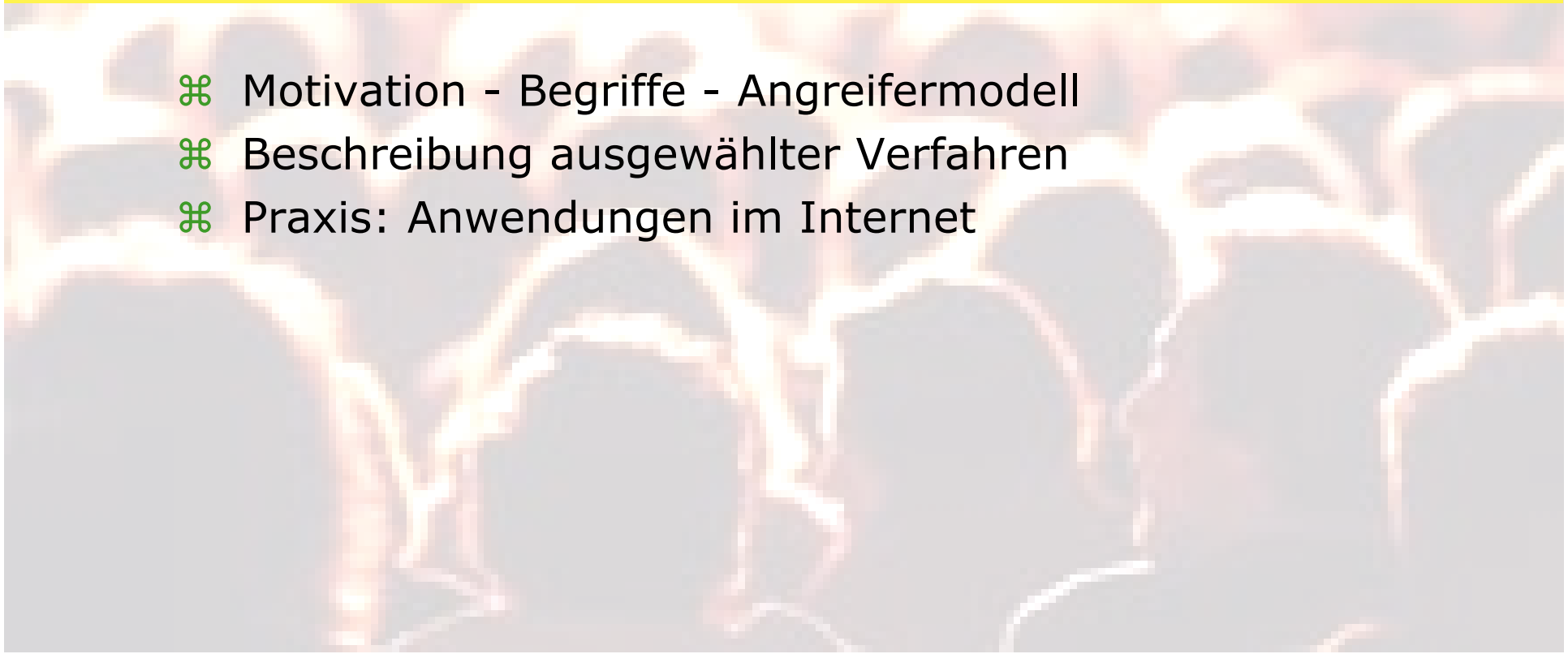
AN.ON
ANONYMITY.ONLINE

http://www.anon-online.de/

Verfahren zur Anonymität und Unbeobachtbarkeit im Internet

Hannes Federrath

http://www.inf.fu-berlin.de/~feder/



- Motivation - Begriffe - Angreifermodell
- Beschreibung ausgewählter Verfahren
- Praxis: Anwendungen im Internet

> Anonymität im Internet ist eine Illusion

- Wer ist der Gegner?
  - Konkurrenz
  - Geheimdienste fremder Länder
  - Big Brother
  - Systemadministrator
  - Nachbar ...

Funküberwachungsantenne (AN/FLR9)

> Anonymität im Internet ist eine Illusion

- Wer ist der Gegner?
  - Konkurrenz
  - Geheimdienste fremder Länder
  - Big Brother
  - Sys-admin
  - Nachbar ...

Bad Aibling Interception facility of the ECHELON system

Source: http://ig.cs.tu-berlin.de/w2000/ir1/referate2/b-1a/

> Anonymität im Internet ist eine Illusion

Electronic Mail: Log-Dateien zeigen Kommunikationsbeziehungen

```
>tail syslog
Oct 15 16:32:06 from=<feder@tcs.inf.tu-dresden.de>, size=1150
Oct 15 16:32:06 to=<hf2@irz.inf.tu-dresden.de>
```

World Wide Web: Log-Dateien zeigen Interessensdaten

```
wwwtcs.inf.tu-dresden.de>tail access_log
amadeus.inf.tu-dresden.de - - [15/Oct/1997:11:50:01] "GET
/lvbeschr/winter/TechnDS.html HTTP/1.0" - "http://wwwtcs.inf.tu-
dresden.de/IKT/" "Mozilla/3.01 (X11; I; SunOS 5.5.1 sun4u)"
```

Finger: Die Ermittlung eines Rechnerbenutzers ist kein Problem

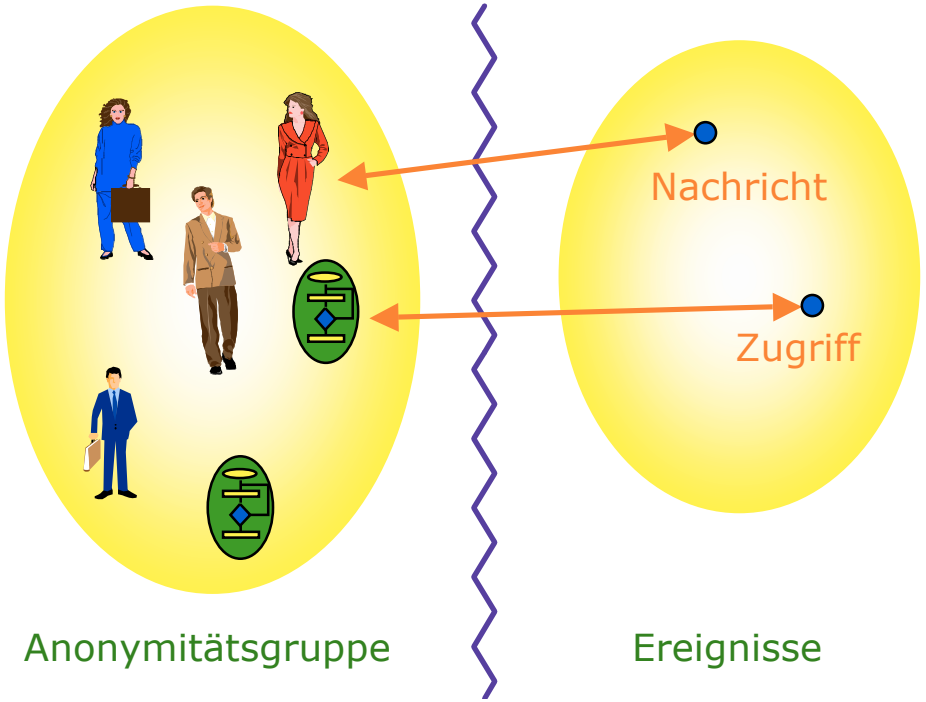
```
ithif19 logs 17 >finger @amadeus.inf.tu-dresden.de
[amadeus.inf.tu-dresden.de]
Login       Name                TTY         Idle    When
feder    Hannes Federrath       console             Wed 11:56
```

Vertraulichkeit; hier: Vertraulichkeit der Verkehrsdaten

- **Unbeobachtbarkeit**
  - Schutz von Sender und/oder Empfänger gegenüber allen Unbeteiligten (inkl. Netzbetreiber)
    - Niemand kann Kommunikationsbeziehungen verfolgen.
    - Unbeobachtbares Senden und/oder Empfangen von Nachrichten
- **Anonymität**
  - Schutz der Identität zusätzlich auch gegenüber dem Kommunikationspartner
    - Anonymität als *Sender* von Nachrichten
    - Anonymität als *Empfänger* von Nachrichten
- **Unverkettbarkeit**
  - Ereignisse werden vom Angreifer bzgl. des Senders und/oder Empfängers als unabhängig erkannt

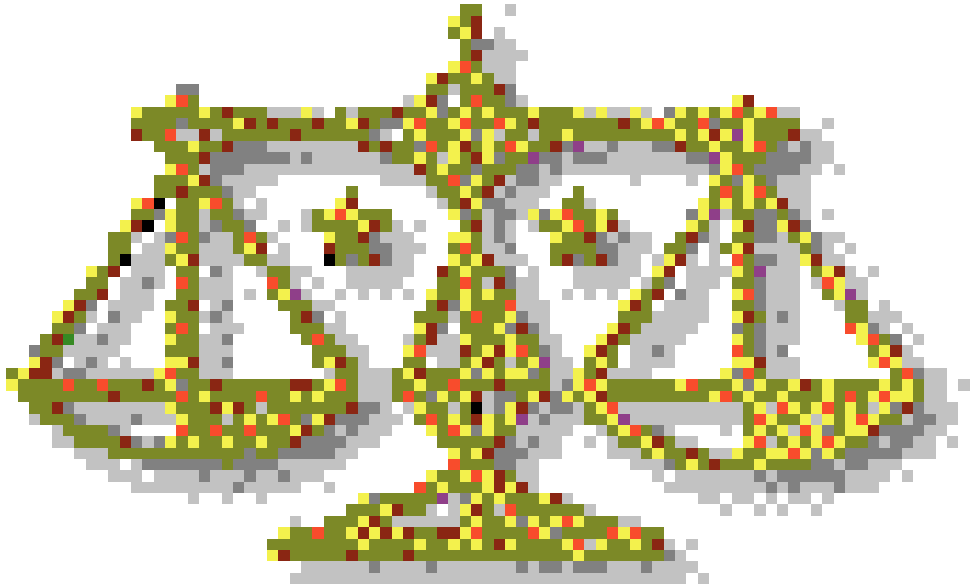
> Anonymitätsgruppe

- Ein einzelnes Ereignis, das durch eine einzelne Person hervorgerufen wurde, kann nicht anonym oder unbeobachtbar sein.
- Wir benötigen eine Gruppe von Personen, die sich alle gleich verhalten: **Anonymitätsgruppe**
  - Jedes Mitglied der Anonymitätsgruppe ist mit der gleichen Wahrscheinlichkeit der Urheber eines Ereignisses.
  - Eine öffentlich bekannte Eigenschaft, die alle Mitglieder der Anonymitätsgruppe erfüllen, kann nicht anonymisiert werden.

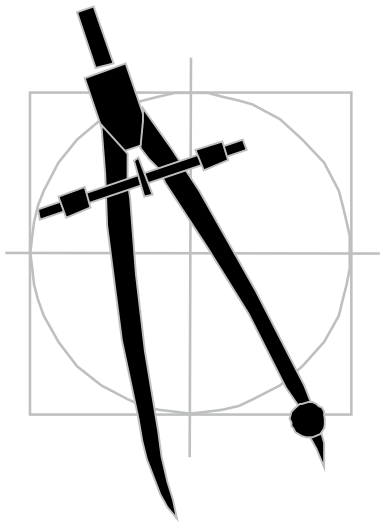


> Juristische Sicht

- Teledienstedatenschutzgesetz (TDDSG)
  - §3(4): Die Gestaltung und Auswahl technischer Einrichtungen für Teledienste hat sich an dem Ziel auszurichten, keine oder so wenige personenbezogene Daten wie möglich zu erheben, zu verarbeiten und zu nutzen.
  - §4(1): Der Diensteanbieter hat dem Nutzer die Inanspruchnahme von Telediensten und ihre Bezahlung anonym oder unter Pseudonym zu ermöglichen, soweit dies technisch möglich und zumutbar ist. Der Nutzer ist über diese Möglichkeiten zu informieren.



- Technischer Datenschutz
  - Systeme so konstruieren, dass unnötige Daten vermieden und nicht miteinander verkettet werden können.
- Zu verschleiern sind:
  - Adressen:
    - Sender, Empfänger, Kommunikationsbeziehung
  - Zeitliche Korrelationen:
    - Zeitpunkte, Dauer
  - Übertragenes Datenvolumen und inhaltliche Korrelationen
  - Orte:
    - Aufenthaltsorte, Bewegungsspuren

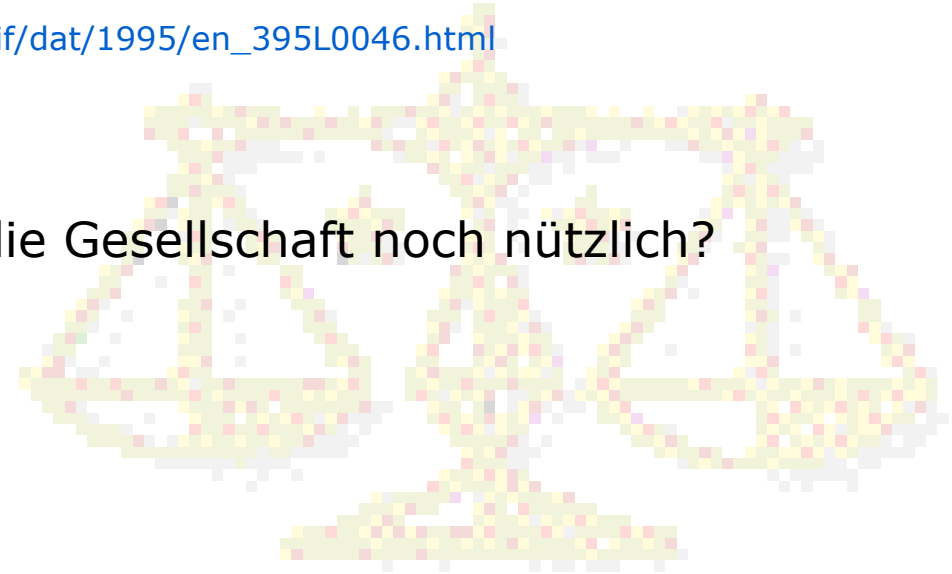


> Politisches und gesellschaftliches Umfeld

- Telekommunikationsüberwachung
  - Telekommunikationsüberwachungsverordnung (TKÜV)
    - http://www.bmwi.de/Homepage/download/telekommunikation_post/TKUEV-Entwurf.pdf
  - Cybercrime Convention
    - http://conventions.coe.int/Treaty/EN/projets/FinalCybercrime.htm

- Datenschutzgesetze
  - Neues Bundesdatenschutzgesetz (BDSG)
    - http://www.bfd.bund.de/information/bdsg_hinweis.html
  - EU-Datenschutzrichtlinie
    - http://europa.eu.int/eur-lex/en/lif/dat/1995/en_395L0046.html

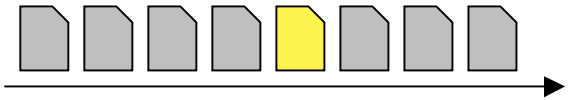
- Offene Frage
  - Wieviel Privatheit ist für die Gesellschaft noch nützlich?



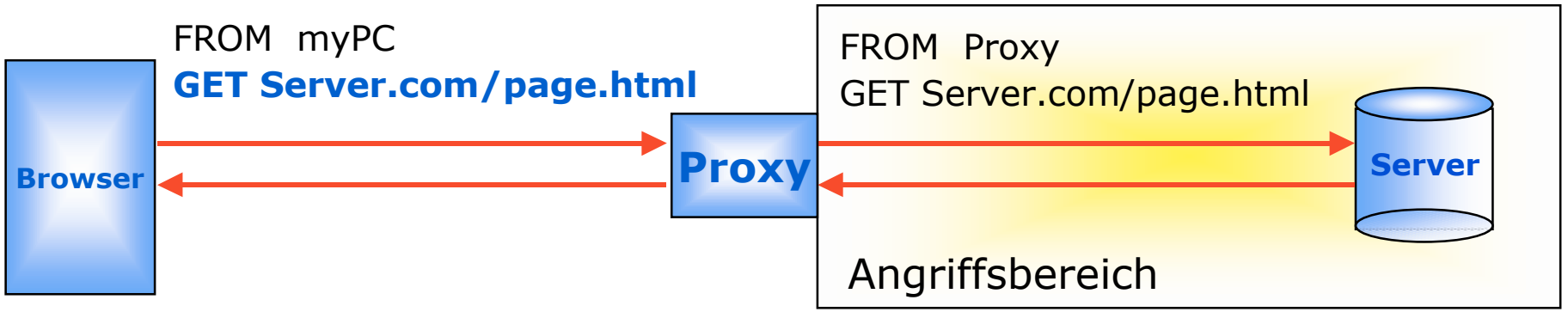
> Grunsätzliche Techniken (1)

- **Verteilung** (Broadcast) + implizite Adressierung
  - Schutz des Empfängers; alle erhalten alles
  - lokale Auswahl

- **Dummy Traffic:** Senden bedeutungsloser Nachrichten
  - Schutz des Senders



- **Proxies** zwischenschalten
  - Server erfährt nichts über Client, Proxy kann mitlesen



> Schutz des Empfängers: Verteilung (Broadcast)

- Adressierung
  - explizite Adressen: Routing
  - implizite Adressen: Merkmal für Station des Adressaten
    - verdeckt: Konzelationssystem
    - offen: Bsp. Zufallszahlengenerator

		Adreßverwaltung	
		öffentliche Adresse	private Adresse
implizite Adres-sierung	verdeckt	sehr aufwendig, für Kontaktaufnahme nötig	aufwendig
	offen	abzuraten	nach Kontaktaufnahme ständig wechseln

Implicit Addresses

- First contact: covered implicit address CIA
  - Recipient publishes public encryption key **c**
  - Sender creates **CIA := c (R , S , M)**
    - Redundancy **R**
    - Seed **S** of a pseudo-random generator **PRG**
    - Message **M** (optional)
  - Recipient decrypts *all* received messages with private key **d**
    - Finds correct **R** for own messages only
- Following addressing: open implicit address OIA
  - **OIA_{i+1} := PRG (i , S)** (i = 0,1,2,…)
  - Sender :
    - calculates next **OIA**
    - encrypts message **M** (optional)
    - Sends **OIA, M**
  - Receiver: Associative memory of all valid **OIA**s to recognize own messages

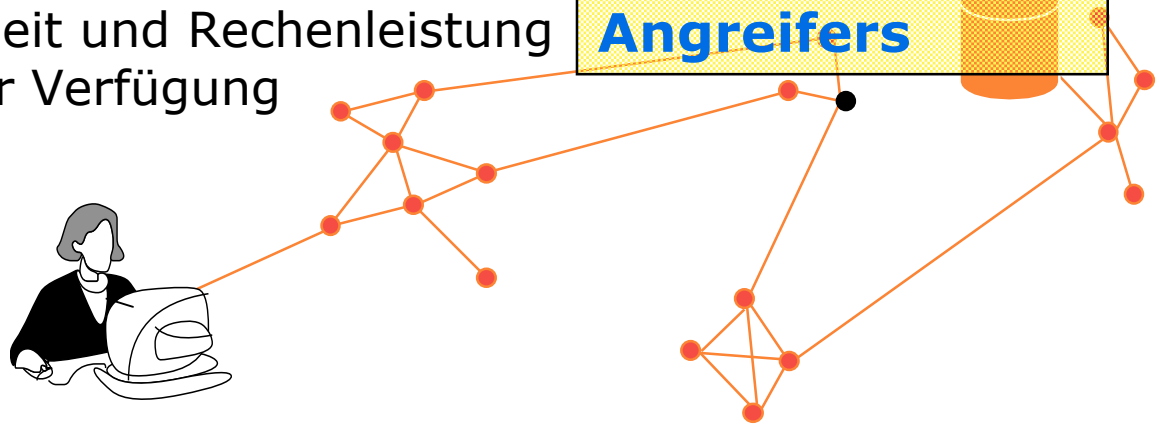
> Grundsätzliche Techniken (2)

- **DC-Netz:** kombiniert u.a. Broadcast, Kryptographie und Dummy Traffic
  - Schutz des Senders
- **Blind-Message-Service:** Unbeobachtbare Abfrage aus von unabhängigen Betreibern replizierten Datenbanken
  - Schutz des Clients
- **MIX-Netz:** kombiniert u.a. hintereinander geschaltete Proxies von unabhängigen Betreibern, Kryptographie und Dummy Traffic
  - Schutz der Kommunikationsbeziehung
  - Effizient in Vermittlungsnetzen
- **Steganographie**
  - Verbergen einer Nachricht in einer anderen

> Angreifermodell

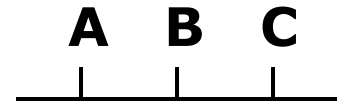
- Angreifer kann:
  - alle Kommunikationsleitungen passiv überwachen (Verkehrsanalysen durchführen),
  - ggf. aktiv eigene Nachrichten beisteuern,
  - ggf. fremde Nachrichten blockieren/verzögern,
  - ggf. selbst eine »Unbeobachtbarkeitsstation« sein
  - Empfänger oder Sender sein
- Angreifer kann nicht:
  - Kryptosysteme brechen,
  - in den persönlichen Rechner eindringen,
  - ist komplexitätstheoretisch beschränkt, d.h. hat begrenzte Zeit und Rechenleistung für seine Angriffe zur Verfügung

Ein sehr starkes Angreifermodell schützt vor dem Unterschätzen des realen Angreifers



> DC-Netz

Chaum, 1988



Broadcastnetz

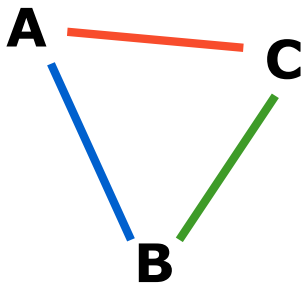
Echte Nachricht von A	00110101
Schlüssel mit B	00101011
Schlüssel mit C	00110110
Summe	00101000

A sendet 00101000

Leere Nachricht von B	00000000
Schlüssel mit A	00101011
Schlüssel mit C	01101111
Summe	01000100

B sendet 01000100

Schlüsselgraph

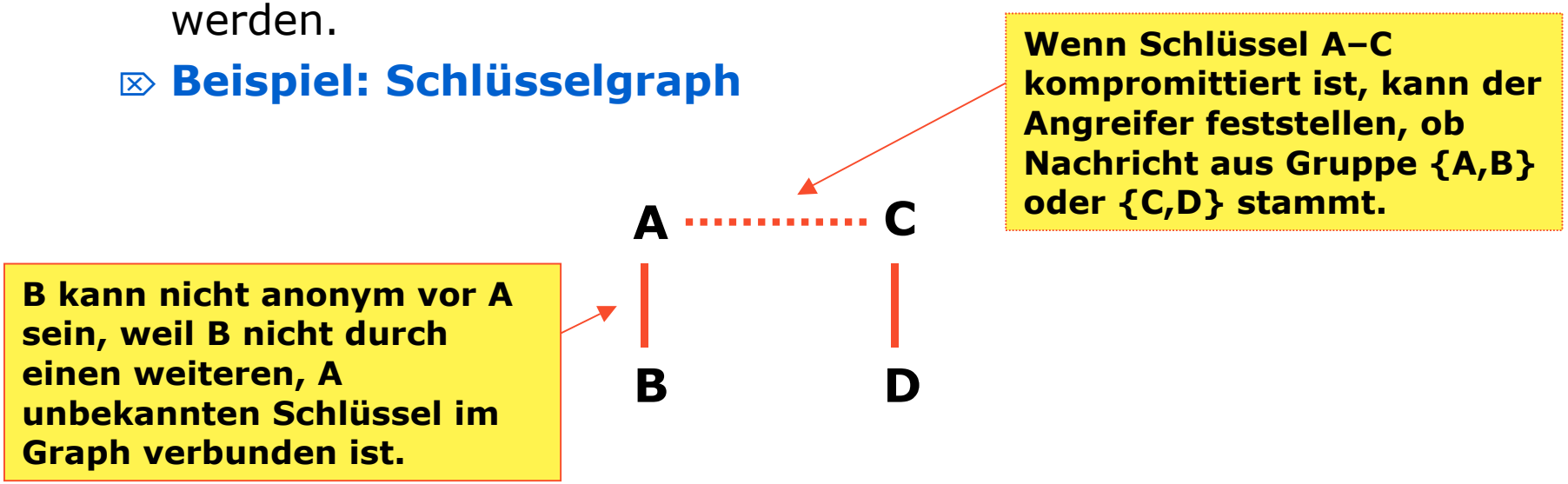


Leere Nachricht von C	00000000
Schlüssel mit A	00110110
Schlüssel mit B	01101111
Summe	01011001

C sendet 01011001

Summe = Echte Nachricht von A 00110101

- Perfekte Unbeobachtbarkeit des Sendens
- Erfordert Synchronisierung der Teilnehmer: Runden
- Zu jedem Zeitpunkt kann immer nur ein Teilnehmer senden
  - Kollisionserkennung und -auflösung nötig
- Bei Punkt-zu-Punkt-Kommunikation
  - implizite Adressierung und Verschlüsselung der Nachricht
- Sicherheitseigenschaft
  - Jede Nachricht ist innerhalb der Teilnehmer unbeobachtbar, die durch einen zusammenhängenden Schlüsselgraph gebildet werden.
  - **Beispiel: Schlüsselgraph**



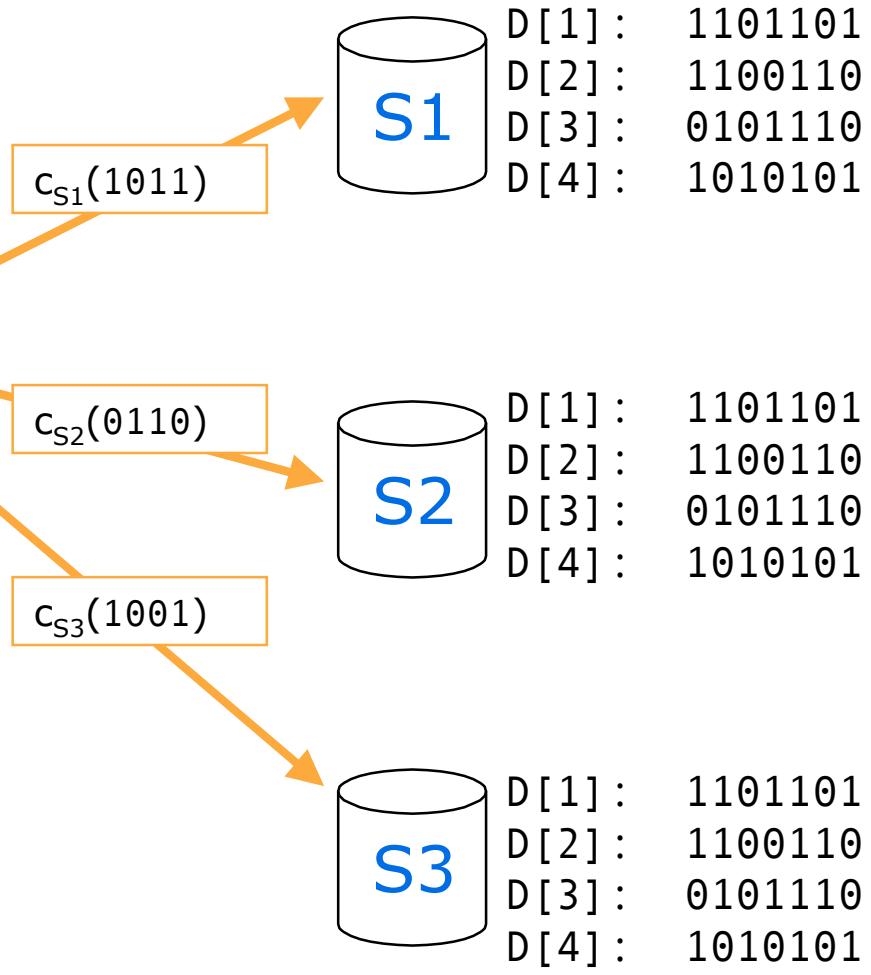
Blind-Message-Service: Anfrage

Cooper, Birman, 1995

Client interessiert sich für D[2]:

Index = 1234

Setze Vektor = 0100
Wähle zufällig request(S1) = 1011
Wähle zufällig request(S2) = 0110
Berechne request(S3) = 1001



- Schutzziel:
  - Client möchte auf Datenbestand zugreifen, ohne dass Datenbank erfährt, wofür sich der Client interessiert
- Replizierte Datenbanken mit unabhängigen Betreibern

> Blind-Message-Service: Antwort

Cooper, Birman, 1995

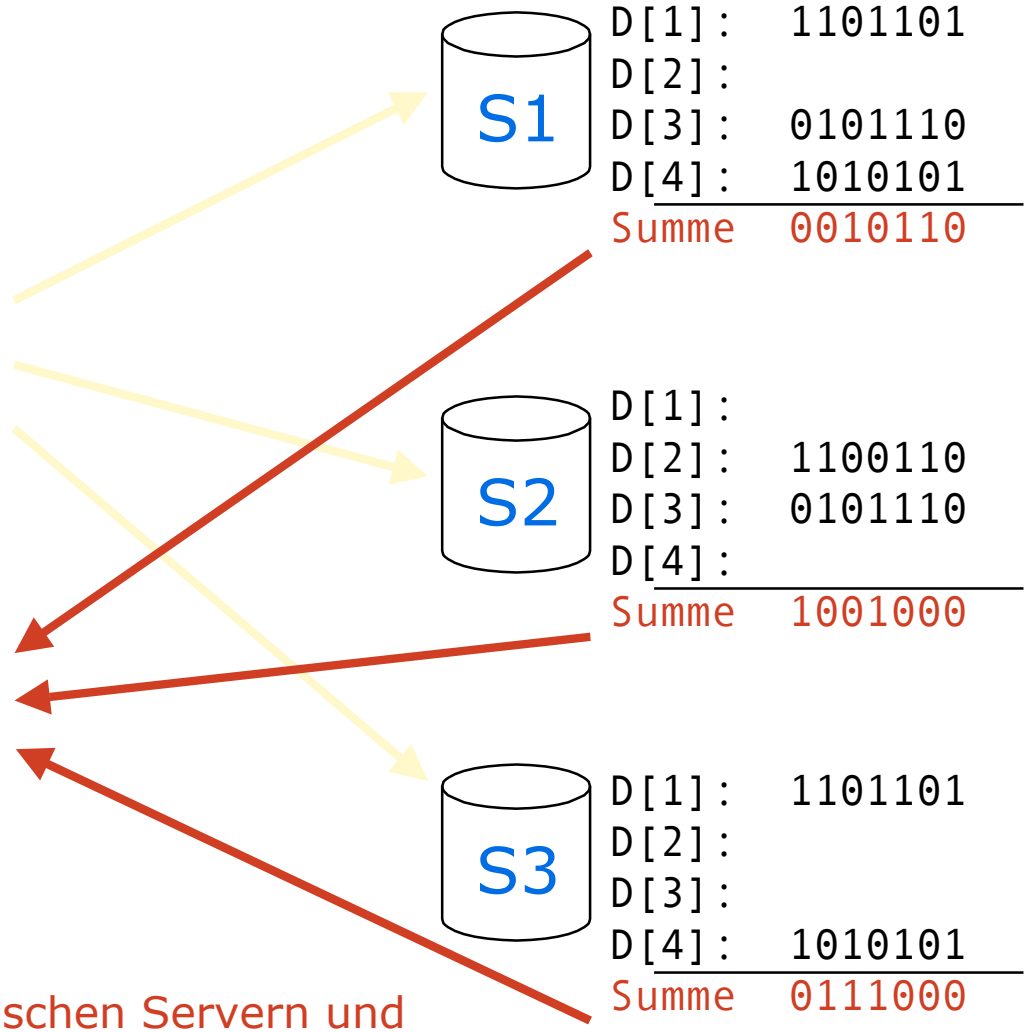
Client interessiert sich für D[2]:

Index = 1234

Setze Vektor = 0100
Wähle zufällig request(S1) = 1011
Wähle zufällig request(S2) = 0110
Berechne request(S3) = 1001

Antworten von
S1: 0010110
S2: 1001000
S3: 0111000

Summe entspricht D[2]: 1100110



Verbindungsverschlüsselung zwischen Servern und Client unbedingt notwendig

Optimierung des Private Message Service

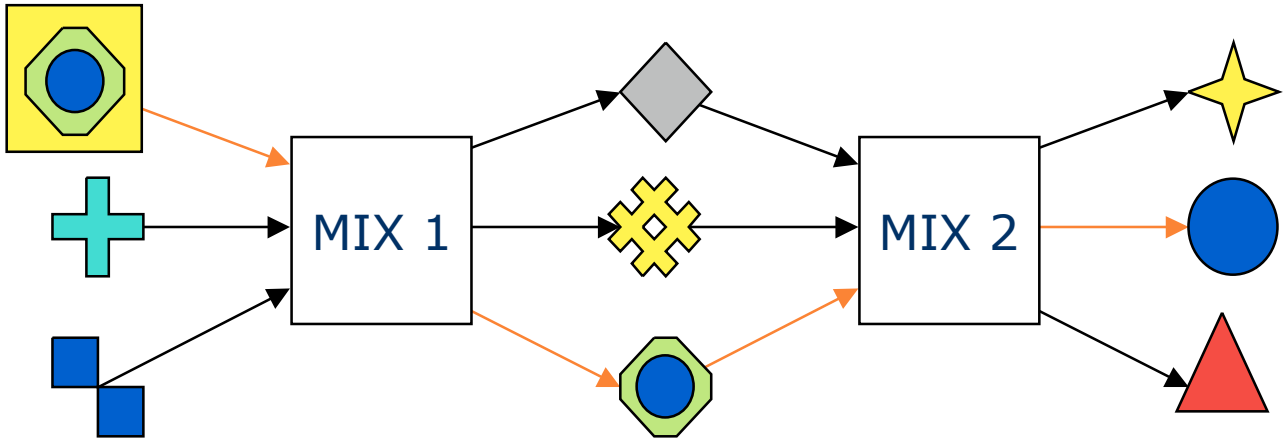
Berthold, Clauß, Köpsell, Pfitzmann 2001

- Optimierung der Kommunikation Client --> Server
  - Verwendung von Pseudozufallsgeneratoren und Registrierungsmechanismus
- Optimierung der Kommunikation Server --> Client
  - Pseudo-One-Time-Pad-Verschlüsselung und Ergebnisüberlagerung auf Serverseite
- Effizientes Abfrageverfahren für Gruppen von Speicherzellen
- Kommunikationskomplexität (ohne Registriermechanismus):
  - D = n + k Bit
    - D:gesamte Anzahl der transferierten Bits (beide Ri.)
    - n:Anzahl der Speicherzellen der Datenbank
    - k:Größe einer Speicherzelle der Datenbank
  - Komplexität vorher: D = m (n + k)
    - m:Anzahl der abgefragten Server
  - Neues Verfahren ist invariant gegenüber Sicherheitsparameter; dafür wurde die informationstheoretische Sicherheit des Grundverfahrens geopfert.

Mix-Netz

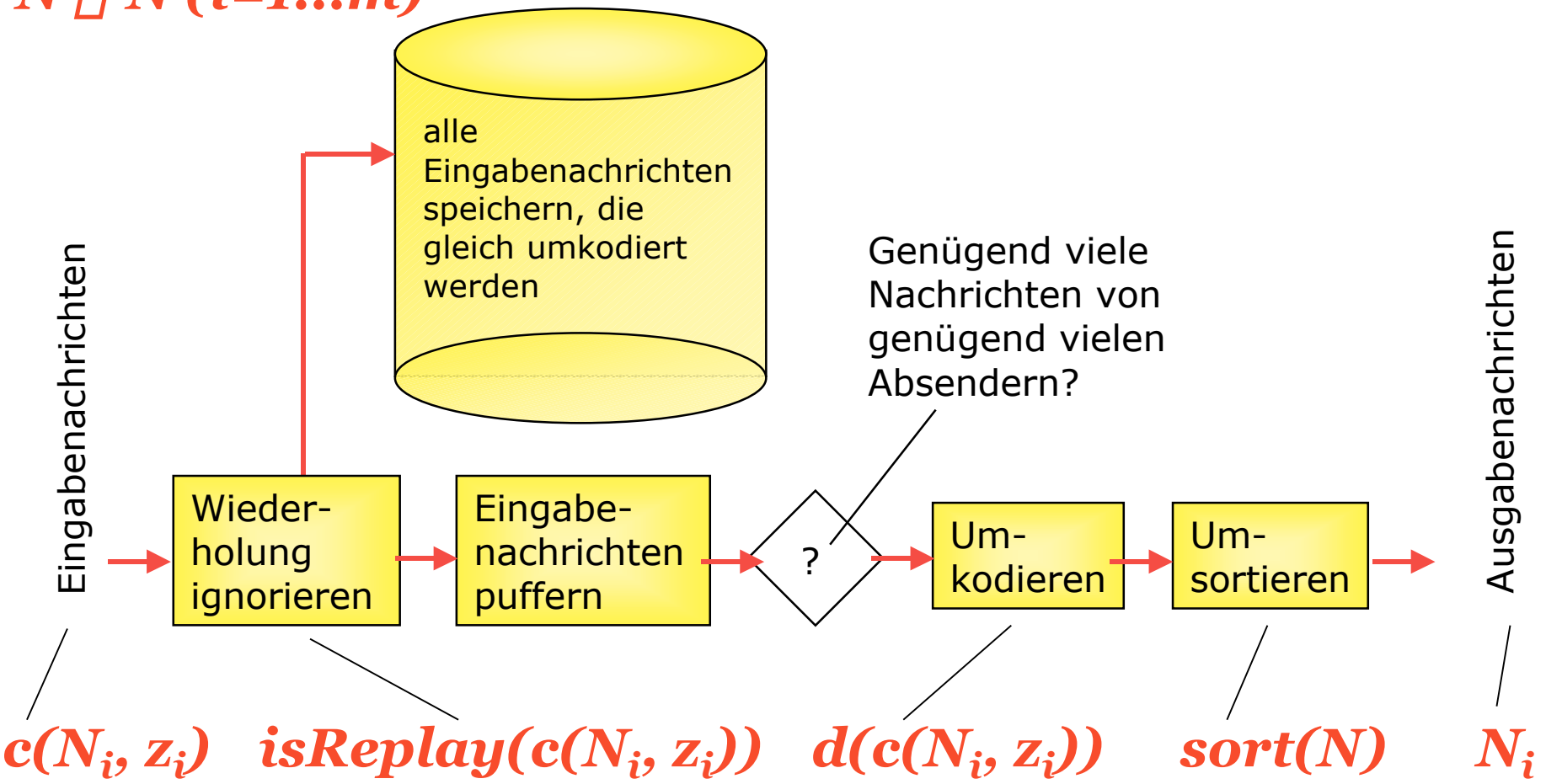
Chaum, 1981

- Grundidee:
  - Nachrichten in einem »Schub«
    - sammeln, Wiederholungen ignorieren, umkodieren, umsortieren, gemeinsam ausgeben.
  - Alle Nachrichten haben die gleiche Länge.
  - Mehr als einen Mix verwenden.
  - Wenigstens ein Mix darf nicht angreifen.
- Schutzziel:
  - Unverkettbarkeit von Sender und Empfänger
  - Schutz der Kommunikationsbeziehung
  - Zuordnung zwischen E- und A-Nachrichten wird verborgen



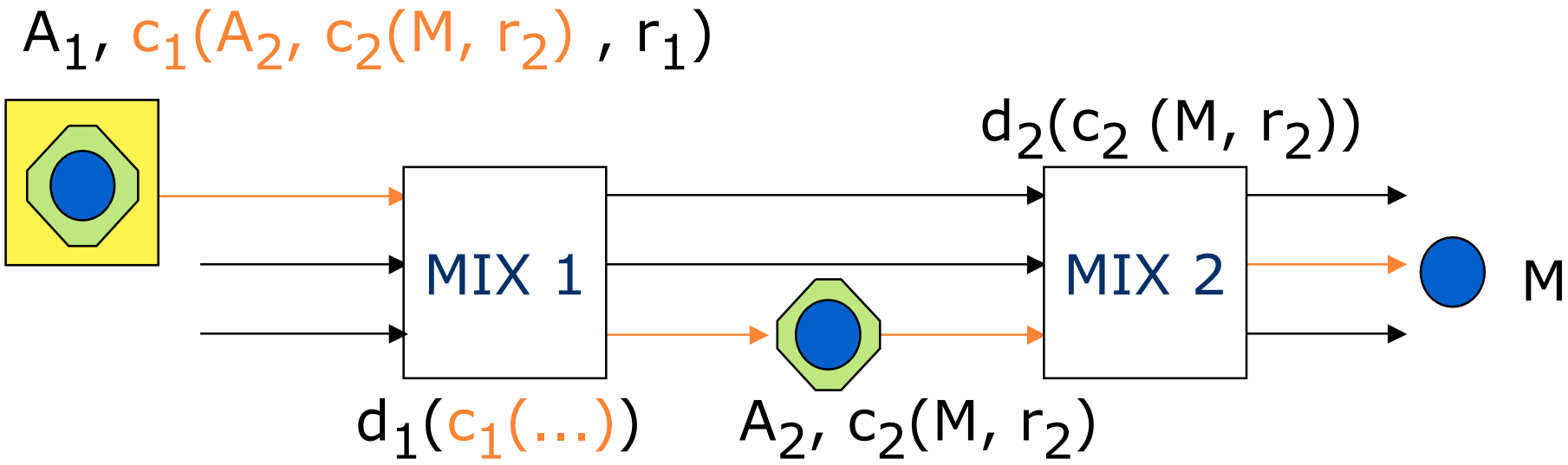
> Blockschaltbild eines Mix

$N = \{N_1, N_2, \ldots, N_m\}$
$N \sqsubset N \ (i=1\ldots m)$

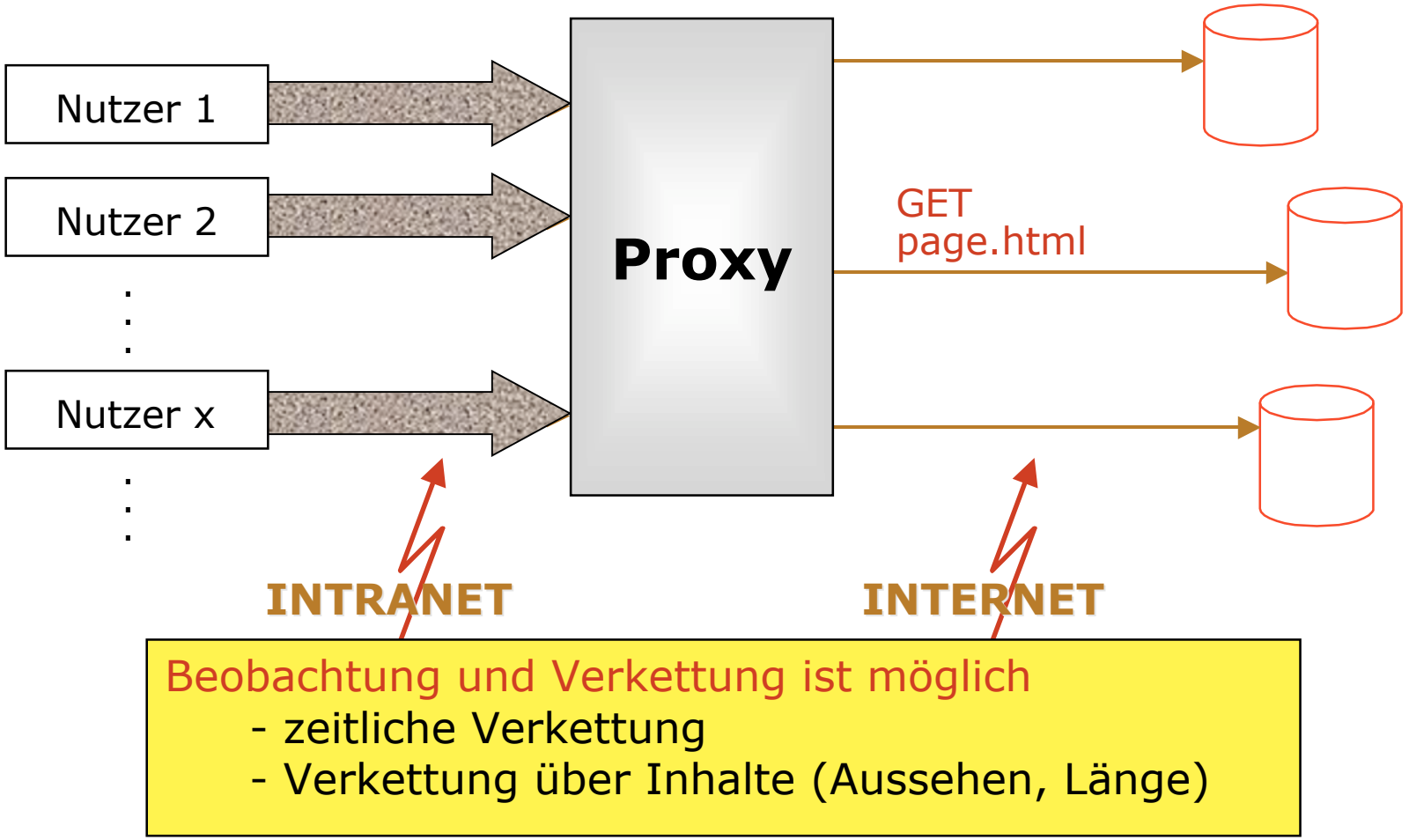


Kryptographische Operationen eines Mix

- Verwendet **asymmetrisches Konzelationssystem**
  - $c_i(...)$ Verschlüsselungsfunktion für Mix i
    - Jeder kann den öffentlichen Schlüssel c_i verwenden
  - $d_i(...)$ private Entschlüsselung von Mix i
    - Nur Mix i kann entschlüsseln
  - A_i Adresse von Mix i
  - r_i Zufallszahl (verbleibt im Mix, wird »weggeworfen«)
  - M (verschlüsselte) Nachricht für Empfänger (inkl. seiner Adresse)



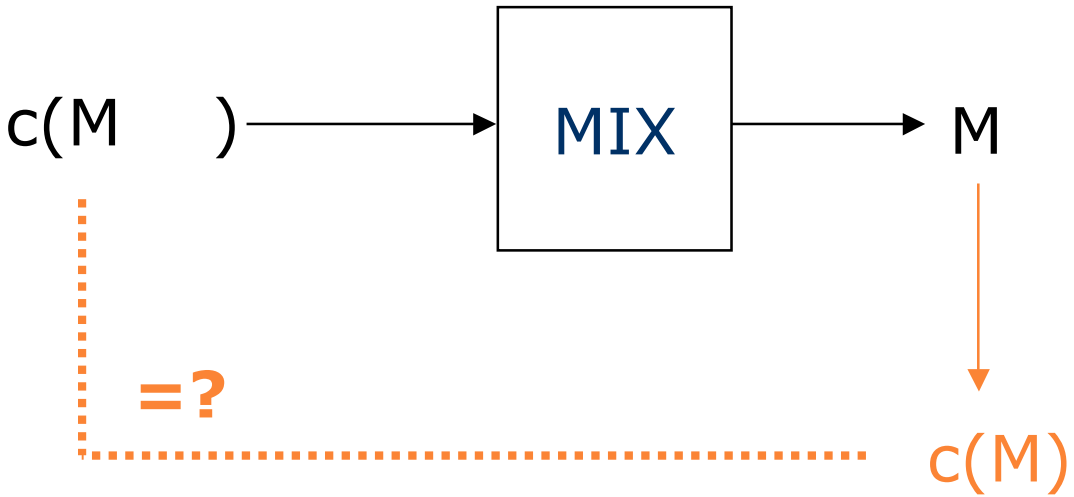
> Mixe: Warum überhaupt umkodieren?



Verschlüsselung zwischen Browser und Proxy verhindert Korrelation über »Aussehen«, aber nicht über Nachrichtenlänge und Zeit und hilft nichts gegen den Proxy.

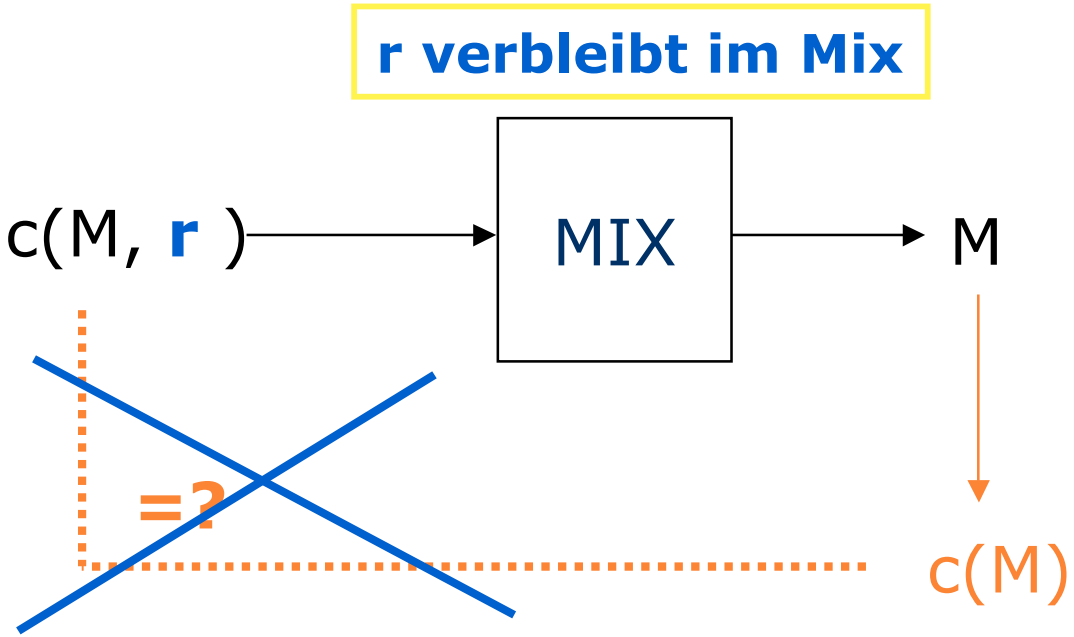
> Mixe: Warum die Zufallszahlen?

- Wenn keine Zufallszahlen r verwendet werden:
  - Jeder kann die Ausgabenachrichten probeweise verschlüsseln, weil c(...) öffentlich ist.
  - Vergleich mit allen eingehenden Nachrichten führt zur Verkettung
  - Es wird ein **indeterministisches Kryptosystem** benötigt
- Eingabenachricht: $c_1(A_2, c_2(M, r_2) , r_1)$



>> Mixe: Warum die Zufallszahlen?

- Wenn keine Zufallszahlen r verwendet werden:
  - Jeder kann die Ausgabenachrichten probeweise verschlüsseln, weil c(...) öffentlich ist.
  - Vergleich mit allen eingehenden Nachrichten führt zur Verkettung
  - Es wird ein **indeterministisches Kryptosystem** benötigt
- Eingabenachricht: $c_1(A_2, c_2(M, r_2) , r_1)$



> Mixe: Warum mehr als ein Mix?

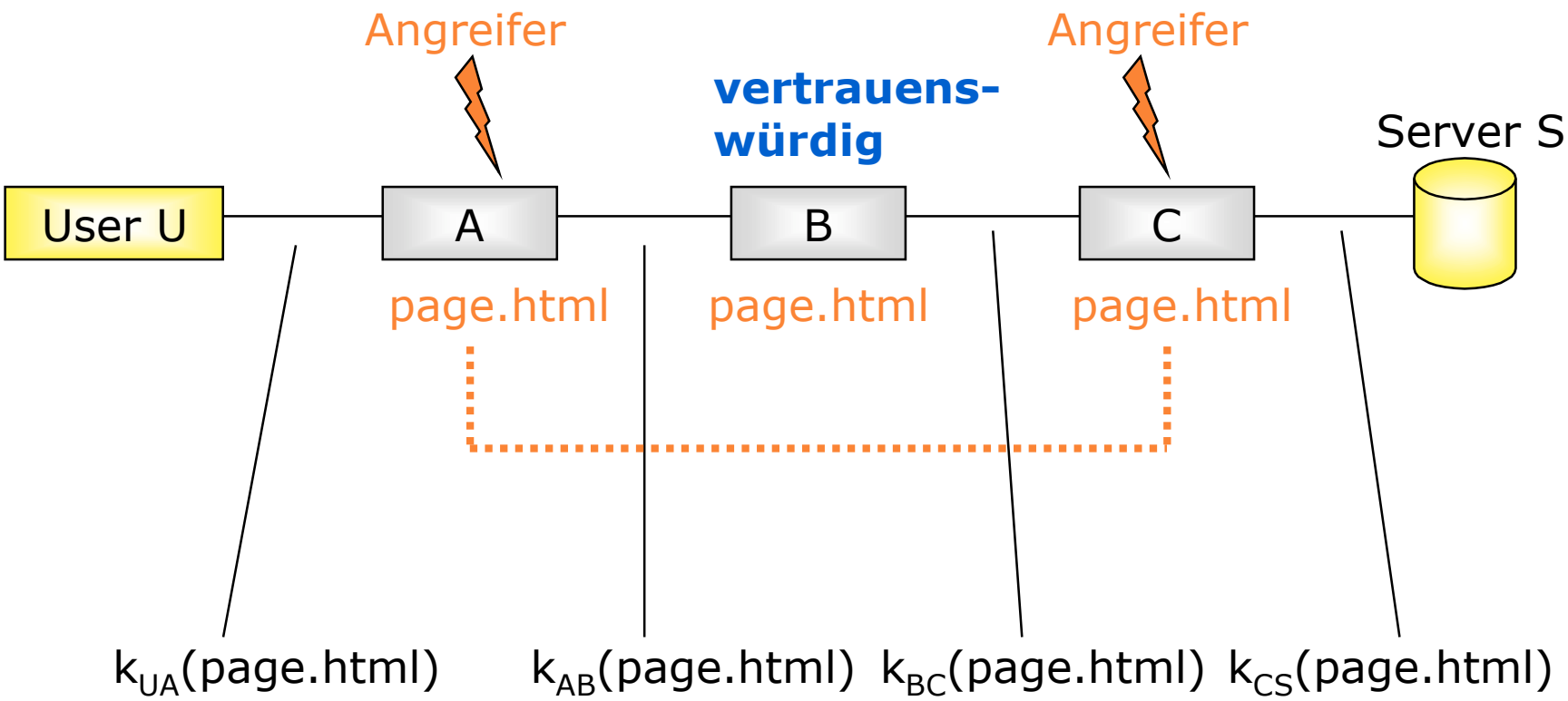
- Schutzziel: Auch Mix soll nicht beobachten können
  - Ein einzelner Mix kennt jedoch E-A-Zuordnung
- Verwende mindestens zwei Mixe
  - erster Mix kennt Sender
  - letzter Mix kennt Empfänger
- Allgemein:
  - Verwende so viele Mixe, dass sich in der Mix-Kette wenigstens ein Dir vertrauenswürdiger Mix befindet
- Praxis:
  - Je mehr Mixe verwendet werden, umso häufiger muss umkodiert werden und es steigt die Verzögerungszeit.
  - Es genügt, wenn ein einziger Mix tatsächlich vertrauenswürdig ist.
  - Lieber sorgfältig wenige Mixe auswählen.
  - Derzeit verwendet man wenigstens drei, besser fünf Mixe, aber das ist rein subjektiv.

> Mixe: Warum »Wiederholungen ignorieren«?

- Wehrt Replay-Angriff ab:
  - Angreifer speichert E-A-Zuordnungen eines Schubs, in dem er eine Nachricht enttarnen möchte.
  - Sendet in späterem Schub die Nachricht erneut an den Mix.
    - Folge ohne »Wiederholungen ignorieren«:
      - Mix entschlüsselt Nachricht, wirft Zufallszahl weg und gibt Nachricht in gleicher Kodierung wie beim 1. Mal aus
      - Vergleich mit allen A-Nachrichten des ersten Schubs ermöglicht E-A-Verkettung für betreffende Nachricht

- Abwehr:
  - Mix prüft, ob eine E-Nachricht bereits gemixt wurde
  - erfordert Speichern *aller* E-Nachrichten (bzw. eines Fingerprints *aller* Nachrichten) oder Mitgabe eines Zeitstempels, solange Mix sein Schlüsselpaar nicht wechselt

> Warum nicht einfach Verbindungsverschlüsselung?



Verbindungsverschlüsselung zwischen den Mixen hilft gegen Außenstehende, aber nicht gegen Betreiber der Mixe.

> Mixe: Warum asymmetrische Kryptographie?

- Mixe lassen sich nur mit asymmetrischem Konzelationssystem realisieren.
  - Ausnahme: Zwischen Sender und erstem Mix genügt symmetrische Verschlüsselung.

- Bei symmetrisch verschlüsselter Nachricht an mittleren Mix:
  - Ein mittlerer Mix darf nicht erfahren, von welchem Sender eine Nachricht stammt. Symmetrische Schlüssel werden jedoch paarweise (Benutzer–Mix) ausgetauscht. Wenn der Mix entschlüsseln können soll, muss er jedoch den passenden Schlüssel kennen. Also kennt er den Absender.

- Asymmetrisches Konzelationssystem:
  - Benötigte n:1-Eigenschaft: Jeder kennt den Schlüssel und kann eine Nachricht für Mix verschlüsseln.

> Warum »...von genügend vielen Absendern?«

Eine Nachricht ist unbeobachtbar in der Gruppe der Nachrichten, die nicht vom Angreifer stammen.

- Ziel: eine bestimmte Nachricht enttarnen

Blocking/Flooding-Attack

- Angriff 1: Angreifer in Rolle »Teilnehmer«
  - Der Angreifer flutet den Mix mit eigenen Nachrichten, so dass im Idealfall die eine zu beobachtende Nachricht und n-1 vom Angreifer bearbeitet werden. (n ist die Schubgröße)
- Angriff 2: Mix greift an
  - Angreifender Mix blockiert die n-1 uninteressanten Nachrichten und ersetzt diese durch eigene
- Beobachtung des Outputs des folgenden Mixes führt zur Überbrückung des Mixes, da n-1 Nachrichten dem Angreifer bekannt sind.

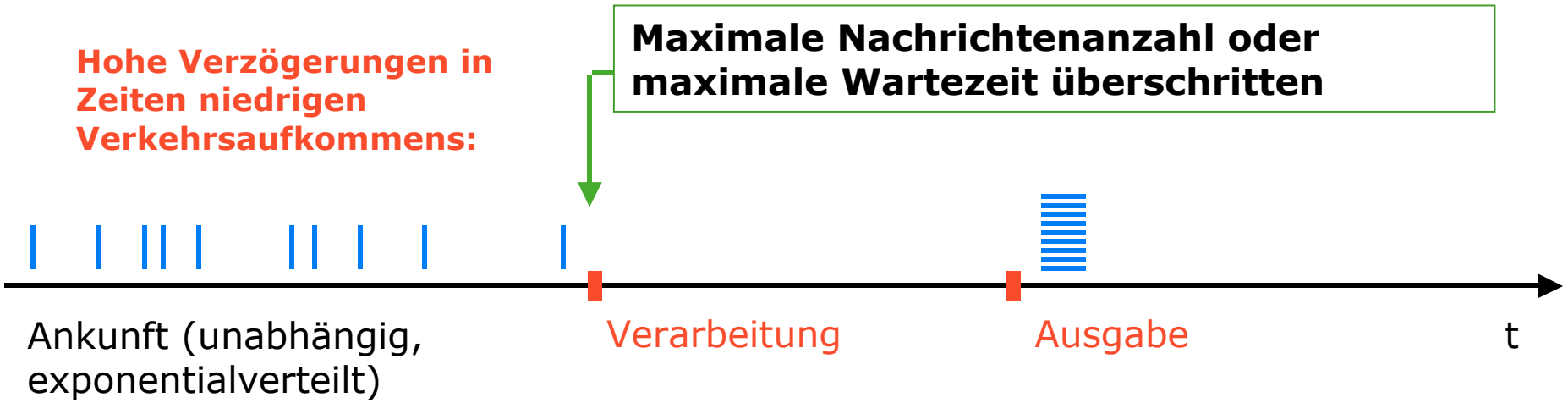
>> Warum »...von genügend vielen Absendern?«

Verhindern der Blocking/Flooding-Attack

- Verhindern Angriff 1: Angreifer in Rolle »Teilnehmer«
  - Jeder Teilnehmer darf nur begrenzte Zahl Eingabenachrichten liefern.
  - Authentisierung der Eingabenachricht gegenüber erstem Mix
- Verhindern Angriff 2: Mix greift an
  - Broadcast-Feedback: Alle Teilnehmer erhalten alle Ausgabeschübe aller Mixe zur Kontrolle. Wenn die eigene Nachricht fehlt, wird Alarm geschlagen.
  - Ticket-Methode: Jeder Teilnehmer erhält für jeden Mix ein »Ticket« (blind geleistete Signatur). Durch organisatorische Maßnahmen wird sichergestellt, dass jeder Teilnehmer je Schub genau 1 Ticket erhält, also den Mix nicht mehr *fluten* kann.
  - Hashwert-Methode: Verhindert, dass Nachrichten aus dem Schub *entfernt* werden, indem Hashwerte der Eingabenachrichten bitweise überlagert (XOR-verknüpft) und mit einem Referenzwert verglichen werden.

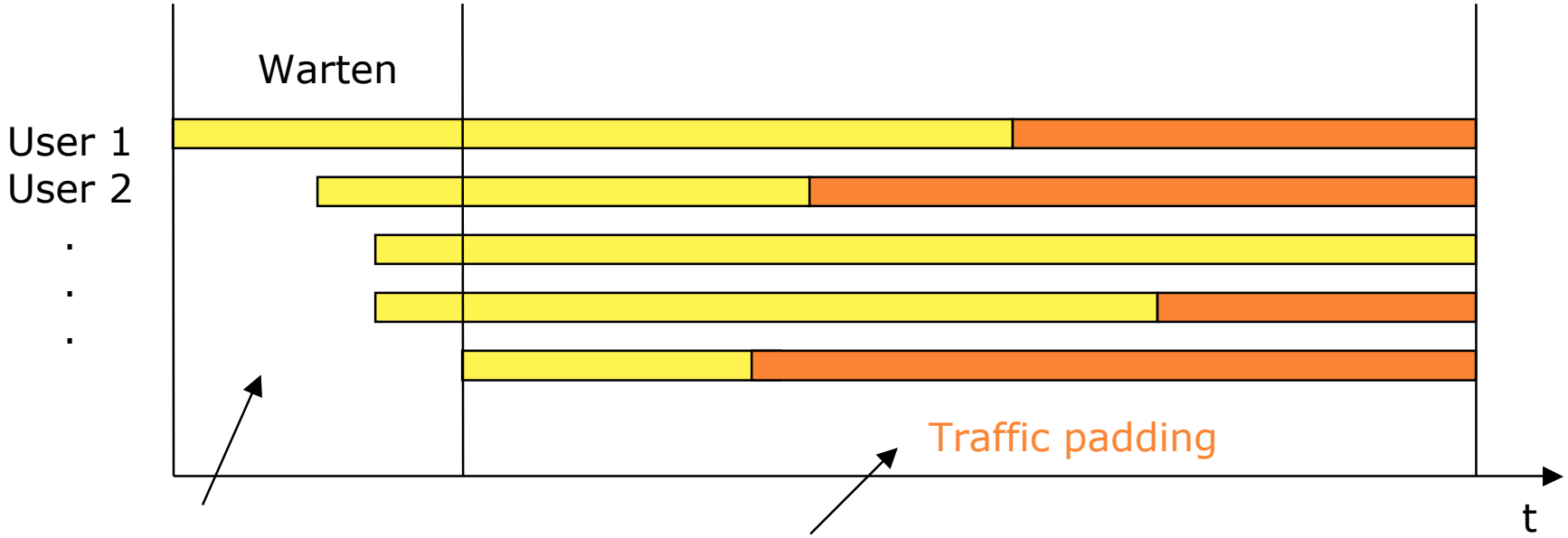
> Echtzeitkommunikation und Mixe

- Mixe sind gut geeignet für wenig zeitkritische Dienste:
  - E-Mail
- Für Echtzeitkommunikation (http, ftp) sind Modifikationen nötig:
  - Nachrichten sammeln führt zu starken Verzögerungen, da der Mix die meiste Zeit auf andere Nachrichten wartet
  - Nachrichtenlängen und Kommunikationsdauer variieren bei verbindungsorientierten Diensten stark
- Veränderungen nötig



> Traffic padding

- Ziel: Verbergen, wann eine Kommunikation beginnt und endet
- Problem: Niemand weiß, wann der letzte Nutzer seine Kommunikation beenden möchte



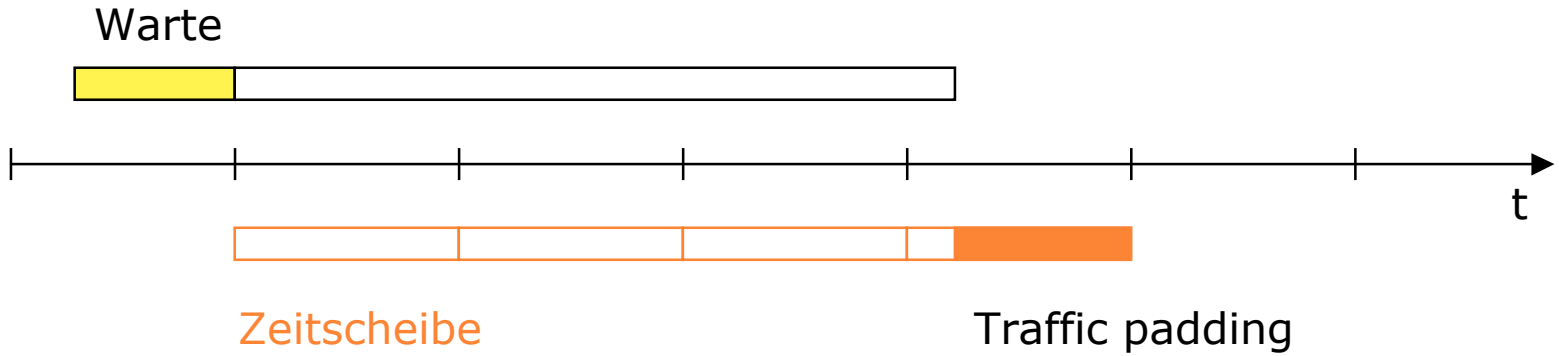
1. Warten, bis genügend Benutzer kommunizieren wollen (Bilden der Anonymitätsgruppe)
   Beispiel: 5 Nutzer

2. Nach Kommunikationsende senden die Nutzer solange Zufallszahlen, bis der letzte Nutzer seine Kommunikation beendet.

3. Problem: Niemand weiß, wann der letzte Nutzer seine Kommunikation beenden möchte, da niemand echte Nachrichten von Traffic padding unterscheiden kann.

> Zeitscheiben und Traffic padding

- Lösung: Zerlegen der Kommunikation in kleine Scheiben, genannt Zeitscheiben oder Volumenscheiben
  - Unbeobachtbarkeit innerhalb der Gruppe aller Nachrichten einer Zeitscheibe
  - Längere Kommunikationsverbindungen setzen sich aus mehreren Zeitscheiben zusammen
  - Zeitscheiben sind nicht verkettbar für Angreifer



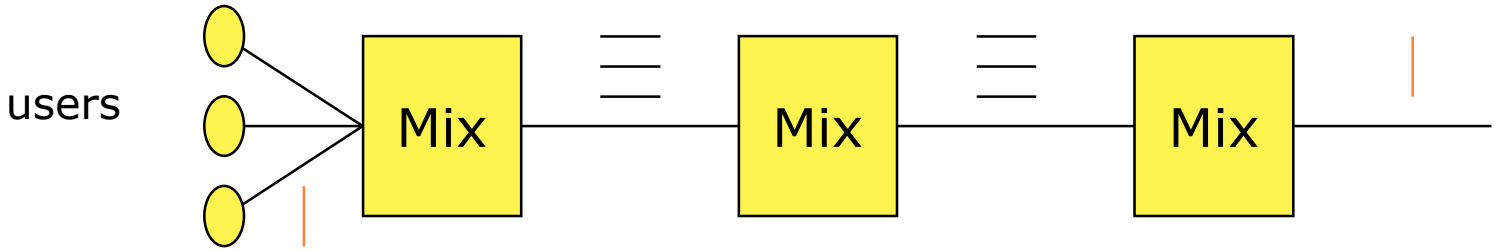
> Dummy traffic

- Ziel: Verkehrsaufkommen in Situationen niedrigen Verkehrs künstlich erhöhen, um Anonymitätsgruppe zu vergrößern

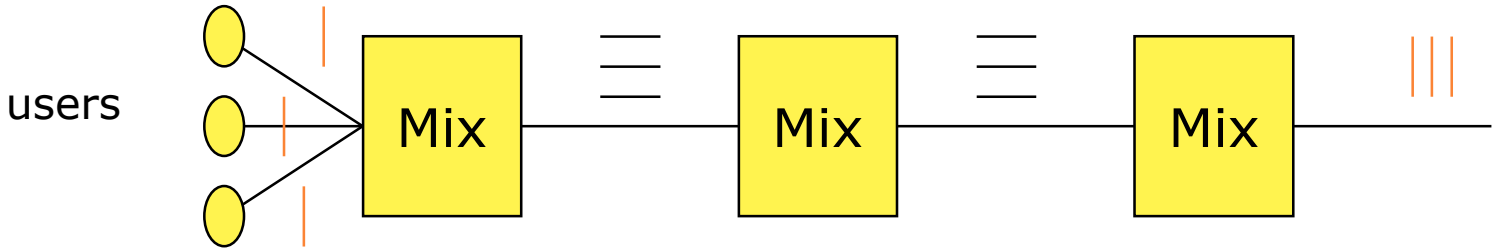
- Manchmal wird der Schub nicht voll, weil zu wenige Teilnehmer Nachrichten senden möchten
- Auswege:
  - Warten, bis weitere Nachrichten eintreffen (führt zu weiteren Verzögerungen)
  - Akzeptieren, dass Anonymitätsgruppe klein bleibt
  - Nutzer, die nichts zu senden haben, senden bedeutungslose Nachrichten
- **Def.: Dummy traffic**. Ein Nutzer sendet ständig Daten. Wenn er keine (verschlüsselten) Nachrichten zu senden hat, sendet er Zufallszahlen, die nicht unterscheidbar sind von echten verschlüsselten Nachrichten.

>> Dummy traffic

⌘ Dummy traffic nur zwischen Mixen reicht nicht aus



⌘ Dummy traffic muss Ende-zu-Ende generiert werden



> Probleme bei Langzeitbeobachtung

- Schnittmengenangriff: Vom Angreifer beobachtete Online-Offline-Phasen können vom Angreifer genutzt werden, um Schnittmengen von Anonymitätsgruppen zu bilden.
- Beispiel:
  - Ein Nutzer zeigt ein sehr konstantes Online-Offline-Verhalten (z.B. online von 20 - 22 Uhr täglich)
  - In dieser Zeit ruft er regelmäßig bestimmte Inhalte ab (Webseiten, seinen E-Mail-Account)
  - Eine große Anzahl weiterer Internetnutzer ist ebenfalls zu dieser Zeit aktiv.
- Der Angreifer beobachtet alle Kommunikationsleitungen, ohne in Mixe einzudringen. Langzeitbeobachtung führt zu unterschiedlichen Anonymitätsgruppen während der relevanten Zeit, die geschnitten werden.
- Wie lange es dauert, die Aktionen eines Benutzers zu verketten, hängt von der Gruppengröße und dem Benutzerverhalten ab.
- Simulation des Schnittmengenangriffs (Berthold, 1999)
- Derzeit ist kein Schutz vor Schnittmengenangriffen in Sicht.

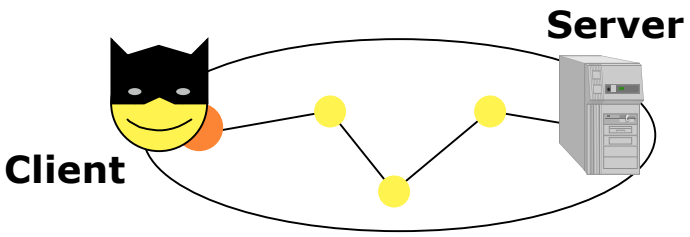
> Praxis: Anonymisierung im Internet

- Anonymisierung von Electronic-Mail:
  - Typ-0-Remailer: anon.penet.fi
    - Header entfernen und anonym/pseudonym weiterleiten
    - Reply möglich, da der echte Absender gespeichert und durch ein Transaktionspseudonym ersetzt wurde
    - Verkettbarkeit über Länge und zeitliche Korrelation
  - Typ-1-Remailer: Cyperpunk-Remailer
    - wie Typ-0, zusätzlich Angabe über Verzögerungszeit bzw. Sendezeit, Kaskadierung
    - PGP-verschlüsselte Mails werden vom Remailer entschlüsselt
    - Verkettbarkeit über Länge, bei niedrigem Verkehrsaufkommen auch über zeitliche Korrelation
  - Typ-2-Remailer: Mixmaster (Cottrel, 1995)
    - sammeln von Nachrichten
    - Mix-Modell im Pool-Mode
    - alle Nachrichten haben gleiche Länge

WORLD WIDE WEB

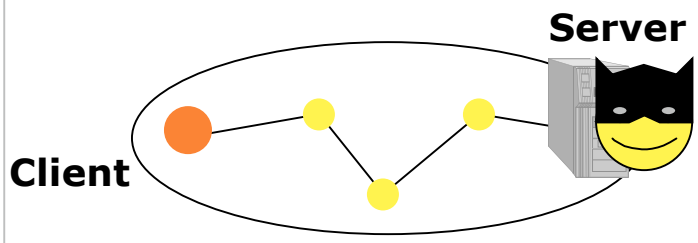
Client-Anonymität

Anonymes Abrufen fremder Inhalte



Server-Anonymität

Anonymes Publizieren von Inhalten



> Server-Anonymität

⌘ Ansätze für Server-Anonymität

- Rewebber Network und TAZ (Temporary Autonomous Zone) Servers (UC Berkeley, Ian Goldberg, Dave Wagner)

- Rewebber.com vormals »Janus« (FernUni Hagen, Andreas Rieke, Thomas Demuth)

```
http://rewebber.com/!RjVi0rawjGR
7Qv3FJlRbyej_Wh10g_9vpPAyeHmrYE1
a4UYt3IaqeQRXXd7oxEvwR8wJ3cnrNbP
IWUtlgW0uRlL0bGkAv3fX8WEcBd1JPWG
PL7OvuV0xtbMPsRbQg0iY=RKLBaUYeds
WTE1nuoh_J5J_yg1CfkaN9jSGkdf51-g
Vyxfupgc8VPsSyFdEeR0dj9kMHuPvLiv
f8mc44QBN0fMVJjpeyHSa79KdTQ5EGlP
cNLSD7YLSc1gKI3X8nk15s=RXbQaqm0A
_MMJaz9wchn_pI48xhTzgndt5Hk09VTo
XKPD7YbKVyiDZytva_sUBcdqpmPXTzAp
o1Pu1Rky8CxRfnC9BvQqof853n99vkuG
6i8DOat-NrOIndpz5xgwZKc=/
```

⌘ Funktionsprinzip:

- Verschlüsselte URL wird von einem zwischengeschalteten Proxy entschlüsselt und aufgerufen.

⌘ Anwendung/Motivation:

- Verhindern von Zensur
- Ermöglichen von *free speech*

> Anonymisierung von Verbindungen (HTTP, FTP)

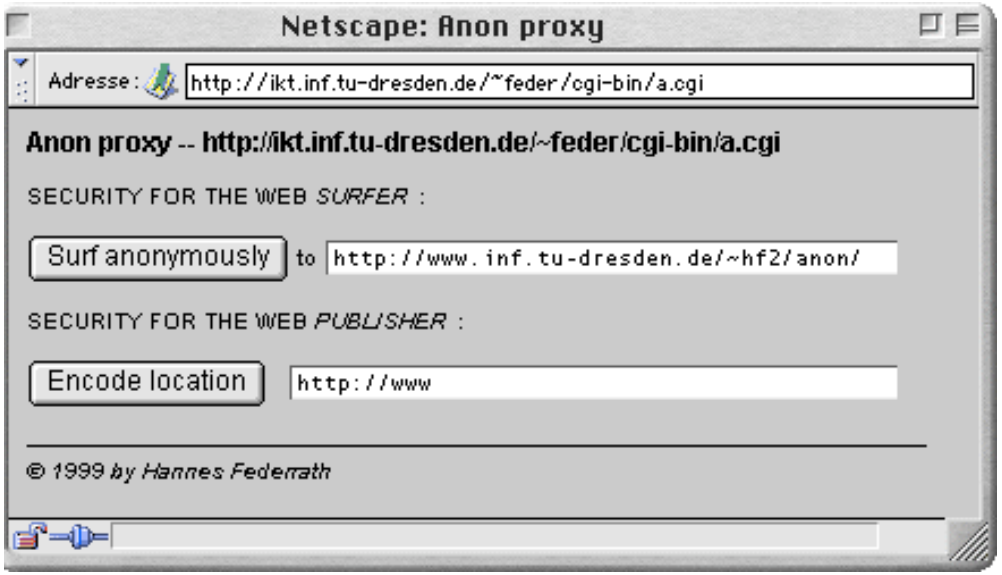
⌘ Client-Anonymität

- Einfache Proxies (teilweise mit Filterfunktion: Cookies, JavaScript, active content)
  - Anonymizer.com (Lance Cottrel)
  - Aixs.net
  - ProxyMate.com (Lucent Personal Web Assistant, Bell Labs)
  - Rewebber.com (Andreas Rieke, Thomas Demuth, FernUni Hagen)
  - Jeder entsprechend konfigurierte Web-Proxy

- Verkehrsanalysen berücksichtigende Verfahren
  - Onion-Routing (Naval Research Center)
  - Crowds (Mike Reiter, Avi Rubin AT&T)
  - Freedom (Ian Goldberg, Zero-Knowledge Inc.)
  - WebIncognito (Privada)
  - Web-Mixe/JAP (TU Dresden)

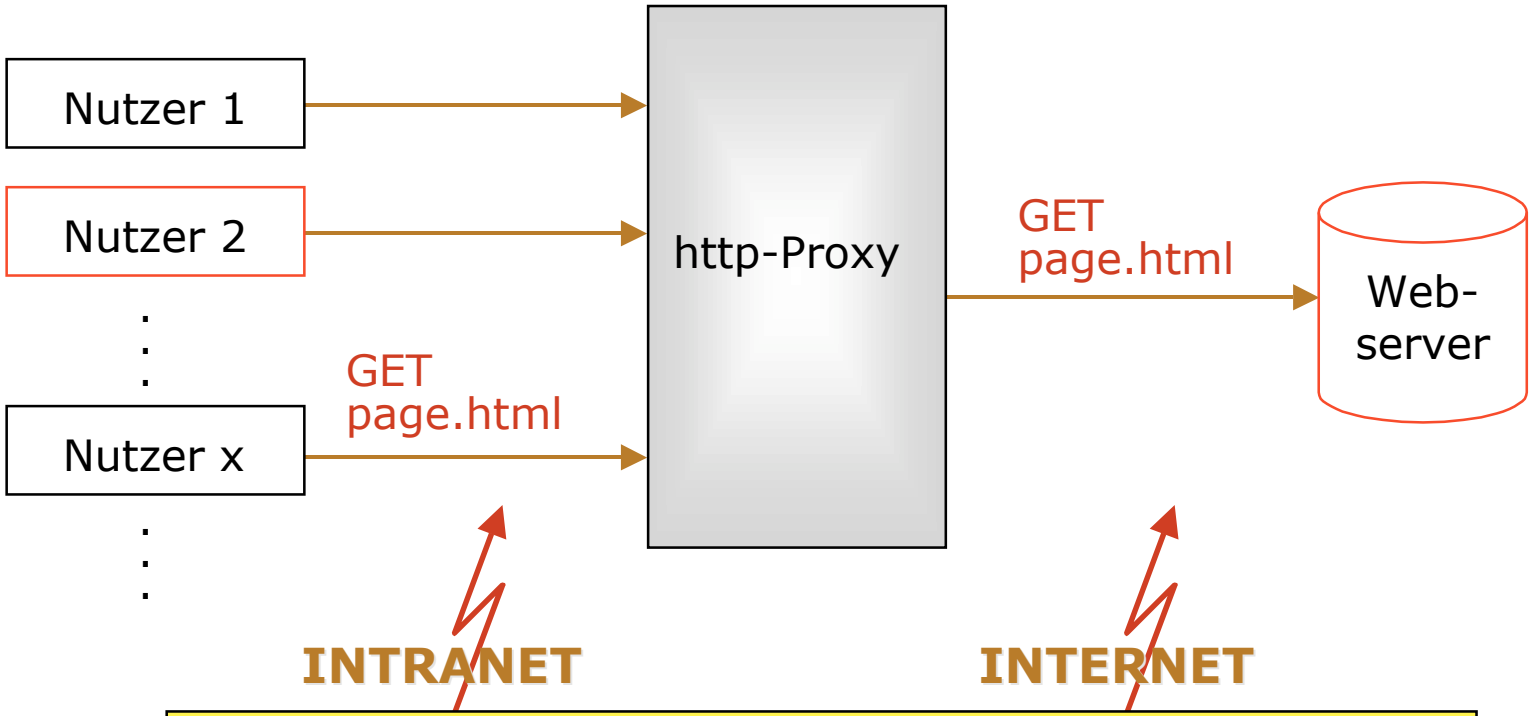
> Einfache Proxies

- Server besitzt keinerlei Information über den wirklichen Absender eines Requests
- **Kein Schutz gegen den Betreiber des Proxy**
- **Kein Schutz gegen Verkehrsanalysen**



- **Arbeitsprinzipien für Webzugriff:**
  1. **Formularbasiert**
     - URL eingeben
     - Proxy stellt Anfrage und versieht eingebettete URLs mit einem Präfix
  2. **Browserkonfiguration ändern**
     - »use proxy«

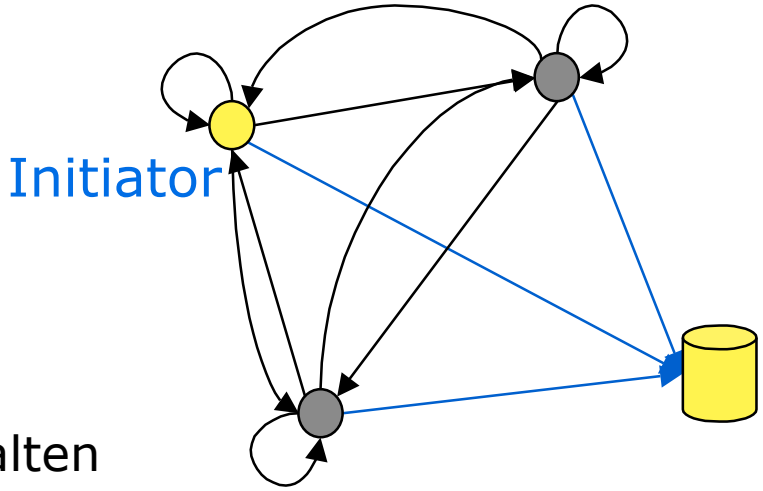
>> Einfache Proxies



Beobachtung und Verkettung ist möglich
- zeitliche Verkettung
- Verkettung über Inhalte (Aussehen, Länge)

Verschlüsselung zwischen Browser und Proxy verhindert Korrelation über »Aussehen«, aber nicht über Nachrichtenlänge und Zeit und hilft nichts gegen den Proxy.

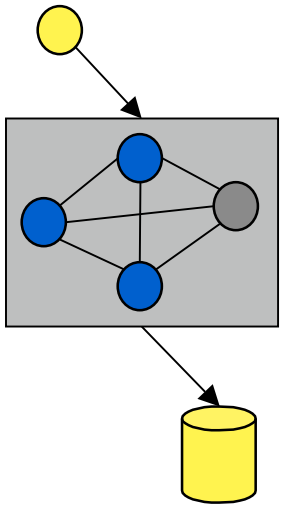
- **Webanfrage wird mit einer Wahrscheinlichkeit P direkt an Server geschickt oder alternativ (mit 1-P) an anderen Teilnehmer (Jondo)**
- **Symmetrische Verbindungsverschlüsselung** zwischen den Nutzern
  - Verkettung über Kodierung verhindern
  - Jedoch zeitliche Verkettung möglich
- Eingebettete Objekte (**Images** etc.) werden vom letzen Jondo angefordert.
  - Anfrage-Bursts unterbinden
- **Sicherheitseigenschaft:**
  Nutzer kann stets behaupten, sein Jondo habe die Anfrage zur Weiterleitung erhalten
- **Schwächen:**
  - zeitliche Korrelationen bleiben erhalten
  - Jondos können mitlesen (problematisch bei personalisierten Sites)

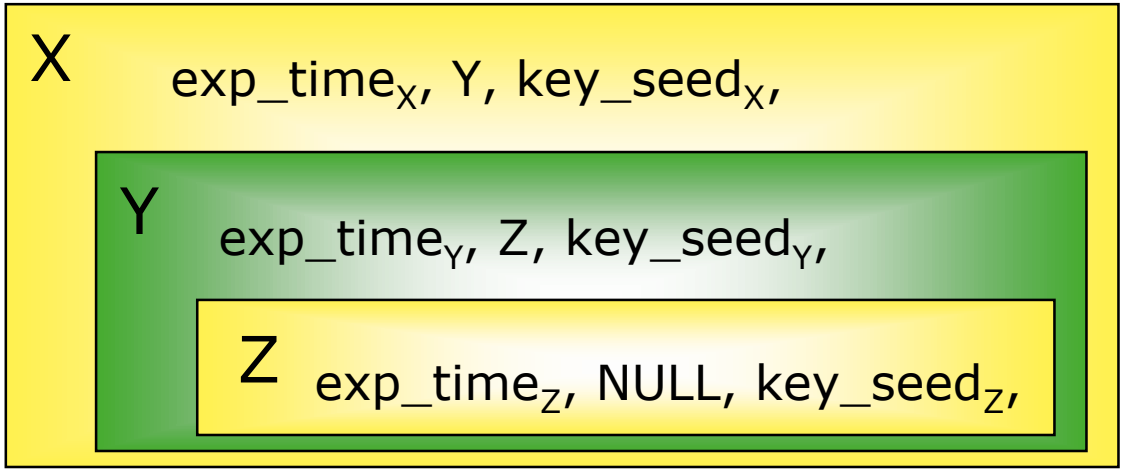


> Onion Routing

US Naval Research Center, Syverson et. al, 1996

- Ziel des Projektes: Schutz von Routing-Information in verbindungsorientierten Diensten
- Mix-basiert mit Verfallszeit für die Datenbank
- Dummy Traffic zwischen Mixen (Onion-Routern)
- **First-Hop-Last-Hop-Attack:** Verkettung von Anfangs- und Endpunkt einer Kommunikation (bei geringer Verkehrslast) möglich
  - Zetliche Verkettung
  - Nachrichtenlänge





Jede »Schale« ist mit dem public key des nächsten Onion-Routers verschlüsselt

> Andere Konzepte

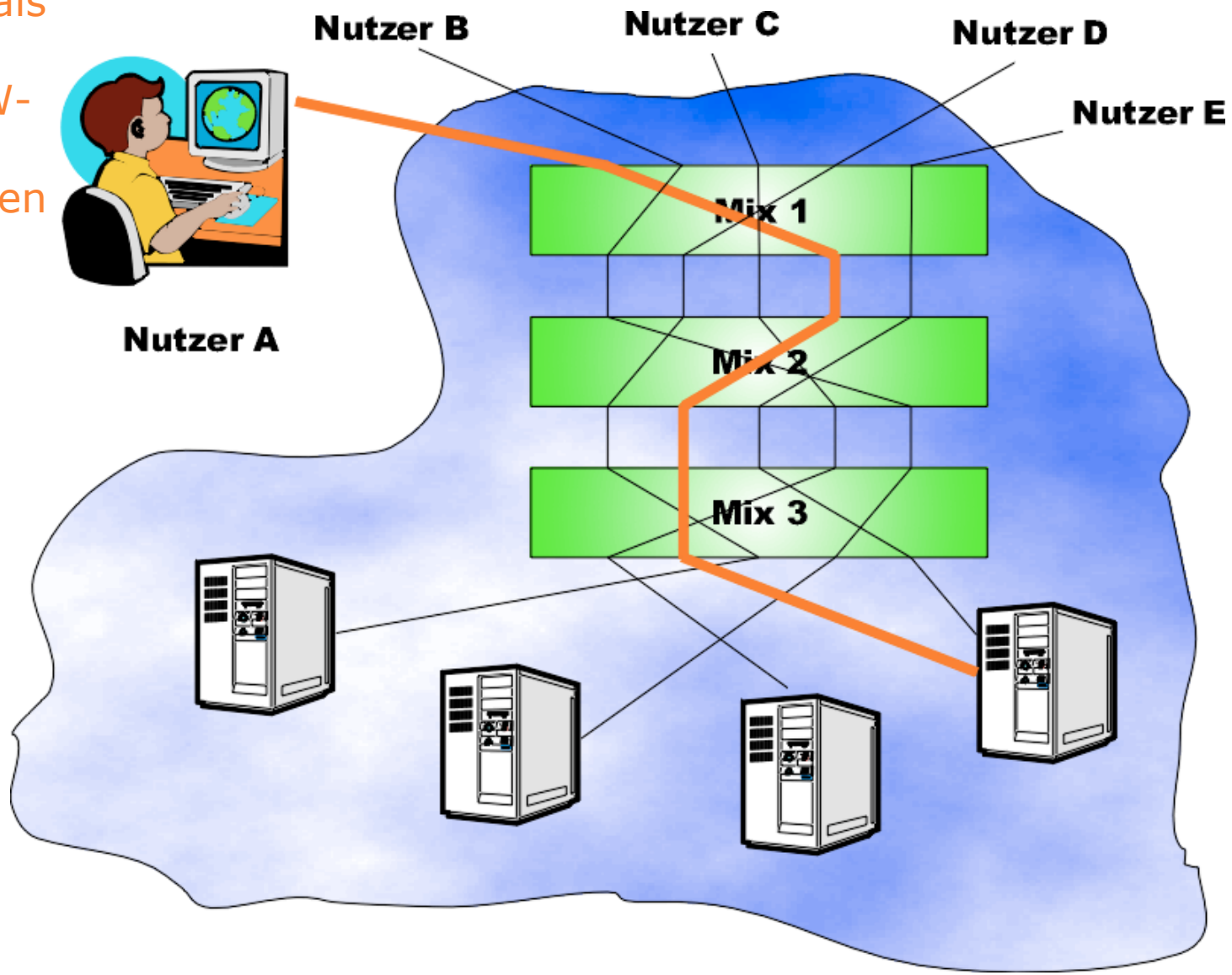
- Privada WebIncognito
  - keinerlei Angaben zum technischen Konzept
  - »is based on a patent-pending technology«

- Zero-Knowledge Freedom (im Herbst 2001 abgeschaltet)
  - Mix-basiert (Anonymous Internet Proxy, AIP)
  - Dummy Traffic zwischen Mixen
  - Link Padding Envelope Shaper
    - Berücksichtigung des Nutzerverhaltens und der Verkehrssituation bei der Generierung von Dummy Traffic

- WebMixe/JAP
  - Mix-basiert
  - Dummy-Traffic von Teilnehmer
  - Rückmeldung über Größe der Anonymitätsgruppe
  - geplant: Identitätsmanagement

JAP/WebMixe

- JAP wird als Proxy für den WWW-Browser eingetragen



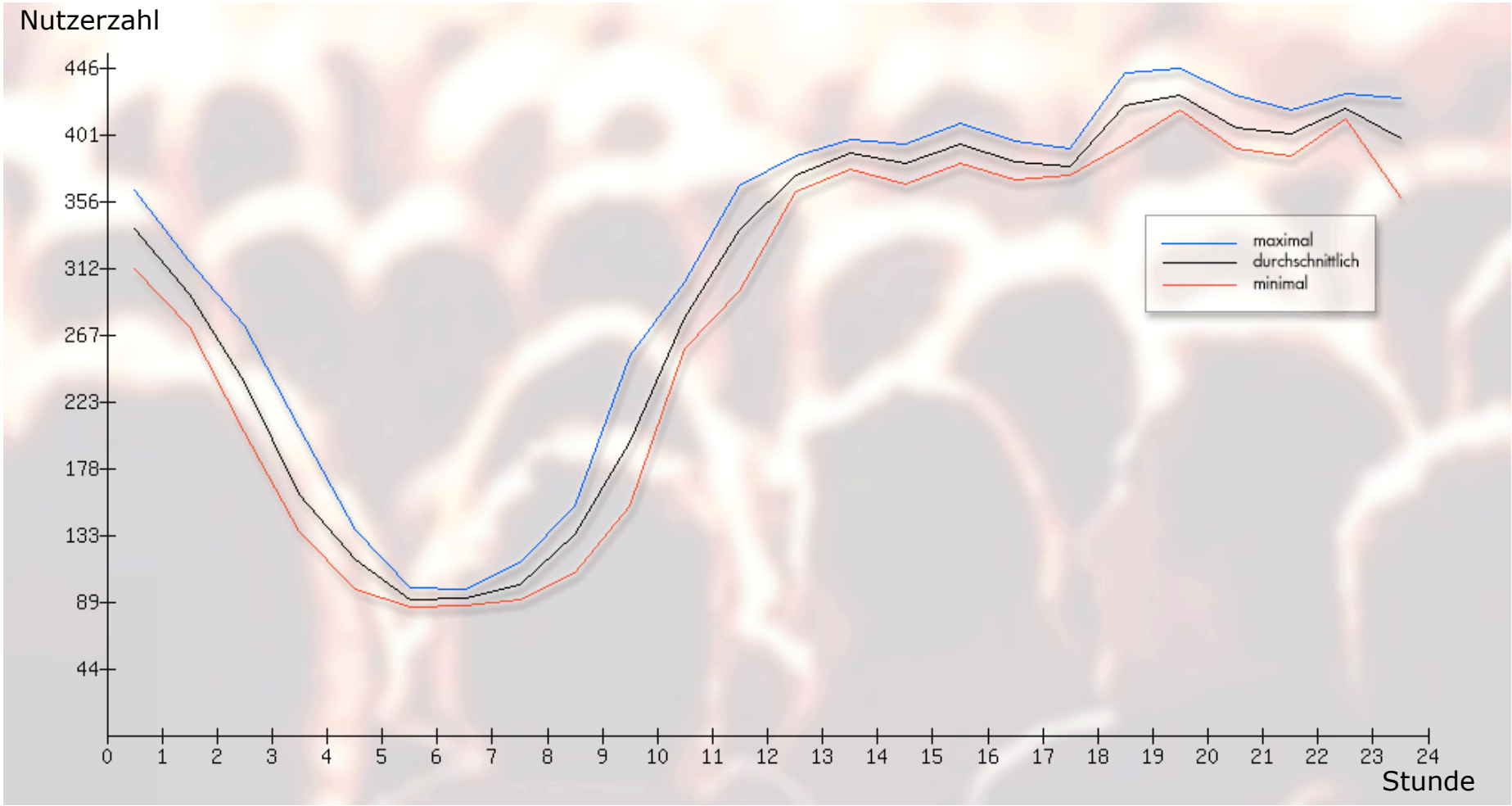
Technische Daten, Nutzerzahlen

JAP.inf.tu-dresden.de

- Entwicklung eines praktisch nutzbaren Systems zum unbeobachtbaren Surfen im Internet
  - Schutz von personenbezogenen Daten bei der Benutzung des Internet
  - Verhinderung von »Profiling« und kommerzieller Nutzung
- Implementierung bestehend aus:
  - Java Client Programm »JAP«
  - Mix-Server (C++)
  - Info-Service (Java)
- Schätzung:
  - insgesamt ca. 18000 Nutzer
- Netzwerkverkehr ist zur Zeit der Hauptengpass:
  - ca. 1000 Gigabyte pro Monat
  - bei bis zu 650 Nutzern gleichzeitig online
  - zu Spitzenzeiten etwa 2000 Transaktionen (URLs) pro Minute
- 3 Mix-Kaskaden im Betrieb

Nutzung

⌘ Typischer Verlauf der Nutzerzahl eines Tages



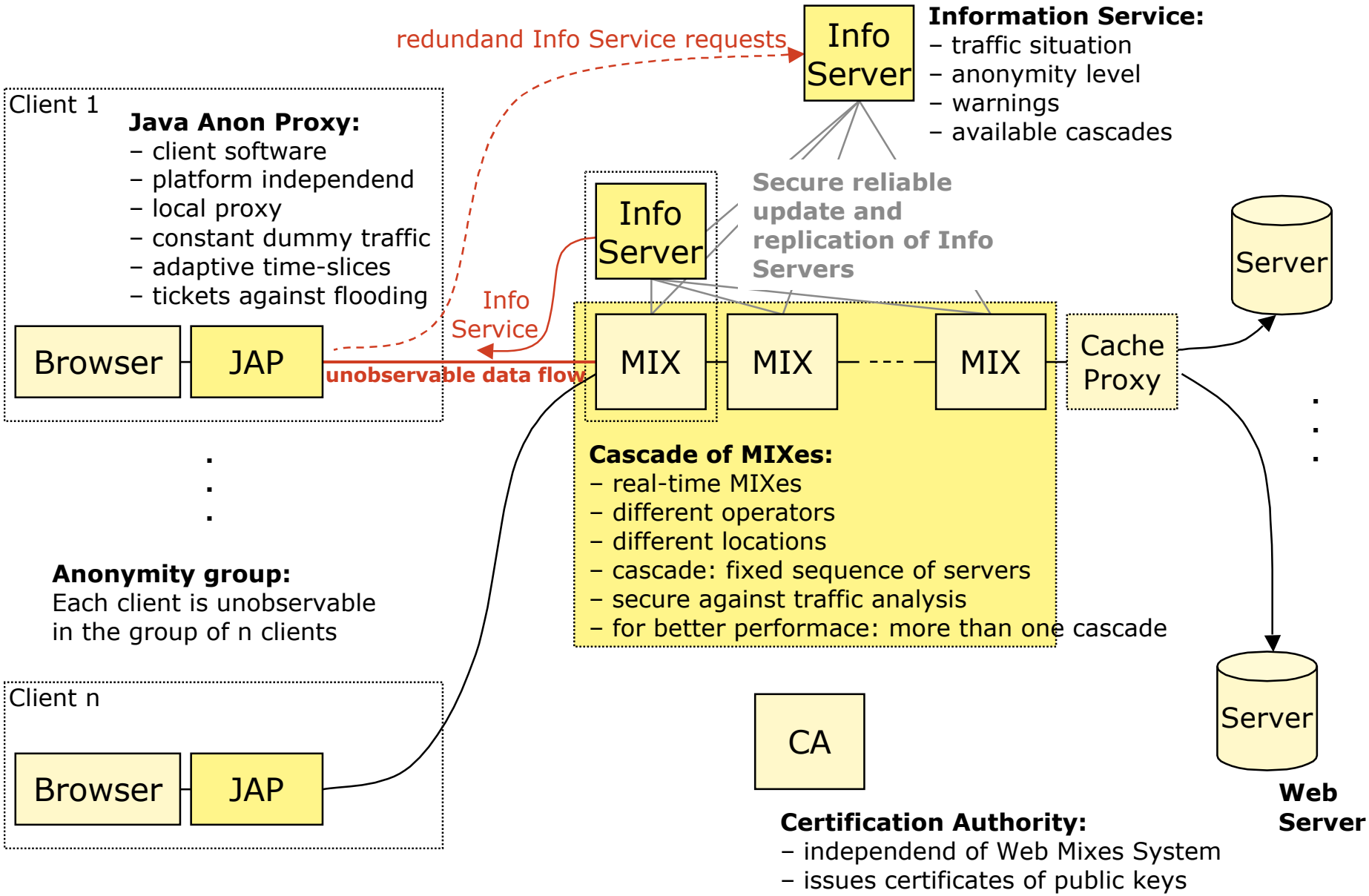
Positive Erfahrungen

- Vorstellung auf der CeBit 2001 und 2002
  - Im Gegensatz zu 1997 wird heute nicht mehr gefragt, wogegen man sich eigentlich schützen soll.
- Größeres Interesse am Datenschutz und im Bewusstsein um Bedrohungen
  - Hohe Bereitschaft praktikable Lösungen zum Selbstdatenschutz einzusetzen
- Kommerzielles Interesse
  - Vermarktung als Dienstleistung geplant

Negative Erfahrungen

- Sehr schwer vermittelbar, warum ein System sicher bzw. unsicher ist
  - Verbreitete Vorstellung: ständig wechselnde IP-Adresse = hohe Anonymität
- Missbrauchsfälle aufgetreten
  - Dienst zur Zeit auf Web-Zugriffe beschränkt, obwohl allgemeiner anonymer TCP/IP möglich wäre
  - Nach juristischer Prüfung ist der Dienst legal, jedoch Überlegungen zur Deanonymisierung
  - Neue Forschungsfrage: Wie kann begründete Deanonymisierung ohne Massenüberwachung durchgeführt werden?
- Länder (Saudi Arabien) haben Zugang zum Dienst gesperrt
  - Forschungsfrage: Anonymisieren des Anonymisierungsdienstes

> WebMixe: Architektur



JAP Website

JAP Anonymity & Privacy

ANONYMITY IS NOT A CRIME

Kostenloser Download von JAP

http://jap.inf.tu-dresden.de

Weitere Informationen zur Anonymität im Internet

http://www.inf.tu-dresden.de/~hf2/anon/