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/
Aims of Teaching at Universities

Science shall clarify

*How something is.*

But additionally, and even more important

*Why it is such*

or

*How could it be*

(and sometimes, *how should it be*).

“*Eternal truths*” (i.e., knowledge of long-lasting relevance) should make up more than 90% of the teaching and learning effort at universities.
1. Education to **honesty** and a **realistic self-assessment**

2. Encouraging realistic **assessment of others**, e.g., other persons, companies, organizations

3. Ability to gather **security and data protection requirements**
   - Realistic protection goals
   - Realistic attacker models / trust models
Realistic protection goals/attacker models: Technical solution possible?
General Aims of Education in IT-security (sorted by priorities)

1. Education to honesty and a realistic self-assessment
2. Encouraging realistic assessment of others, e.g., other persons, companies, organizations
3. Ability to gather security and data protection requirements
   - Realistic protection goals
   - Realistic attacker models / trust models
4. Validation and verification, including their practical and theoretical limits
5. Security and data protection mechanisms
   - Know and understand as well as
   - Being able to develop

In short: Honest IT security experts with their own opinion and personal strength.
1. Education to **honesty** and a **realistic self-assessment**

As teacher, you should make clear
- your strengths and weaknesses as well as
- your limits.

**Oral examinations:**
- Wrong answers are much worse than “I do not know”.
- Possibility to explicitly exclude some topics at the very start of the examination (if less than 25% of each course, no downgrading of the mark given).
- Offer to start with a favourite topic of the examined person.
- Examining into depth until knowledge ends – be it of the examiner or of the examined person.
General Aims of Education in IT-security

How to achieve?

1. Education to **honesty** and a **realistic self-assessment**
2. Encouraging realistic **assessment of others**, e.g., other persons, companies, organizations

Tell, discuss, and evaluate case examples and anecdotes taken from first hand experience.
1. Education to **honesty** and a **realistic self-assessment**
2. Encouraging realistic **assessment of others**, e.g., other persons, companies, organizations
3. Ability to gather **security and data protection requirements**
   - Realistic protection goals
   - Realistic attacker models / trust models

**Tell, discuss, and evaluate case examples (and anecdotes) taken from first hand experience.**

Students should develop scenarios and discuss them with each other.
1. Education to **honesty** and a **realistic self-assessment**

2. Encouraging realistic **assessment of others**, e.g., other persons, companies, organizations

3. Ability to gather **security and data protection requirements**
   - Realistic protection goals
   - Realistic attacker models / trust models

4. **Validation** and **verification**, including their practical and theoretical **limits**

   Work on case examples and discuss them.

**Anecdotes!**
1. Education to **honesty** and a **realistic self-assessment**
2. Encouraging realistic **assessment of others**, e.g., other persons, companies, organizations
3. Ability to gather **security and data protection requirements**
   - Realistic protection goals
   - Realistic attacker models / trust models
4. **Validation** and **verification**, including their practical and theoretical **limits**
5. Security and data protection **mechanisms**
   - Know and understand as well as
   - Being able to develop

**Whatever students can discover by themselves in exercises should not be taught in lectures.**
...but no this way!

First stupid and silly now wise as Goethe this has accomplished the power of the Nuremberg Funnel

Nuremberg Funnel (German: Nürnberger Trichter)
Postcard from around 1940
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
Protection Goals: Definitions

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.

⇒ *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

- Availability

Accountability

- Reachability

Legal Enforceability

implies

+ strengthens

- weakens
Observability of users in switched networks

countermeasure encryption

- link encryption

- interceptor
- possible attackers
- (telephone) exchange
  - operator
  - manufacturer (Trojan horse)
  - employee

radio

television

videophone

phone

internet
Observability of users in switched networks

- countermeasure encryption
  - end-to-end encryption

Possible attackers:
- operator
- manufacturer (Trojan horse)
- employee
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.

countermeasure encryption
• link encryption
• end-to-end encryption

radio

television

videophone

phone

internet

network termination

telephone exchange
• operator
• manufacturer (Trojan horse)
• employee

interceptor
possible attackers

communication partner

data on interests: Who? What?
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 2018: 1 hard drive disk with 500 GByte for ≈ 22 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
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
Anonymous plain old letter post is substituted by „surveillanceable“ e-Mails

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/]
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**
Notions of Privacy: Contextual Integrity

- 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
Degress of Anonymity


• exemplified with sender anonymity:
  – absolute anonymity: unobservability, “no observable effects”
  – beyond suspicion: no more likely than any other potential sender
  – probable innocence: no more likely to be sender than not to be sender
  – possible innocence: nontrivial probability that 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
Attacker (-model)

Questions:

• How widely distributed? (stations, lines)
• observing / modifying?
• How much computing capacity? (computationally unrestricted, computationally restricted)
Realistic protection goals/attacker models: Technical solution possible?
Facebook helps you connect and share with the people in your life.

Create an account
It's free and always will be.

First name
Surname
Mobile number or email address
Re-enter mobile number or email address
New password

Birthday

Day  Month  Year

Why do I need to provide my date of birth?

Female  Male

By clicking Create an account, you agree to our Terms and that you have read our Data Policy, including our Cookie Use.

Create an account
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**  
(if possible: switch channels)

**explicit addresses:**  
- routing

**implicit addresses:**  
- attribute for the station of the addressee

- invisible $\iff$ encryption system
- visible

<table>
<thead>
<tr>
<th></th>
<th>public address</th>
<th>private address</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>implicit address</strong></td>
<td>very costly, but necessary to establish contact</td>
<td>costly</td>
</tr>
<tr>
<td><strong>visible</strong></td>
<td>should not be used</td>
<td>change after use</td>
</tr>
</tbody>
</table>
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 overlay structure
  - store-and-forward like
  - pull-based
Equivalence of Encryption Systems and Implicit Addressing

invisible public address  \iff\  asymmetric encryption system

invisible private address  \iff\  symmetric encryption system
Broadcast vs. Queries

- **Broadcast of separate messages to all recipients**
  - broadcaster
    - message 1
    - message 2
    - message 3
    - message 4
    - ...
  - everybody can query all messages
  - everybody can query all messages

- **Message service**
  - message service
    - message 1
    - message 2
    - message 3
    - message 4
    - ...
  - everybody can query all messages
Example for message service

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

Memory cells

Message service

51

Message 1
Message 2
Message 3
Message 4

Memory cells

Message service

3 servers used for superposed querying

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:

\[ \begin{align*}
!x &= \text{message 1 XOR message 4} & \text{XOR pad}_x \\
!y &= \text{message 1 XOR message 2} & \text{XOR pad}_y \\
!z &= \text{message 2 XOR message 3 XOR message 4} & \text{XOR message 4} \\
\end{align*} \]

From this follows by local superposition of the pads

\[ !x \text{ XOR } !y \text{ XOR } !z = \text{message 3} \text{ XOR message 2} \]

(Equal to the sum of the wanted (**) memory cells)

Generated by servers themselves when starting circulation

Pseudo random short

User

Invert bit of the memory cell of interest

Query vectors

Query multiple memory cells

David A. Cooper, Kenneth P. Birman 1995
Efficiency improvements: A. Pfitzmann 2001
**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**

- $c_{S1}(1011)$
- $c_{S2}(0110)$
- $c_{S3}(1001)$
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

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

3 servers used for superposed querying

response of the message service:

\[ !x = \text{message 1 XOR message 4} \]
\[ !y = \text{message 1 XOR message 2} \]
\[ !z = \text{message 2 XOR message 3 XOR message 4} \]

from this follows by local superposition of the pads

\[ !x \text{ XOR } !y \text{ XOR } !z \Rightarrow \text{message 3 XOR message 2} \]

(equal to the sum of the wanted (**) memory cells)

query vectors
query multiple memory cells
WIEVIEL DEUTSCHPRACHIGE MUTTERSPRACHLER?
re-writable memory cell = implicit address
re-writing = addition mod 2 (enables to read many cells in one step)
canals 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 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.
Invisible implicit addresses using “query and superpose” (1)

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 $\sum_s PBG_s(t)$ (possible in one step.)
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
Proof of anonymity for a RING access method

Flow of the message frame around the ring

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.

alternatives: 123... n+1

Digital signal regeneration:
A. Pfitzmann 1983 - 1985
• 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

Crowds (Reiter, Rubin, 1998)
Link encrypted communication between two adjoining GNUnet users

Indirecting of a request (sender address will be rewritten)

Forwarding of a request (original sender address is preserved)

Response to user according to the given sender address

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

User A

User B

User C

User D

User E

User F

User G

User H

encrypted block

B_{\text{enc}} = E_{h(B)}(B)
Buses...

- Amos Beimel, Shlompi 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
   \( \Rightarrow \) 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)
• A wants to send some message $m$ to $D$
• depicted is one seat of the bus
(Universal) Re-encryption


• Re-encryption:
  – given: public key \( e \), \( c = \text{Enc}(e,m) \)
  – create: \( c' = \text{Enc}(e,m) \) with \( c' \neq c \)

• Universal Re-encryption:
  – Re-encryption without knowing \( e \)
    \( \Rightarrow \) avoids linkability (same recipient…)

• Implementation:
  – Recall ElGamal:
    • \( e = g^x \)
    • \( \text{Enc}(m) = (g^y, m \cdot e^y) \)
    • Homomorphic property: \( \text{Enc}(m_1) \cdot \text{Enc}(m_2) = \text{Enc}(m_1 \cdot m_2) \)
  – Re-encryption:
    • \( \text{Enc}(m)^z = (g^y \cdot g^z, m \cdot e^y \cdot e^z) = (g^{y+z}, m \cdot e^{y+z}) = (g^{y'}, m \cdot e^{y'}) \)
  – Universal Re-encryption:
    • Idea: \( \text{Enc}(m) = [ \text{Enc}(m), \text{Enc}(1) ] = [ (g^y, m \cdot e^y), (g^{y'}, e^{y'}) ] \)
    • \( \text{Enc}(m)^{z,z'} = [ \text{Enc}(m) \cdot \text{Enc}(1)^z, \text{Enc}(1)^{z'} ] = [ (g^{y+y' \cdot z}, m \cdot e^{y+y' \cdot z}), (g^{y' \cdot z'}, e^{y' \cdot z'}) ] = [ (g^{y''}, m \cdot e^{y''}), (g^{y'''}, e^{y'''}) ] \)
Proxy Re-encryption:
- given: \( c = \text{Enc}(e, m) \), \( e' \)
- create: \( c' = \text{Enc}(e', m) \)

\( \Rightarrow \) Will not reveal plaintext \( m \)

Threshold Proxy Re-encryption:
- Proxy is distributed among \( n \) entities
- \( k \) of \( n \) are necessary for re-encryption
- Use case: plaintext \( m \) can only be read by the holder of \( e' \), iff at least \( k \) entities “agree”
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

- 4 seats → one per recipient of D
- 4 seats → one per sender of D
- ? seats → e.g. shortest path B to E not unique

- tradeoff: time vs. communication complexity
  → spanning subgraph sufficient
Buses – time and communication tradeoff

- 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
Fault tolerance of the RING-network

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
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.
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

A sends 00101000

B sends 01000100

C sends 01011001

Sum = True Message from A 00110101

Key Graph

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

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

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

Note: In this example “sum” means XOR

Chaum, 1988
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:
- dependent on each other

Transmission topology:
Reservation scheme

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

only different to “1” if “+” ≠ “⊕”

≥ one round-trip delay

time

reservation frame

message frame
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.

$\Rightarrow$ 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:

$\leq 2^S - 1$ station

addition $\text{mod } 2^L$

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

overflow area for addition of messages

overflow area for addition of counters
Pairwise superposed receiving

\[
S_1 \quad \text{(X+Y)-X = Y} \quad S_2 \quad \text{(X+Y)-Y = X}
\]

without superposed receiving

with pairwise superposed receiving
Collision resolution algorithm with mean calculation and superposed receiving
Global superposed receiving (2 messages equal)

Collision resolution algorithm with mean calculation and superposed receiving

<table>
<thead>
<tr>
<th>S₁</th>
<th>7</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>S₂</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>S₃</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>S₄</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S₅</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

≥ one round-trip delay

∅ = 6

∅ = 3

∅ = 4

∅ = 11

≤

≤

≤

≤

≤

≤
Analogy between Vernam cipher and superposed sending

Vernam cipher

K + M = C ⇔ M = C - K

abelian group

M₁ + K = O₁

M₂ - K = O₂
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 $m-1 \rightarrow m$
Proof of sender anonymity: induction step

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

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

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 $O_L + K'$. 

minimal connectedness: only connected by one key
Proof of sender anonymity: induction step

Theorems:
\[ O_m = M_m + K \]
\[ O_L + K = M_L - K + \ldots \]

minimal connectedness: only connected by one key

Attacker observes \( O_1, O_2, \ldots O_m \).

For each combination of messages \( M'_1, M'_2, \ldots M'_m \)
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

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

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 *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.

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

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!

1?
\[ s \in \mathbb{Z}^*_p \text{, randomly chosen} \]

(\text{so user cannot compute } e \text{ such that } s \equiv \alpha^e) \]

\[ x := s^{b \alpha^y} \mod p \quad \text{with } 0 \leq y \leq p-2 \]

commit \( x \)

open \( y \)

2?
Let \( 2^u \) be the smallest number that does not divide \( p - 1 \)

\[ y := y_1, b, y_2 \quad \text{with } 0 \leq y \leq p-2 \text{ and } |y_2| = u - 1 \]

\[ x := \alpha^{y} \mod p \]

commit \( x \)

open \( y \)
Blobs based on factoring assumption

1?

prover

\[ n := p \cdot q \]

\[ s := t^2 \mod n \]

\[ n, s \]

\[ s \in \text{QR}_n \]

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

commit

\[ x \]

open

\[ y \]

2?

prover

\[ n := p \cdot q \]

\[ s \in \text{QR}_n, \left(\frac{s}{n}\right) = 1 \]

\[ n, s \]

\[ n = p \cdot q, s \notin \text{QR}_n \]

commit

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

open

\[ x \]

\[ y \]
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.)

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

known = known to \textit{all} stations truthful to the protocol
Collisions in the reservation phase

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

Problem: Attacker A could await the output of the users truthful to the protocol and than 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
- Fault detection
- Error recovery of the PRGs, initialization of the access protocol

**F-mode**
- Sender and recipient are not protected
- Fault localization
- Taking defective components out of operation
Fault tolerance: sender-partitioned DC-network

1. Write and read access to the DC-network
2. Read access to the DC-network

widest possible spread of a fault of station 3

... of a fault of station 5
Protection of the communication relation: MIX-network

D. Chaum 1981 for electronic mail

\[ c_1(z_4, M_1) \rightarrow \text{MIX}_1 \text{ batches, discards repeats,} \]
\[ d_1(c_1(z_i, M_i)) = (z_i, M_i) \]

\[ c_2(z_3, M_3) \rightarrow \text{MIX}_2 \text{ batches, discards repeats,} \]
\[ d_2(c_2(z_i, M_i)) = (z_i, M_i) \]
The Mix protocol

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

\[ c_1 (z_4, z_1, M_1) \]
\[ c_1 (z_5, z_2, M_2) \]
\[ c_1 (z_6, z_3, M_3) \]

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

\[ c_2 (z_3, M_3) \]
\[ c_2 (z_1, M_1) \]
\[ c_2 (z_2, M_2) \]

MIX\_2 batches, discards repeats,
\[ d_2(c_2(z_i, M_i)) = (z_i, M_i) \]
Basic functions of a MIX

input messages

MIX

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
Properties of MIXes

**MIXes** should be designed, produced, operated, and maintained independently.

Messages of the same length:
- Buffer
- Re-encrypt
- Change order

Batch-wise:

Each message processed only once:
- Inside each batch
- Between the batches

Symmetric encryption system only for:
- First
- Last

MIX

Asymmetric 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
Maximal protection

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

\[ S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_4 \rightarrow S_5 \rightarrow S_6 \rightarrow \text{Mix 1} \rightarrow \text{Mix 2} \rightarrow \text{Mix 3} \]
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
Mix 2a
Mix 3a
Mix 1b
Mix 2b
Mix 3b
Mix 1c
Mix 2c
Mix 3c

S_1
S_2
S_3
S_4
S_5
S_6
3 honest Mixes / Anonymity Set Size: 2
Re-encryption scheme for sender anonymity

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

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

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

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

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

\[ I_1 = k_0(l) \]

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

message header \( H \)

message content \( I \)

encryption

decryption

observable transfer

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

For sender anonymity

Message header

For recipient anonymity

Message content

3rd party, to hold the anonymous return addresses for anonymous query

delivery using sender anonymity scheme

Encryption

Decryption

Unobservable transfer
Indirect re-encryption scheme maintaining message length

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

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

$M_j$

$M_{j+1}$

blocks with message contents

blocks with random contents

$H_j$

$H_{j+1}$

$H_{j+2}$

blocks with message contents

blocks with random contents

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

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

decrypt with $d_j$

re-encrypt with $k_j$
Minimally message expanding re-encryption scheme maintaining message length

\[
M_j \quad H_j \quad M_{j+1} \quad H_{j+1}
\]

- \( M_j \): Message at time \( j \)
- \( H_j \): Hash of \( M_j \)
- \( M_{j+1} \): Message at time \( j+1 \)
- \( H_{j+1} \): Hash of \( M_{j+1} \)
- \( k_j, A_{j+1}, C_j \):
  - \( k_j \): Encryption key
  - \( A_{j+1} \):
    - \( b_j \):
      - Random contents
    - \( b \):
      - Message contents
  - \( C_j \):
    - \( b \):
      - Random contents
    - \( b_{j-n_j} \):
      - Random contents

- Decrypt with \( d_j \)
- Re-encrypt with \( k_j \)

- If \( k^{-1}(k(M)) = M \)
- And \( k(k^{-1}(M)) = M \)
Recall: Diffie-Hellman key agreement

**Key generation:**
\[ y \in \mathbb{Z}_p^* \]
\[ gy \mod p \]

Calculating shared key:
\[ (gx)^y \mod p \]

**Random number 1**
\[ x \in \mathbb{Z}_p^* \]
\[ gx \mod p \]

**Domain of trust**

**Calculating shared key:**
\[ (gy)^x \mod p \]

**Secret area**

**Calculating shared key:**
\[ (g^x)^y \mod p \]

**Domain of trust**

**Publicly known:**
\[ p \text{ and } g \in \mathbb{Z}_p^* \]
\[ p, g \]

**Random number 2**

**Area of attack**

Calculated keys are equal, because:
\[ (g^y)^x = g^{yx} = g^{xy} = (g^x)^y \mod p \]
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 $M_i$: $k_i = y_{Mi}^{x_i} \mod p$
  – 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 \( y \)
      \( \Rightarrow \) linkability!
  
  – Solution:
    • calculate \( y_{i+1} \) from \( y_i \)
Mix Packets based on Diffie-Hellman Key Agreement

– Solution:

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

$y_{i+1} = g^{x_{i+1}} \mod p$
$= g^{x_i b_i} \mod p$
$= y_i^{b_i} \mod p$

⇒ mix $M_i$ can calculate $y_{i+1}$ from $y_i$!
⇒ **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:

\[(z,M)^c \quad |z|=b \quad |M|=B\]

\[\text{attacker multiplies } M \text{ with factor } f \text{ and compares}\]

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

\[2^b \cdot 2^B \leq n-1 \text{ hold always and normally } b \ll B.\]

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

\[(z,M) = z \cdot 2^B + M \quad \text{ and} \]

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

Let the identifiers $z'$ and $M'$ be defined by

\[(z,M)\cdot f \equiv z'\cdot 2^B + M' \Rightarrow \]

\[z\cdot 2^B\cdot f + M\cdot f \equiv z'\cdot 2^B + M' \Rightarrow \]

\[2^B\cdot (z\cdot f - z') \equiv M' - M\cdot f \Rightarrow \]

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

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

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

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
<table>
<thead>
<tr>
<th>Complexity of the basic methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>unobservability of neighboring lines and stations as well as digital signal regeneration</td>
</tr>
<tr>
<td>unobservability of neighboring lines and stations as well as digital signal regeneration</td>
</tr>
<tr>
<td>attacker model</td>
</tr>
<tr>
<td>RING-network</td>
</tr>
<tr>
<td>DC-network</td>
</tr>
<tr>
<td>MIX-network</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>expense per user</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

$n = \text{number of users}$

$k = \text{connectedness key graph of DC-networks respectively number of MIXes}$
Encryption in layer models

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)

OSI layers

7 application
6 presentation
5 session
4 transport
3 network
2 data link
1 physical
0 medium

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>addressing</td>
<td>addressing</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
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
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
Time-slice channels (1)

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

$S_0$

\[
\begin{align*}
\bullet & \quad \bullet & \quad \bullet \\
\text{TS-set up: } x & \quad & \text{TR-set up: } x
\end{align*}
\]

$\text{call request: } c_G(k, \text{ s}_R, \text{ and s}_G)$

$S_1$

\[
\begin{align*}
\text{TS-set up: PBG(s}_G,1) & \quad & \text{TR-set up: PBG(s}_R,1) \\
\text{TR-set up: PBG(s}_R,1) & \quad & \text{TR-set up: PBG(s}_G,1)
\end{align*}
\]

query and superpose

instead of broadcast
Time-slice channels (2)

This setup of receiving channels is a very flexible scheme for recipient anonymity.
• 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 establishes circuit
⑤ tells rendezvous point
⑥ establishes circuit

Directory Service

Introduction Point

Rendezvous Point

Hidden Service
Connection configuration later (1)

Station $R$  MIXes($R$)  LE($R$)  LE($G$)  MIXes($G$)  Station $G$

$S_0$

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

$S_1$

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

Call request: $c_G(k, sR, and sG)$

From $P$ to $P$

TS-setup: PBG($sQ,0$)
TR-setup: PBG($sQ,1$)
Connection configuration later (2)

$S_2$

- TS-setup: $\text{PBG}(s_G,2)$
- TR-setup: $\text{PBG}(s_R,2)$

- $\text{PBG}(s_R,1)$

- throw away

- replace

$S_{t-1}$

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

$S_t$

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

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

- $k(\text{dial tone, data})$
Query and superpose to receive the call requests

Query and superpose:

- Each station has to query in each time slice (else the anonymity set degenerates)
- Each station should inquiry 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.
2. Filter
3. Generation of visible implicit addresses
4. Restrict the region
5. 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 keep options consider early enough.

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
Payment system is **secure** if

- user can transfer the rights received,
- user can lose 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
Pseudonyms

- Person pseudonyms
  - Public person pseudonym
    - Phone number
  - Non-public person pseudonym
    - Account number
  - Anonymous-person pseudonym
    - Biometric, DNA (as long as no register)
  - Business-relationship pseudonym
    - Pen name
  - Transaction pseudonym
    - One-time password

Role pseudonyms

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 $M$:  
$s_A(M)$

$X \rightarrow M, s_A(M) \rightarrow Y$

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

**graphical notation**

sender $X$

recipient $Y$

document $M$
Authenticated anonymous declarations between business partners that can be de-anonymized

trusted third party A

identification

trusted third party B

identification

user X

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)
Authenticated anonymous declarations between business partners that can be de-anonymized

trusted third party $A$

trustees for identities

trusted third party $B$

identification

user $X$

confirmation

know

document

for $p_G(X,g)$

$p_G(X,g)$

$p_A$

identification

user $Y$

confirmation

know

document

for $p_G(Y,g)$

$p_G(Y,g)$

$p_B$

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
   \[ p_l(Y,g) \]

4. Delivery to customer
   checked by \( T \)

5. Money

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)
   - \( p_T \)

3. **delivery to trustee**
   - \( p_L(Y, g) \)

4. **delivery to customer**
   - checked by \( p_T \)

5. **money**
   - \( p_T \)

\( p_K(X, g) \)

\( p_T \)

\( p_T \)

\( p_T \)

\( p_T \)
Security for completely anonymous business partners using active trustee who can (not) check the goods

trustee for values

[1] order
delivery is $p_L(Y,g)$ + „money“ for distributor

$\rho_X(X,g)$

customer $X$

[4.1] wait
[4] delivery to customer

delivery is $p_L(Y,g)$

$\rho_T$ checked by $T$

order of the customer (money is deposited)

[2] $\rho_T$

[3] delivery to trustee

$\rho_L(Y,g)$

[5] money

$\rho_T$

$\rho_T$

$\rho_T$

merchant $Y$
Anonymously transferable standard values

current owner:
  digital pseudonym

value number: $v_n$

former owners

  digital pseudonym 1, transfer order 1
  digital pseudonym 2, transfer order 2
  digital pseudonym 3, transfer order 3

.....

Anonymously transferable standard value
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 $z$: \( \text{Hash}(\text{Transaction}, \text{Timestamp}, z) = 00000\ldots (0|1)^* < w \)
  - $w$ adjusted over timer

https://www.blockchain.info
Basic scheme of a secure and anonymous digital payment system

1. Choice of pseudonyms
   - $p_Z(X,t)$ chooses a pseudonym $p_E(Y,t)$, which is similar to $p_Z(Y,t)$.

2. Transfer order of the payer
   - $p_Z(X,t)$ transfers the right to $p_E(Y,t)$.

3. Authentication by the witness
   - $p_E(Y,t)$, the witness, confirms ownership of $p_E(Y,t)$.

4. Receipt for the payer
   - $p_E(Y,t)$ confirms the transfer of the right to $p_E(Y,t)$.

5. Authentication for the recipient
   - $p_E(Y,t)$ confirms the transfer of the right to $p_Z(X,t)$.

The process ensures secure and anonymous transactions.
Transformation of the authentication by the witness

[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
\[ p_Z^B(X,t) \text{ owns the right} \]
\[ \text{transfer the right to } p_E^B(Y,t) \]

[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
\[ p_Z(X,t) \text{ have got the right from } p_Z(X,t) \]

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

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

[1] payer Y

[2] EUR 10

[3] EUR 10


recipient Z

witness B

pB

pB

pB

pB

pB

pB

pB
Transformation of the authentication by the witness

[payer X]

- authentication of ownership
  - $p_{ZB}(X,t)$ owns the right
- transfer order of the payer
  - transfer the right to $p_{E}(Y,t)$
- choice of pseudonyms
  - $p_{E}(Y,t) \approx p_{E}(Y,t)$
  - $p_{Z}(X,t) \approx p_{Z}(X,t)$

[witness B]

- authentication by the witness
  - $p_{E}(Y,t)$ owns the right, got from $p_{Z}(X,t)$

[recipient Y]

- receipt for the payer
  - have got the right from $p_{Z}(X,t)$
- authentication for the recipient
  - have transferred the right to $p_{E}(Y,t)$
- choice of pseudonyms
  - $p_{ZB}(Y,t+1)$ owns the right
- transfer of the right
  - $p_{ZB}(Y,t+1)$ has transferred the right to $p_{E}(Y,t)$

[pB]

- $p_{B}$

- $p_{B}$

- $p_{B}$
The next round: Y in the role payer to recipient Z

[0] $p_B^Z(Y, t+1)$ owns the right

[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
$p_B^Z(Y, t+1)$ owns the right to $p_E^B(Z, t+1)$

[3] authentication by the witness
$p_E^B(Z,t+1)$ owns the right, got from $p_Z^B(Y, t+1)$

[4] receipt for the payer
have got the right from $p_Z(Y, t+1)$.

[5] authentication for the recipient
have transferred the right to $p_E(Z, t+1)$.
Signature system for signing blindly

Key generation

Key for testing of signature, publicly known

Random number

Key for signing, kept secret

Text

Random number

Text with signature and test result

"Pass" or "fail"

Blinded text

Blinded text with signature

Unblind and test
RSA as digital signature system
with collision-resistant hash function \( h \)

### Key Generation

- **\( p, q \)** prime numbers
- \( n := p \cdot q \)
- \( t \) with \( \gcd(t, (p-1)(q-1)) = 1 \)
- \( s \equiv t^{-1} \mod (p-1)(q-1) \)

### Keys

- **\( t, n \)** key for testing of signature, publicly known
- **\( s, n \)** key for signing, kept secret

### Test

- **\( x, (h(x))^s \mod n \)**
  - "pass" or "fail"
  - \( h(1. \text{ comp.}) \equiv (2. \text{ comp.})^t \mod n \)

### Signing

- **\( x, (h(x))^s \mod n \)**
- \( (h(\bullet))^s \mod n \)

- **\( x \)** text
- **\( \text{test:} \)**
- **\( \text{signing:} \)**

### Security Parameter

- \( \text{random number} \)
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. Authentication of ownership:
   - \( p_{Z}^{B}(X,t) \) owns the right
   - Transfer order of the payer
   - Transfer the right to \( p_{E}^{B}(Y,t) \)

2. Transfer order of the payer:
   - \( p_{Z}(X,t) \approx p_{Z}^{B}(X,t) \)
   - \( p_{Z}(X,t) \approx p_{Z}(B,X,t) \)

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:
   - \( p_{E}(Y,t), p_{E}^{B}(Y,t) \)
   - \( p_{Z}(X,t) \)

5. Authentication for the recipient:
   - \( p_{E}(Y,t), p_{E}^{B}(Y,t) \)
   - \( p_{Z}(X,t) \)
   - \( p_{Z}(X,t) \approx p_{Z}(B,X,t) \)
   - \( p_{Z}(X,t) \approx p_{Z}(B,X,t) \)

payer X

recipient Y

witness B as secure device
Secure device: 2nd 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
$p_Z^B(X,t)$

[3] authentication by the witness
$p_E^B(Y,t)$

[4] receipt for the payer
$p_E(Y,t)$

[5] authentication for the recipient
$p_E(Y,t)$

pZ^B(X,t)

sym. encryption system suffices
Offline payment system

Payment systems with security by Deanonymizability

\[ k \quad \text{security parameter} \]

\[ l \quad \text{identity of the entity giving out the banknote} \]

\[ r_i \quad \text{randomly chosen} \quad (1 \leq i \leq k) \]

\[ C \quad \text{commitment scheme with information theoretic secrecy} \]

blindly signed banknote:

\[ s_{\text{Bank}}(C(r_1), \ C(r_1 \oplus l), \ C(r_2), \ C(r_2 \oplus l), \ldots, \ C(r_k), \ C(r_k \oplus l)), \]

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 \( (\exists i : \text{bank gets to know } r_i \text{ and } r_i \oplus i) \geq 1-e^{-c \cdot k} \)

(original owner identifiable)
Secure and anonymous digit. payment system with accounts

[pK(X) → pE(Y,t)]

[pE(Y,t) ≈ pE^B(Y,t)]

[pZ(X,t) ≈ pZ^B(X,t)]

[pZ(X,t)]

[pZ^B(X,t)]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]

[pB]
Personal identifier

845 authorizes A:

A notifies 845:

845 pays B €

B certifies 845:

C pays 845 €
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

![Driving license image](image)
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.
Anonymous Credentials
More complete View

Inspector can deanonymise

Taken from EU project ABC4Trust [https://abc4trust.eu/download/Deliverable_D2.2.pdf]
Usage of Anonymous Credentials

User A

User B

User X

Credentials issuing Organisation

Service provider

User A has driving-license

User A has driving-license

User B has driving-license

User X has driving-license
Data Publishing – Use-Case

Collection → Anonymization → Publishing...

Technische Universität Dresden

Privacy and Security
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>Sensitive</th>
<th>Non-sensitive</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZIP</td>
<td>Age</td>
<td>Sex</td>
</tr>
<tr>
<td>47677</td>
<td>43</td>
<td>Male</td>
</tr>
<tr>
<td>47602</td>
<td>22</td>
<td>Female</td>
</tr>
<tr>
<td>47678</td>
<td>45</td>
<td>Female</td>
</tr>
<tr>
<td>47905</td>
<td>31</td>
<td>Male</td>
</tr>
<tr>
<td>47909</td>
<td>36</td>
<td>Male</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>ZIP</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>
Data Publishing – Anonymization ($k$-Anonymity)

- **Groups of $k$ records** $\Rightarrow$ 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
Goldwasser and Micali (1982)

Nothing is learned about the plaintext from the ciphertext

- Anything known about the plaintext after seeing the ciphertext was known before seeing the ciphertext
- Encryption of either “dog” or “cat”: ciphertext leaks no further information about which has been encrypted
Absolute Privacy (Dalenius 1977)

- Access to a statistical database should not enable one to learn anything about an individual that could not be learned without access.

Proven to be impossible to achieve.

(Dwork 2006)
Absolut Privacy Problem

Impossibility result (Dwork 2006) on Absolute Privacy (Dalenius 1977)

Problem: Auxiliary Information and Utility of Database

Example:

- Knowing the height of a person is a privacy breach
- Auxiliary Information: “Terry Gross is two inches shorter than the average Lithuanian woman”
- Database: Reveals average heights of women of different nationalities

Semantic Security:

- Ciphertext does not reveal any information (no average height)
If there exists no Semantic Security equivalence for Privacy is everything lost?
Differential Privacy (Dwork 2006)

• Bounds privacy leakage for participating in a database

Definition

A randomized function \( K \) gives \( \epsilon \)-differential privacy if for all data sets \( D_1 \) and \( D_2 \) differing on at most one element, and all \( S \subseteq \text{Range}(K) \),

\[
Pr[K(D_1) \in S] \leq e^\epsilon \cdot Pr[K(D_2) \in S]
\]
Differential Privacy – Parameter

$$Pr[K(D_1) \in S] \leq \exp(\epsilon) \cdot Pr[K(D_2) \in S]$$

Difference between participating in a database or not:
- For large $\epsilon$ the output of $K(\cdot)$ can vary a lot
- For small $\epsilon$ the output of $K(\cdot)$ can only vary slightly

Small $\epsilon$:
- Higher privacy, lower utility

Large $\epsilon$:
- Lower privacy, higher utility
Differential Privacy – Context

\[ Pr[K(D_1) \in S] \leq \exp(\epsilon) \cdot Pr[K(D_2) \in S] \]

**NOT** a property of a dataset, but of a mechanism \( K() \)
- \( K() \) must introduce some randomness (add noise)
- Not sufficient: Sampling, Generalization, Suppression
- Often used: Perturbation, Randomized Response
PINQ – Privacy INtegrated Queries (MS Research 2009)
Differential Privacy – Non-Interactive Setting

Releasing a sanitized version of a database:

- Perturbed Histogram
- In general: statistics about database

Typical approach:
Calculate statistic then add noise.
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
   - TRUSTED
     - Trusted system
     - Encrypted data
   - PUBLIC
     - Cloud processing

2. Crypted Processing
   - Cloud
     - Processed data

3. Decrypting Result
   - TRUSTED
     - Decrypted data
   - PUBLIC
     - Output
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

$E(a) \ast E(b) = E(a+b)$

Computation e.g. in the Cloud
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 $\rightarrow$ Secure encryption
  - Key Escrow encryption without permanent surveillance $\rightarrow$ Encryption without Key Escrow
  - Symmetric authentication $\rightarrow$ Encryption
  - Multimedia communication $\rightarrow$ Steganography
  - Keys for communication and secret signature keys can be replaced at any time $\rightarrow$ Key Escrow to backup keys is nonsense
- Proposals to regulate cryptography harm the good guys only
Steganography

- **cover**
- **emb**
- **secret message**
- **sender**
- **attacker**
- **stegotext**
- **key**
- **extracting**
- **recipient**
- **cover***
- **emb**
- **secret message**
Steganography

- **Domain of trust**
- **Area of attack**
- **Domain of trust**

**Process Flow**

1. **Cover** input
2. **Embedding** (key, secret message)
3. Stegotext output
4. **Extracting** (key)
5. **Cover** output

**Roles**
- **Sender**
- **Recipient**
- **Attacker**
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 bit 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

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

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

\[ \rightarrow \] Encryption without Key Escrow
Key Escrow encryption without permanent surveillance

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

$k_{esc}(A, c_A)$

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

hybrid encryption can be used
Key Escrow encryption without permanent surveillance

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

**Sender A**

Kennt $k_{AB}$

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

Berechnet

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

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

Wählt zufällig MAC$'_1 \ldots$ MAC$'_n$ mit

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

Überträgt (die Mengenklammern bedeuten „zufällige Reihenfolge“)

$$\{(b_1, \text{MAC}_1), (a_1, \text{MAC}'_1)\} \ldots$$

$$\{(b_n, \text{MAC}_n), (a_n, \text{MAC}'_n)\}$$

**Empfänger B**

Kennt $k_{AB}$

falsely authenticated messages

form

intermingle

separate

Probiert, ob

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

und empfängt den passenden Wert $b_1$

... probiert, ob

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

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

normales Authentikationsprotokoll

Ignoriert Nachrichten mit falscher Sequenz

Ignoriert Nachrichten mit falscher Authentizität

gibt die übrigbleibenden weiter

empfangen wird mit größter Wahrscheinlichkeit

$k_{AB}$
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: \( x \quad y \)

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

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

Diffie-Hellman Public-Key Agreement

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

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

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

\[ f(C, g^{yx}) = f(S, g^{xy}) \]
Digital Signatures

Key Escrow without permanent surveillance

Multimedia communication

Encryption

Key exchange, multiple encryption

Steganography

Cryptoregulation ignores technical constraints
Exchanging new keys is more efficient and more secure than Key Recovery.

Key Recovery for communication is nonsense.

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
Key Recovery – for which keys?

<table>
<thead>
<tr>
<th>Encryption</th>
<th>Protecting communication</th>
<th>Protecting long-term storage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Key</td>
<td>Key</td>
</tr>
<tr>
<td></td>
<td>Recovery</td>
<td>Recovery</td>
</tr>
<tr>
<td></td>
<td>functionally unnecessary</td>
<td>useful</td>
</tr>
<tr>
<td></td>
<td>but additional security risk</td>
<td></td>
</tr>
</tbody>
</table>

- **Encryption**
  - Symmetric (MACs)
  - Asymmetric (dig. signature)
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-Proxies

MIXes
  Cascade: AN.ON
  P2P: TOR

All this exists abroad without regulation – as long as we do not have a global home policy
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