

IT-SICHERHEIT UND DATENSCHUTZ

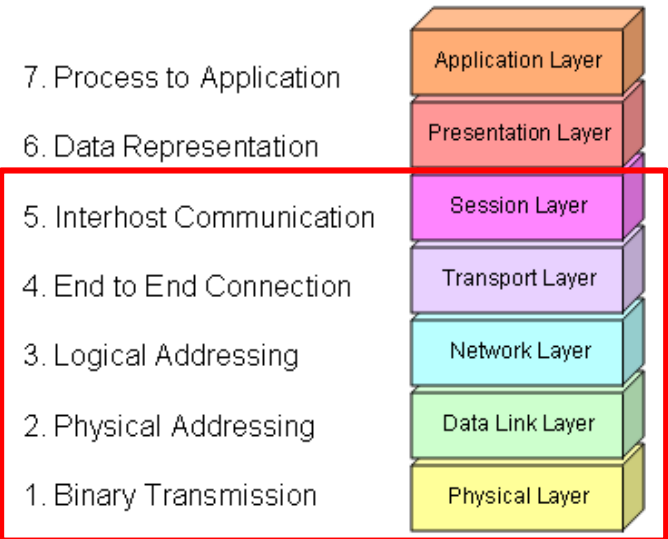
KAPITEL 9 - NETZ-DATENSCHUTZ

buchmann@hft-leipzig.de



LERNZIEL UND AUFBAU DIESES KAPITELS

- Grundlagen
 - Datenschutzverfahren auf Protokollebene
 - Mixe nach Chaum
- Praktische Implementierungen
 - JonDo (vormals JAP)
 - TOR
 - Freenet
- Lernziele
 - Sie können zwischen den Gefahren für den Datenschutz differenzieren, die sich durch die IP-Adresse beim Datenabruf und bei der Datenbereitstellung ergeben.
 - Ihnen ist das Konzept der Mixe im Detail vertraut.
 - Sie können dafür gebräuchliche Schutzverfahren sowie deren Vor- und Nachteile diskutieren.



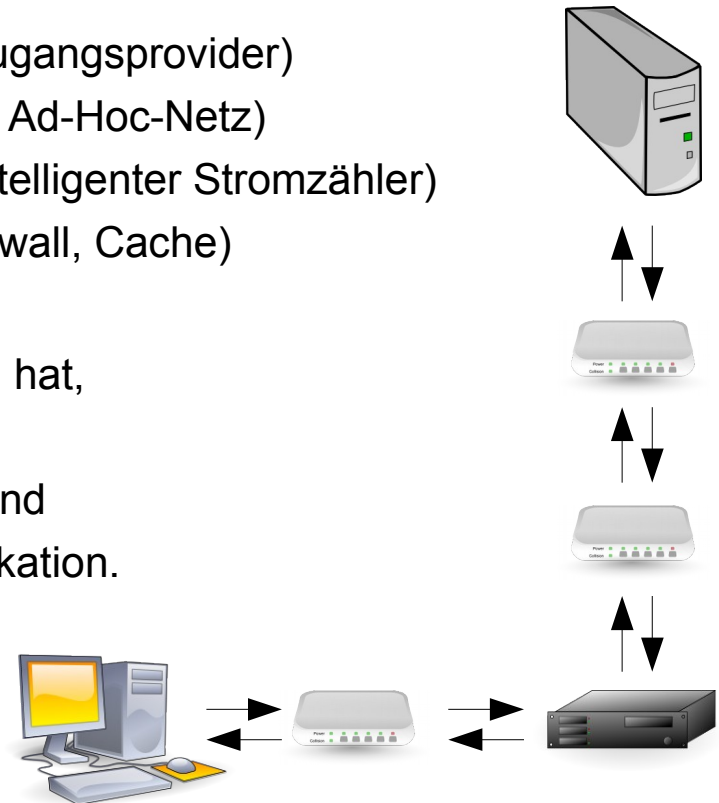
ZIELE DES DATENSCHUTZES

- ➔ 1 grundlegendes Gewährleistungsziel **Datenminimierung**
 - *Umfang* der verarbeiteten Daten, *Zahl* der zugreifenden Stellen, *Ausmaß* der Verfügungsgewalt
- 6 elementare Gewährleistungsziele
- ➔ - **Verfügbarkeit**: Daten für vorgesehene Zwecke verfügbar
- ➔ - **Integrität**: Daten halten Spezifikation aus Zweckbindung ein
- ➔ - **Vertraulichkeit**: Kein Zugriff durch Unbefugte
- ➔ - **Nichtverkettung**: Kein Kombinieren von Daten ohne Erlaubnis
- **Transparenz**: Betroffene, Betreiber, Kontrollinstanzen können erkennen, wo welche Daten zu welchem Zweck vorliegen
- **Intervenierbarkeit**: Betroffenenrechte (Benachrichtigung, Auskunft, Berichtigung, Sperrung, Löschung)

EINFACHSTE ANONYMISIERUNGSVERFAHREN FÜR DIE IP-ADRESSE

DATENSCHUTZPROBLEM IP-ADRESSE

- Viele verteilte Systeme kommunizieren über das Internet-Protokoll
 - Ethernet, WLAN-Protokolle, Powerline, diverse Funktechnologien
- Im verteilten System kann jeder
 - Einstiegspunkt (Basisstation, Internet-Zugangspvoder)
 - Weiterleiter (Router, Zwischenknoten im Ad-Hoc-Netz)
 - Endpunkt (Webserver, Sensorknoten, intelligenter Stromzähler)
 - zwischengeschaltete Dienst (Proxy, Firewall, Cache)
- ermitteln,
 - dass eine Kommunikation stattgefunden hat,
 - wer die Kommunikationspartner waren,
 - welcher Art diese Kommunikation war, und
 - den Inhalt (unverschlüsselter) Kommunikation.



ANGREIFER

- **schwache Angreifer**
 - betreibt Webserver, Router, oder Analysedienst (vgl. Web-Bugs)
 - kann Informationen von wenigen ausgewählten Rechnern zusammenführen
- **starker Angreifer**
 - z.B. Geheimdienst mit starken Ressourcen, Betreiber eines Internet-Backbones
 - kann sehr große Teile der Internet-Kommunikation belauschen

UNTERSCHIEDLICHE ANGREIFERMODELLE

- Senderanonymität
 - Absender einer Nachricht kennt Empfänger, Empfänger erfährt nicht die Identität des Senders
- Empfängeranonymität
 - Absender einer Nachricht kennt den Empfänger nicht, Nachricht wird über einen anonymen Briefkasten weitergeleitet
- Abstreitbarkeit, vollständig anonymer Datenaustausch
 - Speicherort von Daten ebenfalls unbekannt, d.h. kein anonymer Briefkasten mehr vorhanden, über den sich Kommunikationsvorgänge erkennen ließen
 - Nachrichten werden über kryptographische Hash-Signaturen adressiert

ANONYMITÄT BEI SCHWACHEN ANGREIFERN

- VPN über Trusted Third Party, z.B. <http://www.anonymizer.com>
 - leitet eingehende Verbindungen unter der eigenen IP-Adresse weiter
 - löscht Cookies, identifizierende Attribute, Referer etc.

Anonymizer®
Trusted / Proven / Secure

Consumer Enterprise Government About Us

Anonymous Surfing™

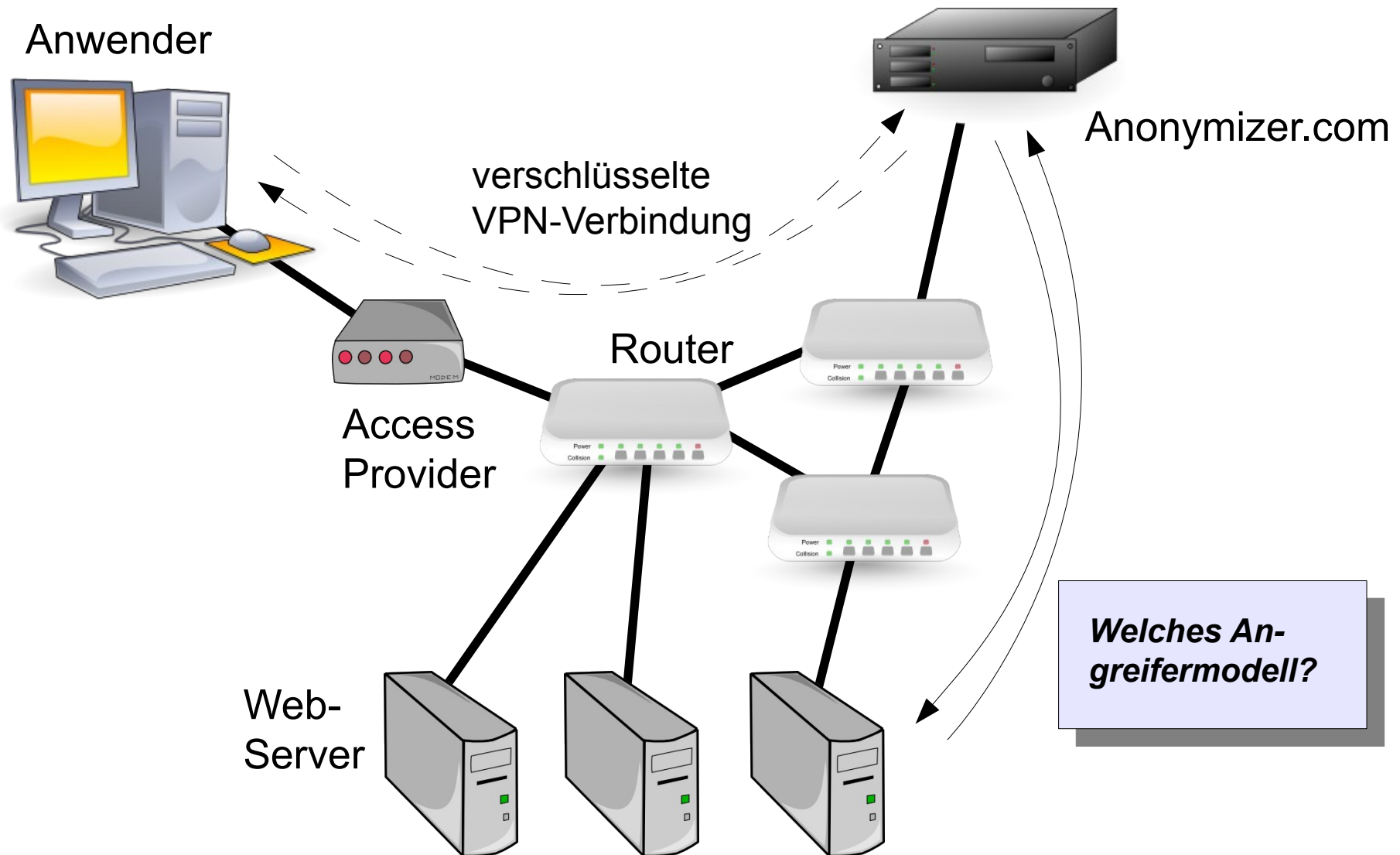
..and it's free to try

Go way beyond simply hiding your computer's IP address with Anonymizer Anonymous Surfing™

- Safe Online Shopping
- Secure Internet Banking
- Phishing & Pharming Alerts
- Wireless PC Security

Weitere: iPredator, Relakks, PRQ, Ivacy, Linkideo Perfect Privacy, Surfonym, SwissVPN, Cinipac, FlashVPN, TorrentPrivacy, TrilightZone, StrongVPN, VPNUK, HotSpotShield...

ANONYMIZER.COM



PRIVOXY

- Privacy-Proxy zum selber-aufsetzen
 - d.h., benötigt einen Rechner, um darauf die Software zu installieren
- Leistet dasselbe wie Anonymizer.com
 - Cookie-Management
 - säubern von HTTP-Headern (Referer, Browser-Informationen, etc.)
 - Filter zum entfernen von Web-Bugs, Scripten
- Open Source, <http://www.privoxy.org/>

SOURCEFORGE.NET®

Privoxy

Summary

Tracker

Mailing Lists

Code

Services

Download

Documentation

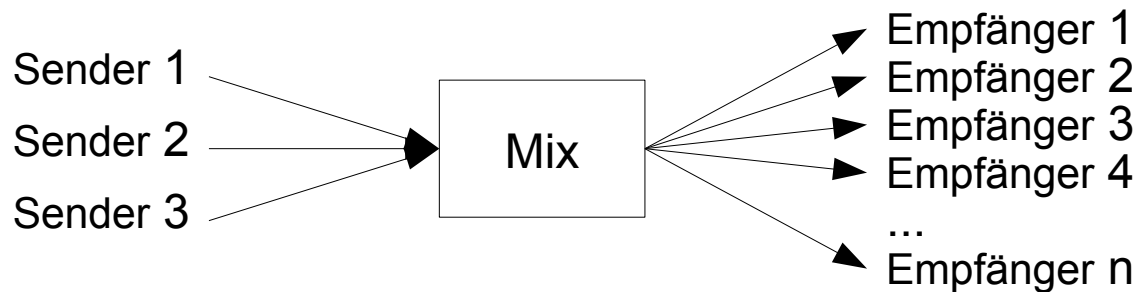
ANONYMITÄT BEI STARKEN ANGREIFERN

- Es werden gebraucht:
 - starke **Public-Key-Verschlüsselung**
 - möglichst **viele unabhängige Rechner**, welche Nachrichten für den Nutzer weiterleiten
- Idee: Mix-Kaskaden
 - Nutzer verbinden sich nicht direkt mit dem Server, sondern über mehrere Zwischenstationen
 - Kommunikation zwischen zwei Stationen erfolgt verschlüsselt
 - Anfragen eines Nutzers werden zwischen denen anderer Nutzer versteckt
- Implementierungen: JonDo, TOR, Freenet
 - Sender-, Empfängeranonymität sowie Abstreitbarkeit möglich

MIXE UND MIX-KASKADEN

MIXE

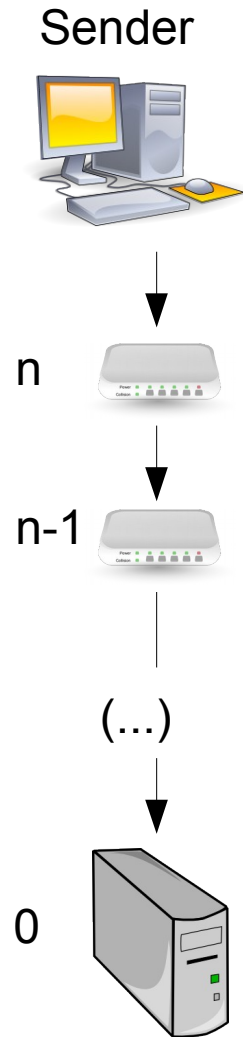
- D. Chaum: *Untraceable Electronic Mail*, Comm. ACM'81
- Mix vermittelt zwischen vielen Sendern und Empfängern



- Jede eingehende Nachricht
 1. entschlüsseln (*Public-Key-Kryptografie*)
 2. kryptografisch transformieren
(*Eingabe-Nachricht* ≠ *Ausgabenachricht*)
 3. Umsortieren, in zufälliger Reihenfolge
(*nicht Eingangsreihenfolge!*)
 4. weiterleiten zum Ziel (oder weiterem Mix)

VERSCHLÜSSELUNG IN MIX-KASKADEN

- Symbole
 - K öffentlicher Schlüssel
 - R Zufallszeichenfolge
 - A Adresse der nächsten Station
- Arbeitsweise eines Mix
 - Mix n erhält $K_n(R_n, K_{n-1}(R_{n-1}, K_{n-2}(R_{n-2} \dots \text{Msg}, A_0), A_{n-2}), A_{n-1})$
 - mit K_n verschlüsselte Nachricht auspacken, R_n löschen, Adresse A_{n-1} lesen
 - sendet $K_{n-1}(R_{n-1}, K_{n-2}(R_{n-2} \dots \text{Msg}, A_0), A_{n-2})$ an A_{n-1}
 - Letzter Mix sendet Msg unverschlüsselt an Adressaten A_0 (kann natürlich auch verschlüsselt erfolgen)



ENTSCHLÜSSELUNG IN MIX-KASKADEN

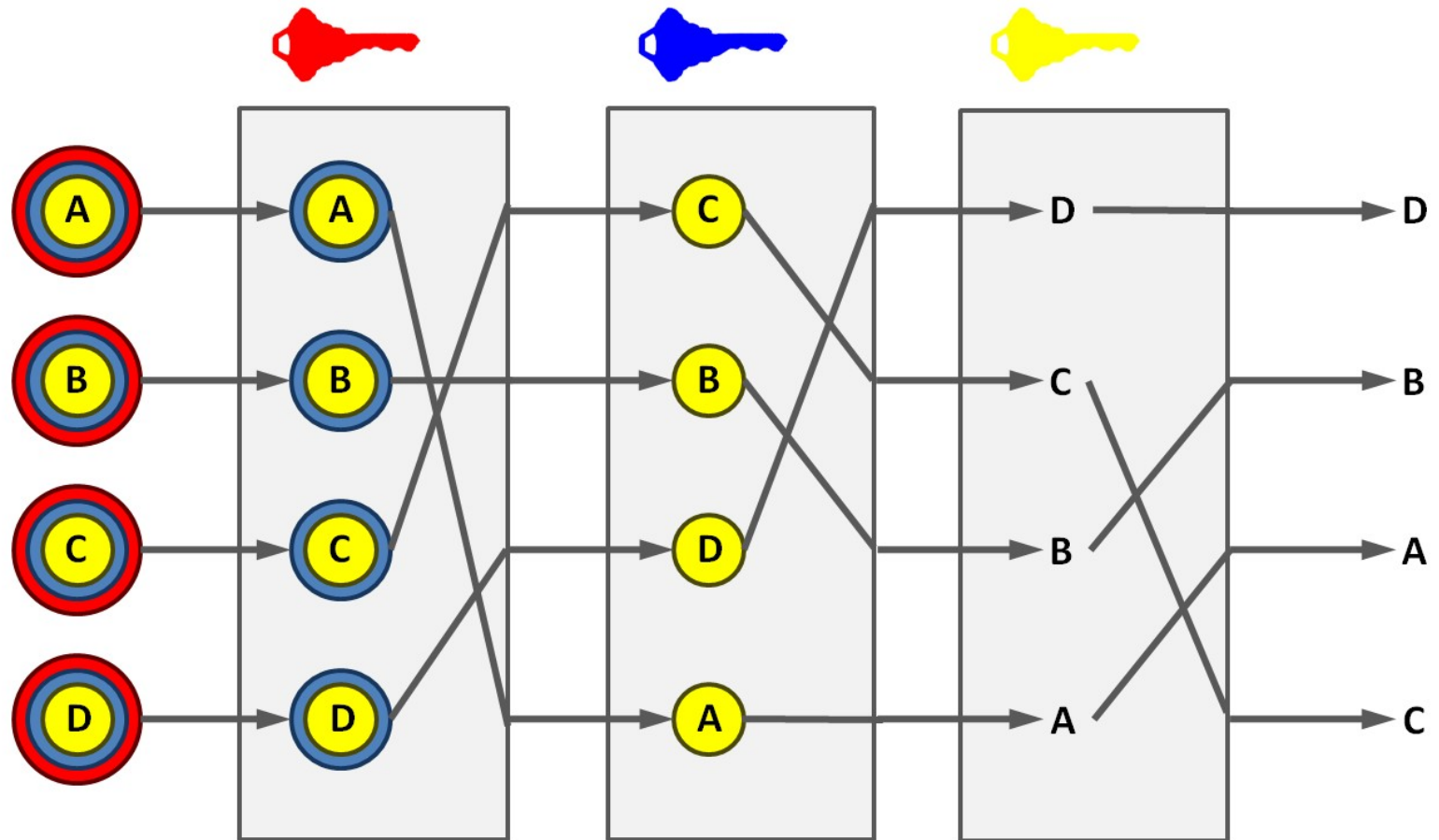
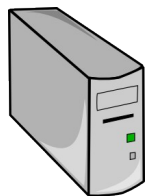
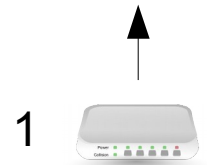


Bild: Wikimedia CC

WAS IST MIT DER ANTWORT?

- Empfänger soll nicht den Absender feststellen, muss aber trotzdem antworten können
- Prinzip: Adresse für Antwort verschlüsselt mitsenden, R dient jeweils als Schlüssel für die Antwort
 - Absender X, Einmalschlüssel K_X , Empfänger 0
 - Antwortadresse $K_1(R_1, A_X)$, K_X wird zusammen mit ursprünglicher Msg mitgesendet
 - Mix 0 kann Adresse A_X nicht entschlüsseln, sondern nur der letzte Mix in der Kaskade
 - Mix 1 erhält von Mix 0 Antwort $K_1(R_1, A_X)$, $K_X(R_0, \text{Msg})$ und sendet $R_1(K_X(R_0, \text{Msg}))$ an A_X
 - Verfahren lässt sich ebenfalls kaskadieren

Absender X



Empfänger 0

EIGENSCHAFTEN DER MIXE NACH CHAUM

- Senderanonymität
 - Absender muss Empfänger kennen
- Nur öffentliche Informationen werden übertragen
 - Absender muss kennen:
 - Empfänger
 - öffentliche Schlüssel $K_0 \dots K_n$ aller Mixe
 - Jeder Mix m kennt
 - eigenen privaten Schlüssel K_m^{-1}
- Aber: einige *praktische* Probleme
 - Public-Key-Verschlüsselung ist *langsam*
 - statistische Angriffe über die Zahl der weitergeleiteten Pakete je Mix
 - Replay-Angriffe (Wiedereinspielen von gültigen Paketen)

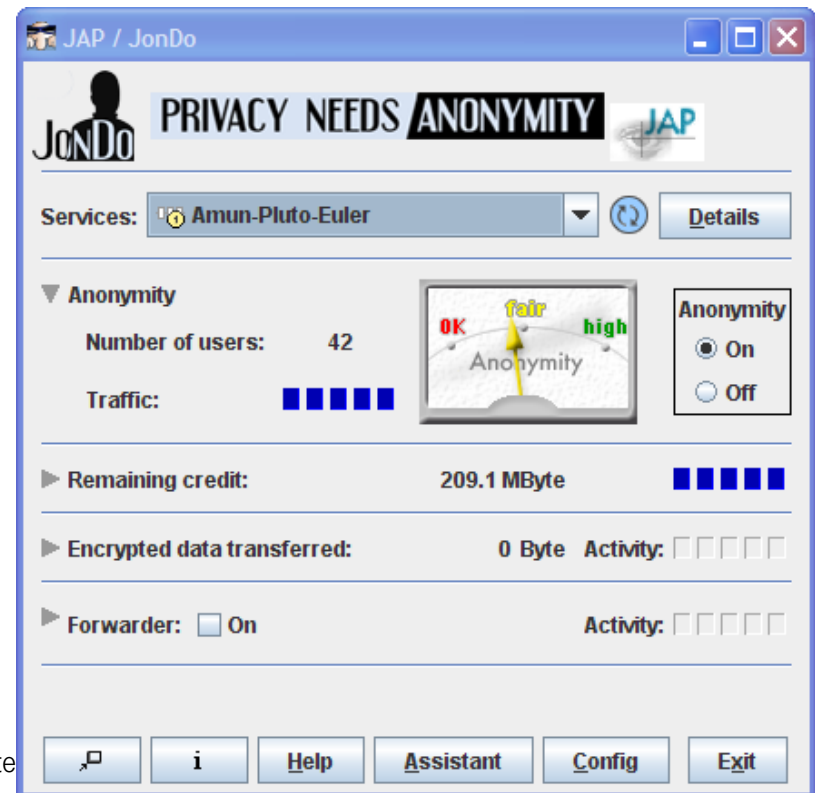
VERBERGEN DER IP-ADRESSE MIT JONDO

JONDO (JAP ANON PROXY)

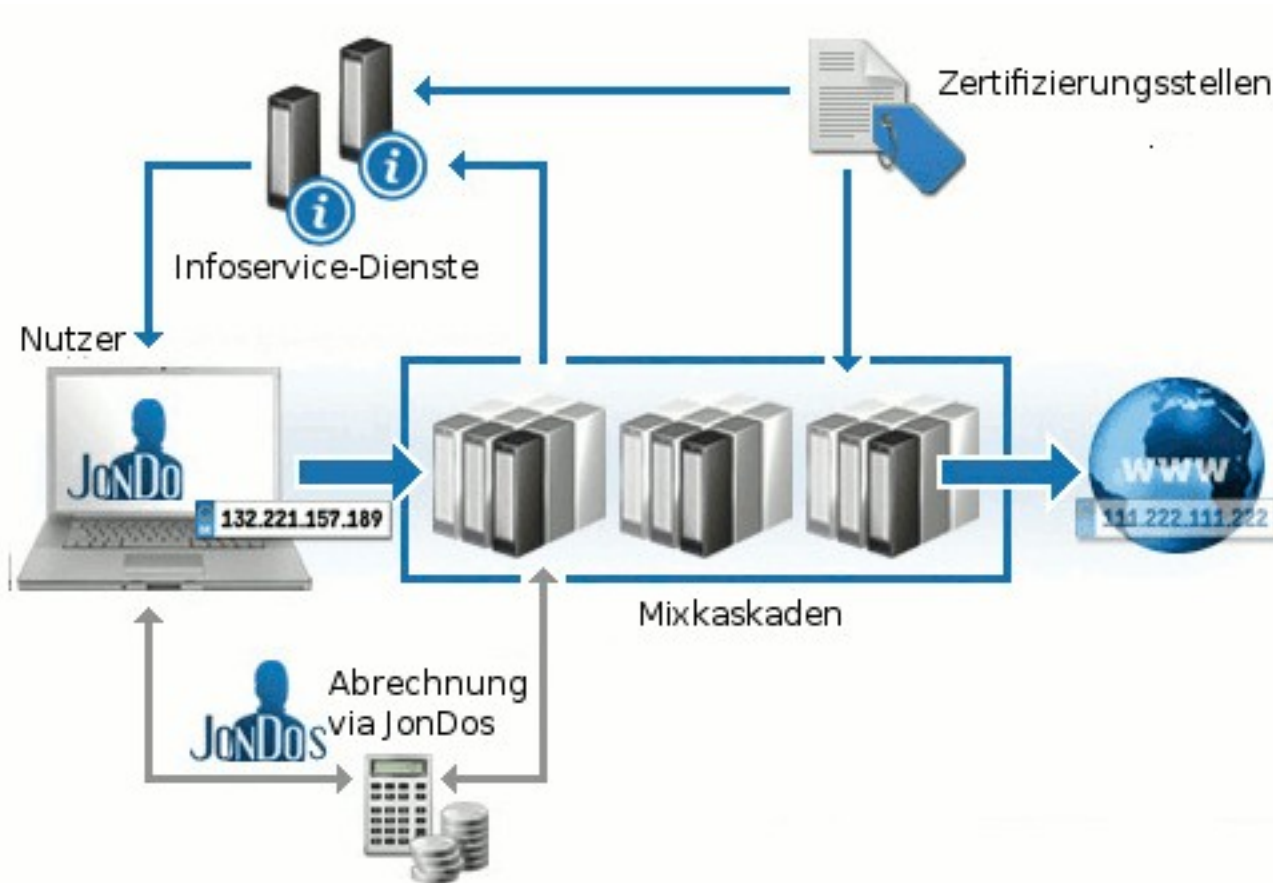
- Entwickelt als Java Anon Proxy seit 2000 vom AN.ON-Projekt
 - Uni Regensburg, TU Dresden, gefördert mit Mitteln der DFG und vom BMBF, seit 2006 keine Finanzierung mehr
- Weiterentwicklung unter dem Namen JonDo
 - <http://www.anonym-surfen.de/>
- Ziele:
 - **Senderanonymität** beim Abruf von Webseiten
 - **Geschlossenes Verfahren**, Sicherheit durch Zertifizierungsinstanz
 - Dienstgüte; sowohl bzgl. Netzparametern (Durchsatz, Latenzzeit etc.) als auch Sicherheit (Anonymität)

JONDO

- Im Gegensatz zu Chaum **feste Kanäle** mit **symmetrischer Verschlüsselung**
→ Problem mit der Rücksendeadresse entfällt,
symm. Verschlüsselung ist effizienter als Public-Key
- grafische Client-Komponente
 - Proxy im Webbrowser
- Info-Service informiert über Schlüssel und aktive Mixe
- kostenpflichtige Premium-Mixe



ARCHITEKTUR VON JONDO



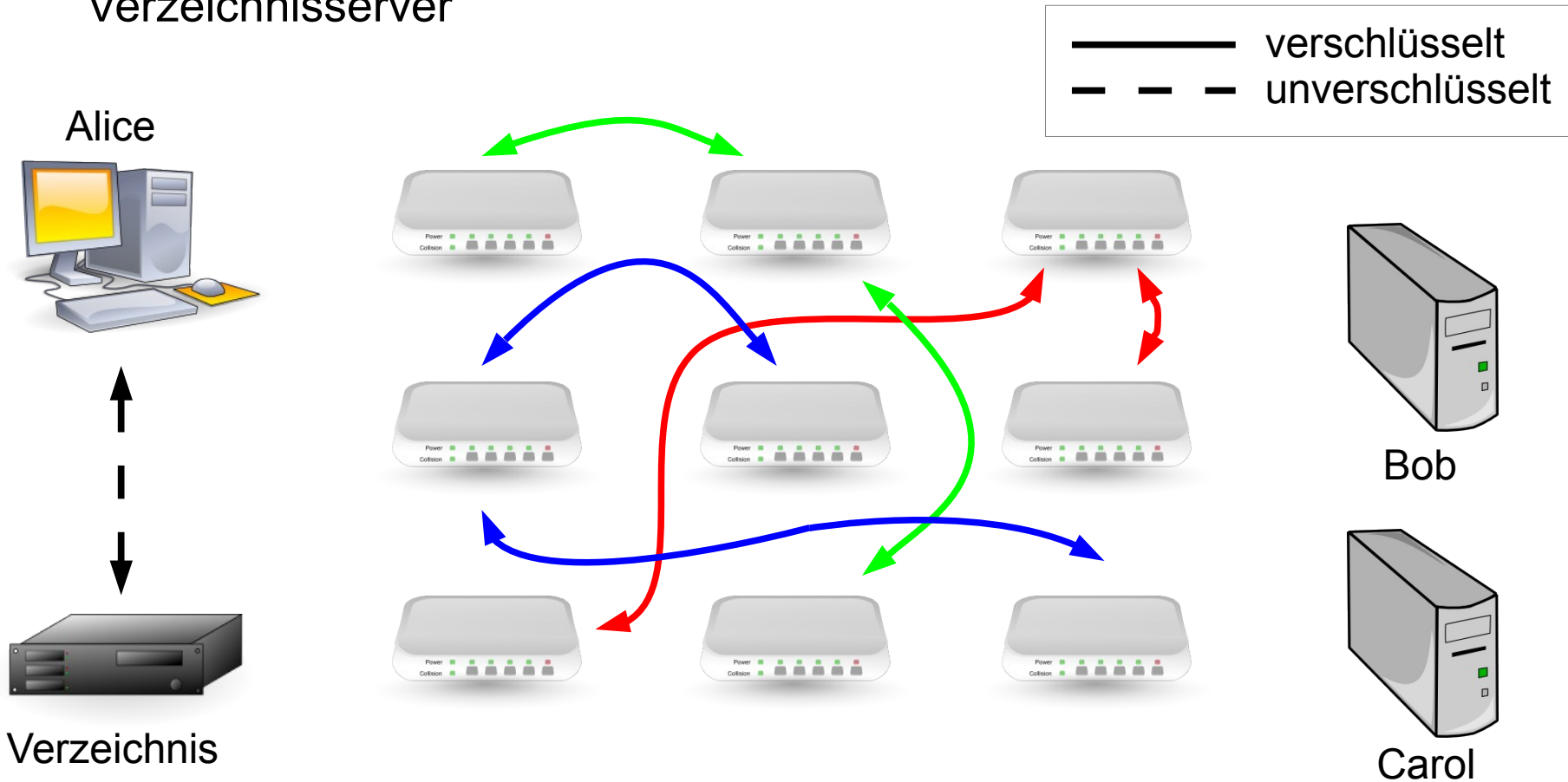
Quelle: JonDos GmbH

MIX-KASKADEN

- Statische Verbindung zwischen zwei Kommunikationsendpunkten über mehrere Mixe hinweg
 - Verbindungsorientierter Vollduplex-Betrieb
(*Gegensatz zu paketorientiert bei Chaum*)
 - vom Sender einer Nachricht initiiert
- Sicherheitsfeatures
 - Datenstrom wird auf mehrere MIX-Pakete aufgeteilt, die alle gleich groß sind (auffüllen mit Zufallszahlen)
 - Zufällige Pakete an beliebige Empfänger, um Zuordnung zu verhindern
 - Symmetrische Verschlüsselung;
(langsame) Public-Key-Verschlüsselung nur zum Aushandeln der symmetrischen Schlüssel
 - Prüfung der Zertifikate von Info-Service und Mixen

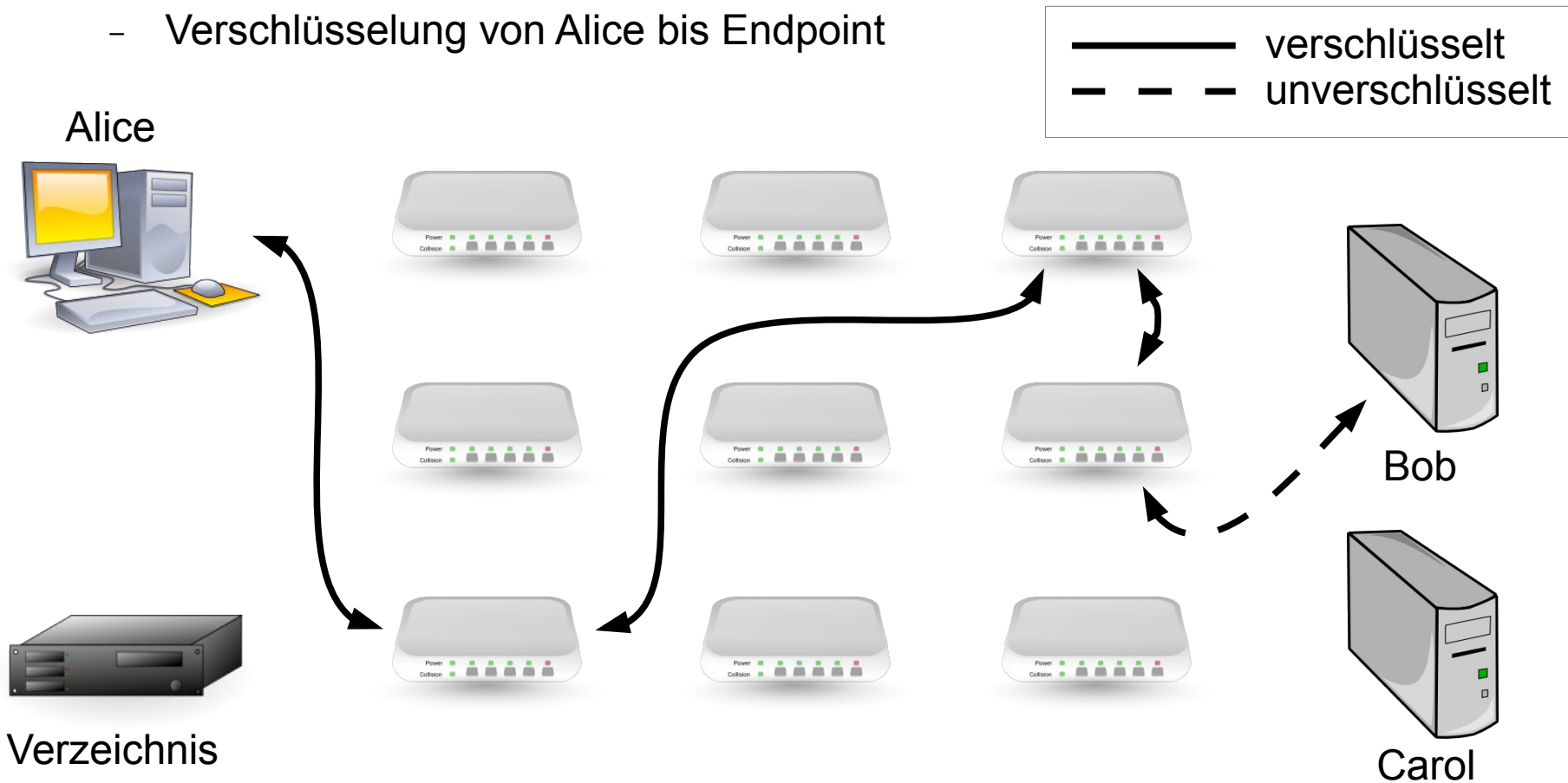
FUNKTIONSWEISE VON JONDO (1/3)

- Alice erhält Liste von verfügbaren Mix-Kaskaden vom Verzeichnisserver



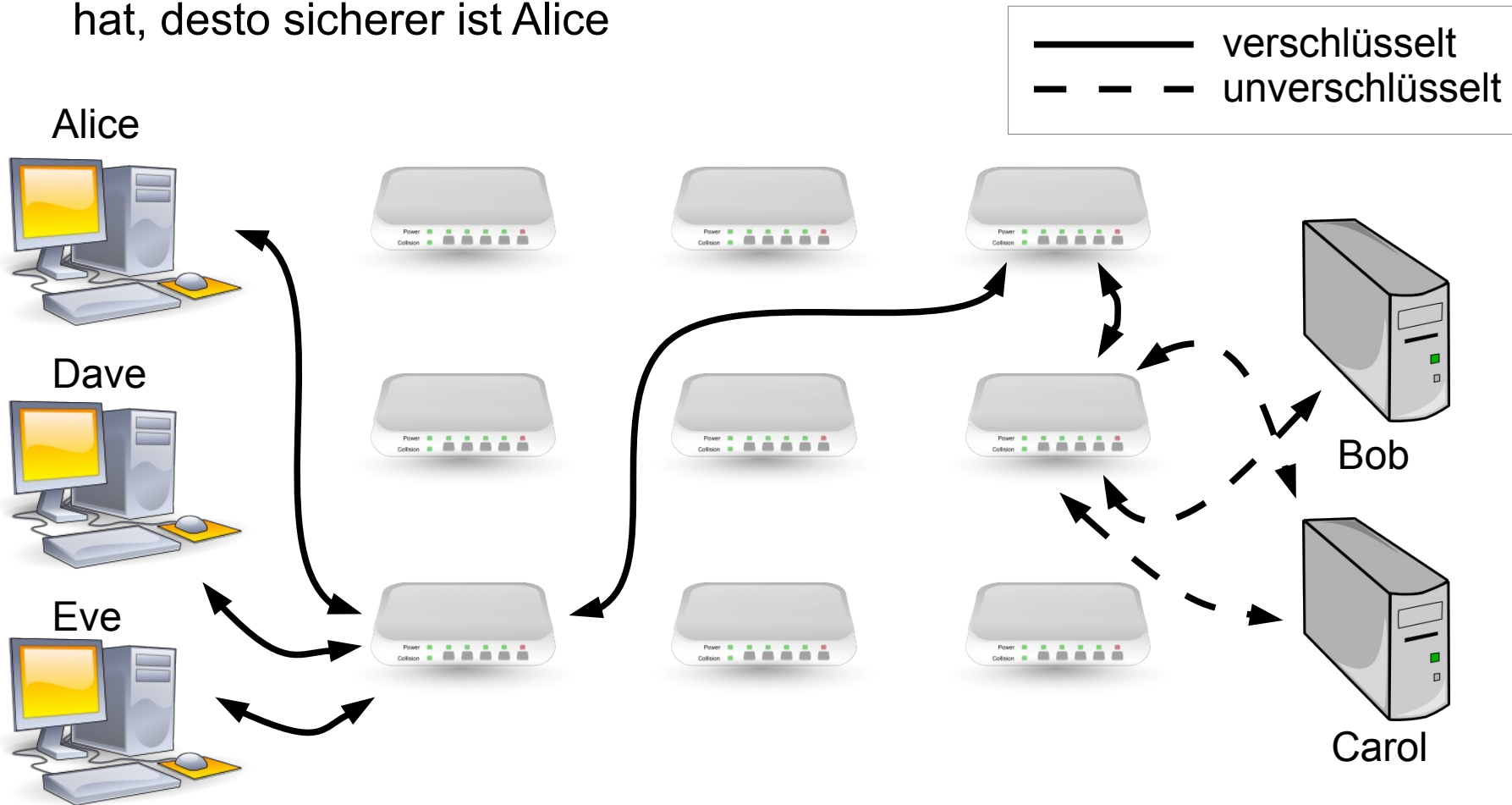
FUNKTIONSWEISE VON JONDO (2/3)

- Alice wählt eine Kaskade aus
 - Verschlüsselung von Alice bis Endpoint



FUNKTIONSWEISE VON JONDO (3/3)

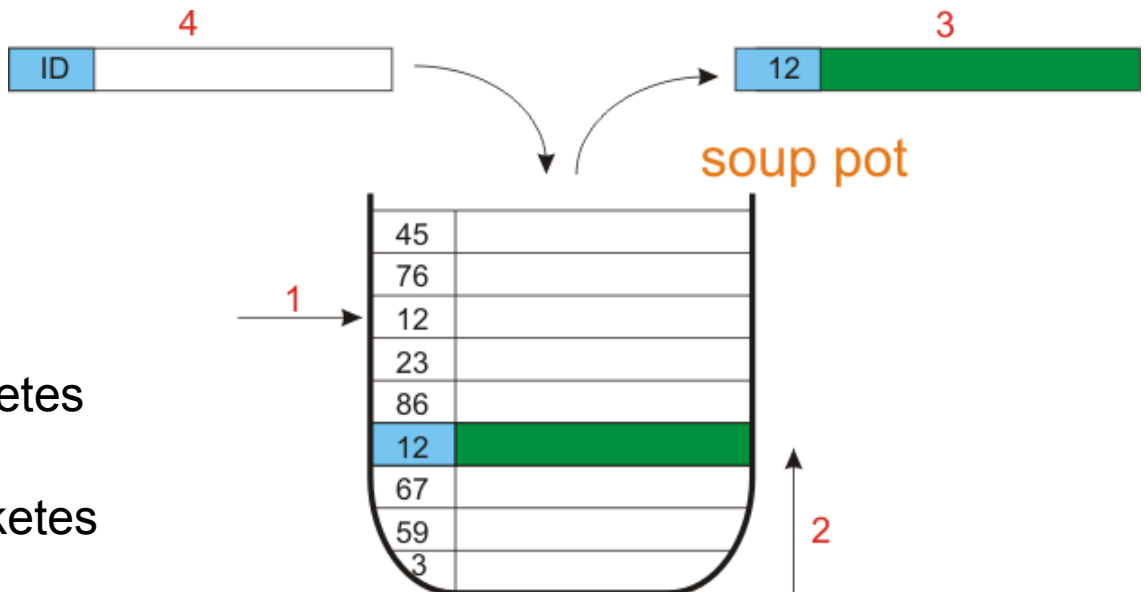
- Je mehr Teilnehmer die (statische) Kaskade hat, desto sicherer ist Alice



BETRIEB EINES MIXES

- Zertifikat von der Zertifizierungsstelle ausstellen lassen
 - Zuverlässigkeitsprüfung, Schutz vor manipulierten Mixen
- Alle 10 Minuten Status-Update an den Info-Service
 - Auslastung, Name, Betreiber, Zahl der gemixten Pakete
- Beim Eintreffen eines MixPaketes:

1. Zufällige Auswahl eines MixPaketes (hat *Kanal-ID*)
2. Suchen nach dem ältesten MixPaket mit *Kanal-ID* im Pool
3. Ausgabe des MixPaketes
4. Hinzufügen des empfangenen MixPaketes



STATUS DER JONDO-MIXE

Status der Mixkaskaden

Name	Nutzer	Verfügbarkeit	Geschwindigkeit	Antwortzeit
<u>Premium-Kaskaden (JonDonym)</u>				
<u>Koala-SpeedPartner-Titan</u>	20	hervorragend	n/a	n/a
<u>Koelsch-Montesquieu-UranusGB</u>	9	hervorragend	n/a	n/a
<u>Born-Niagara-Speedster</u>	6	hervorragend	n/a	n/a
<u>Chomsky-Tulpe-Raiden</u>	-1	hervorragend	n/a	n/a
<u>Kostenfreie Kaskaden (JonDonym)</u>				
<u>SpeedPartner-Cyrax</u>	124 / 600	akzeptabel	100 kbit/s	750-500 ms
<u>Test/Experimental-Dienste</u>				
<u>VPN Test (JAP)</u>	13	unbekannt	n/a	n/a
<u>Dresden (JAP)</u>	238	unbekannt	100 kbit/s	750-500 ms

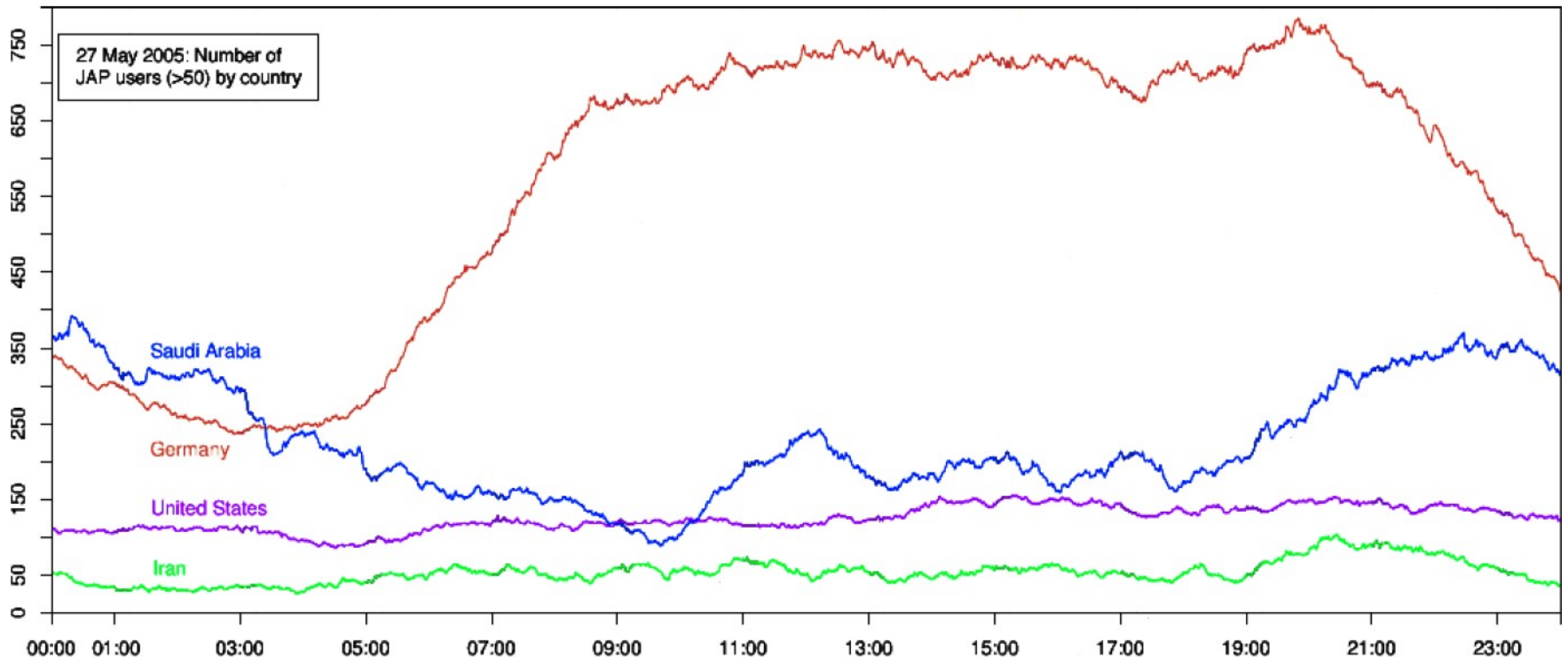
JONDO UND STRAFVERFOLGUNG

- Aufdeckung von Nutzerverbindungen erfordert eine behördliche Anweisung an **jeden** Mix-Betreiber in einer Kaskade (D: Gerichtsbeschluss nach §100a StPO)
 - Durch wechselnde Schlüssel der Kaskade nur **vor** der Nutzung möglich
 - Aufzeichnung einzelner Verbindungen erst nach Anordnung
 - Schwierig, wenn Mixe in unterschiedlichen Ländern stehen
 - Für Premium-Kaskaden bislang noch nicht erfüllt
- Historie der Überwachungsanordnungen
 - 2006: 1 Anordnung an einzelnen Mix
 - 2009: 1 Anordnung an einzelnen Mix bzgl. 2 Email-Zugänge
 - 2010/11: 1 Anordnung an einzelne Betreiber bzgl. eines JonDo-Kontos
 - 2012/13: 1 Anordnung an alle deutschen Mixe bzgl. eines JonDo-Kontos
 - 2014: 1 Anordnung, konnte nicht bedient werden
 - 2015-heute: nichts

WER SIND DIE NUTZER?

- Europa 60%
- Asien 27%
- USA 12%
- Rest 1%

Quelle: <http://anon.inf.tu-dresden.de>



STÄRKEN UND SCHWÄCHEN DES ANSATZES

- Verschleiert nur die IP-Adresse
 - Cookies, Identifizierung durch Quasi-Identifizierer in Suchmaschinen- oder Formulardaten möglich
- Anonymität, solange nicht alle Betreiber einer Kaskade kooperieren
 - jedenfalls wenn der Benutzer keine Fehler macht → Cookies, etc.
- Statische Kaskaden
 - Wenn Angreifer errät, welche Kaskade eine Zielperson nutzt, genügt der Angriff von 3 Rechnern
- Kontrollierte Umgebung
 - Mix-Betreiber muss sich zertifizieren lassen
 - Schutz gegen Mißbrauch, anders als z.B. bei TOR
 - Premium-Server für hohe Performanz, Verfügbarkeit
 - Mix-Server einer Kaskade sind Single Point of Failure und Bottleneck bei zuvielen parallelen Nutzern
 - Insgesamt relativ wenige Mixe, lässt sich daher leicht sperren

VERBERGEN DER IP-ADRESSE MIT TOR

TOR

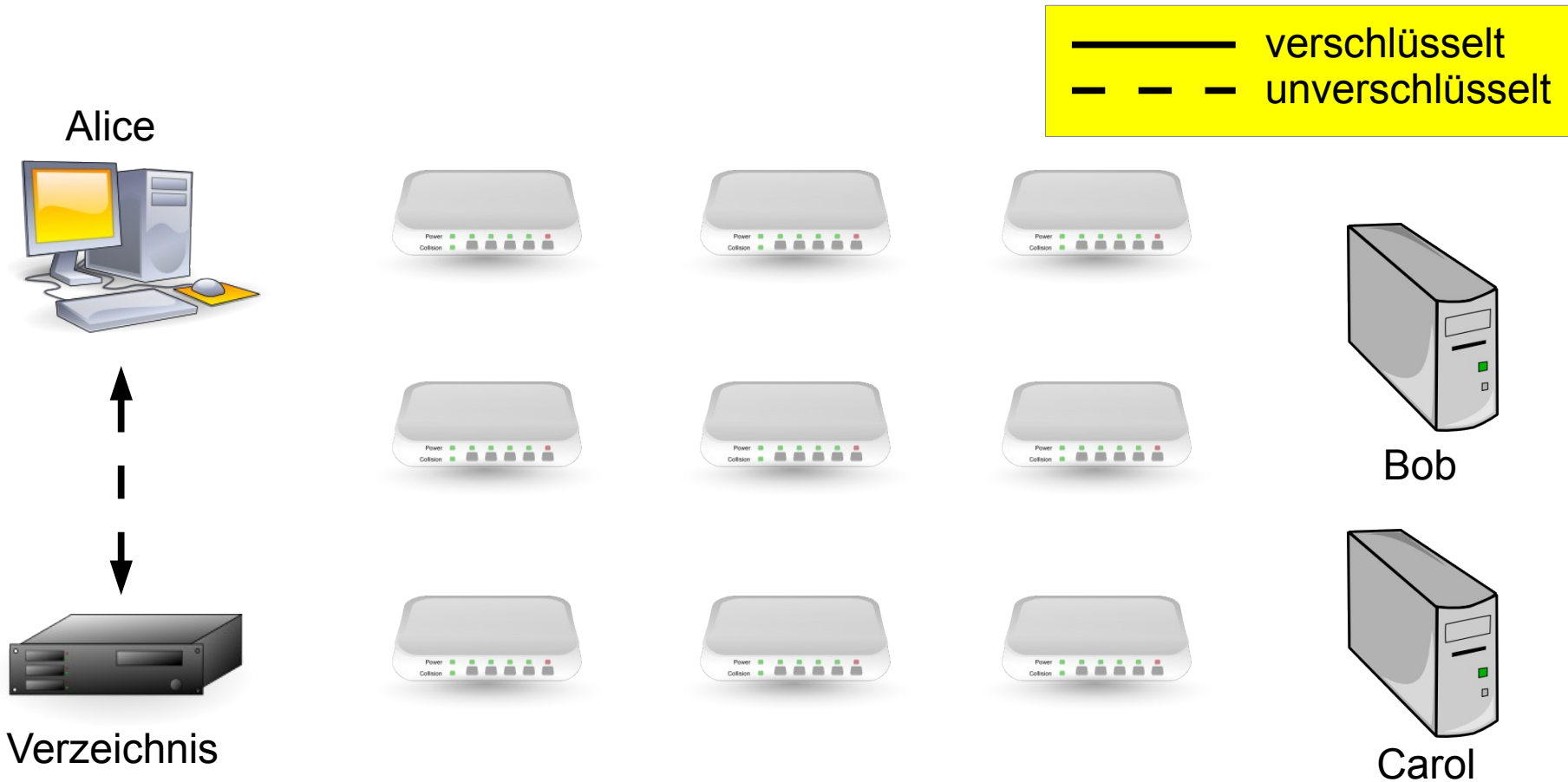
- Projekt der Electronic Frontier Foundation
- Merkmale von Tor
 - **Sender-Anonymität**
 - Onion-Routing (nichts anderes als das Mix-Modell)
 - **Empfänger-Anonymität**
 - Hidden Services: Anonyme Serveradressen
 - **Offenes Verfahren**
 - Jeder kann teilnehmen (im Gegensatz zu JonDo mit Zertifizierungsinstanz)
 - **Freie Rede, überall**
 - Bridge-Knoten von Freiwilligen in die Netze von China, Syrien, Iran, etc.
 - Zufällige Verbindungen für jede Netzwerkverbindung
 - d.h. Kanal wird neu gewählt, sobald Server die Verbindung schließt.



Roger Dingledine et al.: Tor: The Second-Generation Onion Router, Proceedings of the SSYM 2004

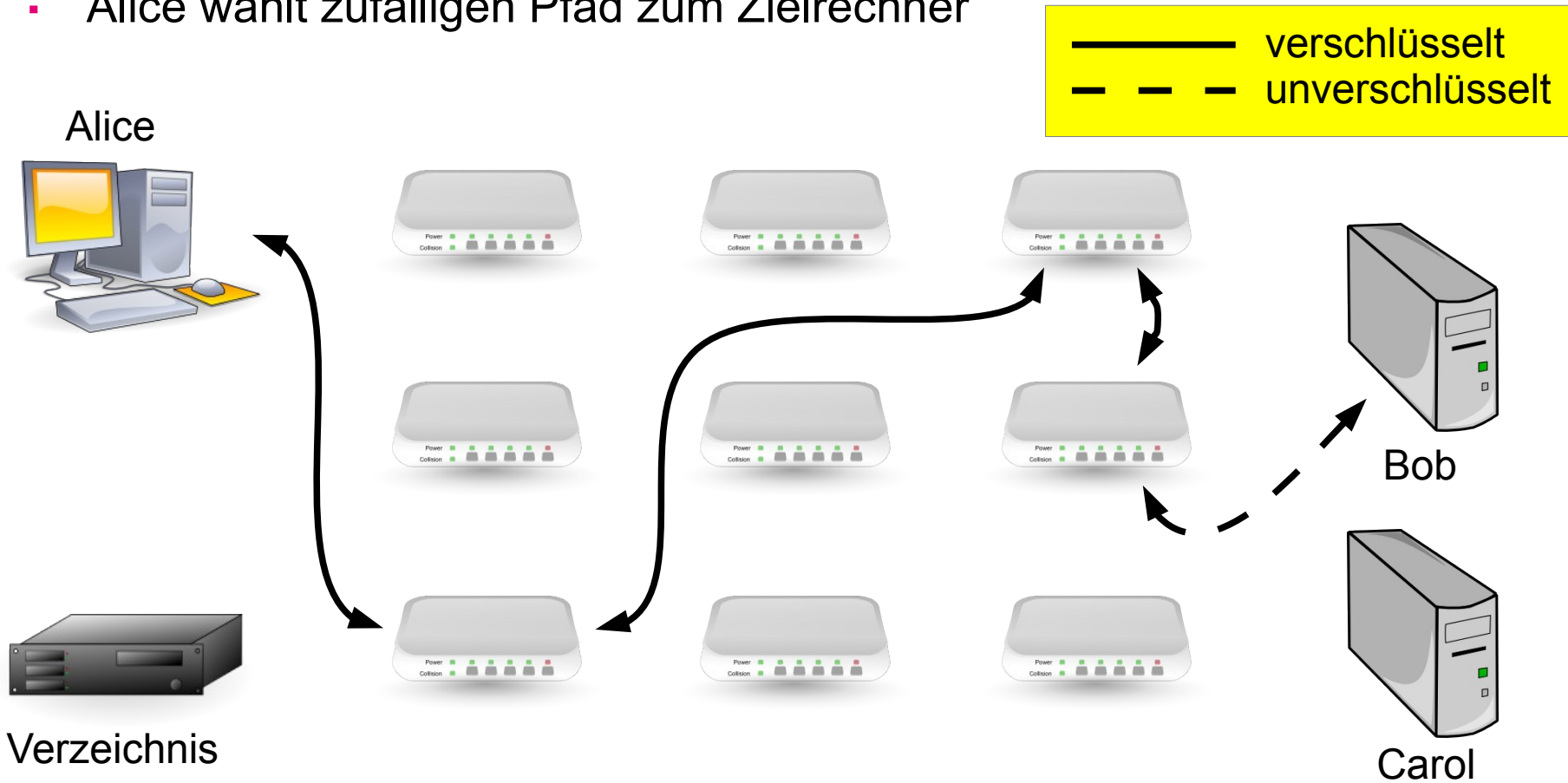
FUNKTIONSWEISE VON TOR (1/3)

- Alice erhält Liste von Torknoten vom Verzeichnisserver



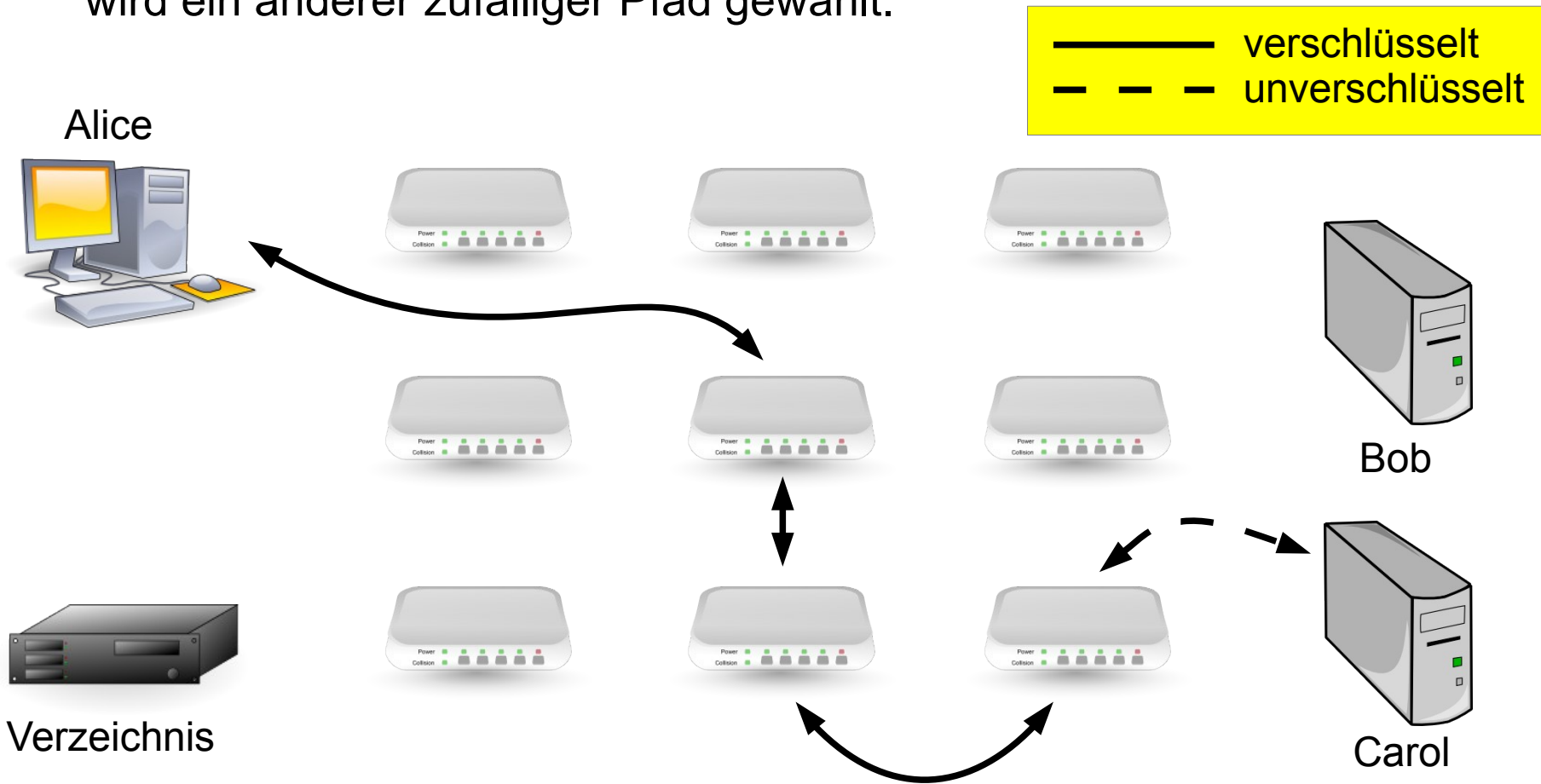
FUNKTIONSWEISE VON TOR (2/3)

- Alice wählt zufälligen Pfad zum Zielrechner



FUNKTIONSWEISE VON TOR (3/3)

- Wenn Alice auf einen anderen Rechner zugreifen will, wird ein anderer zufälliger Pfad gewählt.



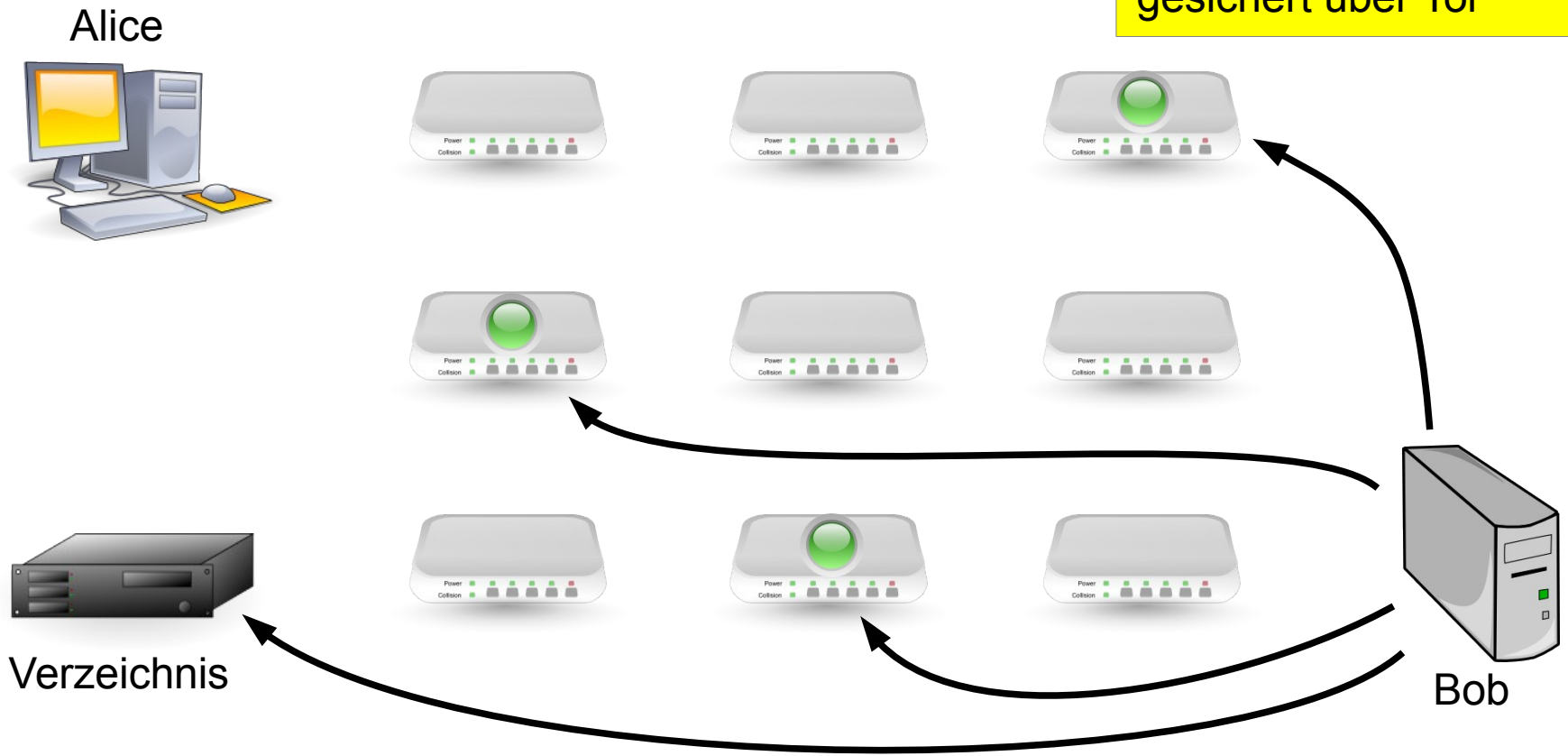
HIDDEN SERVICES

- Ziel: Empfänger-Anonymität
 - Dienste anbieten, ohne die eigene Identität preiszugeben
 - herkömmliches WWW: whois-Anfrage verrät Betreiber
- Bestandteile
 - **Introduction Points** als Verbindung zum Server
 - **Descriptor** auf Verzeichnisserver, der Public Keys und Introduction Points enthält
 - Anonymer Dienst wird über einen eigens generierten **Public Key** identifiziert, der sonst nirgends gebraucht wird
 - **Rendezvous-Knoten** zur anonymen Verbindungsaufnahme

TOR HIDDEN SERVICES (1/4)

- Bob wählt zufällige Introduction Points aus, und teilt die dem Verzeichnisserver mit

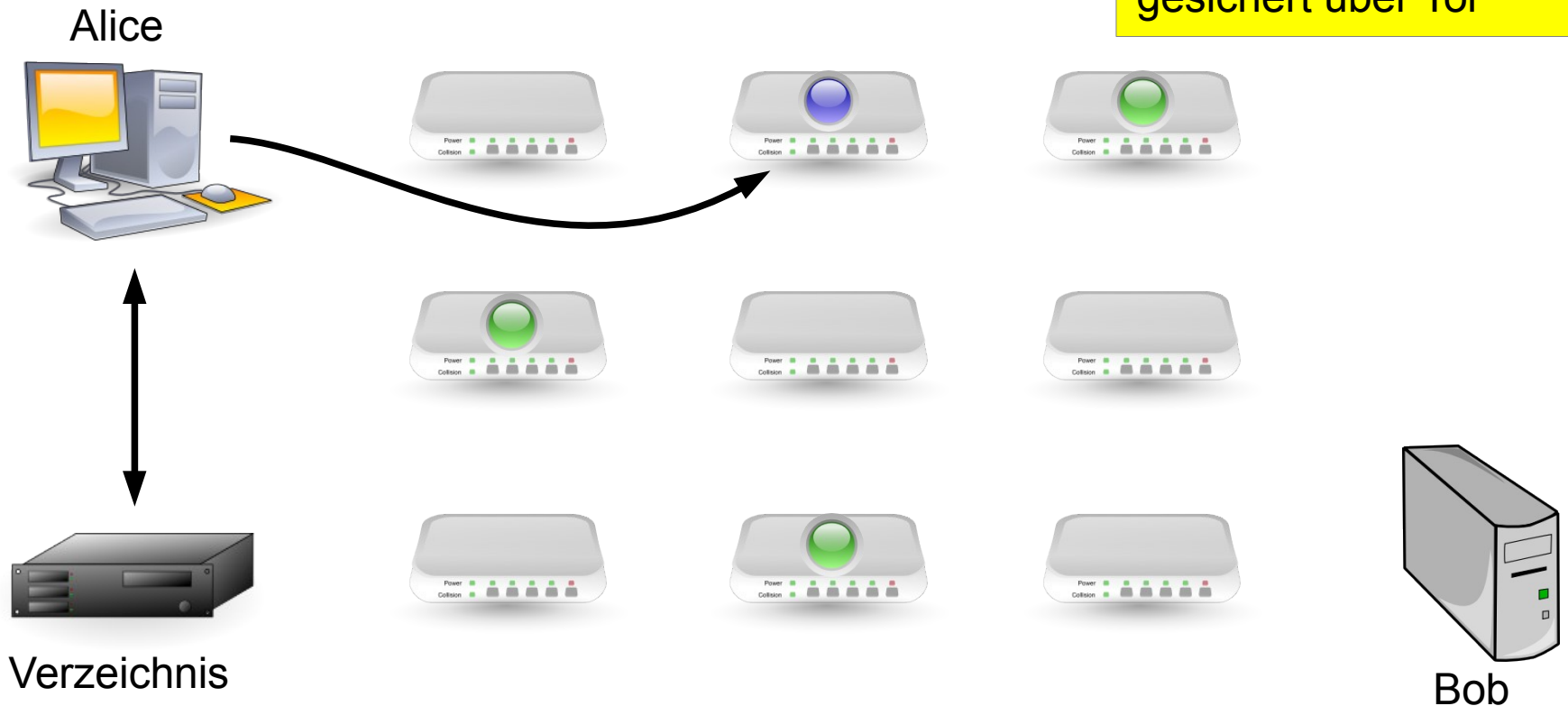
Anm.: alle Verbindungen gesichert über Tor



TOR HIDDEN SERVICES (2/4)

- Alice fragt Verzeichnisserver nach den Introduction Points von *bob.onion* und legt Rendezvous-Knoten fest

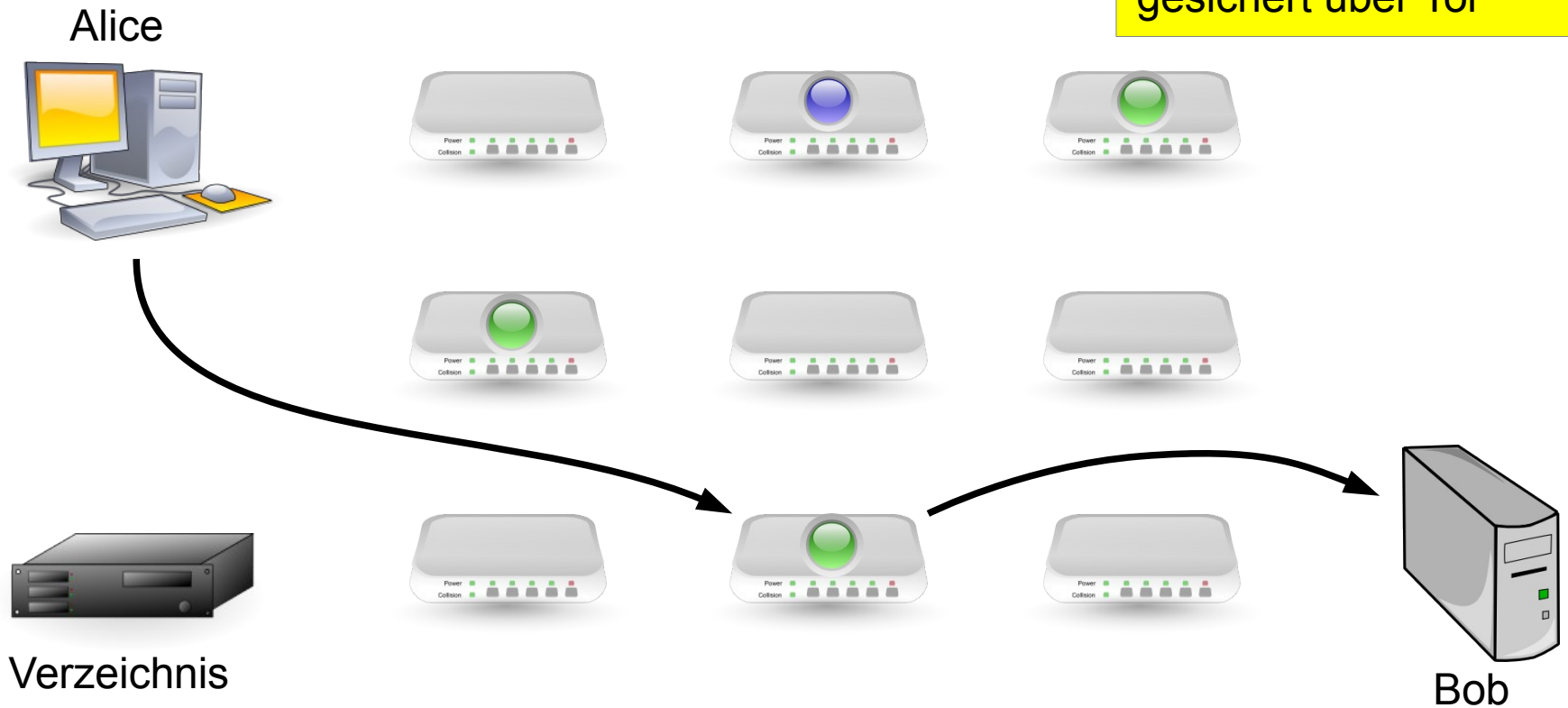
Anm.: alle Verbindungen gesichert über Tor



TOR HIDDEN SERVICES (3/4)

- Alice schickt Bob über Introduction-Point ein One-Time-Secret und ihren Rendezvous-Knoten

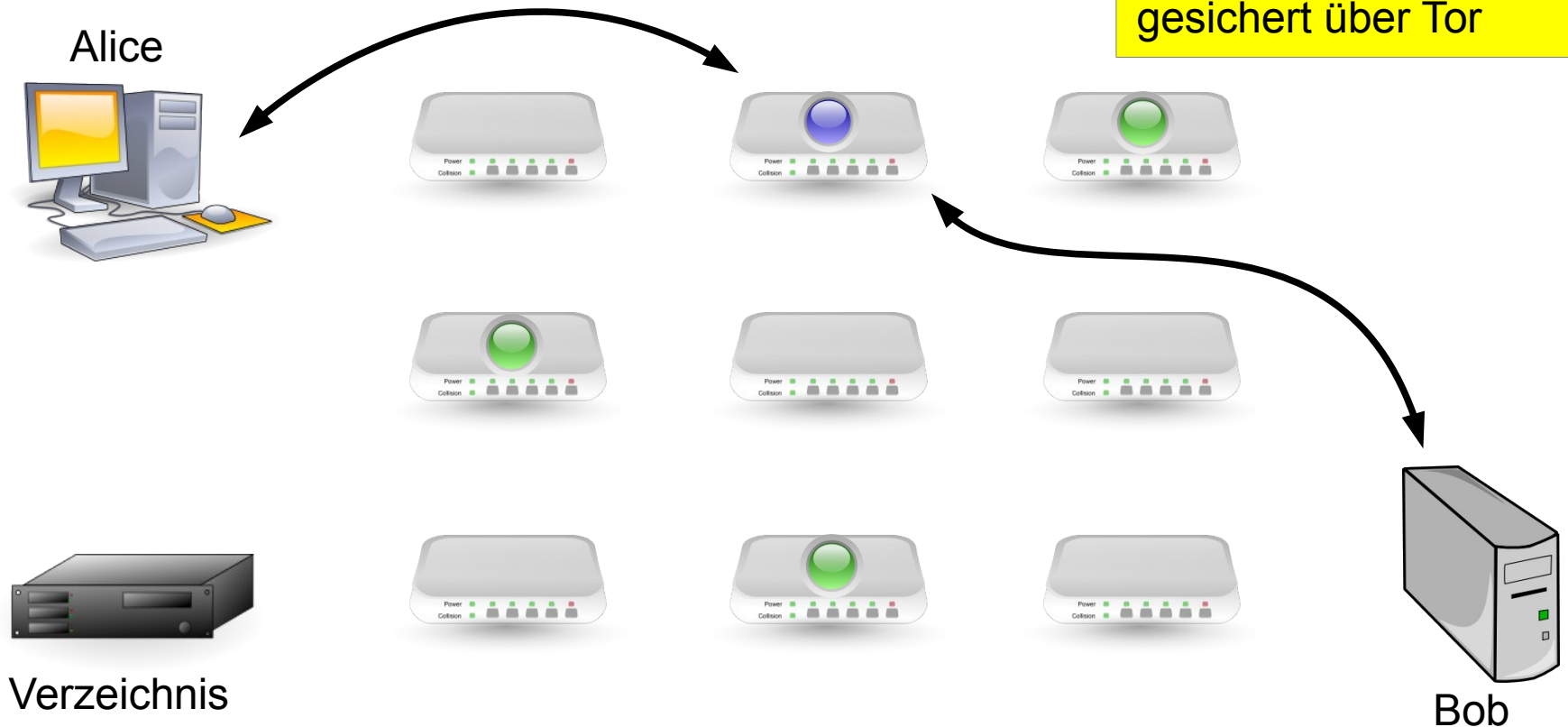
Anm.: alle Verbindungen gesichert über Tor



TOR HIDDEN SERVICES (4/4)

- Bob antwortet mit dem One-Time-Secret über Rendezvous-Knoten
- für beide anonyme Verbindung ist hergestellt

Anm.: alle Verbindungen gesichert über Tor



Warum keine Kommunikation über Introduction Points?

WER SIND DIE NUTZER VON TOR?

- Stand: 16.04.2019

Country	Mean daily users
Russia	10118 (18.57 %)
Iran	4819 (8.85 %)
United States	4768 (8.75 %)
Turkey	3690 (6.77 %)
India	2531 (4.65 %)
Indonesia	1848 (3.39 %)
China	1690 (3.10 %)
Germany	1636 (3.00 %)
Belarus	1460 (2.68 %)
Brazil	1416 (2.60 %)

- Stand: 12.05.2021

Country	Mean daily users
United States	456040 (21.29 %)
Russia	318090 (14.85 %)
Germany	181842 (8.49 %)
Netherlands	101248 (4.73 %)
Lithuania	71198 (3.32 %)
Turkey	69810 (3.26 %)
France	65365 (3.05 %)
Indonesia	60219 (2.81 %)
United Kingdom	60194 (2.81 %)
Ukraine	50571 (2.36 %)

Quelle: <https://metrics.torproject.org>

PERFORMANZ VON TOR

Time to complete 50 KiB request to public server

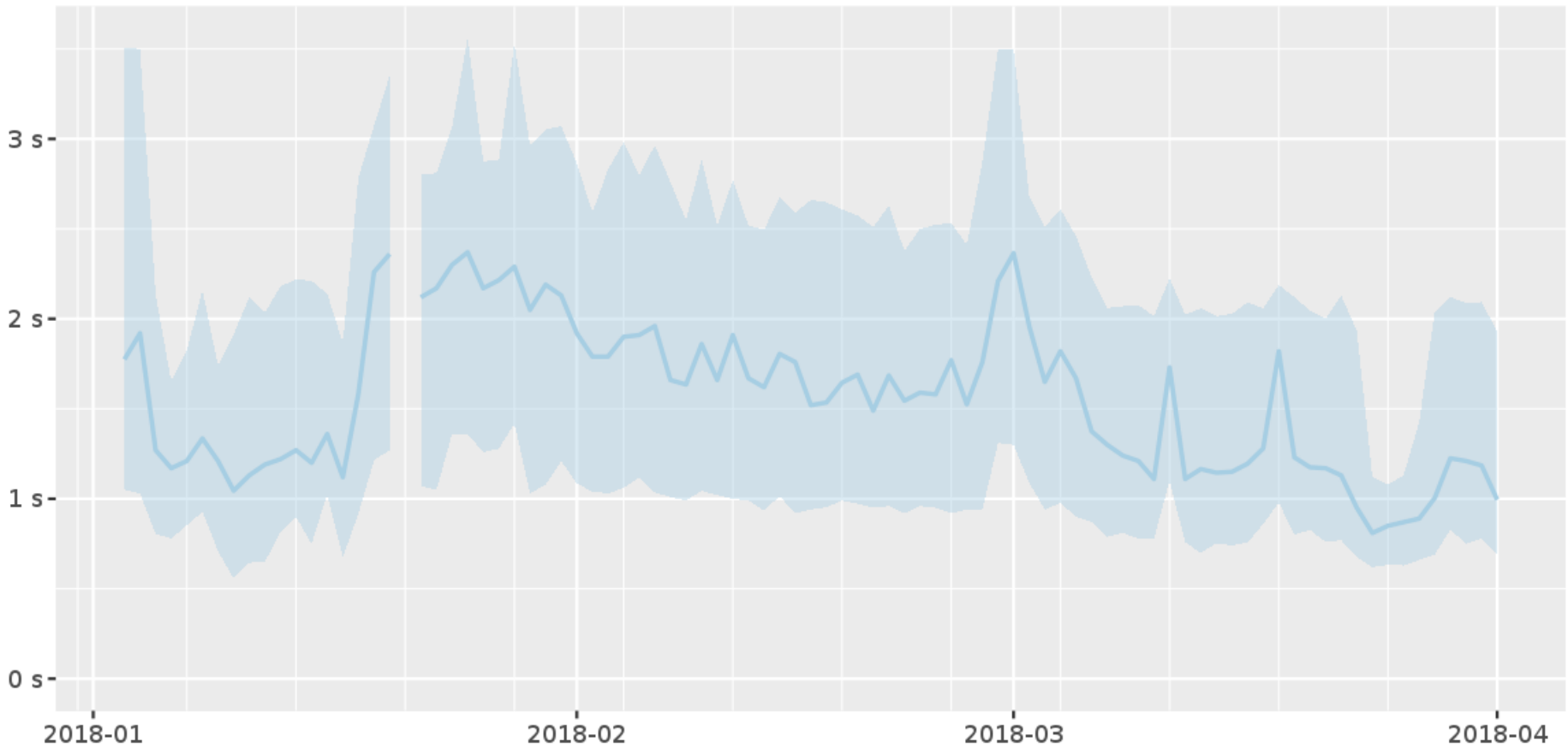
Measured times on all sources per day



Median

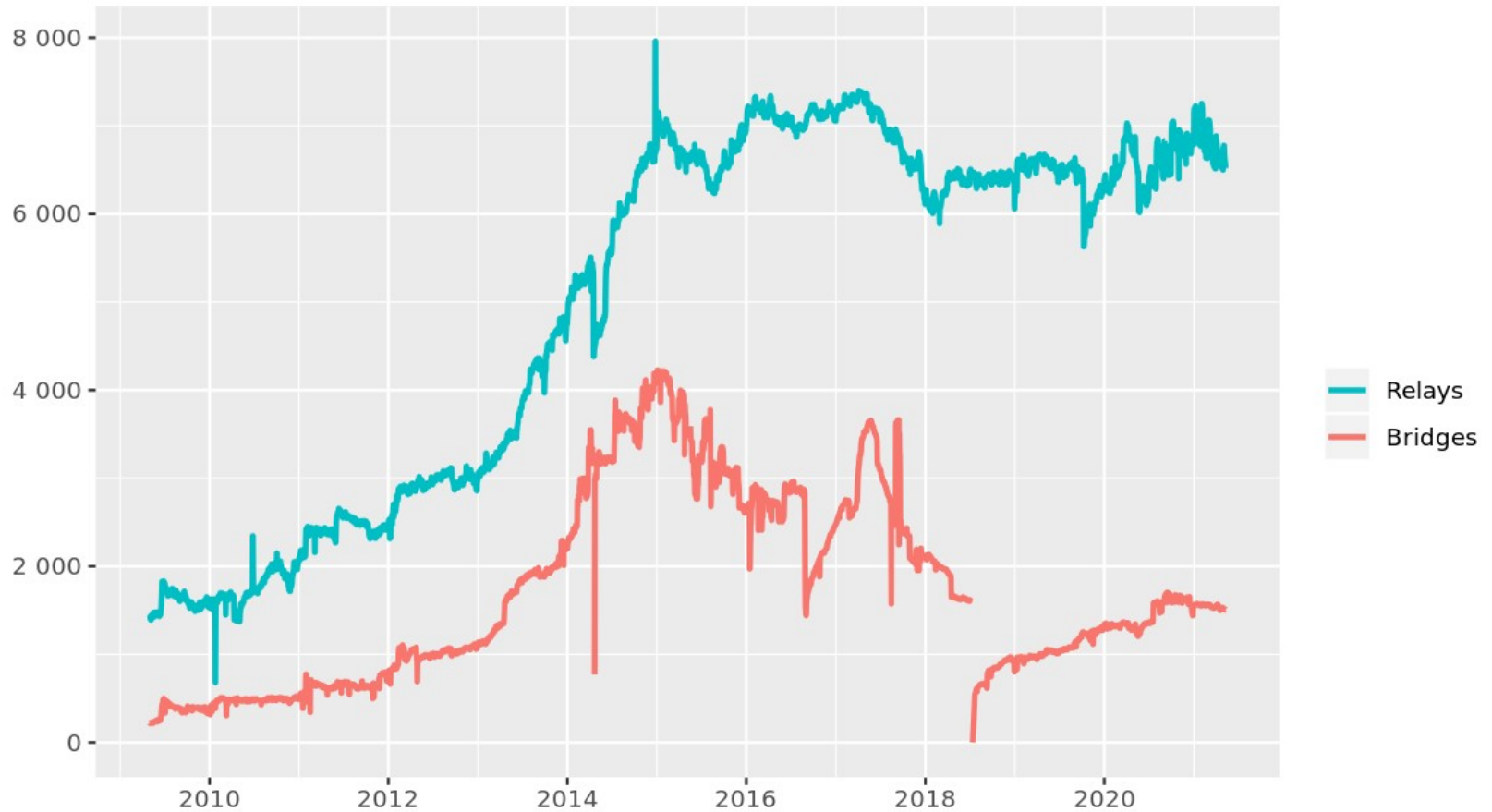


1st to 3rd quartile



ZAHL DER RELAYS UND BRIDGES VON TOR

Number of relays



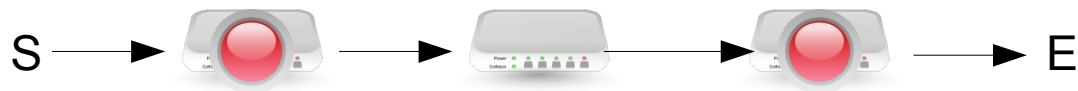
The Tor Project - <https://metrics.torproject.org/>

GRENZEN DES ANSATZES

- Datenspuren auf verschiedenen 'logischen Ebenen'
 - IP-Adresse ist nicht alles
 - Beispiel Internet
 - Web-Bugs, Cookies, Scripte etc. können Nutzer identifizieren
- einige praktische Probleme
 - Endpoints-Angriff
 - Route Selection-Angriff
 - Selective Disruption-Angriff
 - Seitenkanal-Angriffe wie Clock Skew
 - Performanz

ENDPOINTS-ANGRIFF

- Angreifer kontrolliert den ersten und letzten Knoten einer Verbindung
 - Statistischer Angriff über Paketzahl und zeitliche Abfolge



- Angreifer kontrolliert x von N Knoten, v Verbindungen:
Wahrscheinlichkeit p , beide Endpunkte zu kontrollieren ist

$$p = v * (x/n)^2$$

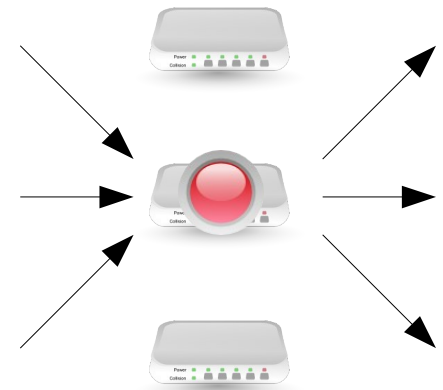
für große $v \rightarrow 100\%$

- Abhilfe: **Entry Guards**
 - Auswahl von maximal 3 Tor-Knoten als Entry Guards
 - Auswahl nach freien Ressourcen, Zuverlässigkeit, Unabhängigkeit
 - Einstiegsknoten werden aus Menge der Entry Guards gewählt
 - Solange Entry Guards nicht vom Angreifer übernommen, ist TOR-Route sicher

Paul Syverson et al.: Towards an Analysis of Onion Routing Security. Workshop on Design Issues in Anonymity and Unobservability, 2000

ROUTE SELECTION-ANGRIFF

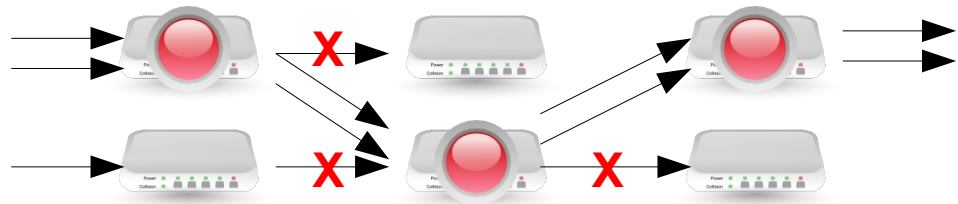
- Auswahl von Tor-Routen entsprechend freier Übertragungskapazität
 - Tor-Knoten geben dafür selbst freie Kapazität an
- Route Selection Angriff
 - einschleusen von kompromittierten Tor-Knoten mit sehr vielen freien Ressourcen (oder Manipulation dieser Angaben)
 - diese Knoten werden von Tor bevorzugt genutzt
→ weniger Rechner können viele Daten abfangen
- Lösung (seit 2009): Kontrolle der Ressourcenangaben durch vertrauenswürdige Server „von außen“
 - Aber: Aufwand für den Angreifer nur größer geworden, nicht unmöglich



Øverlier, L. et al.: Locating hidden servers. Proceedings of the IEEE Symposium on Security and Privacy, 2006

SELECTIVE DISRUPTION-ANGRIFF

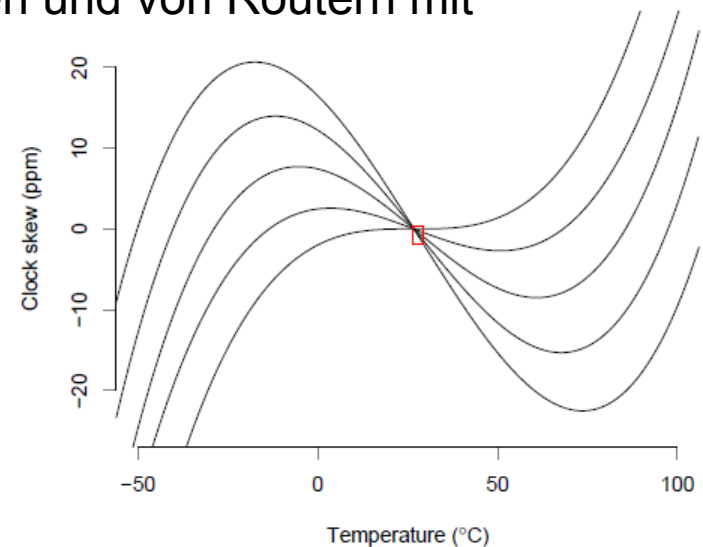
- Eine Tor-Route ist sicher, solange nicht alle Knoten auf einem Pfad kompromittiert sind
- Selective Disruption-Angriff
 - Ein Teil aller Router im Tor-Netz ist kompromittiert
 - Ein kompromittierter Router stört absichtlich alle Pfade, die ehrliche Router enthalten
 - Tor baut neue Routen auf
→ Wahrscheinlichkeit, dass eine Route nur noch kompromittierte Knoten enthält, steigt



*Kevin Bauer et al.: On the Optimal Path Length for Tor.
Hot Topics in Privacy Enhancing Technologies (HotPETS), 2010*

SEITENKANALANGRIFFE

- Beispiel: TCP-Timestamps werden vom Network Stack des Betriebssystems genutzt,
 - um Roundtrip-Zeiten zu schätzen
 - um Fehler in der Reihenfolge von Paketsequenzen zu entdecken
 - werden von Firewalls durchgelassen und von Routern mit weitergeleitet
- Clock Skew-Angriff
 - Abweichungen der Zeit hängen von Hardware und sogar Temperatur des Rechners ab
 - → Fingerabdruck des Rechners; statistische Angriffe auf die Tor-Route möglich



Stephen Murdoch et al.: Hot or not: Revealing hidden services by their clock skew, ACM Conference on Computer and Communications Security, 2006

VERGLEICH DER ANSÄTZE SOWEIT

Quelle:
<https://www.anonym-surfen.de>,
 d.h., der Betreiber
 von JonDo

	 JonDonym	 TOR	 VPN	 PROXY
Vertraulichkeit	↑ unabhängige Betreiber	→ zufällige Verteilung	↓ einzelner Anbieter	↓ einzelner Anbieter
Geschwindigkeit	↑ hoch (Premium)	↓ gering	↑ hoch	→ eher gering
Internationale Verteilung	↑ 2-3 Länder	↗ 1-3 Länder	↓ einzelner Server	↓ einzelner Server
Sicherer Browser	↑ JonDoFox & JonDoBrowser	↑ TorBrowser	↓ keiner	↓ keiner
Anonymisierbare Aktivitäten	↗ fast alles* (Premium)	↗ fast alles*	↑ alles	↓ i.d.R. nur Surfen
Schutz gegen Website-Profile	↑ jeder Zugriff einzeln	→ 10-minütiger Wechsel	↓ Profiling möglich	↓ Profiling möglich
Technik der Anonymisierung	↑ Forschung, Open Source	↑ Forschung, Open Source	↓ Closed Source	→ Standard-Techniken
Verschlüsselte Kommunikation	↑ mehrfach	↑ mehrfach	↗ einfach	↓ i.d.R. keine

* JonDonym und Tor können VoIP und P2P-Filesharing nicht sicher anonymisieren, SMTP-Port 25 ist gesperrt.

Es gibt hier keinen Vergleich mit Web(seiten)-Proxies (Anonymouse, Proxify, Megaproxy,...).
 Wie Sie in unserem IP-Check sehen können ↗, haben diese einen klaren Nachteil: sie funktionieren nicht.

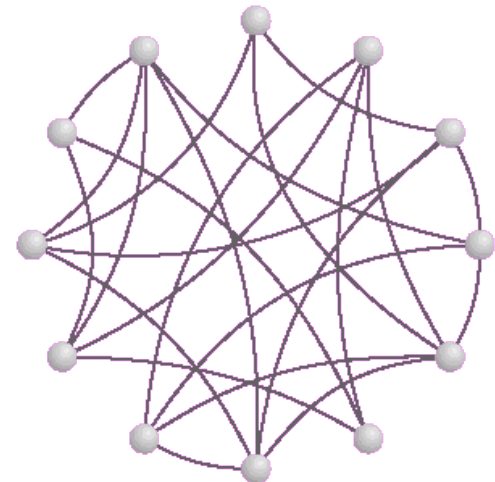
FREENET

- **Drei Ziele: Zensurreistenz, anonymer Datenaustausch, plausible Abstreitbarkeit**
 - TOR Hidden Services: es existiert ein Server im Netz, der Daten speichert und einer Person zugeordnet ist
 - Freenet: *Abstreitbarkeit* für Informationsanbieter
 - Daten anonym und verteilt speichern, Peer-to-Peer Ansatz
 - keine Quelle, die man vom Netz nehmen könnte → Löschung und Zensur unmöglich
- entwickelt seit 2000, Open Source
<http://freenetproject.org>
- *seit 2008 “Darknet” Feature, direkte Kommunikation nur mit vertrauenswürdigen Knoten*



DAS PEER-TO-PEER MODELL

- viele *unabhängige* Knoten
- jeder Knoten kennt einige ausgewählte (strukturierte P2P-Systeme) oder zufällige (unstrukturierte Systeme) andere Knoten
 - lokales, unvollst. Wissen über Netzwerktopologie
- jeder Knoten leitet Nachrichten an einen anderen weiter, der in der Netzwerktopologie näher am Ziel ist
 - Small-World-Eigenschaft (vgl. Milgram-Experiment '69 6 degrees of separation)



FREENET-ARCHITEKTUR

- Jeder Knoten speichert
 - Daten
 - Routing-Tabelle mit ausgewählten Knoten
 - IP-Adresse und gespeicherte Daten des Knotens
- Daten werden über **ortsunabhängige** Schlüssel referenziert
 - Daten dürfen zwischen den Knoten “umziehen”
 - häufig abgefragte Daten werden repliziert, unnötige gemäß Least-recently-used aus den Caches gelöscht
- Anfragen nach diesen Schlüsseln werden über Ketten von Freenet-Knoten weitergeleitet
 - Verschlüsselung zwischen den Knoten
 - Routing-Topologie lernt über die Zeit

REFERENZIERUNG ÜBER SCHLÜSSEL

- Zwei relevante Schlüsseltypen
 - **Content-hash key**
 - kryptische Signatur der Daten,
z.B.: CHK@SVbD9[...]4yFCB,bA7qLNJ[...]8kSi6bbNQ,AAEA--8
 - vergleichbar mit Inodes im Dateisystem
 - **Signed-subspace key**
 - Namespaces, in denen jeder lesen, aber nur der Ersteller schreiben kann
z.B: SSK@GB3wu[..]o-eHK35w,c63Ez[..]3YDduXDs,AQABAAE/mysite-4
 - vergleichbar mit Datei- und Verzeichnisnamen
- Verbreitung der Schlüssel als Lesezeichen, in öffentlichen Directories oder über Freenet-Suchmaschinen
- Die Schlüssel enthalten einen kryptographischen Schlüssel
 - es ist unmöglich, die Daten auf der eigenen Festplatte zu entschlüsseln

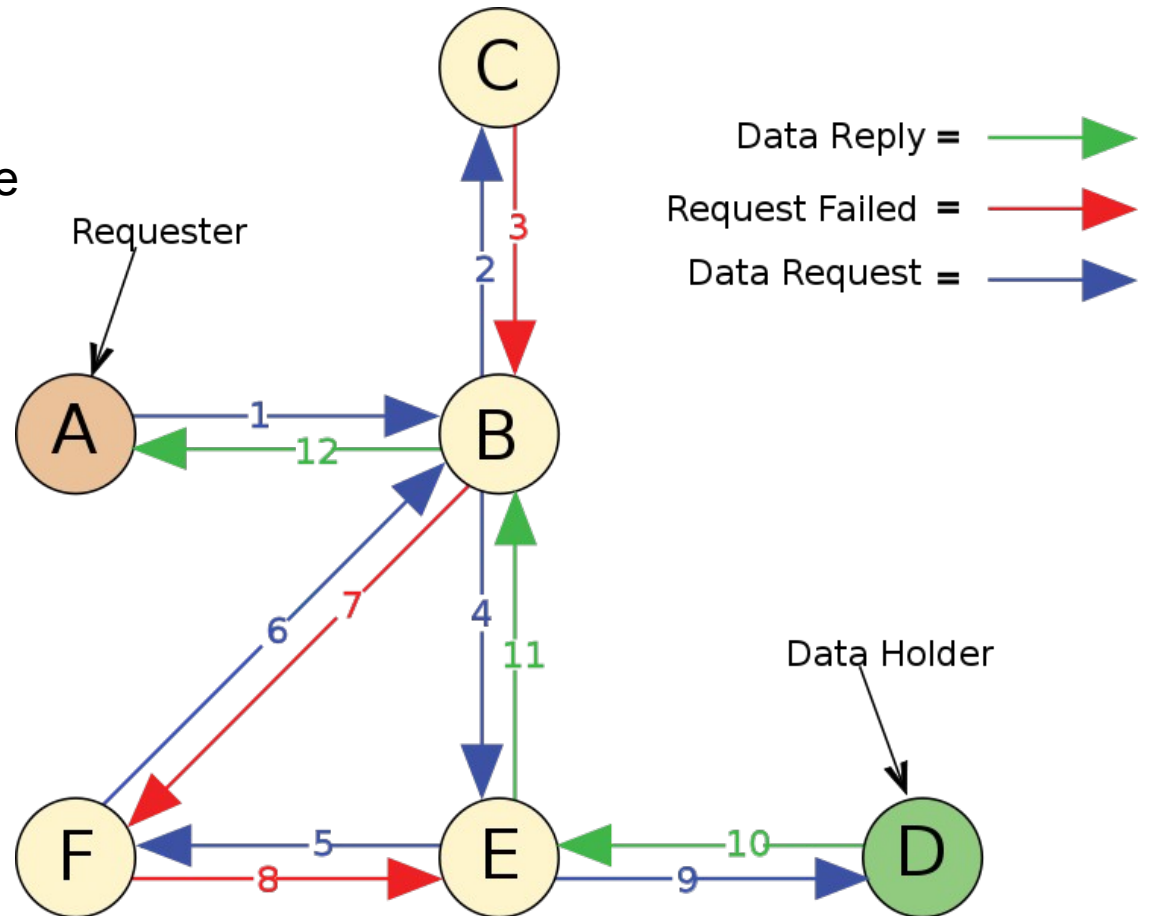
DATEN ABRUFEN (1/2)

1. Schlüssel beschaffen
2. Anfrage mit Schlüssel an einen Freenet-Knoten senden
3. Knoten erhält Anfragen
 - merken, von wem die Nachricht kam
 - zum Schlüssel passende Daten lokal gespeichert?
 - Ja: Antwort zurücksenden
 - Nein: Anfrage an den Knoten senden, dessen Schlüsselbereiche dem gesuchten Schlüssel am ähnlichsten sind (*nächste Folie*)
4. Knoten leitet Antwort zurück
 - Daten im eigenen Repository zwischenspeichern, anderen mitteilen, dass er nun diese Daten kennt
 - Routingtabelle updaten

DATEN ABRUFEN (2/2)

- Query Routing in Freenet: Hillclimbing + Backtracking

- erste Anfrage eines neuen Knotens wird zufällig geroutet
- funktioniert besser, je öfter die Knoten die Routingtabelle geupdatet haben



ANONYMITÄT IN FREENET

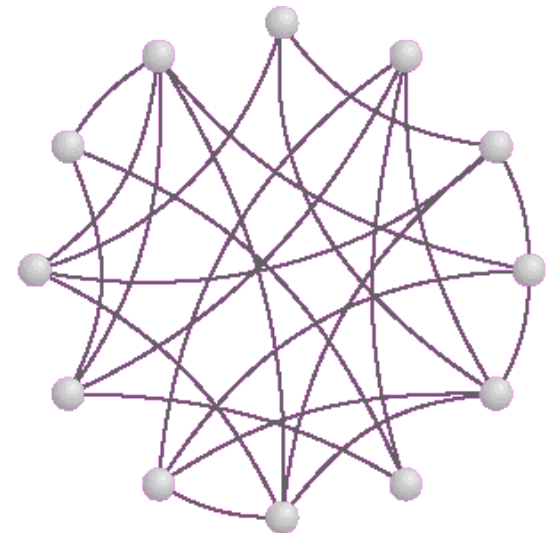
- Anfrager bleibt anonym
 - Nachricht sucht sich ihr Ziel anhand Schlüssel selbst,
 - Verbindungen nur jeweils zwischen zwei Knoten,
 - dem Schlüssel ist nicht anzusehen, was er bezeichnet
- Ersteller der Daten bleibt anonym
 - Daten werden durch ihre Schlüssel adressiert, und an beliebiger Stelle ins Freenet eingespeist
- Zensurresistenz
 - Daten können prinzipiell an beliebigen Stellen gespeichert werden
 - beliebte Daten häufig in den Caches der Knoten
 - Daten werden nur verschlüsselt abgespeichert, d.h. Freenet-Betreiber weiß nicht welche Daten er hostet

DAS DARKNET-FEATURE IN FREENET

- Grundsätzliches Problem in allen anonymen Netzen: das Einschleusen von manipulierten Knoten
 - mit wenigen Rechnern/Ressourcen gezielt angreifen, wenn “die richtigen Stellen” in der Netztopologie bekannt/erraten
- Opennet
 - offenes Freenet, jeder kann direkt teilnehmen
 - Freenet-Knoten konnektieren automatisch den nächsten Knoten in Richtung des Ziels
 - Installation enthält eine Liste offener Knoten
- Darknet
 - abgeschlossenes, privates Friend-to-Friend-Netzwerk
 - manuelle Routingtabellen, keine Kommunikation mit unbekanntem Knoten
 - Einschleusen erschwert
 - Herausforderung: trotzdem ein verbundenes Netzwerk schaffen

UMSETZUNG VON DARKNET

- Routingtabellen werden nur noch von Hand angelegt
 - Betreiber müssen sich persönlich kennen
- Implementierung eines strukturierten Overlays
 - Abbildung aller Schlüssel auf den Wertebereich $[0,1]$
 - Jeder Darknet-Knoten erhält ein kleines Intervall aus dem Wertebereich
 - gezielte Weiterleitung möglich, solange Netzwerk verbunden
- Knoten können ihre Intervalle tauschen (swapping)
 - es bilden sich Cluster von Knoten, die
 - effizient kommunizieren und
 - ähnliche Daten speichern

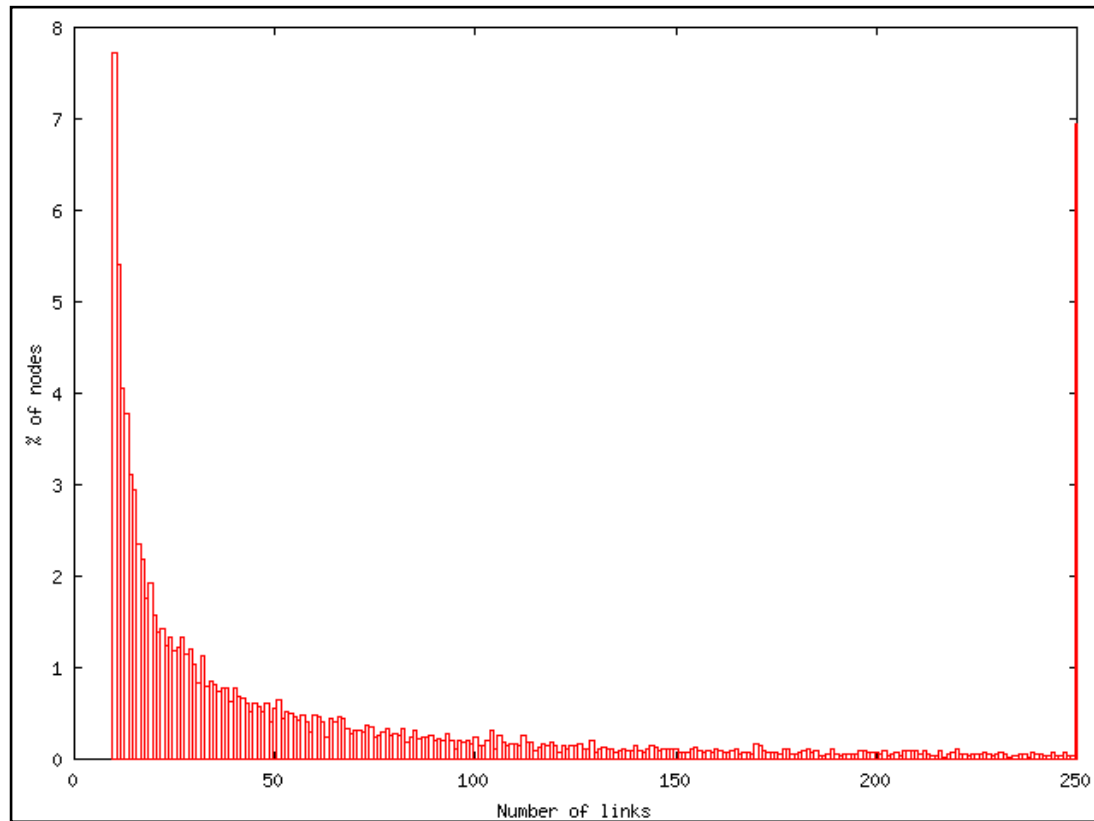


GRENZEN DES ANSATZES (1/2)

- Keine Garantien
 - Routing-Verfahren kann Daten nicht finden, obwohl sie im Netz gespeichert sind
 - selten gesuchte Daten können verschwinden (Least-Recently-Used-Ansatz)
 - der erste gefragte Knoten kann gleichzeitig Ersteller und Speicherort der Daten sein
 - erfährt unmittelbar, wer etwas wissen will
- Skalierbarkeit bzgl. Zahl der Knoten, Pfadlänge
 - Small-World-Ansatz ist wenig effizient; strukturierte P2P-Systeme wären besser
 - Darknet-Erweiterung löst das Problem teilweise
- Darknet erfordert, viele andere Darknet-Nutzer zu kennen

GRENZEN DES ANSATZES (2/2)

- Skalenfreies Netz, einige wenige Knoten wichtiger als andere
→ Angriffspunkte!



Quelle: Topics in Reliable Distributed Computing, Yoav Levy 2004

ZUSAMMENFASSUNG

ZUSAMMENFASSUNG

- Die IP-Adresse ist unvermeidlich, jedoch aus Datenschutzsicht problematisch
 - Quasi-Identifizier
 - ggf. Zuordnung zur Person möglich
- Datenschutzansätze basierend auf dem Konzept von Mixen:
 - JonDo
 - TOR
 - Freenet

LITERATUR

- [1] Roger Dingledine et al.: *Tor: The Second-Generation Onion Router*, Proceedings of the SSYM 2004
<http://freehaven.net/tor/tor-design.pdf>
- [2] Ian Clarke et al.; *Freenet: A Distributed Anonymous Information Storage and Retrieval System*, LNCS 2009
<http://citeseer.ist.psu.edu/old/clarke00freenet.html>

MÖGLICHE PRÜFUNGSFRAGEN

- Wie sieht das Datenpaket aus, das in einer Mix-Kaskade nach Chaum von Mix 2 an Mix 3 verschickt wird? Verwenden Sie die Symbole aus der Vorlesung (K, R, A)
- Erläutern Sie, ob und wie weit JonDo gegen Tracking im Netz hilft.
- Wozu werden bei den TOR Hidden Services die Rendezvous-Knoten benötigt? D.h., warum wird der Datenverkehr nicht einfach direkt über die Introduction Points aus dem Verzeichnisserver geleitet?
- Was ist ein ortsunabhängiger Schlüssel und wie ist er in Freenet umgesetzt?