**Seminararbeit**

# Anonymes Kommunizieren mit Mixminion

aus

*George Danezis, Roger Dingledine & Nick Mathewson*

**Mixminion: Design of a Type III Anonymous Remailer Protocol**

Seminar "Peer-to-Peer Netzwerke"

Claudius Korzen

24. Juli 2009

Institut für Informatik
Lehrstuhl für Rechnernetze und Telematik
Albert-Ludwigs-Universität, Freiburg

# Inhaltsverzeichnis

# 1 Einführung

## 1.1 Motivation

Neben dem Schutz des Inhaltes von Nachrichten durch verschiedene Verschlüsselungsverfahren, besteht ein immer größeres Interesse, auch den Absender und/oder Empfänger einer Nachricht vor außenstehenden Dritten geheimzuhalten. Unter anderem angetrieben durch die Einführung der Vorratsdatenspeicherung[1] werden immer mehr Internetbenutzer motiviert, sog. *Remailer* - einen Internetdienst, der die angesprochene Anonymität für Absender und Empfänger durch Entpersonalisierung der Nachrichten zur Verfügung stellt, zu verwenden.

In dieser Ausarbeitung werden wir insbesondere den Remailer **Mixminion** kennenlernen, der z.T. auf früher entwickelte Remailer aufbaut; aber auch viele neue Standards setzt, um vorhandene Sicherheitslücken zu schließen und so Angriffe auf den angebotenen Dienst, die die Anonymität gefährden, zu erschweren.
Nach einer kurzen Einführung eines Sitzungsmodells führen wir zunächst sog. *Mix-Netzwerke* ein – die entscheidende Grundlage für die meisten Remailer – ehe wir uns konkreten Remailer-Protokollen zuwenden.

## 1.2 Grundlagen

### 1.2.1 Sitzungsmodell

Im Folgenden gehen wir von folgendem Sitzungsmodell aus: *Alice* möchte *Bob* eine Nachricht schicken. Beim Versenden der Nachricht sollen die Schutzziele

(a) **Vertraulichkeit**: Nur Befugte dürfen auf Inhalt der Nachricht zugreifen

(b) **Integrität**: Nachricht wurde nicht von Unbefugten verändert

(c) **Anonymität**: Absender und/oder Empfänger bleiben anonym

(d) **Authentizität**: Nachricht stammt tatsächlich vom Verfasser

eingehalten werden. Die verschickte Nachricht soll also weder von Unbefugten gelesen, noch geändert werden dürfen; zusätzlich soll die Nachricht authentifiziert und anonymisiert sein – niemand außer Alice selbst soll also wissen, von wem die Nachricht stammt.
Die oben genannten Schutzziele werden üblicherweise von möglichen Angreifern gefährdet, weshalb wir in unserem Modell von einem Angreifer *Eve* ausgehen, der das gesamte Netzwerk beobachten, Netzwerk-Traffic analysieren und verschickte Datenpakete abfangen, zurückhalten und manipulieren kann. Diese sehr konservativen Annahmen sind nicht unbedingt realistisch, aber notwendig, um sichere Rückschlüsse auf die Sicherheit der untersuchten Remailer schließen zu können.

[1] Verpflichtung von z.B. Providern, Kommunikationen im Internet (z.B. E-Mail-Verkehr) zu protokollieren und die gesammelten Daten eine bestimmte Zeit lang zu speichern

### 1.2.2 Mix-Netzwerke

Das grundlegende Prinzip von Remailern basiert auf der Idee von David Chaum (Chaum (1981)), ein Netzwerk von sog. *Mixen* zu betreiben. Mixe sind üblicherweise Server, die von versch. Personen oder Institutionen betrieben werden und als Nachrichtenübermittler dienen.
Wenn Alice eine Nachricht $N$ an Bob senden will, sucht sich Alice zunächst eine (endliche) Menge $M = (M_1, M_2, \ldots, M_n)$ von Mixes aus dem Netzwerk aus, über die die Nachricht Schritt für Schritt übertragen wird und letztendlich bei Bob eintrifft (vgl. Abb. 1). Jeder Mix führt definierte Aktionen auf die erhaltene Nachricht aus, ehe die-

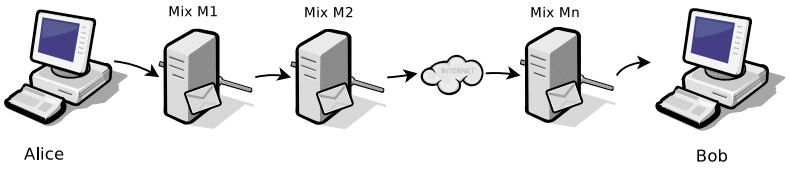



Abbildung 1: Beispiel für ein Mix-Netzwerk

se weitergeleitet wird. So besitzt jeder Mix ein Schlüsselpaar, bestehend aus einem öffentlichem Schlüssel $E_{M_i}$ und einem privaten Schlüssel $D_{M_i}$. Alice verschlüsselt die zu verschickende Nachricht $N$ zusammen mit Routing-Informationen $R_i$ ($i \in [1, n+1]$) nun nacheinander mit den öffentlichen Schlüsseln der ausgewählten Mixes, angefangen mit dem letzten Mix $M_n$ aus $M$:

$$N' = (R_1, E_{M_1}(R_2, E_{M_2}(\ldots(R_n, E_{M_n}(R_{n+1}, E_{Bob}(N)))\ldots)$$

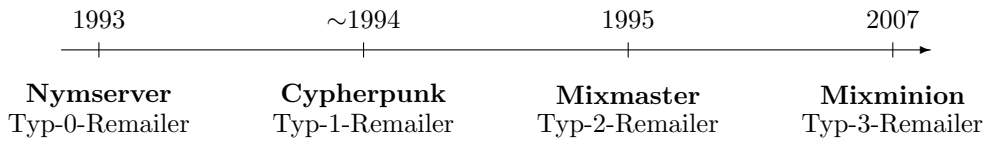
Alice startet das Versenden der Nachricht, indem sie $N'$ an $M_1$ schickt. $M_1$ kann mithilfe von $D_{M_1}$ die erste Verschlüsselungsschicht lösen, woraufhin ihm die Daten $R_2$ und die Nachricht $N'' = E_{M_2}(\ldots(R_n, E_{M_n}(R_{n+1}, E_{Bob}(N)))\ldots)$ zur Verfügung steht, die er an $M_2$ sendet. Die Adresse von $M_2$ wurde der Routing-Information $R_2$ entnommen. $M_2$ entschlüsselt wiederum die zweite Verschlüsselungsschicht und sendet sie an den nächsten Mix. Diese Methode wird solange fortgesetzt, bis schließlich $M_n$ auf $E_{Bob}(N)$ zugreifen kann und diese Nachricht an Bob weiterleitet. Bob kann $E_{Bob}(N)$ nun mit seinem privaten Schlüssel $D_{Bob}$ entschlüsseln und die Nachricht $N$ lesen.
Jeder Mix kennt nur seinen unmittelbaren Vorgänger und seinen unmittelbaren Nachfolger, an den die Nachricht weitergeleitet wird. Deshalb erhält Bob auch keine Kenntnis darüber, wer ihm die Nachricht ursprünglich geschickt hat - für ihn sieht es so aus, als ob die Nachricht von $M_n$ stammt.

Entscheidend bei Mix-Netzwerken ist das Fehlen einer zentralen Instanz, die z.B. den

Ablauf von Sitzungen koordiniert. Dies gestaltet das Netzwerk sehr viel robuster gegen Angriffe, da alle Mixes manipuliert werden müssten, um ein oder mehrere Schutzziel(e) zu gefährden.

# 2 Remailer als Anonymisierungsdienste

Im Laufe der Jahre wurden verschiedene Remailer-Protokolle entwickelt, die größtenteils zwar aufeinander aufbauen, sich aber z.T. erheblich in Funktion und der gebotenen Sicherheit unterscheiden. Der Chronologie entsprechend werden wir zum besseren Verständnis kurz die Prinzipien der Remailer-Protokolle Cypherpunk und Mixmaster kennenlernen, ehe das Hauptaugenmerk auf Mixminion gelegt wird.




Das im Zeitstrahl angebenene Remailer-Protokoll Nymserver bietet statt Anonymität Pseudonymität, indem es E-Mailadressen auf Pseudonyme abbildet. Da es nicht auf einem Mix-Netzwerk basiert, wird dieses Protokoll hier nicht weiter betrachtet.

## 2.1 Typ-1-Remailer: Cypherpunk

Der grundlegende Aufbau von Cypherpunk-Protokollen ist äquivalent zu den Mix-Netzwerken von Chaum (vgl. Kapitel 1.2.2): Alice wählt eine Menge von Mixes aus, verschlüsselt ihre Nachricht nacheinander mit den öffentlichen Schlüsseln dieser Mixe und versendet diese über das Netzwerk. Jeder Mix entschlüsselt eine Schicht mit seinem privaten Schlüssel, bis Mix $M_n$ die Nachricht $E_{Bob}(N)$ an Bob weiterleiten kann. Bob entschlüsselt diese Nachricht und erkennt nur $M_n$ als Absender; er kann also nicht nachvollziehen, von wem die Nachricht ursprünglich stammt.

### 2.1.1 Angriffe gegen Cypherpunk-Protokolle

Bei Verwendung von Cypherpunk-Protokolle ist es Eve relativ leicht möglich, mittels einer Traffic-Analyse den Pfad von Nachrichten durch ein Mix-Netzwerk zu verfolgen. Eve kann alle verschickten Nachrichten beobachten und analysieren, also z.B. den Versendezeitpunkt und die Größe von Nachrichten feststellen. Da die Mixes die Nachrichten in ihrer Größe weitestgehend nicht verändern (Größe der Nachrichten verringert sich bei jedem Entschlüsselungsvorgang nur geringfügig) und die Nachrichten unmittelbar weiter versenden, ist es Eve nun möglicherweise anhand dieser Eigenschaften möglich, die Nachricht zu verfolgen und den Empfänger der Nachricht zu identifizieren (*Traffic-Analyse*).
Zusätzlich könnte Eve die Nachricht von Alice abfangen und immer wieder einspielen,

um Verhaltensmuster von einem Mix zu untersuchen und somit auf den Empfänger zu schließen (*Replay-Angriff*).

## 2.2 Typ-2-Remailer: Mixmaster

Mixmaster baut weitestgehend auf Cypherpunk auf, versucht aber dessen Schwachstellen durch verschiedene Sicherheitsmechanismen zu beheben. So speichert jeder Mix bei Mixmaster-Protokollen Nachrichten zunächst in einem Puffer und verschickt diese – sobald dieser gefüllt ist – in zufälliger Reihenfolge. Dadurch ist es nicht mehr möglich, Nachrichten anhand ihres Versendezeitpunktes zu verfolgen, da dieser nun vom Zufall abhängt. Weiterhin werden Nachrichten in gleich große Blöcke (hier: 20 kB) aufgeteilt, um Nachrichten nicht anhand ihrer Größe verfolgen zu können. Zu kleine Blöcke werden dabei mit Zufallsdaten gefüllt. Damit Bob die komplette Nachricht erhält, ist der letzte Mix dafür zuständig, die Nachricht wieder zusammenzusetzen und schließlich an Bob zu versenden.
Eine Überprüfung der Integrität anhand der Signatur einer Nachricht durch jeden Mix verhindert zusätzlich, dass gefälschte oder wieder eingespielte Nachrichten weitergeleitet werden.

### 2.2.1 Angriffe gegen Mixmaster-Protokolle

Allgemein gelten Mixmaster-Protokolle als sehr sicher, weshalb sie auch eine der beliebtesten Remailer-Protokolle sind. Rein theoretisch sind Mixmaster- Protokolle aber anfällig gegenüber sog. *Flooding-Attacken*:
Eve fängt die Nachricht von Alice ab, bevor diese überhaupt den ersten Mix erreicht. Daraufhin verschickt Eve soviele eigene Nachrichten über die gleichen Mixes, bis deren Puffer gefüllt sind und die Mixes die Nachrichten zur Weiterleitung freigeben. Eve spielt nun die Nachricht von Alice ein. Da Eve den Empfänger seiner eigenen Nachrichten kennt, muss sie nur auf die Nachricht warten, dessen Empfänger von diesem abweicht, um die Nachricht verfolgen zu können. Ein mehrmaliges Anwenden dieses Prinzips kann zur Identifizierung des Empfängers führen.
Wie erwähnt, ist diese Art von Angriff in der Praxis unwahrscheinlich, da Eve sicherstellen müsste, dass tatsächlich nur ihre Nachrichten in den Puffern liegen; zudem müsste sie z.B. die Größe der Puffer von den Mixes kennen. Außerdem müsste sie die Kontrolle über nahezu alle Mixes auf dem Pfad haben, was in der Praxis kaum realisierbar ist.

## 2.3 Typ-3-Remailer: Mixminion

Besonders das eben vorgestellte Protokoll Mixmaster liefert viele Grundlagen für das nun betrachtete Remailer-Protokoll. Obwohl Mixmaster schon als sehr sicher gilt, wurde das Remailer-Protokoll Mixminion entwickelt. Dieses bietet einen entscheidenden Vorteil gegenüber den früheren Remailern Cypherpunk und Mixmaster: Die Möglichkeit des Antwortens auf anonyme Nachrichten. Hierdurch ist es einem Benutzer möglich, auf eine Nachricht zu antworten, obwohl er den Empfänger der Antwort

nicht kennt.
Die bedeutendsten Neuerungen im Überblick:

(a) Antworten auf anonyme Antworten

(b) Verschlüsselten Verbindungen zwischen Mixes: TLS (statt SMTP)

(c) Einführung von Verzeichnisservern

### 2.3.1 Antworten auf anonyme Nachrichten

Wenn Bob auf eine Nachricht von Alice antworten möchte, kann er nicht einfach eine Antwort an Alice senden, da er weder die Email-Adresse noch die Identität von ihr kennt. Um ihr dennoch antworten zu können, verwendet Bob sog. *single-use reply blocks* (SURBs). Ein SURB wird immer vom Absender einer Nachricht, auf die geantwortet werden soll, erstellt und sind aus Sicherheitsgründen nur einmal verwendbar. Bob muss also immer wieder einen neuen SURB verwenden, der von Alice erstellt wurde. Dieser enthält die verschlüsselte E-Mailadresse von Alice sowie einen Pfad durch das Mix-Netzwerk, der zu Alice führt.
Bob antwortet Alice schließlich, indem er seine Nachricht an einen SURB hängt und dem ersten Mix auf dem Pfad durch das Mix-Netzwerk übergibt.

### 2.3.2 Drei verschiedene Nachrichten-Typen

Durch die Hinzunahme von Antwort-Nachrichten unterscheidet Mixminion in 3 verschiedene Nachrichten-Typen:

(a) **"normale" Nachrichten**: Nachrichten, wie sie z.B. schon bei Mixmaster existierten: Alice sendet eine Nachricht an Bob, wobei Bob die Identität von Alice nicht erkennt.

(b) **direkte Antworten**: Bob antwortet auf eine Nachricht von Alice, wobei er nicht weiß, an wen die Antwort gesendet wird.

(c) **anonyme Antworten**: Beide Parteien bleiben anonym; Alice kennt nicht die Identität von Bob und Bob kennt nicht die Identität von Alice.

Aus Sicherheitsgründen müssen die verschieden Typen von Nachrichten untereinander ununterscheidbar sein, da ansonsten Angreifer Rückschlüße auf die Art der Nachrichten schließen und Zusammenhänge zu anderen Nachrichten erstellen könnten.
Nachrichten bestehen in der Regel aus einem *Header* und einer *Payload* (die Teile der eigentlichen Nachricht enthält) und sind 32 kB groß. Ein Pfad wird in zwei Teile aufgeteilt, weshalb auch der Header einer Nachricht in zwei Teile aufgeteilt wird: einem *primären* Header für den ersten Teil des Pfades und einen *sekundären* Header für den zweiten Teil des Pfades.
Die Header enthalten für jeden Mix auf dem Pfad durch das Mix-Netzwerk nochmals einen Subheader, der jeweils eine Prüfsumme über den Rest des Headers enthält. Ein Pfad darf max. 32 Mixes enthalten, weshalb sowohl der primäre als auch der sekundäre

Header max. 16 Subheader enthält. Mit diesen kann später jeder Mix u.a. die Integrität des Pfades überprüfen. Zusätzlich werden in einem Subheader ein *Master Secret* für die Erstellung eines symmetrischen Schlüssels (benötigt für die Verschlüsselung der Nachrichten zur Übertragung) und die Adresse des nächsten Hops gespeichert (vgl. Abb. 2). Bei normalen Nachrichten werden sowohl der primäre als auch der sekundäre

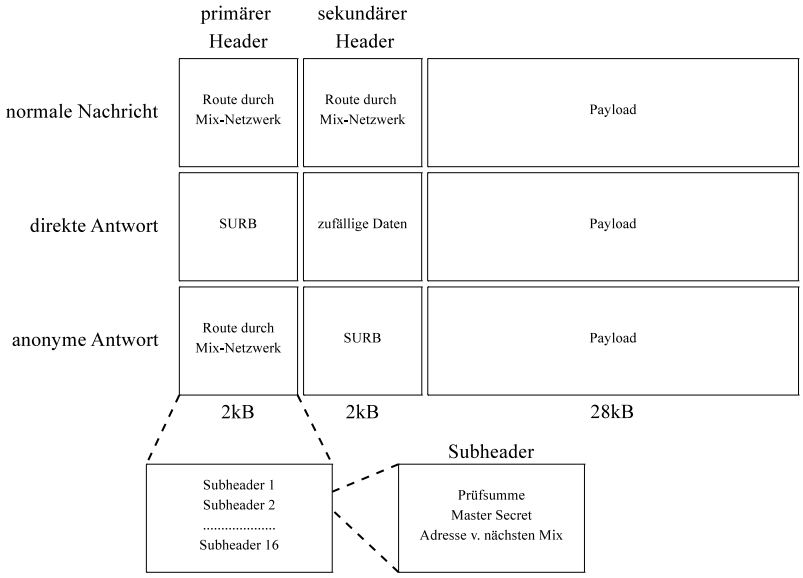



Abbildung 2: Header-Konfiguration der verschiedenen Mixminion-Nachrichten

Headerbereich vom Absender befüllt. Anonyme Antworten enthalten im sekundären Header den SURB des Empfängers, weshalb beide Teilnehmer zunächst ihre SURBs austauschen müssen. Dagegen kann der sekundäre Header bei direkten Antworten mit beliebigen Daten gefüllt werden. Eine Prüfsumme über der Payload wird dafür verwendet, den sekundären Header zu verschlüsseln.

### 2.3.3 Verzeichnisserver

Eine weitere Neuerung gegenüber früheren Remailer-Protokollen ist die Einführung von Verzeichnisservern, die durch eine Gruppe von redundanten Servern gestellt wird. Sie informieren die Benutzer z.B. über den Status und die Schlüssel der Mixe. Die Server müssen hierfür ständig synchronisiert werden, um den aktuellsten Stand des Netzwerkes abzubilden. Ist diese Eigenschaft nicht erfüllt, besteht ein akutes Sicherheitrisiko, wenn Benutzer unterschiedliche Informationen über die Netzwerktopologie haben. Deshalb registriert sich anfangs jeder Mix beim Verzeichnisserver und hält diesen auf den aktuellsten Stand über Status, verwendete Schlüssel, etc. (vgl. Abb. 3). Die Server haben zusätzlich die Aufgabe, sich gegenseitig zu verifizieren. Das heißt,

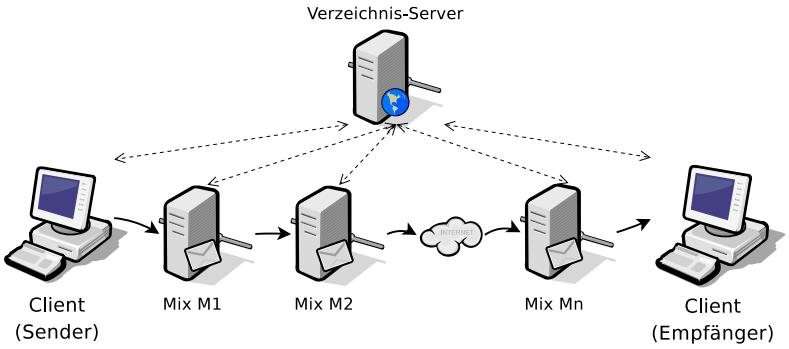



Abbildung 3: Beispiel für ein Mix-Netzwerk mit Verzeichnisserver

dass mehrere unabhängige Server die Glaubwürdigkeit eines jeden Servers bestätigen müssen. Diese Methode beruht auf der Annahme, dass nicht alle Server manipuliert werden, da ansonsten diese Überprüfung für die Benutzer hinfällig wird.

### 2.3.4 Funktionsweise

Bevor Alice eine Nachricht versenden kann, muss sie alle benötigten Informationen, wie z.B. die Verfügbarkeit vom Mixes, zunächst vom Verzeichnisserver einholen, um daraufhin einen Pfad von Mixes auswählen zu können. Das Versenden und Verschlüsseln von Nachrichten geschieht äquivalent zu Mixmaster.
Bevor ein Mix eine Nachricht an einen nächsten Mix weiterleitet, prüft dieser mittels der Prüfsumme im Header die Integrität der Daten und baut eine verschlüsselte Verbindung (über TLS/SSL) zum nächsten Mix auf. Die hierfür benötigten (symmetrischen) Schlüssel werden mithilfe des Master Secrets im Header erstellt. Nachdem die Identität über seine Signatur überprüft wurde und für gültig empfunden wurde, kann die Nachricht weitergeleitet werden. Sobald eine Nachricht übermittelt wurde und die gesicherte Verbindung beendet wurde, werden die erzeugten Schlüssel gelöscht und erneuert.

**Swap-Operation** Wie schon in Kapitel 2.3.2 erwähnt, wird der Pfad durch ein Mix-Netzwerk in zwei Teile aufgeteilt. Zwischen diesen beiden Teilen befindet sich ein sog. *Kreuzungspunkt*. U.a. um die schon kennengelernten Tagging-Attacken zu vermeiden, wird an diesem Kreuzungspunkt eine *swap*-Operation durchgeführt:
Hierbei wird der verschlüsselte, sekundäre Header entschlüsselt und mit dem primären Header vertauscht. Da die Prüfsumme der Payload als Schlüssel für den sekundären Header verwendet wurde, kann dieser nicht mehr entschlüsselt werden, wenn die Payload manipuliert wurde. Die Nachricht kann in diesem Fall nicht weitergeleitet werden.

### 2.3.5 Angriffe gegen Mixminion-Protokolle

Mixminion hat es nie in einen praktischen Einsatz geschafft, da die Entwickler im Jahre 2007 die Weiterentwicklung von Mixminion gestoppt haben. Deshalb existieren auch kaum praktische Erfahrungen mit diesem Protokoll, insbesondere zum Thema Sicherheit. Mixminion befindet sich immer noch in der Testphase, weshalb unentdeckte Fehler im Programm-Code existieren und die Anonymität und Sicherheit für Benutzer gefährden können.
In der Theorie gilt Mixminion als das sicherste Remailer-Protokoll, da neueste Forschungsergebnisse umgesetzt wurden und Mängel früherer Remailer beseitigt wurden. Aus Tabelle 1 kann zusammenfassend entnommen werden, mit welchen Methoden sich Mixminion gegen verschiedenartige Angriffe schützt.

# 3 Zusammenfassung

Wir haben gesehen, wozu Remailer verwendet werden können und welche Sicherheitsanforderungen an diese bestehen, um insbesondere Anonymität zu gewährleisten. Zusätzlich haben wir eine Reihe von möglichen Angriffen kennen gelernt, die ein Mix-Netzwerk gefährden können. Verschiedene Lösungsansätze dienten zur Vermeidung solcher Angriffe, die von den hier vorgestellten Remailer Cypherpunk, Mixmaster und Mixminion auf unterschiedliche Art und Weise umgesetzt wurden. Sicherheitsmängel in den jeweils früheren Remailern machte es allerdings notwendig, sicherere und den aktuellen Standards entsprechenden Protokolle zu entwerfen.
Als das beliebteste und weit verbreiteste Remailer-Protokoll gilt bis heute Mixmaster, obwohl Mixminion zumindest in der Theorie sicherer ist. Da es dieses Protokoll allerdings noch nicht über die Testphase hinaus geschafft hat und eine fehlerfreie Implementierung nicht sichergestellt ist, wird Mixmaster bis auf weiteres das “Standard-Remailer-Protokoll” bleiben.

| **Angriff** | **Verteidigung** |
|---|---|
| Attacken gegen Mixes | |
| Manipulieren eines Mixes | Da Pfad aus mehreren Mixes besteht, keine echte Gefahr (Manipulieren aller Mixe praktisch nicht möglich) |
| Manipulieren eines privaten Schlüssels von Mix | s.o. |
| Wiedereinspielen von Nachrichten | Mixes speichern Prüfsummen der Header: Erkennen von Duplikaten möglich; zusätzl: Nach Erneuerung von Schlüsseln keine Entschlüsselung mehr möglich |
| Nachrichten verzögern, um andere Parameter auszunutzen | Einführen einer Deadline an jedem Mix möglich |
| Nachrichten löschen | Sender muss Nachricht nochmal versenden, sollte dabei gleichen Pfad benutzen |
| Tagging-Attacke | Swap-Operation |
| Flooding-Attacke | Mixes mit Puffern erschweren Angriff |
| Passive Attacken | |
| Traffic-Analyse | Einheitliche Paketgrößen; Zufällige Versende-Reihenfolge durch Mixe; Puffer von Mixen |
| Attacken gegen Verzeichnisserver | |
| Manipulieren eines Verzeichnisservers | Ähnlich wie bei Mixes: Jeder Verzeichnisserver müsste manipuliert werden, damit Gefahr von diesem Angriff ausgeht |
| Ausnutzen von unterschiedlichem Wissen v. Benutzern | Ständige Synchronisierung und Aktualisierung der Verzeichnisserver |

Tabelle 1: Mögliche Angriffe auf Mixminion mit dessen Gegenmaßnahmen

## Literatur

[Chaum 1981] Chaum, David L.: Untraceable electronic mail, return addresses, and digital pseudonyms. In: *Commun. ACM* 24 (1981), Nr. 2, S. 84–90. – ISSN 0001-0782

[Danezis u. a. 2003] Danezis, George ; Dingledine, Roger ; Mathewson, Nick: Mixminion: Design of a Type III Anonymous Remailer Protocol. In: *In Proceedings of the 2003 IEEE Symposium on Security and Privacy*, 2003, S. 2–15

[Kubieziel 2007] Kubieziel, Jens: *Anonym im Netz - Techniken der digitalen Bewegungsfreiheit.* München : Open Source Press, 2007