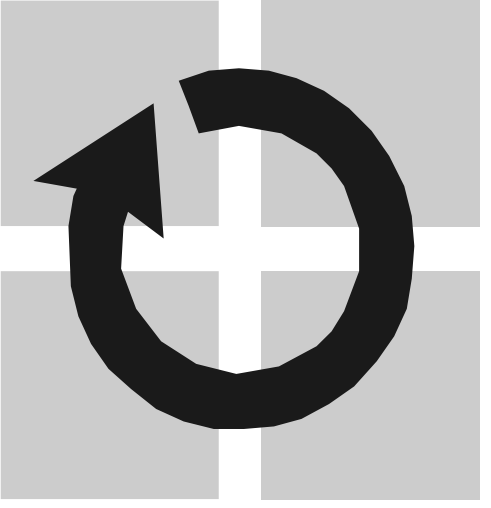
Freehaven und Publius

Anonyme und zensurresistente Verbreitung von Informationen



Christian Hausner
Universität Erlangen-Nürnberg, 2002
christian.hausner@fau.de

Motivation

Warum ist anonymes und zensurresistentes Publizieren wünschenswert?

- Verbreitung neuer (revolutionärer) Ideen
- Aufdecken von Mißständen
- Schutz des Autors / Verlegers
- Erhöhung der Glaubwürdigkeit

Ist das World-Wide-Web ausreichend anonym und zensurresistent?

Nein! ➔ Entwicklung spezieller Informations-Verbreitungs-Systeme

Überblick

I. Designziele

- **i. Anonymität**
- **ii. Zensurresistenz**
- **iii. ...**

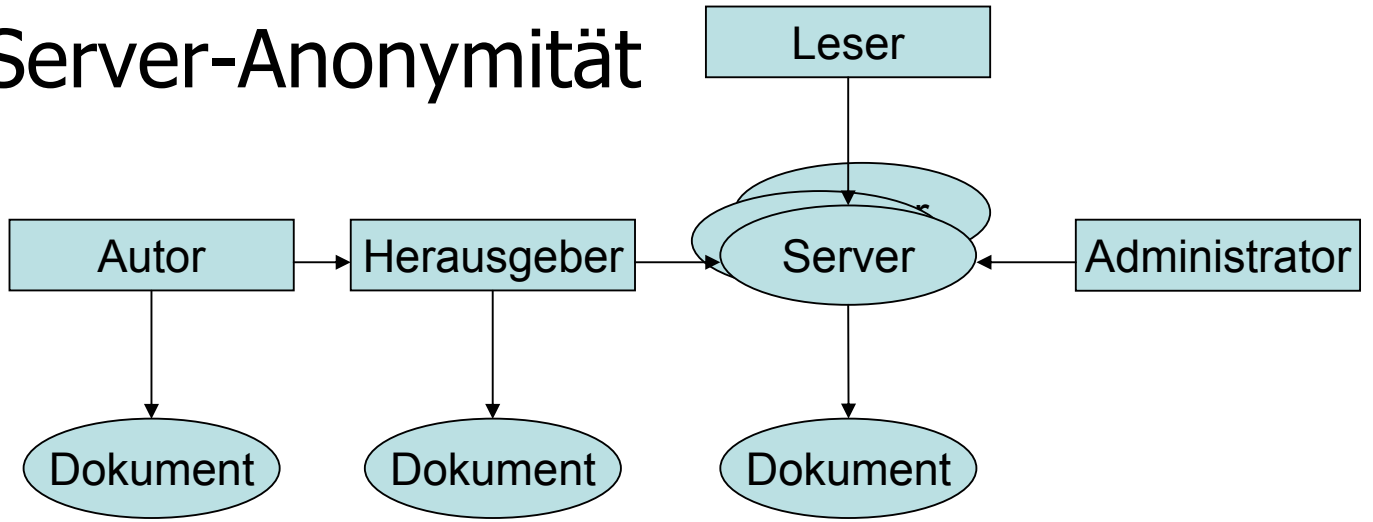
II. Publius

III. Freehaven

IV. Vergleich

Eigenschaften: Anonymität

- Autor-Anonymität
- Herausgeber-Anonymität
- Dokument-Anonymität
- Leser-Anonymität
- Server-Anonymität



Eigenschaften

- Zensurresistenz
  - Angreifern soll jede Möglichkeit genommen werden Informationen zu löschen oder zu verändern
- Sekundäre Designziele
  - Fehlertoleranz: Fehlerhafte oder böswillige Komponenten dürfen keinen Einfluss auf die Verfügbarkeit haben
  - Dezentralität: Kein Single-Point-of-Failure
- Effizienz muss hinten anstehen

Überblick

I. Designziele

II. Publius

- **i. Systemarchitektur**
- **ii. Benutzung**
- **iii. Sicherheitsmechanismen**
- **iv. Implementation**

III. Freehaven

IV. Vergleich

Publius: Systemarchitektur

- Große Anzahl weit verteilter Webserver
- Systemweit verfügbare Liste der Publius-Knoten
- Kommunikation über anonyme Proxy-Server

Publius: Secret-Sharing-Algorithmen

- Informationen können wie folgt aufgespalten werden:
  - n Bruchstücke werden erzeugt
  - k Bruchstücke zur Rekonstruktion (k<n) nötig
  - k-1 Bruchstücke erlauben keinerlei Rückschluss auf die Informationen
  - Typische Werte: n=30; k=3
- Bsp.:
  - Shamir-Secret-Sharing-Algorithmus
  - Information Dispersal Algorithmus

Publius: Veröffentlichung

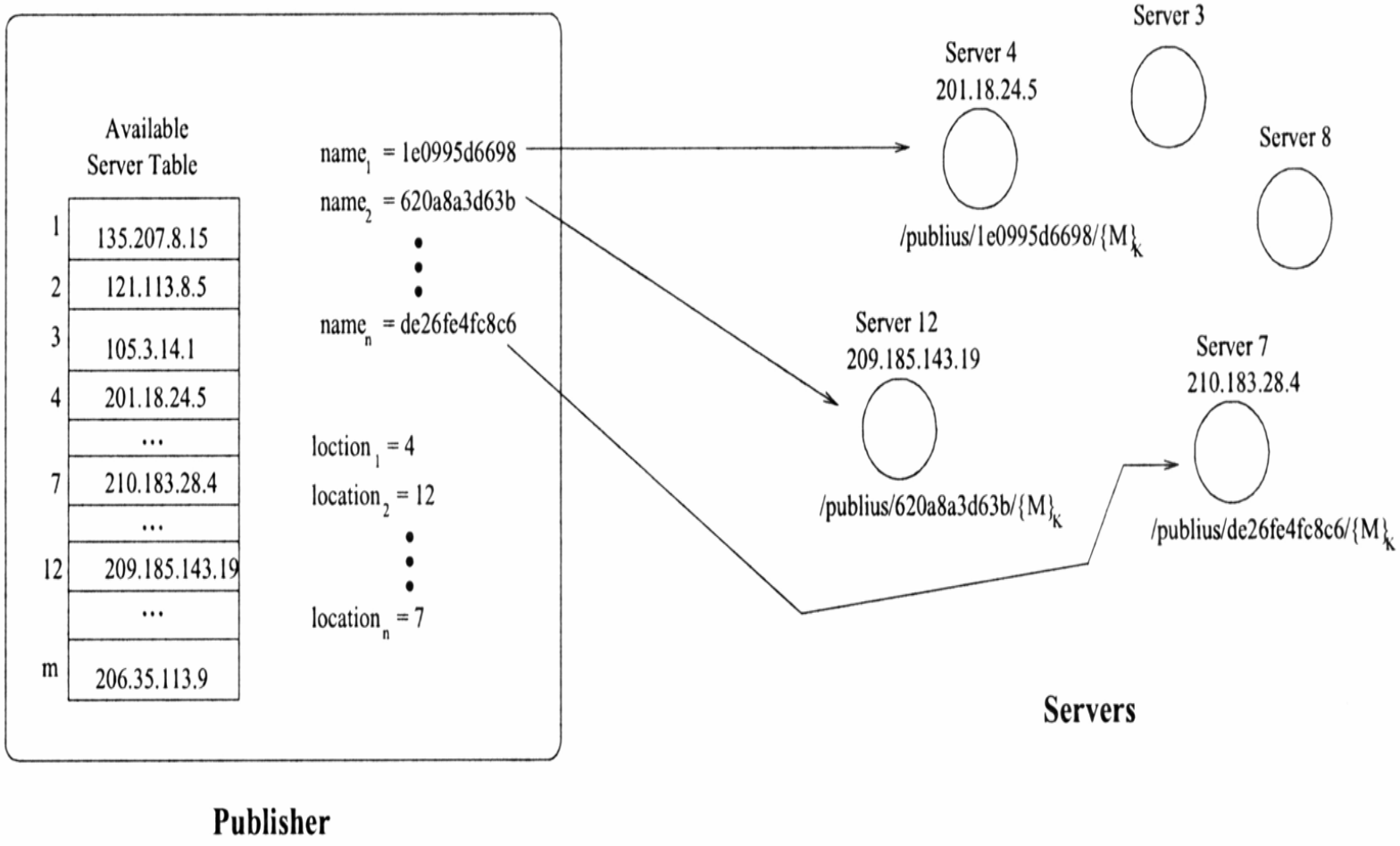
- Erzeugung eines symmetrischen Schlüssels
- Aufspalten des Schlüssels mittels Secret-Sharing-Algorithmus
- MD5-Hash über Kombination von Bruchstück und Inhalten
- Ermittlung von Servern aus der Liste anhand der Hash-Werte
- Upload in Verzeichnis mit dem Namen des Hash-Werts unter den Namen ‚file' bzw. ‚share'

Publius: URL

- Erzeugung einer Publius-URL aus:
  - Prefix
  - Optionen
  - Hashwerten

```
http://!publius!/072LyMOBWJrDw=GT
EaS2GlNNE=NIBsZlvUQP4=sVfdKF7o/kl
=EfUTWGQU7LX=OCk7tkhWTUe=GzWiJyio
75b=QUiNhQWyUW2=fZAX/MJnq67=y4enf
3cLK/0=
```

Publius: Veröffentlichung II



Publius: Abrufen

- Extrahieren von k zufälligen Hash-Werten bzw. Verzeichnisnamen aus URL
- Berechnen von k Serverindizes aus den Hash-Werten
- Download von k Bruchstücken und einer Version der Inhalte
- Rekonstruktion des Schlüssels
- Dechiffrierung der Inhalte

Publius: Update

- Passwortgeschütztes Update: Datei ‚password' mit Hash-Wert des Autor-Passworts
- Änderungen der Inhalte führen zu neuer Publius-URL

➔ Erstellen einer Datei namens ‚update'; Umleitung zu neuer Publius-URL

Publius: Löschen

- Authentifikation mittels Hash-Wert des Passworts
- ‚do not delete'-Option zur Vermeidung von sozialem/rechtlichen Druck auf Autoren und Verleger

Publius: Sicherheitsmechanismen

- Fehlertoleranz und Zensurresistenz durch geeignete Wahl der Aufspaltungs-Parameter (n,k)
- Hash-Werte über Inhalte ➔ Überprüfung der Informationen auf Veränderungen

Publius: Implementation

- Benutzung von HTTP
- Spezielle URLs für Operationen
- Server: CGI-Skript auf einem Webserver
- Client: HTTP-Proxy und Webbrowser

Überblick

I. Designziele

II. Publius

III. Freehaven

- **i. Systemarchitektur**
- **ii. Benutzung**
- **iii. Sicherheitsmechanismen**
- **iv. Implementation**

IV. Vergleich

Freehaven: Systemarchitektur

- Kommunikationssystem
  - Kommunikation per eMail
  - Anonyme Remailer-Systeme
- Servernetzwerk: **servnet**
  - Gemeinschaft von Servern
  - Schlüsselpaar und Reply-Block pro Knoten
  - Listen von Reply-Blocks und Keys der anderen Mitglieder

Freehaven: Remailer-Networks

- Anonymitäts-Problem bei eMails: Passierte Rechner werden im Header vermerkt ➔ Remailer
- Versand in Remailer-Ketten (Reihenfolge vorher festgelegt)
- Nur der letzte Remailer der Kette soll den Empfänger kennen
- Nächste Adresse jeweils mit dem Public-Key des Servers verschlüsselt ➔ Kaskade von ineinander verschachtelten Verschlüsselungen
- Benutzer haben Reply-Blocks (Adressen), in denen sie, wie beschrieben, eine Route zu sich festlegen
- MIX-Master-Remailer-Networks bzw. Cypherpunks-Networks

Freehaven: Veröffentlichung

- Speicherplatz zur Verfügung stellen ➔ Informationen ins Netzwerk einbringen:
  - Betreiben eines Freehaven-Knotens
  - Benutzung eines Introducer-Knotens
- Aufspalten der Informationen (n,k)
- Erzeugung eines Schlüsselpaars für dieses Dokument
- Signieren der Daten
- Zusammenstellen der Freehaven Shares
- Einbringen durch Trading

Freehaven: Share

```
<share>
<PKdoc>cec41f889[…]43662d8c784</PKdoc>
<sharenum>1</sharenum>
<buddynum>0</buddynum>
<totalshares>100</totalshares>
<sufficientshares>60</sufficientshares>
<expiration>2002-07-11-22:25:24
</expiration>
<data>ASCII-Daten</data>
<signature>bdf23f456[…]bf3</signature>
</share>
```

Freehaven: Abrufen

- Identifikation eines Dokuments anhand dessen Public-Key
- Erstellen eines Schlüsselpaars und One-Time-Reply-Blocks für diese Transaktion
- Anfrage-Broadcast:
  - Public-Key des gewünschten Dokuments
  - Public-Key der Transaktion
  - One-Time-Reply-Block
- Überprüfen der Shares durch Server
- Verschlüsselter Versand per eMail mittels Reply-Block
- Rekonstruktion der Informationen

Freehaven: Update & Löschen

- Aus Sicherheitsgründen keine Aktualisierung oder Löschung von Dokumenten vorgesehen
- Entfernen von Informationen automatisch nach Ablauf eines ‚Haltbarkeitsdatums‘

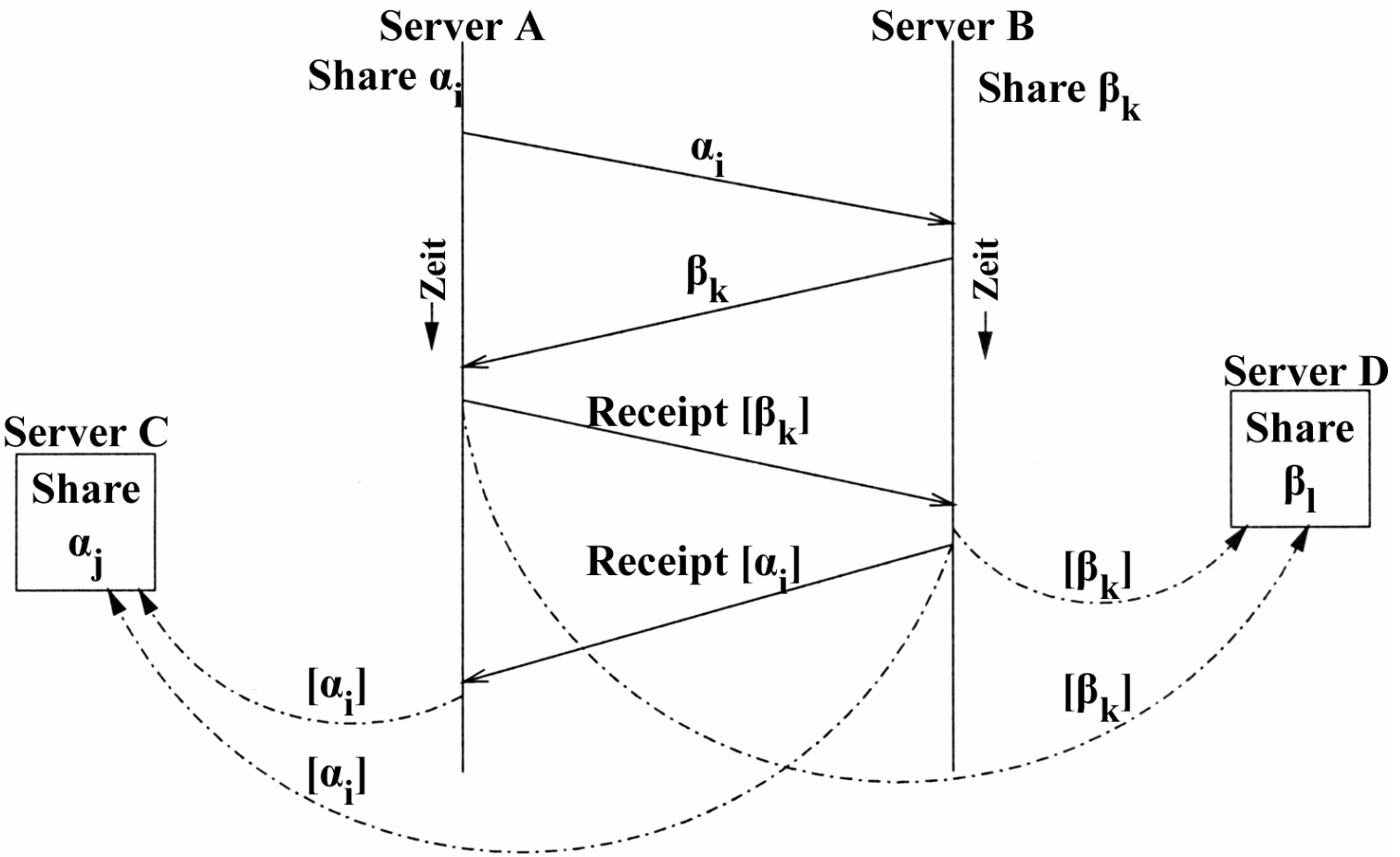
Freehaven: Trading

- Warum Information-Trading?
  - Größeres Maß an Anonymität
  - Dynamisches Netzwerk
  - Längere Haltbarkeitsdaten
  - Rücksicht auf Bedenken von Serveroperatoren
  - Bewegtes Ziel für Angriffe

Freehaven: Trading & Receipts

- Server A schickt ein Angebot, das eines seiner Shares enthält, an den Server B
- Dieser sendet bei Interesse eines seiner Shares zurück
- Versand von Receipts (Quittungen), in denen die Transaktionen beschrieben werden

Freehaven: Trading & Receipts II



Freehaven: Reputation-System

- Pflege von Vertrauenswerten in andere Freehaven-Knoten:
  - Reputation (Guter Ruf)
  - Credibility (Glaubwürdigkeit)
- Änderung der Werte durch Broadcasts:
  - Bei erfolgreichem Trading
  - Bei Fehlverhalten
  - Bei signifikanten Änderungen

Freehaven: Buddy-System

- Gegenseitige Überwachung durch Verbindung zweier Shares
- Benachrichtigung des Partners bei einem Trading-Vorgang
- Periodische Kontrolle des Buddies
- Ggf. Bekanntgabe einer Anomalität, um Vertrauenswerte zu ändern

Freehaven: Implementation

- Proof-of-Design

  Teile des Systems implementiert, um den Grad der erreichten Anonymität und Zensurresistenz zu verifizieren

Überblick

I. Designziele

II. Publius

III. Freehaven

IV. Vergleich

i. Systemarchitektur

ii. Analyse der Anonymität

iii. Angriffe auf die Systeme

Vergleich: Server-Netzwerk

Publius

- Statisches Netzwerk von Publius-Knoten
- Fester Speicherplatz

Freehaven

- Dynamisches Netzwerk
- Wechselnder Speicherplatz durch Trading-Mechanismus

Vergleich: Kommunikationskanäle

Publius

- Benutzung des HTTP-Protokolls
- Anonyme Proxy-Server

Freehaven

- Kommunikation per eMail
- Remailer-Systeme (Mixmaster-, Cypherpunks-Networks)

Vergleich: Update/Delete-Operation

Publius

- Optionale, passwortgeschützte Möglichkeit Informationen zu aktualisieren oder zu löschen

Freehaven

- Verzicht auf Update- oder Delete-Operationen, um Sicherheit zu erhöhen

Vergleich: Sicherheitsmechanismen

Publius

- Bemerken von Veränderungen durch Hash-Werte

Freehaven

- Bemerken von Veränderungen durch Aufspaltung und Signatur

- Spezielle Aufspaltung der Informationen bzw. des Schlüssels und Speicherung auf vielen weit verteilten Knoten

- Reputation-System

Vergleich: Anonymität

- **Autor-Anonymität**

  Eigenverantwortung durch Wahl eines anonymen Kommunikationskanals, um das Dokument zum Verleger zu übertragen

- Herausgeber-Anonymität
- Leser-Anonymität
- Server-Anonymität
- Dokument-Anonymität

Vergleich: Anonymität

- Autor-Anonymität
- **Herausgeber-Anonymität**
  - Publius: Verleger schützt sich durch Benutzung eines Anonymizers / anonymen Proxy-Servers
  - Freehaven: Verlegeridentität wird durch das Share-Trading verschleiert
- Leser-Anonymität
- Server-Anonymität
- Dokument-Anonymität

Vergleich: Anonymität

- Autor-Anonymität
- Herausgeber-Anonymität
- **Leser-Anonymität**
  - Publius: Eigenverantwortung durch Wahl eines anonymen Kommunikationsweges
  - Freehaven: Schutz des Clients durch One-Time-Reply-Blocks innerhalb des verwendeten Remailer-Network
- Server-Anonymität
- Dokument-Anonymität

Vergleich: Anonymität

- Autor-Anonymität
- Herausgeber-Anonymität
- Leser-Anonymität
- **Server-Anonymität**
  - Publius: Server aus der Publius-URL mittels statischer Serverliste identifizierbar
  - Freehaven: Anonyme Reply-Blocks und Broadcasts
- Dokument-Anonymität

Vergleich: Anonymität

- Autor-Anonymität
- Herausgeber-Anonymität
- Leser-Anonymität
- Server-Anonymität
- **Dokument-Anonymität**
  - Publius: Jeder Server hat nur einen Teil des notwendigen Schlüssels
  - Freehaven: Jeder Server hat nur einen Teil der Informationen

Vergleich: Anonymität

Übersicht:

Projekt	Autor	Verleger	Leser	Server	Dokument
Publius	Eigenverantwortung	Eigenverantwortung	Eigenverantwortung	Nein	passiv
Freehaven	Eigenverantwortung	Ja	Ja	Ja	passiv

Angriffe: Zerstörung/Löschung

- Verteilung gleichberechtigter Bruchstücke über eine große Anzahl von Servern auf dem gesamten Globus

  ➔ Kein Schutz vor ausreichend potenten Gegnern

Angriffe: Denial-of-Service

- Keine Mechanismen um Kommunikationswege zu schützen oder Erreichbarkeit zu sichern
- Verlass auf benutzte Kommunikationssysteme

Angriffe: Dataflodding

Publius

- Keine Schutz-Mechanismen implementiert
- Beschränkungen geplant – kein vollkommener Schutz möglich

Freehaven

- Schutz durch Trading-Mechanismus
- Eigener Speicherplatz nötig, um Daten einzubringen ➔ kein Schutz bei gutsituierten Angreifern

Angriffe auf die Anonymität

- Anonymität kann kompromittiert werden durch Kombination von:
  - Intensiven Beobachtungen der Systeme
  - Kenntnissen über Bandbreiten
  - Wissen über Eigenarten des Internet
- Ausüben von rechtlichem/sozialem Druck auf Beteiligte

Angriffe auf Systemeigenheiten

Publius

- Manipulationen der ‚update'-Datei, um Anfragen auf fremde URL umzuleiten und so die Informationen zu zensieren
- …

Freehaven

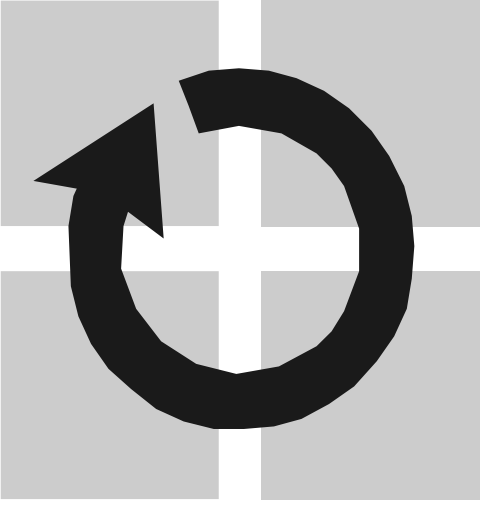
- Ausnutzung des ‚Trading'-Mechanismus, um möglichst viele Shares zu löschen
- Missbrauch des Reputation-Systems
- …

Conclusio

- Anonymität und Zensurresistenz überaus wünschenswert ➔ Entwicklung spezieller Systeme
- Publius: HTTP-basiertes System, das Daten in einem statischem Netzwerk speichert
- Freehaven: eMail-System, in dem Daten dynamisch den Speicherplatz wechseln
- Systeme bieten unterschiedliches Maß an Anonymität und Zensurresistenz ➔ 100%ige Sicherheit ist unmöglich

Freehaven und Publius

Anonyme und zensurresistente Verbreitung von Informationen



Christian Hausner
Universität Erlangen-Nürnberg, 2002
christian.hausner@fau.de