Anonymous & Unobservable Communication

https://dud.inf.tu-dresden.de/sac2

Stefan Köpsell
(Slides [mainly] created by Andreas Pfitzmann)

Technische Universität Dresden, Faculty of Computer Science, D-01187 Dresden
Nöthnitzer Str. 46, Room 3062
Phone: +49 351 463-38272, e-mail: stefan.koepsell@tu-dresden.de, https://dud.inf.tu-dresden.de/
<table>
<thead>
<tr>
<th>Lectures</th>
<th>Staff</th>
<th>SWS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security and Cryptography I, II</td>
<td>Strufe, Köpsell</td>
<td>2/2</td>
</tr>
<tr>
<td>Resilient Networking</td>
<td>Strufe</td>
<td>2/2</td>
</tr>
<tr>
<td>Cryptography and -analysis</td>
<td>Franz</td>
<td>2/1</td>
</tr>
<tr>
<td>Data Security</td>
<td>Franz</td>
<td>2/1</td>
</tr>
<tr>
<td>Information &amp; Coding Theory</td>
<td>Schönfeld</td>
<td>2/1</td>
</tr>
<tr>
<td>Channel Coding</td>
<td>Schönfeld</td>
<td>2/2</td>
</tr>
<tr>
<td>Data Security and Cryptography</td>
<td>Köpsell</td>
<td>0/4</td>
</tr>
<tr>
<td>Security Lab</td>
<td>Köpsell</td>
<td>2/2</td>
</tr>
<tr>
<td><strong>Seminar: Privacy in Online Social Networks</strong></td>
<td>Strufe</td>
<td></td>
</tr>
<tr>
<td>Seminar: Privacy and Security</td>
<td>Köpsell et.al.</td>
<td>2/0</td>
</tr>
<tr>
<td>Seminar: Secure app. development</td>
<td>Borcea-Pfitzmann</td>
<td>2</td>
</tr>
<tr>
<td>Seminar: Security in Computer Systems</td>
<td>Köpsell</td>
<td>2</td>
</tr>
<tr>
<td>Introduction to Data Protection Law</td>
<td>Wagner</td>
<td>2/0</td>
</tr>
</tbody>
</table>
Principles of PETs

- Privacy-enhancing Technologies (PETs)
  - Information suppression tools (Opacity tools)
  - Transparency-enhancing tools (TETs)

- Opacity Tools:
  - Anonymization, pseudonymization, obfuscation

- Transparency-enhancing Tools:
  - Informing user about data collection, purpose etc.
  - Informing about impact of data collection (needed for “informed consent“)
  - Enables checks whether data collection is conform to legal regulation
  - Various techniques: Secure Logging, Audits, Quality Seals, Policies etc.
Transparency-enhancing Tool
Confidentiality ensures that nobody apart from the communicants can discover the content of the communication.

Hiding ensures the confidentiality of the transfer of confidential user data. This means that nobody apart from the communicants can discover the existence of confidential communication.

Anonymity ensures that a user can use a resource or service without disclosing his/her identity. Not even the communicants can discover the identity of each other.

Unobservability ensures that a user can use a resource or service without others being able to observe that the resource or service is being used. Parties not involved in the communication can observe neither the sending nor the receiving of messages.

Integrity ensures that modifications of communicated content (including the sender’s name, if one is provided) are detected by the recipient(s).

Accountability ensures that sender and recipients of information cannot successfully deny having sent or received the information. This means that communication takes place in a provable way.

Availability ensures that communicated messages are available when the user wants to use them.

Reachability ensures that a peer entity (user, machine, etc.) either can or cannot be contacted depending on user interests.

Legal enforceability ensures that a user can be held liable to fulfill his/her legal responsibilities within a reasonable period of time.
Anonymity:
- is the state of being not identifiable within a set of subjects, the *anonymity set*.
- is the stronger, the larger the respective anonymity set is and the more evenly distributed the sending or receiving, respectively, of the subjects within that set is.

⇒ *Anonymity* within a particular setting depends on the number of users
• Unlinkability:
  – of two or more items of interest (IOIs, e.g., subjects, messages, actions, ...) from an attacker’s perspective means that within the system, the attacker cannot sufficiently distinguish whether these IOIs are related or not.

\[ \Rightarrow \textbf{Anonymity in terms of Unlinkability:} \]
Unlinkability between an identity (subject) and the IOI in question (message, data record etc.)
Correlations between protection goals

Confidentiality → Hiding +

Anonymity → Unobservability +

Integrity → Accountability +

Reachability → Legal Enforceability -

Availability → Confidentiality +

implies strengths weaken
Observability of users in switched networks

countermeasure encryption
- link encryption

radio

television

videophone

phone

internet

network termination

interceptor

possible attackers

telephone exchange
- operator
- manufacturer (Trojan horse)
- employee
countermeasure encryption

- end-to-end encryption

Observability of users in switched networks

radio

television

videophone

phone

internet

network termination

interceptor

possible attackers

telephone exchange

- operator
- manufacturer (Trojan horse)
- employee
countermeasure encryption
- link encryption
- end-to-end encryption

**Problem:** traffic data
who with whom?
when? how long?
how much information?

**Aim:** “protect” traffic data (and so data on interests, too) so that they couldn’t be captured.
Reality or fiction?

Since about 1990 reality

Video-8 tape 5 Gbyte

= 3 * all census data of 1987 in Germany
memory costs < 25 EUR

100 Video-8 tapes (or in 2016: 1 hard drive disk with 500 GByte for ≈ 35 EUR) store
all telephone calls of one year:

Who with whom ?
When ?
How long ?
From where ?
With the development of television, and the technical advance which made it possible to receive and transmit simultaneously on the same instrument, private life came to an end.

George Orwell, 1948
Examples of changes w.r.t. anonymity and privacy

Broadcast allows recipient anonymity — it is not detectable who is interested in which programme and information
Examples of changes w.r.t. anonymity and privacy

Internet-Radio, IPTV, Video on Demand etc. support profiling
Remark: Plain old letter post has shown its dangers, but nobody demands full traceability of them …
The massmedia „newspaper“ will be personalised by means of Web, elektronic paper and print on demand.
Privacy & the Cloud?

[http://www.apple.com/icloud/]

[AP] Anonymity & Privacy
ANONYMITY IS NOT A CRIME

[http://www.apple.com/icloud/]
Smart Home
Smart Car
Smart Watch
Smart TV
Smart ...
Types of Data

• Data without any *relation* to *individuals*
  – Simulation data
  – Measurements from experiments

• Data *with relation* to *individuals*
  – Types
    – Content
    – Meta data
  – Revelation
    – Consciously
    – Unconsciously
Notions of Privacy: Right to be let alone


• Reason: “snapshot photography” (recent innovation at that time)
  – allowed newspapers to publish photographs of individuals without obtaining their consent.
  – private individuals were being continually injured
  – this practice weakened the “moral standards of society as a whole”

• Consideration:
  – basic principle of common law: individual shall have full protection in person and in property
  – “it has been found necessary from time to time to define anew the exact nature and extent of such protection”
  – “Political, social, and economic changes entail the recognition of new rights”

• Conclusion:
  – “right to be let alone”
Notions of Privacy: Data Protection

- **Principles**
  - collect and process personal data **fairly and lawfully**
  - **purpose binding**
    - keep it only for one or more specified, explicit and lawful purposes
    - use and disclose it only in ways compatible with these purposes
  - **data minimization**
    - adequate, relevant and not excessive wrt. the purpose
    - retained no longer than necessary
  - **transparency**
    - inform who collects which data for which purposes
    - inform how the data is processed, stored, forwarded etc.
  - **user rights**
    - access to the data, correction, deletion
  - keep the data safe and secure
• Helen Nissenbaum: *Privacy as Contextual Integrity*, Washington Law Review, 2004

• close relation to data protection principles:
  – purpose binding

• Idea:
  – privacy violation, if:
    • violation of **Appropriateness**
      – the context „defines“ if revealing a given information is appropriate
      – **violation**: usage of information disclosed in one context in another context (even if first context is a “public” one)
    • violation of **Distribution**
      – the context „defines“ which information flows are appropriated
      – **violation**: inappropriate information flows
• exemplified with sender anonymity:
  – beyond suspicion: no more likely than any other potential sender
  – probable innocence: no more likely to be the sender than not to be the sender
  – possible innocence: there is a nontrivial probability that the real sender is someone else
Mechanisms to protect traffic data

Protection outside the network

Public terminals
- use is cumbersome

Temporally decoupled processing
- communications with real time properties

Local selection
- transmission performance of the network
- paying for services with fees

Protection inside the network
Questions:

- How widely distributed? (stations, lines)
- Observing / modifying?
- How much computing capacity? (computationally unrestricted, computationally restricted)
Realistic protection goals/attacker models: Technical solution possible?
Questions:

- How widely distributed? (stations, lines)
- Observing / modifying?
- How much computing capacity? (computationally unrestricted, computationally restricted)

Unobservability of an event E
For attacker holds for all his observations B: $0 < P(E|B) < 1$
perfect: $P(E) = P(E|B)$

Anonymity of an entity

Unlinkability of events

If necessary: partitioning in classes
Protection of the recipient: Broadcast

### Performance?
more capable transmission system

### Addressing
- **explicit addresses:** routing
- **implicit addresses:** attribute for the station of the addressee

<table>
<thead>
<tr>
<th>Address distribution</th>
<th>Public Address</th>
<th>Private Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>invisible</td>
<td>very costly, but necessary to establish contact</td>
<td>costly</td>
</tr>
<tr>
<td>visible</td>
<td>should not be used</td>
<td>change after use</td>
</tr>
</tbody>
</table>

- invisible <==> encryption system
  - example: pseudo random number (generator), associative memory to detect

A. Pfitzmann, M. Waidner 1985
BitMessage (J. Warren, 2012)

• messaging system based on
  – broadcast
  – implicit invisible private addresses
• python based clients at: bitmessage.org
• address: Hash(public encryption key, public signature test key)
• messages:
  – encrypted using Elliptic Curve Cryptography
  – digitally signed
  – additionally: proof of work
    ➔ Anti-SPAM
• broadcast of messages:
  – P2P-based overly structure
  – store-and-forward like
  – pull-based
Equivalence of Encryption Systems and Implicit Addressing

invisible public address  <=>  asymmetric encryption system

invisible private address  <=>  symmetric encryption system
Broadcast vs. Queries

Broadcast of separate messages to all recipients

Everybody can query all messages
Example for message service

5 servers available, all contain the same messages in equal order

server, which gets the long query vector, starts circulation

servers add responses, which are encrypted with (pseudo-) one-time pads

response of the message service:

\[
!x = \text{message 1} \oplus \text{message 4}
\]

\[
!y = \text{message 1} \oplus \text{message 2}
\]

\[
!z = \text{message 2} \oplus \text{message 3} \oplus \text{message 4}
\]

from this follows by local superposition of the pads

\[
!x \oplus !y \oplus !z \Rightarrow \text{message 3}
\]

(query vectors)

(query multiple memory cells)
User is interested in D[2]:

Index within Request-Vector = 1234
Set Vector = 0100
Chose random Vector (S1) = 1011
Chose random Vector (S2) = 0110
Calculate Vector (S3) = 1001

Calculations: XOR

Private Message Service
User is interested in D[2]:

Index within Request-Vector = 1234

Set Vector = 0100
Chose random Vector (S1) = 1011
Chose random Vector (S2) = 0110
Calculate Vector (S3) = 1001

Server calculates XOR of the requested records

Answer of S1: 0010110
S2: 1001000
S3: 0111000

Sum is D[2]: 1100110

Note: Encryption between Server and Client necessary!
Example for message service

- Message service
- 5 servers available, all contain the same messages in equal order
- 3 servers used for superposed querying
- 2 servers add responses, which are encrypted with (pseudo-) one-time pads
- Start circulation by local superposition of the pads

Response of the message service:
- $\neg x = \text{message 1 XOR message 4}$
- $\neg y = \text{message 1 XOR message 2}$
- $\neg z = \text{message 2 XOR message 3 XOR message 4}$

From this follows by local superposition of the pads
- $\neg x \lor \neg y \lor \neg z \Rightarrow \text{message 3}$ (equal to the sum of the wanted (**))
- Query multiple memory cells

David A. Cooper, Kenneth P. Birman 1995
Efficiency improvements: A. Pfitzmann 2001
“Query and superpose” instead of “broadcast”

re-writable memory cell = implicit address
re-writing = addition mod 2 (enables to read many cells in one step)
channels trivially realizable

Purposes of implicit addresses

Broadcast: Efficiency (evaluation of implicit address should be faster than processing the whole message)
Query and superpose: Medium Access Control; Efficiency (should reduce number of messages to be read)

fixed memory cell = visible implicit address
implementation: fixed query vectors for servers 0\rightarrow 1

Number of addresses linear in the expense (of superposing).

Improvement: Set of re-writable memory cells = implicit address
Message \( m \) is stored in a set of \( a \) memory cells by choosing \( a-1 \) values randomly and choosing the value of the \( a^{th} \) cell such that the sum of all \( a \) cells is \( m \).
For overall \( n \) memory cells, there are now \( 2^n-1 \) usable implicit addresses, but due to overlaps of them, they cannot be used independently.
If collisions occur due to overlap, try retransmit after randomly chosen time intervals.
Any set of cells as well as any set of sets of cells can be queried in one step.
hopping between memory cells = invisible implicit address

Idea: User who wants to use invisible implicit address at time $t$
reads the values from reserved memory cells at time $t-1$.
These values identify the memory cell to be used at time $t$.

Impl.: • Address owner gives each server $s$ a $PBG_s$.
• Each server $s$ replaces at each time step $t$ the content of its
 reserved memory cell $S_{Adr}$ with $PBG_s(t)$:

$$S_{Adr} := PBG_s(t)$$

• User queries via MIXes $\sum_s PBG_s(t)$.
user employs $S\sum_s PBG_s(t)$ for message. $\checkmark\checkmark 1$

• Address owner generates $\sum_s PBG_s(t)$ and reads using “query and superpose”

$$S\sum_s PBG_s(t)$$

before and after the writing of messages, calculates difference.

Improvement: for all his invisible implicit addresses together: $\checkmark\checkmark 2$ (if $\leq 1$ msg)

Address is in so far invisible, that at each point of time only a very little fraction of
all possible combinations of the cells $S_{Adr}$ are readable.
hopping between memory cells = invisible implicit address

can be extended to

hopping between *sets of* memory cells = invisible implicit address
Fault tolerance (and countering modifying attacks)

What if server (intentionally) does

1. not respond or

2. delivers wrong response?

1. Submit the same query vector to another server.

2. Messages should be authenticated so the user can check their integrity and thereby detect whether at least one server did deliver a wrong response. If so, use a disjoint set of servers or lay traps by sending the same query vector to many servers and checking their responses by comparison.
Protection of the sender

Dummy messages
  • don’t protect against addressee of meaningful messages
  • make the protection of the recipient more inefficient

Unobservability of neighboring lines and stations as well as digital signal regeneration
  example: RING-network
Digital signal regeneration:

The analogue characteristics of bits are independent of their true sender.

The idea of physical unobservability and digital signal regeneration can be adapted to other topologies, i.e. tree-shaped CATV networks;

It reappears in another context in Crowds, GNUnet, etc.
Crowds (Reiter, Rubin, 1998)

- Goal: Anonymous Web browsing
- Link-Encryption between two participants
- HTTP-requests /-responses in plain (no end-to-end encryption)
- each user makes random routing decision
User A

User B

User C

User D

User E

User F

User G

User H

① Request \( h(h(h(B))) \) for block B

② Forwarding of a request (original sender address is preserved)

③ Indirecting of a request (sender address will be rewritten)

④ Response to user according to the given sender address

Encrypted block \( B_{\text{enc}} = E_{h(B)}(B) \)
Buses…

• Amos Beimel, Shlomi Dolev: „Buses for Anonymous Message Delivery“, 2002
  – follow-up: Andreas Hirt, Michael J. Jacobson, Jr., Carey Williamson: “A practical buses protocol for anonymous internet communication.”, 2005

• basic ideas follow a city-bus metaphor
  – messages send around contain „seats“, i.e cells dedicated to certain users/messages
  – different protocols proposed: trade-of: communication complexity, time complexity, storage complexity
• Attacker model:
  – global observing outsider
  – observing participants (except sender/receiver!)
  – [modifying attackers are only considered wrt. availability]

• Protection goals achieved
  – sender anonymity
  – recipient anonymity
  – unobservability regarding sending/receiving of messages
Buses
Buses – simple solution

- dummy messages, if nothing to sent
- implicit addressing
- communication complexity: 1
- time complexity: $O(n)$
- storage complexity: $O(n^2)$
1. Idea: just one „seat“ per sender
   - one ring per sender, i.e. broadcast using implicit addresses

2. Idea: sender selects random „seat“
   - problem: replacement of message from other sender
   - birthday paradox
   - $s$ – number of messages sent simultaneously
   - $k$ – some security parameter
   - for bus size $b = k \cdot s^2 \rightarrow P(\text{collision}) \approx 1/k$
   - advantage: sender anonymity against recipient
   - crypto: layered (aka mix-based)
Buses – reduced seats – Example

- A wants to send some message \( m \) to \( D \)
- depicted is one seat of the bus

\[
\begin{align*}
A & \xrightarrow{k_B(k_C(k_D(m)))} B \\
& \xrightarrow{k_C(k_D(m))} C \\
& \xrightarrow{k_D(m)} D \\
& \xrightarrow{k_E^{-1}(\text{random})} E
\end{align*}
\]

- replay attacks!
Buses – reduced time complexity

- 2 buses per link
- messages are transferred from one bus to another according to the shortest path
- number of seats depends on the shortest paths from all senders to all receivers

- tradeoff: time vs. communication complexity
  -> spanning subgraph sufficient

4 seats → one per recipient of D
4 seats → one per sender of D
? seats → e.g. shortest path B to E not unique
• Idea: partition graph into clusters, have one bus per cluster
• achieves sender and recipient anonymity

• basic building blocks:
  – random walk through peer graph
    • simulates broadcast
  – invisible implicit addressing
  – dummy messages
  – strict synchronisation
    • mitigates timing attacks
The Drunk Motorcyclist Protocol for Anonymous Communication

Adaml L. Young, Moti Young, 2014

- dummy or real message

- store for decryption
- forward to random peer (--TTL)

- delete if TTL=0
Requirement
For each possible error, anonymity has to be guaranteed.

Problem
Anonymity: little global information
Fault tolerance: much global information

Principles
Fault tolerance through weaker anonymity in a single operational mode (anonymity-mode)
Fault tolerance through a special operational mode (fault tolerance-mode)
Braided RING

Two RINGs operating if no faults

Reconfiguration of the outer RING if a station fails

Reconfiguration of the inner RING if an outer line fails

Reconfiguration of the outer RING if an outer and inner line fails
Modifying attacks

modifying attacks at

sender anonymity

→ extend the access method

recipient anonymity

service delivery

publish input and output

if dispute: reconfiguration

covered in RING-network by attacker model
Superposed sending (DC-network)

Anonymity of the sender

If stations are connected by keys the value of which is completely unknown to the attacker, tapping all lines does not give him any information about the sender.
Dining Cryptographers

If the coin is heads, then I paid.
If the coin is tails, then I paid.
Now I know one of them paid—but since I can’t see which side the coin landed on, I don’t know which one of them paid.

Dinning Cryptographers

\[ [1] \leftarrow k \oplus 1 \]

\[ [2] \leftarrow k \]

\[ [1] \oplus [2] = 1 \]

DC-Net – Superposed Sending

Chaum, 1988

Key Graph

A ——— C

B ————

Note: In this example “sum” means XOR

True Message from A 00110101
Key with B 00101011
Key with C 00110110
Sum 00101000

A sends 00101000

Empty Message from B 00000000
Key with A 00101011
Key with C 01101111
Sum 01000100

B sends 01000100

Empty Message from C 00000000
Key with A 00110110
Key with B 01101111
Sum 01011001

C sends 01011001

Sum = True Message from A 00110101
Superposed sending (DC-network)

Anonymity of the sender

If stations are connected by keys the value of which is completely unknown to the attacker, tapping all lines does not give him any information about the sender.
Three distinct topologies

station 1

station 2

station 3

key topology independent of the others

superposition topology

transmission topology

dependent on each other
Reservation scheme

\[
\begin{array}{cccccc}
S_1 & 0 & 1 & 0 & 0 & 0 \\
S_2 & 0 & 1 & 0 & 0 & 0 \\
S_3 & 0 & 0 & 0 & 0 & 0 \\
S_4 & 0 & 1 & 0 & 1 & 0 \\
S_5 & 0 & 0 & 1 & 0 & 0 \\
\hline
0 & 3 & 1 & 1 & 0
\end{array}
\]

reservation frame

only different to “1” if “+” \(\neq\) “⊕”

\(T_5\)

\(T_4\)

message frame

\(\geq\) one round-trip delay

time
**Superposed receiving**

Whoever knows the sum of \( n \) characters and \( n-1 \) of these \( n \) characters, can calculate the \( n \)-th character.

**pairwise superposed receiving** (reservation scheme: \( n=2 \))

Two stations send simultaneously. Each subtracts their characters from the sum to receive the character sent by the other station. 

\[ \Rightarrow \text{Duplex channel in the bandwidth of a simplex channel} \]

**global superposed receiving** (direct transmission: \( n \geq 2 \))

Result of a collision is stored, so that if \( n \) messages collide, only \( n-1 \) have to be sent again.

**Collision resolution algorithm using the mean of messages:**

\[ \leq 2^S - 1 \text{ station} \quad \text{addition mod } 2^L \]

<table>
<thead>
<tr>
<th>( S )</th>
<th>( S-1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 ... 0</td>
<td>0 ... 0 1</td>
</tr>
</tbody>
</table>

`counter`

overflow area for addition of messages

overflow area for addition of counters
Pairwise superposed receiving

without superposed receiving

$S_1$

$S_2$

$(X+Y)-X = Y$

with pairwise superposed receiving

$S_1$

$S_2$

$(X+Y)-Y = X$
Collision resolution algorithm with *mean calculation* and *superposed receiving*
Collision resolution algorithm with mean calculation and superposed receiving
Analogy between Vernam cipher and superposed sending

Vernam cipher

$K + M = C \iff M = C - K$

abelian group

$M_1 + K = O_1$

$M_2 - K = O_2$
Proposition:
If stations $S_i$ are connected by uniform randomly distributed keys $K_j$ which are unknown to the attacker, by observing all the $O_i$, the attacker only finds out $\sum_i M_i$ about the $M_i$.

Proof:
$m=1$, trivial
step $m-1 \to m$
Proof of sender anonymity: induction step

Attacker observes $O_1, O_2, ... O_m$.

For each combination of messages $M'_1, M'_2, ... M'_m$

with $\sum_{i=1}^{m} M'_i = \sum_{i=1}^{m} O_i$ there is exactly one compatible combination of keys:

- $K' := O_m - M'_m$
- The other keys are defined as in the induction assumption, where the output of $S_L$ is taken as $O_L + K'$. 
Information-theoretic anonymity in spite of modifying attacks

Problems:

1) The attacker sends messages only to some users. If he gets an answer, the addressee was among these users.

2) To be able to punish a modifying attack at service delivery, corrupted messages have to be investigated. But this may not apply to meaningful messages of users truthful to the protocol.
DC$^+$-net to protect the recipient even against modifying attacks:

if broadcast error then uniformly distributed modification of keys

key between station $i$ and $j$ at time $t$

at station $i$ at time $t$
broadcast character

( skew-) field

$$K_{ij}^t = a_{ij}^t + \sum_{k=t-s}^{t-1} b_{ij}^{t-k} \bullet C_i^k$$

For practical reasons:
Each station has to send within each $s$ successive points in time a random message and observe, whether the broadcast is “correct“.
Modifying attacks at
sender anonymity
recipient anonymity
service delivery

attacker sends message character $\neq 0$,
if the others send their message character as well
$\rightarrow$ no transmission of meaningful information

To be able to punish a modifying attack at service delivery, corrupted messages have to be investigated. But this may \textit{not} apply to meaningful messages of users truthful to the protocol.
Protection of the sender: anonymous trap protocol

- Each user can cause investigating the reservation blobs directly after their sending if the sending of his reservation blobs did not work.
- Each user can authorize investigating of his “collision-free” random message, by opening the corresponding reservation blob.
Blob := committing to 0 or 1, without revealing the value committed to

1) The user committing the value must not be able to change it, but he must be able to reveal it.

In a “digital” world you can get exactly one property without assumptions, the other then requires a complexity-theoretic assumption.

Example:

Given a prime number $p$ and the prime factors of $p-1$, as well as a generator $\alpha$ of $\mathbb{Z}_p^*$ (multiplicative group mod $p$). Using $y$ everybody can calculate $\alpha^y \mod p$.

The inverse can not be done efficiently!

2) The others should not get any information about the value.

1?  
$s \in \mathbb{Z}_p^*$ randomly chosen  
(so user cannot compute $e$ such that $s \equiv \alpha^e$)

$x := s^b \alpha^y \mod p$ with $0 \leq y \leq p-2$

commit \(x\)

open \(y\)

2?

Let $2^u$ be the smallest number that does not divide $p-1$

$y := y_1, b, y_2$  
with $0 \leq y \leq p-2$ and $|y_2| = u-1$

$x := \alpha^y \mod p$

commit \(x\)

open \(y\)
Blobs based on factoring assumption

1?

prover

verifier

\[ n := p \cdot q \]
\[ s := t^2 \mod n \]
\[ s \in \text{QR}_n \]
\[ n, s \]

2?

prover

verifier

\[ n := p \cdot q \]
\[ s \not\in \text{QR}_n, \left(\frac{s}{n}\right) = 1 \]
\[ n = p \cdot q, s \not\in \text{QR}_n \]
\[ n, s \]

\[ x := y^2 s^b \mod n \]
\[ x \]

\[ y \]
Blobs based on asymmetric encryption system

2?

encrypt \( b \) with asymmetric encryption system (recall: public encryption key and ciphertext together uniquely determine the plaintext)

• has to be probabilistic – otherwise trying all possible values is easy

• communicating the random number used to probabilistically encrypt \( b \) means opening the blob

• computationally unrestricted attackers can calculate \( b \) (since they can break any asymmetric encryption system anyway)
Checking the behavior of the stations

To check a station it has to be known:

• All keys with others
• The output of the station
• All the global superposing results received by the station
• At what time the station may send message characters according to the access protocol
  (Can be determined using the global superposition results of the last rounds; These results can be calculated using the outputs of all stations.)

\[
\text{calculated message characters} \quad \text{compare}
\]

known = known to all stations truthful to the protocol
Collisions in the reservation phase

- cannot be avoided completely
- therefore they *must not* be treated as an attack

Problem: Attacker A could await the output of the users truthful to the protocol and then A could choose his own message so that a collision is generated.

Solution: Each station

1. defines its output using a Blob at first, then
2. awaits the Blobs of all other stations, and finally
3. reveals its own Blob's content.
Fault tolerance: 2 modes of operation

**A-mode**
- anonymous transmission of messages using superposed sending

**F-mode**
- sender and recipient are not protected

- fault detection
- fault localization
- taking defective components out of operation

- error recovery of the PRGs, initialization of the access protocol
Fault tolerance: sender-partitioned DC-network

The diagram illustrates a network of stations connected to different DC-networks. Each station is represented by a point, and the connections are indicated by lines between the points and the network labels.

- **Write and read access to the DC-network**: These points represent the ability to access the network for both write and read operations.
- **Read access to the DC-network**: These points indicate the ability to access the network for read operations only.

The network is partitioned such that the widest possible spread of a fault of station 3 affects the network, while a fault of station 5 affects only a smaller portion of the network.
Protection of the communication relation: MIX-network

\[ c_1(z_4, z_5, z_6, M_1, M_2, M_3) \]

**MIX\(_1\)** batches, discards repeats,
\[ d_1(c_1(z_i, M_i)) = (z_i, M_i) \]

**MIX\(_2\)** batches, discards repeats,
\[ d_2(c_2(z_i, M_i)) = (z_i, M_i) \]

D. Chaum 1981 for electronic mail

\[ c_1(z_1, z_2, z_3, M_1, M_2, M_3) \]
Idea: Provide unlinkability between incoming and outgoing messages

A Mix collects messages, changes their coding and forwards them in a different order.

If *all* Mixes work together, they can reveal the way of a given messages.
Protection of the communication relation: MIX-network

D. Chaum 1981 for electronic mail

\[ c_1(z_4, z_1, M_1) \]

\[ c_1(z_5, z_2, M_2) \]

\[ c_1(z_6, z_3, M_3) \]

\[ d_1(c_1(z_i, M_i)) = (z_i, M_i) \]

\[ c_2(z_3, M_3) \]

\[ c_2(z_1, M_1) \]

\[ c_2(z_2, M_2) \]

\[ d_2(c_2(z_i, M_i)) = (z_i, M_i) \]
Basic functions of a MIX

input messages

discard repeats

buffer current input batch

sufficiently many messages from sufficiently many senders? If needed: insert dummy messages

re-encrypt (decrypt or encrypt)

change order

all input messages which were or will be re-encrypted using the same key

output messages

MIX
Properties of MIXes

MIXes should be designed independently produced operated maintained ...

Messages of the same length
- buffer
- re-encrypt
- change order \[\text{batch-wise}\]

Each message processed only once!
- inside each batch
- between the batches

sym. encryption system only for
- first \(\text{MIX}\)
- last

asym. encryption system required for MIXes in the middle
Possibilities and limits of re-encryption

**Aim:** (without dummy traffic)

Communication relation can be revealed only by:

- *all* other senders and recipients together or
- *all* MIXes together which were passed through against the will of the sender or the recipient.

**Conclusions:**

1. **Re-encryption: never decryption directly after encryption**
   
   Reason: to decrypt the encryption the corresponding key is needed;
   
   - before and after the encoding of the message it is the same
   - re-encryption is irrelevant

2. **Maximal protection:  
   MIXes are passed through simultaneously and therefore in the same order**
Mix-network topologies

- cascades: fixed chain of Mixes

- free routes of Mixes: random selection by sender
Mix-network topologies

- restricted routes:
  - dedicated set of last Mix (Tor: Exit-Node)
  - fixed first Mix (Tor: Entry-Guard)
  - restricted set of Node neighbours
**Possibilities and limits of re-encryption**

**Aim:** (without dummy traffic)

Communication relation can be revealed only by:

- *all* other senders and recipients together or
- *all* MIXes together which were passed through against the will of the sender or the recipient.

**Conclusions:**

1. **Re-encryption:** never decryption directly after encryption
   
   Reason: to decrypt the encryption the corresponding key is needed;
   
   ➔ before and after the encoding of the message it is the same
   ➔ re-encryption is irrelevant

2. **Maximal protection:**
   MIXes are passed through simultaneously and therefore in the same order
Maximal protection

Pass through MIXes in the same order
Best case:

- Anonymity set size: 6
- 1 honest Mix
Maximal protection

Best case:
- Anonymity set size: 6
- 1 honest Mix

Alternative Architecture, therefore:
Pass through all honest MIXes in the same order.
Maximal protection

Best case:
- Anonymity set size: 6
- 1 honest Mix

Alternative Architecture, therefore:
Pass through all honest MIXes in the same order.
Problem: You don’t know which is honest…
Therefore:
Pass through all MIXes in the same order.
3 honest Mixes / Anonymity Set Size: 4

Mix 1a
S1

Mix 1b
S2

Mix 1c
S3

Mix 2a
S4

Mix 2b
S5

Mix 2c
S6

Mix 3a

Mix 3b

Mix 3c
3 honest Mixes / Anonymity Set Size: 2
Re-encryption scheme for sender anonymity

indirect re-encryption scheme for sender anonymity

\[ M_{n+1} = c_{n+1}(M) \]

\[ M_i = c_i(z_i, A_{i+1}, M_{i+1}) \text{ for } i = n, \ldots, 1 \]

\[ M_i = c_i(k_i, A_{i+1}); k_i(M_{i+1}) \]
Indirect re-encryption scheme for recipient anonymity

\[ H_{m+1} = e \]

\[ H_j = c_j(k_j, A_{j+1}, H_{j+1}) \text{ for } j = m, \ldots, 0 \]

message header \( H \)

message content \( l \)

\[ l_1 = k_0(l) \]

\[ l_j = k_{j-1}(l_{j-1}) \text{ for } j = 2, \ldots, m+1 \]

encryption \downarrow \text{ observable transfer} \uparrow \text{ decryption}

observable transfer

unobservable transfer
Indirect re-encryption scheme for sender and recipient anonymity

1. S ➔ MIX₁ ➔ MIX₂ ➔ MIX₃
2. MIX₄ ➔ MIX₅ ➔ R

- **Message header**
  - d₃ k₃
  - 3rd party, to hold the anonymous return addresses for anonymous query
  - pickup using recipient anonymity scheme, initiated using sender anonymity scheme

- **Message content**
  - c₁ k₁, d₁ k₁, c₂ k₂, d₂ k₂, c₃ k₃, d₃ k₃
  - for sender anonymity

- **Encryption**
  - unobservable transfer

- **Decryption**
  - observable transfer

- **Key Exchange**
  - k₁, k₂, k₃, k₄, k₅
Indirect re-encryption scheme maintaining message length

\[ H_{m+1} = [e] \]

\[ H_j = [c_j(k_j, A_{j+1})], \ k_j(H_{j+1}) \quad \text{for} \ j = m,..,1 \]
Indirect re-encryption scheme maintaining message length for special symmetric encryption systems

\[ M_j \]

blocks with message contents

blocks with random contents

\[ k_{j+1}(H_{j+2}) \]

\[ H_{j+1} \]

\[ H_j \]

\[ k_j(H_{j+1}) \]

\[ k_j, A_{j+1} \]

\[ M_{j+1} \]

blocks with message contents

blocks with random contents

\[ k^{-1}(k(M)) = M \]

\[ k(k^{-1}(M)) = M \]
Minimally message expanding re-encryption scheme maintaining message length

If $k^{-1}(k(M)) = M$

and $k(k^{-1}(M)) = M$
Recall: Diffie-Hellman key agreement

**Key Generation:**
- $x \in \mathbb{Z}_p^*$
- $g^x \mod p$

**Calculating Shared Key:**
- $(g^y)^x \mod p = g^{yx} = g^{xy} = (g^x)^y \mod p$

**Publicly Known:**
- $p$ and $g \in \mathbb{Z}_p^*$

**Random Number:**
- Random number 1
- Random number 2

**Domain of Trust:**
- Area of attack
Recall: Diffie-Hellman key agreement – “modes of operation”

- **static – static**
  - sender & recipient use long time static DH keys

- **ephemeral – static**
  - recipient: long time static DH key
  - sender: newly create random DH-key („session key“)
  - new DH secret with every key exchange
  - ElGamal encryption system

- **static – ephemeral**

- **ephemeral – ephemeral**
  - sender & recipient use newly create random DH-keys
  - forward secrecy
Mix Packets based on Diffie-Hellman Key Agreement

- first idea:
  - ephemeral – static mode
  - user creates DH key for every mix $M_i$:
    - $x_i, y_i = g^{x_i} \mod p$
    - secret $k_i$ shared with $M_i$: $k_i = y_{M_i}^{x_i} \mod p$
  - layered encryption:
    - $y_i, k_i(y_{i+1}, k_{i+1}(\ldots))$
  - overhead:
    - per mix: size of $y_i$
Mix Packets based on Diffie-Hellman Key Agreement

• more efficient idea:
  – ephemeral-static – static mode
    ➔ ephemeral: sender creates new DH key for every packet
    ➔ static: same DH key for all mixes!
  – user creates DH key (same for every mix $M_i$):
    • $x, y = g^x \mod p$
    • secret $k_i$ shared with $M_i$: $k_i = y_{M_i}^x \mod p$
  – layered encryption:
    • $y, k_i(k_{i+1}(…))$
Mix Packets based on Diffie-Hellman Key Agreement

- layered encryption:
  - $y, k_i(k_{i+1}(\ldots))$

- How to achieve?
  - Problem:
    - all mixes know $y$
    - linkability!
  - Solution:
    - calculate $y_{i+1}$ from $y_i$
Mix Packets based on Diffie-Hellman Key Agreement

– Solution:

• calculate \( y_{i+1} \) from \( y_i \)
• \( x_{i+1} = x_i^{b_i} \mod p \)
• \( b_{i+1} = \text{Hash}(y_i, k_i) \)

\[
\begin{align*}
\quad y_{i+1} &= g^{x_{i+1}} \mod p \\
&= g^{x_i b_i} \mod p \\
&= y_i^{b_i} \mod p
\end{align*}
\]

\( \Rightarrow \) mix \( M_i \) can calculate \( y_{i+1} \) from \( y_i \)!
\( \Rightarrow \text{only} \ M_i \) can calculate \( y_{i+1} \) from \( y_i \)!
Implementation of MIXes using RSA without redundancy predicate and with contiguous bit strings (David Chaum, 1981) is insecure:

\[(z, M) \rightarrow \text{MIX} \rightarrow ((x, y)) \rightarrow (x, y) \pmod{n} \rightarrow y \]

\(|z| = b \quad |M| = B\)

Attacker multiplies \(M\) with factor \(f\) and compares

\[(z, M)^c \cdot f^c \approx M \cdot f\]

Unlinkability, if many factors \(f\) are possible.

\[2^b \cdot 2^B \leq n - 1\] hold always and normally \(b << B\).

If the random bit strings are the most significant bits, it holds

\[(z, M) = z \cdot 2^B + M\]

and

\[(z, M) \cdot f \equiv (z \cdot 2^B + M) \cdot f \equiv z \cdot 2^B \cdot f + M \cdot f.\]
Breaking the direct RSA-implementation of MIXes (2)

Let the identifiers \( z' \) and \( M' \) be defined by

\[
(z,M)\cdot f \equiv z'\cdot 2^B + M' \quad \Rightarrow
\]
\[
z\cdot 2^B\cdot f + M\cdot f \equiv z'\cdot 2^B + M' \quad \Rightarrow
\]
\[
2^B\cdot (zf - z') \equiv M' - M\cdot f \quad \Rightarrow
\]
\[
z\cdot f - z' \equiv (M' - M\cdot f) \cdot (2^B)^{-1} \quad (1)
\]

If the attacker chooses \( f \leq 2^b \), it holds

\[-2^b < zf - z' < 2^{2b} \quad (2)\]

The attacker replaces in (1) \( M \) and \( M' \) by all output-message pairs of the batch and tests (2).

(2) holds, if \( b \ll B \), very probably only for one pair \((P1, P2)\). \( P1 \) is output message to \((z, M)^c\), \( P2 \) to \((z, M)^c\cdot f^c\).

If (2) holds for several pairs, the attack is repeated with another factor.
Fault tolerance in MIX-networks (1)

2 alternative routes via disjoint MIXes

MIX$_i$ or MIX$_{i''}$ can substitute MIX$_i$
Fault tolerance in MIX-networks (2)

In each step, one MIX can be skipped.
### Complexity of the basic methods

<table>
<thead>
<tr>
<th>attacker model</th>
<th>RING-network</th>
<th>DC-network</th>
<th>MIX-network</th>
</tr>
</thead>
<tbody>
<tr>
<td>physically limited</td>
<td>unobservability of neighboring lines and stations as well as digital signal regeneration</td>
<td>computationally restricted w.r.t. service delivery</td>
<td>computationally restricted not even well analyzed asymmetric encryption systems are known which are secure against adaptive active attacks</td>
</tr>
<tr>
<td>O(n) (≥ n/2) transmission</td>
<td>O(n) (≥ n/2) transmission</td>
<td>O(k\cdot n) key</td>
<td></td>
</tr>
</tbody>
</table>

\[ n = \text{number of users} \]
\[ k = \text{connectedness key graph of DC-networks respectively number of MIXes} \]
In the OSI model it holds:

Layer $n$ doesn’t have to look at Data Units (DUs) of layer $n+1$ to perform its service. So layer $n+1$ can deliver $(n+1)$-DUs encrypted to layer $n$.

For packet-oriented services, the layer $n$ typically furnishes the $(n+1)$-DUs with a $n$-header and possibly with an $n$-trailer, too, and delivers this as $n$-DU to layer $n-1$. This can also be done encrypted again.

and so on.

All encryptions are independent with respect to both the encryption systems and the keys.
Arranging it into the OSI layers (1)

<table>
<thead>
<tr>
<th>OSI layers</th>
<th>7 application</th>
<th>6 presentation</th>
<th>5 session</th>
<th>4 transport</th>
<th>3 network</th>
<th>2 data link</th>
<th>1 physical</th>
<th>0 medium</th>
</tr>
</thead>
</table>

- **End-to-end encryption**
- **Link encryption**
### Arranging it into the OSI layers (2)

<table>
<thead>
<tr>
<th>OSI layers</th>
<th>broadcast</th>
<th>query</th>
<th>MIX-network</th>
<th>DC-network</th>
<th>RING-network</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 application</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 presentation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 session</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 transport</td>
<td>implicit</td>
<td>implicit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>addressing</td>
<td>addressing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 network</td>
<td>broadcast</td>
<td>query and superpose</td>
<td>buffer and re-encrypt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 data link</td>
<td></td>
<td></td>
<td>anonymous access</td>
<td>anonymous access</td>
<td></td>
</tr>
<tr>
<td>1 physical</td>
<td>channel selection</td>
<td></td>
<td>superpose keys and messages</td>
<td>digital signal regeneration</td>
<td></td>
</tr>
<tr>
<td>0 medium</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ring</td>
</tr>
</tbody>
</table>

- **Yellow** has to preserve anonymity against the communication partner
- **Green** end-to-end encryption
- **Orange** has to preserve anonymity
- **Brown** realizable without consideration of anonymity
Solution for the ISDN: telephone MIXes

Aims: ISDN services on ISDN transmission system
- 2 independent 64-kbit/s duplex channels on a 144-kbit/s subscriber line
- hardly any additional delay on established channels
- establish a channel within 3 s
- no additional traffic on the long distance network

Network structure
Solution for the ISDN: telephone MIXes (1989)

**Aims: ISDN services on ISDN transmission system**
- 2 independent 64-kbit/s duplex channels on a 144-kbit/s subscriber line
- hardly any additional delay on established channels
- establish a channel within 3 s
- no additional traffic on the long distance network

**Network structure**

![Network diagram with MIX1, MIXm, MIX'm, MIX', R, G, LE(R), LE(G)]
Time-slice channels (1)

station $R$  MIXes($R$)  LE($R$)  LE($G$)  MIXes($G$)  station $G$

$S_0$

\[ \begin{align*}
\text{TS-setup: } x \\
\text{TR-setup: } x 
\end{align*} \]

\[ \begin{align*}
\text{call request: } c_G(k, s_R, \text{and } s_G) \\
\text{query and superpose instead of broadcast}
\end{align*} \]

\[ \begin{align*}
\text{TS} \\
\text{TR} \
\end{align*} \]

\[ \begin{align*}
\text{TS-setup: } \text{PBG}(s_G, 1) \\
\text{TR-setup: } \text{PBG}(s_R, 1)
\end{align*} \]

$S_1$

\[ \begin{align*}
\text{TS} \\
\text{TR} \
\end{align*} \]

\[ \begin{align*}
\text{TS-setup: } \text{PBG}(s_R, 1) \\
\text{TR-setup: } \text{PBG}(s_G, 1)
\end{align*} \]
This setup of receiving channels is a very flexible scheme for recipient anonymity.
Tor

- basic building block:
  - symmetric encrypted channels → called: circuits
  - multiple streams multiplexed over one circuit

- Mix packet: cells
  - 512 bytes

- asymmetric crypto for key exchange: Diffie-Hellman
  - telescopically
    - CREATE-Cell sent to next Tor node over already established circuit
Tor: Hidden Services

① establishes circuit
② publishes introduction point anonymously
③ searches for introduction point
④ asks for rendezvous point
⑤ establishes circuit
⑥ establishes circuit

Directory Service
Hidden Service
Introduction Point
Rendezvous Point
Connection configuration later (1)

station $R$ \hspace{1cm} MIXes($R$) \hspace{1cm} LE($R$) \hspace{1cm} LE($G$) \hspace{1cm} MIXes($G$) \hspace{1cm} station $G$

$S_0$

- TS-setup: $x$
- TR-setup: $x$

$S_1$

- TS-setup: PBG(sG,1)
- TR-setup: PBG(sR,1)

call request: $c_G(k, sR, \text{and } sG)$

from $P$ to $P$ \hspace{1cm} PBG(sQ,0) \hspace{1cm} TR \hspace{1cm} TS
Connection configuration later (2)

$S_2$

TS-setup: $\text{PBG}(sG, 2)$
TR-setup: $\text{PBG}(sR, 2)$

throw away

$\text{PBG}(sR, 1)$

replace

from P $\text{PBG}(sQ, 1)$

to P

$S_{t-1}$

TS-setup: $\text{PBG}(sG, t-1)$
TR-setup: $\text{PBG}(sR, t-1)$

$S_t$

TS-setup: $\text{PBG}(sR, t-1)$

TR-setup: $\text{PBG}(sG, t-1)$

$\text{PBG}(sG, t-1)$

k(dial tone, data)
Query and superpose: 

- *Each* station has to query in each time slice (else the anonymity set degenerates)
- *Each* station should inquire *all* its implicit addresses at each query.
  (possible both for visible and invisible addresses without additional expense)

→ The size of the anonymity set is no longer limited by the transmission capacity on the user line, but only by the addition performance of the message servers.
Radio networks (1)

Difference to wired networks
• Bandwidth of transmission remains scarce
• The current place of the user is also to be protected

Assumptions
• Mobile user station is always identifiable and locatable if the station sends.
• Mobile user station is not identifiable and locatable if the station only (passively) receives.

Which measures are applicable?
+ end-to-end encryption
+ link encryption
- dummy messages, unobservability of neighboring lines and stations as well digital signal regeneration, superposed sending

➔ all measures to protect traffic data and data on interests have to be handled in the wired part of the communication network
Radio networks (2)

+ MIXes

1. Broadcast the call request in the whole radio network, only then the mobile station answers. After this the transmission proceeds in one radio cell only.

+ Filter + Generation of visible implicit addresses + Restrict the region

+ Keep the user and SIM anonymous towards the mobile station used.

if the coding in the radio network is different or computing power for encryption is missing
No movement profiles in radio networks

GSM/UMTS – cellular mobile networks

- roaming information in central data bases
- operators of the network can record the information

Alternative concept

- Maintenance of the roaming information in a domain of trust
  - at home (HPC)
  - at trustworthy organizations
- Protection of the communication relationship using MIXes
• **Use Case:**
  – Location-aware Apps

• **Assumptions:**
  – untrusted Apps are interested in location inside a defined geographic region (*application zone*)
  – trusted middleware

• **Idea:**
  – middleware reveals location using App-specific user pseudonyms

• **Problem:**
  – colluding Apps

• **Solution:**
  – Mix Zones: no location tracing at all
Mix Zones: User Privacy in Location-aware Services

[Alastair R. Beresford, Frank Stajano, 2004]

- Timing information!
Conclusions & Outlook (1)

Using the network transactions between anonymous partners explicit proof of identity is possible at any time

Protection of traffic data and data on interests requires appropriate network structure

consider early enough

keep options

Networks offering anonymity can be operated in a “trace users mode” without huge losses in performance, the converse is not true!
Trustworthy data protection in general or only at individual payment for interested persons?

- Concerning traffic data, the latter is technically inefficient.
- The latter has the contrary effect (suspicion).
- Everyone should be able to afford fundamental rights!
Electronic Banking

Motivation

• Banking using paper forms – premium version
  Customer gets the completely personalized forms from the bank in which only the value has to be filled in. No signature!

Electronic banking – usual version
  Customer gets card and PIN, TAN from his/her bank.
  
http://www.cl.cam.ac.uk/research/security/banking/

Upcoming / Current
  Customer gets chip card from Bank with
  or
  - key for MAC
  - key pair for digital signature

• Map exercise of US secret services: observe the citizens of the USSR (1971, Foy 75)

Main part (Everything a little bit more precise)

• Payment system is secure ...
  - MAC, digital signature
  - payment system using digital signatures

• Pseudonyms (person identifier ↔ role-relationship pseudonyms)
Chip & PIN Problem

Verify PIN

PIN ok

Verify PIN

PIN ok
Chip & PIN Problem

Verify PIN

PIN ok

Verified by Signature

Signed

Transaction Record
Security properties of digital payment systems

digital (integrity, availability)

Payment system is **secure** if

- user can transfer the rights received, via communication network immaterial, digital
- user can loose a right only if he is willing to,
- if a user who is willing to pay uniquely denotes another user as recipient, only this entity receives the right,
- user can prove transfers of rights to a third party if necessary (receipt problem), and
- the users cannot increase their rights even if they collaborate, without the committer being identified.

Problem: messages can be copied perfectly
Solution: witness accepts only the *first* (copy of a) message
Scalability concerning the protection of anonymity
Distinction between:

1. **Initial linking** between the pseudonym and its holder

2. Linkability due to the use of the pseudonym **across** different contexts
Pseudonyms: Initial linking to holder

**Public pseudonym:**
The linking between pseudonym and its holder may be publicly known from the very beginning.

Phone number with its owner listed in public directories

**Initially non-public pseudonym:**
The linking between pseudonym and its holder may be known by certain parties (trustees for identity), but is not public at least initially.

Bank account with bank as trustee for identity,
Credit card number ...

**Initially unlinked pseudonym:**
The linking between pseudonym and its holder is – at least initially – not known to anybody (except the holder).

Biometric characteristics; DNA (as long as no registers)
Pseudonyms: Use across different contexts => partial order

A → B stands for “B enables stronger unlinkability than A”
Notations: transfer of a signed message from $X$ to $Y$

**functional notation**

signing the message $M$: 
$s_A(M)$

$X \overset{M, s_A(M)}{\rightarrow} Y$

test the signature: 
$t_A(M, s_A(M))$ ?

**graphical notation**

sender $X$  

recipient $Y$
Authenticated anonymous declarations between business partners that can be de-anonymized

trusted third party $A$

identification

user $X$

confirmed

for $p_G(Y,g)$

know $p_G(X,g)$

$p_A$

confirmed

for $p_G(Y,g)$

know $p_G(X,g)$

$p_B$

trusted third party $B$

identification

user $Y$

Generalization:

$X \rightarrow B_1 \rightarrow B_2 \rightarrow ... \rightarrow B_n \rightarrow Y$

$B'_1 \rightarrow B'_2 \rightarrow ... \rightarrow B'_m$

error / attack tolerance (cf. MIXes)
Authenticated anonymous declarations between business partners that can be de-anonymized

trusted third party $A$

user $X$

trustees for identities

trusted third party $B$

user $Y$

Generalization:

$$X \rightarrow B_1 \rightarrow B_2 \rightarrow \ldots \rightarrow B_n \rightarrow Y$$

$$B'_1 \rightarrow B'_2 \rightarrow \ldots \rightarrow B'_m$$

error / attack tolerance (cf. MIXes)
Security for completely anonymous business partners using active trustee who can check the goods

1. Order: Merchant is $p_L(Y,g)$ + "money" for merchant

2. Order of the customer (money is deposited)

3. Delivery to trustee

4. Delivery to customer checked by $T$

5. Money

Trustee $T$

Customer $X$

Merchant $Y$
Security for completely anonymous business partners using active trustee who can **not** check the goods

1. Order
   - Delivery is $p_L(Y,g)$ + “money” for distributor

2. Order of the customer
   - (Money is deposited)

3. Delivery to trustee
   - $p_L(Y,g)$

4. Delivery to customer
   - Checked by

5. Money
   - Checked by

Customer $X$

Trustee $T$

Merchant $Y$
Security for completely anonymous business partners using active trustee who can (not) check the goods

trustee for values

[1] order of the customer (money is deposited)

[2] delivery to trustee

[3] order of the customer

[4] delivery to customer

[5] money

[4.1] wait

trustee T

pT

pK(X, g)

customer X

pT

merchant Y

pT

customer X

trustee T

money

pL(Y, g)

delivery to customer

checkered by

pT

merchant Y
### Anonymously transferable standard values

**Current owner:**
- Digital pseudonym

**Value number:** $v_n$

**Value:** 10 $\$

**Former owners:**
- Digital pseudonym 1, transfer order 1
- Digital pseudonym 2, transfer order 2
- Digital pseudonym 3, transfer order 3
- .....
• Key feature: Bitcoin transfer between pseudonyms (Bitcoin addresses)
• Bitcoin pseudonym ≡ public key of ECDSA
• Sender signs transfer
• Double spending protection:
  – Bitcoin network keeps history of all transactions
  – Transactions have timestamps → only oldest is valid
    • Bitcoin network works as “distributed time server”
  – Binding of transaction and timestamp: „proof-of-work“:
    • search for $z$: $\text{Hash}(\text{Transaction, Timestamp, } z) = 00000… (0|1)^* < w$
    • $w$ adjusted over timer
• https://www.blockchain.info
Basic scheme of a secure and anonymous digital payment system

1. Choice of pseudonyms
   - \( p_E(Y,t) \approx p_E(B,Y,t) \)
   - \( p_Z(X,t) \approx p_Z(B,X,t) \)

2. Transfer order of the payer
   - \( p_Z(X,t) \) owns the right
   - \( p_Z(B,X,t) \) have transferred the right to \( p_E(B,Y,t) \)

3. Authentication by the witness
   - \( p_E(B,Y,t) \) owns the right, got from \( p_Z(B,X,t) \)

4. Receipt for the payer
   - \( p_E(Y,t) \)
   - \( p_Z(X,t) \) have got the right from \( p_Z(B,X,t) \)

5. Authentication for the recipient
   - \( p_E(Y,t) \)
   - \( p_Z(X,t) \) have transferred the right to \( p_E(Y,t) \)
Transformation of the authentication by the witness

[1] choice of pseudonyms
\[ p_E(Y,t) \approx p_E(Y,t) \]
\[ p_Z(X,t) \approx p_Z(X,t) \]

[2] transfer order of the payer
\[ p_Z^B(X,t) \] owns the right
transfer the right to \[ p_E^B(Y,t) \]

[3] authentication by the witness
\[ p_E^B(Y,t) \] owns the right, got from \[ p_Z^B(X,t) \]

[4] receipt for the payer
\[ p_E(Y,t) \]
have got the right from \[ p_Z(X,t) \]

[5] authentication for the recipient
\[ p_Z(X,t) \] have transferred the right to \[ p_E(Y,t) \]

[6] \[ p_Z^B(Y,t+1) \] owns the right
Transformation of the authentication by the witness: Simplified Steps

1. **Payer Y**
   - Pays EUR 10 to **witness B**

2. **witness B**
   - Receives EUR 10 from **Payer Y**

3. **witness B**
   - Transfers EUR 10 to **recipient Z**

4. **recipient Z**
   - Receives EUR 10 from **witness B**
Transformation of the authentication by the witness

1. **choice of pseudonyms**
   - $p_B(X,t)$ owns the right
   - $p_E(Y,t) \approx p_B(Y,t)$
   - $p_Z(X,t) \approx p_B(X,t)$

2. **transfer order of the payer**
   - $p_B(X,t)$ owns the right
   - transfer the right to $p_E(Y,t)$

3. **authentication by the witness**
   - $p_E(B,Y,t)$ owns the right, got from $p_B(X,t)$

4. **receipt for the payer**
   - have got the right from $p_Z(X,t)$

5. **authentication for the recipient**
   - have transferred the right to $p_E(Y,t)$
   - $p_Z(X,t)$

6. **$p_B(Y,t+1)$ owns the right**
The next round: Y in the role payer to recipient Z

[0] payer Y
- \( p_{Z,Y}^{B}(t+1) \) owns the right
- \( p_{B} \)

[1] choice of pseudonyms
- \( p_{E}(Z,t+1) \approx p_{E}^{B}(Z,t+1) \)
- \( p_{Z}(Y,t+1) \approx p_{Z}^{B}(Y,t+1) \)

[2] transfer order of the payer
- \( p_{Z,Y}^{B}(t+1) \) owns the right
- \( p_{Z}(Y,t+1) \)
- \( p_{E}(Z,t+1) \)

[3] authentication by the witness
- \( p_{E}^{B}(Z,t+1) \) owns the right, got from \( p_{Z,Y}^{B}(t+1) \)

[4] receipt for the payer
- \( p_{E}(Z,t+1) \)
- \( p_{Z}(Y,t+1) \)
- \( p_{B} \)

[5] authentication for the recipient
- \( p_{Z}(Y,t+1) \)
- \( p_{E}(Z,t+1) \)
- \( p_{B} \)

The witness B
- \( p_{B} \)

Authentication of ownership
- \( p_{Z,Y}^{B}(t+1) \) owns the right
- \( p_{Z}(Y,t+1) \)
- \( p_{E}(Z,t+1) \)
- \( p_{Z}(Y,t+1) \)
- \( p_{E}(Z,t+1) \)

Transferred right to
- \( p_{E}^{B}(Z,t+1) \)
- \( p_{Z}(Y,t+1) \)
- \( p_{B} \)

Have transferred the right to
- \( p_{E}^{B}(Z,t+1) \)
- \( p_{Z}(Y,t+1) \)
- \( p_{B} \)

Have got the right from
- \( p_{Z}(Y,t+1) \)
- \( p_{E}(Z,t+1) \)
- \( p_{B} \)
Signature system for signing blindly

key generation

key for testing of signature, publicly known

t

random number

key for signing, kept secret

blind

random number’ z’
x

blind text

z'(x)

signing

s

text with signature

and test result

x, s(x), “pass” or “fail”

unblind and test

blinded text

blinded text with signature

z'(x), s(z'(x))
RSA as digital signature system with collision-resistant hash function h

key generation:
- p, q prime numbers
- n := p•q
- t with gcd(t, (p-1)(q-1)) = 1
- s ≡ t⁻¹ mod (p-1)(q-1)

key for testing of signature, publicly known: t, n

random number

security parameter

key for signing, kept secret: s, n

text with signature and test result: x, (h(x))^s mod n, "pass" or "fail"

test:
- h(1. comp.) ≡ (2. comp.)^t mod n ?

signing:
- (h(x))^s mod n

text with signature x, (h(x))^s mod n
One time convertible authentication

**Recipient**

choose pseudonym

\[ p \]

(test key of arbitrary sign. system)

Collision-resistant hash function \( h \)

\[ p, h(p) \]

choose \( r \in \mathbb{Z}_n^* \)

\[ (p, h(p)) \cdot r^t \]

\[ (p, h(p))^s \cdot r \]

multiply with \( r^{-1} \)

get

\[ (p, h(p))^s \]

**Issuer (i.e. witness)**

RSA test key \( t, n \), publicly known

\[ ((p, h(p)) \cdot r^t)^s \]
Secure device: 1st possibility

[1] choice of pseudonyms
\[ p_E(Y,t) = p_E^B(Y,t) \]
\[ p_Z(X,t) = p_Z^B(X,t) \]

[2] transfer order of the payer
\[ p_Z^B(X,t) \]
\[ p_B \]

[3] authentication by the witness
\[ p_E^B(Y,t) \] owns the right, got from \[ p_Z^B(X,t) \]

[4] receipt for the payer
\[ p_E(Y,t) \] have got the right from \[ p_Z(X,t) \]

[5] authentication for the recipient
\[ p_E(Y,t) \] have transferred the right to \[ p_E(Y,t) \].

Payer X

Recipient Y
Secure device: 2nd possibility

[1] choice of pseudonyms
\[ p_{Z}(X,t) = p_{Z}^{B}(X,t) \]
\[ p_{E}(Y,t) = p_{E}^{B}(Y,t) \]

[2] transfer order of the payer
\[ p_{Z}^{B}(X,t) \text{ owns the right} \]

[3] authentication by the witness
\[ p_{E}^{B}(Y,t) \text{ owns the right, got from } p_{Z}^{B}(X,t) \]

[4] receipt for the payer
\[ p_{E}(Y,t) \text{ have got the right from } p_{Z}(X,t). \]

[5] authentication for the recipient
\[ p_{Z}(X,t) \text{ have transferred the right to } p_{E}(Y,t). \]

sym. encryption system suffices
Offline payment system

Payment systems with security by Deanonymizability

$k$ security parameter

$l$ identity of the entity giving out the banknote

$r_i$ randomly chosen ($1 \leq i \leq k$)

$C$ commitment scheme with information theoretic secrecy

blindly signed banknote:

$s_{Bank}(C(r_1), C(r_1 \oplus l), C(r_2), C(r_2 \oplus l), \ldots, C(r_k), C(r_k \oplus l)),$

recipient decides, whether he wants to get revealed $r_i$ or $r_i \oplus l$.
(one-time pad preserves anonymity.)

Hand-over to two honest recipients:
probability ($\exists i: $ bank gets to know $r_i$ and $r_i \oplus i$) $\geq 1 - e^{-c \cdot k}$
(original owner identifiable)
Secure and anonymous digit. payment system with accounts

The system involves the following steps:

1. **Choice of Pseudonyms**
   - \( p_{E}(Y, t) \approx p_{E}(Y, t) \)
   - \( p_{Z}(X, t) \approx p_{Z}(X, t) \)

2. **Transfer of Order of the Payer**
   - \( p_{Z}(X, t) \) transfers the right to \( p_{E}(Y, t) \).

3. **Authentication by the Witness**
   - \( p_{E}(Y, t) \) owns the right, got from \( p_{Z}(X, t) \).

4. **Receipt for the Payer**
   - \( p_{E}(Y, t) \) have got the right from \( p_{Z}(X, t) \).

5. **Authentication for the Recipient**
   - \( p_{E}(Y, t) \) have transferred the right to \( p_{E}(Y, t) \).

6. **Transfer of Right**
   - \( p_{E}(Y, t) \) transfers the right to \( p_{B}(Y, t) \).

7. **Authentication of Ownership**
   - \( p_{B}(Y, t) \) owns the right, got from \( p_{Z}(X, t) \).

8. **Transfer Order of the Payer**
   - \( p_{Z}(X, t) \) transfers the right to \( p_{E}(Y, t) \).
Personal identifier

845 authorizes A:

A notifies 845:

845 pays B €

B certifies 845:

C pays 845 €

Organisation

A

Organisation

B

Organisation

C

DOSSIER

Smith
Role pseudonyms
(business-relationship and transaction pseudonyms)
Usually: one identity per user

Problem: Linkability of records
Many Partial-Identities per user

Management / disclosure / linkability under the control of the user
many services need only a **few data**

revealing that data under a **Pseudonym** prevents unnecessary linkability with other data of the user

**different actions / data** are initially unlinkable if one uses different pseudonyms

**Example: Car Rental**

necessary data:
- Possession of a driving license valid for the car wanted
Anonymous Credentials

 Credential = Attestation of an attribute of a user (e.g. „User has driving license“)

 Steps:
 - Organisation issues credentials
 - User shows credential to service provider

 Properties:
 - User can show credentials under different pseudonyms (transformation)
 - Usage of the same credential with different pseudonyms prevents linkability against the service provider and the issuer.
Usage of Anonymous Credentials

User A has driving-license

Credentials issuing Organisation

have driving-license

Service provider

have driving-license

User B

have driving-license

User X

have driving-license
Data Publishing – Use-Case

Collection → Anonymization → Publishing
Data Publishing – Classification of Data

- Explicit identifiers must be removed
- Link between **Quasi-IDs** and sensitive attributes needs to be obfuscated

<table>
<thead>
<tr>
<th>Quasi ID</th>
<th>Age</th>
<th>Sex</th>
<th>Disease</th>
<th>Salary</th>
<th>Q1</th>
<th>Q2</th>
</tr>
</thead>
<tbody>
<tr>
<td>47677</td>
<td>43</td>
<td>Male</td>
<td>Heart</td>
<td>3.000</td>
<td>a1</td>
<td>13</td>
</tr>
<tr>
<td>47602</td>
<td>22</td>
<td>Female</td>
<td>Flu</td>
<td>5.000</td>
<td>a5</td>
<td>4</td>
</tr>
<tr>
<td>47678</td>
<td>45</td>
<td>Female</td>
<td>Hepatitis</td>
<td>6.000</td>
<td>a4</td>
<td>22</td>
</tr>
<tr>
<td>47905</td>
<td>31</td>
<td>Male</td>
<td>HIV</td>
<td>4.000</td>
<td>a1</td>
<td>12</td>
</tr>
<tr>
<td>47909</td>
<td>36</td>
<td>Male</td>
<td>Flu</td>
<td>10.000</td>
<td>a2</td>
<td>8</td>
</tr>
</tbody>
</table>
Quasi-IDs: an Example

- Re-identification through directly linking shared attributes

- 87% of US population show characteristics to be uniquely identifiable through \{ZIP, Date of birth, Sex\} (Census 1990)

Data Publishing – Classification of Data

- Explicit identifiers must be removed
- Link between Quasi-IDs and sensitive attributes needs to be obfuscated
  - Generalization & Suppression
  - Anatomization & Permutation
  - Perturbation

<table>
<thead>
<tr>
<th>Quasi ID</th>
<th>Sensitive</th>
<th>Non-sensitive</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ZIP</td>
<td>Age</td>
</tr>
<tr>
<td>47677</td>
<td>47677</td>
<td>43</td>
</tr>
<tr>
<td>47602</td>
<td>47602</td>
<td>22</td>
</tr>
<tr>
<td>47678</td>
<td>47678</td>
<td>45</td>
</tr>
<tr>
<td>47905</td>
<td>47905</td>
<td>31</td>
</tr>
<tr>
<td>47909</td>
<td>47909</td>
<td>36</td>
</tr>
</tbody>
</table>
Data Publishing – Anonymization (\(k\)-Anonymity)

- Groups of \(k\) records resulting in \(k\)-anonymous table
- Probability \(1/k\) to link correct entry to known quasi-identifier
- Tradeoff between privacy and utility
  - larger groups normally result in less accurate data
- **Problem: Homogeneity in sensitive attributes**
  - Solution: \(l\)-diversity \(\Rightarrow\) at least \(l\) different values for each sensitive attribute in each equivalence class
  - **Problem:** meaning of “different”: different kinds of cancer \(\Rightarrow\) cancer
    - Solution: \(t\)-closeness
Privacy-Preserving Data Mining

- Secure Computations
  - min. 2 parties
  - distributed inputs or outsourced computations
  - different requirements
  - no single point of trust
  - protocol design

- Secure string matching
  - sequence comparisons
  - similarity between strings
  - fuzzy text search
  - basis for text mining
Privacy-Preserving Data Mining
Secure Multi-Party Computations

Secret Sharing
Secure Computation
Result Delivery
Privacy-Preserving Data Mining
Homomorphic Encryption

1 Encryption

2 Crypted Processing

3 Decrypting Result
Computation with secret inputs

- inputs could be from different parties

Based on the properties of a Homomorphism:

\[ f(a) \circ f(b) = f(a+b) \]

In principle: arbitrary "circuits" / algorithms computable

- huge overhead!

Secure Computation—Homomorphic Encryption
Cryptography and the impossibility of its legal regulation

- Cryptography *(you already know)*
- Steganography
- Proposals to regulate cryptography
- Technical limits of regulating cryptography
  - Secure digital signatures → Secure encryption
  - Key Escrow encryption without permanent surveillance → Encryption without Key Escrow
  - Symmetric authentication → Encryption
  - Multimedia communication → Steganography
  - Keys for communication and secret signature keys can be replaced at any time → Key Escrow to backup keys is nonsense
- Proposals to regulate cryptography harm the good guys only
Steganography
Steganography

Domain of trust → Embedding → Stegotext → Extracting → Domain of trust

Cover → emb → Secret message → Sender → Stegotext → Recipient → emb → Secret message

Area of attack
Steganography: Secrecy of secrecy

- exactly the same
- cannot be detected
- as much as possible

no changes
Steganography: Watermarking and Fingerprinting

- Correlation is enough
- Some 100 bits are enough
Proposals to regulate cryptography?

- Would you regulate cryptography to help fight crime?
- If so: How?
Proposals to regulate cryptography!

- Outlaw encryption
- Outlaw encryption – with the exception of small key lengths
- Outlaw encryption – with the exception of Key Escrow or Key Recovery systems
- Publish public encryption keys only within PKI if corresponding secret key is escrowed
- Obligation to hand over decryption key to law enforcement during legal investigation
Secure digital signatures —> Secure encryption

A does not need a certificate for $c_A$ issues by CA
Key Escrow encryption without permanent surveillance

\[ \text{\( k_{esc}(A,c_A) \)} \]

\[ \text{\( c_A(\text{secret message}) \)} \]

\[ \longrightarrow \text{ Encryption without Key Escrow} \]
Key Escrow encryption without permanent surveillance

\[ k_{esc}(A, c_A) \]

A

\[ k_{esc}(c_A(\text{secret message})) \]

B

employ Key Escrow additionally to keep your encryption without Key Escrow secret
Key Escrow encryption without permanent surveillance

A

B

\( k_{esc}(A, c_A) \)

\( k_{esc}(c_A(k_{AB}), k_{AB}(\text{secret message})) \)

hybrid encryption can be used
Key Escrow encryption without permanent surveillance

\[ k_{esc}(A,k_{AB}) \]

\[ k_{esc}(k_{AB}(\text{secret message})) \]

if surveillance is not done or even cannot be done retroactively, prejudgment: symmetric encryption alone does the job
Symmetric authentication → Encryption

**Sender** $A$

Kenn $k_{AB}$

Zu übertragen sei Nachricht $b_1, \ldots, b_n$ mit $b_i \in \{0, 1\}$

Berechnet

$MAC_1 := \text{code}(k_{AB}, b_1)$ ... $MAC_n := \text{code}(k_{AB}, b_n)$

Sei $a_1, \ldots, a_n$ die bitweise invertierte Nachricht.

Wählt zufällig $MAC'_1$ ... $MAC'_n$ mit

$MAC'_1 \circ \text{code}(k_{AB}, a_1)$ ... $MAC'_n \circ \text{code}(k_{AB}, a_n)$

Überträgt (die Mengenklammern bedeuten „zufällige Reihenfolge“)

{$(b_1, MAC_1), (a_1, MAC'_1)$} ... {$b_n, MAC_n), (a_n, MAC'_n)$}

**Empfänger** $B$

Kenn $k_{AB}$

falsely authenticated messages

form

intermingle

separate

Probiert, ob

{$MAC_1 = \text{code}(k_{AB}, b_1)$ oder $MAC'_1 = \text{code}(k_{AB}, a_1)$}

und empfängt den passenden Wert $b_1$

... probiert, ob

{$MAC_n = \text{code}(k_{AB}, b_n)$ oder $MAC'_n = \text{code}(k_{AB}, a_n)$}

und empfängt den passenden Wert $b_n$
Symmetric authentication → Encryption

**Sender A**

Kennt $k_{AB}$

Zu übertragen sei Nachricht $b_1, \ldots, b_n$ mit $b_i \in \{0, 1\}$

Berechnet

$\text{MAC}_1 := \text{code}(k_{AB}, b_1) \ldots \text{MAC}_n := \text{code}(k_{AB}, b_n)$

Überträgt

$(1, b_1, \text{MAC}_1), \ldots, (n, b_n, \text{MAC}_n)$

**Empfänger B**

Kennt $k_{AB}$

**Komplementgenerierer**

Hört die Nachricht $b_1, \ldots, b_n$ ab.

Bildet $a_1, \ldots, a_n$, die bitweise invertierte Nachricht.

Wählt zufällig $\text{MAC}'_1 \ldots \text{MAC}'_n$ und mischt in den Nachrichtenstrom von Sender A an die passenden Stellen

$(1, a_1, \text{MAC}'_1), \ldots, (n, a_n, \text{MAC}'_n)$

Überträgt die Mischung

**Abhörer**

kann $a_i$ und $b_i$ nicht unterscheiden

**falsely authenticated messages**

form and intermingle
without knowing the key

normal Authentikationsprotokoll
Ignoriert Nachrichten mit falscher Seque
Ignoriert Nachrichten mit falscher Authe
gibt die übrigbleibenden weiter
empfangen wird mit größter Wahrschein

$b_1, \ldots, b_n$
Exchanging keys outside the communication network is easy for **small closed groups**, in particular it is easy for criminals and terrorists.

**Large open groups** need a method of key exchange which works without transmitting suspicious messages within the communication network – asymmetric encryption cannot be used directly for key exchange.

Solution:

**Diffie-Hellman Public-Key Agreement**

Uses public keys of a commonly used digital signature systems (DSS, developed and standardized by NSA and NIST, USA)
Key exchange without message exchange

Diffie-Hellman Public-Key Agreement

secret: \( x \) \( y \)

public: \( g^x \) \( g^y \)

\((g^y)^x = g^{yx} = g^{xy} = (g^x)^y\)
Key exchange for steganography!

Diffie-Hellman Public-Key Agreement

secret: \( x \) \hspace{1cm} y

public: \( g^x \) \hspace{1cm} \( g^y \)

\((g^y)^x = g^{yx} = g^{xy} = (g^x)^y\)

\(f(C, g^{yx}) = f(S, g^{xy})\)
Summary

Cryptoregulation ignores technical constraints

<table>
<thead>
<tr>
<th>Digital Signatures</th>
<th>Encryption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key Escrow without permanent surveillance</td>
<td>Key exchange, multiple encryption</td>
</tr>
<tr>
<td>Multimedia communication</td>
<td>Steganography</td>
</tr>
</tbody>
</table>
Exchanging new keys is more efficient and more secure than Key Recovery —> Key Recovery for communication is nonsense

Communication

Encryption: generate new one(s) and exchange

Authenticate/encrypt and transmit message(s) once more

Authentication: generate new one(s) and exchange using CA

Dig. Signature: already generated digital signatures can still be tested; generate new key-pair for new digital signatures and, if you like, let certify your new public key

Long-term storage

Symmetric Authentication

Encryption

Key Recovery makes sense
<table>
<thead>
<tr>
<th>Encryption</th>
<th>protecting communication</th>
<th>protecting long-term storage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Encryption</strong></td>
<td><strong>Key Recovery</strong></td>
<td><strong>Key Recovery</strong></td>
</tr>
<tr>
<td>symmetric (MACs)</td>
<td>functionally unnecessary, but additional security risk</td>
<td>useful</td>
</tr>
<tr>
<td>asymmetric (dig. signature)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Proposals to regulate cryptography harm the good guys only

- Outlaw encryption
- Outlaw encryption – with the exception of small key lengths
- Outlaw encryption – with the exception of Key Escrow or Key Recovery systems
- Publish public encryption keys only within PKI if corresponding secret key is escrowed
- Obligation to hand over decryption key to law enforcement during legal investigation

- Steganography
- In addition steganography
- Use Key Escrow or Key Recovery system for bootstrap
- Run PKI for your public encryption keys yourself
- Calculate one-time-pad accordingly
(Im-)Possibility to regulate anonymous/pseudonymous communication

- Explicit techniques *(you already know the theory)*
- Workarounds
(Im-)Possibility to regulate anonymous/pseudonymous communication

Anon-Proxy

MiXes
  Cascade: AN.ON
  P2P: TOR

All this exists abroad without regulation – as long as we do not have a global home policy
(Im-)Possibility to regulate anonymous/pseudonymous communication

But even domestic:

- Public phones,
- Prepaid phones,
- Open unprotected WLANs,
- Insecure Bluetooth mobile phones,
- ...

Data retention is nearly nonsense,

Since „criminals“ will use workarounds, cf. above
• 14.7. Martin Übung
• 16.7. Benjamin Kellerman „dudle“ – privacy preserving meeting scheduling based on DC-net ideas
• 21.7. Computation on encrypted data
• 23.7 Stefanie: “freenet – a privacy-preserving P2P system“
Group Signatures
(Chaum, van Heyst 1991)

• Idea: digital signature on behalf of a group without revealing which group member did sign

• Setting:
  – Group Manager (can be distributed):
    • generates group key pair
    • join / leave of group members
    • revoke anonymity of group members
  – Join:
    • member learns his private key for signing
  – Leave:
    • private key of the member is revoked
  – Signing:
    • every member of group
  – Verification:
    • everybody with the help of the group public key
Properties of a Group Signature Scheme

• Soundness and Completeness
  – valid signatures always verify correctly
  – invalid signatures always fail verification.

• Unforgeable
  – only group members can create valid signatures

• Anonymity
  – given a message and its signature, the signing group member cannot be determined without the group manager's private key

• Traceability
  – group manager can trace which group member issued a signature

• Unlinkability
  – given two messages and their signatures, only group manager can tell if the signatures were from the same signer or not
Properties of a Group Signature Scheme

- **No Framing**
  - colluding group members (and manager) cannot forge a signature of a non-participating group member

- **Unforgeable tracing verification**
  - group manager cannot falsely accuse a signer of creating a signature he did not create

- **Coalition resistance**
  - colluding group members cannot generate a signature that the group manager cannot trace to one of the colluding group members
Zero Knowledge Proof of Knowledge (ZKP)