374,8	374,8	374,8
		anon5:	371,5	368,5	368,5	368,5	368,5
		anon6:	368,2	364,8	364,8	364,8	364,8
		anon7:	365,9	362,8	362,8	362,8	362,8
		anon8:	363,7	359,9	359,9	359,9	359,9
		anon9:	357,8	354,3	354,3	354,3	354,3
		anon10:	347,0	342,4	342,4	342,4	342,4
		Entropie:	18,3394	18,3394	18,3394	18,3394	18,3394
		Nutzer1:	84,1978%	84,1978%	84,1978%	84,1978%	84,1978%
		Nutzer2:	5,09%	5,09%	5,09%	5,09%	5,09%
		Nutzer2-9:	9,6563%	9,6563%	9,6563%	9,6563%	9,6563%
		Nutzer10+:	6,1459%	6,1459%	6,1459%	6,1459%	6,1459%
		anon1:	1169,4	1169,4	1169,4	1169,4	1169,4
		anon2:	397,0	397,0	397,0	397,0	397,0
0,01%	90%	anon3:	387,6	387,6	387,6	387,6	387,6
		anon4:	380,3	380,3	380,3	380,3	380,3
		anon5:	375,1	375,1	375,1	375,1	375,1
		anon6:	368,5	368,5	368,5	368,5	368,5
		anon7:	363,1	363,1	363,1	363,1	363,1
		anon8:	359,1	359,1	359,1	359,1	359,1
		anon9:	353,3	353,3	353,3	353,3	353,3
		anon10:	346,3	346,3	346,3	346,3	346,3
		Entropie:	18,3384	18,3383	18,3382	18,3382	18,3382
		Nutzer1:	84,1656%	84,1598%	84,1591%	84,159%	84,159%
		Nutzer2:	5,1035%	5,1042%	5,1035%	5,1035%	5,1035%
		Nutzer2-9:	9,6746%	9,6773%	9,6771%	9,6772%	9,6772%
		Nutzer10+:	6,1598%	6,1629%	6,1638%	6,1638%	6,1638%
		anon1:	1163,3	1163,3	1163,3	1163,3	1163,3
		anon2:	399,3	399,3	399,3	399,3	399,3
0,01%	10%	anon3:	390,0	390,0	390,0	390,0	390,0
		anon4:	383,1	383,1	383,1	383,1	383,1
		anon5:	375,1	375,1	375,1	375,1	375,1
		anon6:	369,9	369,8	369,8	369,8	369,8
		anon7:	364,4	364,4	364,4	364,4	364,4
		anon8:	359,9	359,9	359,9	359,9	359,9
		anon9:	354,5	354,5	354,5	354,5	354,5
		anon10:	345,6	345,6	345,6	345,6	345,6

p_{random}	p_{fixed}		1.Tick	2.Tick	3.Tick	4.Tick	5.Tick
0,01%	50%	Entropie:	18,3379	18,3378	18,3378	18,3378	18,3378
		Nutzer1:	84,1327%	84,1321%	84,132%	84,132%	84,132%
		Nutzer2:	5,123%	5,1234%	5,1234%	5,1234%	5,1234%
		Nutzer2-9:	9,7136%	9,7143%	9,7143%	9,7143%	9,7143%
		Nutzer10+:	6,1537%	6,1537%	6,1537%	6,1537%	6,1537%
		anon1:	1163,7	1163,8	1163,9	1163,9	1163,9
		anon2:	397,9	397,9	397,9	397,9	397,9
		anon3:	388,1	388,0	388,0	388,0	388,0
		anon4:	379,5	379,5	379,5	379,5	379,5
		anon5:	373,1	373,1	373,1	373,1	373,1
		anon6:	368,0	368,0	367,9	367,9	367,9
		anon7:	364,6	364,6	364,6	364,6	364,6
		anon8:	356,7	356,7	356,7	356,7	356,7
		anon9:	352,2	352,2	352,2	352,2	352,2
		anon10:	348,1	348,1	348,1	348,1	348,1
0,01%	0%	Entropie:	18,3384	18,3379	18,3379	18,3379	18,3379
		Nutzer1:	84,1564%	84,1485%	84,1485%	84,1485%	84,1485%
		Nutzer2:	5,1086%	5,108%	5,108%	5,108%	5,108%
		Nutzer2-9:	9,6988%	9,6976%	9,6976%	9,6976%	9,6976%
		Nutzer10+:	6,1448%	6,1538%	6,1538%	6,1538%	6,1538%
		anon1:	1152,3	1152,2	1152,2	1152,2	1152,2
		anon2:	394,6	394,6	394,6	394,6	394,6
		anon3:	385,1	385,1	385,1	385,1	385,1
		anon4:	379,4	379,4	379,4	379,4	379,4
		anon5:	374,0	374,0	374,0	374,0	374,0
		anon6:	369,2	369,2	369,2	369,2	369,2
		anon7:	363,9	363,9	363,9	363,9	363,9
		anon8:	357,4	357,4	357,4	357,4	357,4
		anon9:	352,5	352,5	352,5	352,5	352,5
		anon10:	345,5	345,5	345,5	345,5	345,5

Varationskoeffizienten - Fälschung mit Hilfe der Fingerprints anderer Fälscher

p_{random}	p_{fixed}		1.Tick	2.Tick	3.Tick	4.Tick	5.Tick
		Entropie:	0,02%	0,02%	0,02%	0,02%	0,02%
		Nutzer1:	0,08%	0,08%	0,08%	0,08%	0,08%
		Nutzer2:	0,91%	0,93%	0,93%	0,93%	0,93%
		Nutzer2-9:	0,65%	0,65%	0,65%	0,65%	0,65%
		Nutzer10+:	0,5%	0,39%	0,39%	0,39%	0,39%
		anon1:	2,34%	1,31%	1,31%	1,31%	1,31%
		anon2:	3,45%	3,45%	3,45%	3,45%	3,45%
1%	0%	anon3:	2,49%	2,48%	2,48%	2,48%	2,48%
		anon4:	1,75%	1,82%	1,82%	1,82%	1,82%
		anon5:	2,5%	2,43%	2,43%	2,43%	2,43%
		anon6:	2,16%	2,2%	2,2%	2,2%	2,2%
		anon7:	2,02%	2,17%	2,17%	2,17%	2,17%
		anon8:	2,08%	2,32%	2,32%	2,32%	2,32%
		anon9:	2,86%	2,66%	2,66%	2,66%	2,66%
		anon10:	4,24%	4,12%	4,12%	4,12%	4,12%
		Entropie:	0,02%	0,01%	0,01%	0,01%	0,01%
		Nutzer1:	0,07%	0,07%	0,07%	0,07%	0,07%
		Nutzer2:	0,56%	0,61%	0,63%	0,6%	0,62%
		Nutzer2-9:	0,48%	0,48%	0,48%	0,48%	0,47%
		Nutzer10+:	0,62%	0,44%	0,44%	0,47%	0,48%
		anon1:	2,99%	14,27%	10,28%	10,28%	10,28%
		anon2:	4,14%	70,59%	51,56%	53,78%	53,87%
1%	10%	anon3:	2,59%	2,21%	2,87%	5,13%	5,22%
		anon4:	2,0%	2,16%	3,05%	4,65%	4,98%
		anon5:	1,84%	4,1%	3,41%	3,25%	3,23%
		anon6:	2,1%	4,04%	2,95%	2,77%	2,66%
		anon7:	1,84%	4,04%	4,13%	2,38%	2,22%
		anon8:	2,01%	4,63%	4,24%	2,15%	2,2%
		anon9:	1,67%	3,63%	4,11%	1,99%	2,16%
		anon10:	1,9%	3,86%	3,67%	2,16%	2,16%
		Entropie:	0,01%	0,01%	0,01%	0,01%	0,01%
		Nutzer1:	0,04%	0,05%	0,04%	0,04%	0,04%
		Nutzer2:	0,69%	0,64%	0,63%	0,68%	0,67%
		Nutzer2-9:	0,6%	0,65%	0,58%	0,61%	0,61%
		Nutzer10+:	0,59%	0,65%	0,52%	0,54%	0,57%
		anon1:	2,35%	4,42%	4,42%	4,42%	4,42%
		anon2:	2,52%	10,11%	10,45%	10,5%	10,51%
1%	50%	anon3:	2,21%	3,15%	3,33%	3,3%	3,21%
		anon4:	2,42%	1,93%	2,31%	2,52%	2,67%
		anon5:	2,02%	2,39%	2,64%	2,46%	2,55%
		anon6:	2,2%	2,53%	2,64%	2,63%	2,53%
		anon7:	2,12%	3,18%	2,96%	2,83%	2,76%
		anon8:	2,36%	3,08%	2,79%	2,6%	2,51%
		anon9:	2,39%	3,15%	2,77%	2,77%	2,71%
		anon10:	3,34%	3,06%	3,1%	3,07%	3,05%
		Entropie:	0,02%	0,02%	0,02%	0,02%	0,02%
		Nutzer1:	0,1%	0,1%	0,1%	0,1%	0,1%
		Nutzer2:	0,85%	0,84%	0,85%	0,85%	0,85%
		Nutzer2-9:	0,63%	0,63%	0,64%	0,64%	0,64%
		Nutzer10+:	0,78%	0,79%	0,79%	0,79%	0,79%
		anon1:	3,14%	3,1%	3,1%	3,1%	3,1%
		anon2:	3,15%	3,18%	3,2%	3,2%	3,2%
1%	90%	anon3:	2,04%	2,17%	2,14%	2,14%	2,14%
		anon4:	2,49%	2,36%	2,33%	2,3%	2,3%
		anon5:	2,34%	2,45%	2,48%	2,48%	2,48%
		anon6:	2,17%	2,15%	2,14%	2,14%	2,14%
		anon7:	2,3%	2,32%	2,27%	2,19%	2,15%
		anon8:	2,34%	2,36%	2,33%	2,28%	2,19%
		anon9:	2,25%	2,15%	2,15%	2,15%	2,15%
		anon10:	3,05%	3,19%	3,19%	3,19%	3,19%
		Entropie:	0,01%	0,01%	0,01%	0,01%	0,01%
		Nutzer1:	0,04%	0,04%	0,04%	0,04%	0,04%
		Nutzer2:	0,93%	0,92%	0,92%	0,92%	0,92%
		Nutzer2-9:	0,46%	0,46%	0,46%	0,46%	0,46%
		Nutzer10+:	0,74%	0,73%	0,73%	0,73%	0,73%
		anon1:	3,73%	3,73%	3,73%	3,73%	3,73%
		anon2:	2,6%	2,6%	2,6%	2,6%	2,6%
0,01%	0%	anon3:	1,18%	1,18%	1,18%	1,18%	1,18%
		anon4:	0,89%	0,89%	0,89%	0,89%	0,89%
		anon5:	0,77%	0,77%	0,77%	0,77%	0,77%
		anon6:	1,0%	1,0%	1,0%	1,0%	1,0%
		anon7:	1,29%	1,29%	1,29%	1,29%	1,29%
		anon8:	1,73%	1,73%	1,73%	1,73%	1,73%
		anon9:	1,29%	1,29%	1,29%	1,29%	1,29%
		anon10:	2,12%	2,12%	2,12%	2,12%	2,12%
		Entropie:	0,01%	0,01%	0,01%	0,01%	0,01%
		Nutzer1:	0,07%	0,07%	0,07%	0,07%	0,07%
		Nutzer2:	0,87%	0,87%	0,86%	0,87%	0,87%
		Nutzer2-9:	0,67%	0,67%	0,67%	0,67%	0,67%
		Nutzer10+:	0,79%	0,78%	0,78%	0,79%	0,79%
		anon1:	2,34%	2,34%	2,34%	2,34%	2,34%
		anon2:	2,88%	2,88%	2,88%	2,88%	2,88%
0,01%	10%	anon3:	2,37%	2,37%	2,37%	2,37%	2,37%
		anon4:	1,66%	1,66%	1,66%	1,66%	1,66%
		anon5:	1,96%	1,96%	1,96%	1,96%	1,96%
		anon6:	1,67%	1,68%	1,68%	1,68%	1,68%
		anon7:	2,15%	2,15%	2,15%	2,15%	2,15%
		anon8:	1,75%	1,75%	1,75%	1,75%	1,75%
		anon9:	2,16%	2,16%	2,16%	2,16%	2,16%
		anon10:	2,09%	2,09%	2,09%	2,09%	2,09%

p_{random}	p_{fixed}		1.Tick	2.Tick	3.Tick	4.Tick	5.Tick
0,01%	50%	Entropie:	0,01%	0,01%	0,01%	0,01%	0,01%
		Nutzer1:	0,1%	0,1%	0,1%	0,1%	0,1%
		Nutzer2:	1,01%	1,01%	1,01%	1,01%	1,01%
		Nutzer2-9:	0,79%	0,79%	0,79%	0,79%	0,79%
		Nutzer10+:	0,45%	0,45%	0,45%	0,45%	0,45%
		anon1:	3,16%	3,16%	3,16%	3,16%	3,16%
		anon2:	2,52%	2,52%	2,52%	2,52%	2,52%
		anon3:	2,17%	2,13%	2,13%	2,13%	2,13%
		anon4:	2,4%	2,4%	2,4%	2,4%	2,4%
		anon5:	2,67%	2,67%	2,67%	2,67%	2,67%
		anon6:	2,49%	2,49%	2,53%	2,53%	2,53%
		anon7:	2,59%	2,59%	2,59%	2,59%	2,59%
		anon8:	2,59%	2,59%	2,59%	2,59%	2,59%
		anon9:	2,64%	2,64%	2,64%	2,64%	2,64%
		anon10:	2,91%	2,91%	2,91%	2,91%	2,91%
0,01%	90%	Entropie:	0,01%	0,01%	0,01%	0,01%	0,01%
		Nutzer1:	0,08%	0,08%	0,08%	0,08%	0,08%
		Nutzer2:	0,91%	0,91%	0,91%	0,91%	0,91%
		Nutzer2-9:	0,57%	0,57%	0,57%	0,57%	0,57%
		Nutzer10+:	0,8%	0,8%	0,8%	0,8%	0,8%
		anon1:	3,52%	3,52%	3,52%	3,52%	3,52%
		anon2:	3,55%	3,55%	3,55%	3,55%	3,55%
		anon3:	2,34%	2,34%	2,34%	2,34%	2,34%
		anon4:	2,69%	2,69%	2,69%	2,69%	2,69%
		anon5:	2,49%	2,49%	2,49%	2,49%	2,49%
		anon6:	1,99%	1,99%	1,99%	1,99%	1,99%
		anon7:	1,89%	1,89%	1,89%	1,89%	1,89%
		anon8:	1,98%	1,98%	1,98%	1,98%	1,98%
		anon9:	2,11%	2,11%	2,11%	2,11%	2,11%
		anon10:	2,7%	2,7%	2,7%	2,7%	2,7%

A.8. Detaillierte Tabellen zur Randomisierung von Fingerprints

Varationskoeffizienten - eingeschränkte Fälschung von Fingerprints

p_{random}	$p_{randomTick}$		1.Tick	2.Tick	10.Tick	20.Tick	30.Tick	50.Tick	70.Tick	90.Tick	100.Tick
100%	100%	AnzKorr:	1,0	2,0	10,0	20,0	30,0	50,0	70,0	90,0	100,0
		AnzFing:	0,0	994,2	1000,0	1000,0	1000,0	1000,0	1000,0	1000,0	1000,0
100%	10%	AnzKorr:	1,0	1,6482	6,8653	13,396	19,9021	32,9084	45,952	58,958	65,4648
		AnzFing:	0,0	0,0	7,9	239,8	616,6	943,8	993,7	999,2	999,7
100%	1%	AnzKorr:	1,0	1,098	1,8711	2,8315	3,7824	5,7017	7,6211	9,5195	10,4671
		AnzFing:	0,0	0,0	0,0	0,0	0,0	0,1	1,4	6,0	10,5
10%	100%	AnzKorr:	1,0	1,6508	6,8487	13,3288	19,7892	32,6584	45,4467	58,1656	64,4921
		AnzFing:	349,2	999,8	1000,0	1000,0	1000,0	1000,0	1000,0	1000,0	1000,0
10%	10%	AnzKorr:	1,0	1,0942	1,8431	2,7773	3,7241	5,6308	7,5541	9,4741	10,4447
		AnzFing:	348,2	388,3	667,0	871,6	953,4	995,3	999,6	1000,0	1000,0
10%	1%	AnzKorr:	1,0	1,0083	1,0908	1,1921	1,2923	1,4957	1,6896	1,8848	1,9865
		AnzFing:	347,6	350,9	383,1	421,4	459,9	530,4	595,3	652,4	679,2
1%	100%	AnzKorr:	1,0	1,0967	1,868	2,829	3,7833	5,6811	7,5619	9,4244	10,3517
		AnzFing:	903,3	1000,0	1000,0	1000,0	1000,0	1000,0	1000,0	1000,0	1000,0
1%	10%	AnzKorr:	1,0	1,0112	1,0938	1,1946	1,3002	1,5075	1,7053	1,9086	2,0091
		AnzFing:	901,7	911,9	963,1	987,4	996,6	999,7	1000,0	1000,0	1000,0
1%	1%	AnzKorr:	1,0	1,0006	1,0073	1,0169	1,0266	1,0459	1,0644	1,0811	1,092
		AnzFing:	905,0	905,5	911,5	919,3	926,6	939,0	949,8	957,2	961,5

Varationskoeffizienten - eingeschränkte Fälschung von Fingerprints

p_{random}	$p_{randomTick}$		1.Tick	2.Tick	10.Tick	20.Tick	30.Tick	50.Tick	70.Tick	90.Tick	100.Tick
100%	100%	AnzKorr:	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
		AnzFing:	nan%	0,1609%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
100%	10%	AnzKorr:	0,0%	0,973%	0,6656%	0,5211%	0,4769%	0,3487%	0,2438%	0,19%	0,1506%
		AnzFing:	nan%	nan%	30,7207%	4,5292%	2,6173%	0,7009%	0,1687%	0,0981%	0,0458%
100%	1%	AnzKorr:	0,0%	0,8745%	1,6421%	1,5035%	0,9495%	0,9034%	0,8984%	0,8048%	0,7177%
		AnzFing:	nan%	nan%	nan%	nan%	nan%	300,0%	65,4654%	27,8887%	21,4021%
10%	100%	AnzKorr:	0,0%	1,2502%	2,6951%	2,9289%	3,0202%	3,078%	3,1211%	3,1294%	3,1367%
		AnzFing:	5,9103%	0,04%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
10%	10%	AnzKorr:	0,0%	1,1072%	2,3195%	2,722%	2,7616%	2,5792%	2,5225%	2,8521%	2,8113%
		AnzFing:	3,361%	2,8283%	2,0535%	0,9547%	0,7753%	0,2012%	0,0664%	0,0%	0,0%
10%	1%	AnzKorr:	0,0%	0,1936%	0,7251%	1,0406%	1,1893%	1,3692%	1,7456%	1,7317%	1,7578%
		AnzFing:	4,7012%	4,5942%	5,3322%	4,3514%	4,0916%	3,3004%	2,4792%	2,0544%	2,1415%
1%	100%	AnzKorr:	0,0%	0,8095%	4,2607%	5,9404%	6,7666%	7,5238%	7,9332%	8,1871%	8,2799%
		AnzFing:	0,9828%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
1%	10%	AnzKorr:	0,0%	0,296%	1,2654%	1,3442%	1,7504%	2,59%	2,6829%	3,2151%	3,6784%
		AnzFing:	0,8447%	0,6909%	0,6009%	0,3112%	0,234%	0,0458%	0,0%	0,0%	0,0%
1%	1%	AnzKorr:	0,0%	0,049%	0,2474%	0,6164%	0,7279%	0,9995%	1,073%	1,5938%	1,7865%
		AnzFing:	1,2125%	1,2211%	1,1039%	0,973%	0,832%	0,6005%	0,7748%	0,7005%	0,6824%

A.9. Detaillierte Tabellen zum Fälschen von zufälligen Fingerprints

Werte ohne Hops - Fälschen von zufälligen Fingerprints

p_{fake}	p_{visit}	p_{fixed}		1.Tick	2.Tick	10.Tick	20.Tick	30.Tick	50.Tick	70.Tick	90.Tick	100.Tick
100%	0,01%	90%	fullsizeAnon:	9,4552	9,8834	15,1066	18,0419	21,3274	24,5958	27,3448	29,1555	30,3356
			fullsizeForgedAnon:	9,4552	9,8834	15,1066	18,0419	21,3274	24,5958	27,3448	29,1555	30,3356
			numFingerprints:	1,0	1,0014	1,0122	1,0228	1,0341	1,0524	1,0696	1,0859	1,0938
			percentTouched:	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%
			detectableFixed:	34,8893%	34,9497%	35,4315%	35,8734%	36,2724%	36,8818%	37,3761%	37,7952%	37,9815%
			numChanges:	0,0	0,0014	0,013	0,0255	0,0403	0,0671	0,0938	0,1222	0,1362
			sizeForgedAnon:	9,4552	9,359	10,6894	10,6462	10,9319	10,9426	10,6212	12,0154	11,0116
			anon1:	1156,1	1083,8	1181,3	1107,0	1097,1	1068,6	1020,6	1386,0	1116,9
			anon2:	396,1	438,7	605,4	577,0	618,4	678,2	634,1	562,8	624,8
			anon3:	385,3	413,3	443,4	511,8	515,4	550,1	522,8	446,6	534,6
			anon4:	378,7	400,6	379,1	465,2	431,5	455,9	432,9	415,6	455,5
			anon5:	372,8	368,0	367,7	383,2	367,9	387,5	390,3	384,7	394,8
			anon6:	366,6	360,6	349,4	344,0	335,7	356,3	334,0	358,8	349,3
			anon7:	362,9	354,8	325,2	327,8	312,4	328,3	323,5	329,5	324,8
			anon8:	357,7	350,8	304,2	306,5	299,2	306,1	313,6	298,9	313,1
			anon9:	353,5	346,8	298,5	287,9	291,4	291,6	294,2	278,8	301,6
			anon10:	346,4	335,3	291,6	275,7	278,3	262,6	285,9	266,9	287,9
			Nutzer1:	84,1733%	84,154%	84,0252%	83,8882%	83,7674%	83,5362%	83,3348%	83,1406%	83,048%
			Nutzer2:	5,0878%	5,0857%	5,0712%	5,0665%	5,0622%	5,0695%	5,0814%	5,0963%	5,1037%
			Nutzer2-9:	9,6726%	9,6699%	9,6709%	9,7002%	9,7375%	9,8389%	9,9387%	10,0453%	10,1111%
			Nutzer10+:	6,1541%	6,1761%	6,3038%	6,4116%	6,4951%	6,6249%	6,7265%	6,8141%	6,8409%
			Entropie:	18,3388	18,3372	18,3246	18,3172	18,3098	18,3001	18,2929	18,2853	18,2831
1%	100%	0%	fullsizeAnon:	9,4933	18,9977	95,0604	189,3652	281,3252	462,9022	638,6942	813,8659	900,154
			fullsizeForgedAnon:	1,0832	1,1794	1,9503	2,9263	3,8628	5,763	7,6112	9,4961	10,4465
			numFingerprints:	1,0	2,0	9,999	19,9966	29,9915	49,9755	69,9517	89,921	99,9026
			percentTouched:	3,9834%	6,0517%	16,3531%	25,7986%	33,7202%	46,7128%	56,8856%	65,0252%	68,4856%
			detectableFixed:	0,0%	7,2161%	99,8926%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%
			numChanges:	0,0	1,0	8,9999	18,9997	28,9994	48,999	68,9985	88,9981	98,998
			sizeForgedAnon:	9,4933	10,5291	10,3531	10,5896	10,4331	10,4254	10,2718	9,9178	10,2356
			anon1:	1150,2	1149,2	1150,0	1149,9	1149,5	1150,6	1150,1	1148,8	1149,8
			anon2:	399,5	401,4	399,3	400,2	400,9	399,8	399,0	399,5	399,9
			anon3:	392,1	392,1	391,9	392,0	392,4	392,2	391,3	391,8	392,1
			anon4:	383,9	384,0	384,2	384,0	383,9	383,9	383,6	384,1	384,2
			anon5:	377,8	378,1	377,2	377,6	377,4	376,8	377,6	376,8	377,4
			anon6:	371,8	371,3	371,6	371,5	371,8	370,9	371,5	371,1	370,3
			anon7:	365,4	367,1	366,0	365,9	365,8	365,6	365,8	365,7	365,7
			anon8:	359,5	359,5	359,5	359,0	359,0	359,5	359,2	359,1	359,1
			anon9:	355,2	354,6	354,8	355,7	355,5	354,0	355,2	355,2	354,9
			anon10:	349,9	349,6	349,1	349,7	349,8	349,7	349,5	349,4	350,1
			Nutzer1:	84,1473%	82,5245%	82,5109%	82,5081%	82,5086%	82,5079%	82,5105%	82,5075%	82,5049%
			Nutzer2:	5,126%	6,6191%	6,6248%	6,6334%	6,6333%	6,6325%	6,6261%	6,6307%	6,6435%
			Nutzer2-9:	9,6635%	11,2805%	11,2919%	11,2988%	11,2967%	11,3019%	11,2988%	11,3009%	11,3034%
			Nutzer10+:	6,1892%	6,195%	6,1972%	6,1931%	6,1946%	6,1903%	6,1906%	6,1917%	6,1916%
			Entropie:	18,3374	18,32	18,3199	18,3198	18,3199	18,3199	18,3199	18,32	18,32
1%	1%	0%	fullsizeAnon:	9,4277	19,7982	103,76	206,7473	309,4297	512,6003	714,7001	910,2237	1005,88
			fullsizeForgedAnon:	1,084	2,157	10,8368	21,652	32,4685	54,0417	75,4861	96,8282	107,4084
			numFingerprints:	1,0	2,0	9,9989	19,9955	29,9906	49,9747	69,951	89,9192	99,9015
			percentTouched:	3,9333%	5,3295%	12,5813%	19,3236%	25,1252%	35,0381%	43,3818%	50,5182%	53,7219%
			detectableFixed:	0,0%	6,9024%	99,9277%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%
			numChanges:	0,0	1,0	8,9997	18,9994	28,9991	48,9986	68,9983	88,9975	98,9973
			sizeForgedAnon:	9,4277	11,3706	12,2572	11,6756	11,4715	11,4919	11,7203	11,8332	11,4527
			anon1:	1163,2	1161,9	1165,4	1163,4	1162,2	1164,4	1163,8	1164,6	1163,3
			anon2:	394,7	393,8	393,6	395,5	394,6	393,7	394,4	394,9	393,7
			anon3:	383,8	384,9	384,6	385,3	384,1	383,8	384,3	384,7	383,9
			anon4:	377,4	379,1	380,0	377,8	377,7	377,9	377,6	377,0	378,6
			anon5:	372,8	373,1	372,8	371,4	372,7	371,8	371,9	372,4	371,7
			anon6:	367,9	368,3	368,7	367,8	367,2	368,7	367,9	368,0	368,3
			anon7:	364,1	363,2	364,5	363,9	365,0	363,0	363,7	364,3	363,7
			anon8:	360,0	359,3	360,4	361,1	362,3	360,6	359,7	360,4	360,2
			anon9:	355,9	356,3	356,2	356,2	355,4	357,0	355,9	355,5	356,7
			anon10:	347,7	348,1	348,9	348,6	348,0	346,8	349,2	346,8	348,3
			Nutzer1:	84,1515%	82,8371%	82,827%	82,8282%	82,83%	82,8291%	82,8268%	82,8284%	82,8277%
			Nutzer2:	5,0945%	5,581%	5,5845%	5,5823%	5,5823%	5,5813%	5,5884%	5,5789%	5,5852%
			Nutzer2-9:	9,6852%	10,9945%	11,0079%	11,0016%	10,9998%	11,0046%	11,0055%	11,0045%	11,0048%
			Nutzer10+:	6,1633%	6,1684%	6,1651%	6,1702%	6,1702%	6,1663%	6,1677%	6,1671%	6,1675%
			Entropie:	18,3378	18,3174	18,3173	18,3173	18,3172	18,3173	18,3173	18,3173	18,3173
1%	1%	10%	fullsizeAnon:	9,2273	19,3154	102,0084	202,7316	300,3329	485,9759	661,2888	830,9771	914,5335
			fullsizeForgedAnon:	1,0772	2,1429	10,7463	21,4795	32,1121	53,3185	74,2368	95,0513	105,4753
			numFingerprints:	1,0	1,9925	9,9206	19,8104	29,6808	49,3694	68,9862	88,5357	98,2843
			percentTouched:	3,9743%	5,3329%	12,5915%	19,3359%	25,1056%	34,9852%	43,3148%	50,4379%	53,6276%
			detectableFixed:	0,0%	9,4616%	99,5006%	99,7904%	99,8477%	99,8837%	99,8964%	99,9028%	99,9049%
			numChanges:	0,0	0,9925	8,927	18,8453	28,7631	48,5983	68,4354	88,2704	98,1887
			sizeForgedAnon:	9,2273	11,1358	11,2106	11,6782	12,0836	11,6296	11,0702	12,068	11,7878
			anon1:	1155,7	1155,2	1153,7	1155,4	1158,1	1155,6	1154,1	1156,7	1156,0
			anon2:	396,0	396,1	396,1	398,2	394,5	396,5	396,2	397,4	397,6
			anon3:	384,4	383,5	384,2	385,4	385,2	382,9	383,9	384,5	384,6
			anon4:	379,2	378,3	379,2	379,2	380,9	377,7	380,1	378,8	379,7
			anon5:	372,0	372,6	372,3	372,4	372,0	373,5	373,0	372,8	372,6
			anon6:	367,1	366,6	368,0	366,6	367,7	367,2	366,9	368,9	367,0
			anon7:	362,5	362,0	363,3	363,3	363,0	362,7	363,0	363,8	363,5
			anon8:	356,0	355,5	357,5	356,1	357,3	356,7	356,0	357,9	355,9
			anon9:	350,5	350,7	350,1	351,1	350,2	351,4	351,7	351,3	350,8
			anon10:	342,8	344,7	344,0	343,6	343,0	343,4	342,3	343,2	343,9
			Nutzer1:	84,1789%	82,8681%	82,8483%	82,8472%	82,8476%	82,848%	82,848%	82,8407%	82,8447%
			Nutzer2:	5,0686%	5,5527%	5,5677%	5,5722%	5,5623%	5,5632%	5,563%	5,5841%	5,58%
			Nutzer2-9:	9,6442%	10,9483%	10,9671%	10,9653%	10,9691%	10,9651%	10,9651%	10,9733%	10,9665%
			Nutzer10+:	6,1769%	6,1836%	6,1846%	6,1875%	6,1833%	6,1869%	6,1869%	6,1861%	6,1888%
			Entropie:	18,3379	18,3176	18,3173	18,3172	18,3173	18,3172	18,3173	18,3173	18,3172

p_{fake}	p_{visit}	p_{fixed}		1.Tick	2.Tick	10.Tick	20.Tick	30.Tick	50.Tick	70.Tick	90.Tick	100.Tick
1%	1%	50%	fullsizeAnon:	9,5222	17,4605	69,3514	115,2331	151,2566	208,748	254,6004	294,7957	312,8181
			fullsizeForgedAnon:	1,0869	1,5034	4,7943	8,6497	12,2564	18,8904	24,9058	30,5626	33,2403
			numFingerprints:	1,0	1,555	5,8498	10,9922	15,9454	25,3757	34,2643	42,6957	46,7595
			percentTouched:	3,9067%	5,0344%	10,5852%	15,3757%	19,3977%	26,1839%	31,9309%	36,9229%	39,1943%
			detectableFixed:	0,0976%	11,6474%	67,3764%	73,8137%	76,6646%	79,7729%	81,3761%	82,464%	82,807%
			numChanges:	0,0	0,555	4,9509	10,4343	15,9183	26,893	37,8572	48,8186	54,3004
			sizeForgedAnon:	9,5222	9,8507	10,2808	10,85	10,9415	10,5223	10,8621	11,2485	10,2895
			anon1:	1147,7	1148,1	1146,3	1148,4	1148,6	1147,8	1149,4	1149,7	1147,3
			anon2:	394,1	392,8	393,6	394,9	393,8	395,2	395,3	394,6	393,1
			anon3:	383,4	382,7	384,3	383,2	383,0	381,7	382,2	382,9	382,4
			anon4:	378,5	377,0	378,0	377,9	378,8	378,1	377,2	379,1	377,0
			anon5:	372,5	372,4	373,2	372,0	372,4	372,8	372,9	372,9	371,0
			anon6:	368,6	367,6	368,8	368,9	369,0	368,6	367,8	368,8	367,7
			anon7:	365,6	363,7	364,2	365,0	365,5	365,3	364,6	363,7	364,5
			anon8:	360,7	360,1	360,3	359,7	361,2	359,3	359,3	358,7	360,2
			anon9:	354,6	353,8	354,5	355,3	354,8	354,4	354,5	354,7	356,1
			anon10:	348,1	346,8	348,1	348,2	347,9	347,2	347,2	348,6	347,8
			Nutzer1:	84,135%	83,4213%	83,0866%	83,0016%	82,9548%	82,9061%	82,8804%	82,8623%	82,8554%
			Nutzer2:	5,0963%	5,493%	5,7935%	5,8679%	5,9181%	5,9626%	5,9872%	6,0113%	6,0164%
			Nutzer2-9:	9,7058%	10,4081%	10,7423%	10,8241%	10,8735%	10,9255%	10,9478%	10,9641%	10,9747%
			Nutzer10+:	6,1591%	6,1705%	6,1711%	6,1743%	6,1717%	6,1684%	6,1718%	6,1736%	6,1699%
			Entropie:	18,338	18,3282	18,3245	18,3235	18,323	18,3227	18,3223	18,3221	18,3221
1%	1%	90%	fullsizeAnon:	9,4686	12,363	18,5141	20,461	21,4575	22,6741	23,3278	23,7891	23,9876
			fullsizeForgedAnon:	1,0841	1,1408	1,3082	1,3777	1,4172	1,4651	1,4938	1,5144	1,5226
			numFingerprints:	1,0	1,0372	1,247	1,444	1,6011	1,8626	2,0699	2,2419	2,3192
			percentTouched:	3,9463%	4,2735%	5,4372%	6,1024%	6,509%	7,0702%	7,436%	7,7115%	7,8268%
			detectableFixed:	34,9309%	37,0481%	43,748%	46,8313%	48,5655%	50,8754%	52,2701%	53,268%	53,6216%
			numChanges:	0,0	0,0372	0,3296	0,6943	1,0502	1,7646	2,4767	3,1857	3,542
			sizeForgedAnon:	9,4686	9,9141	9,8027	9,9959	9,9951	9,9157	9,9898	9,9155	9,728
			anon1:	1160,1	1161,9	1160,6	1161,8	1162,2	1161,8	1162,0	1161,6	1160,5
			anon2:	392,1	393,1	393,6	392,8	392,3	391,9	392,5	392,3	392,7
			anon3:	382,5	383,6	384,2	383,2	382,7	383,1	383,0	383,1	383,6
			anon4:	376,2	375,8	375,8	375,5	376,2	376,2	375,8	376,0	375,8
			anon5:	372,4	371,9	371,4	371,7	371,3	371,6	371,6	371,3	371,8
			anon6:	368,0	368,4	368,1	368,3	368,2	367,7	367,8	368,3	368,4
			anon7:	364,4	364,2	364,0	364,4	363,7	364,4	363,9	363,8	363,9
			anon8:	358,1	357,8	358,0	358,4	358,0	357,5	358,5	358,0	358,5
			anon9:	352,6	353,1	353,6	353,1	353,5	352,6	352,5	352,9	353,2
			anon10:	346,5	345,6	346,2	346,2	345,7	345,8	347,0	345,8	346,1
			Nutzer1:	84,1905%	84,173%	84,1179%	84,0796%	84,056%	84,0219%	83,9985%	83,9848%	83,978%
			Nutzer2:	5,0885%	5,0998%	5,1419%	5,1738%	5,1918%	5,2196%	5,24%	5,2513%	5,2614%
			Nutzer2-9:	9,6726%	9,6888%	9,7441%	9,7807%	9,805%	9,8382%	9,8645%	9,8767%	9,8824%
			Nutzer10+:	6,137%	6,1382%	6,138%	6,1396%	6,139%	6,1399%	6,137%	6,1385%	6,1396%
			Entropie:	18,3393	18,3391	18,3384	18,3379	18,3376	18,3373	18,337	18,3369	18,3368
1%	0,01%	0%	fullsizeAnon:	9,172	120,4041	978,9176	1833,067	2539,3373	3606,0572	4342,878	4920,429	5140,9753
			fullsizeForgedAnon:	1,0789	99,9876	870,9818	1625,0521	2239,8114	3136,33	3700,984	4065,0975	4199,7325
			numFingerprints:	1,0	2,0	9,9973	19,9922	29,9836	49,9657	69,9395	89,8959	99,8739
			percentTouched:	3,9009%	3,9255%	4,0999%	4,3613%	4,5916%	5,0059%	5,4067%	5,8049%	5,9813%
			detectableFixed:	0,0%	6,9618%	99,9067%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%
			numChanges:	0,0	1,0	8,9984	18,9962	28,9933	48,9887	68,9833	88,9781	98,9751
			sizeForgedAnon:	9,172	112,2336	122,4392	102,426	106,9603	118,4316	109,3939	120,9457	109,6289
			anon1:	1172,8	1180,3	1181,5	1167,8	1192,3	1193,0	1172,5	1203,3	1171,2
			anon2:	395,9	419,9	408,5	403,7	410,9	403,8	420,1	428,9	410,9
			anon3:	387,8	396,5	387,2	386,1	386,7	391,5	385,0	393,9	387,5
			anon4:	379,3	378,9	377,2	376,8	379,0	376,7	377,4	381,4	378,0
			anon5:	374,5	372,3	371,0	371,6	371,4	373,7	373,4	375,1	371,6
			anon6:	368,6	366,3	365,1	366,2	365,9	365,2	365,2	367,1	366,7
			anon7:	362,3	359,7	358,6	360,3	359,4	358,6	358,6	361,3	358,9
			anon8:	357,4	354,4	355,1	354,4	353,7	353,4	353,4	356,4	353,4
			anon9:	353,9	349,9	349,8	349,8	350,2	350,7	349,8	351,5	349,8
			anon10:	349,3	346,7	345,3	345,3	345,7	345,5	345,3	346,0	345,3
			Nutzer1:	84,1716%	83,3684%	83,3686%	83,3676%	83,3674%	83,3684%	83,3682%	83,3681%	83,3681%
			Nutzer2:	5,1048%	5,0327%	5,0332%	5,033%	5,033%	5,0327%	5,0328%	5,0331%	5,033%
			Nutzer2-9:	9,67%	9,5322%	9,5338%	9,5332%	9,5327%	9,5334%	9,5322%	9,5338%	9,5327%
			Nutzer10+:	6,1583%	7,0994%	7,0975%	7,0992%	7,0999%	7,0982%	7,0996%	7,0981%	7,0991%
			Entropie:	18,338	18,2761	18,2749	18,277	18,2771	18,2757	18,2763	18,2755	18,2762
1%	0,01%	10%	fullsizeAnon:	9,7169	97,2893	753,9224	1420,3383	1985,0469	2833,5312	3450,5603	3923,3062	4106,2608
			fullsizeForgedAnon:	1,0875	78,5469	658,5654	1237,4461	1711,1609	2401,2038	2865,4965	3186,8931	3308,9446
			numFingerprints:	1,0	1,8802	8,859	17,6162	26,3547	43,8239	61,2677	78,6717	87,3843
			percentTouched:	3,9868%	4,0153%	4,1958%	4,4129%	4,6281%	5,0225%	5,4228%	5,8163%	5,9843%
			detectableFixed:	0,0%	8,086%	92,9566%	95,8402%	96,8229%	97,7781%	98,181%	98,4985%	98,6011%
			numChanges:	0,0	0,8802	7,8613	16,6326	25,39	42,9265	60,4573	77,9747	86,7442
			sizeForgedAnon:	9,7169	88,6763	99,6809	87,7988	92,8153	89,4197	91,8793	90,6944	83,9825
			anon1:	1173,3	1170,5	1191,7	1161,6	1182,5	1182,0	1202,2	1161,6	1161,4
			anon2:	397,0	425,1	427,0	392,4	416,3	424,4	406,1	463,6	420,5
			anon3:	385,4	395,8	383,0	380,6	385,1	388,6	389,0	387,4	382,4
			anon4:	378,8	377,4	377,7	375,4	375,9	376,6	376,2	377,6	376,6
			anon5:	373,2	371,9	371,8	368,6	370,6	369,7	369,1	372,3	369,1
			anon6:	368,2	366,9	365,4	365,1	366,0	366,4	365,8	367,6	366,1
			anon7:	364,0	361,5	363,8	361,2	361,8	361,5	362,4	362,9	361,2
			anon8:	361,3	358,1	358,1	357,6	358,0	357,8	359,5	359,2	357,6
			anon9:	356,2	352,9	352,7	352,7	352,7	354,1	353,0	352,8	352,7
			anon10:	351,2	348,4	348,2	348,2	348,7	348,6	348,2	348,4	348,2
			Nutzer1:	84,1585%	83,4742%	83,3647%	83,339%	83,3307%	83,3218%	83,3164%	83,3144%	83,3139%
			Nutzer2:	5,0977%	5,0294%	5,0562%	5,076%	5,0826%	5,0914%	5,0986%	5,0985%	5,0986%
			Nutzer2-9:	9,6771%	9,5476%	9,6497%	9,6766%	9,6834%	9,6908%	9,6926%	9,6981%	9,6966%
			Nutzer10+:	6,1645%	6,9782%	6,9856%	6,9844%	6,9859%	6,9874%	6,991%	6,9874%	6,9895%
			Entropie:	18,3374	18,2858	18,2826	18,283	18,2828	18,2832	18,2831	18,2827	18,2832

p_{fake}	p_{visit}	p_{fixed}		1.Tick	2.Tick	10.Tick	20.Tick	30.Tick	50.Tick	70.Tick	90.Tick	100.Tick
1%	0,01%	50%	fullsizeAnon:	9,46	18,3089	62,0817	108,471	142,2393	199,5212	243,3546	283,9899	301,3201
			fullsizeForgedAnon:	1,0831	3,6431	20,1812	35,9438	48,5004	68,065	82,902	94,9049	100,1545
			numFingerprints:	1,0	1,1312	2,1678	3,431	4,6953	7,216	9,6764	12,1084	13,3039
			percentTouched:	3,9813%	4,0027%	4,1903%	4,3903%	4,599%	5,015%	5,3984%	5,7893%	5,9722%
			detectableFixed:	0,0908%	1,9414%	20,2615%	27,9043%	32,5306%	38,3771%	42,0116%	44,811%	46,0089%
			numChanges:	0,0	0,1312	1,1762	2,4689	3,769	6,3859	8,9755	11,5862	12,8862
			sizeForgedAnon:	9,46	14,0303	14,7435	18,8468	9,7807	13,2888	12,3721	14,4917	14,489
			anon1:	1171,3	1181,0	1175,0	1194,0	1160,8	1171,2	1164,0	1176,1	1175,6
			anon2:	394,2	403,3	404,9	398,5	399,2	408,1	403,5	407,0	398,4
			anon3:	386,3	384,8	391,3	387,3	386,2	393,1	391,4	386,7	387,8
			anon4:	379,5	378,1	383,3	380,1	377,9	380,5	383,6	379,8	379,6
			anon5:	374,2	372,3	375,0	374,5	372,8	373,1	374,7	373,1	375,2
			anon6:	368,4	365,6	369,0	368,4	368,1	366,1	366,2	367,8	368,5
			anon7:	363,0	360,5	362,7	361,3	362,1	360,5	362,3	363,6	363,8
			anon8:	358,8	356,4	357,7	356,8	357,1	356,3	357,3	357,2	357,7
			anon9:	354,3	352,4	351,3	353,0	352,8	352,2	352,0	352,6	353,6
			anon10:	349,0	346,7	345,2	347,2	344,8	347,9	346,5	346,4	345,7
			Nutzer1:	84,1705%	84,084%	83,9108%	83,8191%	83,7568%	83,6749%	83,6197%	83,5825%	83,5652%
			Nutzer2:	5,1061%	5,0953%	5,1092%	5,1514%	5,192%	5,2461%	5,2919%	5,3194%	5,3344%
			Nutzer2-9:	9,6714%	9,6673%	9,822%	9,9106%	9,9658%	10,0484%	10,1059%	10,1384%	10,1579%
			Nutzer10+:	6,1581%	6,2487%	6,2672%	6,2703%	6,2774%	6,2767%	6,2744%	6,2791%	6,2768%
			Entropie:	18,3386	18,3344	18,3303	18,3288	18,3284	18,3273	18,3264	18,3261	18,3258
1%	0,01%	90%	fullsizeAnon:	9,1895	9,4421	11,3346	12,6527	13,6309	14,8178	15,7074	16,324	16,5563
			fullsizeForgedAnon:	1,0789	1,0863	1,1359	1,1687	1,1963	1,231	1,2546	1,2738	1,2808
			numFingerprints:	1,0	1,0017	1,0122	1,0248	1,0345	1,0544	1,0723	1,0897	1,0979
			percentTouched:	2,9677%	2,9802%	3,0673%	3,1828%	3,2456%	3,4013%	3,5437%	3,6588%	3,7196%
			detectableFixed:	34,887%	34,9594%	35,4192%	35,8611%	36,2415%	36,8714%	37,4263%	37,9239%	38,156%
			numChanges:	0,0	0,0017	0,0134	0,0282	0,0411	0,0696	0,0996	0,129	0,1433
			sizeForgedAnon:	9,1895	9,0042	9,3494	8,8718	8,8048	8,7894	10,0703	8,933	8,9045
			anon1:	1158,3	1157,1	1159,6	1156,9	1156,0	1156,2	1163,0	1157,4	1156,4
			anon2:	402,4	402,3	402,1	402,3	403,9	401,7	402,0	402,7	402,7
			anon3:	392,4	392,3	392,4	392,7	391,7	393,2	392,5	392,4	392,4
			anon4:	378,6	379,1	378,6	379,7	379,7	379,3	379,0	379,4	380,1
			anon5:	374,1	374,4	373,7	374,0	374,7	373,6	374,1	373,9	373,8
			anon6:	369,9	370,1	369,8	369,6	370,0	370,4	369,4	370,2	371,2
			anon7:	364,1	364,8	363,8	364,1	364,5	363,9	363,7	363,2	363,1
			anon8:	359,0	359,1	358,9	358,1	358,6	359,4	357,8	358,4	358,0
			anon9:	353,5	353,5	353,3	353,2	352,9	353,9	353,2	353,3	354,3
			anon10:	349,4	349,7	348,8	349,7	349,9	349,4	348,4	349,1	350,3
			Nutzer1:	84,1879%	84,1876%	84,1857%	84,1832%	84,1817%	84,1785%	84,1754%	84,1727%	84,1715%
			Nutzer2:	5,0969%	5,0972%	5,0982%	5,0999%	5,1006%	5,1027%	5,1051%	5,1061%	5,1069%
			Nutzer2-9:	9,6447%	9,6451%	9,6472%	9,6499%	9,6509%	9,6552%	9,657%	9,6598%	9,6602%
			Nutzer10+:	6,1673%	6,1673%	6,1671%	6,1669%	6,1674%	6,1663%	6,1676%	6,1675%	6,1682%
			Entropie:	18,3389	18,3389	18,3389	18,3389	18,3389	18,3388	18,3387	18,3387	18,3387
0,01%	100%	0%	fullsizeAnon:	6,4852	12,8499	101,1991	208,2339	306,0174	486,5471	651,4892	807,978	898,9499
			fullsizeForgedAnon:	1,0	1,0	1,0144	1,0242	1,0288	1,0379	1,0532	1,0754	1,0833
			numFingerprints:	1,0	2,0	9,9982	19,9961	29,9937	49,984	69,9603	89,9447	99,9236
			percentTouched:	0,0479%	0,1092%	0,771%	1,3084%	1,7707%	2,4602%	2,9659%	3,4712%	3,7039%
			detectableFixed:	0,0%	8,061%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%
			numChanges:	0,0	1,0	9,0	19,0	29,0	49,0	68,9977	88,9977	98,9977
			sizeForgedAnon:	6,4852	7,3647	8,2236	13,8699	7,0955	9,946	7,8273	10,3165	16,6781
			anon1:	1163,2	1163,2	1163,2	1163,4	1163,2	1163,3	1163,3	1163,2	1163,5
			anon2:	396,6	396,6	396,6	396,6	396,6	396,6	396,6	396,8	396,6
			anon3:	389,0	389,0	389,1	389,0	389,1	389,0	389,0	389,0	389,0
			anon4:	381,8	381,7	381,8	381,8	381,8	381,8	381,7	381,7	381,9
			anon5:	374,8	375,0	374,8	374,8	374,8	374,8	374,8	374,8	374,9
			anon6:	369,0	369,0	369,1	369,2	369,0	369,0	369,0	369,1	369,1
			anon7:	365,6	365,5	365,6	365,6	365,5	365,5	365,5	365,5	365,5
			anon8:	361,0	361,0	361,0	361,0	361,0	361,0	361,0	361,3	361,0
			anon9:	352,0	352,0	352,0	352,0	352,0	352,0	352,0	352,0	352,1
			anon10:	347,0	347,0	347,0	347,0	347,0	347,1	347,0	347,2	347,2
			Nutzer1:	84,166%	84,1507%	84,1505%	84,1509%	84,1506%	84,1509%	84,1504%	84,1505%	84,151%
			Nutzer2:	5,0984%	5,1122%	5,1132%	5,1117%	5,1127%	5,1114%	5,1131%	5,113%	5,1112%
			Nutzer2-9:	9,6679%	9,6832%	9,6833%	9,6831%	9,6835%	9,6822%	9,6836%	9,6834%	9,683%
			Nutzer10+:	6,1661%	6,1661%	6,1662%	6,166%	6,1659%	6,1668%	6,166%	6,1661%	6,166%
			Entropie:	18,3379	18,3377	18,3377	18,3377	18,3377	18,3377	18,3377	18,3377	18,3377
0,01%	100%	10%	fullsizeAnon:	5,0262	17,2675	85,5254	179,6769	264,792	424,185	558,1142	711,551	774,2049
			fullsizeForgedAnon:	1,0	1,0	1,0148	1,0239	1,0278	1,0313	1,0487	1,0528	1,0577
			numFingerprints:	1,0	2,0	9,9717	19,9332	29,8656	49,6284	69,2821	88,8917	98,6649
			percentTouched:	0,0384%	0,1536%	0,7283%	1,2711%	1,7347%	2,5145%	3,0652%	3,565%	3,815%
			detectableFixed:	0,0%	7,0762%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%
			numChanges:	0,0	1,0	8,9918	18,9918	28,9877	48,9792	68,9682	88,9572	98,953
			sizeForgedAnon:	5,0262	13,2413	8,2195	7,9917	5,607	13,7422	11,5093	12,3611	8,3077
			anon1:	1164,0	1164,1	1164,0	1164,0	1164,0	1164,2	1164,2	1164,2	1164,1
			anon2:	398,2	398,2	398,2	398,2	398,2	398,3	398,2	398,4	398,3
			anon3:	385,6	385,6	385,5	385,7	385,5	385,5	385,6	385,5	385,5
			anon4:	380,5	380,8	380,6	380,5	380,6	380,5	380,6	380,5	380,5
			anon5:	378,7	378,7	378,7	378,7	378,7	378,7	378,7	378,7	378,7
			anon6:	368,6	368,6	368,7	368,6	368,6	368,7	368,6	368,6	368,6
			anon7:	365,7	365,7	365,7	365,7	365,7	365,7	365,7	365,7	365,7
			anon8:	361,0	361,0	361,0	361,0	361,0	361,0	361,0	361,0	361,0
			anon9:	356,8	356,8	357,0	357,0	356,8	356,8	356,8	356,9	356,8
			anon10:	352,5	352,4	352,4	352,4	352,4	352,5	352,4	352,4	352,4
			Nutzer1:	84,1896%	84,1744%	84,174%	84,174%	84,1739%	84,1738%	84,174%	84,1739%	84,174%
			Nutzer2:	5,0753%	5,0888%	5,0897%	5,0902%	5,09%	5,0904%	5,0896%	5,0902%	5,09%
			Nutzer2-9:	9,6316%	9,6461%	9,6468%	9,6463%	9,647%	9,6469%	9,6466%	9,6469%	9,6471%
			Nutzer10+:	6,1788%	6,1795%	6,1792%	6,1797%	6,1791%	6,1793%	6,1795%	6,1792%	6,1789%
			Entropie:	18,3378	18,3376	18,3377	18,3376	18,3377	18,3377	18,3377	18,3376	18,3377

p_{fake}	p_{visit}	p_{fixed}		1.Tick	2.Tick	10.Tick	20.Tick	30.Tick	50.Tick	70.Tick	90.Tick	100.Tick
0,01%	100%	50%	fullsizeAnon:	8,5532	13,2997	52,6474	89,5071	122,6261	165,6667	203,0547	239,4279	253,4155
			fullsizeForgedAnon:	1,0	1,0039	1,0039	1,0121	1,0204	1,0204	1,0289	1,0484	1,0484
			numFingerprints:	1,0	1,8285	7,3964	13,2795	18,715	28,3103	36,9787	44,9527	48,7951
			percentTouched:	0,0821%	0,1245%	0,477%	0,7863%	1,0513%	1,392%	1,6915%	1,9628%	2,0789%
			detectableFixed:	0,2174%	21,6598%	83,7008%	83,9136%	83,9136%	83,9136%	83,9136%	83,9136%	83,9136%
			numChanges:	0,0	0,8285	6,9001	14,4818	22,0601	37,0875	52,1751	67,2554	74,8454
			sizeForgedAnon:	8,5532	6,4455	8,9286	5,6088	8,5176	11,0423	7,659	9,6397	11,9405
			anon1:	1146,3	1146,2	1146,3	1146,2	1146,3	1146,5	1146,3	1146,3	1146,4
			anon2:	395,6	395,6	395,7	395,6	395,6	395,6	395,6	395,7	395,6
			anon3:	388,8	388,8	388,8	388,8	388,9	388,8	388,8	388,9	388,8
			anon4:	381,4	381,3	381,3	381,3	381,3	381,3	381,3	381,3	381,3
			anon5:	375,0	375,1	375,1	375,0	375,0	375,0	375,0	375,1	375,0
			anon6:	371,8	371,8	371,9	371,8	371,8	371,8	371,8	371,9	371,8
			anon7:	365,8	365,7	365,6	365,7	365,6	365,6	365,6	365,7	365,7
			anon8:	361,2	361,2	361,2	361,2	361,3	361,2	361,2	361,2	361,2
			anon9:	357,3	357,3	357,3	357,3	357,3	357,3	357,3	357,3	357,4
			anon10:	352,9	353,0	352,8	352,8	352,8	352,8	352,8	352,9	352,8
			Nutzer1:	84,1644%	84,1512%	84,1505%	84,1508%	84,1505%	84,1507%	84,1507%	84,1504%	84,1508%
			Nutzer2:	5,1114%	5,1238%	5,1242%	5,1235%	5,1244%	5,124%	5,1236%	5,1246%	5,1236%
			Nutzer2-9:	9,6752%	9,6888%	9,689%	9,6887%	9,6892%	9,689%	9,689%	9,6894%	9,6888%
			Nutzer10+:	6,1603%	6,1601%	6,1605%	6,1605%	6,1603%	6,1602%	6,1603%	6,1602%	6,1604%
			Entropie:	18,3383	18,3381	18,3381	18,3381	18,3381	18,3381	18,3381	18,3381	18,3381
0,01%	100%	90%	fullsizeAnon:	6,9089	8,2848	11,1928	12,1587	12,7244	13,1812	13,425	13,5081	13,5864
			fullsizeForgedAnon:	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
			numFingerprints:	1,0	1,1214	1,5577	1,7917	1,9483	2,1111	2,2261	2,2941	2,3273
			percentTouched:	0,061%	0,0744%	0,1047%	0,1148%	0,1207%	0,1252%	0,1277%	0,1287%	0,1295%
			detectableFixed:	36,6868%	44,7596%	55,1215%	55,1215%	55,1215%	55,1215%	55,1215%	55,1215%	55,1215%
			numChanges:	0,0	0,1214	0,8347	1,6854	2,5498	4,3071	5,9741	7,639	8,4905
			sizeForgedAnon:	6,9089	7,0908	6,8626	7,1117	7,064	6,8752	6,5302	6,9242	6,7824
			anon1:	1145,4	1145,4	1145,4	1145,4	1145,4	1145,4	1145,4	1145,4	1145,4
			anon2:	398,9	398,9	398,9	399,0	398,9	398,9	398,9	398,9	398,9
			anon3:	386,9	386,9	386,9	386,9	386,9	386,9	386,9	386,9	386,9
			anon4:	381,8	381,8	381,8	381,8	381,8	381,8	381,8	381,8	381,8
			anon5:	376,6	376,6	376,6	376,6	376,6	376,6	376,6	376,6	376,6
			anon6:	370,1	370,1	370,1	370,1	370,1	370,1	370,1	370,1	370,1
			anon7:	365,6	365,6	365,6	365,6	365,6	365,6	365,6	365,6	365,6
			anon8:	360,1	360,1	360,1	360,1	360,1	360,1	360,1	360,1	360,1
			anon9:	354,3	354,3	354,3	354,3	354,3	354,3	354,3	354,3	354,3
			anon10:	344,7	344,7	344,7	344,7	344,7	344,7	344,7	344,7	344,7
			Nutzer1:	84,2129%	84,2112%	84,2103%	84,2103%	84,2103%	84,2103%	84,2104%	84,2103%	84,2103%
			Nutzer2:	5,079%	5,0803%	5,0814%	5,0813%	5,0812%	5,0812%	5,0811%	5,0814%	5,0813%
			Nutzer2-9:	9,636%	9,6377%	9,6386%	9,6384%	9,6384%	9,6382%	9,6386%	9,6386%	9,6385%
			Nutzer10+:	6,1511%	6,1511%	6,1511%	6,1513%	6,1512%	6,1515%	6,151%	6,1511%	6,1511%
			Entropie:	18,3398	18,3397	18,3397	18,3397	18,3397	18,3397	18,3397	18,3397	18,3397
0,01%	1%	0%	fullsizeAnon:	10,2108	23,6749	96,1525	197,6053	299,5258	465,1314	644,3465	830,5428	941,5305
			fullsizeForgedAnon:	1,0042	1,0157	1,0835	1,1544	1,298	1,5557	1,7916	2,0352	2,1229
			numFingerprints:	1,0	2,0	9,9981	19,9981	29,9957	49,9877	69,9697	89,947	99,9248
			percentTouched:	0,0851%	0,2076%	0,7924%	1,3362%	1,8025%	2,606%	3,2184%	3,7279%	3,983%
			detectableFixed:	0,0%	8,668%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%
			numChanges:	0,0	1,0	9,0	19,0	29,0	49,0	69,0	89,0	98,9979
			sizeForgedAnon:	10,2108	14,4642	9,4225	11,4338	9,5785	16,5016	7,7367	10,3191	14,097
			anon1:	1167,6	1167,9	1167,6	1167,6	1167,6	1167,8	1167,6	1167,6	1167,7
			anon2:	398,1	398,1	398,1	398,1	398,2	398,1	398,1	398,1	398,2
			anon3:	384,7	384,8	384,7	384,8	384,8	384,8	384,7	384,7	384,7
			anon4:	380,3	380,3	380,3	380,3	380,3	380,3	380,3	380,3	380,6
			anon5:	374,0	374,0	374,0	374,0	374,0	374,1	374,1	374,2	374,0
			anon6:	369,2	369,0	369,1	369,0	369,0	369,0	369,0	369,0	369,1
			anon7:	366,3	366,1	366,2	366,3	366,1	366,2	366,1	366,1	366,1
			anon8:	361,6	361,5	361,5	361,5	361,5	361,7	361,5	361,5	361,6
			anon9:	355,6	355,7	355,6	355,9	355,6	355,6	355,6	355,7	355,7
			anon10:	348,9	348,9	349,0	348,9	348,9	348,9	348,9	349,0	348,9
			Nutzer1:	84,1563%	84,1393%	84,1395%	84,1393%	84,1396%	84,1394%	84,1392%	84,1394%	84,1394%
			Nutzer2:	5,1067%	5,1225%	5,1219%	5,1229%	5,1214%	5,1222%	5,1227%	5,1223%	5,1223%
			Nutzer2-9:	9,6982%	9,7153%	9,7148%	9,7148%	9,7145%	9,7146%	9,7153%	9,715%	9,7151%
			Nutzer10+:	6,1455%	6,1454%	6,1457%	6,1459%	6,1458%	6,146%	6,1455%	6,1456%	6,1455%
			Entropie:	18,3384	18,3382	18,3382	18,3382	18,3382	18,3382	18,3383	18,3382	18,3382
0,01%	1%	10%	fullsizeAnon:	12,2476	19,0509	91,1115	171,5567	259,8295	403,0325	560,1036	703,2397	784,5662
			fullsizeForgedAnon:	1,0	1,0156	1,1256	1,2153	1,2989	1,548	1,7434	1,9703	2,0855
			numFingerprints:	1,0	1,9886	9,8857	19,7387	29,5688	49,1399	68,6536	88,0779	97,7904
			percentTouched:	0,1255%	0,2023%	0,8157%	1,3759%	1,7959%	2,5396%	3,1954%	3,7796%	4,0677%
			detectableFixed:	0,0%	12,0769%	98,9634%	99,6726%	99,8077%	99,8077%	99,8077%	100,0%	100,0%
			numChanges:	0,0	0,9886	8,8957	18,7719	28,6625	48,4098	68,1607	87,9053	97,78
			sizeForgedAnon:	12,2476	7,8034	8,3539	8,781	5,3285	7,8085	8,3008	10,8948	13,3595
			anon1:	1161,8	1161,6	1161,7	1161,6	1161,6	1161,6	1161,7	1161,7	1161,8
			anon2:	393,6	393,6	393,5	393,6	393,5	393,5	393,5	393,5	393,5
			anon3:	384,2	384,0	384,0	384,0	384,0	384,0	384,1	384,0	384,0
			anon4:	378,5	378,3	378,3	378,3	378,4	378,4	378,3	378,3	378,3
			anon5:	373,9	374,0	373,9	373,9	373,9	373,9	373,9	374,1	373,9
			anon6:	367,7	367,9	367,7	368,1	367,7	367,7	367,8	367,7	367,7
			anon7:	363,3	363,3	363,5	363,4	363,4	363,5	363,3	363,3	363,4
			anon8:	358,0	357,9	357,9	357,9	357,9	357,9	358,0	357,9	358,0
			anon9:	353,2	353,1	353,1	353,2	353,1	353,2	353,1	353,1	353,3
			anon10:	346,9	347,0	346,9	346,9	346,9	346,9	347,0	347,0	346,9
			Nutzer1:	84,1605%	84,1423%	84,1419%	84,142%	84,142%	84,1423%	84,1422%	84,1427%	84,1425%
			Nutzer2:	5,1083%	5,1256%	5,1262%	5,1258%	5,1257%	5,1253%	5,1254%	5,1242%	5,1248%
			Nutzer2-9:	9,6988%	9,7175%	9,7175%	9,7171%	9,7175%	9,717%	9,717%	9,7167%	9,7168%
			Nutzer10+:	6,1407%	6,1402%	6,1405%	6,1409%	6,1405%	6,1408%	6,1408%	6,1406%	6,1407%
			Entropie:	18,3386	18,3384	18,3384	18,3384	18,3384	18,3384	18,3384	18,3384	18,3384

p_{fake}	p_{visit}	p_{fixed}		1.Tick	2.Tick	10.Tick	20.Tick	30.Tick	50.Tick	70.Tick	90.Tick	100.Tick
0,01%	1%	50%	fullsizeAnon:	20,3553	37,2369	96,9467	147,1838	191,0589	253,9454	299,801	338,9718	356,2285
			fullsizeForgedAnon:	1,009	1,009	1,0372	1,0608	1,1104	1,1981	1,273	1,3286	1,3514
			numFingerprints:	1,0	1,5776	6,1194	11,4813	16,6124	26,3757	35,5822	44,3429	48,5533
			percentTouched:	0,1405%	0,2833%	0,7564%	1,1421%	1,4685%	1,8982%	2,2193%	2,4929%	2,614%
			detectableFixed:	0,0%	11,7853%	68,5243%	75,5625%	78,6804%	82,9129%	84,7414%	86,1172%	86,1172%
			numChanges:	0,0	0,5776	5,2544	11,0175	16,7911	28,334	39,8606	51,4614	57,1967
			sizeForgedAnon:	20,3553	19,5218	20,2344	9,7344	21,7135	19,7165	12,6643	13,5846	10,0291
			anon1:	1163,5	1163,6	1163,5	1163,3	1163,5	1163,6	1163,3	1163,3	1163,2
			anon2:	394,9	395,0	395,1	394,9	395,0	394,9	394,9	395,0	394,9
			anon3:	383,3	383,2	383,3	383,3	383,4	383,3	383,3	383,2	383,2
			anon4:	377,0	377,1	377,1	377,0	377,1	377,0	377,1	377,1	377,1
			anon5:	371,8	371,8	371,7	371,7	371,9	371,7	371,7	371,9	371,8
			anon6:	365,9	365,9	365,8	365,9	366,0	365,8	365,8	365,8	365,8
			anon7:	363,1	362,9	363,1	362,9	362,9	363,0	363,1	362,9	363,0
			anon8:	357,1	357,0	357,0	356,9	357,1	357,0	357,0	357,1	356,9
			anon9:	352,2	352,3	352,2	352,3	352,2	352,3	352,2	352,2	352,4
			anon10:	347,2	347,2	347,2	347,2	347,3	347,2	347,2	347,2	347,2
			Nutzer1:	84,1937%	84,1858%	84,1826%	84,1819%	84,1817%	84,1809%	84,1807%	84,1803%	84,1804%
			Nutzer2:	5,1134%	5,1202%	5,1234%	5,1237%	5,1235%	5,1245%	5,1249%	5,1255%	5,1252%
			Nutzer2-9:	9,6656%	9,6732%	9,6766%	9,677%	9,6773%	9,6777%	9,6781%	9,6785%	9,6788%
			Nutzer10+:	6,1407%	6,141%	6,1408%	6,1411%	6,141%	6,1414%	6,1412%	6,1412%	6,1408%
			Entropie:	18,3397	18,3396	18,3396	18,3396	18,3396	18,3396	18,3395	18,3395	18,3395
0,01%	1%	90%	fullsizeAnon:	9,1361	10,0396	12,077	14,2901	14,8022	15,7544	16,1799	16,7561	17,0182
			fullsizeForgedAnon:	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
			numFingerprints:	1,0	1,0401	1,2564	1,4565	1,6273	1,9308	2,1883	2,401	2,4848
			percentTouched:	0,0729%	0,0806%	0,0995%	0,1185%	0,123%	0,1315%	0,1353%	0,1403%	0,1425%
			detectableFixed:	34,7152%	37,243%	43,1041%	45,5618%	48,3461%	50,3528%	52,6477%	53,7497%	54,2161%
			numChanges:	0,0	0,0401	0,3141	0,632	0,9561	1,5731	2,2137	2,8313	3,1306
			sizeForgedAnon:	9,1361	6,2844	9,278	7,2434	8,9269	7,4463	9,7919	5,8477	9,7844
			anon1:	1181,7	1181,6	1181,7	1181,6	1181,7	1181,6	1181,7	1181,6	1181,7
			anon2:	393,3	393,3	393,3	393,3	393,3	393,3	393,3	393,3	393,3
			anon3:	382,5	382,5	382,5	382,5	382,5	382,5	382,5	382,5	382,5
			anon4:	378,4	378,4	378,4	378,4	378,4	378,4	378,4	378,4	378,4
			anon5:	371,4	371,4	371,4	371,4	371,4	371,4	371,4	371,4	371,4
			anon6:	368,6	368,6	368,5	368,7	368,5	368,7	368,6	368,5	368,6
			anon7:	361,0	361,0	361,0	361,0	361,0	361,0	361,0	361,0	361,0
			anon8:	356,2	356,2	356,2	356,2	356,2	356,2	356,2	356,2	356,2
			anon9:	351,2	351,2	351,2	351,2	351,2	351,2	351,2	351,2	351,2
			anon10:	346,0	346,0	346,0	346,0	346,0	346,0	346,0	346,0	346,0
			Nutzer1:	84,1811%	84,181%	84,1802%	84,18%	84,1797%	84,1794%	84,179%	84,179%	84,1789%
			Nutzer2:	5,1073%	5,1075%	5,1082%	5,1082%	5,1082%	5,1085%	5,1092%	5,1089%	5,1092%
			Nutzer2-9:	9,6645%	9,6647%	9,6654%	9,6657%	9,666%	9,6658%	9,6666%	9,6667%	9,6667%
			Nutzer10+:	6,1543%	6,1543%	6,1543%	6,1543%	6,1543%	6,1548%	6,1543%	6,1543%	6,1544%
			Entropie:	18,3391	18,3391	18,3391	18,3391	18,3391	18,3391	18,3391	18,3391	18,3391
0,01%	0,01%	0%	fullsizeAnon:	10,0191	13,05	85,3251	188,0219	290,6955	504,697	685,8271	881,745	968,7487
			fullsizeForgedAnon:	1,0	1,8547	9,2601	17,1146	23,1343	31,6424	37,0336	40,4777	41,713
			numFingerprints:	1,0	2,0	10,0	19,9982	29,9874	49,9676	69,9466	89,9176	99,9011
			percentTouched:	0,0929%	0,1145%	0,4891%	0,8956%	1,2368%	1,8281%	2,3035%	2,7313%	2,9092%
			detectableFixed:	0,0%	7,347%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%
			numChanges:	0,0	1,0	9,0	19,0	29,0	48,998	68,998	88,998	98,998
			sizeForgedAnon:	10,0191	4,0309	6,6501	10,4247	20,0694	10,1452	13,5067	14,9426	10,348
			anon1:	1190,0	1189,9	1189,9	1190,0	1190,2	1189,9	1190,1	1190,0	1189,9
			anon2:	395,8	395,7	395,7	395,7	395,7	395,7	395,7	395,7	396,0
			anon3:	385,2	385,1	385,1	385,1	385,4	385,1	385,1	385,4	385,1
			anon4:	381,1	381,1	381,1	381,1	381,1	381,2	381,1	381,1	381,1
			anon5:	377,5	377,3	377,3	377,3	377,3	377,3	377,3	377,3	377,4
			anon6:	369,2	369,1	369,1	369,1	369,3	369,1	369,1	369,2	369,3
			anon7:	363,9	363,8	363,8	363,8	363,9	363,8	363,8	363,8	363,8
			anon8:	352,4	352,4	352,4	352,5	352,5	352,4	352,4	352,4	352,4
			anon9:	349,0	349,0	349,1	349,0	349,1	349,0	349,0	349,1	349,0
			anon10:	342,0	342,0	342,0	342,1	342,0	342,1	342,0	342,2	342,0
			Nutzer1:	84,1812%	84,168%	84,1686%	84,1684%	84,1686%	84,1681%	84,1684%	84,1689%	84,1686%
			Nutzer2:	5,0842%	5,0898%	5,0883%	5,0889%	5,0886%	5,0902%	5,0889%	5,088%	5,0884%
			Nutzer2-9:	9,6644%	9,678%	9,6773%	9,6775%	9,6768%	9,6769%	9,677%	9,6764%	9,6767%
			Nutzer10+:	6,1544%	6,154%	6,1541%	6,1542%	6,1546%	6,155%	6,1546%	6,1547%	6,1547%
			Entropie:	18,3384	18,3382	18,3382	18,3382	18,3382	18,3382	18,3382	18,3382	18,3382
0,01%	0,01%	10%	fullsizeAnon:	7,9412	19,5405	95,4057	191,1544	288,2965	474,732	640,1549	784,2852	861,0393
			fullsizeForgedAnon:	1,0	1,8062	7,2807	12,8673	17,3427	23,989	28,2546	31,4506	32,5966
			numFingerprints:	1,0	1,8785	8,9642	17,8227	26,6717	44,2657	61,8366	79,3859	88,1237
			percentTouched:	0,0662%	0,1589%	0,5539%	0,9267%	1,2567%	1,8352%	2,2785%	2,6926%	2,872%
			detectableFixed:	0,0%	6,8779%	94,0415%	96,643%	96,8096%	97,8246%	98,2056%	98,2056%	98,2056%
			numChanges:	0,0	0,8785	7,9642	16,8293	25,6906	43,3759	61,0697	78,7656	87,5579
			sizeForgedAnon:	7,9412	12,8678	17,6629	9,8428	15,1857	13,2649	11,0253	11,0642	6,5775
			anon1:	1166,4	1166,5	1166,5	1166,4	1166,6	1166,7	1166,4	1166,3	1166,3
			anon2:	393,6	393,6	393,6	393,8	393,6	393,6	393,6	393,6	393,6
			anon3:	383,9	384,0	384,4	383,9	383,9	383,9	383,9	384,2	383,9
			anon4:	380,1	380,0	380,0	380,0	380,0	380,0	380,0	380,2	380,1
			anon5:	375,6	375,6	375,6	375,6	375,6	375,6	375,6	375,7	375,6
			anon6:	370,5	370,5	370,5	370,5	370,5	370,5	370,5	370,5	370,5
			anon7:	366,4	366,4	366,4	366,4	366,4	366,4	366,7	366,4	366,4
			anon8:	359,0	359,0	359,0	359,0	359,2	359,0	359,0	359,0	359,0
			anon9:	355,7	356,1	355,7	355,7	355,7	355,7	355,7	355,7	355,7
			anon10:	351,0	351,0	351,0	351,0	351,0	351,0	351,0	351,0	351,0
			Nutzer1:	84,1376%	84,1267%	84,125%	84,1245%	84,1247%	84,1247%	84,1248%	84,1249%	84,1243%
			Nutzer2:	5,1073%	5,1118%	5,1131%	5,1143%	5,1137%	5,1137%	5,1135%	5,1127%	5,1146%
			Nutzer2-9:	9,6946%	9,7045%	9,7073%	9,7077%	9,7071%	9,7077%	9,7069%	9,7074%	9,7079%
			Nutzer10+:	6,1677%	6,1688%	6,1677%	6,1678%	6,1682%	6,1676%	6,1683%	6,1677%	6,1678%
			Entropie:	18,3377	18,3375	18,3375	18,3375	18,3375	18,3375	18,3375	18,3375	18,3375

p_{fake}	p_{visit}	p_{fixed}		1.Tick	2.Tick	10.Tick	20.Tick	30.Tick	50.Tick	70.Tick	90.Tick	100.Tick
0,01%	0,01%	50%	fullsizeAnon:	8,211	13,4422	40,3291	58,5809	72,3333	108,5319	137,7281	159,4365	169,2004
			fullsizeForgedAnon:	1,0043	1,0413	1,1535	1,269	1,359	1,5345	1,6474	1,774	1,8575
			numFingerprints:	1,0	1,1254	2,0896	3,3511	4,5996	7,0273	9,3795	11,6647	12,8084
			percentTouched:	0,0657%	0,1168%	0,3353%	0,493%	0,5989%	0,8694%	1,1288%	1,2983%	1,3701%
			detectableFixed:	0,0%	2,0076%	18,3922%	28,1825%	33,0982%	39,3146%	42,7469%	45,1704%	46,4153%
			numChanges:	0,0	0,1254	1,0932	2,3632	3,6367	6,1401	8,5912	11,0378	12,2653
			sizeForgedAnon:	8,211	9,3948	11,3801	8,014	8,6372	6,7938	7,8051	18,8732	8,1689
			anon1:	1150,7	1150,7	1150,7	1150,7	1150,7	1150,6	1150,6	1151,1	1150,6
			anon2:	392,9	392,9	393,1	393,0	392,9	392,9	393,0	392,9	393,0
			anon3:	386,9	386,9	386,9	386,8	386,8	386,9	386,7	386,8	386,8
			anon4:	380,4	380,5	380,4	380,4	380,4	380,4	380,4	380,4	380,4
			anon5:	374,4	374,5	374,4	374,4	374,5	374,4	374,5	374,4	374,6
			anon6:	371,6	371,6	371,8	371,7	371,6	371,7	371,8	371,8	371,7
			anon7:	366,2	366,2	366,2	366,2	366,2	366,3	366,3	366,3	366,2
			anon8:	362,7	362,7	362,7	362,7	362,7	362,7	362,8	362,7	362,7
			anon9:	356,5	356,5	356,6	356,5	356,6	356,5	356,5	356,5	356,5
			anon10:	349,3	349,2	349,2	349,2	349,2	349,2	349,2	349,2	349,2
			Nutzer1:	84,1662%	84,1647%	84,1616%	84,1604%	84,1598%	84,1593%	84,1587%	84,1578%	84,1577%
			Nutzer2:	5,1024%	5,1039%	5,1065%	5,108%	5,1085%	5,1087%	5,1089%	5,1102%	5,1101%
			Nutzer2-9:	9,6792%	9,6808%	9,6841%	9,6857%	9,686%	9,6866%	9,6872%	9,6881%	9,688%
			Nutzer10+:	6,1546%	6,1544%	6,1543%	6,1539%	6,1541%	6,1541%	6,1541%	6,1541%	6,1543%
			Entropie:	18,3381	18,3381	18,338	18,3381	18,338	18,338	18,338	18,338	18,338
0,01%	0,01%	90%	fullsizeAnon:	7,28	7,28	7,6116	8,5608	9,5686	11,4102	11,9744	12,6254	12,6841
			fullsizeForgedAnon:	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
			numFingerprints:	1,0	1,0	1,007	1,0158	1,0254	1,0598	1,0768	1,0981	1,1059
			percentTouched:	0,0724%	0,0724%	0,0764%	0,0879%	0,0998%	0,1181%	0,1247%	0,1308%	0,1314%
			detectableFixed:	36,3756%	36,3756%	36,551%	37,0805%	37,5123%	38,2775%	38,4529%	39,2183%	39,5892%
			numChanges:	0,0	0,0	0,007	0,0176	0,0272	0,0615	0,0925	0,1196	0,1333
			sizeForgedAnon:	7,28	7,28	7,3396	6,1566	6,2328	5,4654	8,0126	7,834	6,5296
			anon1:	1172,6	1172,6	1172,6	1172,5	1172,5	1172,5	1172,6	1172,6	1172,5
			anon2:	399,7	399,7	399,7	399,7	399,7	399,7	399,7	399,7	399,7
			anon3:	390,1	390,1	390,1	390,1	390,1	390,1	390,1	390,1	390,1
			anon4:	382,6	382,6	382,6	382,6	382,6	382,6	382,6	382,6	382,6
			anon5:	376,1	376,1	376,1	376,1	376,1	376,1	376,1	376,1	376,1
			anon6:	369,4	369,4	369,4	369,4	369,4	369,4	369,4	369,4	369,4
			anon7:	365,6	365,6	365,6	365,6	365,6	365,6	365,6	365,6	365,6
			anon8:	361,4	361,4	361,4	361,4	361,4	361,4	361,4	361,4	361,4
			anon9:	355,2	355,2	355,2	355,3	355,3	355,2	355,3	355,3	355,4
			anon10:	345,4	345,4	345,4	345,4	345,4	345,4	345,4	345,4	345,5
			Nutzer1:	84,1801%	84,1801%	84,18%	84,18%	84,18%	84,18%	84,18%	84,1799%	84,1799%
			Nutzer2:	5,0833%	5,0833%	5,0833%	5,0833%	5,0833%	5,0834%	5,0833%	5,0833%	5,0834%
			Nutzer2-9:	9,661%	9,661%	9,6611%	9,6611%	9,6611%	9,6611%	9,6611%	9,6612%	9,6613%
			Nutzer10+:	6,1589%	6,1589%	6,1589%	6,1589%	6,1589%	6,1589%	6,1589%	6,1589%	6,1588%
			Entropie:	18,3378	18,3378	18,3378	18,3378	18,3378	18,3378	18,3378	18,3378	18,3378

Varationskoeffizienten ohne Hops - Fälschen von zufälligen Fingerprints

p_{fake}	p_{visit}	p_{fixed}		1.Tick	2.Tick	10.Tick	20.Tick	30.Tick	50.Tick	70.Tick	90.Tick	100.Tick
			fullsizeAnon:	2,0819%	7,4024%	12,2248%	7,392%	9,5984%	7,7788%	6,6315%	5,705%	3,1755%
			fullsizeForgedAnon:	2,0819%	7,4024%	12,2248%	7,392%	9,5984%	7,7788%	6,6315%	5,705%	3,1755%
			numFingerprints:	0,0%	0,0886%	0,1786%	0,1204%	0,1418%	0,152%	0,2053%	0,1954%	0,267%
			percentTouched:	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
			detectableFixed:	0,1578%	0,1747%	0,2142%	0,1574%	0,2047%	0,1524%	0,1883%	0,1866%	0,2229%
			numChanges:	-	65,4395%	15,1837%	6,0161%	7,0183%	6,4494%	5,2528%	5,2325%	5,23%
			sizeForgedAnon:	2,0819%	3,5048%	13,8877%	13,8406%	16,2823%	12,7653%	13,1792%	14,1113%	10,1791%
			anon1:	2,7146%	8,2053%	33,7891%	35,5354%	42,2445%	35,0104%	36,3796%	31,2454%	30,2839%
			anon2:	3,4808%	23,3478%	21,2286%	23,2368%	24,0387%	16,5466%	15,2207%	21,9157%	15,7137%
			anon3:	2,551%	23,8237%	23,4083%	21,9916%	21,3054%	26,1949%	18,6104%	18,6059%	17,8344%
100%	0,01%	90%	anon4:	2,5521%	18,4707%	22,0802%	24,4472%	20,0203%	28,8145%	16,9175%	16,2441%	20,9577%
			anon5:	1,9704%	1,635%	22,437%	18,9414%	22,4236%	19,7176%	16,0284%	18,0749%	15,8541%
			anon6:	2,2894%	2,9327%	17,5414%	16,7317%	13,8848%	18,6374%	11,3804%	14,8301%	11,5484%
			anon7:	2,4194%	2,8095%	14,9491%	14,7978%	9,8716%	19,7041%	10,67%	13,8694%	11,8371%
			anon8:	2,8125%	3,0244%	4,6531%	11,0846%	7,0691%	17,8825%	10,8795%	11,9447%	11,3546%
			anon9:	2,496%	3,5987%	5,123%	5,7295%	6,4862%	18,208%	11,5416%	10,1117%	10,3491%
			anon10:	3,9713%	5,1277%	5,0569%	7,1373%	5,378%	13,7663%	12,8214%	8,9506%	10,4992%
			Nutzer1:	0,0796%	0,0761%	0,0761%	0,0712%	0,0748%	0,0741%	0,0767%	0,0822%	0,0795%
			Nutzer2:	0,9719%	0,9725%	0,9772%	0,9779%	1,0221%	1,1496%	1,008%	1,1059%	1,0682%
			Nutzer2-9:	0,6733%	0,6935%	0,65%	0,7057%	0,6393%	0,7015%	0,7595%	0,8304%	0,817%
			Nutzer10+:	0,7835%	0,7906%	0,6516%	0,854%	0,743%	0,6857%	0,6922%	0,8354%	0,7578%
			Entropie:	0,0164%	0,013%	0,0129%	0,0133%	0,0174%	0,0228%	0,0194%	0,0257%	0,0165%
			fullsizeAnon:	10,665%	5,7039%	1,9611%	2,3197%	1,5759%	1,7956%	1,5319%	1,4469%	1,4588%
			fullsizeForgedAnon:	1,6507%	1,6085%	1,6276%	2,4132%	1,9299%	2,4661%	2,6425%	2,4989%	2,3393%
			numFingerprints:	0,0%	0,0065%	0,0046%	0,003%	0,002%	0,0035%	0,007%	0,0053%	0,0048%
			percentTouched:	1,4847%	0,9605%	0,8227%	0,8039%	0,8325%	0,8025%	0,754%	0,6592%	0,6174%
			detectableFixed:	-	4,8636%	0,0549%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
			numChanges:	-	0,0131%	0,0019%	0,0011%	0,0009%	0,0008%	0,0007%	0,0006%	0,0006%
			sizeForgedAnon:	10,665%	4,8309%	5,5513%	6,3862%	7,0976%	6,7603%	7,187%	10,2042%	8,0831%
			anon1:	2,2446%	2,255%	2,2629%	2,2579%	2,3499%	2,2166%	2,1998%	2,1614%	2,2885%
			anon2:	3,0126%	3,1339%	3,1699%	2,7459%	3,0518%	2,805%	2,607%	2,7812%	2,8978%
			anon3:	3,0826%	3,1597%	3,0523%	2,9198%	3,246%	2,9533%	2,9672%	3,1813%	3,0208%
1%	100%	0%	anon4:	2,8307%	3,0658%	2,9718%	2,7437%	3,1224%	2,8664%	2,6488%	2,7784%	2,7442%
			anon5:	2,3846%	2,2611%	2,3257%	2,2849%	2,227%	2,1521%	2,1292%	2,3997%	2,4262%
			anon6:	2,214%	2,5238%	2,562%	2,4114%	2,432%	2,5045%	2,4501%	2,4563%	2,5016%
			anon7:	2,0669%	2,1256%	2,1652%	2,0902%	2,3669%	1,9193%	2,0487%	2,1533%	2,1148%
			anon8:	2,3447%	2,2087%	2,1805%	2,1756%	2,062%	2,298%	2,1593%	2,4355%	2,1837%
			anon9:	2,3003%	2,1815%	2,1309%	1,8437%	2,0757%	2,2135%	2,3649%	2,3243%	2,3079%
			anon10:	1,7914%	1,7124%	1,6774%	1,6229%	1,6561%	1,735%	1,4603%	1,4605%	1,7675%
			Nutzer1:	0,0794%	0,0863%	0,0823%	0,083%	0,0817%	0,0831%	0,0835%	0,0843%	0,0854%
			Nutzer2:	0,9197%	0,8564%	0,6876%	0,7967%	0,7391%	0,7405%	0,8114%	0,6866%	0,7001%
			Nutzer2-9:	0,7559%	0,7499%	0,6567%	0,6855%	0,692%	0,7077%	0,6829%	0,6768%	0,6964%
			Nutzer10+:	0,4781%	0,4812%	0,4918%	0,5441%	0,5237%	0,5059%	0,4648%	0,4083%	0,499%
			Entropie:	0,0105%	0,0106%	0,0107%	0,011%	0,0111%	0,0105%	0,0112%	0,01%	0,0106%
			fullsizeAnon:	8,4388%	4,1652%	3,8195%	3,2549%	2,2845%	1,7208%	1,6857%	1,4968%	1,2146%
			fullsizeForgedAnon:	1,341%	1,6417%	1,5903%	1,4532%	1,2263%	1,2506%	1,1676%	1,1734%	1,1702%
			numFingerprints:	0,0%	0,0%	0,0033%	0,0051%	0,0058%	0,0065%	0,0049%	0,0051%	0,005%
			percentTouched:	2,6143%	1,924%	0,5927%	0,5043%	0,4824%	0,3924%	0,3762%	0,3358%	0,3243%
			detectableFixed:	-	3,8994%	0,0255%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
			numChanges:	-	0,0%	0,002%	0,0016%	0,0014%	0,0008%	0,0008%	0,0005%	0,0006%
			sizeForgedAnon:	8,4388%	6,8493%	7,7824%	14,9011%	13,0614%	10,2238%	13,9871%	12,1671%	8,9912%
			anon1:	1,8394%	1,709%	1,8956%	1,76%	1,915%	1,8992%	1,595%	1,8196%	1,7063%
			anon2:	2,2001%	2,3017%	2,1802%	1,8726%	2,7275%	1,9742%	2,149%	2,5258%	2,6665%
			anon3:	1,2754%	1,0092%	1,0733%	0,7962%	1,3798%	1,407%	1,2096%	1,3609%	1,4664%
1%	1%	0%	anon4:	1,2608%	1,0776%	1,2892%	1,0965%	1,3241%	1,1562%	1,2931%	1,3525%	1,0578%
			anon5:	1,4031%	1,4506%	1,2397%	1,5097%	1,2645%	1,1895%	1,3943%	1,4169%	1,3721%
			anon6:	1,0347%	1,0235%	0,8751%	0,8494%	0,806%	0,8917%	0,9909%	0,9873%	1,0796%
			anon7:	1,5314%	1,2545%	1,2223%	1,1816%	1,2374%	1,3038%	1,486%	1,5627%	1,5112%
			anon8:	1,3494%	1,3524%	1,3829%	1,1713%	1,3066%	1,3369%	1,2621%	1,2421%	1,2771%
			anon9:	1,1951%	1,3286%	0,8771%	0,9949%	1,2596%	1,0254%	1,2404%	1,1057%	0,9716%
			anon10:	1,908%	1,5117%	1,8945%	1,5618%	1,5151%	1,8182%	1,624%	2,0544%	1,4866%
			Nutzer1:	0,0883%	0,0849%	0,0782%	0,0873%	0,0803%	0,082%	0,0821%	0,0881%	0,0838%
			Nutzer2:	1,0913%	1,2131%	1,0649%	1,0753%	0,9488%	0,9717%	0,9592%	1,1171%	0,9814%
			Nutzer2-9:	0,6266%	0,5879%	0,5476%	0,5702%	0,5842%	0,5437%	0,5517%	0,6078%	0,5401%
			Nutzer10+:	0,6306%	0,5517%	0,6076%	0,6814%	0,6344%	0,5928%	0,5658%	0,5973%	0,5482%
			Entropie:	0,0136%	0,012%	0,0137%	0,0121%	0,0126%	0,0131%	0,0127%	0,0119%	0,0126%
			fullsizeAnon:	7,0202%	9,471%	3,7305%	3,7328%	2,893%	2,1457%	1,8798%	1,5941%	1,3759%
			fullsizeForgedAnon:	1,1896%	2,1605%	0,947%	1,0693%	1,0945%	1,1545%	1,2013%	1,2005%	1,2287%
			numFingerprints:	0,0%	0,0692%	0,118%	0,1137%	0,12%	0,13%	0,1312%	0,1348%	0,1343%
			percentTouched:	3,8402%	3,3096%	0,7468%	0,5988%	0,4851%	0,4961%	0,5172%	0,472%	0,4627%
			detectableFixed:	-	4,0898%	0,0912%	0,0736%	0,0586%	0,0547%	0,0533%	0,0497%	0,0451%
			numChanges:	-	0,1389%	0,1326%	0,1205%	0,1248%	0,123%	0,127%	0,1318%	0,1319%
			sizeForgedAnon:	7,0202%	13,9667%	13,0758%	10,7287%	14,7551%	14,4898%	12,1674%	10,0817%	9,6618%
			anon1:	2,896%	3,0725%	3,049%	3,0888%	3,2877%	3,2323%	2,8624%	2,8303%	3,0101%
			anon2:	2,8188%	2,785%	3,6697%	3,1622%	3,0466%	3,0625%	3,4289%	3,5107%	3,724%
			anon3:	2,072%	1,6543%	2,0121%	1,8464%	1,9627%	1,9768%	1,8908%	1,6085%	2,0247%
1%	1%	10%	anon4:	1,8825%	1,7051%	1,3994%	1,3894%	1,9129%	1,6194%	1,719%	1,4736%	1,9462%
			anon5:	1,5015%	1,2714%	1,7075%	1,8339%	1,8153%	1,5986%	1,6828%	1,8884%	1,6769%
			anon6:	1,6319%	1,126%	1,6304%	1,392%	1,6767%	1,5396%	1,7386%	1,8798%	1,5747%
			anon7:	1,8639%	1,9019%	2,0304%	1,8873%	1,7938%	1,9614%	2,1516%	2,1249%	2,0485%
			anon8:	2,3061%	2,1834%	2,6008%	2,6446%	2,1717%	2,5542%	2,4488%	2,3953%	2,0855%
			anon9:	1,9234%	1,7253%	2,0496%	1,9462%	2,3816%	2,0211%	2,2026%	2,0881%	2,035%
			anon10:	2,1626%	2,1789%	2,3292%	2,1395%	2,2845%	2,0558%	1,6632%	2,0304%	2,2086%
			Nutzer1:	0,0611%	0,067%	0,0713%	0,0694%	0,0723%	0,0654%	0,0692%	0,0665%	0,071%
			Nutzer2:	0,6854%	0,7596%	0,96%	0,7471%	0,8074%	0,5383%	0,7051%	0,617%	0,8127%
			Nutzer2-9:	0,4791%	0,4803%	0,5159%	0,5222%	0,5239%	0,5126%	0,5055%	0,4837%	0,46%
			Nutzer10+:	0,6866%	0,6889%	0,7374%	0,7339%	0,6618%	0,8375%	0,6614%	0,8642%	0,673%
			Entropie:	0,0122%	0,0122%	0,0121%	0,0124%	0,0114%	0,0124%	0,0111%	0,0129%	0,0123%

P_{fake}	P_{visit}	P_{fixed}		1.Tick	2.Tick	10.Tick	20.Tick	30.Tick	50.Tick	70.Tick	90.Tick	100.Tick
			fullsizeAnon:	13,1683%	7,7336%	4,7105%	4,9936%	4,5696%	4,3187%	4,238%	4,0661%	4,171%
			fullsizeForgedAnon:	2,811%	3,3573%	4,1837%	4,2745%	4,267%	4,2638%	4,3436%	4,4308%	4,4793%
			numFingerprints:	0,0%	0,5744%	1,0905%	1,249%	1,3033%	1,3458%	1,4074%	1,4472%	1,4577%
			percentTouched:	2,2022%	1,7799%	1,1589%	1,1438%	1,1865%	1,318%	1,3614%	1,3421%	1,3564%
			detectableFixed:	38,4135%	4,7787%	1,0901%	0,9538%	0,927%	1,0315%	1,0407%	0,8725%	0,7919%
			numChanges:	-	1,6094%	1,2852%	1,3225%	1,3341%	1,3306%	1,3541%	1,345%	1,3374%
			sizeForgedAnon:	13,1683%	8,5105%	7,9264%	11,3173%	7,5853%	6,7687%	14,5917%	8,6595%	13,3673%
			anon1:	2,9068%	3,1904%	3,1693%	3,121%	2,841%	3,1155%	3,3446%	2,9555%	3,177%
			anon2:	2,7055%	2,3132%	2,2616%	2,4719%	2,4298%	3,1315%	2,9976%	2,8247%	2,4988%
			anon3:	1,0575%	1,4025%	1,0347%	1,2667%	1,3466%	1,2179%	1,3934%	1,4086%	1,1107%
1%	1%	50%	anon4:	1,5371%	1,448%	1,7388%	1,8389%	1,6049%	1,5532%	1,4799%	1,4178%	1,4673%
			anon5:	1,3543%	1,19%	1,3861%	1,5254%	1,5294%	1,5724%	1,4364%	1,471%	1,3205%
			anon6:	1,3521%	1,4507%	1,1101%	1,5017%	1,4391%	1,55%	1,4631%	1,3438%	1,4134%
			anon7:	1,5864%	1,6907%	1,7313%	1,7966%	1,6336%	1,8729%	1,6511%	1,9249%	1,4685%
			anon8:	1,8558%	1,8689%	2,0322%	1,7803%	1,8356%	2,034%	2,2509%	2,3058%	1,9346%
			anon9:	2,3567%	2,0992%	2,653%	2,5395%	2,2541%	2,4082%	1,8975%	2,2415%	2,2481%
			anon10:	3,0631%	3,0997%	3,0032%	2,8771%	3,0783%	3,1228%	2,9451%	2,8577%	3,3427%
			Nutzer1:	0,0606%	0,0713%	0,0637%	0,0643%	0,0678%	0,0732%	0,0705%	0,0707%	0,0688%
			Nutzer2:	0,5061%	0,6195%	0,485%	0,5526%	0,6419%	0,6575%	0,637%	0,5498%	0,5481%
			Nutzer2-9:	0,4684%	0,5054%	0,4178%	0,3923%	0,4122%	0,4739%	0,4101%	0,459%	0,4337%
			Nutzer10+:	0,3593%	0,4191%	0,4651%	0,4231%	0,388%	0,4019%	0,4305%	0,3884%	0,4089%
			Entropie:	0,0095%	0,0096%	0,0099%	0,0096%	0,0088%	0,0095%	0,0095%	0,0094%	0,0094%
			fullsizeAnon:	11,3068%	11,9802%	10,4236%	10,1582%	9,9212%	9,7886%	9,7827%	9,8225%	9,7894%
			fullsizeForgedAnon:	1,699%	2,7726%	4,1643%	4,5892%	4,7969%	5,033%	5,2059%	5,3521%	5,3797%
			numFingerprints:	0,0%	0,3244%	1,0464%	1,5818%	1,7781%	2,1698%	2,5199%	2,6602%	2,6997%
			percentTouched:	3,4125%	3,6123%	3,2148%	3,0775%	2,5726%	2,5868%	2,6084%	2,6402%	2,6383%
			detectableFixed:	1,9419%	2,0732%	1,614%	1,3126%	1,1376%	1,3275%	1,2244%	1,1253%	1,105%
			numChanges:	-	9,0522%	5,2526%	5,4196%	5,1863%	5,0243%	4,9438%	5,0475%	5,1381%
			sizeForgedAnon:	11,3068%	13,4924%	15,5098%	10,6801%	10,2612%	12,1947%	12,9796%	10,7057%	9,0739%
			anon1:	3,7547%	3,7565%	3,788%	3,7113%	3,707%	3,7954%	3,6865%	3,6464%	3,5798%
			anon2:	1,9638%	2,2566%	2,1534%	2,1506%	1,9911%	2,1423%	1,8381%	2,033%	2,0595%
			anon3:	0,973%	0,912%	0,8773%	0,9538%	0,8903%	0,8292%	0,9486%	1,0005%	0,8662%
1%	1%	90%	anon4:	1,4646%	1,6177%	1,5415%	1,6599%	1,479%	1,4838%	1,2583%	1,4123%	1,5868%
			anon5:	1,3597%	1,2865%	1,3473%	1,2961%	1,2635%	1,3785%	1,2576%	1,1366%	1,0812%
			anon6:	1,3145%	1,1336%	1,1993%	1,3024%	1,0918%	1,3546%	1,3141%	1,2853%	1,2451%
			anon7:	1,5485%	1,4415%	1,6299%	1,6836%	1,7348%	1,719%	1,5421%	1,7506%	1,7529%
			anon8:	1,8626%	1,9636%	1,9553%	1,7918%	1,9791%	1,971%	1,9655%	1,9433%	1,9376%
			anon9:	1,544%	1,6438%	1,6931%	1,7014%	1,8518%	1,7214%	1,529%	1,635%	1,6644%
			anon10:	2,2513%	2,3585%	2,2142%	2,2847%	2,2183%	2,3522%	2,0941%	2,3486%	2,1211%
			Nutzer1:	0,0811%	0,0817%	0,0836%	0,0834%	0,0852%	0,0856%	0,0848%	0,0829%	0,0838%
			Nutzer2:	0,7497%	0,7406%	0,7035%	0,6721%	0,7077%	0,6941%	0,7026%	0,6828%	0,7237%
			Nutzer2-9:	0,5604%	0,5503%	0,5769%	0,5763%	0,5801%	0,5717%	0,5742%	0,5339%	0,54%
			Nutzer10+:	0,7002%	0,6872%	0,6992%	0,6897%	0,6731%	0,7364%	0,6513%	0,6798%	0,6931%
			Entropie:	0,0152%	0,0152%	0,0154%	0,0157%	0,0156%	0,0154%	0,0153%	0,0154%	0,0155%
			fullsizeAnon:	7,386%	11,0025%	6,8116%	4,4085%	4,7162%	3,611%	3,091%	2,4738%	2,5897%
			fullsizeForgedAnon:	1,0849%	10,4198%	5,4158%	3,1767%	2,9424%	2,1037%	1,8681%	1,6995%	1,5715%
			numFingerprints:	0,0%	0,0%	0,0127%	0,0127%	0,0265%	0,0253%	0,0257%	0,0216%	0,0265%
			percentTouched:	2,5541%	2,7185%	2,5616%	2,5619%	2,1318%	1,9355%	1,7309%	1,347%	1,4993%
			detectableFixed:	-	8,6569%	0,0527%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
			numChanges:	-	0,0%	0,0094%	0,0071%	0,0071%	0,0068%	0,0047%	0,0052%	0,0044%
			sizeForgedAnon:	7,386%	11,7538%	16,4645%	7,8694%	22,6752%	22,9598%	18,013%	16,0629%	10,5153%
			anon1:	4,0571%	4,2506%	6,1657%	4,7801%	6,6025%	7,1737%	4,0337%	4,2414%	4,8125%
			anon2:	2,0472%	9,8334%	8,0704%	5,9047%	7,5581%	8,0718%	10,6361%	8,5688%	6,4448%
			anon3:	1,9082%	7,0521%	2,0231%	2,021%	1,865%	6,2726%	1,8107%	5,1648%	1,7167%
1%	0,01%	0%	anon4:	1,4102%	2,356%	1,4316%	1,8983%	2,054%	1,9034%	1,6428%	2,5399%	1,4296%
			anon5:	1,5581%	2,0458%	1,6262%	2,0537%	1,708%	1,7993%	1,4824%	2,1442%	1,6334%
			anon6:	1,262%	1,8315%	1,4044%	1,8553%	1,5816%	1,368%	1,368%	1,9241%	1,3528%
			anon7:	1,4712%	2,0241%	1,5834%	2,036%	1,6564%	1,5436%	1,5436%	2,0714%	1,5486%
			anon8:	1,1414%	1,708%	1,4162%	1,6511%	1,2262%	1,1885%	1,1885%	1,7305%	1,1885%
			anon9:	0,9578%	0,8521%	0,786%	0,786%	0,8642%	0,9716%	0,786%	1,2739%	0,786%
			anon10:	1,1017%	1,2838%	0,9782%	0,9782%	1,0593%	1,0047%	0,9782%	1,1415%	0,9782%
			Nutzer1:	0,1054%	0,1097%	0,1106%	0,1108%	0,1105%	0,1097%	0,1105%	0,1113%	0,1103%
			Nutzer2:	0,8641%	0,7483%	0,7433%	0,7399%	0,747%	0,749%	0,7414%	0,7472%	0,7458%
			Nutzer2-9:	0,7234%	0,6937%	0,696%	0,6927%	0,7041%	0,6958%	0,6851%	0,7002%	0,7032%
			Nutzer10+:	0,7323%	0,6804%	0,7016%	0,6957%	0,6859%	0,6747%	0,6866%	0,7024%	0,6835%
			Entropie:	0,0188%	0,0278%	0,0258%	0,0187%	0,0262%	0,0324%	0,0232%	0,0162%	0,0244%
			fullsizeAnon:	9,608%	14,6524%	4,4326%	2,6537%	2,7203%	2,8604%	2,4227%	2,1023%	1,9351%
			fullsizeForgedAnon:	1,8032%	21,3491%	6,287%	3,9249%	2,6365%	1,8843%	1,6357%	1,4523%	1,4015%
			numFingerprints:	0,0%	0,9875%	0,6211%	0,3654%	0,4129%	0,5193%	0,5091%	0,4915%	0,4852%
			percentTouched:	2,6818%	2,3211%	1,7438%	1,7435%	1,8381%	1,6455%	1,5514%	1,4882%	1,5905%
			detectableFixed:	-	13,3588%	0,6007%	0,2243%	0,1387%	0,1815%	0,1407%	0,1492%	0,1501%
			numChanges:	-	2,1093%	0,7018%	0,3926%	0,4107%	0,4836%	0,4564%	0,4343%	0,4235%
			sizeForgedAnon:	9,608%	16,0747%	15,676%	10,8179%	17,4722%	16,6081%	18,0301%	14,5513%	10,3012%
			anon1:	3,1057%	4,6966%	5,969%	3,1356%	3,8441%	4,6094%	4,9245%	3,1104%	3,1262%
			anon2:	3,4666%	9,3544%	10,2687%	3,6098%	9,5428%	13,578%	7,2133%	15,0758%	9,86%
			anon3:	3,2722%	6,0032%	3,4678%	3,4442%	3,3284%	5,6217%	5,7321%	4,5334%	3,4952%
1%	0,01%	10%	anon4:	2,6447%	2,869%	2,7157%	2,7405%	2,6482%	2,8554%	2,7721%	3,8106%	2,9165%
			anon5:	3,2857%	2,8162%	2,963%	3,024%	2,8228%	3,1219%	3,0615%	3,4649%	3,1466%
			anon6:	3,0674%	3,2671%	3,0009%	3,0314%	2,9172%	3,0979%	3,0972%	3,2558%	3,1535%
			anon7:	3,1564%	3,2783%	2,9369%	3,1948%	3,1943%	3,1643%	3,2255%	3,2075%	3,1948%
			anon8:	2,6978%	2,9167%	2,7966%	2,8715%	2,8376%	2,8551%	3,2517%	3,2509%	2,8715%
			anon9:	2,79%	2,8687%	2,8069%	2,8069%	2,8069%	2,5273%	2,8187%	2,8367%	2,8069%
			anon10:	2,7067%	2,6469%	2,5616%	2,5616%	2,5907%	2,4212%	2,5616%	2,6469%	2,5616%
			Nutzer1:	0,0468%	0,0544%	0,0489%	0,0476%	0,0472%	0,0485%	0,049%	0,0493%	0,0471%
			Nutzer2:	0,574%	0,519%	0,5158%	0,5211%	0,4693%	0,4785%	0,5801%	0,5236%	0,5279%
			Nutzer2-9:	0,3395%	0,3368%	0,3557%	0,3176%	0,3115%	0,3142%	0,3221%	0,3289%	0,2967%
			Nutzer10+:	0,6119%	0,7021%	0,5817%	0,5926%	0,5552%	0,5448%	0,6031%	0,5428%	0,5867%
			Entropie:	0,0112%	0,0124%	0,0128%	0,0092%	0,0134%	0,0146%	0,011%	0,0128%	0,0145%

p_{fake}	p_{visit}	p_{fixed}		1.Tick	2.Tick	10.Tick	20.Tick	30.Tick	50.Tick	70.Tick	90.Tick	100.Tick
			fullsizeAnon:	7,5397%	34,5093%	16,1082%	11,5163%	10,5292%	8,0057%	6,9946%	5,428%	5,4316%
			fullsizeForgedAnon:	1,0982%	9,2096%	8,0238%	5,6674%	5,7378%	5,6487%	5,1433%	4,9674%	4,9975%
			numFingerprints:	0,0%	1,1726%	2,8262%	2,46%	2,792%	2,8228%	2,6016%	2,443%	2,4556%
			percentTouched:	2,1562%	2,1304%	1,6219%	1,5846%	1,644%	1,6092%	1,0219%	1,1485%	0,9877%
			detectableFixed:	48,7416%	19,7959%	4,7666%	3,4401%	2,7997%	2,2595%	2,0477%	1,9652%	1,8839%
			numChanges:	-	10,1108%	5,4134%	3,4627%	3,3043%	3,2989%	3,0036%	2,6762%	2,7343%
			sizeForgedAnon:	7,5397%	46,6725%	33,2058%	38,7771%	21,4995%	32,158%	25,8761%	42,1334%	44,7237%
			anon1:	2,3093%	3,564%	3,4134%	3,3871%	2,3285%	2,7263%	2,5465%	3,2476%	3,9759%
			anon2:	2,3382%	6,0888%	4,7371%	3,9282%	3,7186%	5,137%	2,4238%	6,1846%	2,5652%
			anon3:	1,9953%	2,0123%	3,0239%	2,7738%	2,2268%	4,3148%	2,5142%	1,7426%	2,3087%
1%	0,01%	50%	anon4:	2,089%	2,301%	1,7735%	2,9354%	2,1382%	3,3061%	2,5654%	1,7096%	1,8183%
			anon5:	2,1238%	1,8766%	2,2659%	2,8491%	2,0527%	2,5609%	2,7374%	1,8625%	1,8342%
			anon6:	2,2192%	2,0766%	1,8736%	2,0065%	2,5526%	2,8214%	2,5436%	1,9294%	1,8415%
			anon7:	2,3245%	2,4095%	2,2637%	2,5034%	2,493%	2,4444%	2,3617%	2,0404%	1,7332%
			anon8:	2,4355%	2,4467%	2,5502%	2,6877%	2,7932%	2,7842%	2,7338%	2,135%	2,3822%
			anon9:	2,587%	2,6474%	2,6826%	2,6393%	2,9287%	2,678%	2,8124%	2,4666%	2,783%
			anon10:	2,3454%	2,3004%	2,4021%	3,0202%	2,4049%	2,1832%	2,6174%	2,2811%	2,1764%
			Nutzer1:	0,0555%	0,0527%	0,0499%	0,0514%	0,0561%	0,0548%	0,0564%	0,0548%	0,0585%
			Nutzer2:	0,9774%	0,9954%	0,9105%	0,9946%	1,0117%	0,8913%	0,9337%	0,8306%	0,7802%
			Nutzer2-9:	0,492%	0,4779%	0,4966%	0,4059%	0,4869%	0,5282%	0,4877%	0,5152%	0,5063%
			Nutzer10+:	0,5592%	0,5929%	0,5464%	0,4907%	0,5544%	0,5826%	0,5906%	0,6439%	0,5566%
			Entropie:	0,0099%	0,0117%	0,0105%	0,0095%	0,0098%	0,0112%	0,0101%	0,01%	0,0095%
			fullsizeAnon:	8,9559%	9,9351%	12,325%	8,2241%	8,6901%	6,2584%	5,8922%	6,5043%	6,2163%
			fullsizeForgedAnon:	1,6482%	1,9928%	2,849%	2,4239%	2,827%	2,3383%	2,2308%	2,6388%	2,5858%
			numFingerprints:	0,0%	0,0815%	0,1339%	0,2107%	0,2751%	0,392%	0,6053%	0,6751%	0,6807%
			percentTouched:	3,0468%	3,0135%	2,4967%	2,3889%	2,3821%	3,2757%	3,6233%	3,6784%	3,8074%
			detectableFixed:	2,0486%	1,9798%	1,8051%	1,8738%	1,8365%	1,8626%	1,659%	1,6912%	1,576%
			numChanges:	-	48,6606%	13,2813%	11,3862%	9,1741%	7,6147%	7,9533%	7,7033%	6,9817%
			sizeForgedAnon:	8,9559%	9,6442%	10,7004%	13,2343%	13,1663%	14,5668%	11,0574%	10,0635%	10,2025%
			anon1:	3,0735%	3,0182%	3,2064%	3,2761%	3,1735%	3,1356%	3,1555%	3,1465%	3,0652%
			anon2:	3,0093%	3,0057%	2,9182%	3,0465%	3,1599%	3,0693%	3,109%	3,0353%	3,234%
			anon3:	2,6709%	2,6857%	2,6044%	2,6854%	2,6062%	2,46%	2,6088%	2,9061%	2,5869%
1%	0,01%	90%	anon4:	1,9127%	2,1314%	1,8647%	1,6575%	1,6783%	1,7047%	2,0025%	1,8382%	1,832%
			anon5:	2,0268%	2,0278%	2,1208%	1,8447%	2,0009%	2,0251%	1,9804%	2,1074%	2,1261%
			anon6:	1,642%	1,6632%	1,6661%	1,541%	1,809%	1,5981%	1,5794%	1,9359%	1,6244%
			anon7:	1,9784%	1,8865%	1,9545%	2,0049%	2,0649%	2,3813%	2,1546%	2,2564%	2,1561%
			anon8:	2,5039%	2,4921%	2,289%	2,4102%	2,6753%	2,3747%	2,3006%	2,0775%	2,3203%
			anon9:	2,4799%	2,4799%	2,557%	2,4513%	2,5105%	2,5601%	2,3443%	2,3885%	2,4865%
			anon10:	2,9052%	2,8854%	2,8635%	2,6272%	2,602%	2,7308%	2,6655%	2,6423%	2,8434%
			Nutzer1:	0,0582%	0,0581%	0,0581%	0,0584%	0,0587%	0,0585%	0,0589%	0,059%	0,0592%
			Nutzer2:	0,3544%	0,3533%	0,3521%	0,3507%	0,3565%	0,3762%	0,3846%	0,3953%	0,3928%
			Nutzer2-9:	0,4886%	0,4886%	0,481%	0,4793%	0,4824%	0,4773%	0,4833%	0,48%	0,4971%
			Nutzer10+:	0,5781%	0,5802%	0,5785%	0,5906%	0,5935%	0,5761%	0,5933%	0,5872%	0,594%
			Entropie:	0,0143%	0,0142%	0,0143%	0,0144%	0,0143%	0,0142%	0,0142%	0,0143%	0,0142%
			fullsizeAnon:	87,6711%	45,8173%	30,3942%	17,2441%	12,2368%	5,5907%	6,3729%	6,569%	7,815%
			fullsizeForgedAnon:	0,0%	0,0%	2,1738%	2,3831%	2,3034%	3,3401%	3,7874%	4,474%	3,882%
			numFingerprints:	0,0%	0,0%	0,0526%	0,0396%	0,0325%	0,0427%	0,0333%	0,031%	0,0286%
			percentTouched:	93,8244%	48,429%	25,1566%	10,0809%	8,9812%	8,3127%	9,7444%	9,2917%	10,0565%
			detectableFixed:	-	53,4771%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
			numChanges:	-	0,0%	0,0%	0,0%	0,0%	0,0%	0,0099%	0,0077%	0,0069%
			sizeForgedAnon:	87,6711%	73,2483%	73,4597%	107,3669%	70,4863%	82,7218%	90,632%	92,0358%	80,2488%
			anon1:	1,9554%	1,9554%	1,9554%	1,9473%	1,9554%	1,9572%	1,9444%	1,9554%	1,9384%
			anon2:	2,5891%	2,5891%	2,5891%	2,5891%	2,5891%	2,5891%	2,5891%	2,5893%	2,5891%
			anon3:	1,8926%	1,8926%	1,9006%	1,8926%	1,8971%	1,8926%	1,8926%	1,8926%	1,8926%
			anon4:	1,9628%	1,9712%	1,9628%	2,0111%	1,994%	1,9733%	1,9712%	1,9712%	2,0533%
0,01%	100%	0%	anon5:	1,6308%	1,6696%	1,6308%	1,6308%	1,6308%	1,6308%	1,6308%	1,6308%	1,672%
			anon6:	1,4237%	1,4237%	1,4512%	1,4475%	1,4237%	1,4237%	1,4237%	1,4359%	1,4359%
			anon7:	1,5817%	1,6244%	1,5722%	1,6466%	1,6244%	1,6244%	1,6244%	1,6244%	1,6244%
			anon8:	1,2634%	1,2634%	1,2634%	1,2634%	1,2634%	1,2634%	1,2634%	1,3334%	1,2634%
			anon9:	1,4925%	1,4925%	1,4925%	1,4925%	1,4925%	1,4925%	1,4925%	1,4925%	1,4837%
			anon10:	1,6149%	1,6149%	1,6149%	1,6149%	1,6149%	1,6269%	1,6149%	1,5922%	1,5922%
			Nutzer1:	0,0574%	0,0555%	0,0555%	0,0553%	0,0554%	0,0557%	0,0556%	0,0551%	0,0554%
			Nutzer2:	0,6612%	0,6415%	0,6356%	0,6456%	0,6525%	0,652%	0,6419%	0,6424%	0,63%
			Nutzer2-9:	0,4609%	0,4621%	0,4525%	0,4589%	0,4545%	0,4566%	0,4556%	0,4513%	0,4572%
			Nutzer10+:	0,959%	0,9577%	0,9527%	0,9586%	0,9575%	0,9622%	0,9526%	0,9573%	0,9564%
			Entropie:	0,0122%	0,0121%	0,012%	0,0121%	0,0121%	0,012%	0,012%	0,012%	0,0121%
			fullsizeAnon:	83,1516%	78,1748%	46,1764%	37,8596%	27,0598%	22,8803%	17,874%	17,3755%	17,3438%
			fullsizeForgedAnon:	0,0%	0,0%	3,2244%	3,7805%	3,7037%	4,4468%	4,5532%	4,9342%	5,4716%
			numFingerprints:	0,0%	0,0%	0,2841%	0,3093%	0,3663%	0,4516%	0,5324%	0,6172%	0,6847%
			percentTouched:	95,5874%	75,8084%	38,9145%	20,487%	15,8523%	13,7679%	11,1947%	8,8683%	8,588%
			detectableFixed:	-	54,1469%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
			numChanges:	-	0,0%	0,1507%	0,0713%	0,0845%	0,0586%	0,0507%	0,0488%	0,0434%
			sizeForgedAnon:	83,1516%	79,4757%	98,7927%	62,8114%	88,0003%	90,0075%	71,5315%	114,8848%	88,8888%
			anon1:	2,4284%	2,4335%	2,4284%	2,4284%	2,4284%	2,4294%	2,4355%	2,4346%	2,4225%
			anon2:	2,843%	2,843%	2,843%	2,843%	2,843%	2,8539%	2,843%	2,8045%	2,8161%
			anon3:	1,8161%	1,8672%	1,857%	1,8335%	1,857%	1,857%	1,8744%	1,857%	1,857%
0,01%	100%	10%	anon4:	1,9178%	1,8673%	1,899%	1,9178%	1,899%	1,9178%	1,9315%	1,9178%	1,9178%
			anon5:	1,7156%	1,7156%	1,7156%	1,7156%	1,7156%	1,7156%	1,7156%	1,7156%	1,7156%
			anon6:	1,759%	1,759%	1,7411%	1,759%	1,759%	1,7621%	1,759%	1,759%	1,759%
			anon7:	1,8548%	1,8548%	1,8548%	1,8548%	1,8548%	1,8548%	1,8548%	1,8548%	1,8548%
			anon8:	1,5473%	1,5473%	1,5473%	1,5473%	1,5473%	1,5473%	1,5473%	1,5473%	1,5473%
			anon9:	1,8793%	1,8793%	1,8874%	1,9163%	1,8793%	1,8793%	1,8793%	1,9022%	1,8793%
			anon10:	1,6986%	1,7036%	1,7036%	1,7036%	1,7036%	1,708%	1,7036%	1,7036%	1,7036%
			Nutzer1:	0,0616%	0,0623%	0,0622%	0,0622%	0,0625%	0,0624%	0,0623%	0,0623%	0,0621%
			Nutzer2:	0,856%	0,8576%	0,8673%	0,8791%	0,8839%	0,897%	0,8813%	0,8922%	0,8653%
			Nutzer2-9:	0,7211%	0,7247%	0,7291%	0,7245%	0,7296%	0,7255%	0,7218%	0,7284%	0,7234%
			Nutzer10+:	0,7384%	0,7387%	0,7439%	0,7399%	0,7474%	0,7364%	0,7359%	0,7384%	0,7385%
			Entropie:	0,0104%	0,0104%	0,0105%	0,0105%	0,0105%	0,0105%	0,0105%	0,0105%	0,0104%

P_{fake}	P_{visit}	P_{fixed}		1.Tick	2.Tick	10.Tick	20.Tick	30.Tick	50.Tick	70.Tick	90.Tick	100.Tick
			fullsizeAnon:	102,1824%	94,8871%	66,0066%	56,9469%	48,7957%	47,9163%	46,0771%	46,478%	45,6366%
			fullsizeForgedAnon:	0,0%	1,1719%	1,1719%	2,5187%	3,2104%	3,2104%	3,1198%	5,7608%	5,7608%
			numFingerprints:	0,0%	2,924%	5,0698%	6,2687%	7,5976%	8,9446%	10,3285%	11,2296%	11,6164%
			percentTouched:	122,9773%	107,7035%	71,1691%	55,4004%	47,3455%	45,0267%	42,4108%	40,5038%	40,2377%
			detectableFixed:	300,0%	30,7849%	6,2893%	6,486%	6,486%	6,486%	6,486%	6,486%	6,486%
			numChanges:	-	6,4532%	4,6145%	4,6223%	4,8345%	4,8311%	4,8168%	4,8411%	4,7798%
			sizeForgedAnon:	102,1824%	81,9629%	93,6975%	45,4855%	78,2628%	85,5144%	103,6825%	98,0282%	80,8899%
			anon1:	3,1927%	3,1939%	3,1927%	3,1939%	3,1927%	3,1695%	3,1927%	3,1927%	3,1903%
			anon2:	1,8479%	1,8479%	1,8573%	1,8479%	1,8479%	1,8479%	1,8479%	1,812%	1,8479%
			anon3:	1,7997%	1,7997%	1,7997%	1,7997%	1,7757%	1,7997%	1,7979%	1,7997%	1,8199%
			anon4:	1,7479%	1,7477%	1,7477%	1,7477%	1,7477%	1,7477%	1,7477%	1,7477%	1,7477%
0,01%	100%	50%	anon5:	2,1598%	2,1673%	2,1007%	2,1598%	2,1598%	2,1598%	2,1598%	2,1007%	2,1598%
			anon6:	2,0931%	2,0931%	2,0352%	2,0931%	2,0931%	2,0931%	2,0931%	2,0352%	2,0931%
			anon7:	1,94%	1,9947%	2,007%	1,9947%	2,007%	2,007%	2,007%	2,0207%	2,0207%
			anon8:	2,0895%	2,0895%	2,0895%	2,0895%	2,0714%	2,0895%	2,0895%	2,0895%	2,0895%
			anon9:	2,3783%	2,3783%	2,3783%	2,3783%	2,3783%	2,3783%	2,3783%	2,3783%	2,355%
			anon10:	2,2756%	2,2804%	2,302%	2,302%	2,302%	2,302%	2,302%	2,2756%	2,302%
			Nutzer1:	0,0407%	0,0395%	0,0396%	0,0395%	0,0397%	0,0394%	0,0392%	0,0395%	0,0395%
			Nutzer2:	1,0087%	1,0123%	0,9975%	0,9968%	1,0052%	1,0033%	1,0001%	1,008%	0,9996%
			Nutzer2-9:	0,643%	0,6337%	0,6357%	0,6394%	0,6383%	0,6333%	0,6301%	0,6337%	0,636%
			Nutzer10+:	0,5964%	0,594%	0,5973%	0,6026%	0,5974%	0,5954%	0,5915%	0,5938%	0,5949%
			Entropie:	0,0089%	0,0089%	0,0089%	0,0089%	0,0089%	0,0089%	0,0089%	0,0089%	0,0088%
			fullsizeAnon:	94,064%	76,8267%	59,0508%	57,3236%	55,4346%	53,2394%	52,9686%	52,4534%	52,4136%
			fullsizeForgedAnon:	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
			numFingerprints:	0,0%	5,1838%	12,6154%	14,3914%	17,0497%	20,4272%	21,9219%	23,8154%	23,9776%
			percentTouched:	105,6362%	84,8597%	62,3004%	59,9543%	57,4585%	55,1598%	54,7728%	54,1866%	54,1432%
			detectableFixed:	12,0364%	7,8177%	8,572%	8,572%	8,572%	8,572%	8,572%	8,572%	8,572%
			numChanges:	-	47,8835%	33,4259%	32,4099%	32,1341%	32,1847%	31,8768%	33,1013%	33,2564%
			sizeForgedAnon:	94,064%	92,5661%	95,8968%	97,4391%	92,705%	94,4696%	101,862%	96,785%	97,2487%
			anon1:	1,8718%	1,8718%	1,8718%	1,8718%	1,8718%	1,8718%	1,8718%	1,8718%	1,8718%
			anon2:	1,8093%	1,8093%	1,8093%	1,8246%	1,8093%	1,8093%	1,8093%	1,8093%	1,8093%
			anon3:	1,5656%	1,5656%	1,5656%	1,5656%	1,5656%	1,5656%	1,5656%	1,5656%	1,5656%
			anon4:	1,6474%	1,6474%	1,6474%	1,6474%	1,6474%	1,6474%	1,6474%	1,6474%	1,6474%
0,01%	100%	90%	anon5:	2,016%	2,016%	2,016%	2,016%	2,016%	2,016%	2,016%	2,016%	2,016%
			anon6:	1,1618%	1,1618%	1,1618%	1,1618%	1,1618%	1,1618%	1,1618%	1,1618%	1,1618%
			anon7:	1,3299%	1,3299%	1,3299%	1,3299%	1,3299%	1,3299%	1,3299%	1,3299%	1,3299%
			anon8:	1,7583%	1,7583%	1,7583%	1,7583%	1,7583%	1,7583%	1,7583%	1,7583%	1,7583%
			anon9:	1,8978%	1,8978%	1,8978%	1,8978%	1,8978%	1,8978%	1,8978%	1,8978%	1,8978%
			anon10:	3,015%	3,015%	3,015%	3,015%	3,015%	3,015%	3,015%	3,015%	3,015%
			Nutzer1:	0,0552%	0,0554%	0,0553%	0,0552%	0,0552%	0,0553%	0,0552%	0,0552%	0,0553%
			Nutzer2:	0,698%	0,7117%	0,7132%	0,7126%	0,7198%	0,715%	0,7189%	0,7237%	0,7138%
			Nutzer2-9:	0,3139%	0,312%	0,3087%	0,3051%	0,3068%	0,3043%	0,3086%	0,3087%	0,3089%
			Nutzer10+:	0,7096%	0,7054%	0,7076%	0,7125%	0,7111%	0,7077%	0,7083%	0,7066%	0,7074%
			Entropie:	0,0148%	0,0148%	0,0148%	0,0148%	0,0148%	0,0148%	0,0148%	0,0148%	0,0148%
			fullsizeAnon:	100,4623%	78,5552%	31,1821%	29,5491%	18,7716%	13,0813%	8,8271%	9,5902%	8,6623%
			fullsizeForgedAnon:	1,2448%	2,5064%	5,0343%	7,8514%	8,8541%	11,7682%	13,3596%	13,9529%	14,6241%
			numFingerprints:	0,0%	0,0%	0,0577%	0,0288%	0,0289%	0,0266%	0,0227%	0,0281%	0,0339%
			percentTouched:	104,7379%	73,8642%	22,5621%	18,1675%	12,4677%	8,2847%	7,2833%	7,2715%	6,9335%
			detectableFixed:	-	48,1299%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
			numChanges:	-	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0064%
			sizeForgedAnon:	100,4623%	114,5215%	77,0617%	92,2079%	71,6555%	69,5948%	110,1514%	66,566%	75,454%
			anon1:	2,9446%	2,9171%	2,9353%	2,9298%	2,9396%	2,9594%	2,9446%	2,9353%	2,9498%
			anon2:	1,9046%	1,9046%	1,9046%	1,9046%	1,9283%	1,9046%	1,9046%	1,9046%	1,892%
			anon3:	1,9558%	2,019%	1,9558%	1,9406%	2,019%	1,9648%	1,9558%	1,9558%	1,9558%
			anon4:	2,084%	2,084%	2,084%	2,084%	2,084%	2,084%	2,084%	2,084%	2,012%
0,01%	1%	0%	anon5:	2,1121%	2,1121%	2,1121%	2,1121%	2,1121%	2,0893%	2,0893%	2,1506%	2,1121%
			anon6:	2,3136%	2,2995%	2,3067%	2,2995%	2,2995%	2,2995%	2,2995%	2,2995%	2,3067%
			anon7:	2,3166%	2,2605%	2,2445%	2,3166%	2,2605%	2,2412%	2,2605%	2,2605%	2,2605%
			anon8:	2,2269%	2,2276%	2,2276%	2,2276%	2,2276%	2,2632%	2,2276%	2,2276%	2,2811%
			anon9:	1,9123%	1,9276%	1,9123%	1,9404%	1,9123%	1,9123%	1,9123%	1,8651%	1,8861%
			anon10:	2,4803%	2,4803%	2,5338%	2,4803%	2,4803%	2,4803%	2,4803%	2,5338%	2,4803%
			Nutzer1:	0,0663%	0,0661%	0,066%	0,0662%	0,0661%	0,066%	0,0662%	0,0661%	0,0661%
			Nutzer2:	0,8597%	0,8497%	0,8568%	0,8634%	0,8613%	0,859%	0,8503%	0,8628%	0,8626%
			Nutzer2-9:	0,5718%	0,568%	0,5686%	0,5738%	0,5698%	0,5688%	0,5684%	0,5722%	0,5709%
			Nutzer10+:	0,7566%	0,7547%	0,7586%	0,7591%	0,757%	0,7592%	0,7538%	0,7541%	0,7531%
			Entropie:	0,0132%	0,0131%	0,0131%	0,0132%	0,0131%	0,0131%	0,0131%	0,0131%	0,0131%
			fullsizeAnon:	69,2849%	47,656%	28,8589%	25,4438%	27,7788%	23,2742%	25,2901%	23,3285%	22,0938%
			fullsizeForgedAnon:	0,0%	1,8882%	6,9324%	11,2246%	12,4922%	13,909%	12,6726%	12,9155%	13,5752%
			numFingerprints:	0,0%	0,9012%	1,1091%	1,172%	1,1198%	1,1327%	1,176%	1,1815%	1,2055%
			percentTouched:	71,245%	55,6195%	21,2339%	27,7041%	24,0674%	21,3609%	20,1724%	19,4715%	18,1716%
			detectableFixed:	-	36,1485%	1,3907%	0,6694%	0,578%	0,578%	0,578%	0,0%	0,0%
			numChanges:	-	1,8127%	1,2628%	1,2573%	1,17%	1,0893%	1,1269%	1,1047%	1,1074%
			sizeForgedAnon:	69,2849%	68,4483%	82,6973%	65,7724%	47,4959%	51,7263%	115,527%	90,399%	80,1966%
			anon1:	3,9588%	3,9631%	3,9793%	3,9631%	3,9631%	3,9631%	3,9545%	3,9573%	3,9428%
			anon2:	3,0407%	3,0407%	3,0374%	2,9958%	3,0374%	3,0374%	3,0374%	3,0374%	3,0374%
			anon3:	1,81%	1,8377%	1,8377%	1,8377%	1,8377%	1,8377%	1,8499%	1,8377%	1,8377%
			anon4:	1,9955%	1,9994%	1,9994%	1,9994%	2,0202%	2,0202%	1,9994%	1,9994%	1,9994%
0,01%	1%	10%	anon5:	2,1972%	2,2016%	2,1972%	2,1972%	2,1972%	2,1972%	2,1972%	2,2025%	2,1972%
			anon6:	1,6811%	1,7124%	1,6811%	1,632%	1,6811%	1,6811%	1,7187%	1,6811%	1,6811%
			anon7:	2,1745%	2,1745%	2,1565%	2,1952%	2,1849%	2,2188%	2,1745%	2,1745%	2,209%
			anon8:	2,2693%	2,2368%	2,2368%	2,2368%	2,2368%	2,2368%	2,2693%	2,2368%	2,203%
			anon9:	2,174%	2,1953%	2,1953%	2,225%	2,1953%	2,174%	2,1953%	2,1953%	2,2217%
			anon10:	2,498%	2,4991%	2,498%	2,498%	2,498%	2,498%	2,4991%	2,5123%	2,498%
			Nutzer1:	0,076%	0,0779%	0,078%	0,0784%	0,078%	0,0781%	0,078%	0,0781%	0,0777%
			Nutzer2:	0,9883%	0,9914%	0,9841%	1,0014%	1,0058%	1,0%	1,0029%	0,9859%	0,9828%
			Nutzer2-9:	0,4763%	0,4848%	0,4839%	0,4855%	0,4822%	0,4787%	0,4858%	0,4818%	0,477%
			Nutzer10+:	0,6451%	0,6512%	0,6464%	0,6591%	0,6452%	0,6481%	0,6353%	0,6524%	0,6481%
			Entropie:	0,0155%	0,0157%	0,0157%	0,0157%	0,0157%	0,0157%	0,0157%	0,0157%	0,0157%

p_{fake}	p_{visit}	p_{fixed}		1.Tick	2.Tick	10.Tick	20.Tick	30.Tick	50.Tick	70.Tick	90.Tick	100.Tick
			fullsizeAnon:	105,8211%	69,6727%	40,9033%	36,1258%	34,4084%	34,08%	34,0831%	32,5382%	32,1589%
			fullsizeForgedAnon:	1,8345%	1,8345%	2,8194%	4,5195%	5,6432%	8,8738%	11,0944%	13,2335%	13,2944%
			numFingerprints:	0,0%	4,9593%	11,9935%	12,857%	13,8199%	14,6206%	15,1929%	15,4064%	15,4276%
			percentTouched:	76,5972%	49,0986%	36,5581%	32,3286%	29,7191%	27,5705%	26,0422%	24,3575%	23,8528%
			detectableFixed:	-	47,0548%	11,8814%	9,8344%	9,0551%	7,5161%	5,8562%	5,4477%	5,4477%
			numChanges:	-	13,5461%	14,2469%	14,1649%	14,5656%	14,8368%	14,8171%	14,7129%	14,6133%
			sizeForgedAnon:	105,8211%	53,069%	99,045%	79,8255%	68,294%	51,3528%	86,9255%	48,7256%	69,1563%
			anon1:	3,3914%	3,3994%	3,3847%	3,4144%	3,3797%	3,4161%	3,3896%	3,3905%	3,3946%
			anon2:	3,7328%	3,7138%	3,7429%	3,7328%	3,7017%	3,7328%	3,7328%	3,7619%	3,7328%
			anon3:	2,0775%	2,0442%	2,021%	2,021%	2,0007%	2,021%	2,0775%	2,0442%	2,0442%
0,01%	1%	50%	anon4:	2,6124%	2,6076%	2,6129%	2,6124%	2,6076%	2,6124%	2,6076%	2,6156%	2,6344%
			anon5:	1,882%	1,882%	1,8834%	1,8834%	1,9481%	1,8834%	1,8834%	1,8491%	1,935%
			anon6:	2,1256%	2,1396%	2,1414%	2,1396%	2,1269%	2,1414%	2,1414%	2,1414%	2,1414%
			anon7:	1,9019%	1,9188%	1,8899%	1,9188%	1,9188%	1,8966%	1,8939%	1,9188%	1,9244%
			anon8:	1,978%	1,9568%	1,9846%	1,9631%	1,978%	1,9568%	1,9846%	1,978%	1,9631%
			anon9:	2,2565%	2,228%	2,2565%	2,2098%	2,2565%	2,2886%	2,2565%	2,2565%	2,1988%
			anon10:	1,1861%	1,1861%	1,1861%	1,1861%	1,2014%	1,1861%	1,1861%	1,1861%	1,1861%
			Nutzer1:	0,0696%	0,069%	0,0695%	0,0697%	0,0697%	0,0699%	0,0694%	0,0695%	0,0699%
			Nutzer2:	1,2344%	1,2275%	1,2368%	1,232%	1,2475%	1,2489%	1,2354%	1,2406%	1,2302%
			Nutzer2-9:	0,5994%	0,5974%	0,6068%	0,6109%	0,6118%	0,6068%	0,597%	0,5986%	0,6009%
			Nutzer10+:	0,6849%	0,6825%	0,6808%	0,6778%	0,684%	0,6841%	0,6774%	0,6834%	0,6775%
			Entropie:	0,0128%	0,0126%	0,0128%	0,0127%	0,0127%	0,0127%	0,0127%	0,0127%	0,0127%
			fullsizeAnon:	103,8253%	102,2853%	99,362%	119,5561%	118,3997%	120,155%	118,9652%	121,6939%	122,9352%
			fullsizeForgedAnon:	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
			numFingerprints:	0,0%	3,7213%	15,6321%	25,2415%	30,9775%	39,1721%	43,9556%	50,2231%	52,8351%
			percentTouched:	111,8261%	109,0208%	102,9095%	122,2427%	120,7381%	121,8865%	120,404%	122,92%	124,0997%
			detectableFixed:	15,0298%	11,8164%	12,3724%	11,6814%	10,7507%	9,7219%	9,389%	9,5426%	9,4701%
			numChanges:	-	96,5708%	67,5982%	72,9323%	73,2358%	69,9439%	70,9421%	72,1615%	72,3317%
			sizeForgedAnon:	103,8253%	68,6347%	111,8616%	69,5672%	109,1836%	65,3407%	95,8486%	60,0234%	99,3962%
			anon1:	2,1727%	2,1581%	2,1727%	2,1581%	2,1727%	2,1581%	2,1727%	2,1581%	2,1727%
			anon2:	3,3347%	3,3347%	3,3347%	3,3347%	3,3347%	3,3347%	3,3347%	3,3347%	3,3347%
			anon3:	2,456%	2,456%	2,456%	2,456%	2,456%	2,456%	2,456%	2,456%	2,456%
0,01%	1%	90%	anon4:	2,2399%	2,2399%	2,2399%	2,2399%	2,2399%	2,2399%	2,2399%	2,2399%	2,2399%
			anon5:	1,6164%	1,6164%	1,6164%	1,6164%	1,6164%	1,6164%	1,6164%	1,6164%	1,6164%
			anon6:	1,9344%	1,9344%	1,916%	1,9485%	1,916%	1,9485%	1,9344%	1,916%	1,9344%
			anon7:	2,6512%	2,6512%	2,6512%	2,6512%	2,6512%	2,6512%	2,6512%	2,6512%	2,6512%
			anon8:	2,9089%	2,9089%	2,9089%	2,9089%	2,9089%	2,9089%	2,9089%	2,9089%	2,9089%
			anon9:	2,7776%	2,7776%	2,7776%	2,7776%	2,7776%	2,7776%	2,7776%	2,7776%	2,7776%
			anon10:	3,1844%	3,1844%	3,1844%	3,1844%	3,1844%	3,1844%	3,1844%	3,1844%	3,1844%
			Nutzer1:	0,0617%	0,0616%	0,0617%	0,062%	0,0618%	0,0618%	0,0618%	0,0619%	0,0619%
			Nutzer2:	0,9665%	0,9664%	0,9711%	0,9773%	0,9733%	0,9773%	0,9771%	0,975%	0,9768%
			Nutzer2-9:	0,6046%	0,6047%	0,6083%	0,6107%	0,6101%	0,6136%	0,6113%	0,6126%	0,6127%
			Nutzer10+:	0,774%	0,7743%	0,7755%	0,7748%	0,7745%	0,7762%	0,7743%	0,7747%	0,775%
			Entropie:	0,0108%	0,0108%	0,0108%	0,0108%	0,0108%	0,0108%	0,0107%	0,0107%	0,0108%
			fullsizeAnon:	92,6601%	77,9357%	33,4046%	32,825%	21,2063%	21,713%	20,3638%	14,3069%	15,0524%
			fullsizeForgedAnon:	0,0%	12,2206%	16,2959%	11,9468%	13,1799%	13,6004%	13,9272%	13,9593%	13,9776%
			numFingerprints:	0,0%	0,0%	0,0%	0,0273%	0,058%	0,0617%	0,0524%	0,0419%	0,0403%
			percentTouched:	103,4887%	90,1648%	33,8343%	24,7199%	15,2473%	11,9657%	8,7385%	8,8462%	8,6098%
			detectableFixed:	-	51,5557%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
			numChanges:	-	0,0%	0,0%	0,0%	0,0%	0,0125%	0,0089%	0,0069%	0,0062%
			sizeForgedAnon:	92,6601%	26,1787%	89,5101%	64,5849%	98,3855%	73,2715%	143,5452%	88,2475%	97,1608%
			anon1:	3,1642%	3,1665%	3,1665%	3,1789%	3,1889%	3,1665%	3,1499%	3,1624%	3,1665%
			anon2:	3,378%	3,3982%	3,3982%	3,3982%	3,3982%	3,3982%	3,3982%	3,3982%	3,5641%
			anon3:	2,7346%	2,7419%	2,7419%	2,7419%	2,7927%	2,7419%	2,7419%	2,707%	2,7419%
0,01%	0,01%	0%	anon4:	2,5126%	2,5126%	2,5126%	2,5126%	2,5126%	2,5156%	2,5126%	2,5126%	2,5126%
			anon5:	2,588%	2,5889%	2,5889%	2,5889%	2,5889%	2,5889%	2,5889%	2,5889%	2,575%
			anon6:	1,9487%	1,9289%	1,9289%	1,9289%	2,0374%	1,9289%	1,9289%	1,9823%	1,934%
			anon7:	2,8438%	2,9007%	2,9007%	2,9007%	2,9172%	2,9007%	2,9007%	2,9007%	2,9007%
			anon8:	2,1767%	2,1767%	2,1767%	2,1352%	2,2057%	2,1767%	2,1767%	2,1767%	2,1767%
			anon9:	1,9349%	1,9349%	1,9107%	1,9349%	1,9657%	1,9349%	1,9349%	1,974%	1,9349%
			anon10:	1,7348%	1,7348%	1,7348%	1,6866%	1,7348%	1,7512%	1,7348%	1,6417%	1,7348%
			Nutzer1:	0,0467%	0,0469%	0,047%	0,0466%	0,0469%	0,0469%	0,0465%	0,0464%	0,0467%
			Nutzer2:	1,1658%	1,1804%	1,1787%	1,1475%	1,1882%	1,1585%	1,1696%	1,1622%	1,1783%
			Nutzer2-9:	0,675%	0,6719%	0,6714%	0,6649%	0,6646%	0,6648%	0,6607%	0,6626%	0,6589%
			Nutzer10+:	0,7549%	0,7621%	0,7575%	0,7559%	0,7533%	0,7519%	0,7505%	0,7576%	0,7447%
			Entropie:	0,0143%	0,0143%	0,0144%	0,0143%	0,0143%	0,0144%	0,0143%	0,0143%	0,0144%
			fullsizeAnon:	92,4215%	54,3741%	33,6314%	21,3575%	22,4565%	21,266%	16,2112%	11,6445%	11,0069%
			fullsizeForgedAnon:	0,0%	14,1272%	14,0762%	15,9549%	16,2108%	18,5243%	18,3759%	18,2408%	18,4398%
			numFingerprints:	0,0%	1,7398%	4,2468%	4,4189%	4,1922%	4,2161%	4,3085%	4,3386%	4,3926%
			percentTouched:	107,0649%	70,9187%	29,195%	16,607%	14,9088%	12,9028%	12,7615%	10,7631%	10,9525%
			detectableFixed:	-	33,0393%	2,7589%	2,7738%	2,8183%	1,9763%	1,793%	1,793%	1,793%
			numChanges:	-	3,7204%	4,7801%	4,6651%	4,3606%	4,2392%	4,3539%	4,3399%	4,3627%
			sizeForgedAnon:	92,4215%	91,3039%	75,7131%	91,0139%	154,2324%	134,1263%	86,8442%	94,2119%	64,2242%
			anon1:	1,5299%	1,5379%	1,5066%	1,5299%	1,534%	1,5013%	1,5338%	1,5285%	1,5285%
			anon2:	1,6077%	1,6077%	1,6077%	1,5521%	1,6077%	1,6077%	1,6077%	1,6077%	1,6077%
			anon3:	1,7375%	1,7037%	1,7804%	1,7375%	1,7375%	1,7375%	1,7375%	1,7179%	1,7375%
0,01%	0,01%	10%	anon4:	2,1453%	2,1444%	2,1444%	2,1444%	2,1444%	2,1444%	2,1444%	2,1842%	2,1709%
			anon5:	1,9824%	1,9824%	1,9824%	1,9824%	1,9824%	1,9824%	1,9824%	1,9525%	1,9824%
			anon6:	2,0315%	2,0315%	2,0315%	2,0315%	2,0315%	2,0315%	2,0315%	2,0315%	2,0315%
			anon7:	1,8276%	1,8276%	1,8276%	1,8276%	1,8276%	1,8276%	1,978%	1,8276%	1,8276%
			anon8:	2,0882%	2,0882%	2,0882%	2,0882%	2,1878%	2,0882%	2,0882%	2,0882%	2,0882%
			anon9:	1,8004%	1,8689%	1,8004%	1,8004%	1,8004%	1,8004%	1,8004%	1,8004%	1,8004%
			anon10:	2,0226%	2,0226%	2,0226%	2,0226%	2,0226%	2,0226%	2,0226%	2,0226%	2,0226%
			Nutzer1:	0,0811%	0,0811%	0,0813%	0,0815%	0,0816%	0,0817%	0,082%	0,0816%	0,0816%
			Nutzer2:	0,8415%	0,8278%	0,8332%	0,8467%	0,8357%	0,8384%	0,8514%	0,8399%	0,845%
			Nutzer2-9:	0,7336%	0,7317%	0,7288%	0,7308%	0,72%	0,7277%	0,7447%	0,7277%	0,7217%
			Nutzer10+:	0,6019%	0,6091%	0,6053%	0,6026%	0,5989%	0,6032%	0,6097%	0,5991%	0,6058%
			Entropie:	0,0138%	0,0139%	0,0139%	0,0139%	0,0138%	0,0139%	0,0138%	0,0138%	0,0139%

p_{fake}	p_{visit}	p_{fixed}		1.Tick	2.Tick	10.Tick	20.Tick	30.Tick	50.Tick	70.Tick	90.Tick	100.Tick
0,01%	0,01%	50%	fullsizeAnon:	110,9461%	96,1405%	72,9871%	61,1794%	54,7328%	46,2841%	39,8065%	38,5639%	39,1767%
			fullsizeForgedAnon:	1,2987%	3,726%	8,7281%	10,738%	11,1572%	9,6021%	8,7826%	8,8165%	11,9745%
			numFingerprints:	0,0%	3,3687%	8,1952%	8,9634%	10,1082%	10,4728%	10,123%	10,7186%	11,0792%
			percentTouched:	122,9056%	98,2669%	57,7986%	51,1804%	46,3572%	39,1179%	33,4616%	31,9234%	32,1391%
			detectableFixed:	-	104,7617%	22,9211%	10,3381%	10,5084%	14,5835%	14,0775%	13,7028%	11,9769%
			numChanges:	-	30,2304%	15,5708%	12,7987%	12,8711%	12,0827%	10,8305%	11,0301%	11,2074%
			sizeForgedAnon:	110,9461%	100,7497%	97,3115%	74,5939%	119,909%	114,2001%	40,2024%	76,1948%	76,7728%
			anon1:	2,8655%	2,8781%	2,8655%	2,8815%	2,8655%	2,8818%	2,8818%	2,868%	2,8818%
			anon2:	2,6559%	2,6559%	2,7597%	2,7073%	2,6559%	2,6559%	2,6663%	2,6559%	2,6517%
			anon3:	2,139%	2,139%	2,139%	2,1747%	2,1747%	2,139%	2,2126%	2,1747%	2,2083%
			anon4:	2,0572%	2,0467%	2,0572%	2,0572%	2,0572%	2,0572%	2,0572%	2,0572%	2,0572%
			anon5:	1,9525%	1,9375%	1,9525%	1,9525%	1,9375%	1,9525%	1,9375%	1,9525%	1,9146%
			anon6:	1,7531%	1,7531%	1,7751%	1,7144%	1,7531%	1,7313%	1,7087%	1,6917%	1,7144%
			anon7:	2,2808%	2,2808%	2,2808%	2,2808%	2,2808%	2,2448%	2,2448%	2,2448%	2,2808%
			anon8:	2,1394%	2,1394%	2,1394%	2,1394%	2,1394%	2,1394%	2,2009%	2,1394%	2,1394%
			anon9:	1,2307%	1,2307%	1,2678%	1,2307%	1,2678%	1,2307%	1,2307%	1,2307%	1,2307%
			anon10:	1,7881%	1,7736%	1,7736%	1,7736%	1,7736%	1,7736%	1,7736%	1,7736%	1,7736%
			Nutzer1:	0,0714%	0,0715%	0,072%	0,0721%	0,0717%	0,0711%	0,071%	0,0716%	0,0718%
			Nutzer2:	0,7549%	0,753%	0,7586%	0,7576%	0,7731%	0,7661%	0,7639%	0,7765%	0,7792%
			Nutzer2-9:	0,6982%	0,6993%	0,695%	0,7016%	0,7055%	0,7023%	0,6983%	0,7028%	0,7072%
			Nutzer10+:	0,941%	0,9391%	0,938%	0,9402%	0,941%	0,9411%	0,9433%	0,9399%	0,9402%
			Entropie:	0,0176%	0,0176%	0,0176%	0,0176%	0,0176%	0,0176%	0,0177%	0,0176%	0,0176%
0,01%	0,01%	90%	fullsizeAnon:	92,0189%	92,0189%	98,9357%	117,5574%	111,7094%	94,4905%	97,5518%	92,8723%	92,2965%
			fullsizeForgedAnon:	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
			numFingerprints:	0,0%	0,0%	2,0906%	4,1198%	5,0234%	8,0787%	10,7228%	11,2917%	10,9785%
			percentTouched:	111,5871%	111,5871%	119,1806%	138,9616%	130,6948%	112,0178%	114,9626%	109,4328%	108,7958%
			detectableFixed:	13,8208%	13,8208%	14,0389%	14,9196%	14,1854%	15,5374%	16,0779%	14,8377%	14,318%
			numChanges:	-	-	300,0%	267,8504%	208,5464%	147,1448%	142,8749%	125,3739%	118,4495%
			sizeForgedAnon:	92,0189%	92,0189%	93,2656%	73,024%	75,1141%	75,8885%	98,191%	95,2853%	59,3174%
			anon1:	3,0315%	3,0315%	3,0315%	3,0246%	3,0246%	3,0246%	3,0315%	3,0315%	3,0246%
			anon2:	3,1529%	3,1529%	3,1529%	3,1529%	3,1529%	3,1529%	3,1529%	3,1529%	3,1529%
			anon3:	2,5518%	2,5518%	2,5518%	2,5518%	2,5518%	2,5518%	2,5518%	2,5518%	2,5518%
			anon4:	2,1717%	2,1717%	2,1717%	2,1717%	2,1717%	2,1717%	2,1717%	2,1717%	2,1717%
			anon5:	1,7695%	1,7695%	1,7695%	1,7695%	1,7695%	1,7695%	1,7695%	1,7695%	1,7695%
			anon6:	2,1425%	2,1425%	2,1425%	2,1425%	2,1425%	2,1425%	2,1425%	2,1425%	2,1425%
			anon7:	2,13%	2,13%	2,13%	2,13%	2,13%	2,13%	2,13%	2,13%	2,13%
			anon8:	2,6517%	2,6517%	2,6517%	2,6517%	2,6517%	2,6517%	2,6517%	2,6517%	2,6517%
			anon9:	2,5425%	2,5425%	2,5425%	2,4986%	2,4986%	2,5425%	2,5426%	2,5426%	2,4984%
			anon10:	2,8667%	2,8667%	2,8667%	2,8667%	2,8667%	2,8667%	2,8667%	2,8667%	2,8159%
			Nutzer1:	0,067%	0,067%	0,067%	0,067%	0,0669%	0,0671%	0,0671%	0,0671%	0,067%
			Nutzer2:	0,8784%	0,8784%	0,8779%	0,8776%	0,8776%	0,879%	0,8796%	0,8792%	0,8797%
			Nutzer2-9:	0,6377%	0,6377%	0,6379%	0,6382%	0,6382%	0,6392%	0,6396%	0,6397%	0,6396%
			Nutzer10+:	0,3078%	0,3078%	0,3073%	0,3081%	0,308%	0,3077%	0,3076%	0,3078%	0,3075%
			Entropie:	0,0107%	0,0107%	0,0107%	0,0107%	0,0107%	0,0107%	0,0107%	0,0107%	0,0107%

Werte der Hops - Fälschen von zufälligen Fingerprints

p_{fake}	p_{visit}	p_{fixed}		1.Tick	2.Tick	10.Tick	20.Tick	30.Tick	50.Tick	70.Tick	90.Tick	100.Tick
1%	0,01%	90%	touched1hop:	9,1895	11,4895	26,3428	39,7548	50,4568	69,8111	84,3909	96,4576	100,787
			forgedTouched1hop:	1,0789	1,1109	1,3285	1,5268	1,7009	2,0078	2,2268	2,4314	2,505
			touched2hop:	9,1895	13,1808	45,8041	90,8242	127,2951	203,7332	264,9741	326,3986	348,9497
			forgedTouched2hop:	1,0789	1,1279	1,5649	2,1672	2,7153	3,7966	4,6494	5,5646	5,9026
			touched3hop:	9,1895	13,3935	48,2034	107,8349	153,2355	242,3689	320,3021	390,08	416,9062
			forgedTouched3hop:	1,0789	1,13	1,5846	2,3729	3,0441	4,2849	5,3695	6,4183	6,8176
0,01%	100%	0%	touched1hop:	6,4852	12,8499	101,8123	210,9219	311,9248	500,2464	687,2791	871,754	972,7629
			forgedTouched1hop:	1,0	1,0	1,0144	1,0242	1,0288	1,0379	1,0532	1,0754	1,0833
			touched2hop:	6,4852	12,8499	101,8123	210,9219	311,9248	500,2464	687,2791	871,754	972,7629
			forgedTouched2hop:	1,0	1,0	1,0144	1,0242	1,0288	1,0379	1,0532	1,0754	1,0833
			touched3hop:	6,4852	12,8499	101,8123	210,9219	311,9248	500,2464	687,2791	871,754	972,7629
			forgedTouched3hop:	1,0	1,0	1,0144	1,0242	1,0288	1,0379	1,0532	1,0754	1,0833
0,01%	100%	10%	touched1hop:	5,0262	17,2675	87,1948	186,0195	273,5601	442,2124	602,4497	773,3689	853,9927
			forgedTouched1hop:	1,0	1,0	1,0148	1,0239	1,0278	1,0313	1,0487	1,0528	1,0626
			touched2hop:	5,0262	17,2675	87,1948	186,0195	273,5601	442,2124	602,4497	773,3689	857,778
			forgedTouched2hop:	1,0	1,0	1,0148	1,0239	1,0278	1,0313	1,0487	1,0528	1,0626
			touched3hop:	5,0262	17,2675	87,1948	186,0195	273,5601	442,2124	602,4497	773,3689	857,778
			forgedTouched3hop:	1,0	1,0	1,0148	1,0239	1,0278	1,0313	1,0487	1,0528	1,0626
0,01%	100%	50%	touched1hop:	8,5532	13,3703	55,1945	94,9567	136,8116	183,1118	232,2602	301,762	318,0557
			forgedTouched1hop:	1,0	1,0039	1,0039	1,016	1,0285	1,0285	1,037	1,0721	1,0721
			touched2hop:	8,5532	13,3703	55,1945	99,2978	146,3196	194,9628	245,3934	345,366	363,6869
			forgedTouched2hop:	1,0	1,0039	1,0039	1,016	1,0285	1,0285	1,037	1,0917	1,0917
			touched3hop:	8,5532	13,3703	55,1945	99,2978	146,3196	194,9628	245,3934	349,5581	368,0046
			forgedTouched3hop:	1,0	1,0039	1,0039	1,016	1,0285	1,0285	1,037	1,0917	1,0917
0,01%	100%	90%	touched1hop:	6,9089	8,2848	11,1928	12,1587	12,7244	13,1812	13,425	13,5081	13,5864
			forgedTouched1hop:	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
			touched2hop:	6,9089	8,2848	11,1928	12,1587	12,7244	13,1812	13,425	13,5081	13,5864
			forgedTouched2hop:	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
			touched3hop:	6,9089	8,2848	11,1928	12,1587	12,7244	13,1812	13,425	13,5081	13,5864
			forgedTouched3hop:	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
0,01%	1%	0%	touched1hop:	10,2108	23,6791	102,2235	232,8251	370,5556	697,023	1113,5641	1595,0672	1828,8755
			forgedTouched1hop:	1,0042	1,0157	1,0915	1,189	1,3912	1,8927	2,4437	3,1518	3,3739
			touched2hop:	10,2108	23,6791	102,9062	240,1	401,2965	884,0906	1698,6892	2800,6064	3304,1861
			forgedTouched2hop:	1,0042	1,0157	1,0915	1,1927	1,4146	2,2039	3,3705	5,1297	5,7118
			touched3hop:	10,2108	23,6791	102,9062	240,1	401,2965	932,6957	2104,7544	3894,4035	4690,1823
			forgedTouched3hop:	1,0042	1,0157	1,0915	1,1927	1,4146	2,2826	4,1201	7,2717	8,3014
0,01%	1%	10%	touched1hop:	12,2476	19,8998	98,1643	209,6847	339,0206	623,2205	950,5347	1348,1017	1588,6078
			forgedTouched1hop:	1,0	1,0156	1,1358	1,2808	1,4207	1,9003	2,3044	2,9372	3,2836
			touched2hop:	12,2476	19,8998	98,7503	221,2239	379,7006	843,1749	1390,1033	2380,0166	2977,6815
			forgedTouched2hop:	1,0	1,0156	1,1358	1,2915	1,4706	2,2558	2,9226	4,6532	5,6274
			touched3hop:	12,2476	19,8998	98,7503	221,2239	380,737	928,0961	1617,3518	3344,7851	4451,0504
			forgedTouched3hop:	1,0	1,0156	1,1358	1,2915	1,4706	2,4032	3,3555	6,6899	8,5776
0,01%	1%	50%	touched1hop:	20,3553	40,4132	105,9125	176,9921	240,4542	359,3449	452,8007	579,2447	626,7492
			forgedTouched1hop:	1,009	1,009	1,0372	1,065	1,1182	1,2603	1,3936	1,5325	1,5727
			touched2hop:	20,3553	40,4132	105,9125	179,3921	249,3562	427,6205	567,6511	794,0449	893,8096
			forgedTouched2hop:	1,009	1,009	1,0372	1,065	1,1182	1,291	1,4545	1,682	1,7484
			touched3hop:	20,3553	40,4132	105,9125	179,3921	249,3562	427,6205	579,0293	815,4169	915,9014
			forgedTouched3hop:	1,009	1,009	1,0372	1,065	1,1182	1,291	1,4618	1,7002	1,7666
0,01%	1%	90%	touched1hop:	9,1361	10,0396	12,077	14,2901	14,8022	15,7544	16,1799	16,7561	17,0182
			forgedTouched1hop:	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
			touched2hop:	9,1361	10,0396	12,077	14,2901	14,8022	15,7544	16,1799	16,7561	17,0182
			forgedTouched2hop:	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
			touched3hop:	9,1361	10,0396	12,077	14,2901	14,8022	15,7544	16,1799	16,7561	17,0182
			forgedTouched3hop:	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
0,01%	0,01%	0%	touched1hop:	10,0191	20,5346	609,8271	2162,5232	3972,9025	7105,6579	9780,2341	12100,6304	13062,3299
			forgedTouched1hop:	1,0	1,8547	36,9263	46,7262	46,8	46,8	46,8	46,8	46,8
			touched2hop:	10,0191	20,5346	2259,094	4210,6	5814,8	8595,0	10830,1	12841,3	13678,0
			forgedTouched2hop:	1,0	1,8547	46,795	46,8	46,8	46,8	46,8	46,8	46,8
			touched3hop:	10,0191	20,5346	2299,5	4210,6	5814,8	8595,0	10830,1	12841,3	13678,0
			forgedTouched3hop:	1,0	1,8547	46,8	46,8	46,8	46,8	46,8	46,8	46,8
0,01%	0,01%	10%	touched1hop:	7,9412	26,2003	636,0854	2010,0486	3503,0879	6320,8961	8618,5199	10800,3115	11750,1102
			forgedTouched1hop:	1,0	1,8062	27,8696	40,8546	42,647	43,6448	43,9054	44,3956	44,6347
			touched2hop:	7,9412	26,2003	2447,2277	4235,6215	5754,6286	8454,2114	10502,2579	12468,506	13322,8855
			forgedTouched2hop:	1,0	1,8062	42,0085	43,5923	43,781	44,1572	44,1572	44,5375	44,7342
			touched3hop:	7,9412	26,2003	2481,0934	4238,3659	5754,6432	8454,2222	10502,2579	12468,506	13322,8855
			forgedTouched3hop:	1,0	1,8062	42,0674	43,5923	43,781	44,1572	44,1572	44,5375	44,7342
0,01%	0,01%	50%	touched1hop:	8,211	18,4377	71,5865	135,893	201,1584	388,6301	551,6358	720,2817	801,9411
			forgedTouched1hop:	1,0043	1,0413	1,2365	1,5423	1,8086	2,3264	2,6665	3,1779	3,4767
			touched2hop:	8,211	18,4377	80,3932	200,1721	359,6105	741,5687	1169,9769	1589,4365	1746,7344
			forgedTouched2hop:	1,0043	1,0413	1,268	1,7922	2,4096	3,2698	3,9507	4,7925	5,1302
			touched3hop:	8,211	18,4377	80,408	201,707	383,6424	772,8321	1220,6749	1668,9971	1838,4465
			forgedTouched3hop:	1,0043	1,0413	1,268	1,811	2,4828	3,3232	4,0547	4,9229	5,3088
0,01%	0,01%	90%	touched1hop:	7,28	7,28	7,6116	8,5608	9,5686	11,4102	11,9744	12,6254	12,6841
			forgedTouched1hop:	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
			touched2hop:	7,28	7,28	7,6116	8,5608	9,5686	11,4102	11,9744	12,6254	12,6841
			forgedTouched2hop:	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
			touched3hop:	7,28	7,28	7,6116	8,5608	9,5686	11,4102	11,9744	12,6254	12,6841
			forgedTouched3hop:	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0

Varationskoeffizienten der Hops - Fälschen von zufälligen Fingerprints

P_{fake}	P_{visit}	P_{fixed}		1.Tick	2.Tick	10.Tick	20.Tick	30.Tick	50.Tick	70.Tick	90.Tick	100.Tick
1%	0,01%	90%	touched1hop:	8,9559%	27,0993%	32,1928%	21,0179%	21,5227%	15,6713%	14,7179%	13,8109%	13,2547%
			forgedTouched1hop:	1,6482%	4,1097%	10,2366%	8,9596%	10,4617%	8,9709%	9,1783%	9,73%	10,0998%
			touched2hop:	8,9559%	44,1141%	48,9206%	26,9177%	24,259%	19,6362%	17,4937%	15,1054%	14,8766%
			forgedTouched2hop:	1,6482%	6,3342%	20,284%	16,116%	17,0784%	17,0733%	16,7891%	16,6284%	17,8126%
			touched3hop:	8,9559%	46,602%	48,5964%	25,9147%	22,8135%	21,3344%	15,8065%	17,1623%	17,5753%
			forgedTouched3hop:	1,6482%	6,6906%	20,4995%	17,3826%	18,2343%	19,3514%	16,1029%	18,5082%	20,423%
0,01%	100%	0%	touched1hop:	87,6711%	45,8173%	31,0231%	16,9747%	12,1532%	4,5898%	8,0286%	8,6289%	9,4513%
			forgedTouched1hop:	0,0%	0,0%	2,1738%	2,3831%	2,3034%	3,3401%	3,7874%	4,474%	3,882%
			touched2hop:	87,6711%	45,8173%	31,0231%	16,9747%	12,1532%	4,5898%	8,0286%	8,6289%	9,4513%
			forgedTouched2hop:	0,0%	0,0%	2,1738%	2,3831%	2,3034%	3,3401%	3,7874%	4,474%	3,882%
			touched3hop:	87,6711%	45,8173%	31,0231%	16,9747%	12,1532%	4,5898%	8,0286%	8,6289%	9,4513%
			forgedTouched3hop:	0,0%	0,0%	2,1738%	2,3831%	2,3034%	3,3401%	3,7874%	4,474%	3,882%
0,01%	100%	10%	touched1hop:	83,1516%	78,1748%	46,0241%	39,1482%	28,5017%	25,3558%	22,7175%	23,1718%	24,4569%
			forgedTouched1hop:	0,0%	0,0%	3,2244%	3,7805%	3,7037%	4,4468%	4,5532%	4,9342%	6,2627%
			touched2hop:	83,1516%	78,1748%	46,0241%	39,1482%	28,5017%	25,3558%	22,7175%	23,1718%	24,9849%
			forgedTouched2hop:	0,0%	0,0%	3,2244%	3,7805%	3,7037%	4,4468%	4,5532%	4,9342%	6,2627%
			touched3hop:	83,1516%	78,1748%	46,0241%	39,1482%	28,5017%	25,3558%	22,7175%	23,1718%	24,9849%
			forgedTouched3hop:	0,0%	0,0%	3,2244%	3,7805%	3,7037%	4,4468%	4,5532%	4,9342%	6,2627%
0,01%	100%	50%	touched1hop:	102,1824%	95,5106%	69,2605%	63,3728%	60,4726%	58,138%	55,7356%	67,9978%	66,453%
			forgedTouched1hop:	0,0%	1,1719%	1,1719%	3,5603%	4,6773%	4,6773%	4,4543%	10,71%	10,71%
			touched2hop:	102,1824%	95,5106%	69,2605%	69,9567%	68,0711%	65,5737%	62,1438%	88,6575%	86,9738%
			forgedTouched2hop:	0,0%	1,1719%	1,1719%	3,5603%	4,6773%	4,6773%	4,4543%	15,6507%	15,6507%
			touched3hop:	102,1824%	95,5106%	69,2605%	69,9567%	68,0711%	65,5737%	62,1438%	90,7887%	89,067%
			forgedTouched3hop:	0,0%	1,1719%	1,1719%	3,5603%	4,6773%	4,6773%	4,4543%	15,6507%	15,6507%
0,01%	100%	90%	touched1hop:	94,064%	76,8267%	59,0508%	57,3236%	55,4346%	53,2394%	52,9686%	52,4534%	52,4136%
			forgedTouched1hop:	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
			touched2hop:	94,064%	76,8267%	59,0508%	57,3236%	55,4346%	53,2394%	52,9686%	52,4534%	52,4136%
			forgedTouched2hop:	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
			touched3hop:	94,064%	76,8267%	59,0508%	57,3236%	55,4346%	53,2394%	52,9686%	52,4534%	52,4136%
			forgedTouched3hop:	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
0,01%	1%	0%	touched1hop:	100,4623%	78,5474%	31,1976%	32,606%	25,0883%	18,5851%	15,3543%	15,3987%	15,5648%
			forgedTouched1hop:	1,2448%	2,5064%	6,0489%	11,1614%	13,3605%	19,2417%	23,0014%	25,3859%	26,3287%
			touched2hop:	100,4623%	78,5474%	30,9621%	35,4115%	29,6621%	27,7543%	31,4545%	35,2436%	35,2067%
			forgedTouched2hop:	1,2448%	2,5064%	6,0489%	11,9092%	15,2546%	31,8483%	44,0382%	51,7512%	52,3422%
			touched3hop:	100,4623%	78,5474%	30,9621%	35,4115%	29,6621%	31,3001%	49,4313%	58,1339%	57,7221%
			forgedTouched3hop:	1,2448%	2,5064%	6,0489%	11,9092%	15,2546%	37,2385%	61,7814%	81,2638%	83,3946%
0,01%	1%	10%	touched1hop:	69,2849%	45,4411%	26,3855%	32,1493%	34,9997%	34,3187%	32,523%	35,4019%	33,0785%
			forgedTouched1hop:	0,0%	1,8882%	7,8258%	14,4877%	17,0127%	24,4574%	24,8973%	28,4243%	28,7226%
			touched2hop:	69,2849%	45,4411%	26,64%	33,0254%	45,5754%	49,1%	47,9045%	55,8127%	50,2073%
			forgedTouched2hop:	0,0%	1,8882%	7,8258%	15,6525%	20,4543%	36,1703%	40,7986%	52,848%	48,0494%
			touched3hop:	69,2849%	45,4411%	26,64%	33,0254%	45,6539%	57,68%	58,0875%	74,8016%	66,166%
			forgedTouched3hop:	0,0%	1,8882%	7,8258%	15,6525%	20,4543%	40,7441%	54,1902%	77,1161%	72,0896%
0,01%	1%	50%	touched1hop:	105,8211%	74,0183%	47,0714%	48,1222%	49,6667%	52,8947%	53,7155%	57,5866%	56,9867%
			forgedTouched1hop:	1,8345%	1,8345%	2,8194%	5,4243%	6,8428%	15,131%	19,4187%	26,953%	26,8442%
			touched2hop:	105,8211%	74,0183%	47,0714%	49,5629%	53,6989%	71,3716%	74,9605%	84,3227%	80,8697%
			forgedTouched2hop:	1,8345%	1,8345%	2,8194%	5,4243%	6,8428%	18,6499%	26,0368%	41,8617%	41,2851%
			touched3hop:	105,8211%	74,0183%	47,0714%	49,5629%	53,6989%	71,3716%	77,3474%	87,6972%	83,7115%
			forgedTouched3hop:	1,8345%	1,8345%	2,8194%	5,4243%	6,8428%	18,6499%	26,9561%	44,2698%	43,4566%
0,01%	1%	90%	touched1hop:	103,8253%	102,2853%	99,362%	119,5561%	118,3997%	120,155%	118,9652%	121,6939%	122,9352%
			forgedTouched1hop:	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
			touched2hop:	103,8253%	102,2853%	99,362%	119,5561%	118,3997%	120,155%	118,9652%	121,6939%	122,9352%
			forgedTouched2hop:	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
			touched3hop:	103,8253%	102,2853%	99,362%	119,5561%	118,3997%	120,155%	118,9652%	121,6939%	122,9352%
			forgedTouched3hop:	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
0,01%	0,01%	0%	touched1hop:	92,6601%	86,4891%	32,8339%	28,1167%	18,1421%	13,9786%	9,3104%	9,5256%	9,1237%
			forgedTouched1hop:	0,0%	12,2206%	23,2423%	13,9148%	13,8741%	13,8741%	13,8741%	13,8741%	13,8741%
			touched2hop:	92,6601%	86,4891%	32,6016%	24,7199%	15,2473%	11,9657%	8,7385%	8,8462%	8,6098%
			forgedTouched2hop:	0,0%	12,2206%	13,8868%	13,8741%	13,8741%	13,8741%	13,8741%	13,8741%	13,8741%
			touched3hop:	92,6601%	86,4891%	33,8343%	24,7199%	15,2473%	11,9657%	8,7385%	8,8462%	8,6098%
			forgedTouched3hop:	0,0%	12,2206%	13,8741%	13,8741%	13,8741%	13,8741%	13,8741%	13,8741%	13,8741%
0,01%	0,01%	10%	touched1hop:	92,4215%	58,6891%	34,4617%	23,3122%	18,6395%	16,7639%	15,7506%	13,7984%	13,5393%
			forgedTouched1hop:	0,0%	14,1272%	22,1658%	20,216%	20,0755%	19,7534%	19,5126%	18,6343%	19,1842%
			touched2hop:	92,4215%	58,6891%	29,7015%	17,1883%	14,8939%	13,6671%	13,9047%	11,6737%	11,8537%
			forgedTouched2hop:	0,0%	14,1272%	19,1196%	19,5599%	19,7009%	19,2741%	19,2741%	18,474%	19,1458%
			touched3hop:	92,4215%	58,6891%	29,3237%	17,2009%	14,8938%	13,6669%	13,9047%	11,6737%	11,8537%
			forgedTouched3hop:	0,0%	14,1272%	19,0638%	19,5599%	19,7009%	19,2741%	19,2741%	18,474%	19,1458%
0,01%	0,01%	50%	touched1hop:	110,9461%	136,1054%	102,9775%	98,8239%	72,3042%	48,2542%	40,5209%	38,3428%	38,5717%
			forgedTouched1hop:	1,2987%	3,726%	16,1422%	24,0539%	26,285%	24,6773%	22,6575%	17,3518%	22,575%
			touched2hop:	110,9461%	136,1054%	111,1517%	99,1047%	67,8317%	41,3748%	33,811%	34,1743%	33,543%
			forgedTouched2hop:	1,2987%	3,726%	18,7948%	35,0096%	44,7161%	40,0889%	31,9206%	23,5205%	27,8346%
			touched3hop:	110,9461%	136,1054%	111,1202%	98,1659%	64,0544%	40,4828%	31,4765%	33,0151%	32,9102%
			forgedTouched3hop:	1,2987%	3,726%	18,7948%	35,6469%	46,9015%	41,3855%	32,059%	23,2955%	29,1917%
0,01%	0,01%	90%	touched1hop:	92,0189%	92,0189%	98,9357%	117,5574%	111,7094%	94,4905%	97,5518%	92,8723%	92,2965%
			forgedTouched1hop:	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
			touched2hop:	92,0189%	92,0189%	98,9357%	117,5574%	111,7094%	94,4905%	97,5518%	92,8723%	92,2965%
			forgedTouched2hop:	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
			touched3hop:	92,0189%	92,0189%	98,9357%	117,5574%	111,7094%	94,4905%	97,5518%	92,8723%	92,2965%
			forgedTouched3hop:	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%

B

BEIGELEGTE CD

MD5 Hash der CD:

SHA256 Hash der CD: