Security and Cryptography II

(Version 2024/04/11)

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/

Field of Specialization: Security and Privacy

Lectures	*Staff*	*SWS*
Network Security	Tschorsch	2/2
Peer-to-Peer Systems	Tschorsch	2/2
Security and Cryptography I, II	**Köpsell**	**2/2**
Application Security	**Köpsell**	**2/0**
Cryptography and -analysis	**Franz**	**2/1**
Information & Coding Theory	**Franz**	**2/1**
Data Security and Cryptography	**Köpsell**	**0/4**
Security Lab	**Köpsell**	**2/2**
Computers and Society	**Köpsell**	**2/0**
Introduction to Data Protection Law	**Wagner**	**2/0**

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.

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

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.***

General Aims of Education in IT-security — How to achieve ?

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.

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
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.

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
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!

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
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**

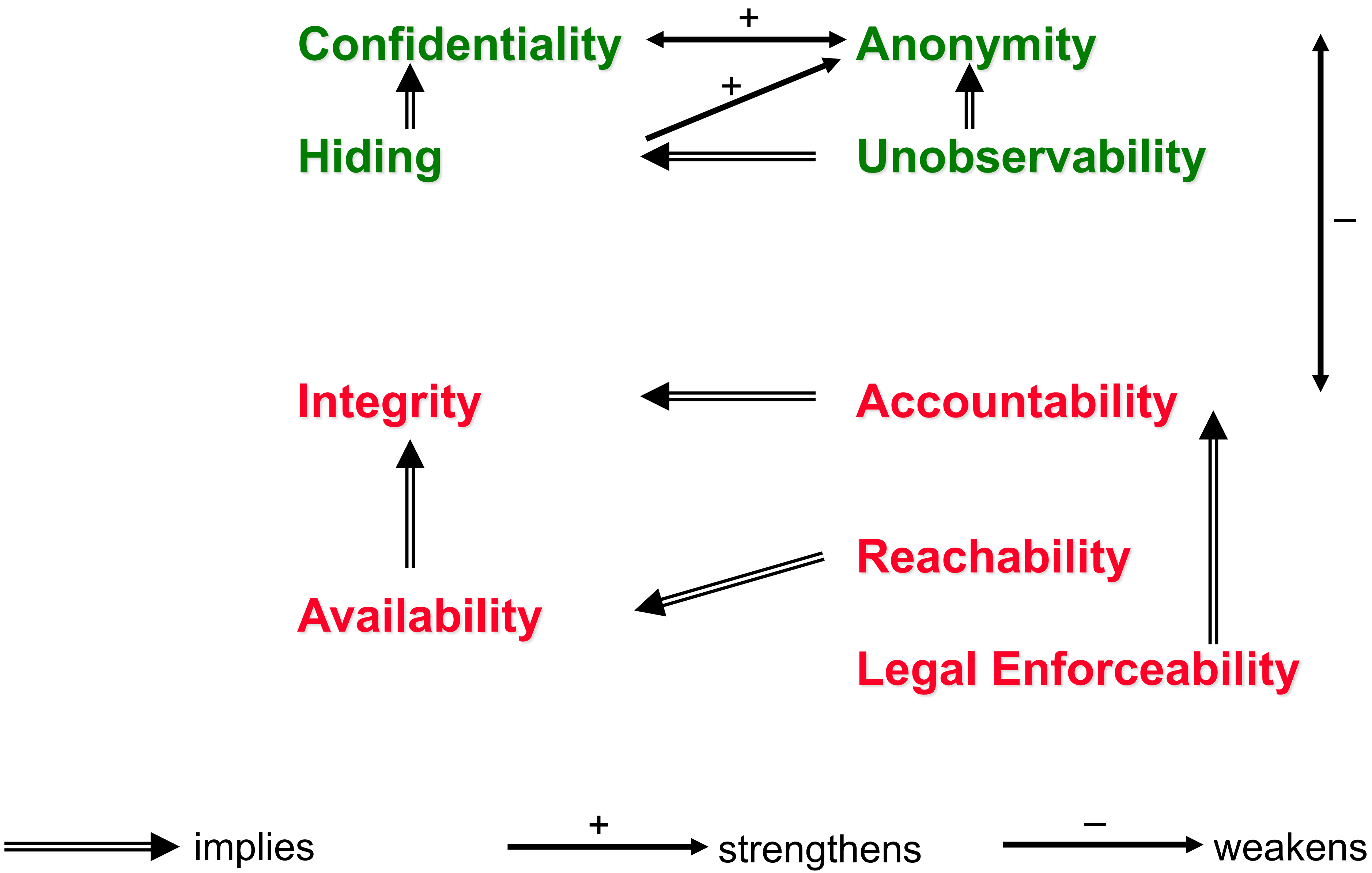

GO ARMY!

- **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.

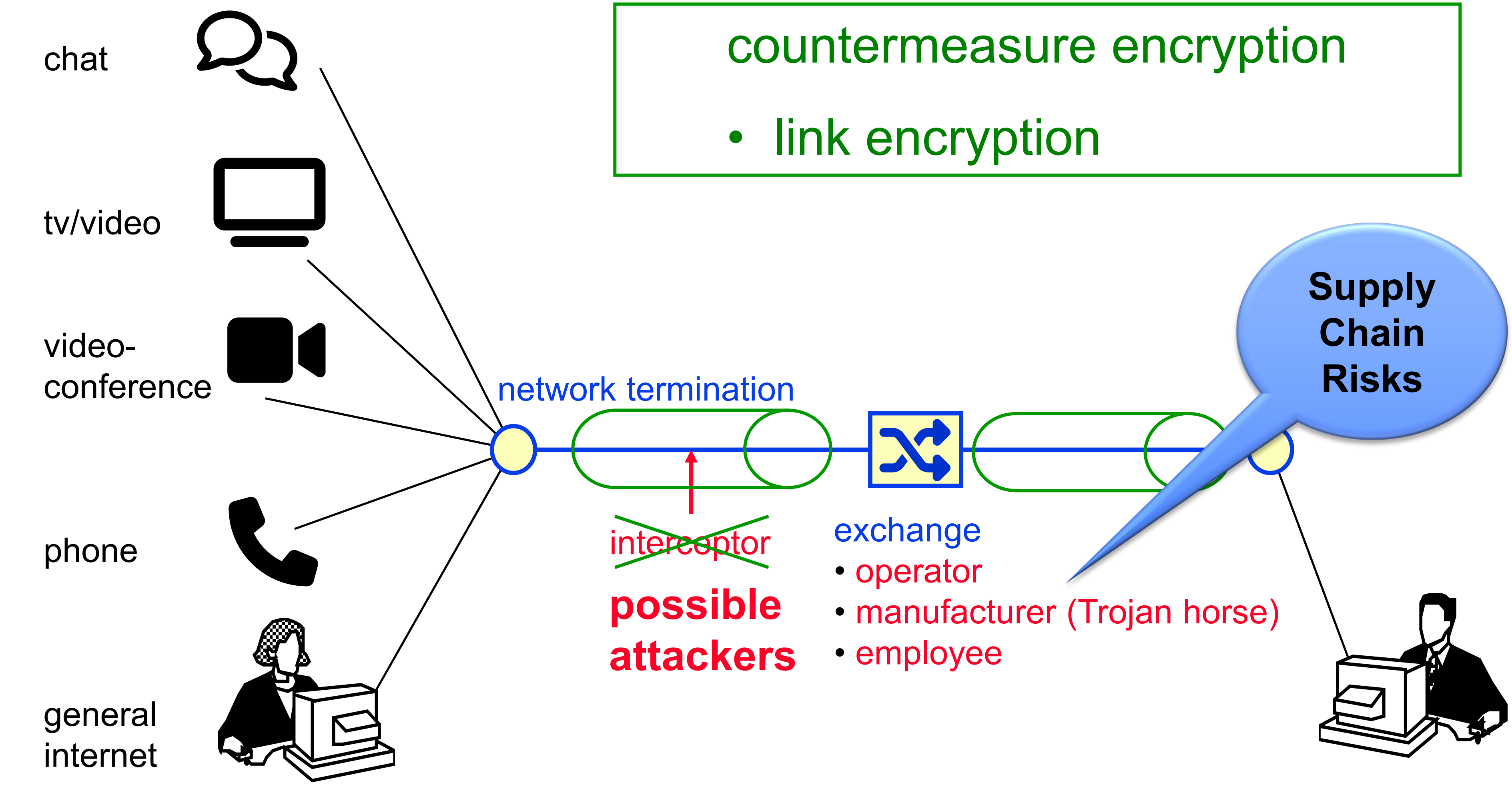
⇨ ***Anonymity* in terms of *Unlinkability*:**

Unlinkabilty between an identity (subject) and the IOI in question (message, data record etc.)

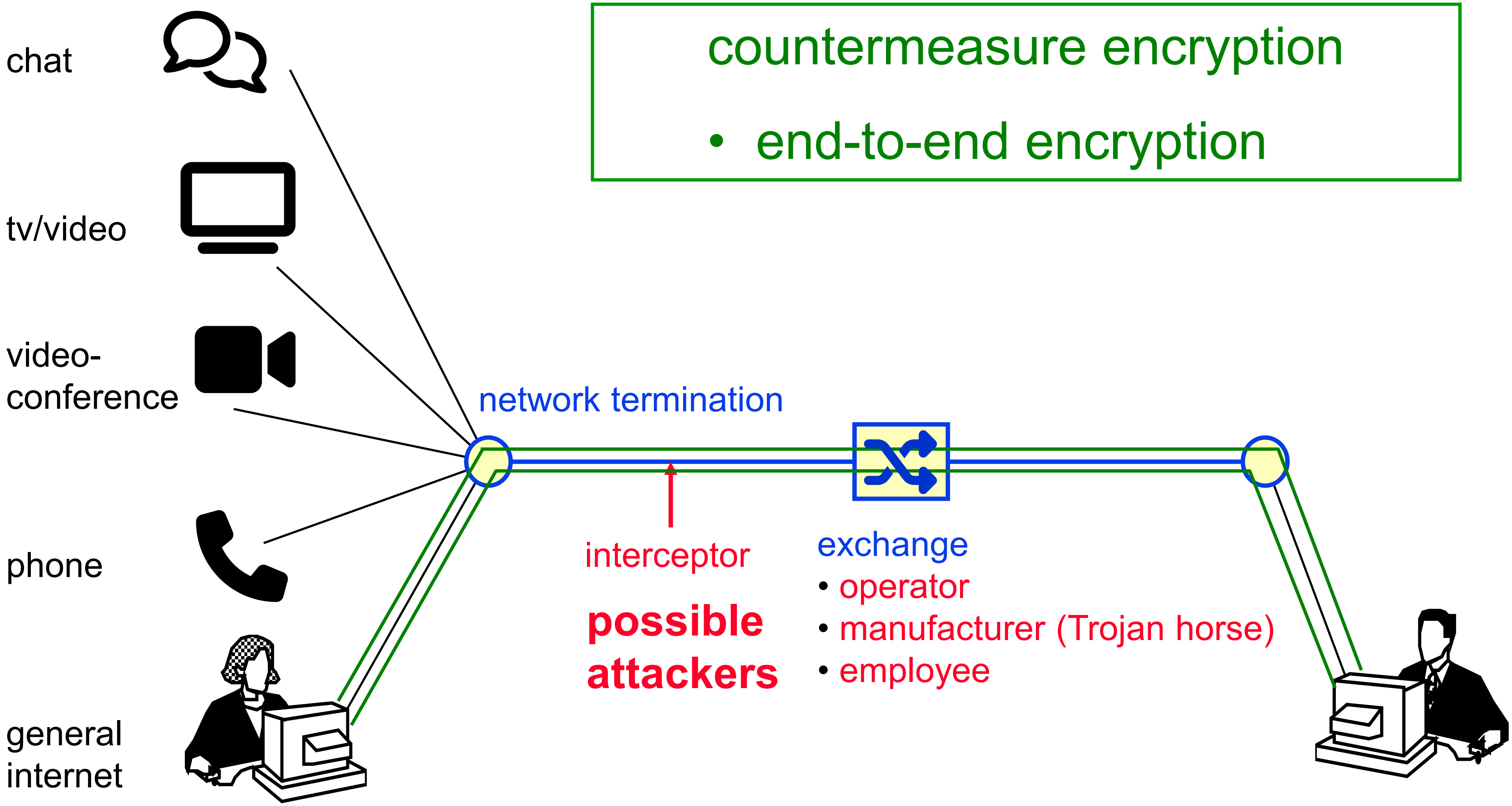
Correlations between protection goals



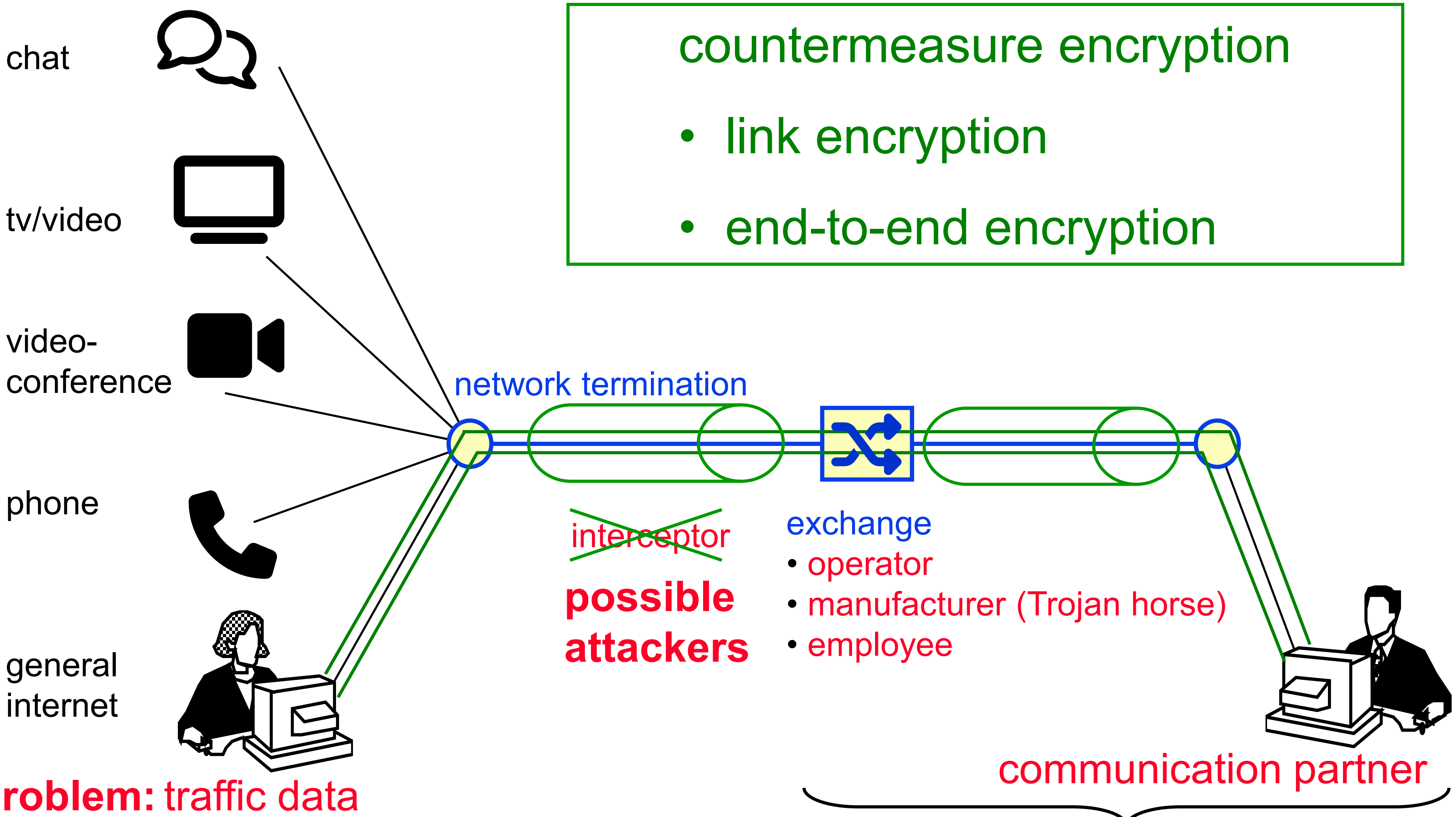
Observability of users in switched networks



Observability of users in switched networks



Observability of users in switched networks



data on interests: Who? What?

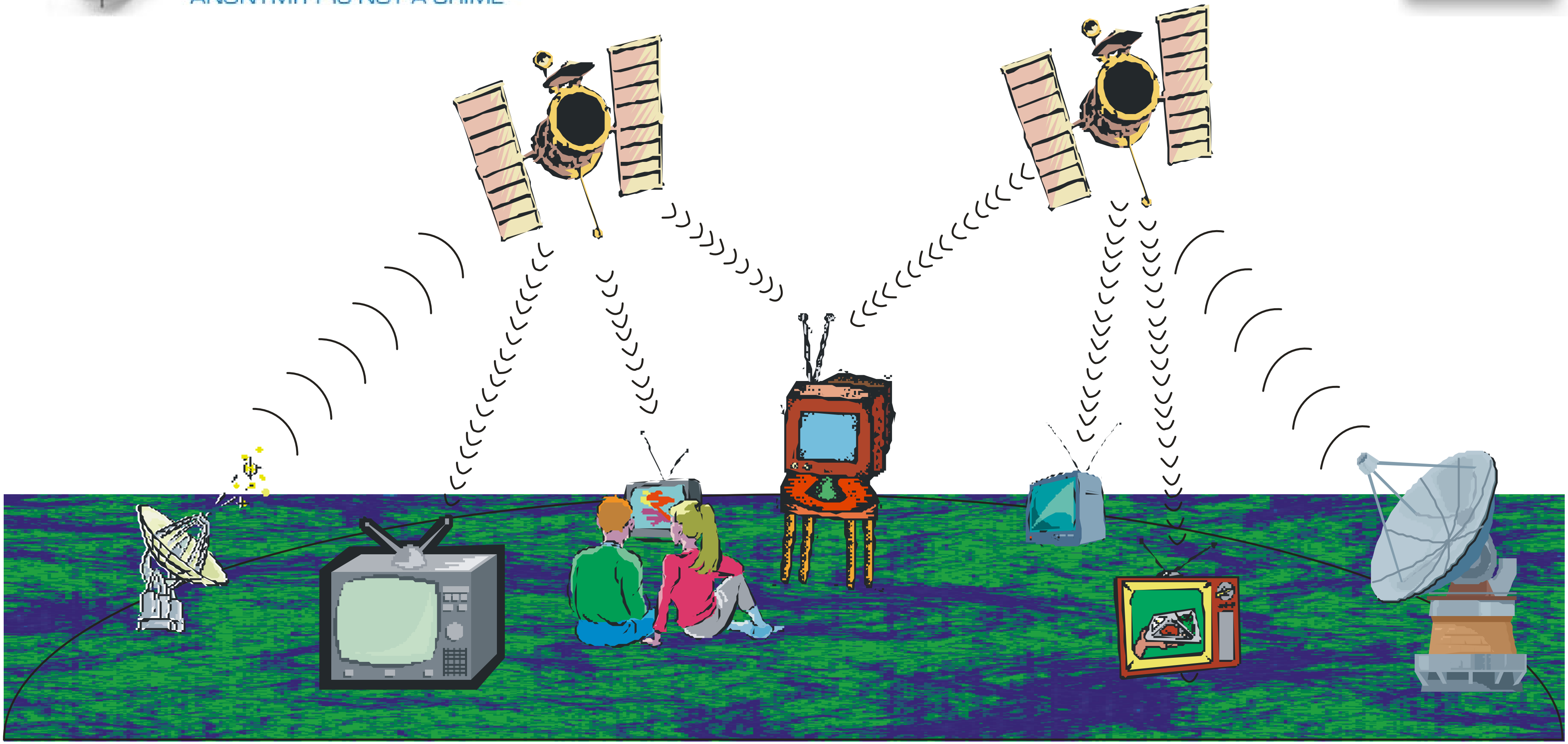
Problem: traffic data
who with whom?
when? how long?
how much information?

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

Excerpt from: 1984

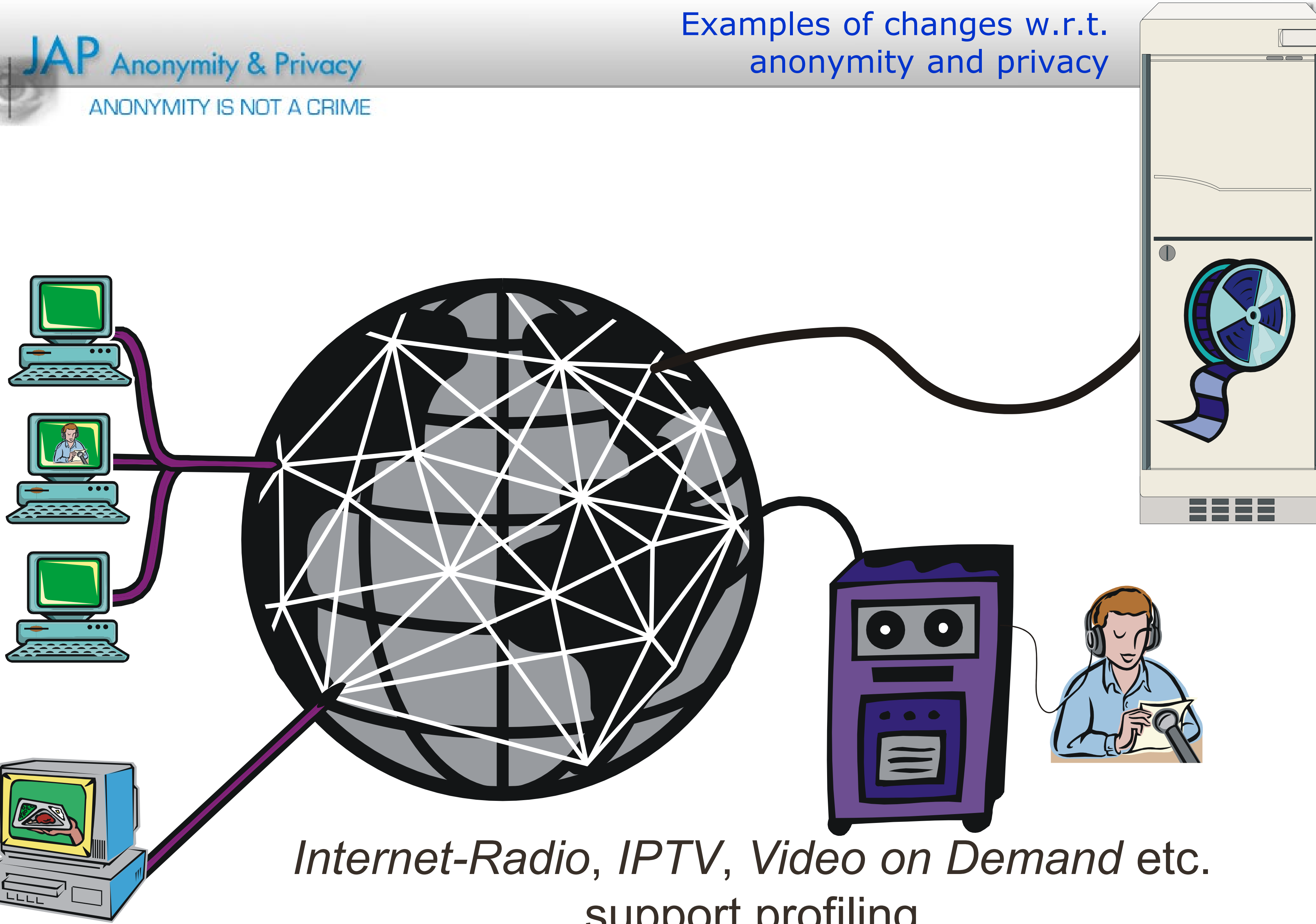
With the development of television,
and the technical advance which
made it possible to receive and transmit
simultaneously on the same instrument,
private life came to an end.

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

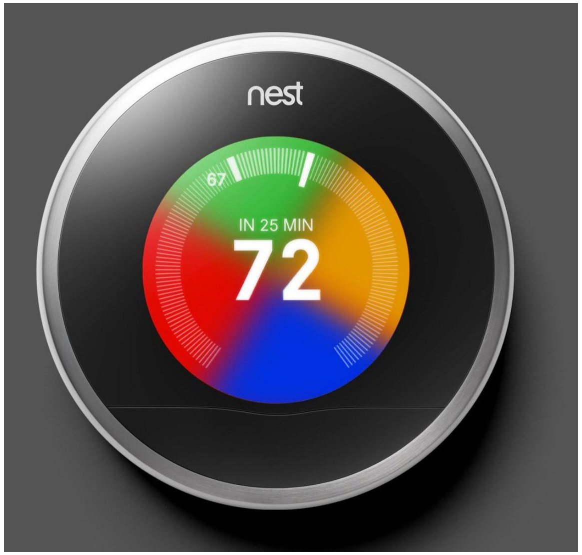
Anonymous plain old letter post is substituted by „surveillanceable" e-Mails

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/]



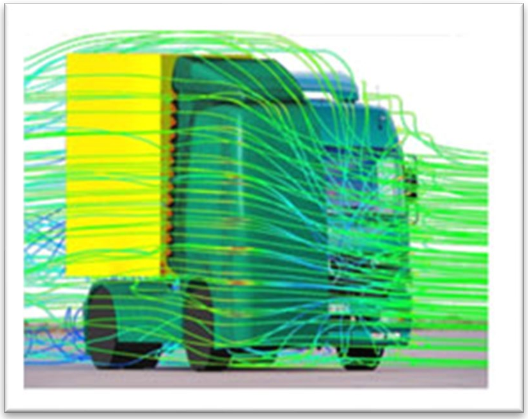
http://www.digitaltrends.com/home/google-just-bought-nest-3-2-billion/

http://www.bmw.de/de/topics/faszination-bmw/connecteddrive/ubersicht.html

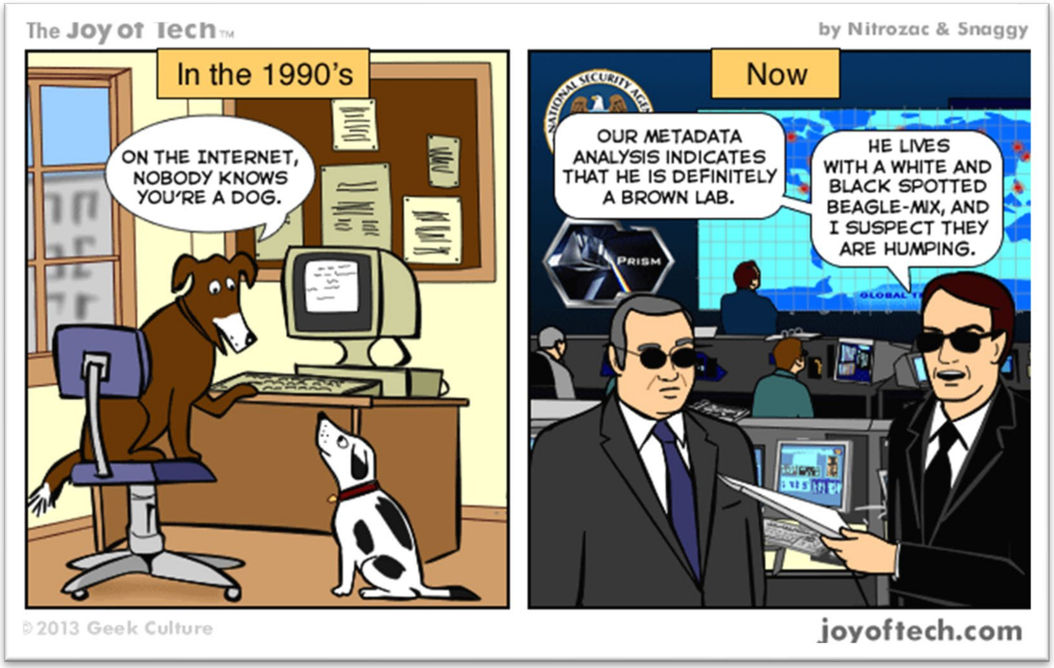
- ⌘ Smart Home
- ⌘ Smart Car
- ⌘ Smart Watch
- ⌘ Smart TV
- ⌘ Smart …

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

- Samuel Warren, Louis Brandeis: "**The Right to Privacy**", Harvard Law Review, Vol. IV, No. 5, 15th December **1890**

- **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

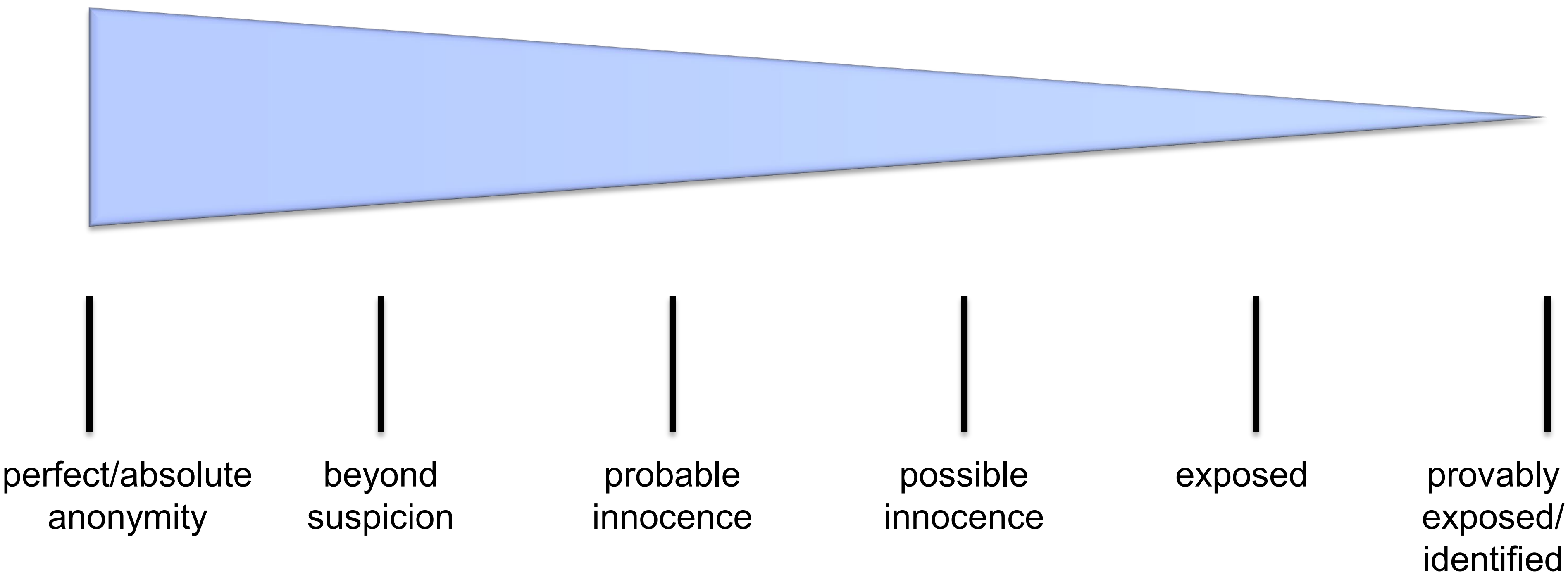
- Principles
  - collect and process personal data **fairly and lawfully**
  - **purpose binding**
    - keep it only for one or more specified, explicit and lawful purposes
    - use and disclose it only in ways compatible with these purposes
  - **data minimization**
    - adequate, relevant and not 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

- 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

Degress of Anonymity

[M. Reiter, A. Rubin: „Crowds: Anonymity for Web Transactions“, 1999]



- 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

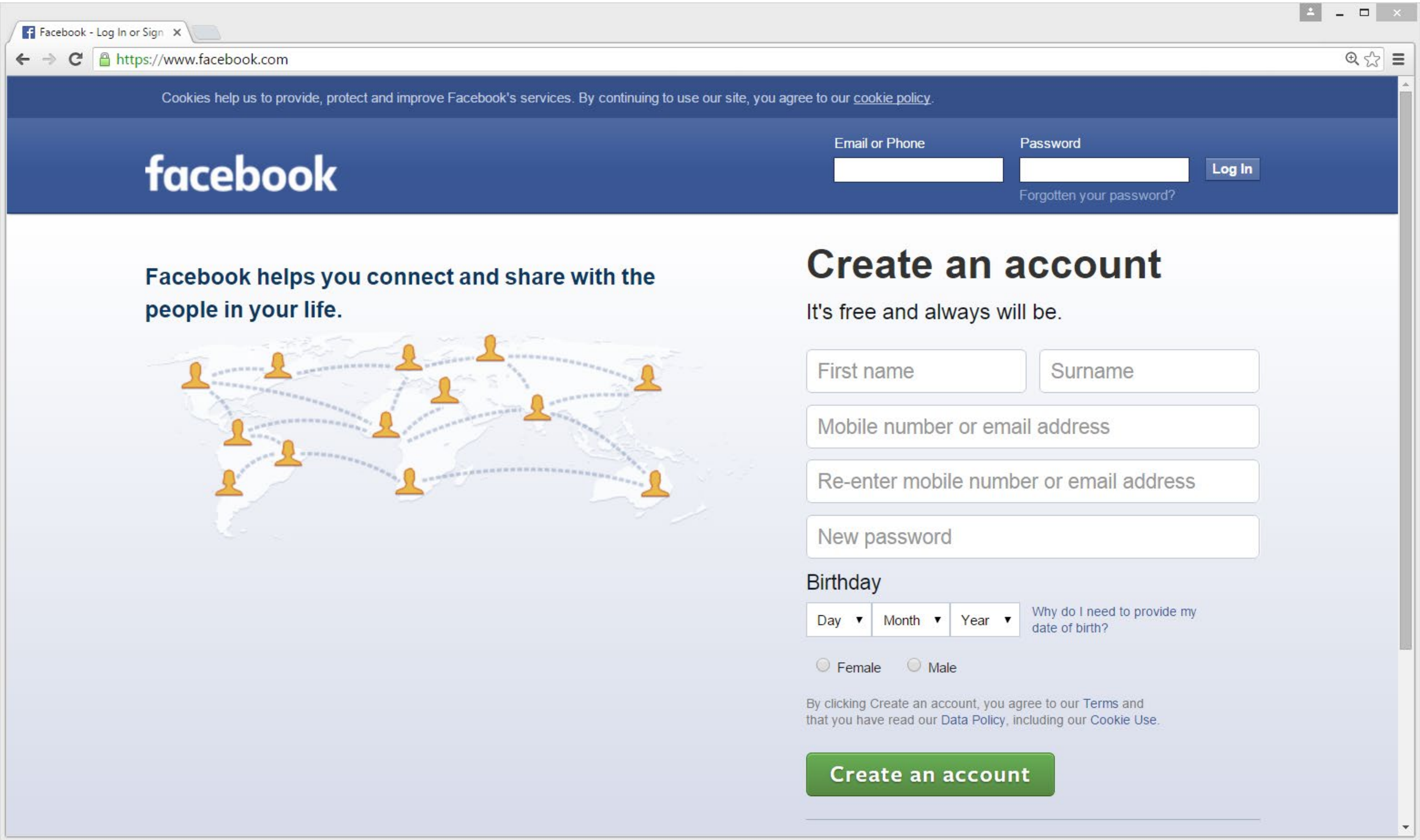
Attacker (-model)

Questions:

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

Realistic protection goals/attacker models: Technical solution possible?

Social Networks



Attacker (-model)

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 O: $0 < P(E|O) < 1$

perfect: $P(E) = P(E|O)$

Anonymity of an entity

Unlinkability of events

if necessary: partitioning in classes

Protection of the recipient: Broadcast

A. Pfitzmann, M. Waidner 1985

Performance? — more capable transmission system

Addressing — (if possible: switch channels)

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

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

		address distribution: public address	address distribution: private address
implicit address	invisible	very costly, but necessary to establish contact	costly
	visible	should not be used	change after use

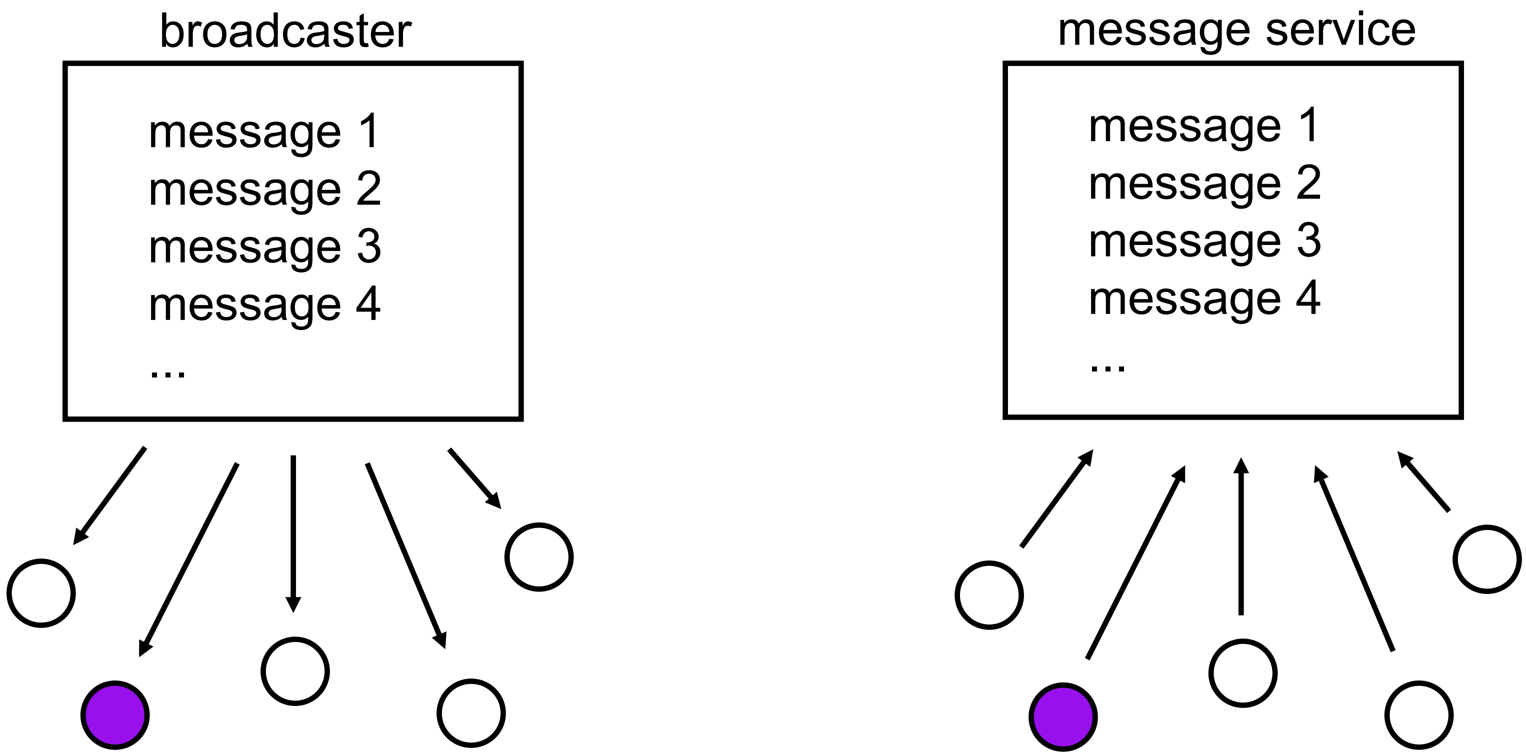
BitMessage (J. Warren, 2012)

- 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	←→	asymmetric encryption system
invisible private address	←→	symmetric encryption system

Broadcast vs. Queries



broadcast of separate messages to all recipients

everybody can query all messages

Private Message Service

User is interested in D[2]:

Index within Request-Vector = 1234

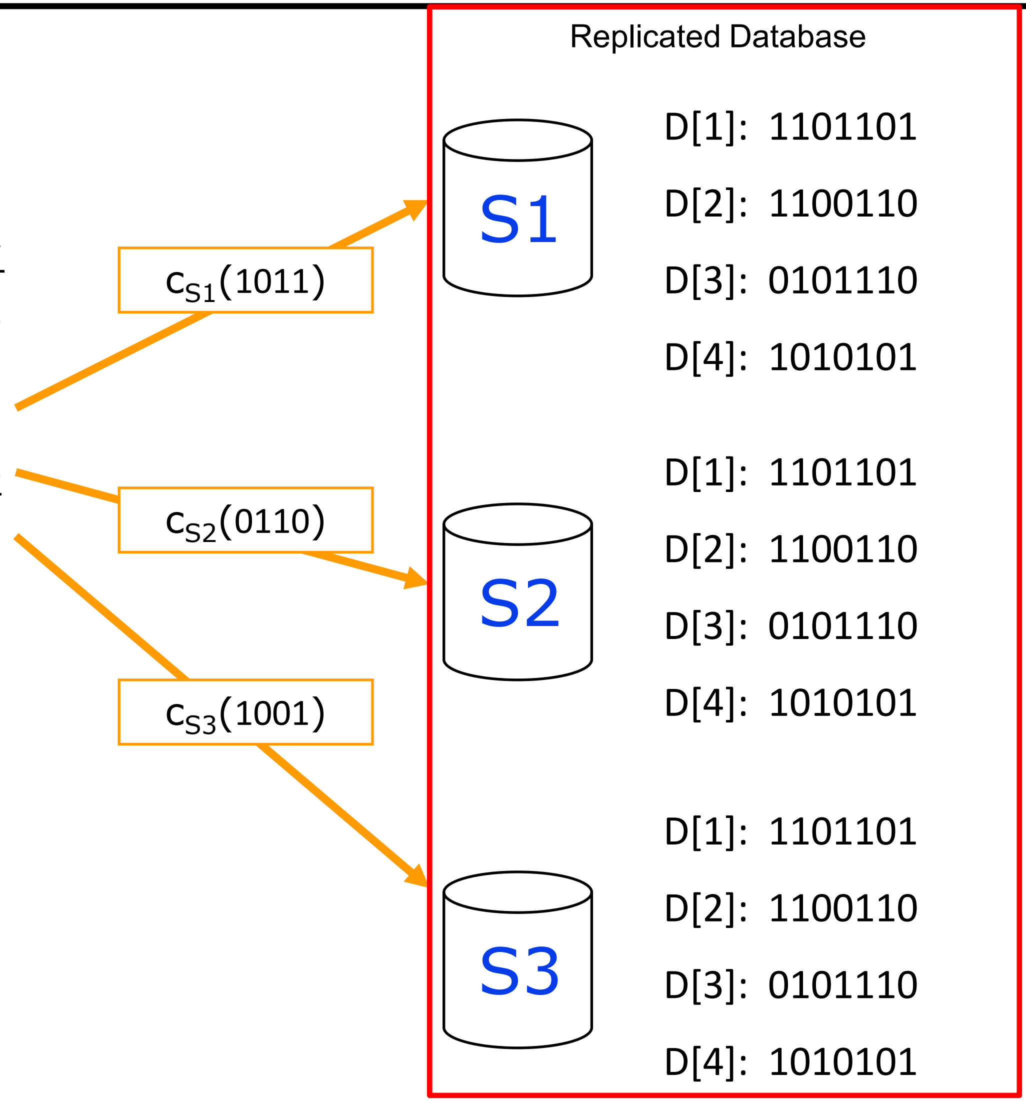
Set Vector = 0100

Chose random Vector (S1) = 1011

Chose random Vector (S2) = 0110

Calculate Vector (S3) = 1001

Calculations: XOR



David A. Cooper, Kenneth P. Birman 1995

Efficiency improvements: A. Pfitzmann 2001

Private Message Service

User is interested in D[2]:

Index within Request-Vector = 1234

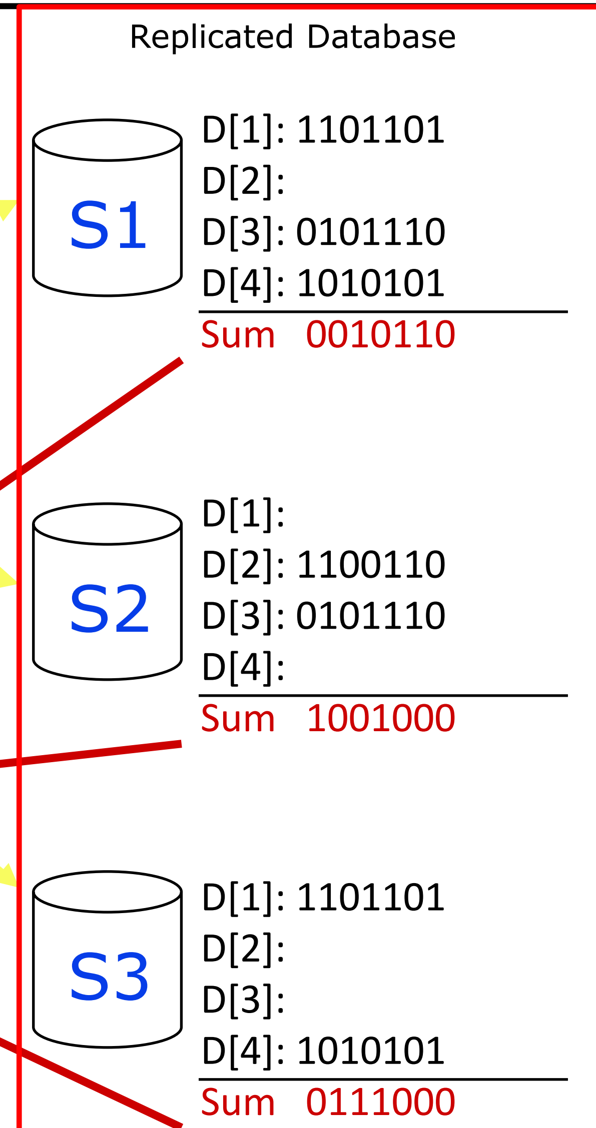
Set Vector = 0100
Chose random Vector (S1) = 1011
Chose random Vector (S2) = 0110
Calculate Vector (S3) = 1001

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!



Reducing Traffic from User to Database

User is interested in D[2]:

Index within Request-Vector = 1234

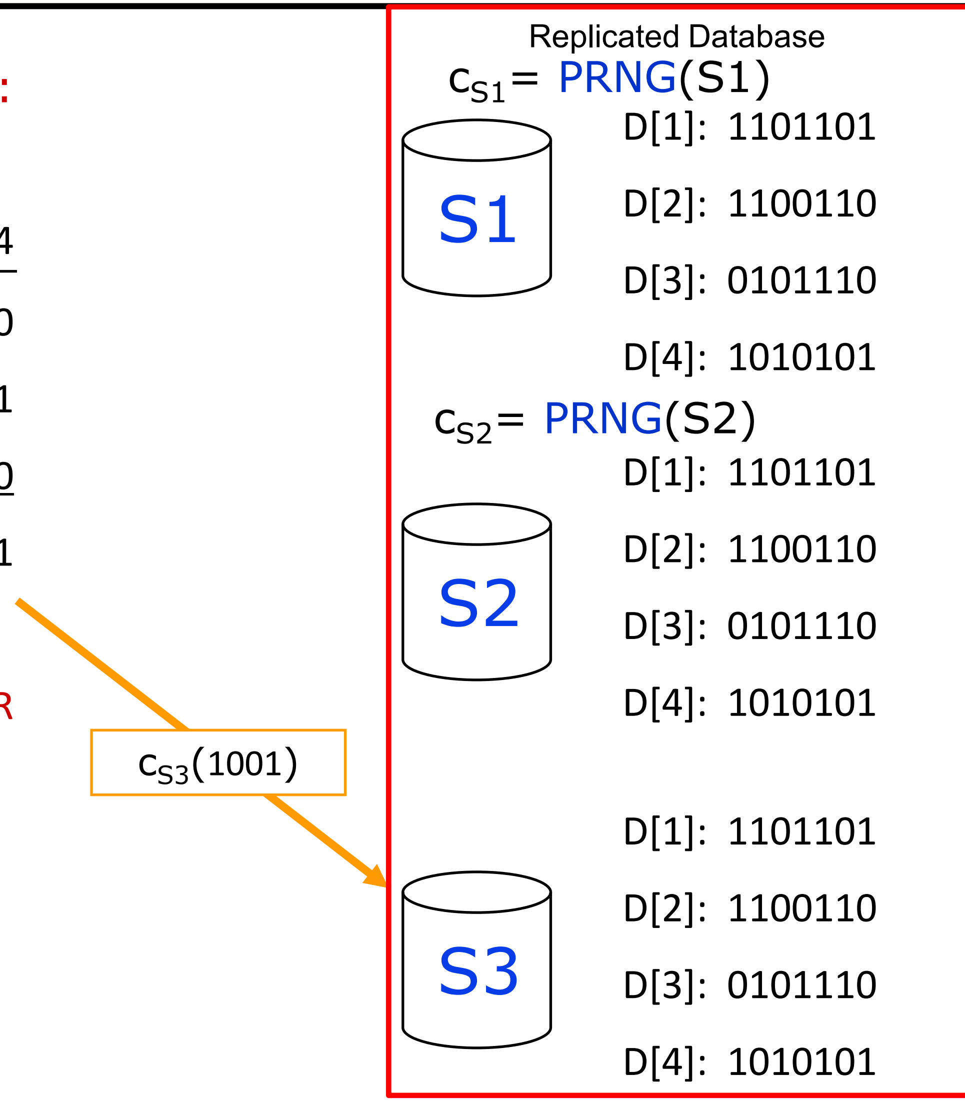
Set Vector = 0100

Generate random Vector PRNG(S1) = 1011

Generate random Vector PRNG(S2) = 0110

Calculate Vector (S3) = 1001

Calculations: XOR



Private Message Service

User is interested in D[2]:

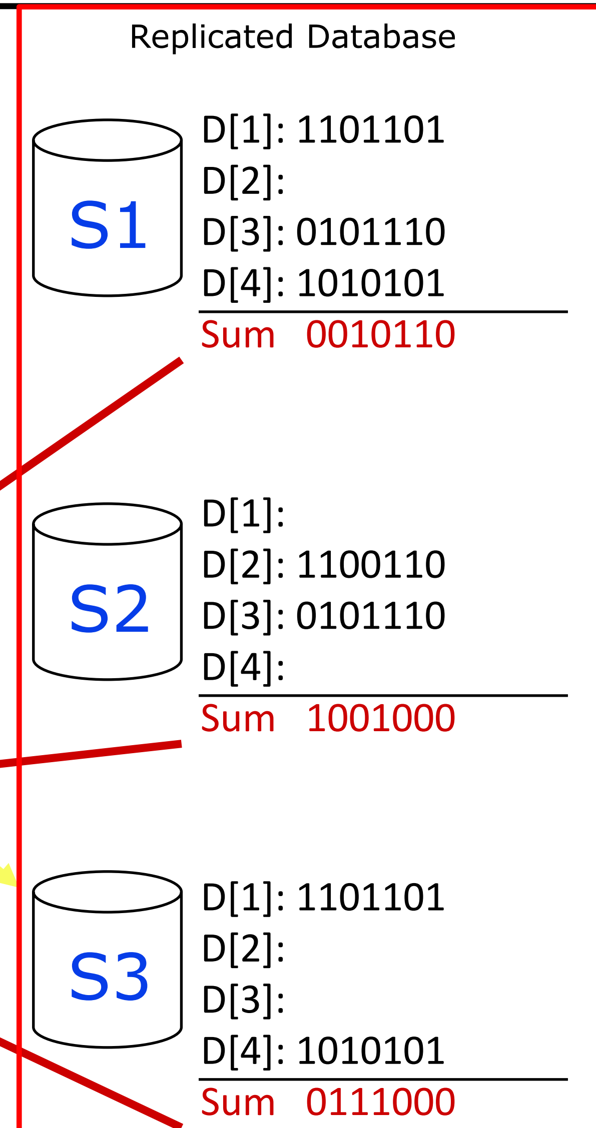
Index within Request-Vector = 1234

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



Reducing the Traffic from Database to User

User is interested in D[2]:

Index within Request-Vector = 1234

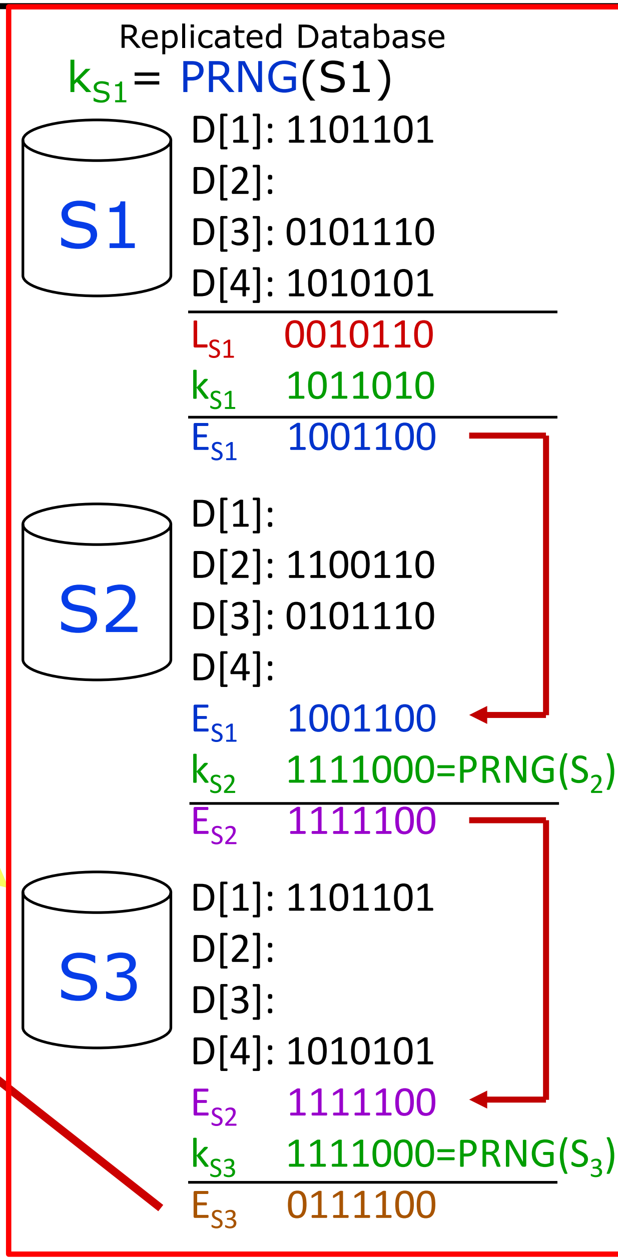
Set Vector = 0100
Chose random Vector (S1) = 1011
Chose random Vector (S2) = 0110
Calculate Vector (S3) = 1001

Server calculates XOR
of the requested records

Answer

E_{S3}	0111100
k_{S1}	1011010
k_{S2}	1111000
k_{S3}	1111000

Sum is D[2]: 1100110



"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 ↗↙ 1

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

Set of re-writable memory cells = implicit address

cell 1	Addr 1	Addr 3	Addr 5
cell 2	Addr 1	Addr 4	Addr 6
cell 3	Addr 2	Addr 3	Addr 6
cell 4	Addr 2	Addr 4	Addr 5

Goal: Increase number of addresses

Idea: Message *m* is stored in a set of *a* memory cells

How: choose a–1 values randomly, choose the value of the a^{th} cell such that the sum of all *a* cells is *m*.

Improvement: For overall *n* memory cells, there are now 2^n–1 usable implicit addresses

Drawback: overlaps → they cannot be used independently

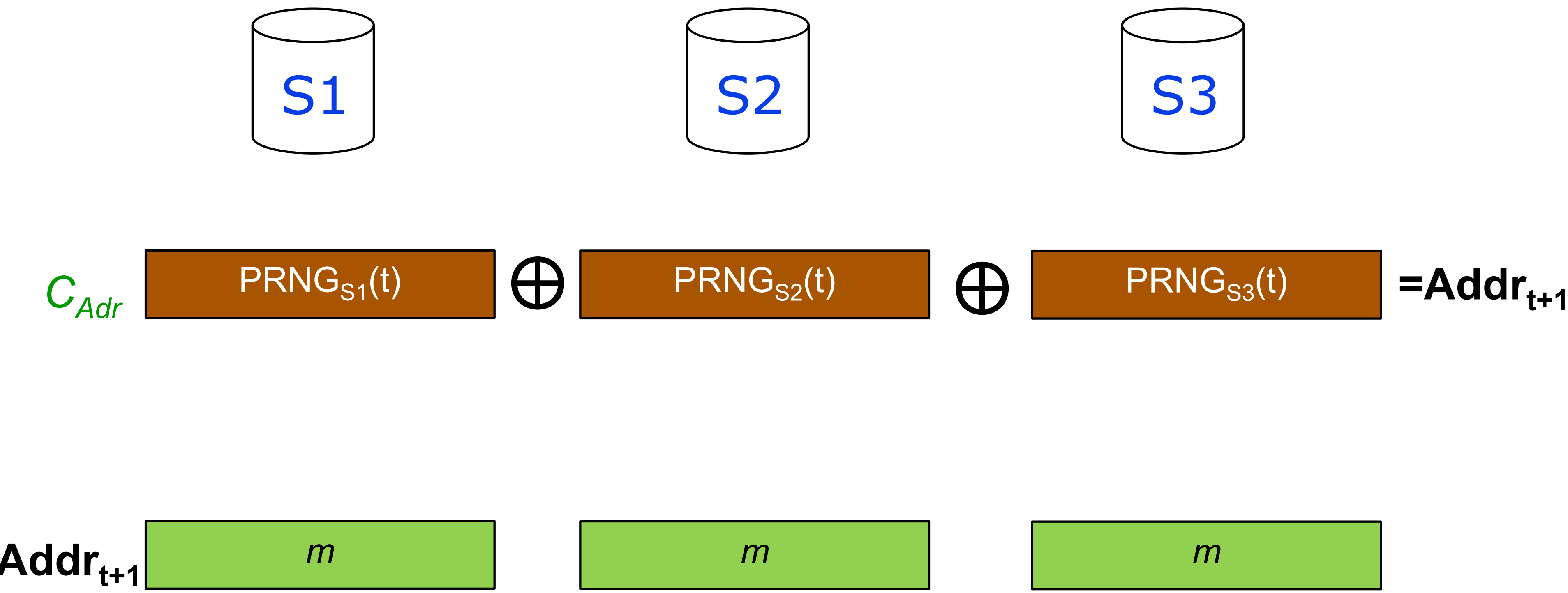
Solution: collision → retransmit after randomly chosen time intervals

Note: 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*+1
reads the values from reserved memory cells at time *t*
These values identify the memory cell to be used at time *t*+1



Invisible implicit addresses using “query and superpose” (2)

hopping between memory cells = invisible implicit address

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

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 C_{Adr} with $PBG_s(t)$:

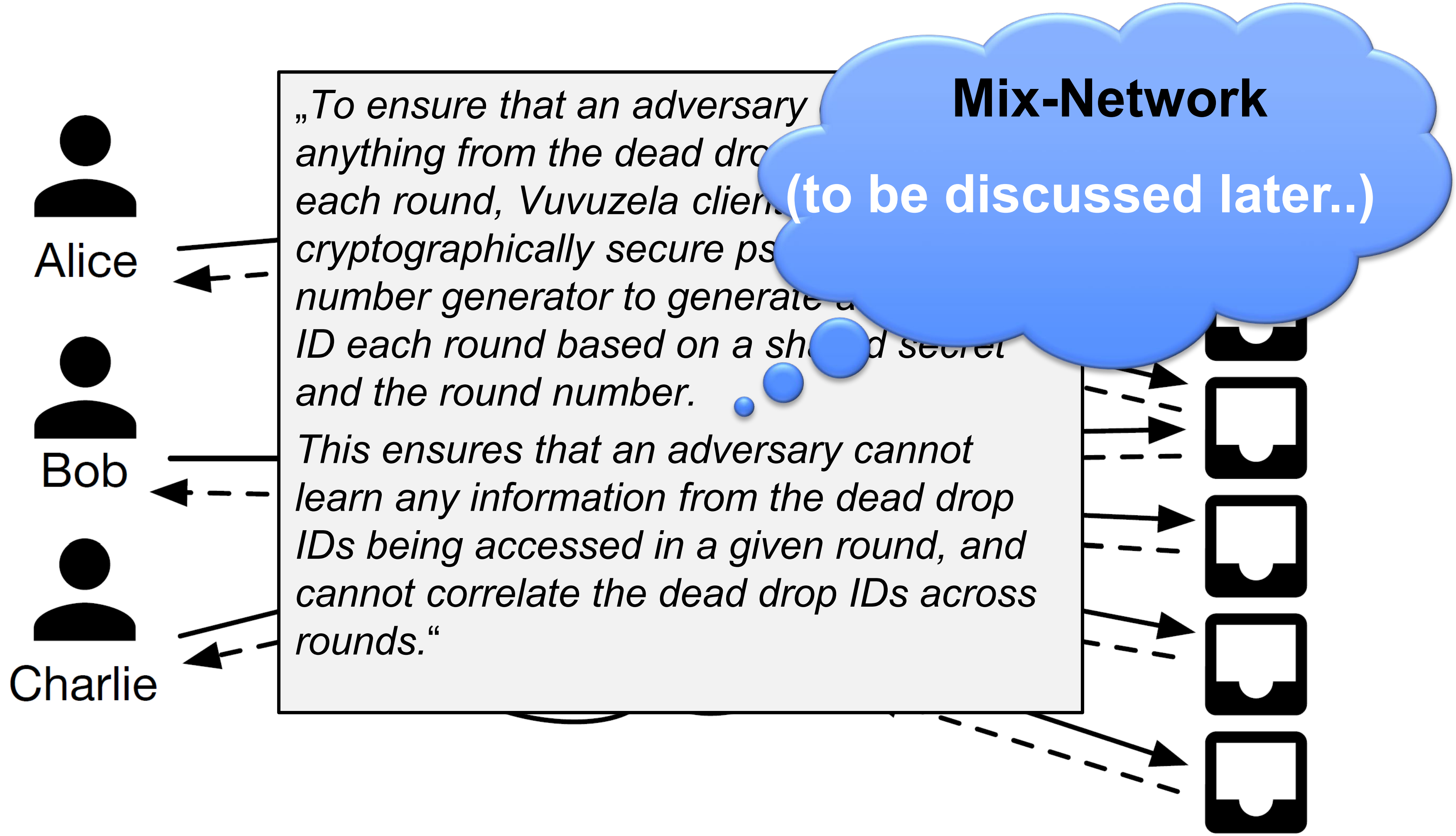
$$C_{Adr} := PBG_s(t)$$

- User queries anonymously (e.g. via MIXes) $\sum_s PBG_s(t)$ (possible in one step)
  user employs $S_{\sum_s PBG_s(t)}$ for message 1 ↗↙ 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: 1↗↙2 (if ≤ 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 C_{Adr} are readable.

Hopping between „cells“ for anonymous chat

[van den Hooff et al.: „Vuvuzela: scalable private messaging resistant to traffic analysis“, 2015]



„To ensure that an adversary anything from the dead dro each round, Vuvuzela clien cryptographically secure ps number generator to generate a ID each round based on a sh d secret and the round number.

This ensures that an adversary cannot learn any information from the dead drop IDs being accessed in a given round, and cannot correlate the dead drop IDs across rounds.“

(1) Users access dead drops

(2) Honest server unlinks users from dead drops and adds cover traffic

(3) Adversary can't tell who is talking to who by looking at dead drop access patterns

Invisible implicit addresses using "query and superpose" (3)

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. authenticated messages ➔ detect modifying attacks
   - use disjoint set of servers
   - lay traps
     - send the same query vector to many servers
     - check their responses by comparison

Protection of the sender

Dummy messages

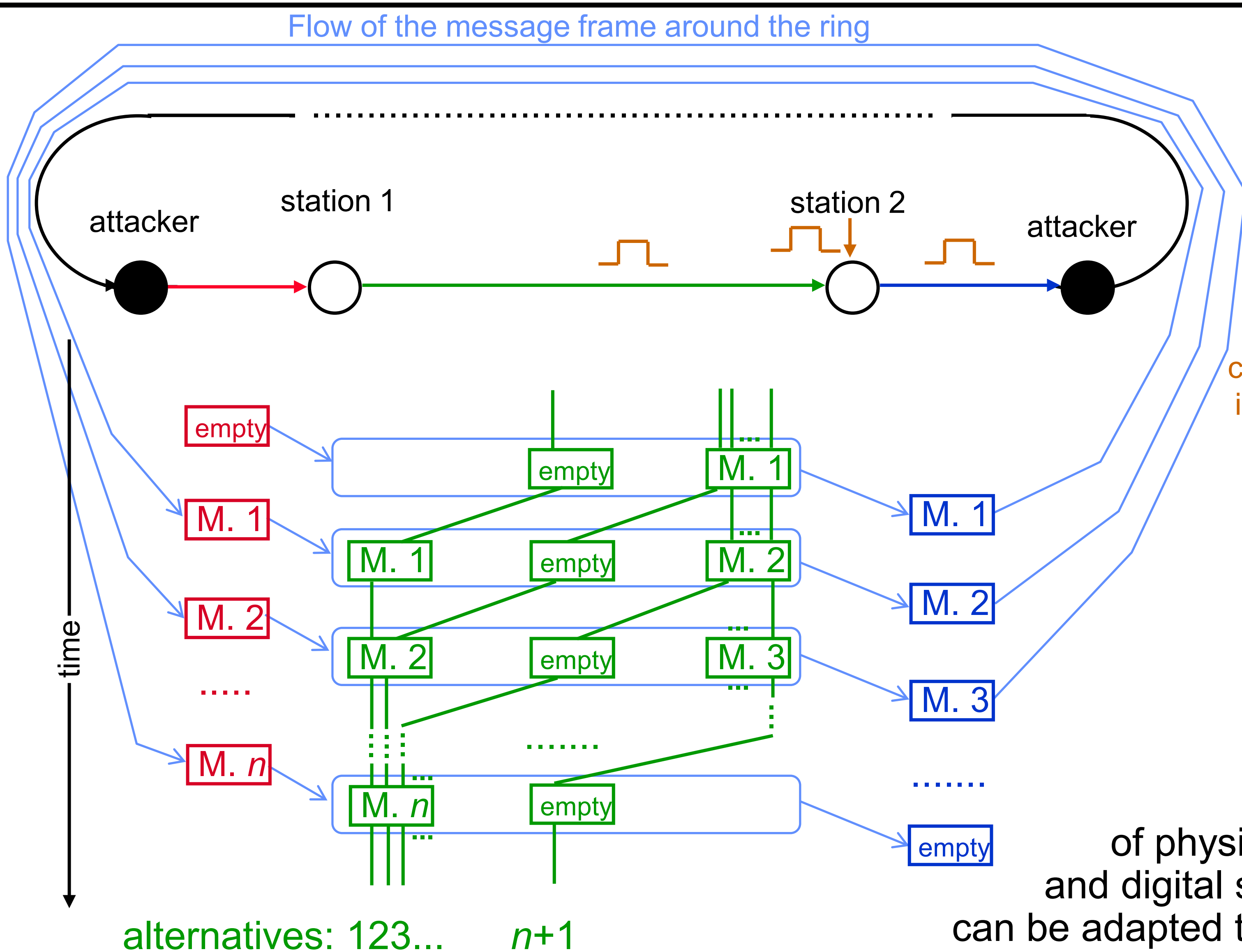
- do not protect against addressee of meaningful messages
- make the protection of the recipient more inefficient

Unobservability of neighboring lines and stations as well as digital signal regeneration

example: RING-network

Proof of anonymity for a RING access method

A. Pfitzmann 1983 - 1985



Digital signal regeneration:

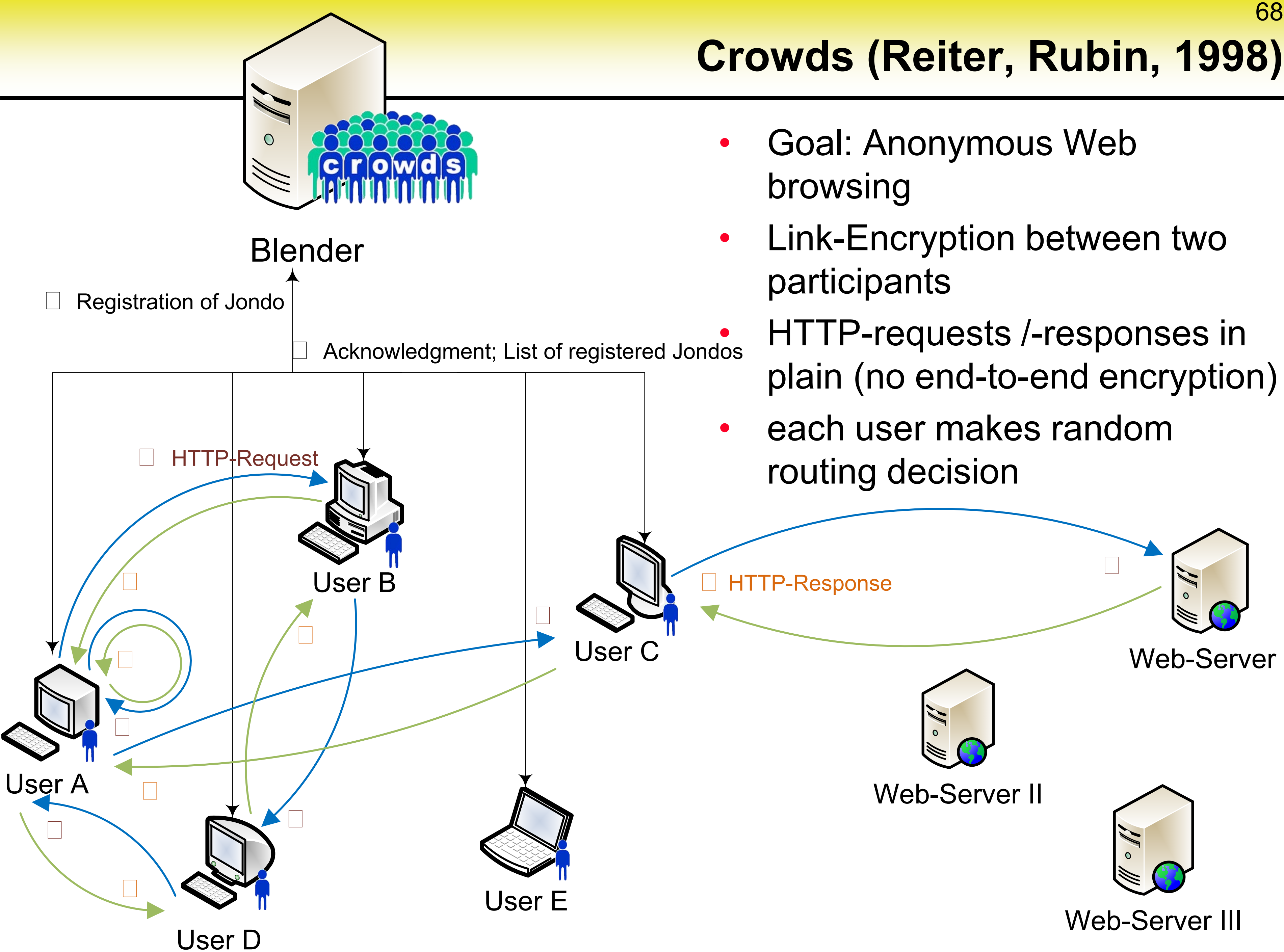
The analogue characteristics of bits are independent of their true sender.

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

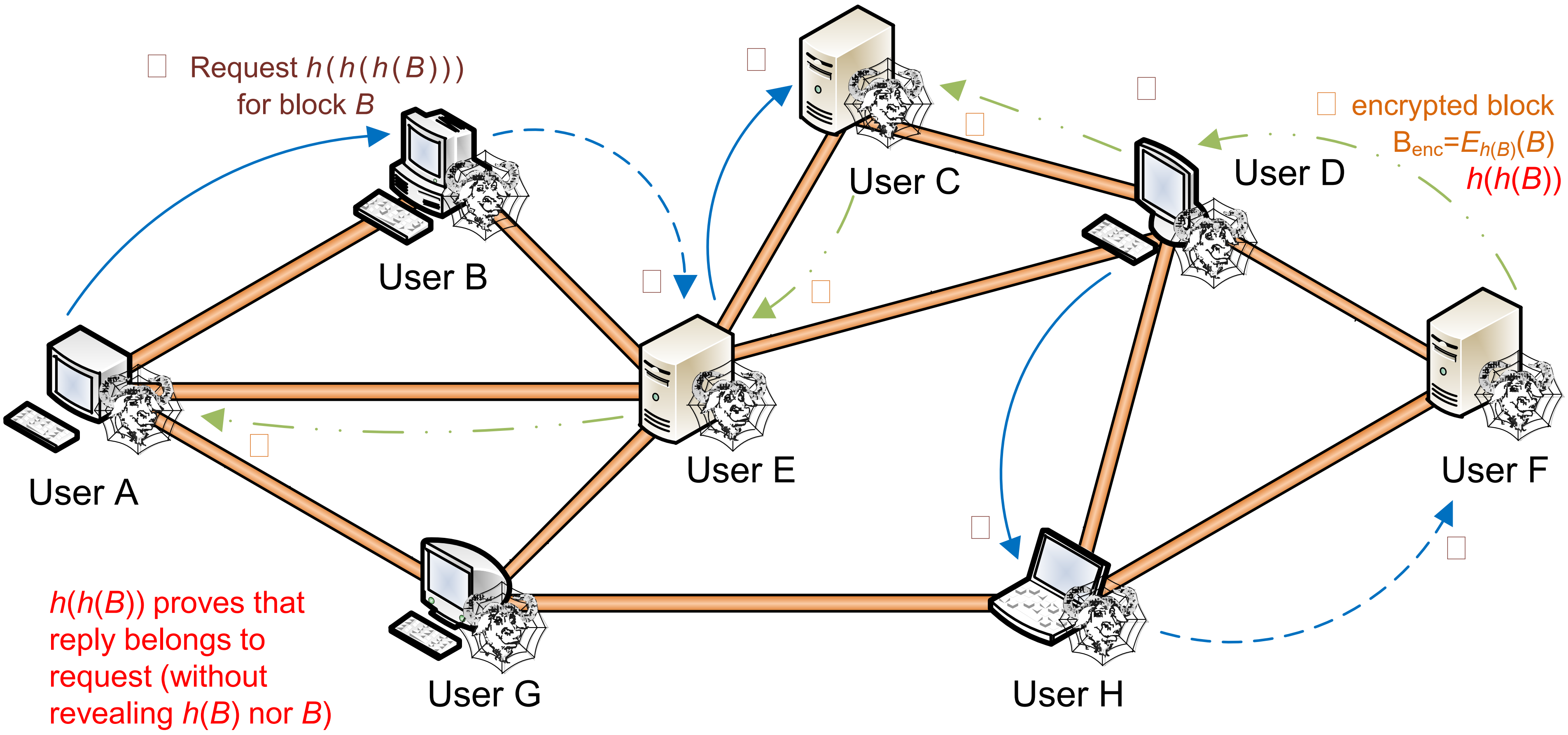
It reappears in another context in Crowds, GNUnet, etc.

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



GNUnet (gnunet.org, 2001)



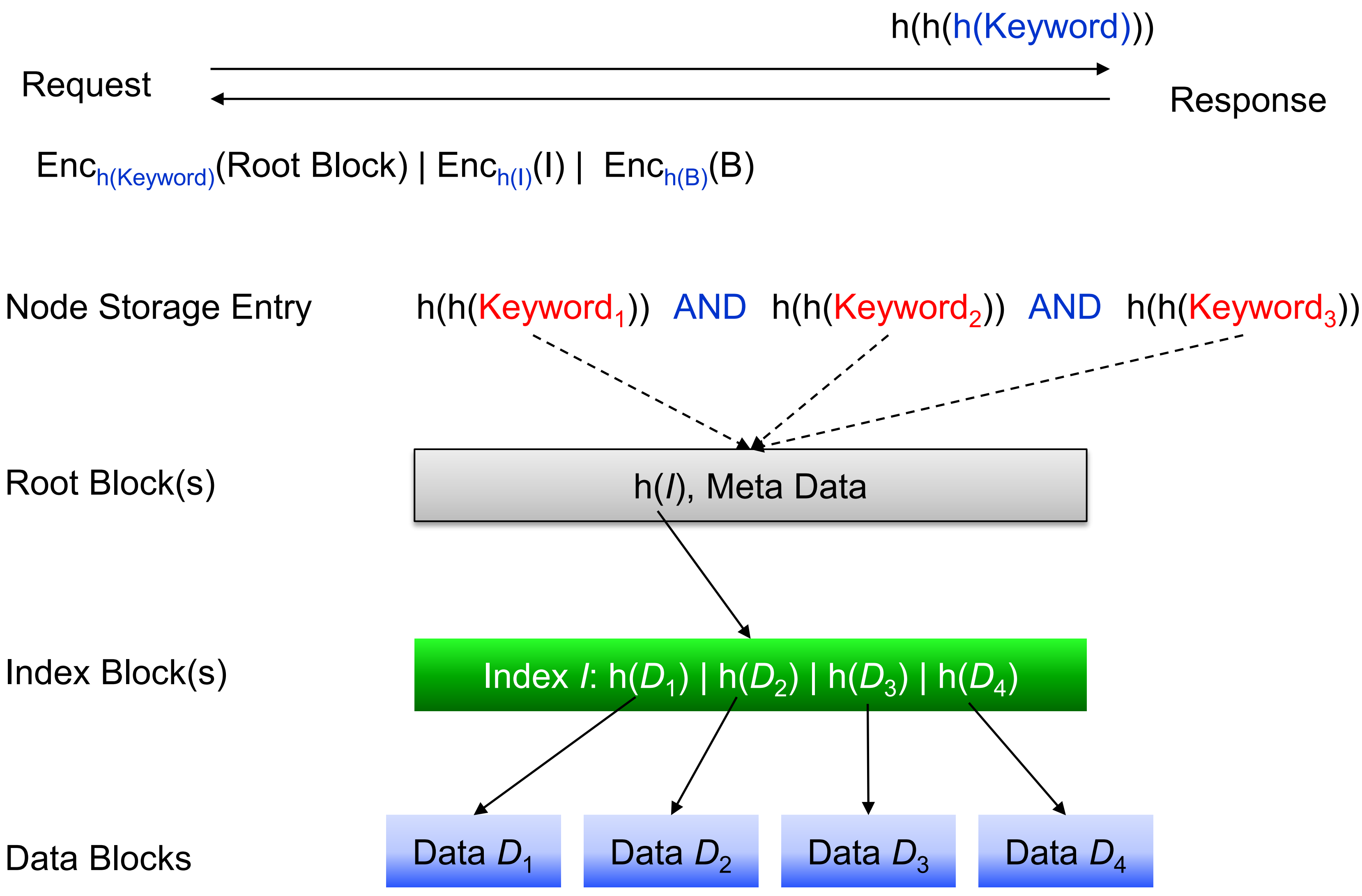
Link encrypted communication between two adjoining GNUnet users

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

Searching in GNUnet



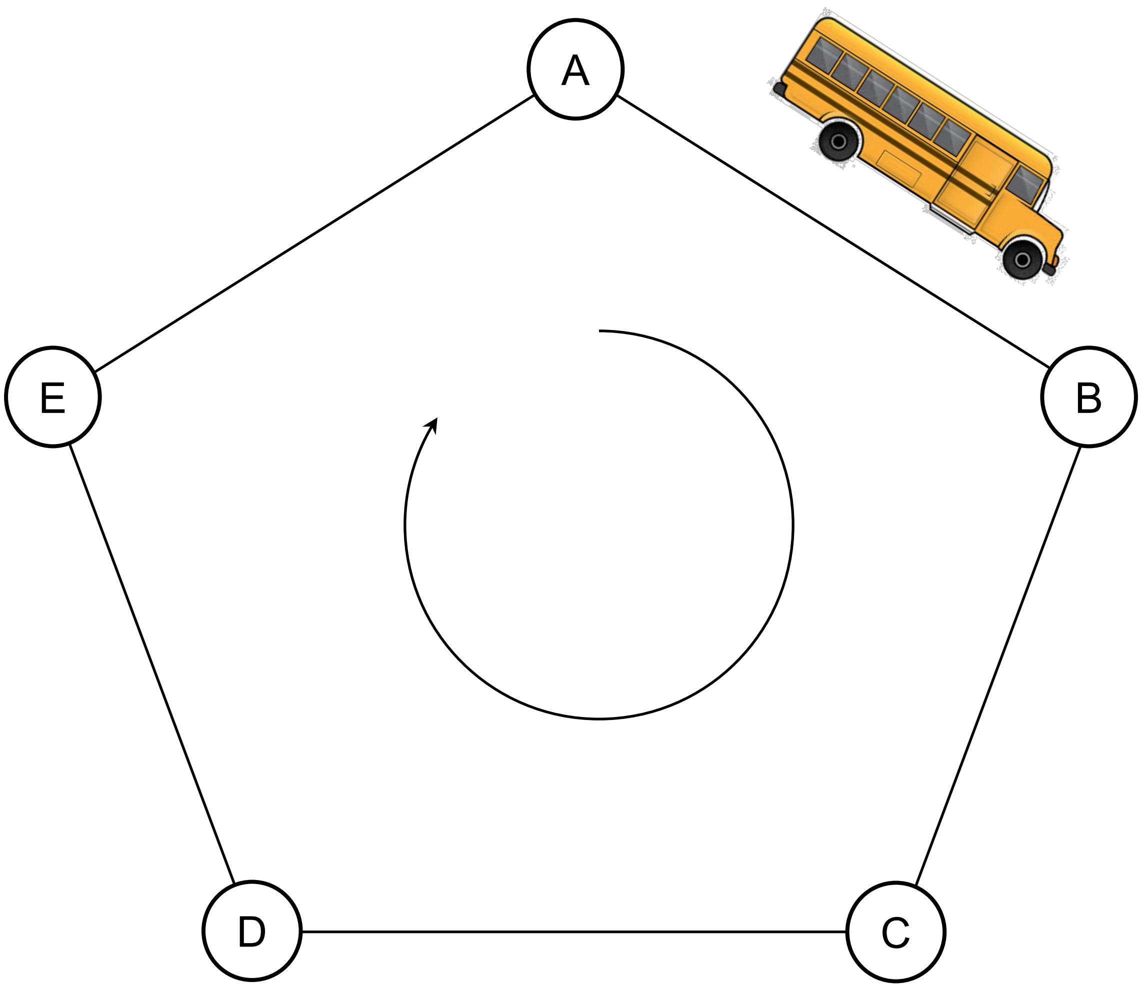
Buses…

- 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
    - follow-up: Andreas Hirt, Michael J. Jacobson, Jr., Carey Williamson: “Taxis: Scalable Strong Anonymous Communication”, 2008
      - follow-up: Adaml L. Young, Moti Young: “The Drunk Motorcyclist Protocol for Anonymous Communication”, 2014
- basic ideas follow a city-bus metaphor
  - messages send around contain „seats“, i.e., cells dedicated to certain users/messages
  - different protocols proposed: trade-off: communication complexity, time complexity, storage complexity

Buses…

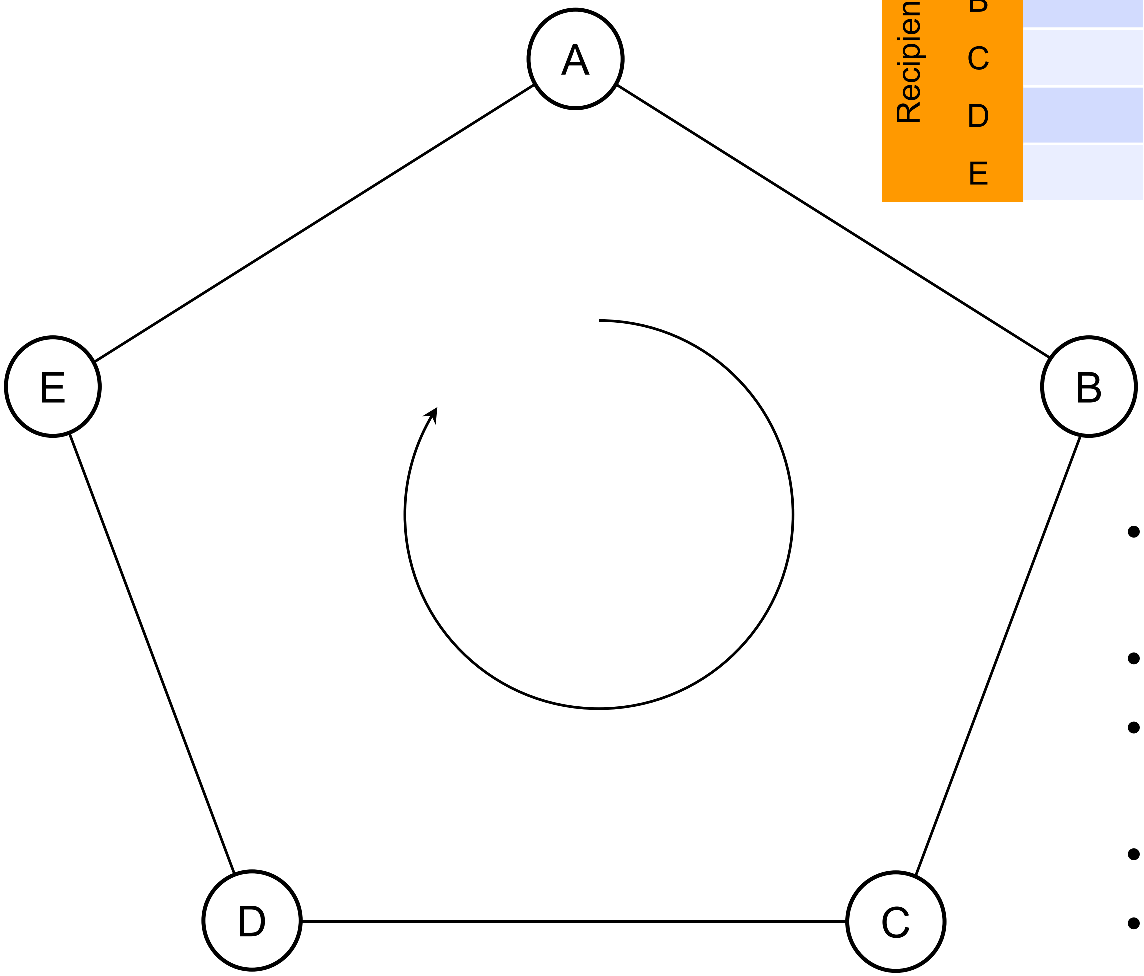
- Attacker model:
  - global observing outsider
  - observing participants (except sender/receiver!)
  - [modifying attackers are only considered wrt. availability]
- Protection goals achieved
  - sender anonymity
  - recipient anonymity
  - unobservability regarding sending/receiving of messages

Buses



Buses – simple solution

Message	Sender A	Sender B	Sender C	Sender D	Sender E
Recipient A	?				
Recipient B					
Recipient C		$m_{B \to C}$			
Recipient D					
Recipient E					



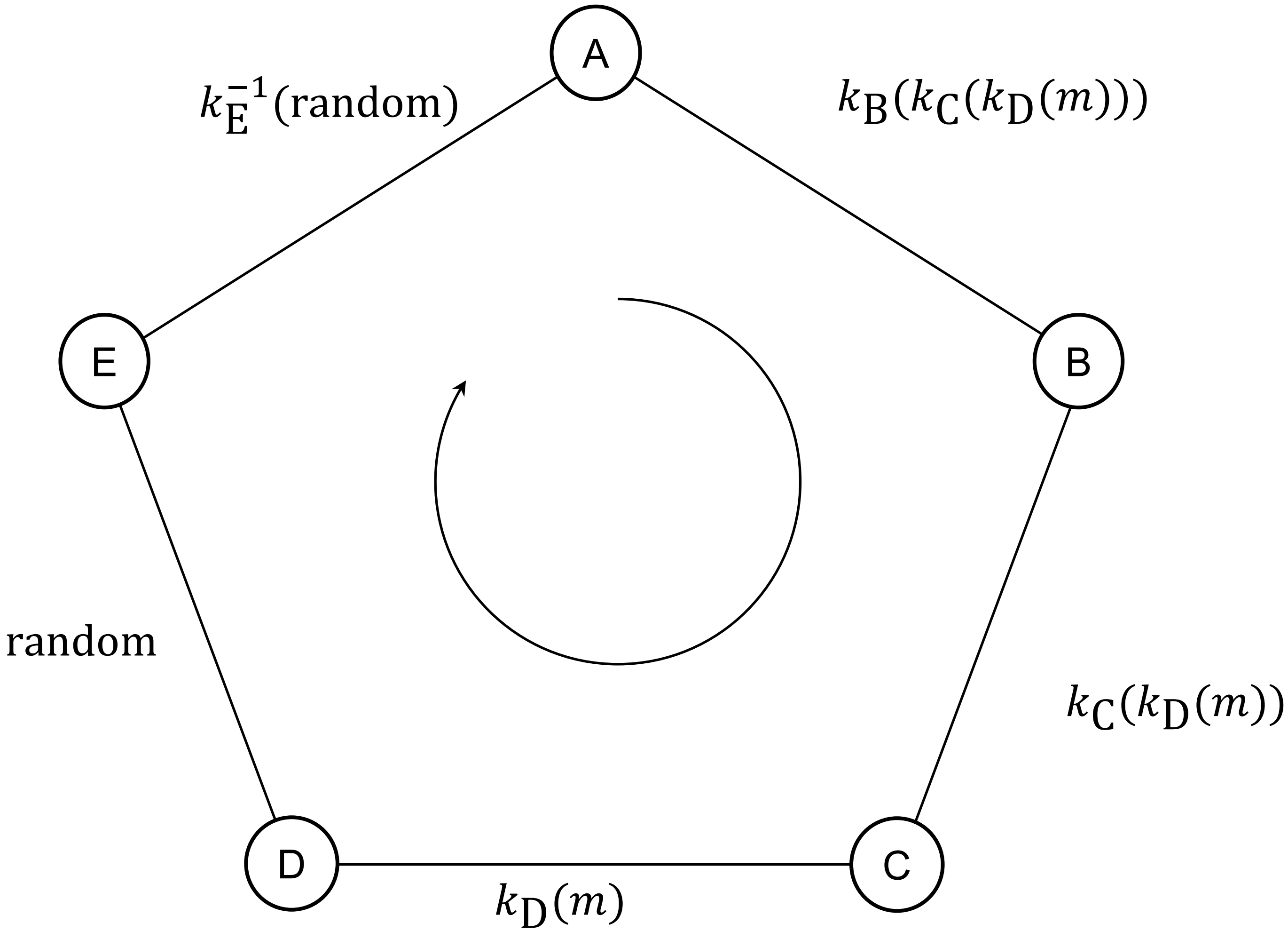
- dummy messages, if nothing to sent
- implicit addressing
- communication complexity: 1
- time complexity: O(n)
- storage complexity: $O(n^2)$

Buses – reducing storage complexity

- 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 sent some message m to *D*
- depicted is one seat of the bus



- replay attacks!

(Universal) Re-encryption

[Golle et al.: „Universal Re-encryption for Mixnets“, 2004]

- Re-encryption:
  - given: public key e, c=Enc(e,m)
  - create: c'=Enc(e,m) with c' ≠ c
- Universal Re-encryption:
  - Re-encryption without knowing e
    - ➔ avoids linkability (same recipient…)
- Implementation:
  - Recall ElGamal:
    - $e=g^x$
    - $Enc(m)=(g^y, m \cdot e^y)$
    - Homomorphic property: $Enc(m_1) \cdot Enc(m_2)=Enc(m_1 \cdot m_2)$
  - Re-encryption:
    - $Enc(m)^z = (g^y \cdot g^z, m \cdot e^y \cdot e^z)=(g^{y+z}, m \cdot e^{y+z})=(g^{y'}, m \cdot e^{y'})$
  - Universal Re-encryption:
    - Idea: $Enc(m) = [\ Enc(m), Enc(1)\ ] = [\ (g^y, m \cdot e^y), (g^{y'}, e^{y'})\ ]$
    - $Enc(m)^{z,z'} = [\ Enc(m) \cdot Enc(1)^z, Enc(1)^{z'}\ ] = [\ (g^{y+y' \cdot z}, m \cdot e^{y+y' \cdot z}), (g^{y' \cdot z'}, e^{y' \cdot z'})\ ]$
      $= [\ (g^{y''}, m \cdot e^{y''}), (g^{y'''}, e^{y'''})\ ]$

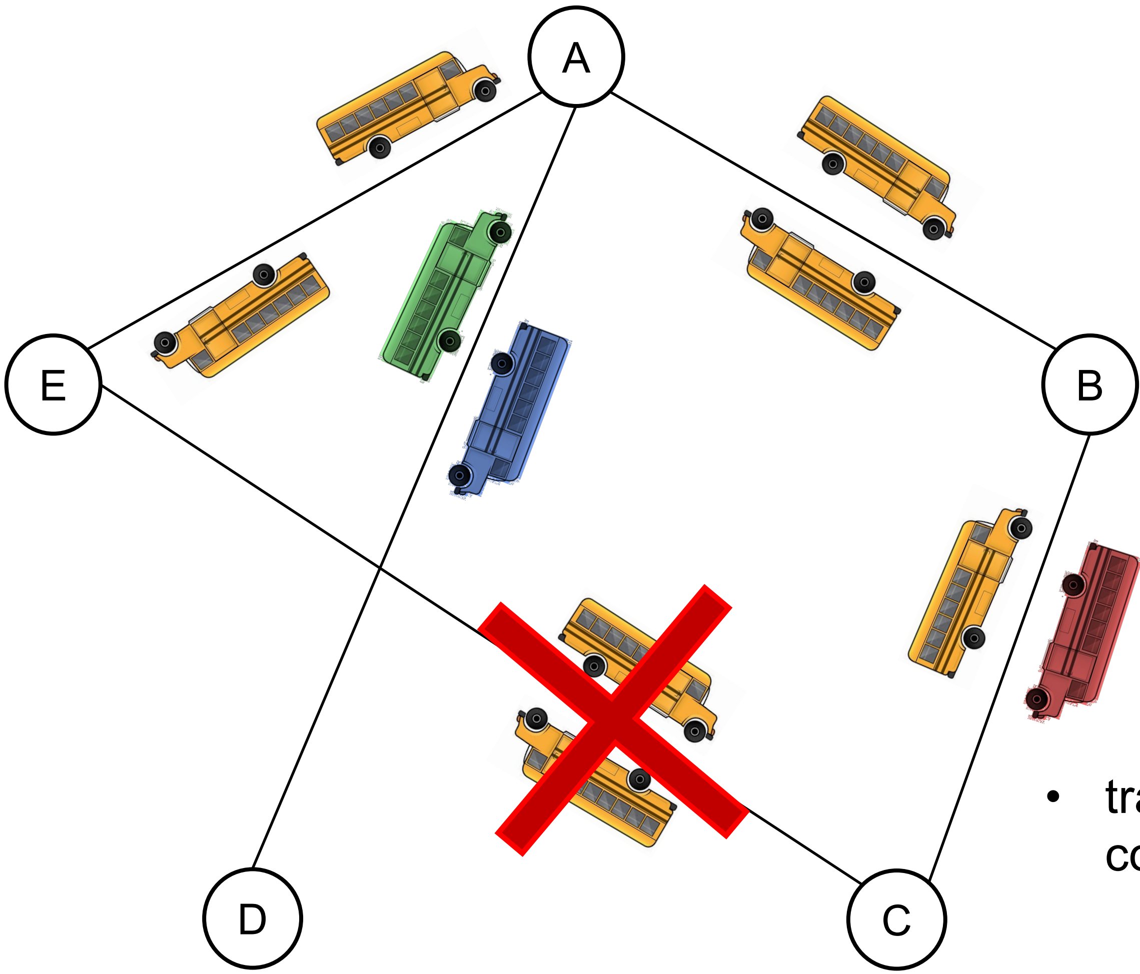
(Threshold) Proxy Re-encryption

- Proxy Re-encryption:
  - given: c=Enc(*e*,m), *e*'
  - create: c'=Enc(*e'*,m)
  - ➔ 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 a 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

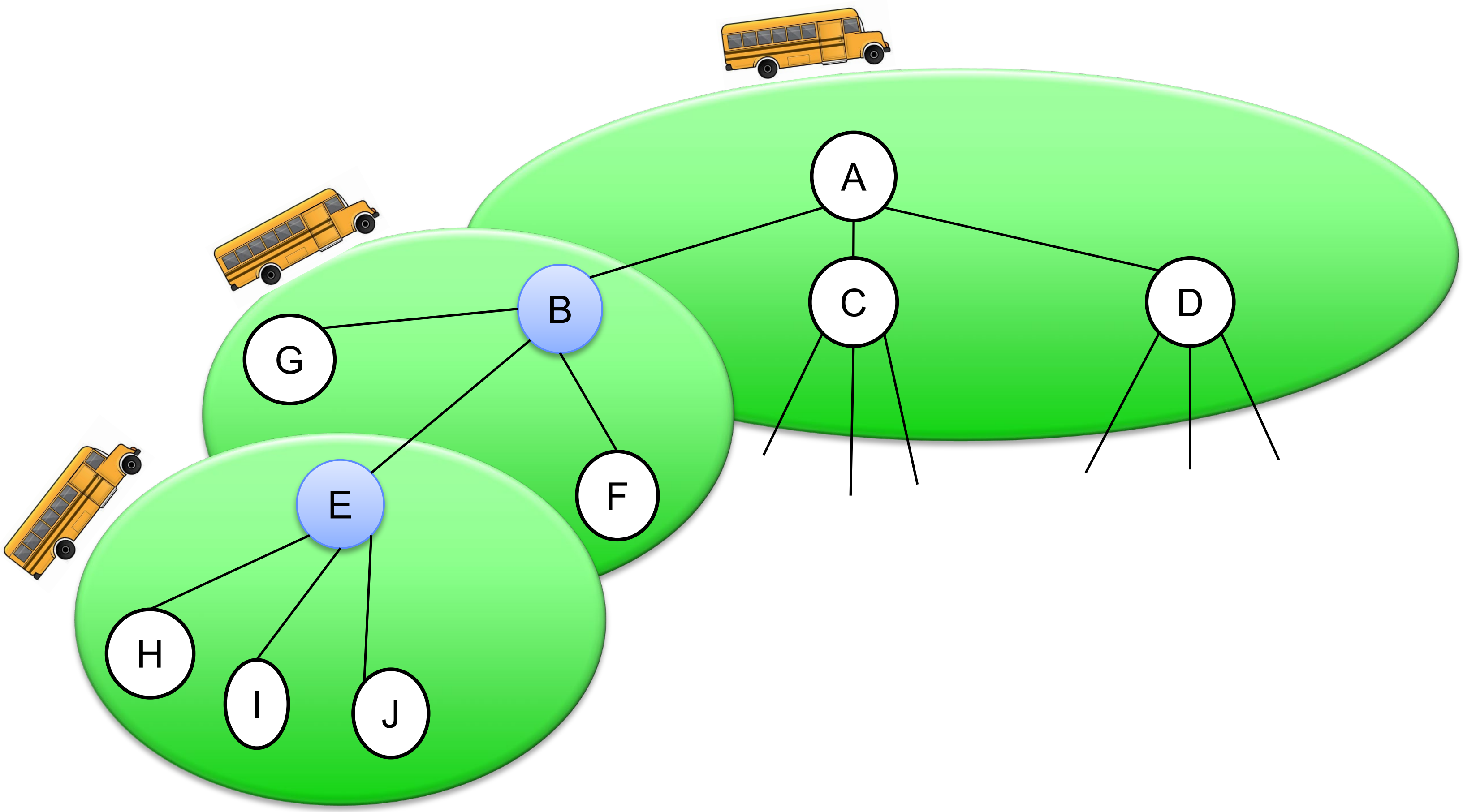
? seats → e.g. shortest path B to E not unique

- tradeoff: time vs. communication complexity

  ➔ spanning subgraph sufficient

Buses – time and communication tradoff

- Idea: partition graph into clusters, have one bus per cluster

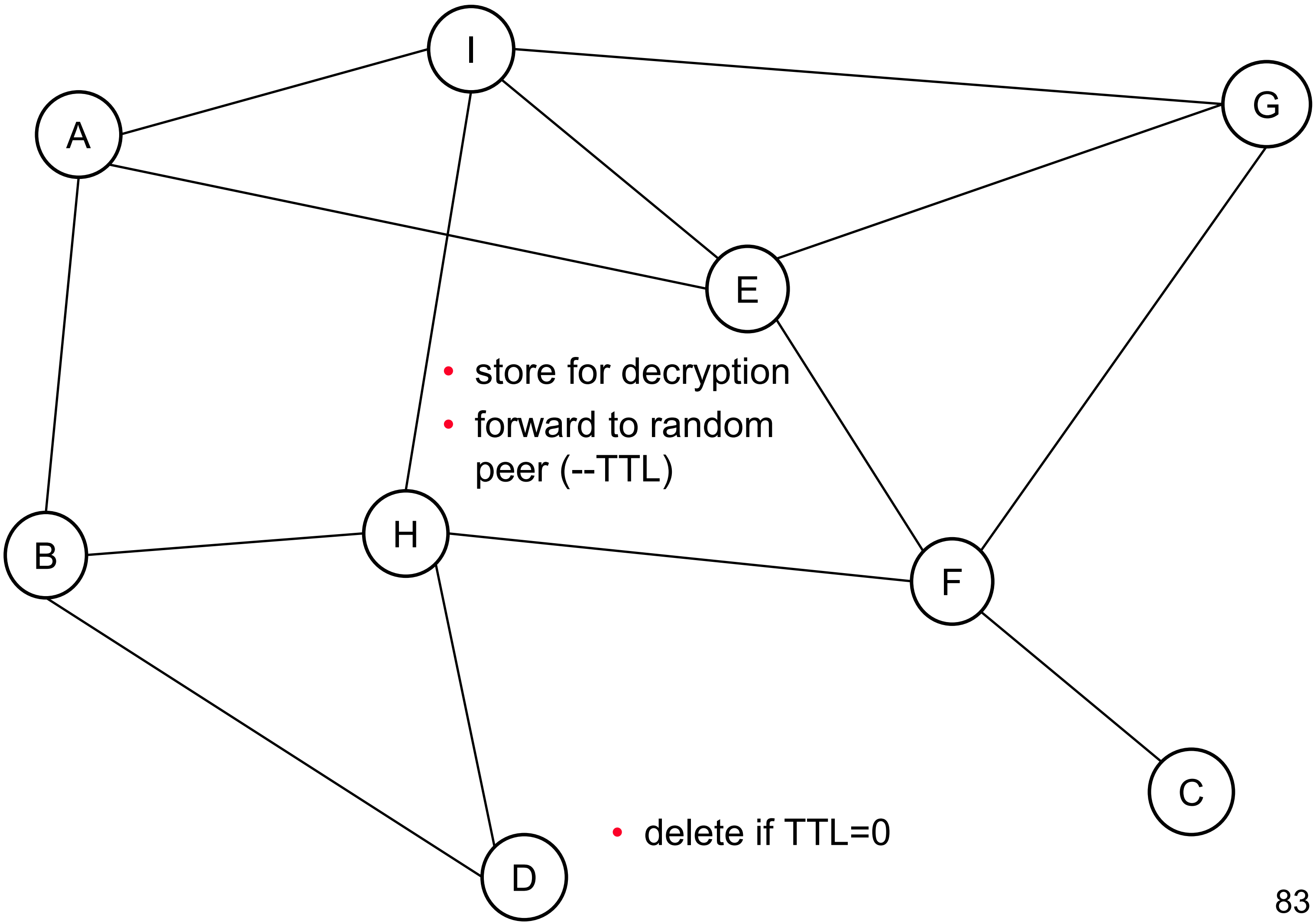


- achieves sender and recipient anonymity
- basic building blocks:
  - random walk through peer graph
    - simulates broadcast
  - invisible implicit addressing
  - dummy messages
  - strict synchronisation
    - mitigates timing attacks

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

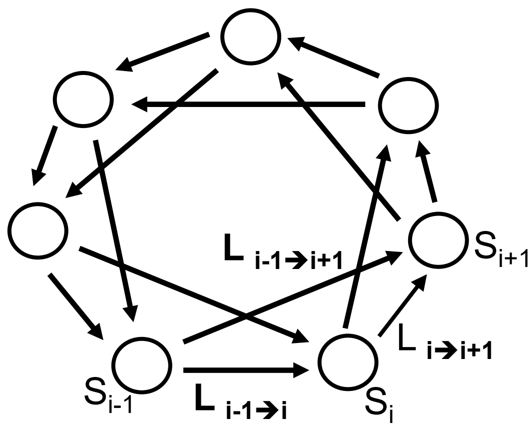
Anonymity: little global information
Fault tolerance: much global information

Principles

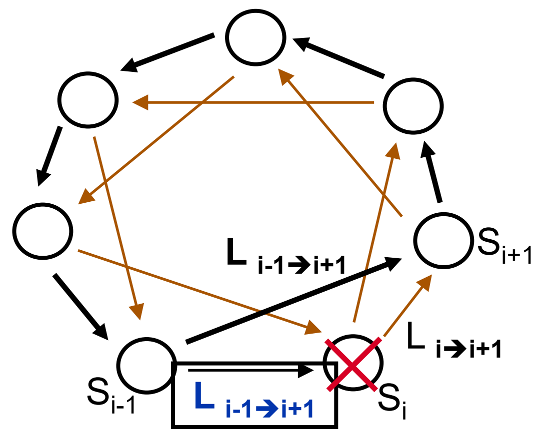
Fault tolerance through weaker anonymity in a single operational mode (anonymity-mode)

Fault tolerance through a special operational mode (fault tolerance-mode)

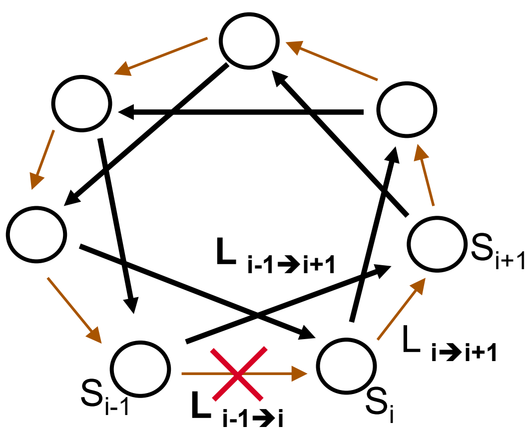
Braided RING



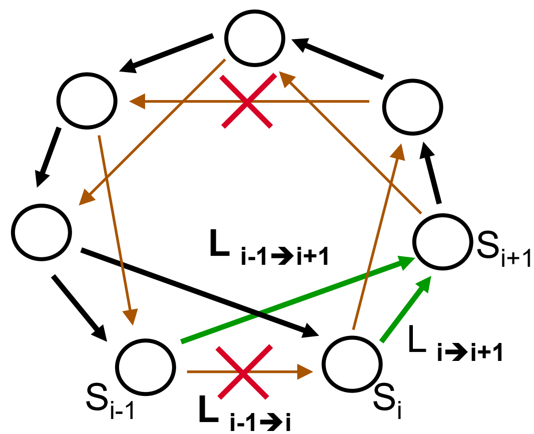
Two RINGs operating if no faults



Reconfiguration of the outer RING if a station fails



Reconfiguration of the inner RING if an outer line fails



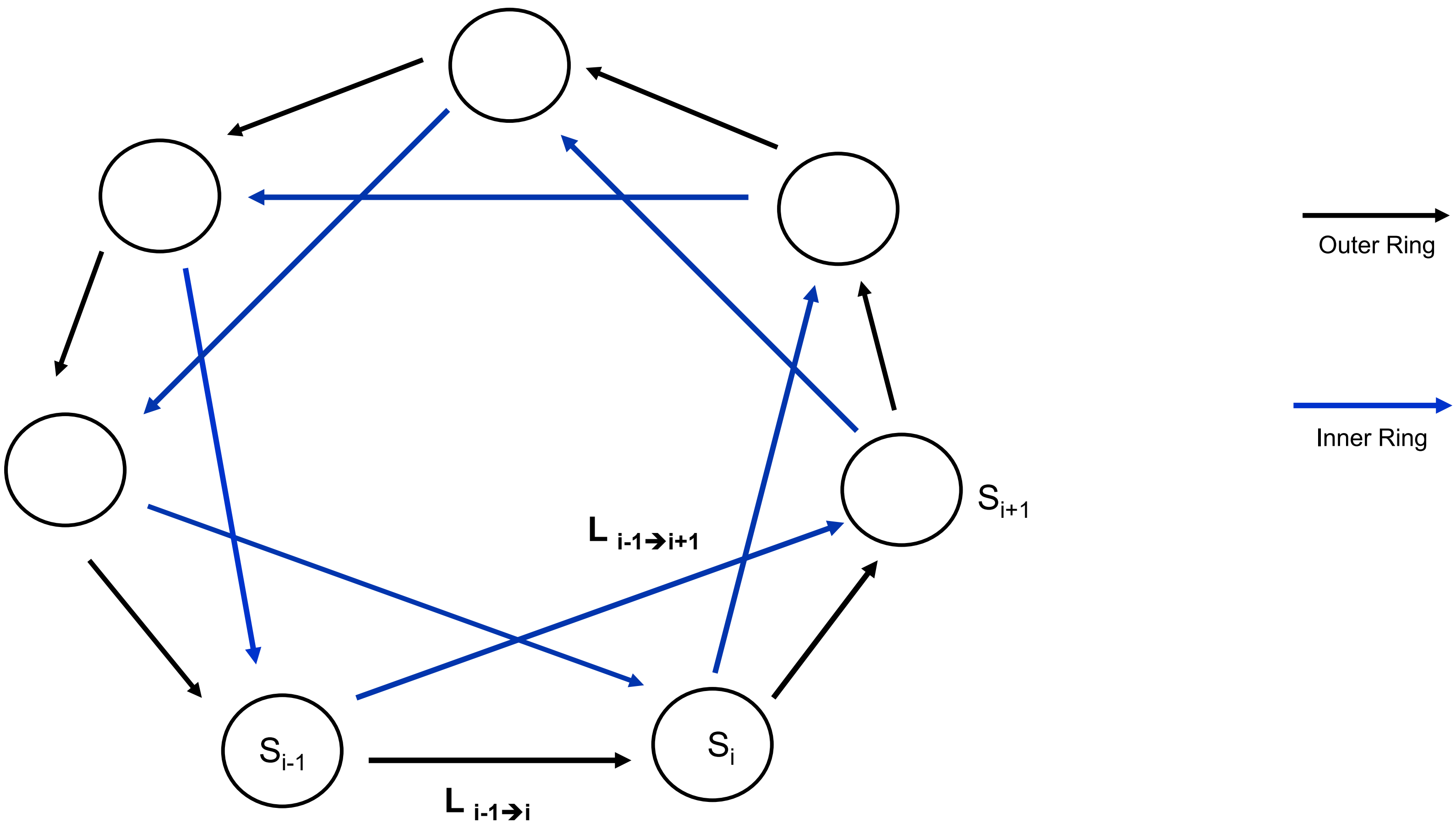
Reconfiguration of the outer RING if an outer and inner line fails

Line used

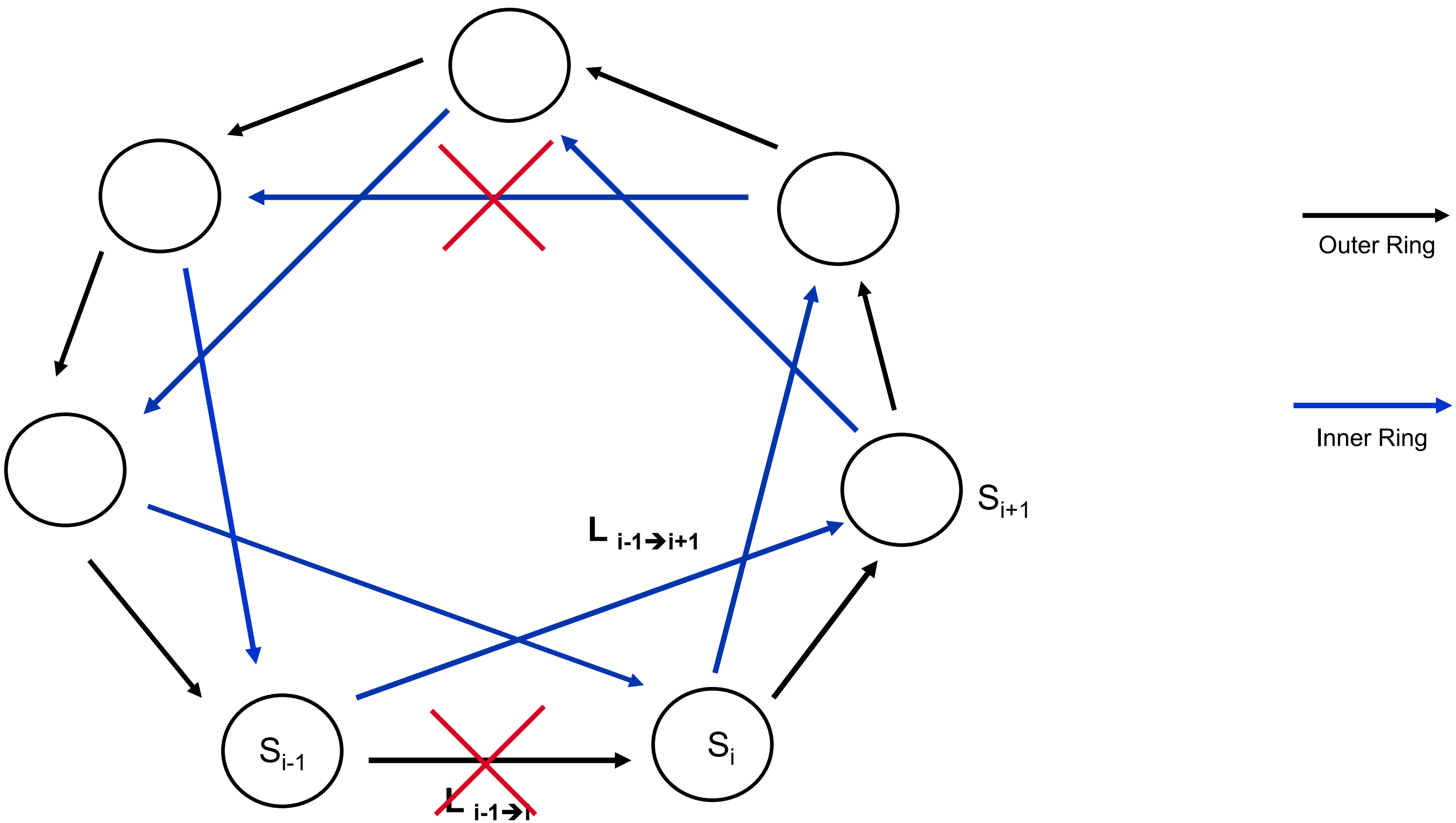
Line not used

Line used to transmit half of the messages

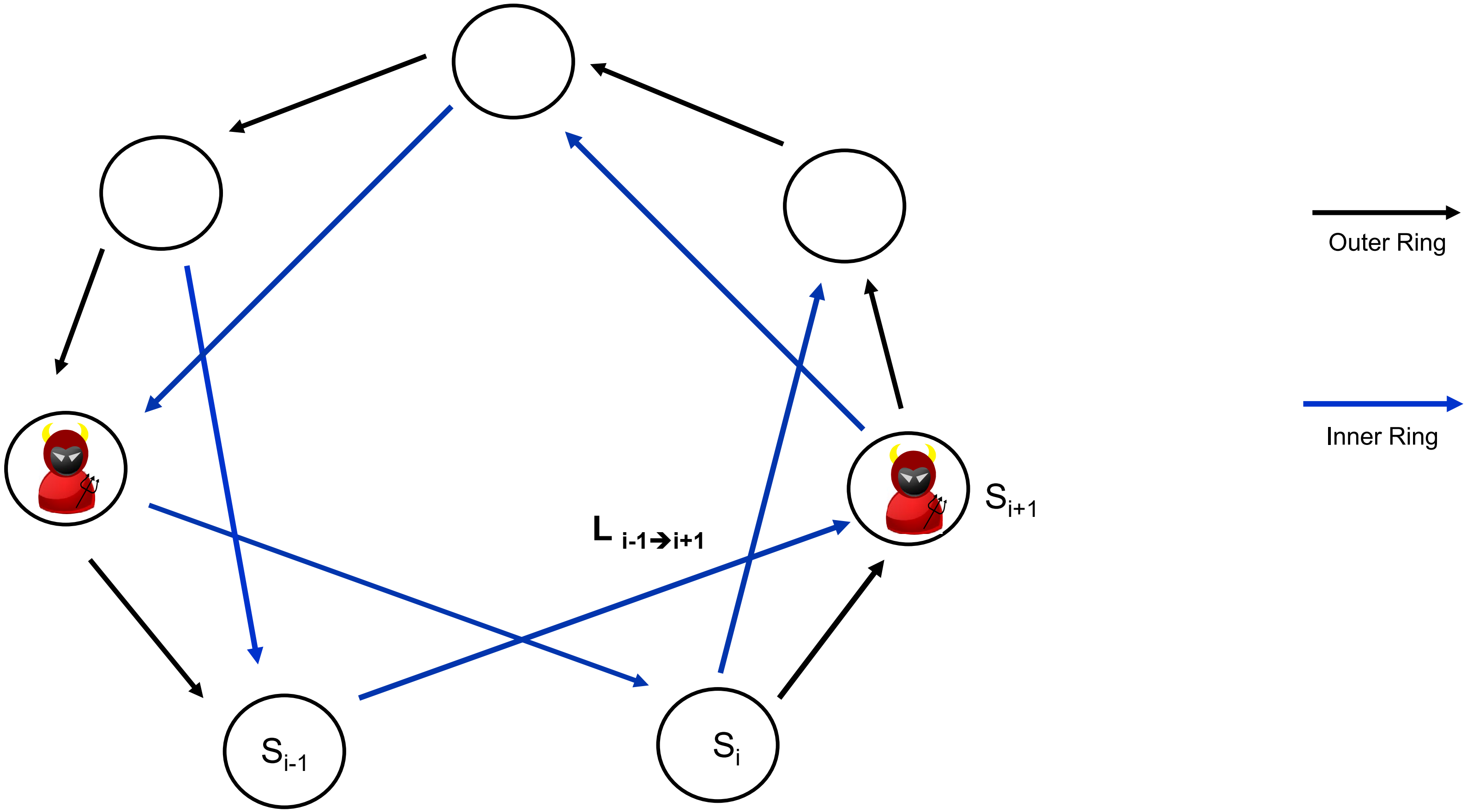
Braided RING



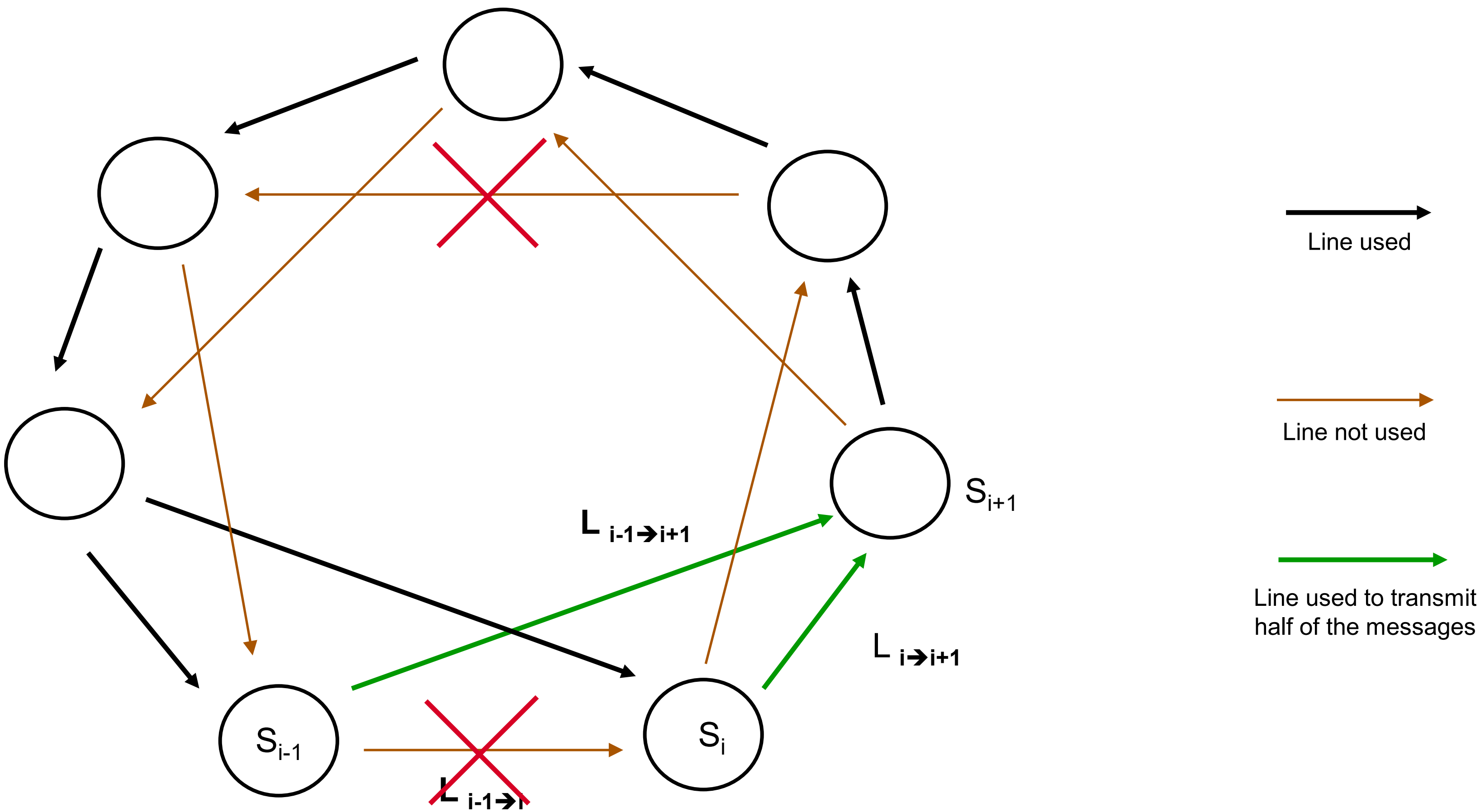
Braided RING



Braided RING

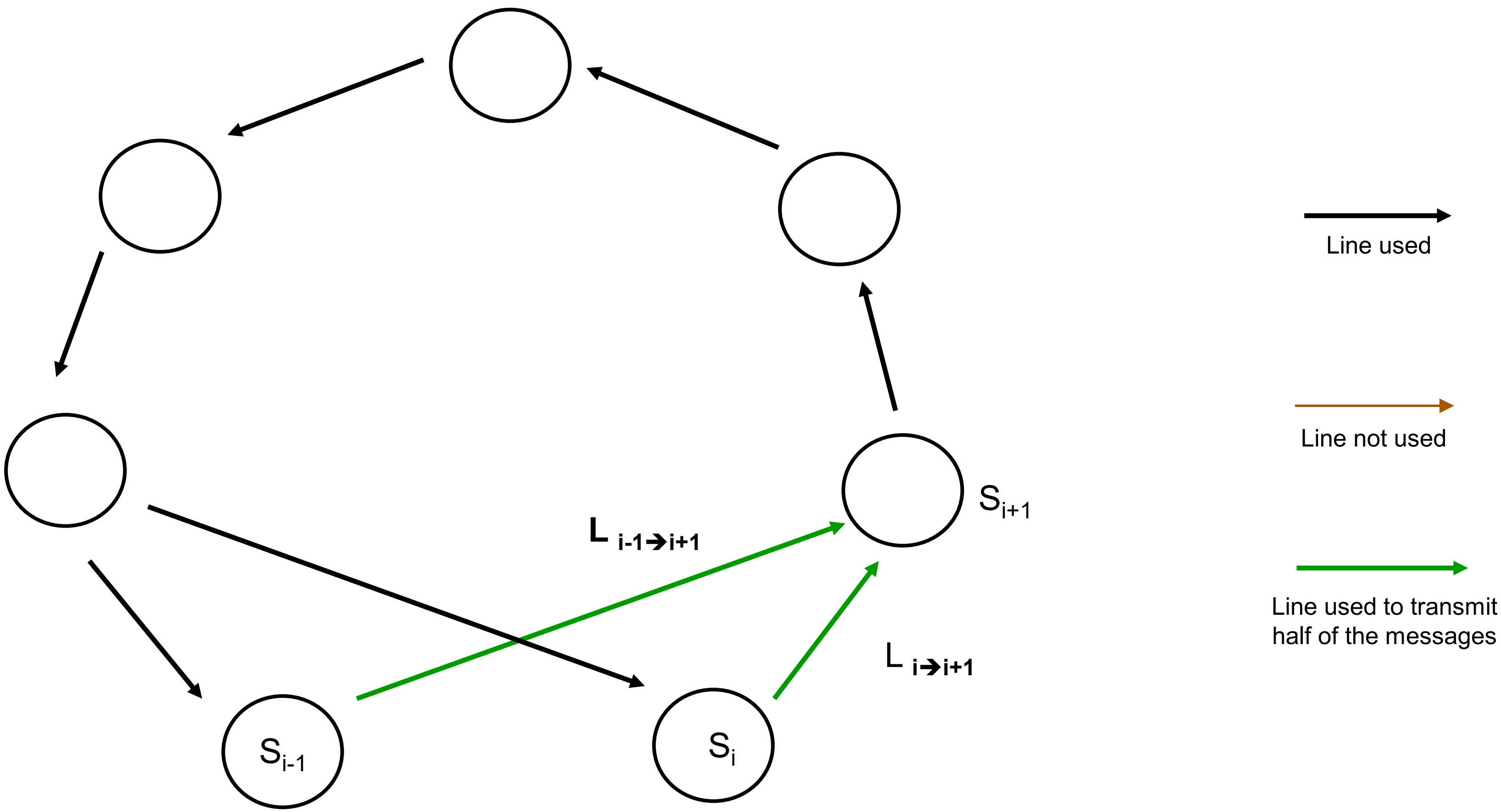


Braided RING



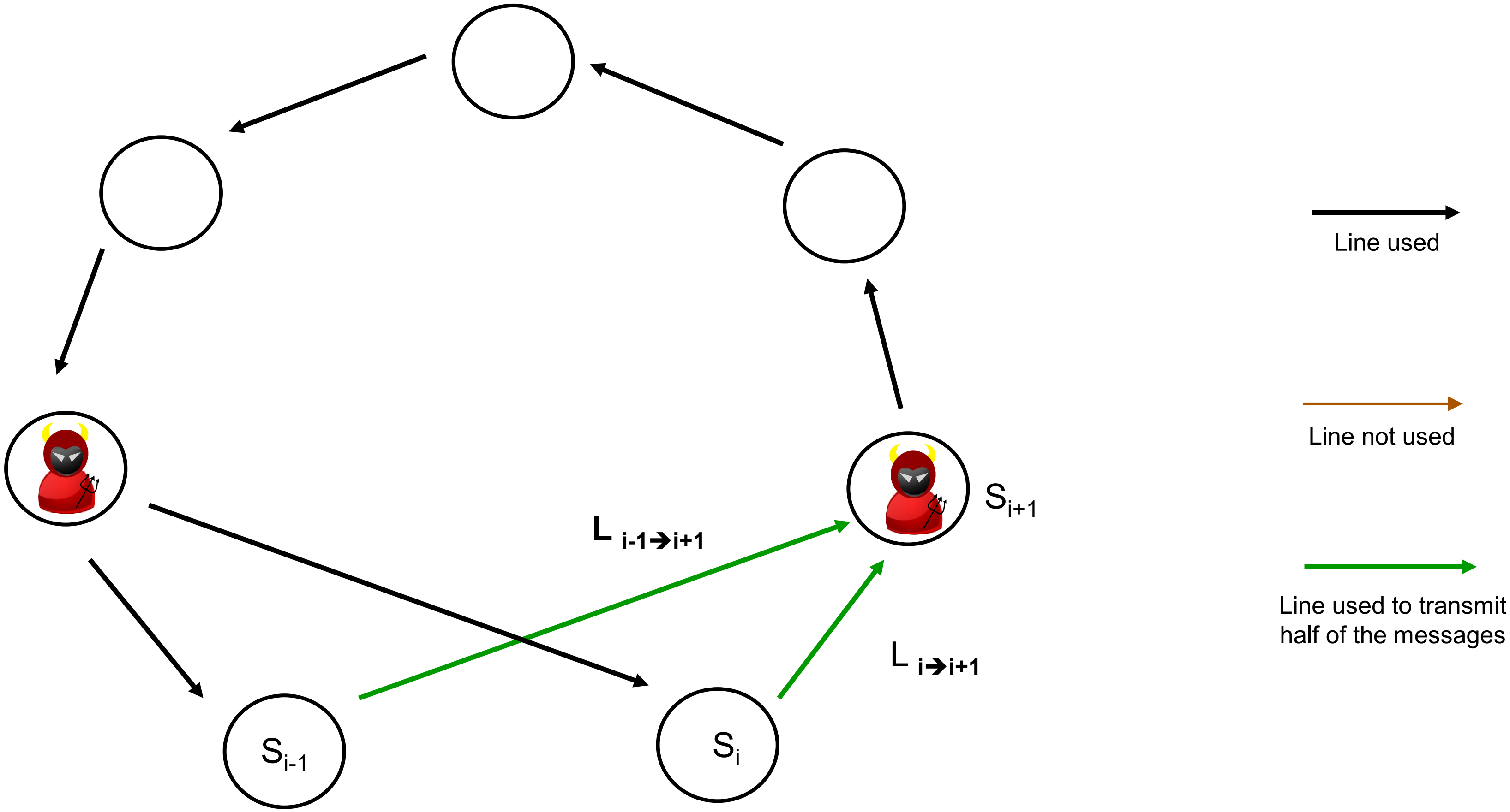
Reconfiguration of the outer RING if an outer and inner line fails

Braided RING



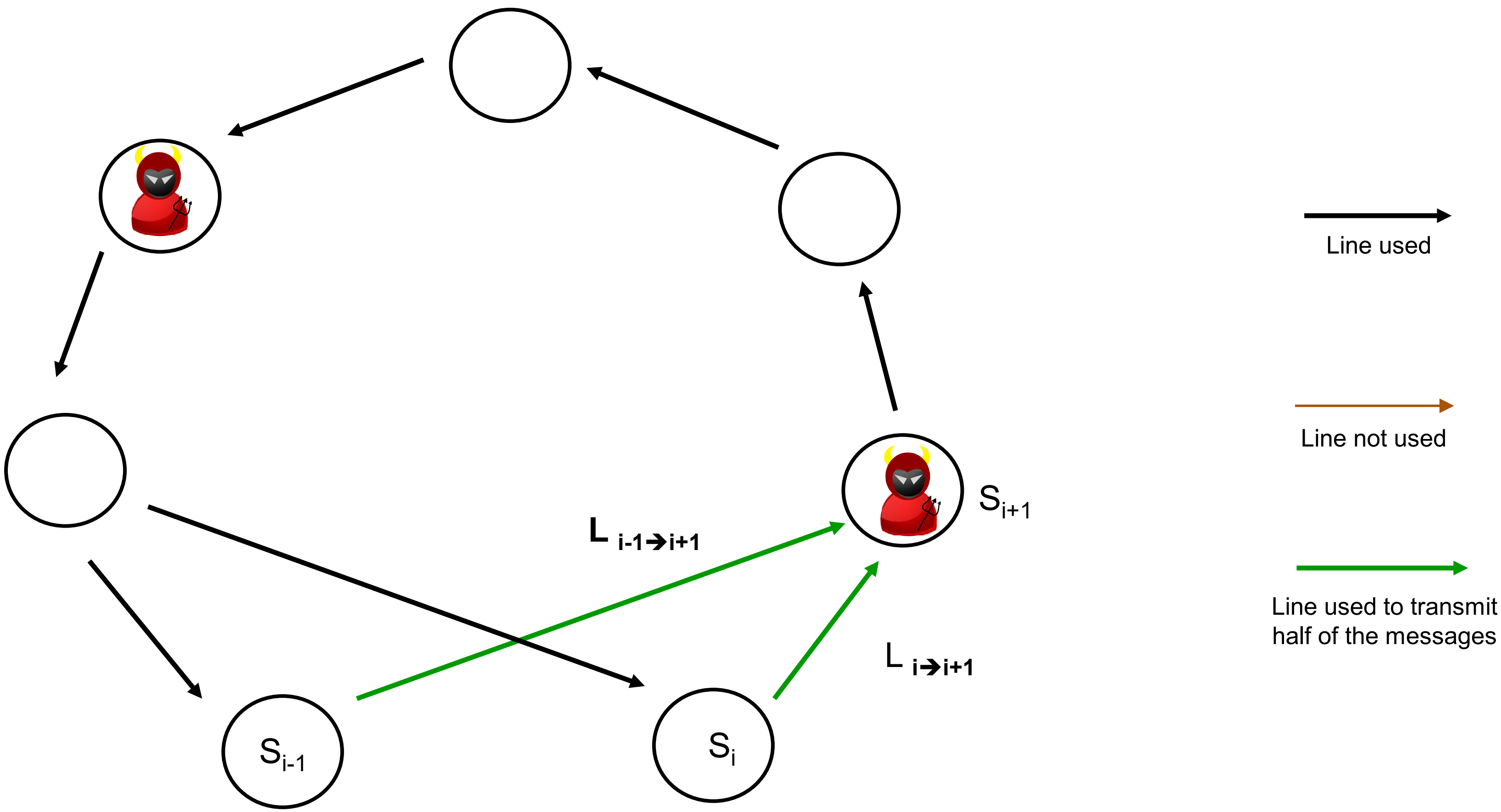
Reconfiguration of the outer RING if an outer and inner line fails

Braided RING



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

Braided RING



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

Modifying attacks

modifying attacks at

covered in RING-network by attacker model

sender anonymity

→ extend the access method

recipient anonymity

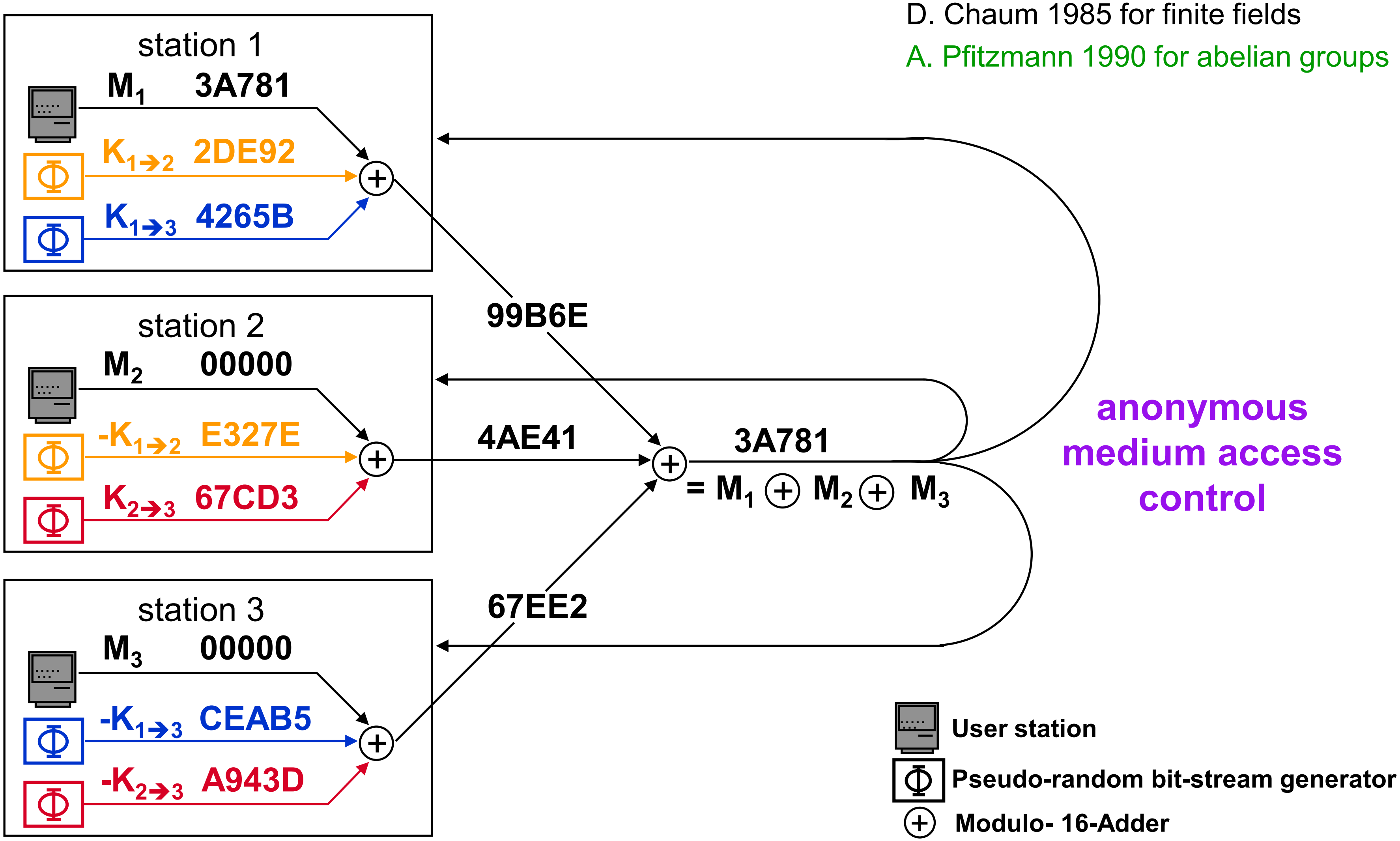
service delivery

publish input and output
if dispute: reconfiguration

Superposed sending (DC-network)

D. Chaum 1985 for finite fields

A. Pfitzmann 1990 for abelian groups



Anonymity of the sender

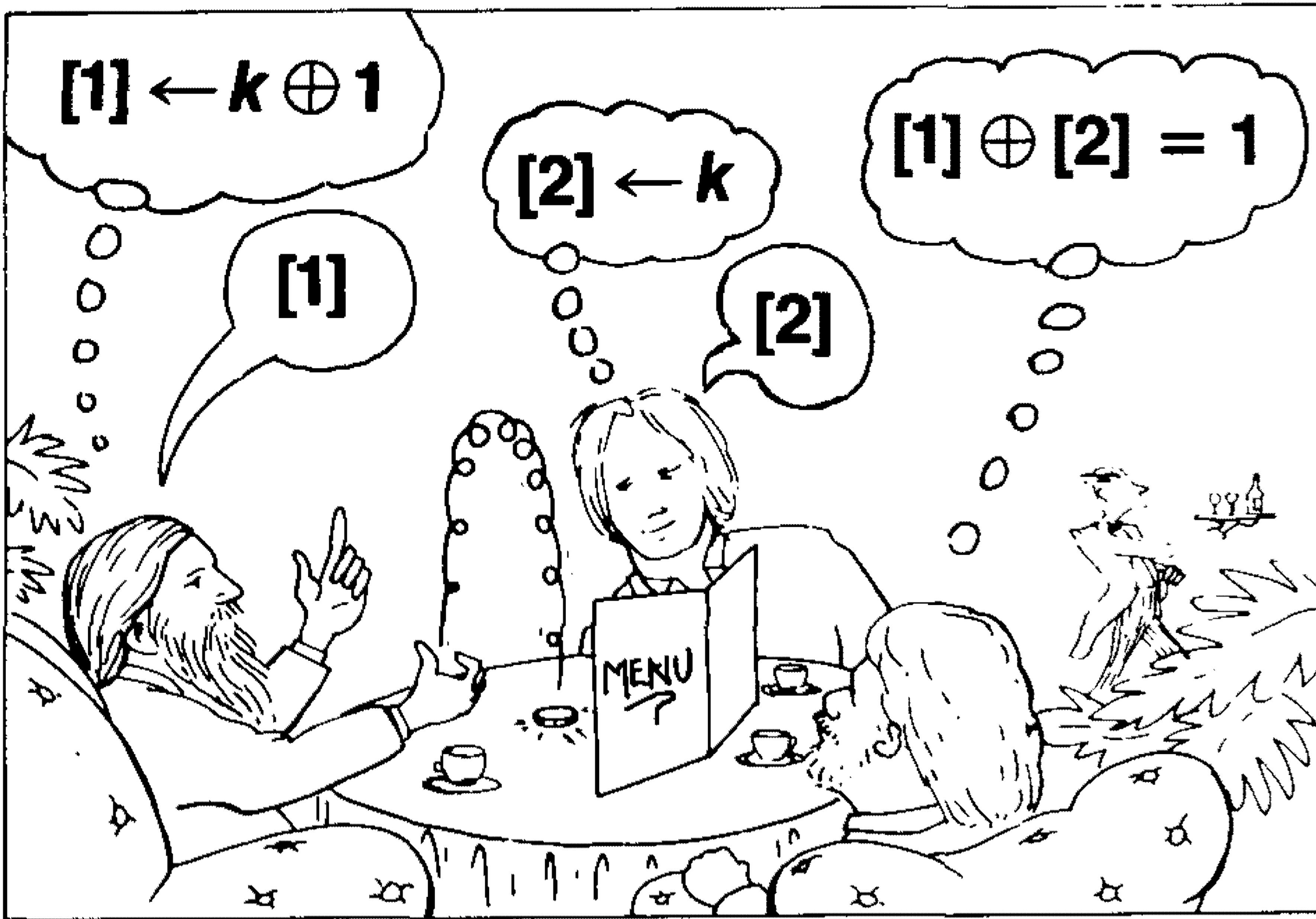
If stations are connected by keys the value of which is completely unknown to the attacker, tapping all lines does not give him any information about the sender.

Dinning Cryptographers

[D. Chaum: „*Security without identification: transaction systems to make big brother obsolete*“, Communications of the ACM, Volume 28, Issue 10, Oct. 1985]

Dinning Cryptographers

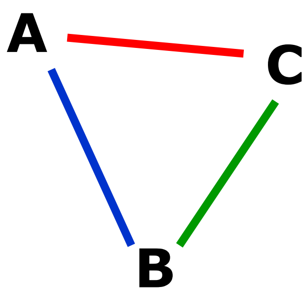
[D. Chaum: „*Security without identification: transaction systems to make big brother obsolete*“, Communications of the ACM, Volume 28, Issue 10, Oct. 1985]



DC-Net – Superposed Sending

Chaum, 1988

Key Graph



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

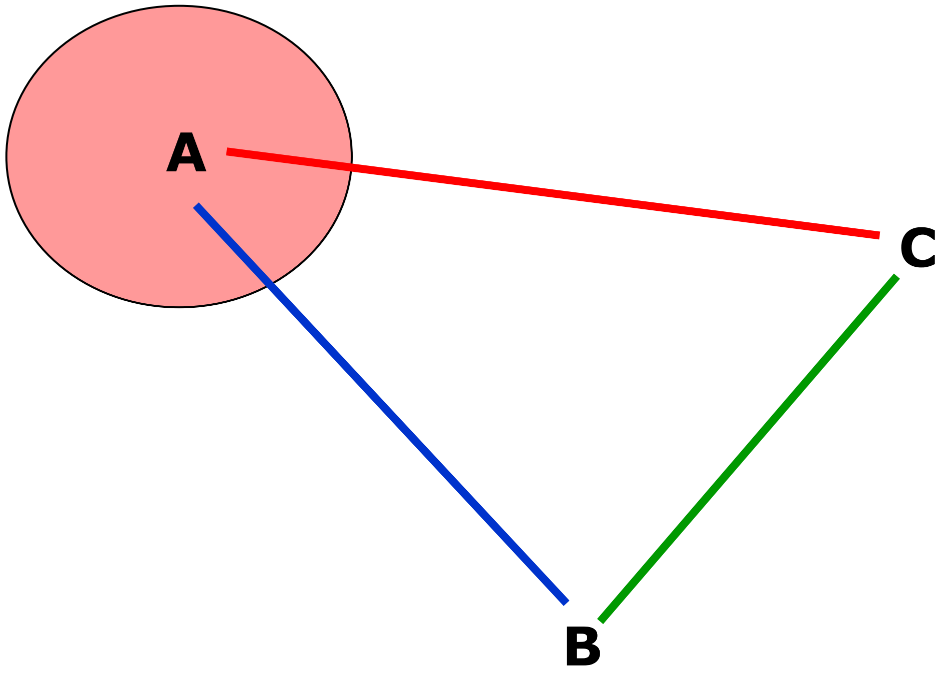
C sends 01011001

Note: In this example "sum" means XOR

Sum = True Message from A 00110101

Security Analysis w.r.t. Insider for 3 Participants

Key Graph



A’s perspective: Degraded DC-Net

B —————————— C

$$L_B = m \oplus k$$

$$G = L_B \oplus L_C = 1$$

$$G = m \oplus k \oplus \bar{m} \oplus k$$

$$G = m \oplus \bar{m}$$

$$G = 1$$

$$L_C = \bar{m} \oplus k$$

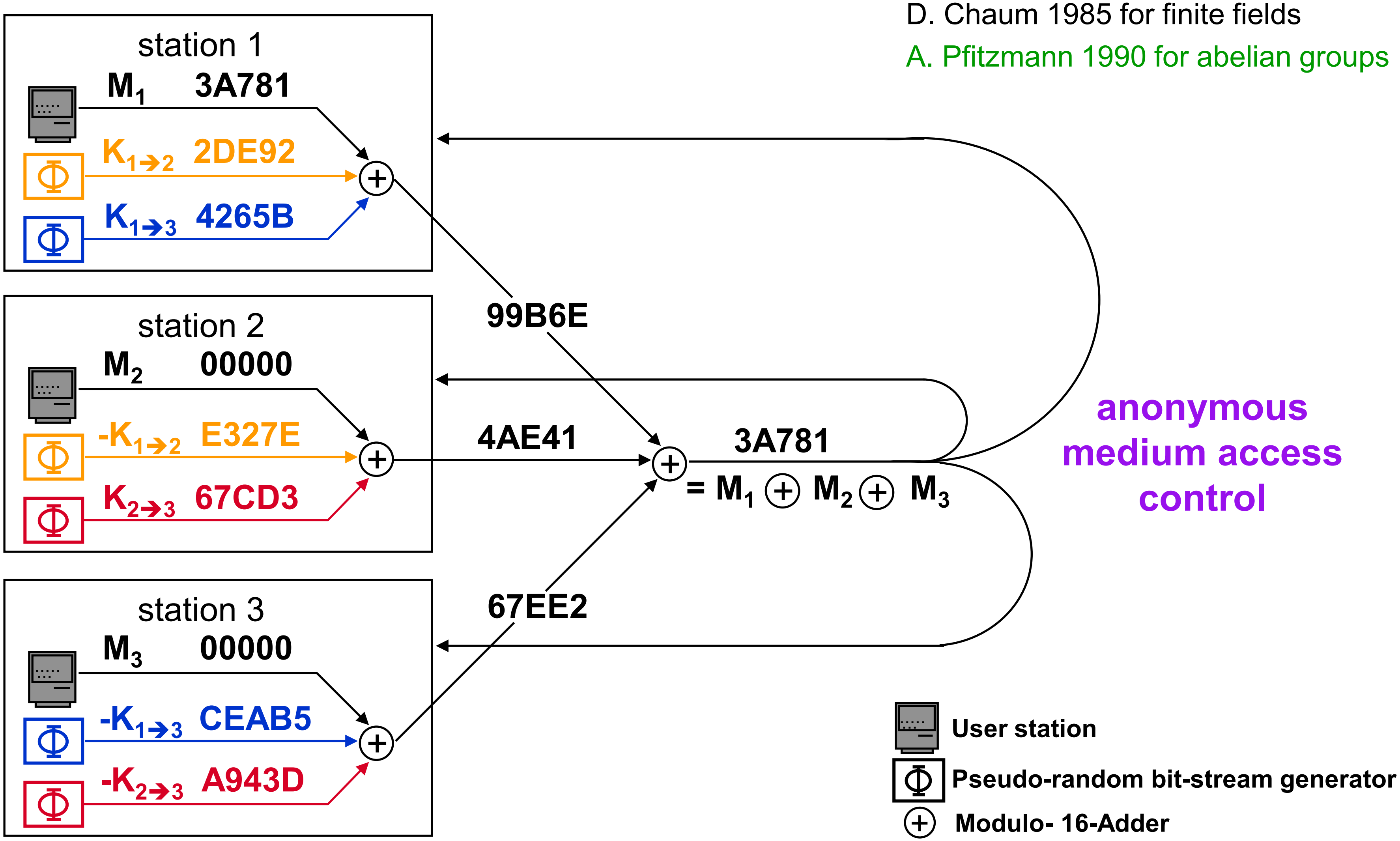
L_B	m_B	k
0	0	0
0	1	1
1	0	1
1	1	0

L_C	m_C	k
1	1	0
1	0	1
0	1	1
0	0	0

Superposed sending (DC-network)

D. Chaum 1985 for finite fields

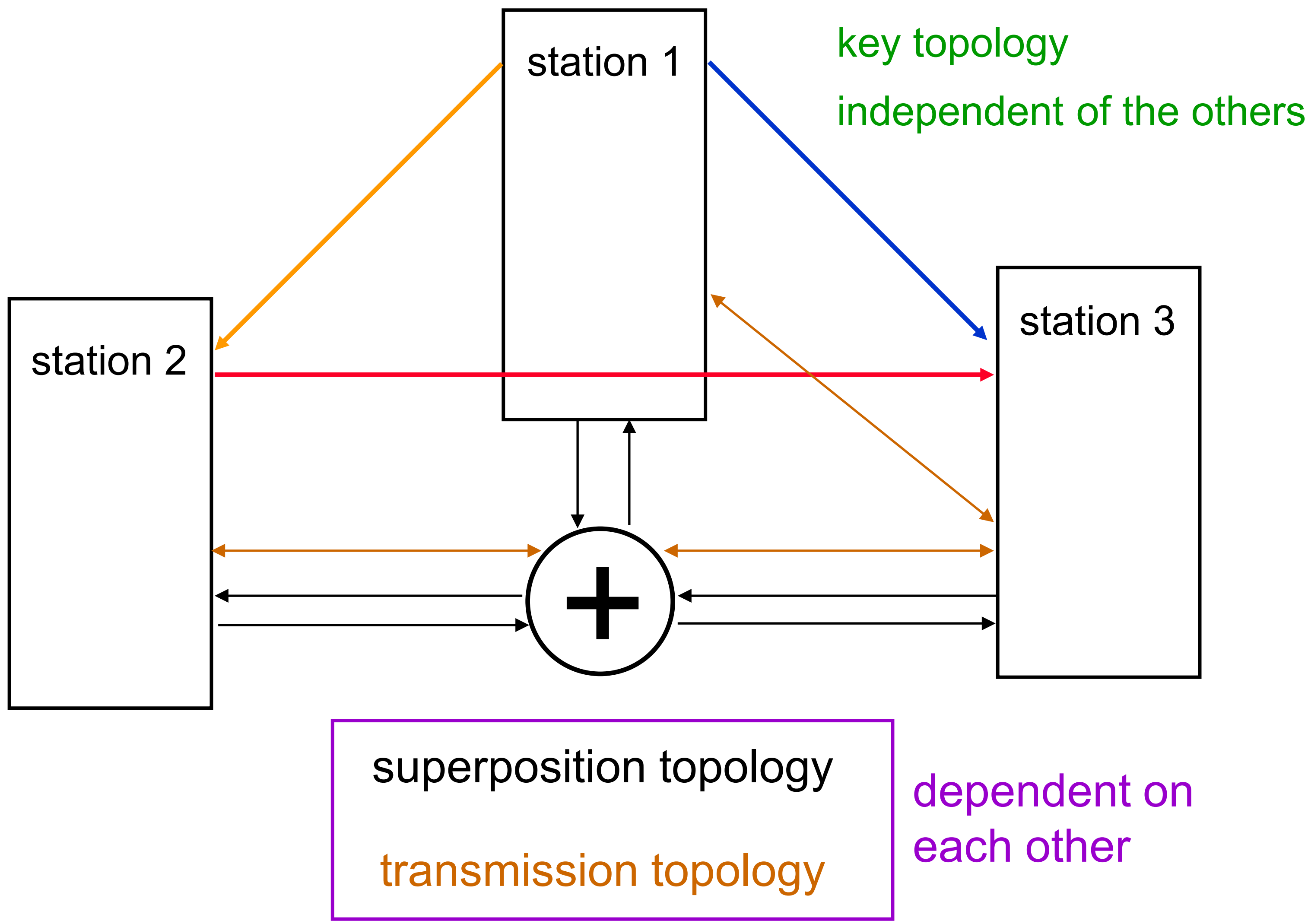
A. Pfitzmann 1990 for abelian groups



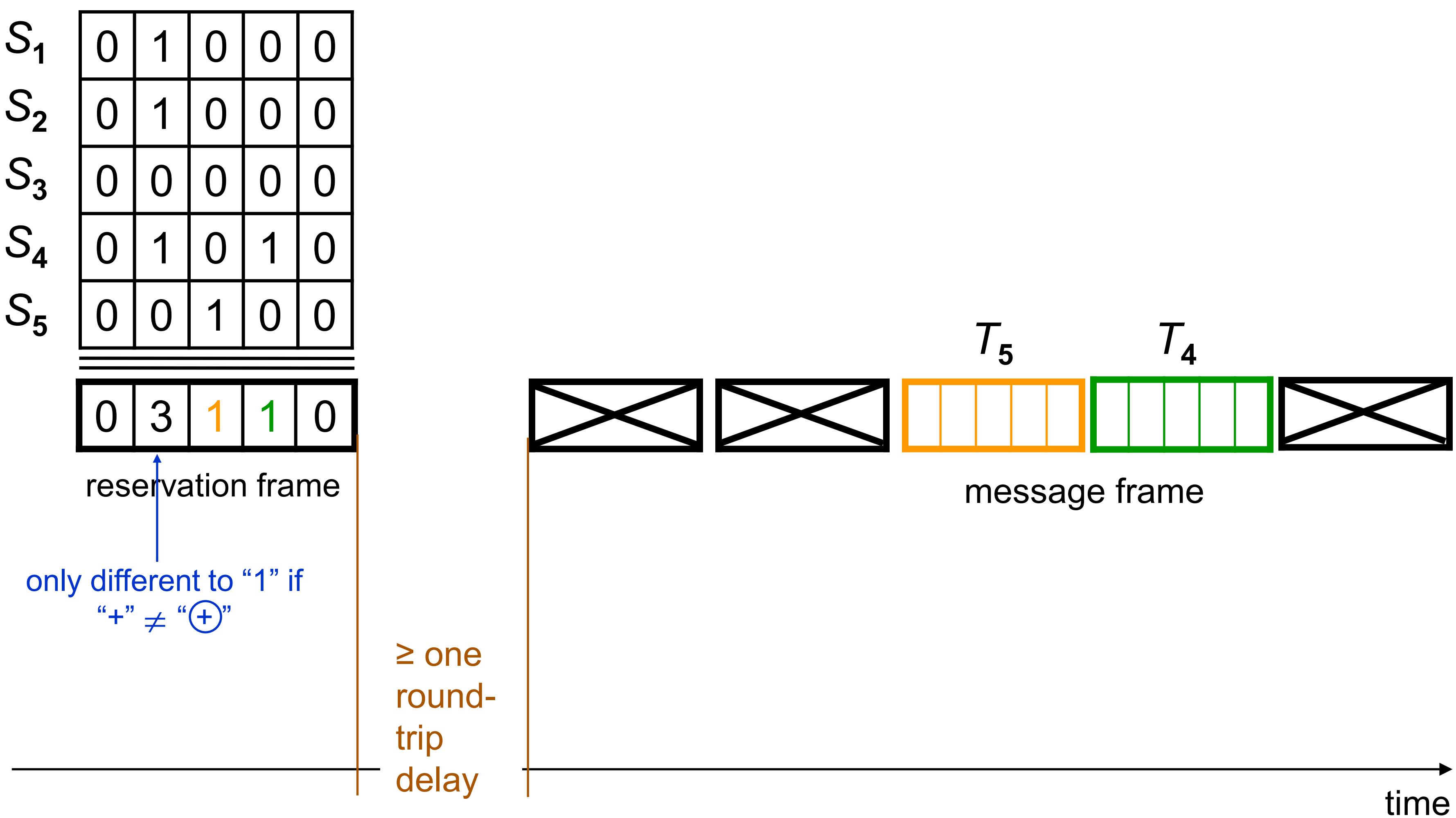
Anonymity of the sender

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



Reservation scheme



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.
==> Duplex channel in the bandwidth of a simplex channel

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

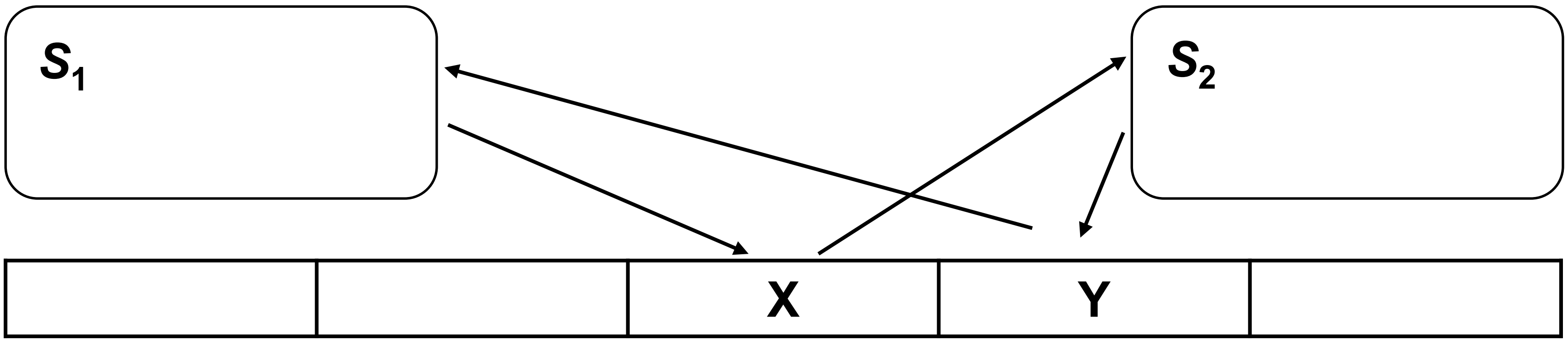
Result of a collision is stored, so that if *n* messages collide, only *n*-1 have to be sent again.

Collision resolution algorithm using the mean of messages:

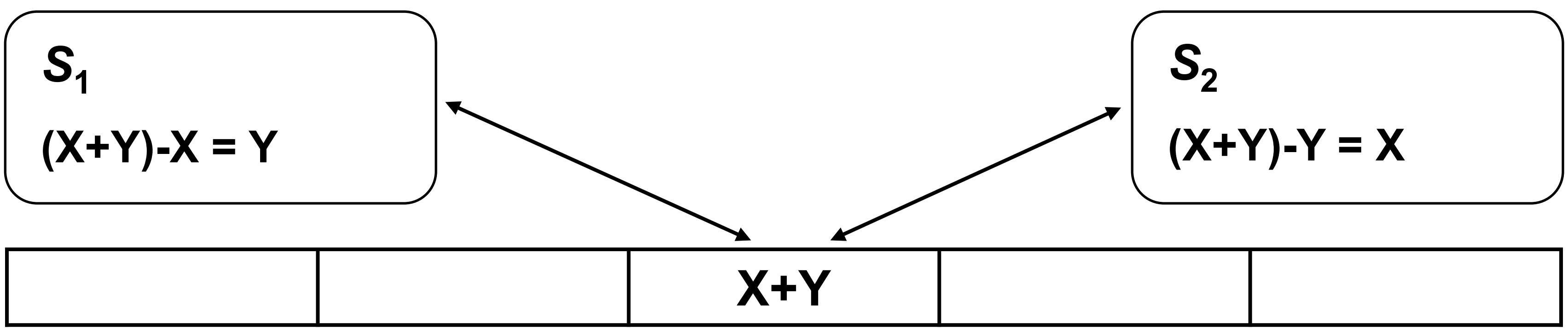
$\leq 2^S - 1$ station

addition mod 2^L

Pairwise superposed receiving

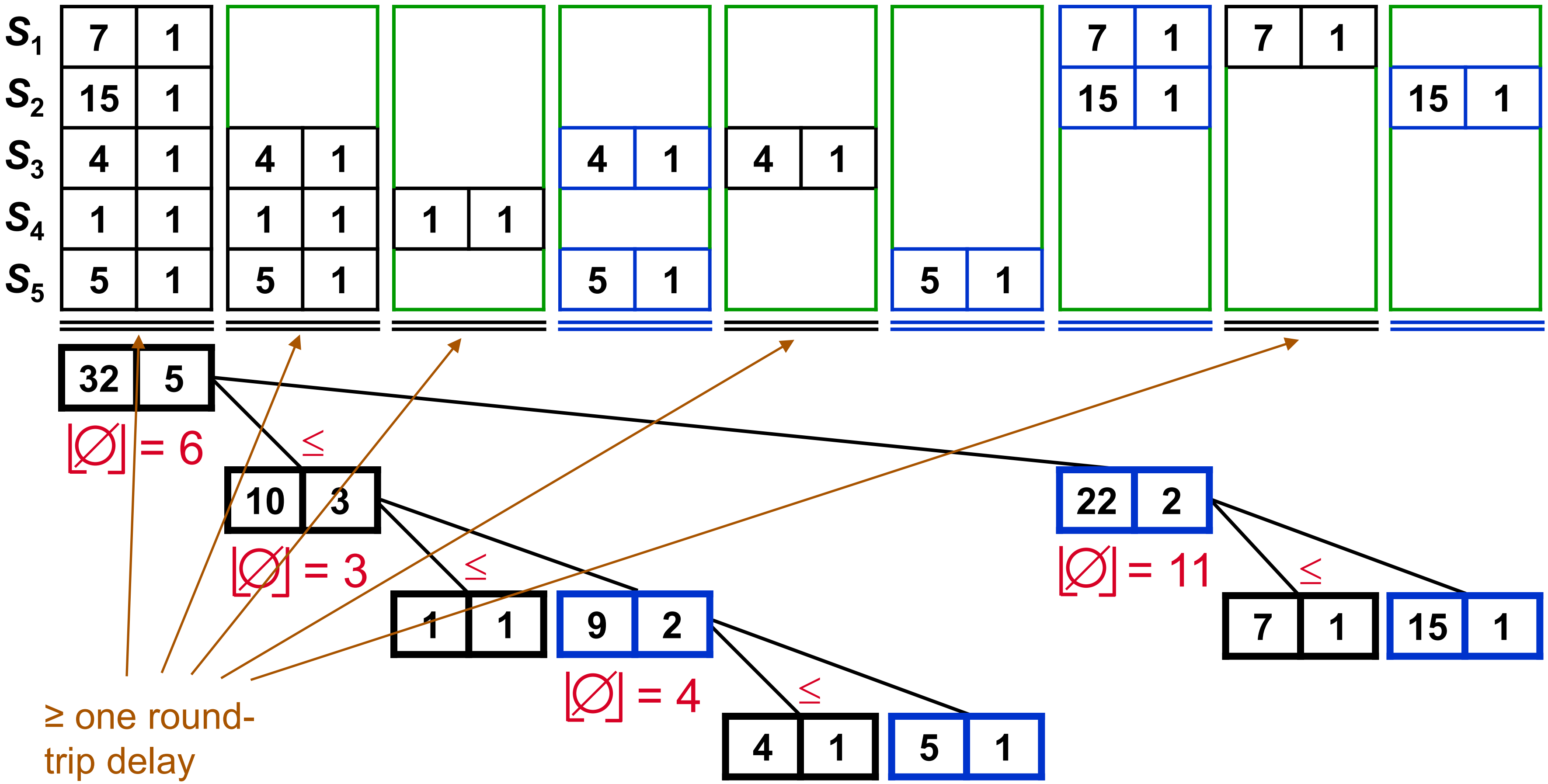


without superposed receiving



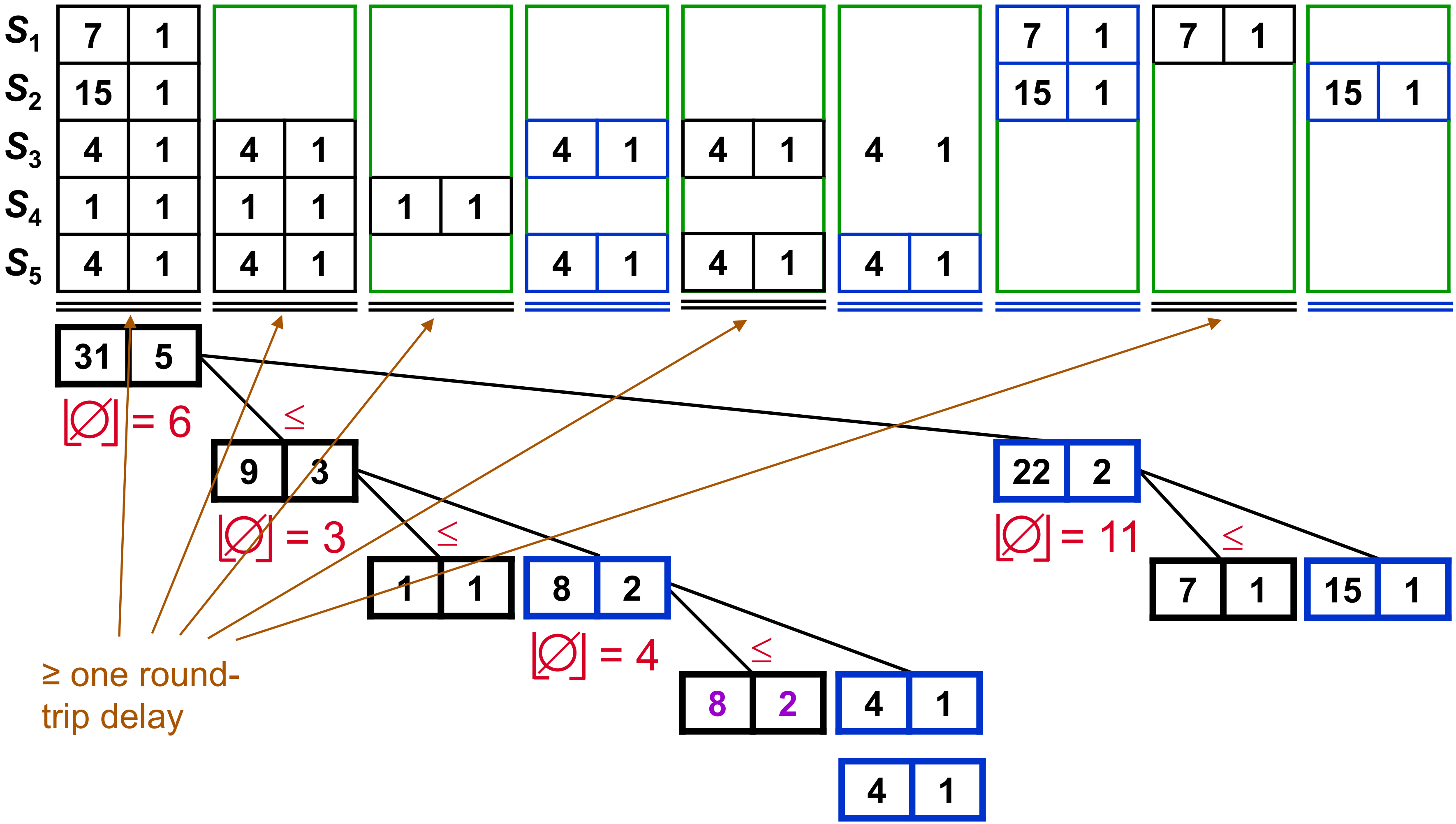
with pairwise superposed receiving

Global superposed receiving



Collision resolution algorithm with mean calculation and superposed receiving

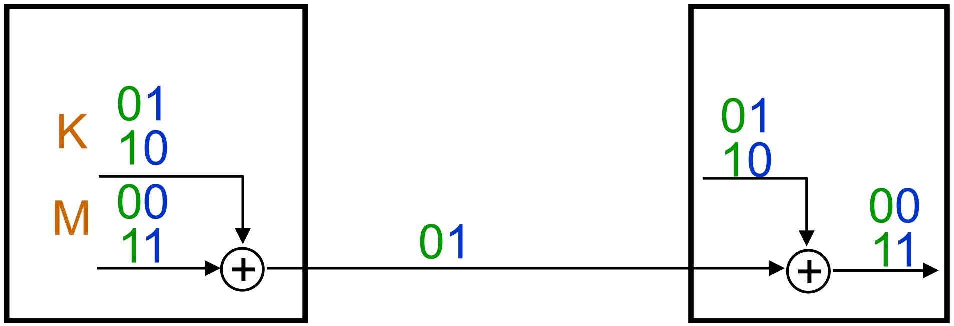
Global superposed receiving (2 messages equal)



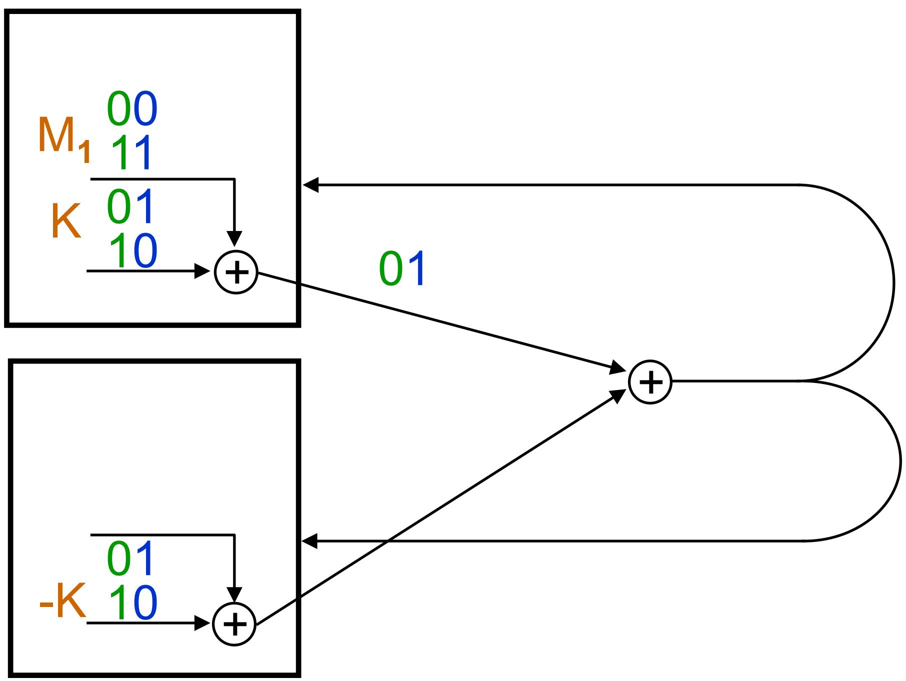
Collision resolution algorithm with mean calculation and superposed receiving

Analogy between Vernam cipher and superposed sending

Vernam cipher



$K + M = C \Leftrightarrow M = C - K$ abelian group



$M_1 + K = O_1$

$M_2 - K = O_2$

Proof of sender anonymity: proposition and start of induction

Proposition:

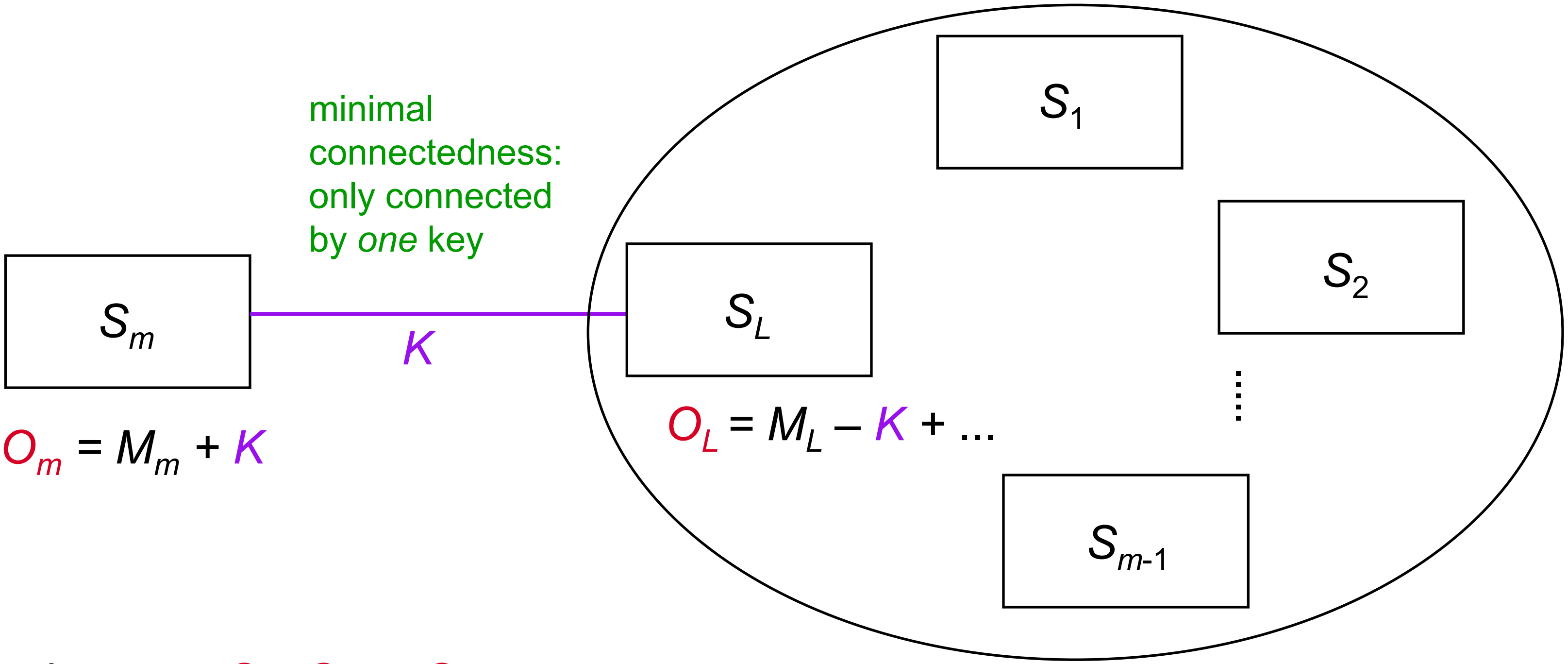
If stations S_i are connected by uniform randomly distributed keys K_j which are unknown to the attacker, by observing all the O_i, the attacker only finds out $\sum_i M_i$ about the M_i.

Proof:

m=1, trivial

step m-1 ➔ m

Proof of sender anonymity: induction step

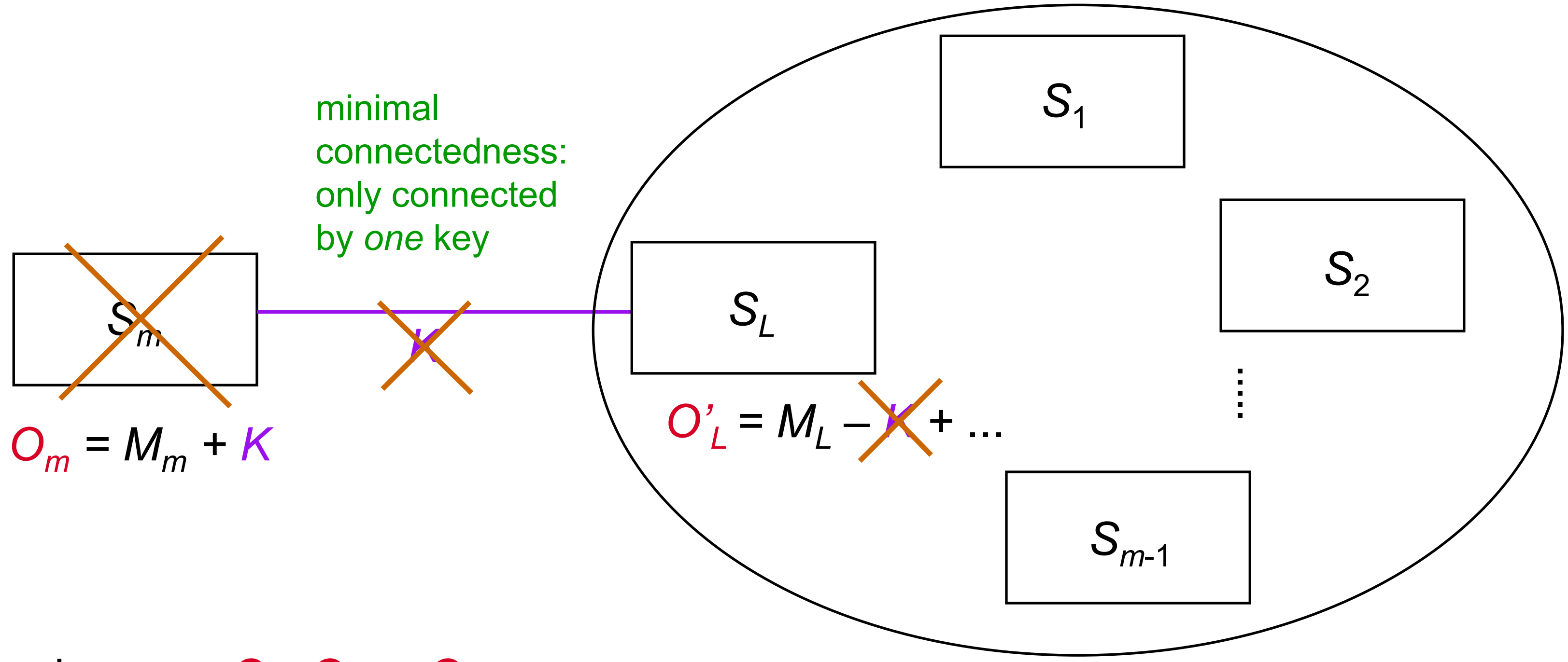


Attacker observes $O_1, O_2, \ldots O_m$.

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 :

- $K' := O_m - M'_m$

Proof of sender anonymity: induction step



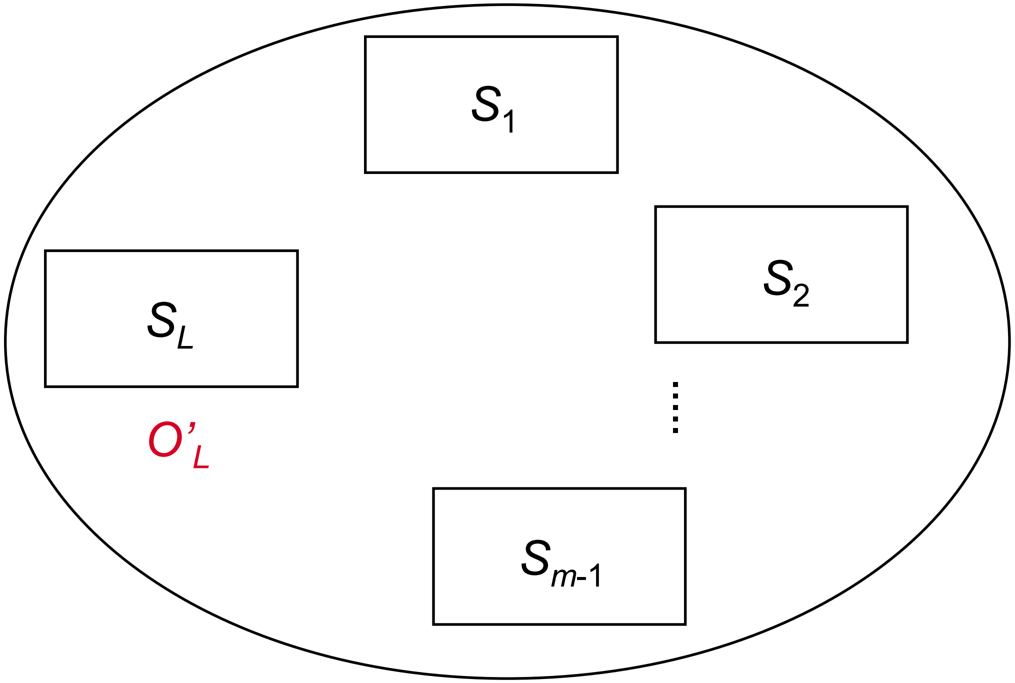
Attacker observes $O_1, O_2, \ldots O_m$.

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 O'_L of S_L is taken as: $O'_L = O_L - K'$.

Proof of sender anonymity: induction step



Attacker observes O_1, O_2, ...O'_L.

For each combination of messages M'_1, M'_2, ... M'_{m-1} with $\sum_{i=1}^{m} M'_i = \sum_{i=1}^{m} O_i$ there is exactly one compatible combination of keys.

Information-theoretic anonymity in spite of modifying attacks

Problems:

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

2) Attack on Availability: 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

Modifying attacks at

- sender anonymity
- recipient anonymity
- service delivery

attacker sends message character $\neq 0$,

if the others send their message character as well

➔ 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.

Blob := committing to 0 or 1, without revealing the value committed to

binding

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

secrecy

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 α of Z^*_p (multiplicative group mod p). Using y everybody can calculate α^y mod p.

The inverse can not be done efficiently!

binding: ☹ secrecy: ☺

$s \in Z^*_p$ randomly chosen

(so user cannot compute e such that $s \equiv \alpha^e$)

$x := s^b \alpha^y \bmod p$ with $0 \le y \le p-2$

commit $\xrightarrow{x}$

open $\xrightarrow{y}$

binding: ☺ secrecy: ☹

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

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

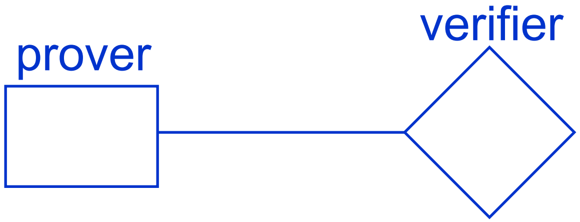
$x := \alpha^y \bmod p$

commit $\xrightarrow{x}$

open $\xrightarrow{y}$

Blobs based on factoring assumption

binding: ☹ secrecy: ☺



$n := p \cdot q$

$s := t^2 \bmod n$

$\xleftarrow{n,\ s}$

$\xleftrightarrow{s \in QR_n}$

commit

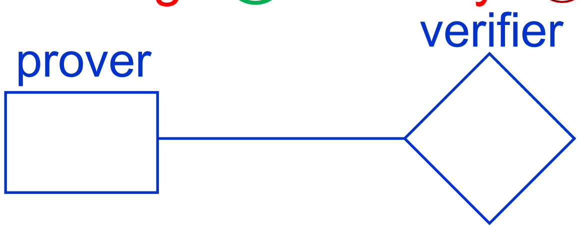
$x := y^2 s^b \bmod n$

$\xrightarrow{x}$

open

$\xrightarrow{y}$

binding: ☺ secrecy: ☹



$n := p \cdot q$

$s \notin QR_n$, $\left(\frac{s}{n}\right) = 1$

$\xrightarrow{n,\ s}$

$\xleftrightarrow{n = p \cdot q,\ s \notin QR_n}$

$x := y^2 s^b \bmod n$

$\xrightarrow{x}$

$\xrightarrow{y}$

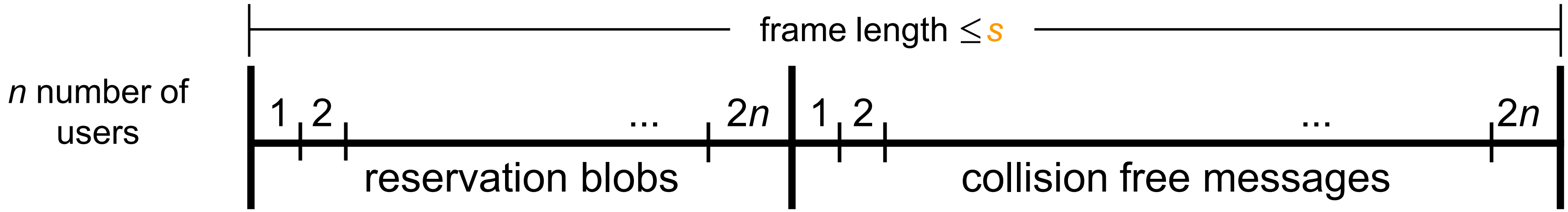
Blobs based on asymmetric encryption system

binding: ☺ secrecy: ☹

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)

Protection of the sender: anonymous trap protocol

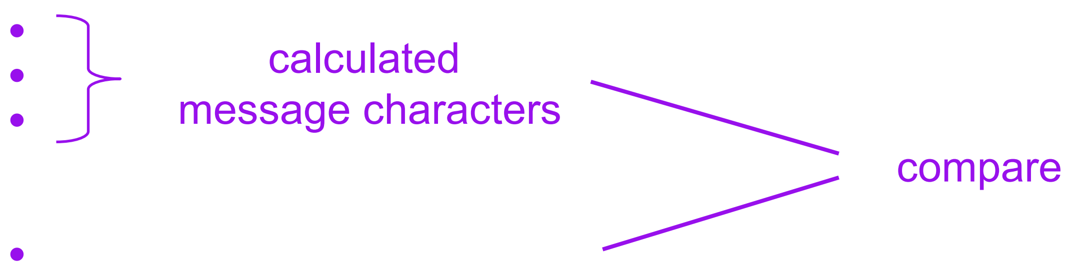


- Each user can cause investigating the reservation blobs directly after their sending if the sending of his reservation blobs did not work.
- Each user can authorize investigating of his “collision-free” random message, by opening the corresponding reservation blob.

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.)



known = known to *all* stations truthful to the protocol

Modifying attacks in the reservation phase

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.

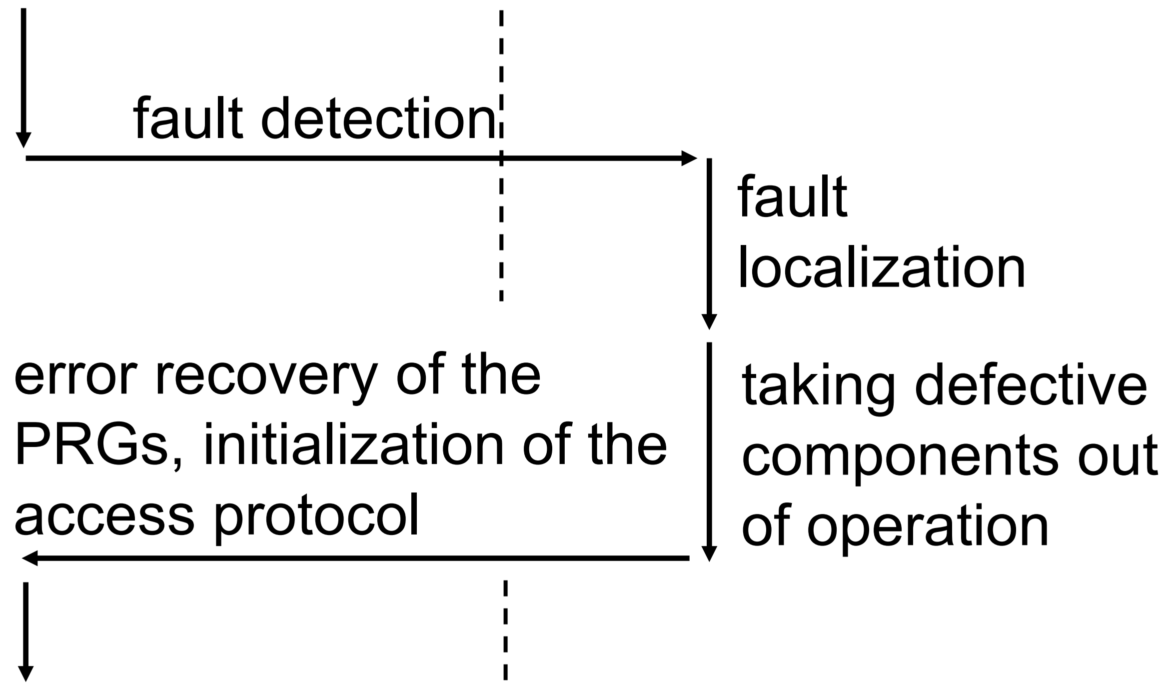
Fault tolerance: 2 modes of operation

A-mode

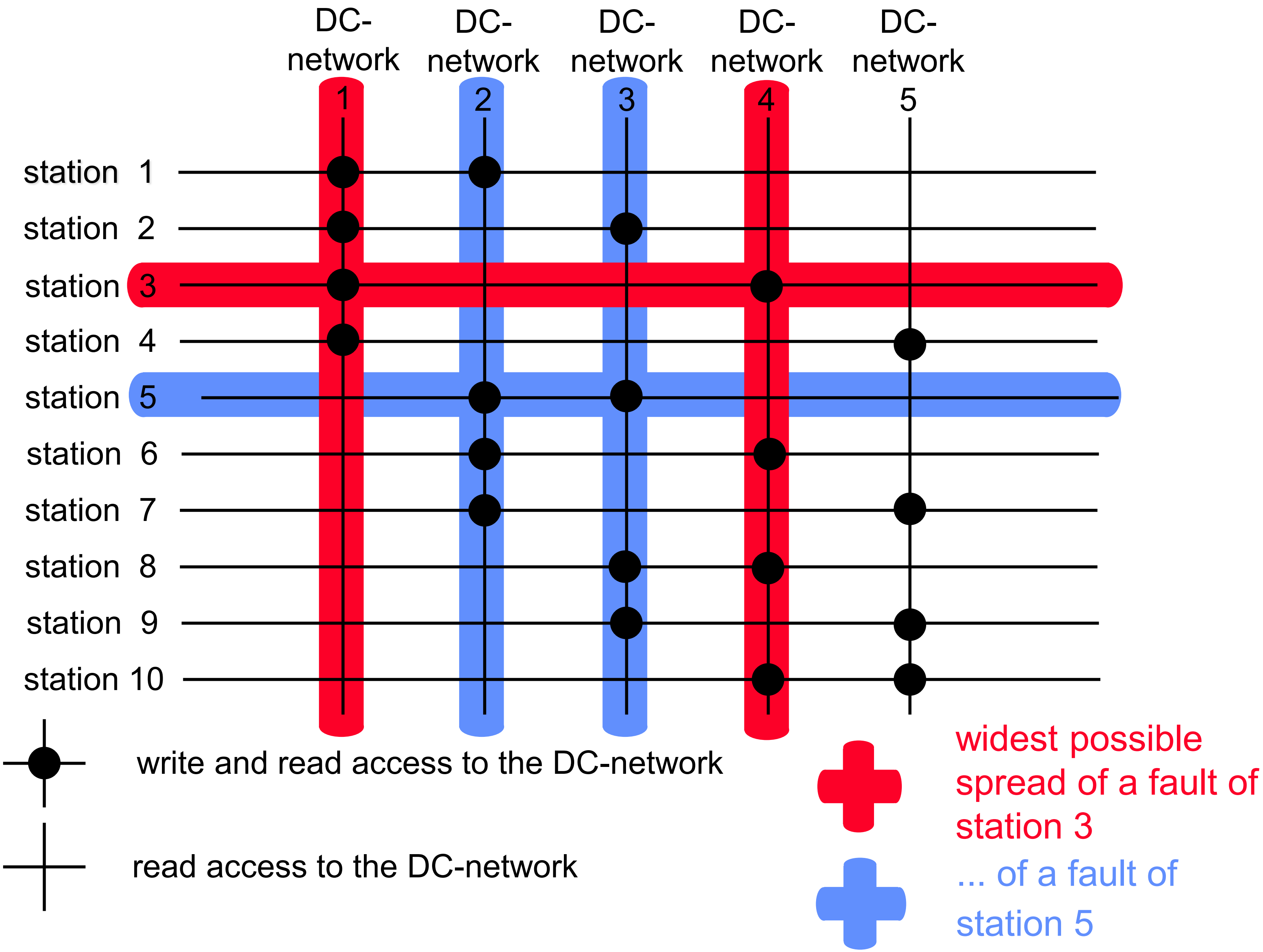
anonymous transmission of messages using superposed sending

F-mode

sender and recipient are not protected

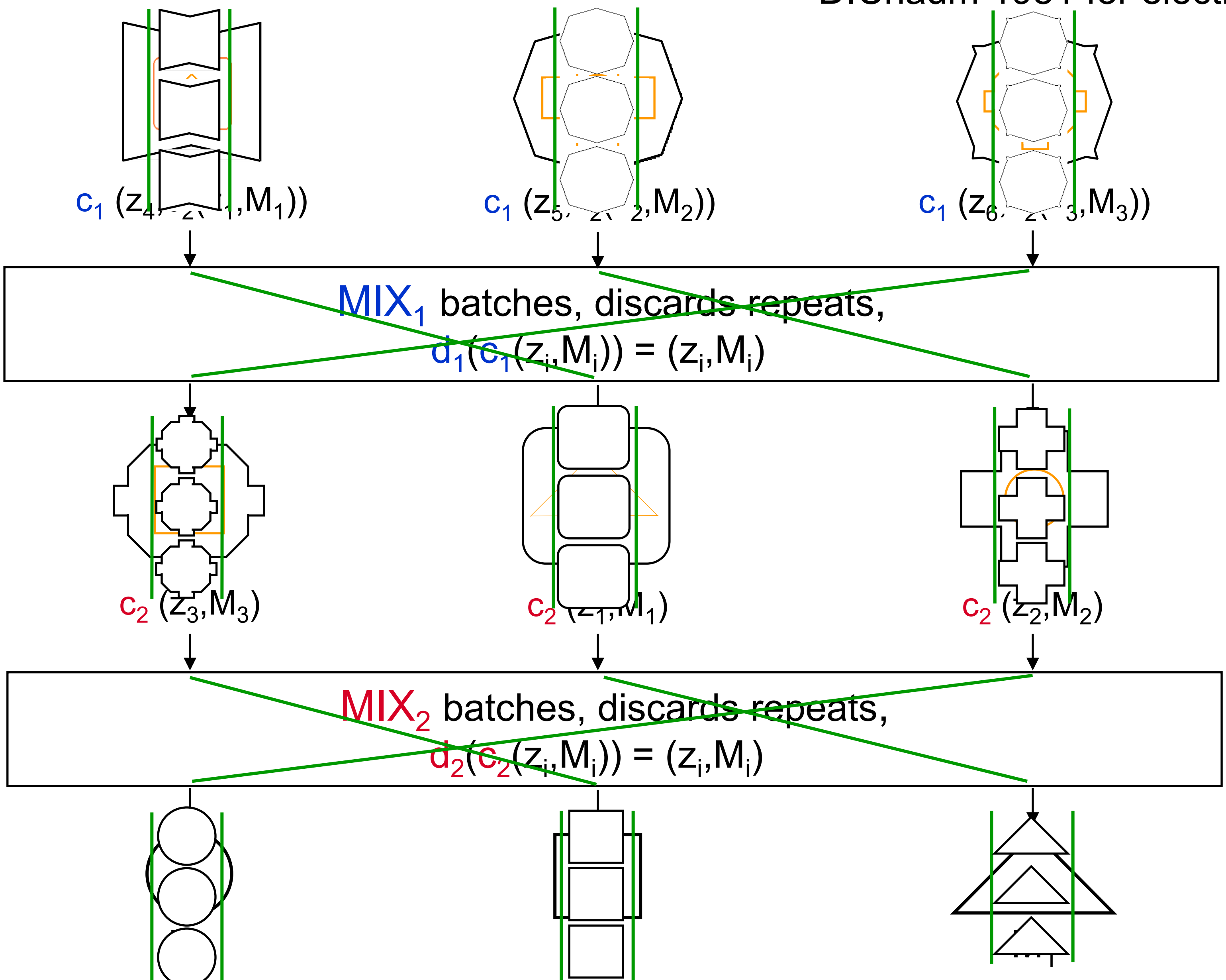


Fault tolerance: sender-partitioned DC-network



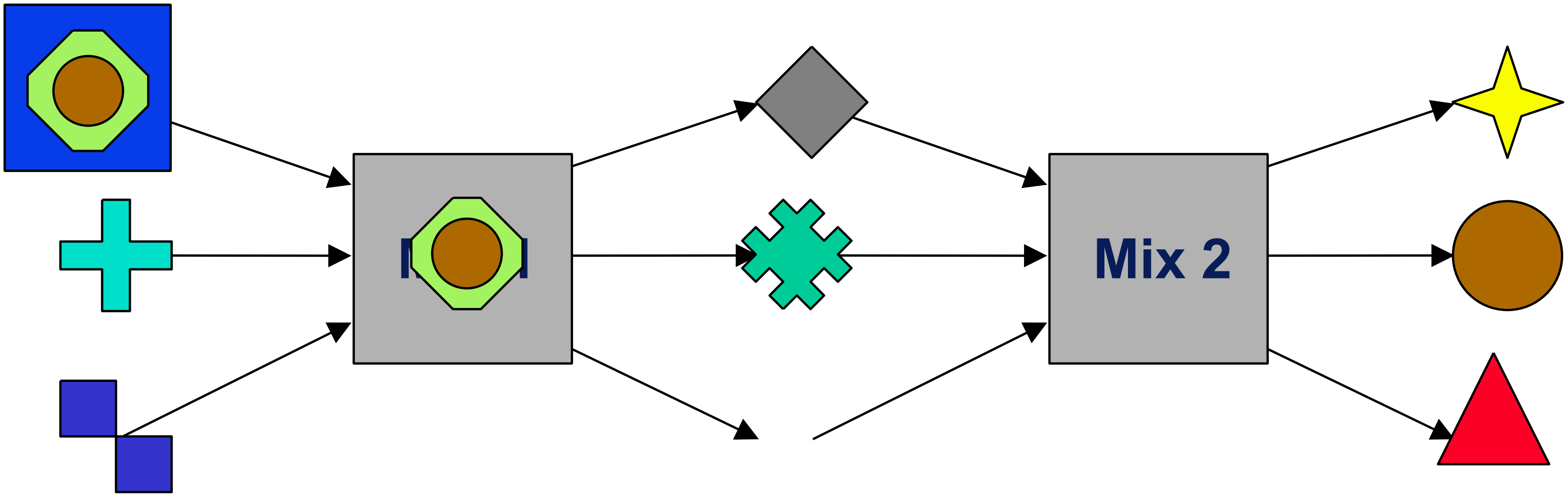
Protection of the communication relation: MIX-network

D.Chaum 1981 for electronic mail



The Mix protocol

Idea: Provide unlinkability between incoming and outgoing messages

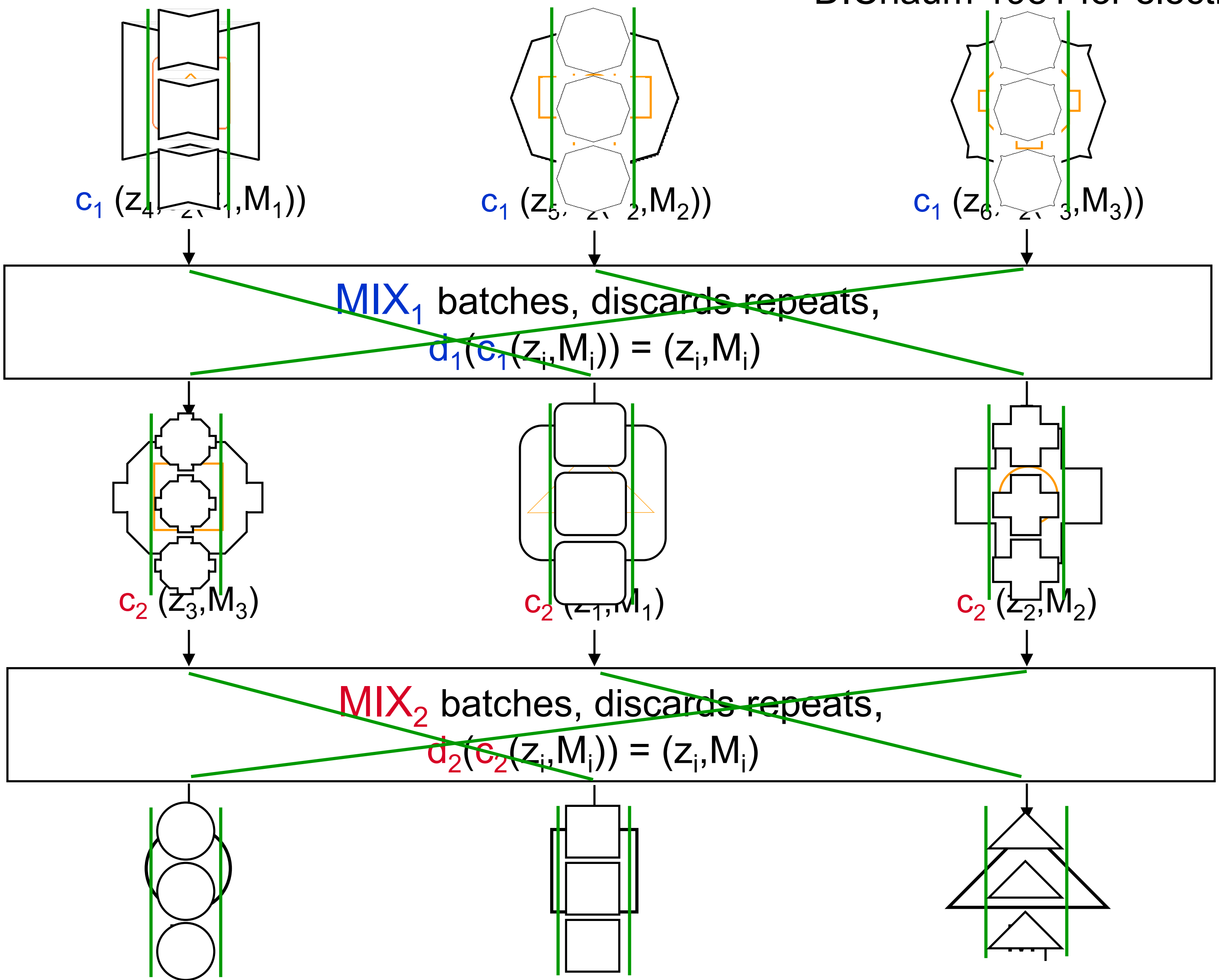


A Mix collects messages, changes their coding and forwards them in a different order.

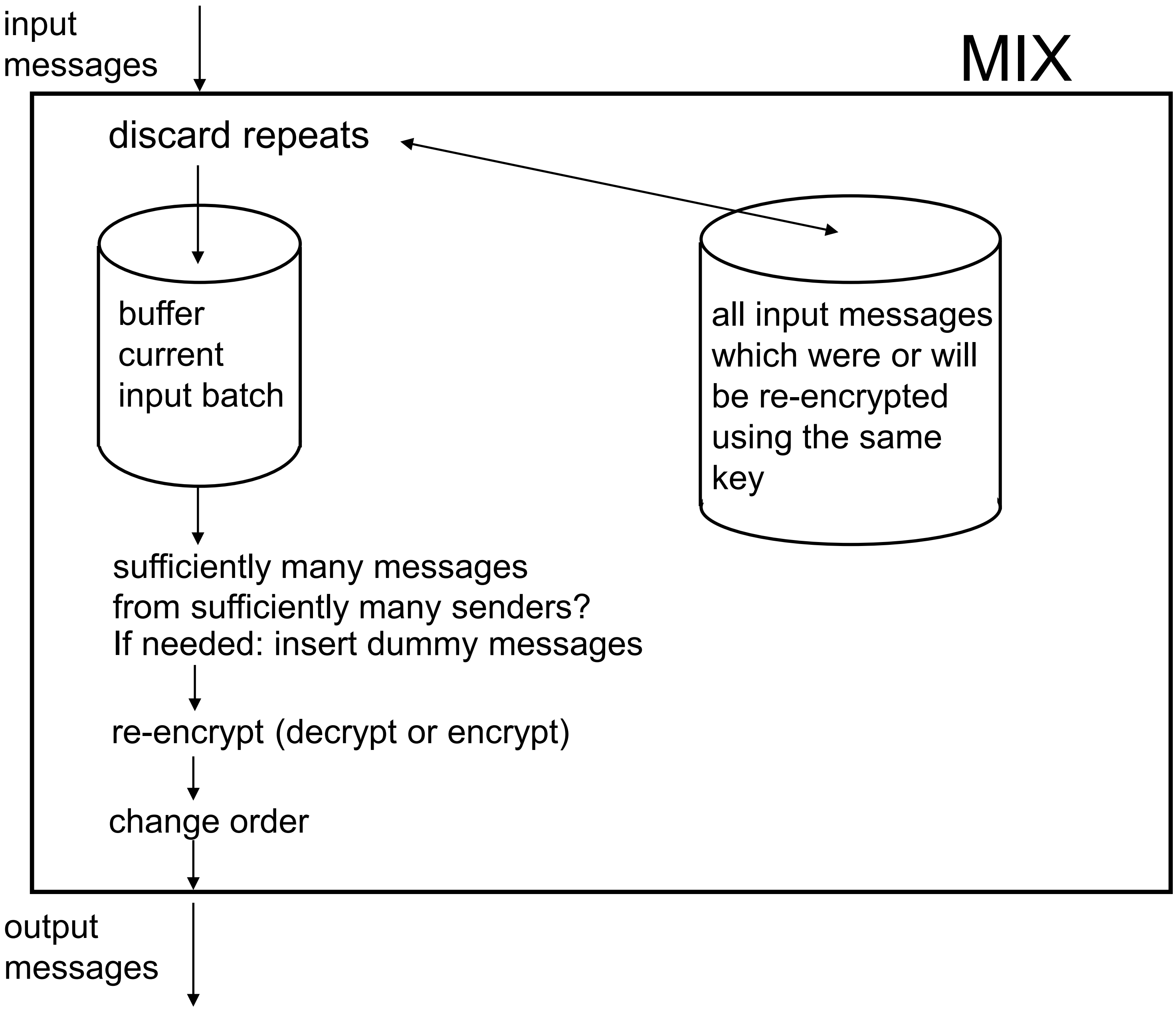
If ***all*** Mixes work together,
they can reveal the way of a given messages.

Protection of the communication relation: MIX-network

D.Chaum 1981 for electronic mail



Basic functions of a MIX



Properties of MIXes

MIXes should be designed independently
produced
operated
maintained ...

Messages of the same length
- buffer
- re-encrypt
- change order

} batch-wise

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

sym. encryption system only for
- first
- last

} MIX

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;

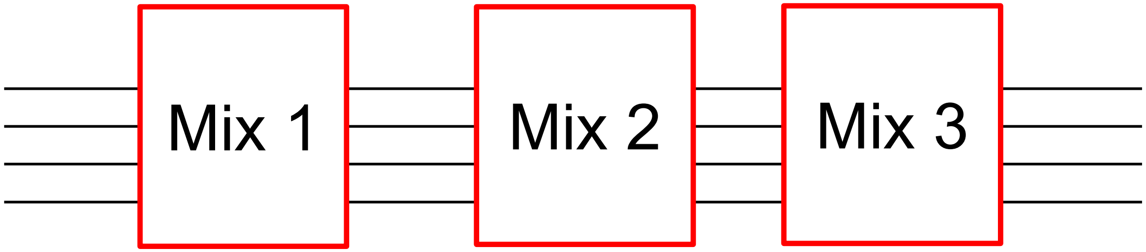
   ➔ before and after the encoding of the message it is the same
   ➔ re-encryption is irrelevant

2. Maximal protection:

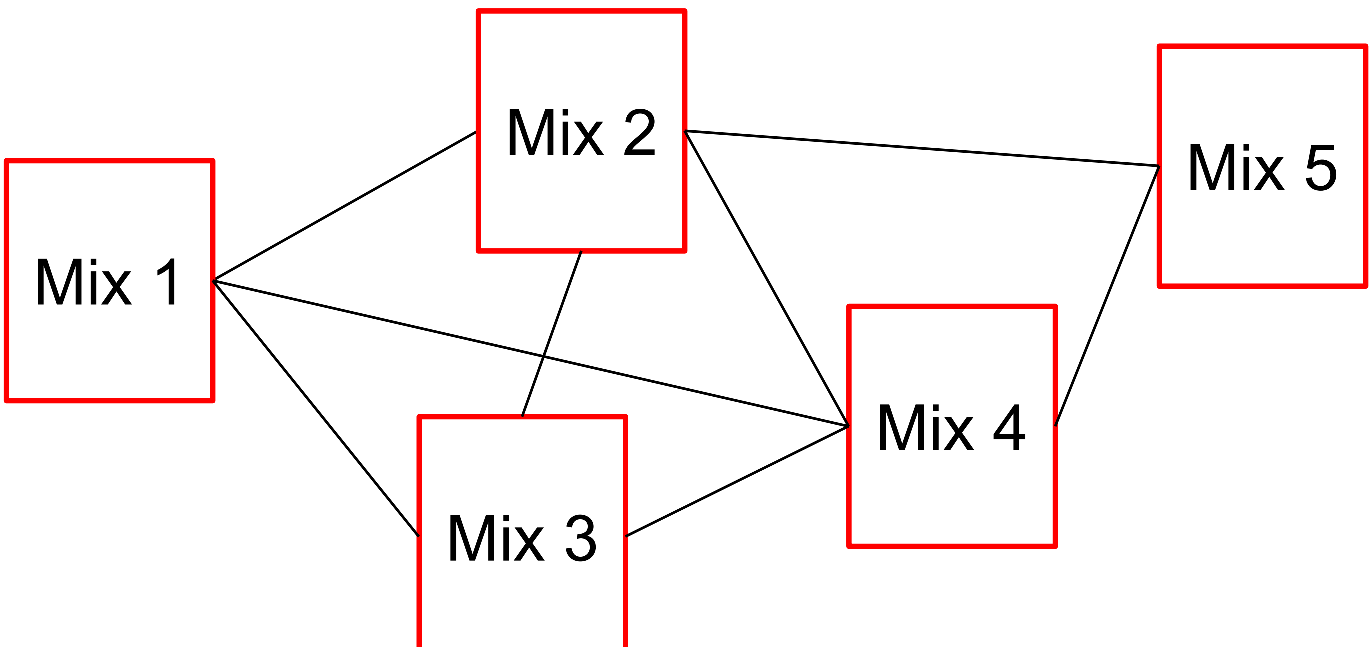
   MIXes are passed through simultaneously and therefore in the same order

Mix-network topologies

- cascades: fixed chain of Mixes

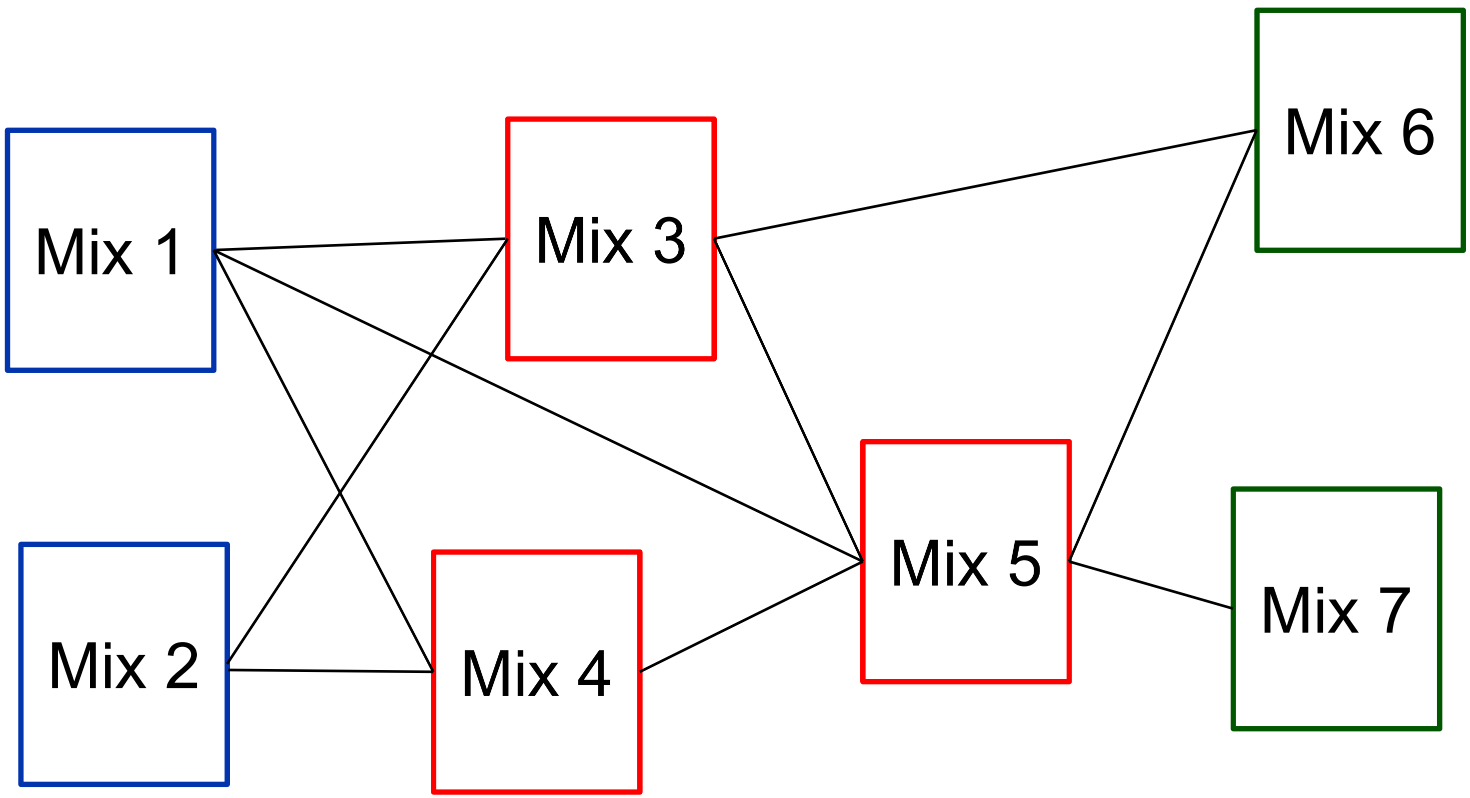


- free routes of Mixes: random selection by sender



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;

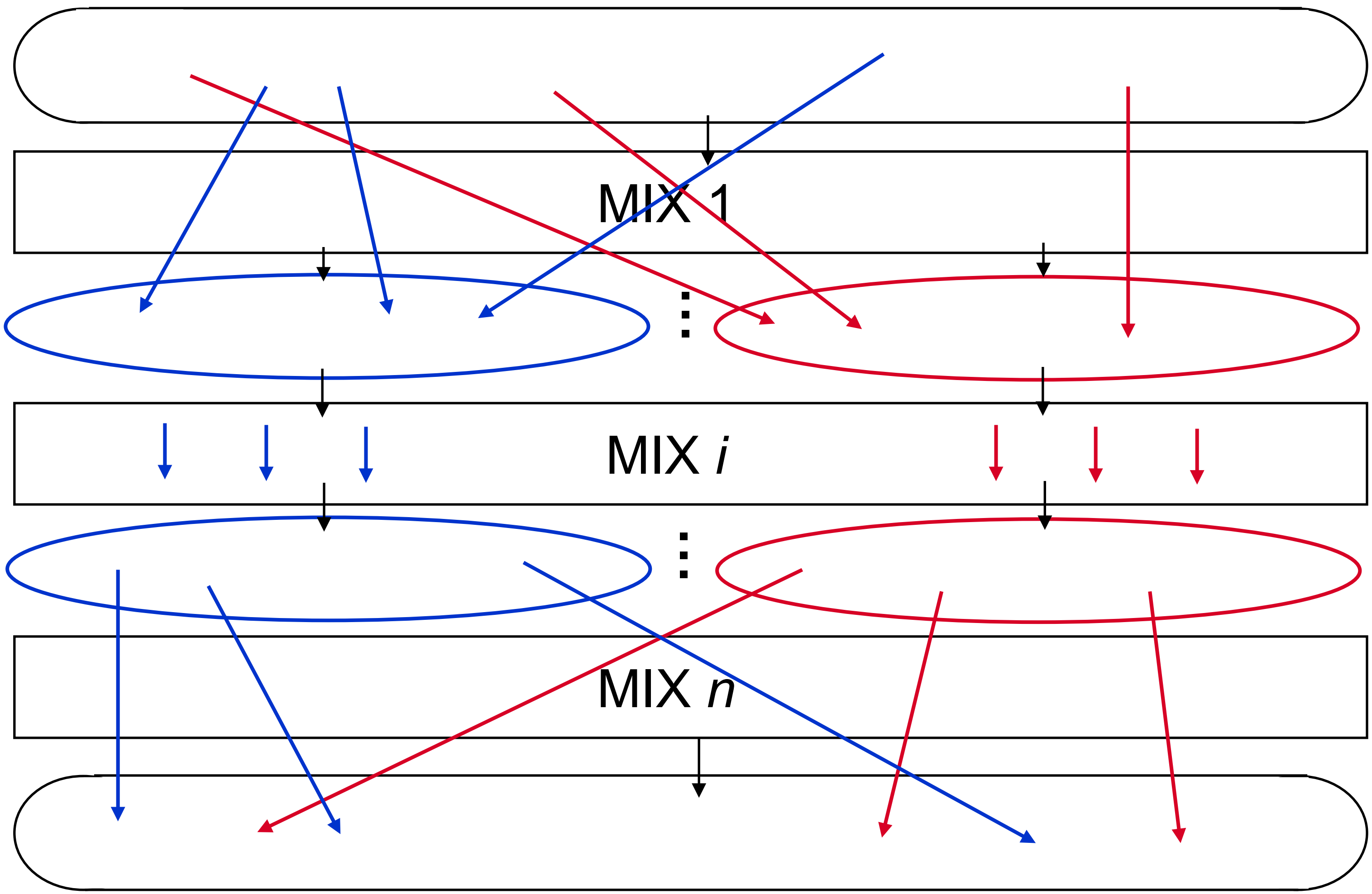
   ➔ before and after the encoding of the message it is the same
   ➔ re-encryption is irrelevant

2. Maximal protection:

   MIXes are passed through simultaneously and therefore in the same order

Maximal protection

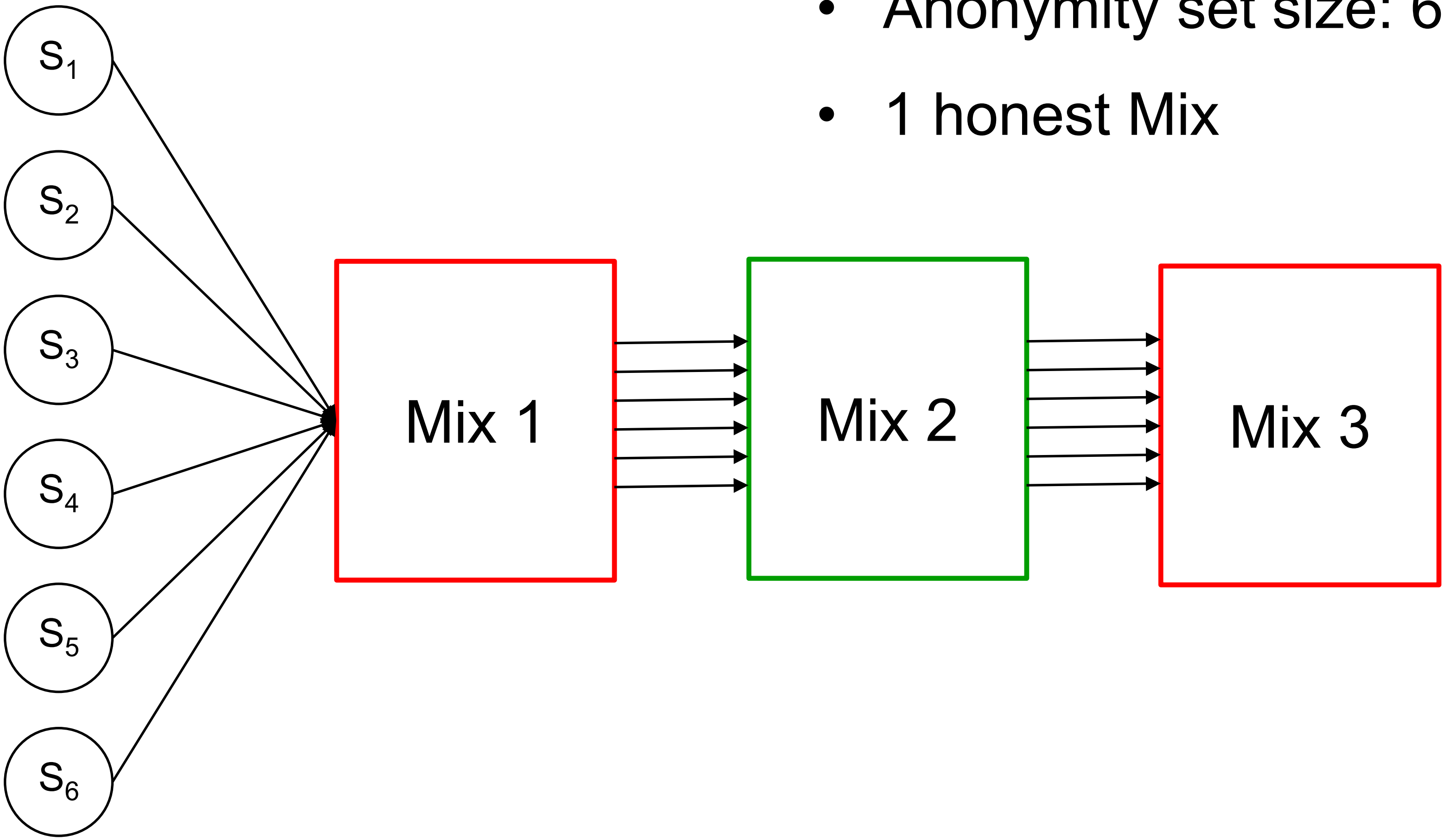
Pass through MIXes in the same order



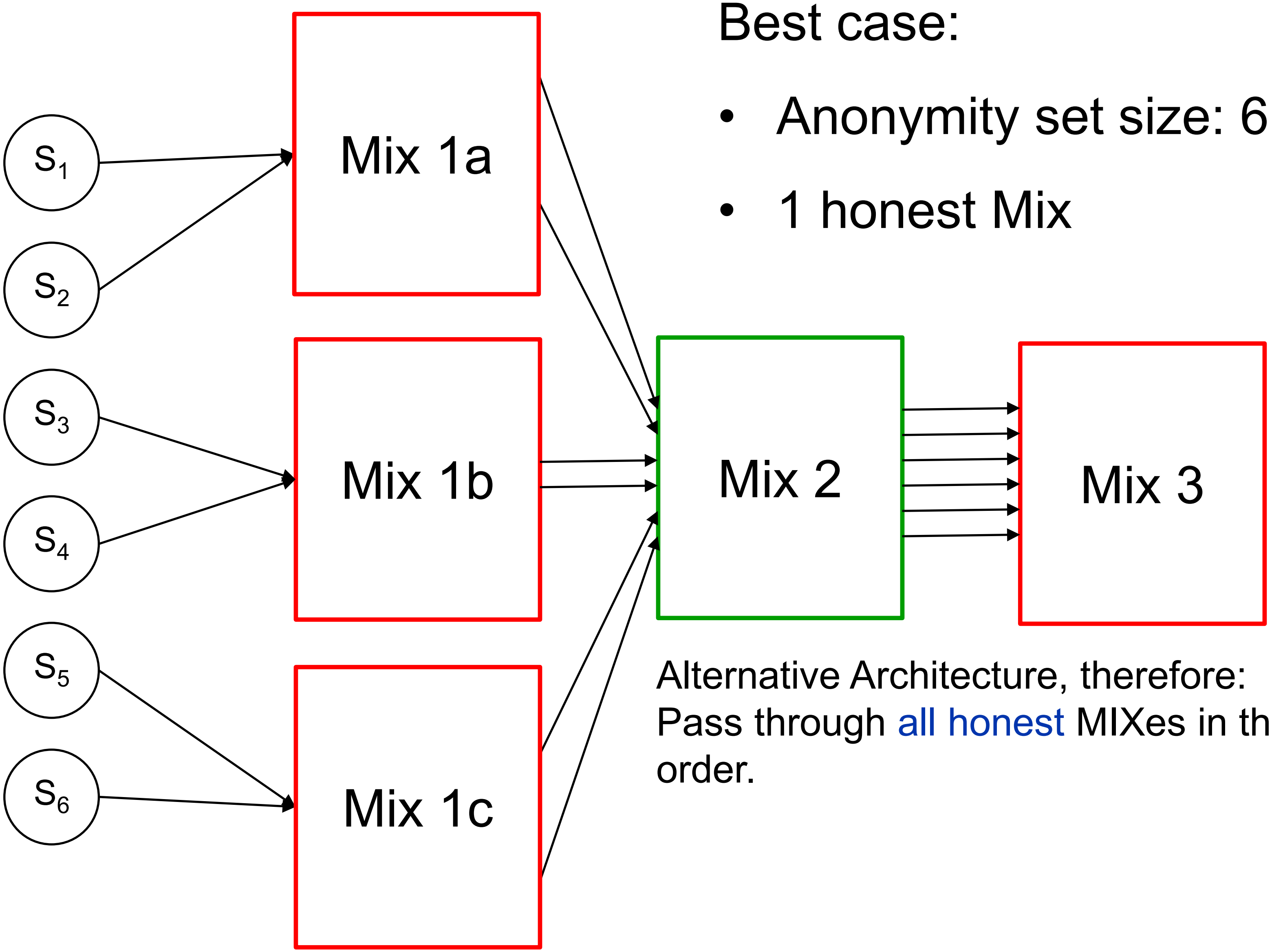
Maximal protection

Best case:

- Anonymity set size: 6
- 1 honest Mix



Maximal protection

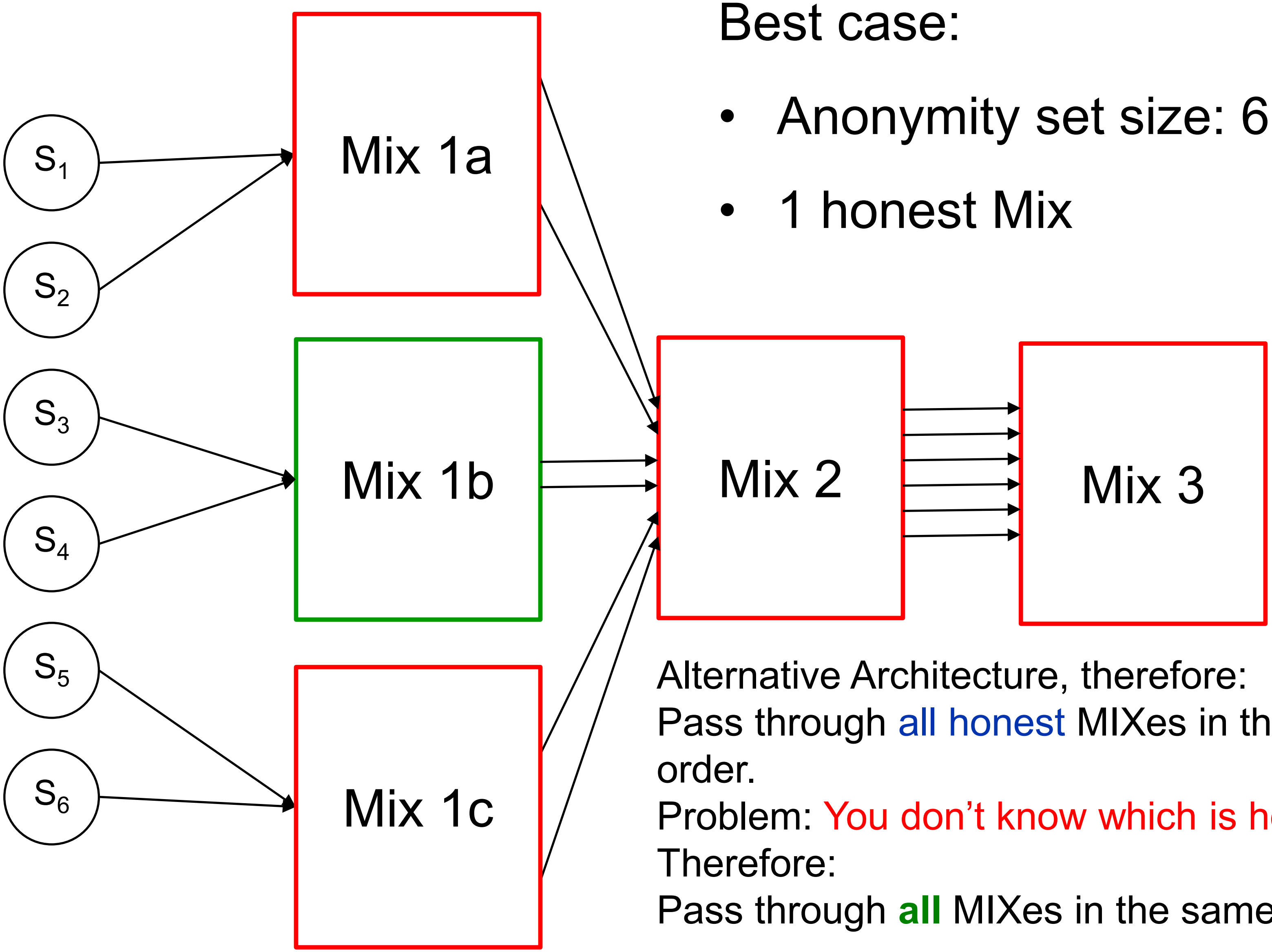


Best case:

- Anonymity set size: 6
- 1 honest Mix

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

Maximal protection

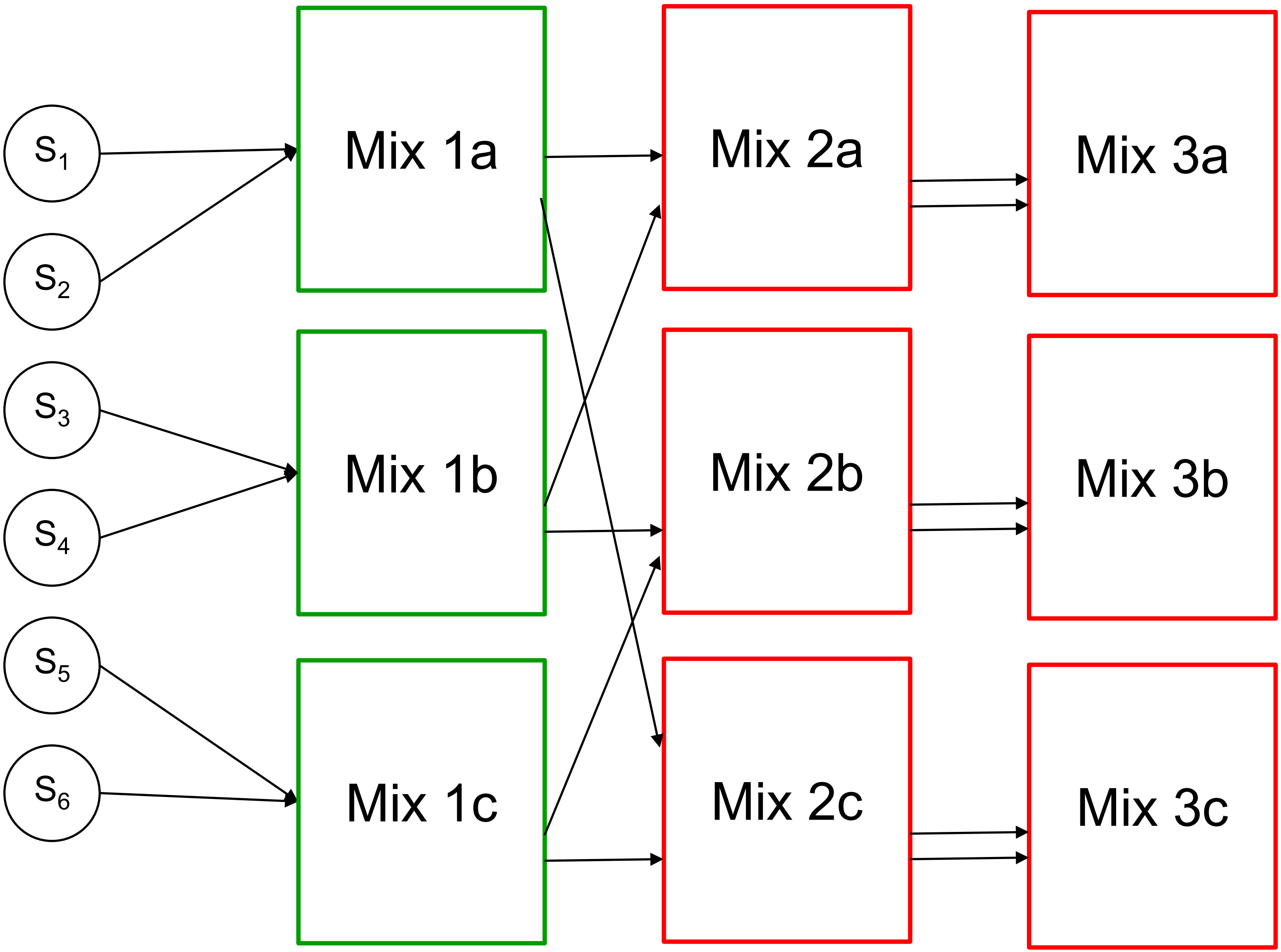


Best case:

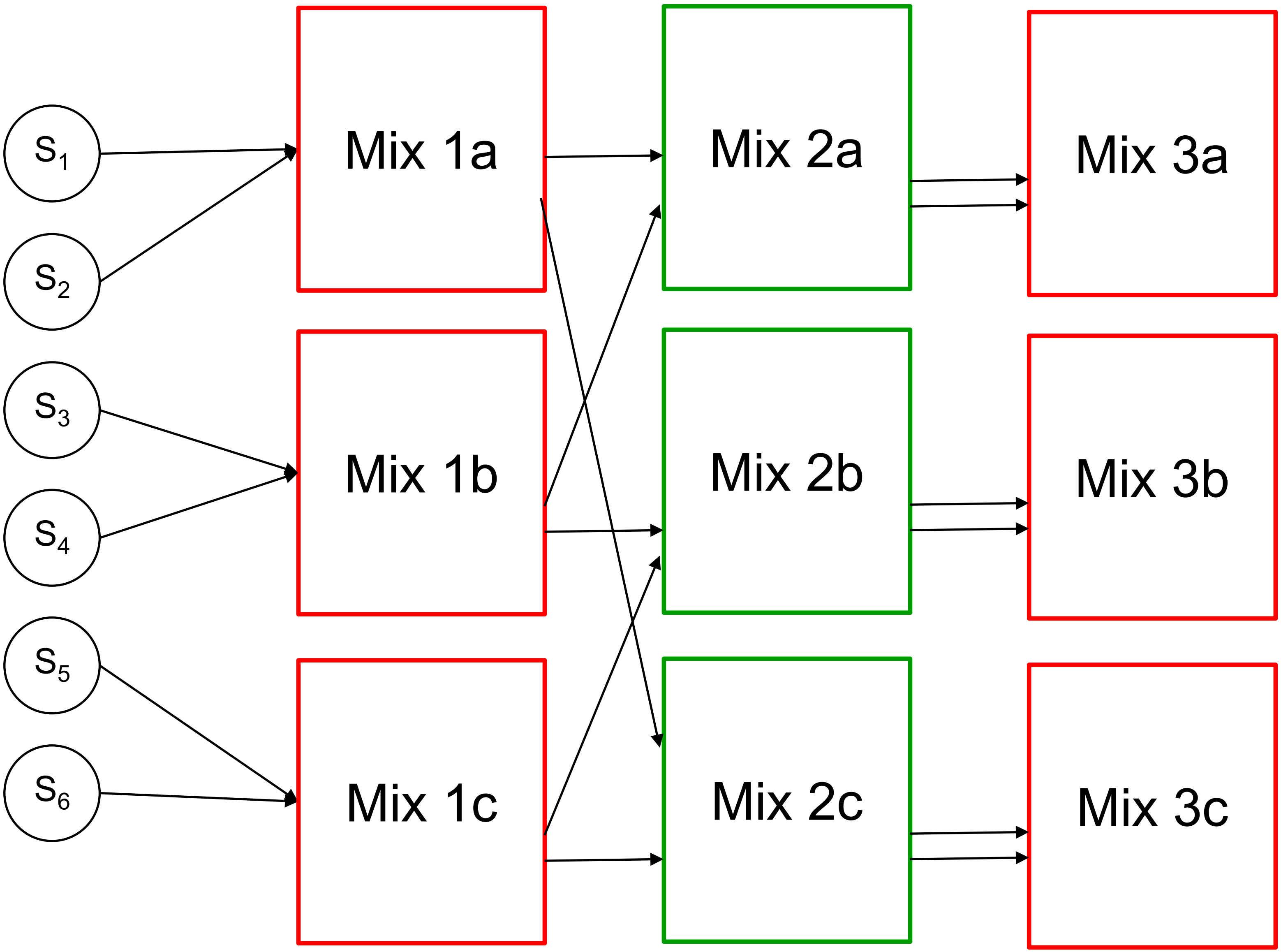
- Anonymity set size: 6
- 1 honest Mix

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.

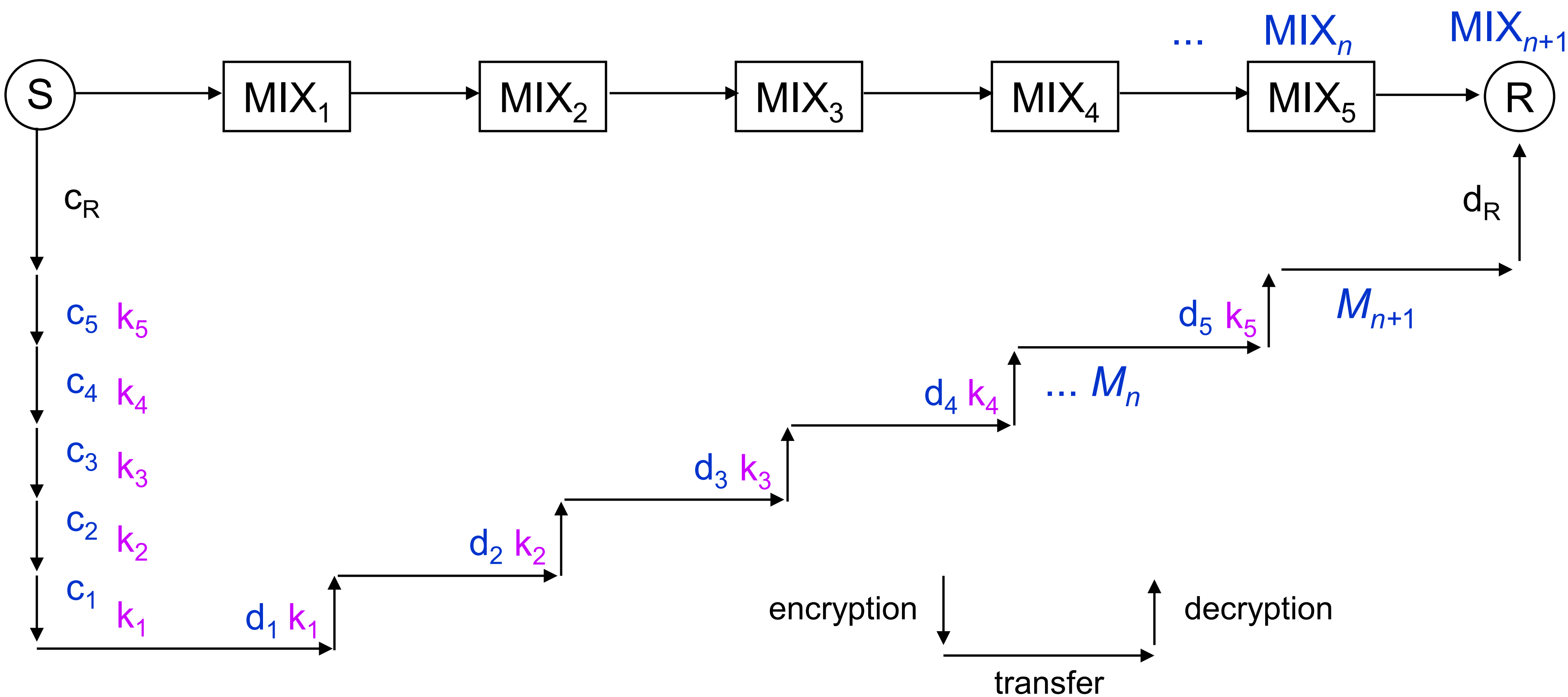
3 honest Mixes / Anonymity Set Size: 4



3 honest Mixes / Anonymity Set Size: 2



Re-encryption scheme for sender anonymity



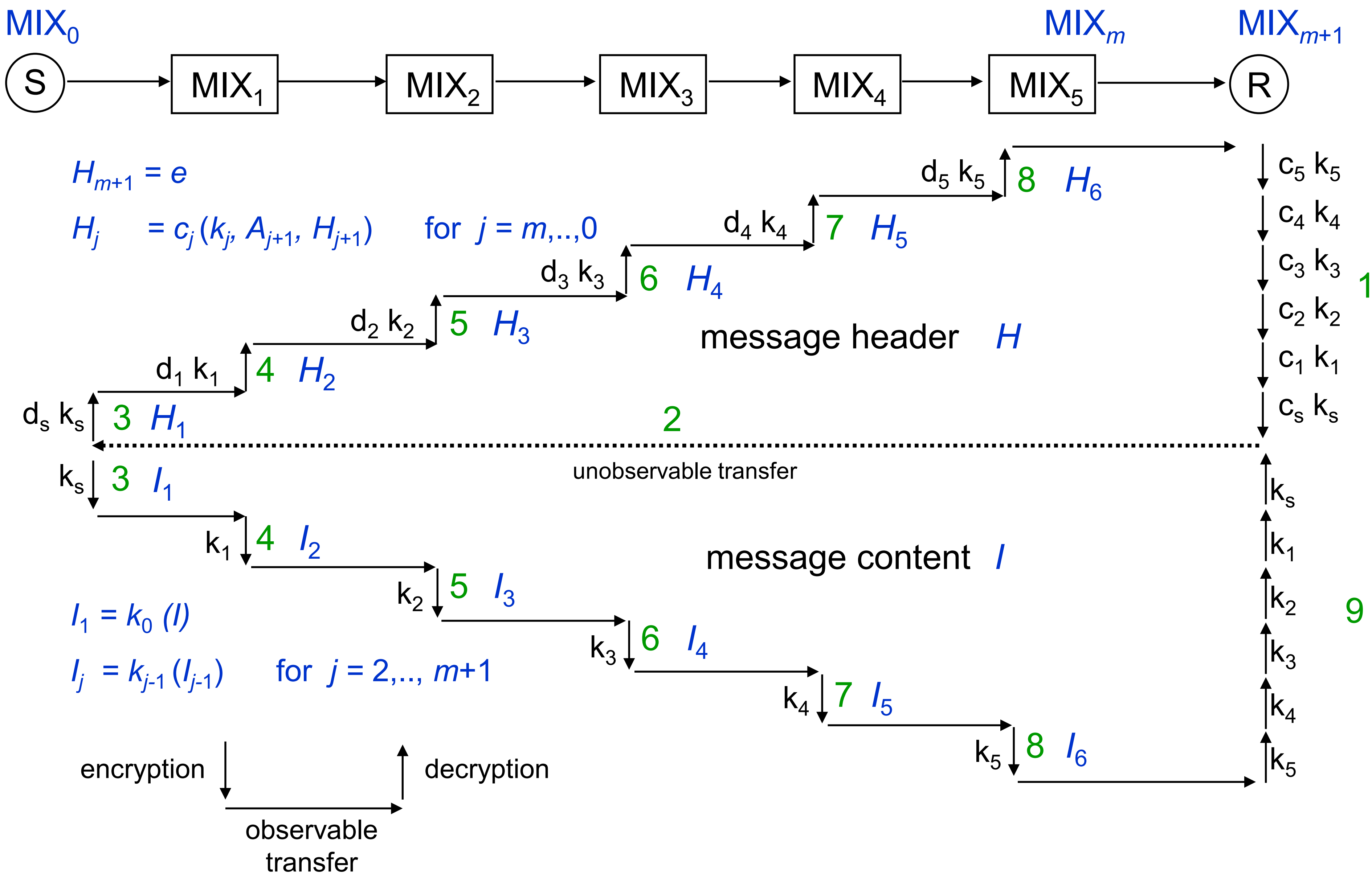
indirect re-encryption scheme for sender anonymity

$M_{n+1} = c_{n+1}(M)$

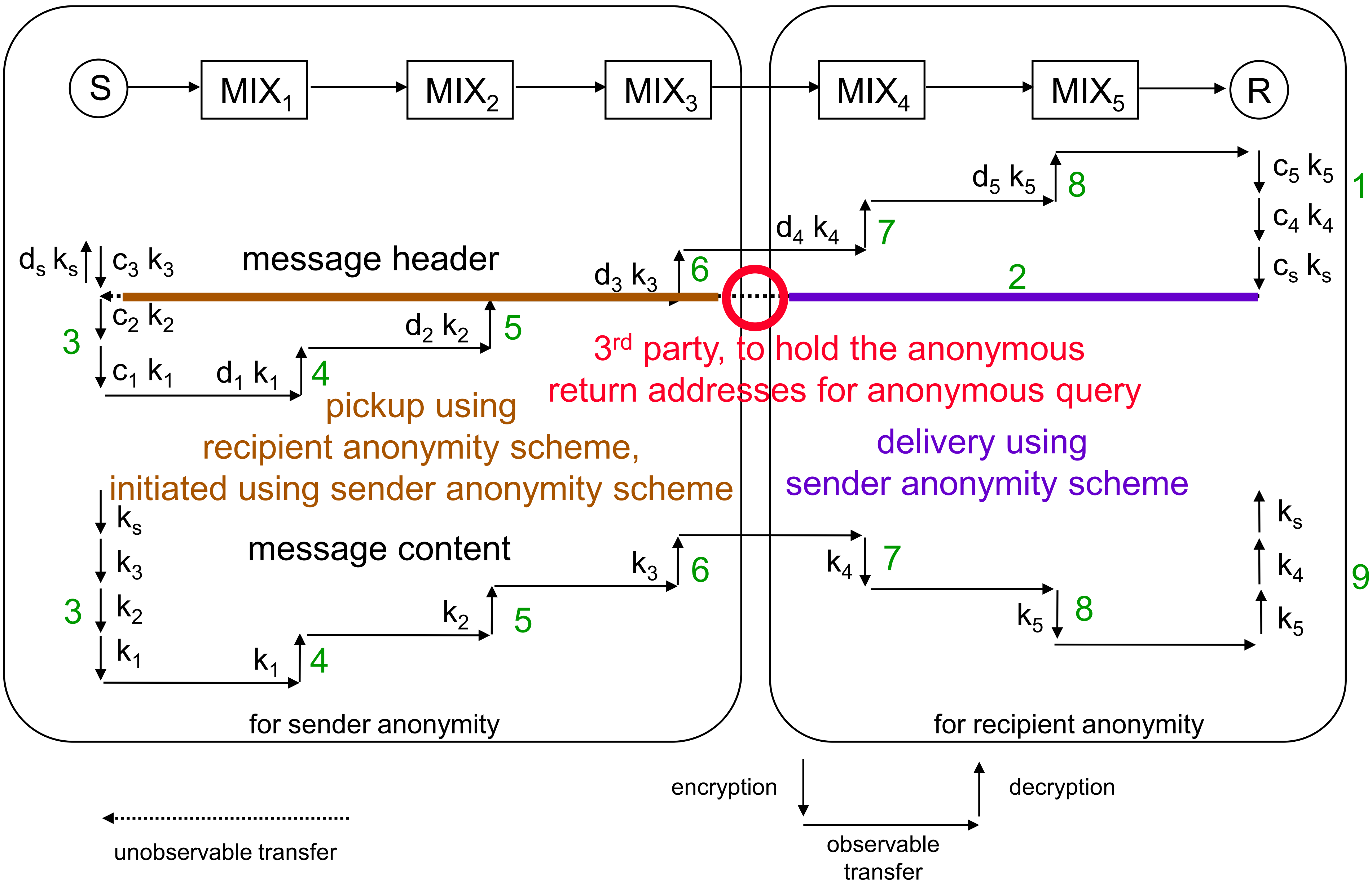
$M_i = c_i(z_i, A_{i+1}, M_{i+1})$ for $i = n,..,1$

$M_i = c_i(k_i, A_{i+1}); k_i(M_{i+1})$

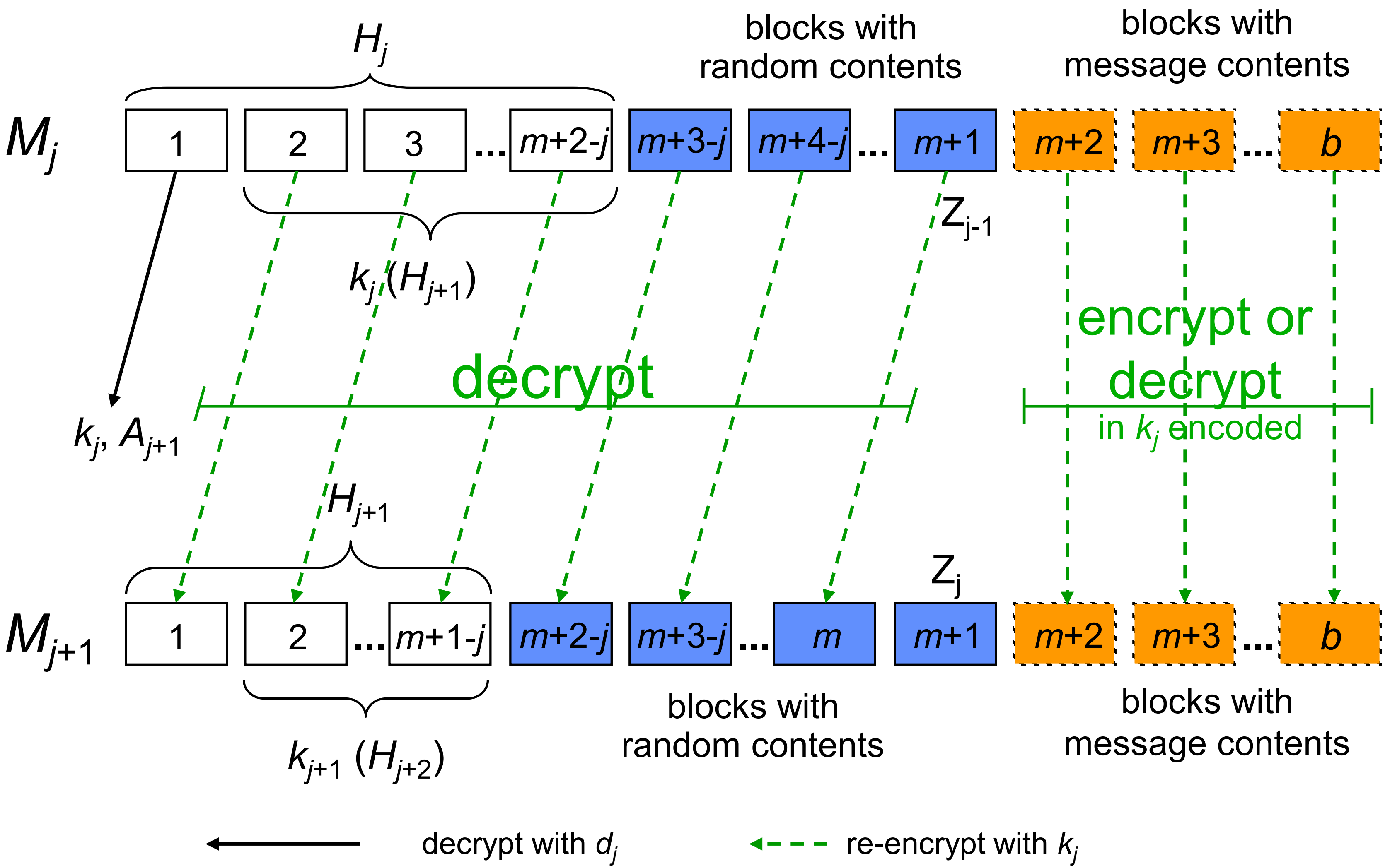
Indirect re-encryption scheme for recipient anonymity



Indirect re-encryption scheme for sender and recipient anonymity



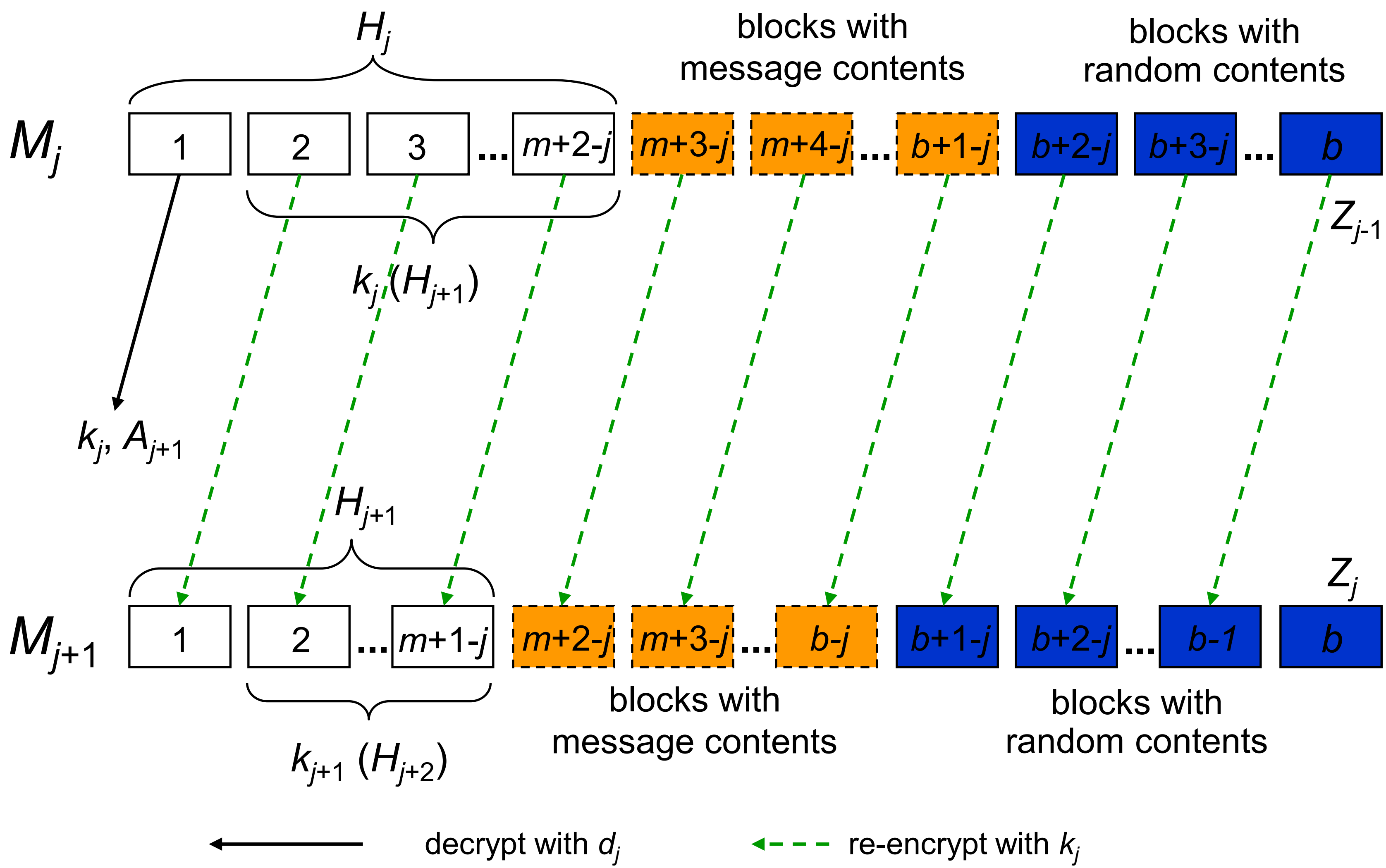
Indirect re-encryption scheme maintaining message length



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

$H_j = [c_j(k_j, A_{j+1})], k_j(H_{j+1})$ for $j = m,..,1$

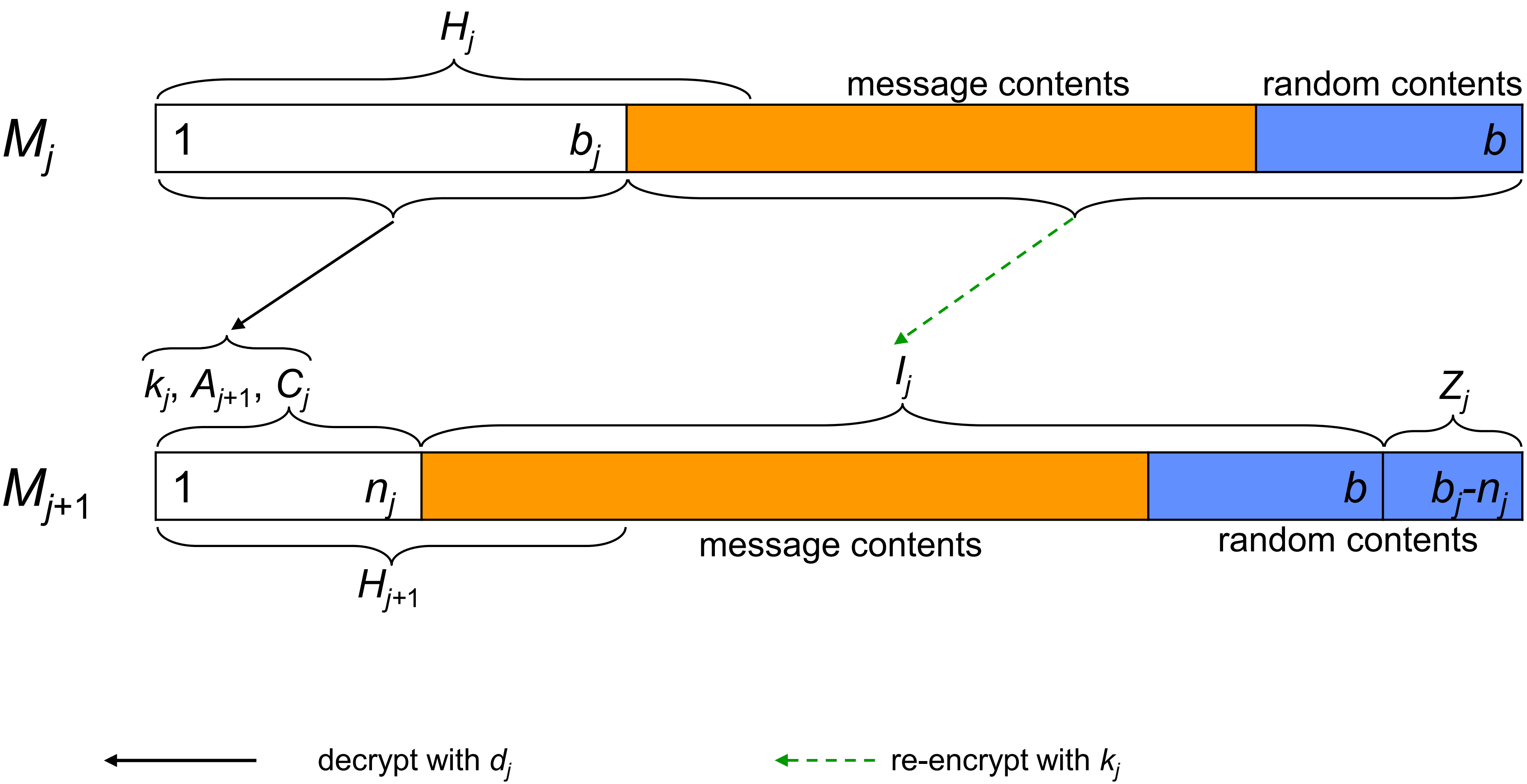
Indirect re-encryption scheme maintaining message length for special symmetric encryption systems



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

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

Minimally message expanding re-encryption scheme maintaining message length



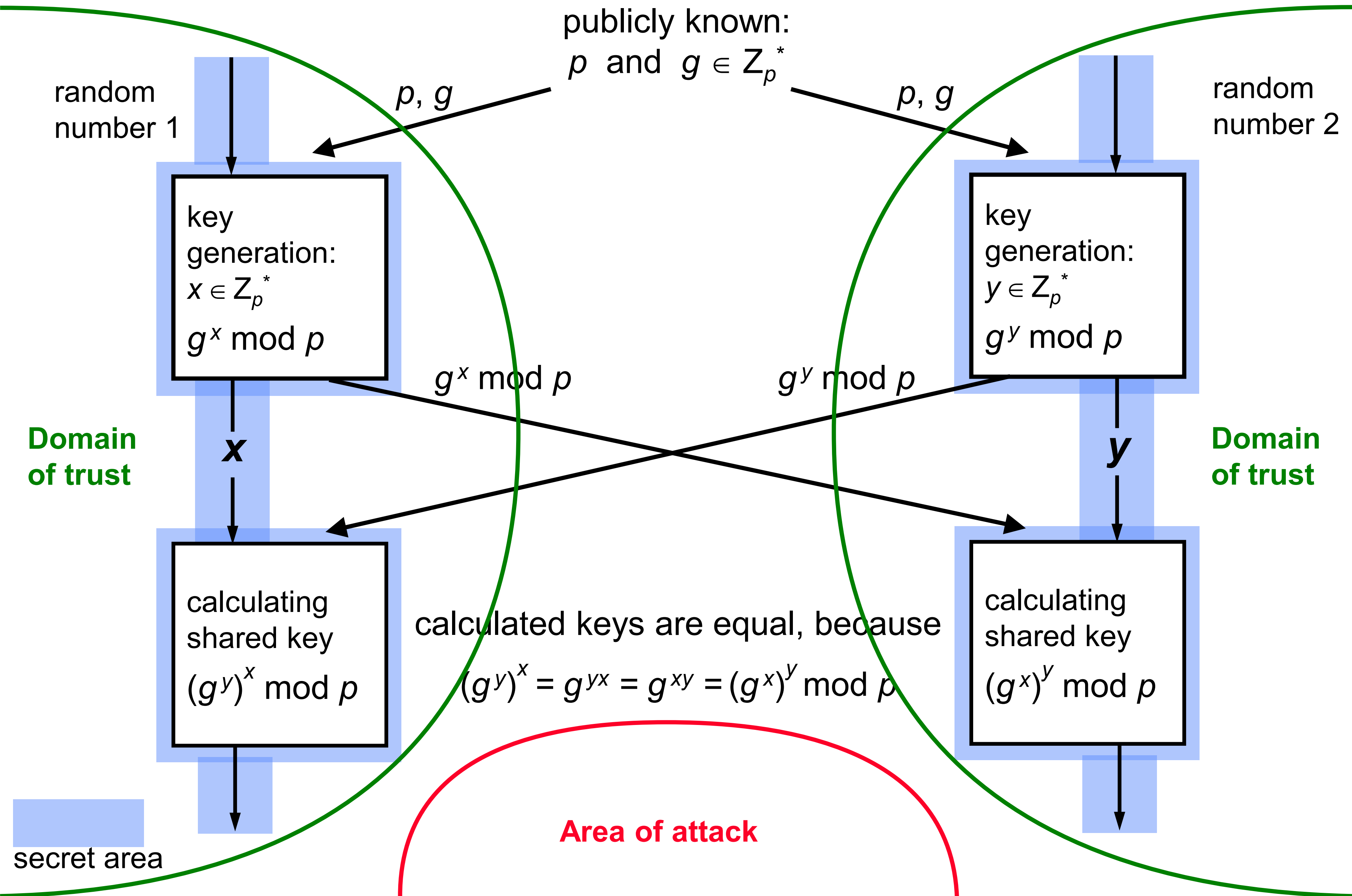
if $k^{-1}(k(M)) = M$

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

Mix Packets based on Diffie-Hellman Key Agreement

Danezis, Goldberg: “Sphinx: A Compact and Provably Secure Mix Format”, 2009

Recall: Diffie-Hellman key agreement



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, $y_i = g^{x_i} \bmod p$
    - secret k_i shared with M_i: $k_i = y_{M_i}^{x_i} \bmod p$
  - 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, $y=g^x \bmod p$
    - secret k_i shared with M_i: $k_i=y_{M_i}{}^x \bmod p$
  - layered encryption:
    - y, $k_i(k_{i+1}(\ldots))$

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

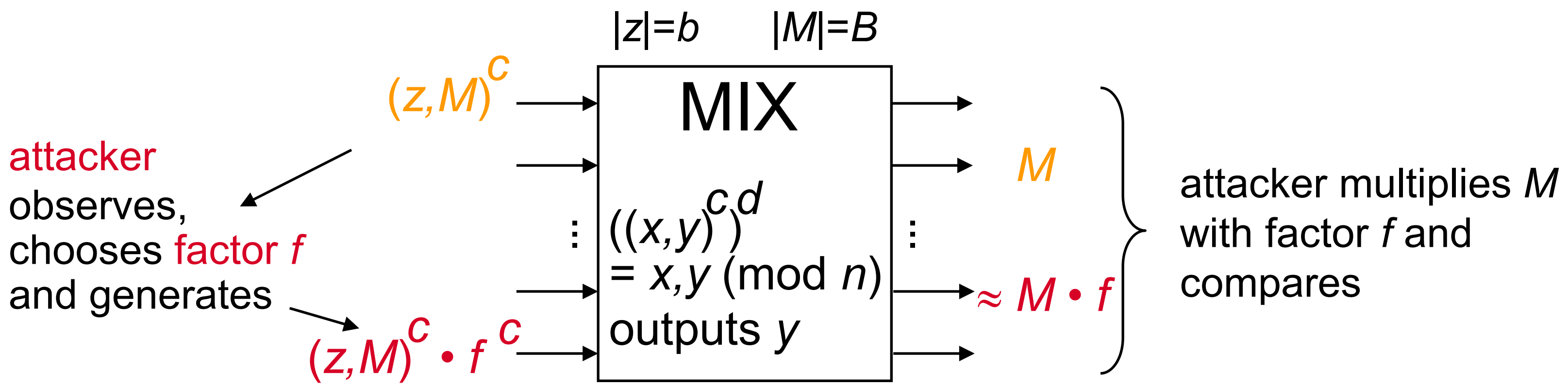
- Solution:
  - calculate y_{i+1} from y_i
  - $x_{i+1} = x_i^{b_i} \bmod p$
  - $b_{i+1} = \mathrm{Hash}(y_i, k_i)$
  - $y_{i+1} = g^{x_{i+1}} \bmod p$
    $= g^{x_i b_i} \bmod p$
    $= y_i^{b_i} \bmod p$

➔ mix M_i can calculate y_{i+1} from y_i!

➔**only** M_i can calculate y_{i+1} from y_i!

Breaking the direct RSA-implementation of MIXes (1)

Implementation of MIXes using RSA without redundancy predicate and with contiguous bit strings (David Chaum, 1981) is insecure:



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

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

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

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

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

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

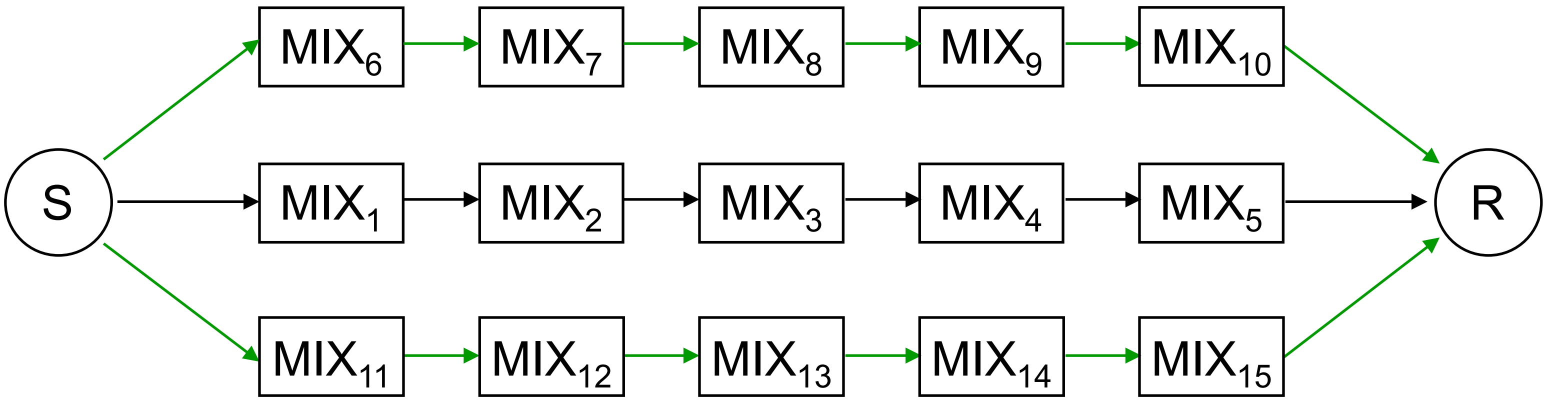
$$-2^b < z\cdot f - z' < 2^{2b} \quad (2)$$

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

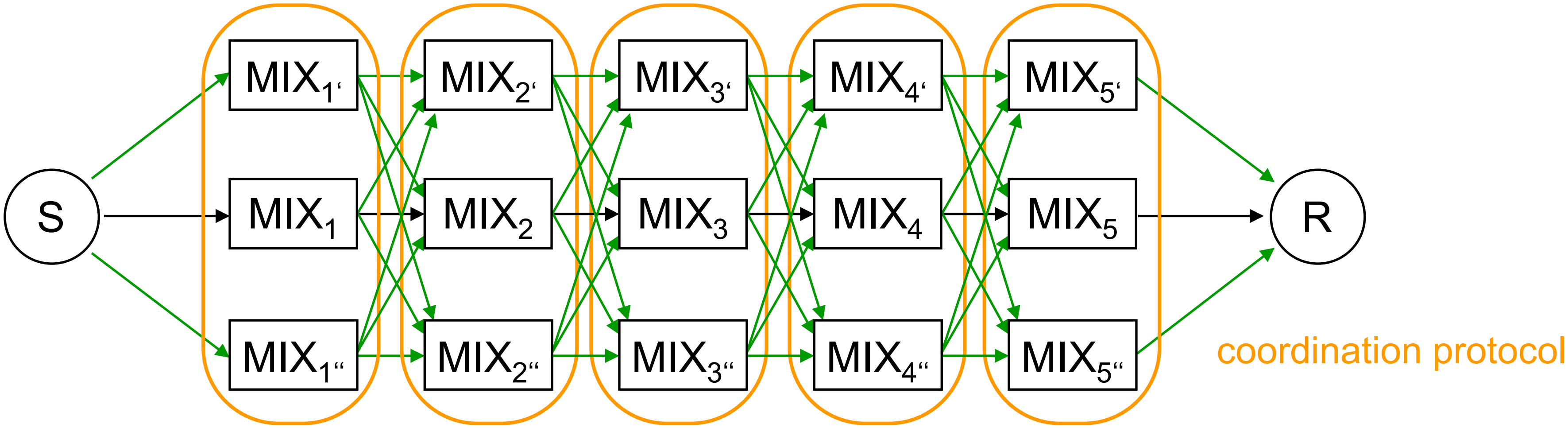
(2) holds, if $b \ll B$, very probably only for one pair (P1,P2). P1 is output message to $(z,M)^c$, P2 to $(z,M)^c\cdot f^c$.

If (2) holds for several pairs, the attack is repeated with another factor.

Fault tolerance in MIX-networks (1)

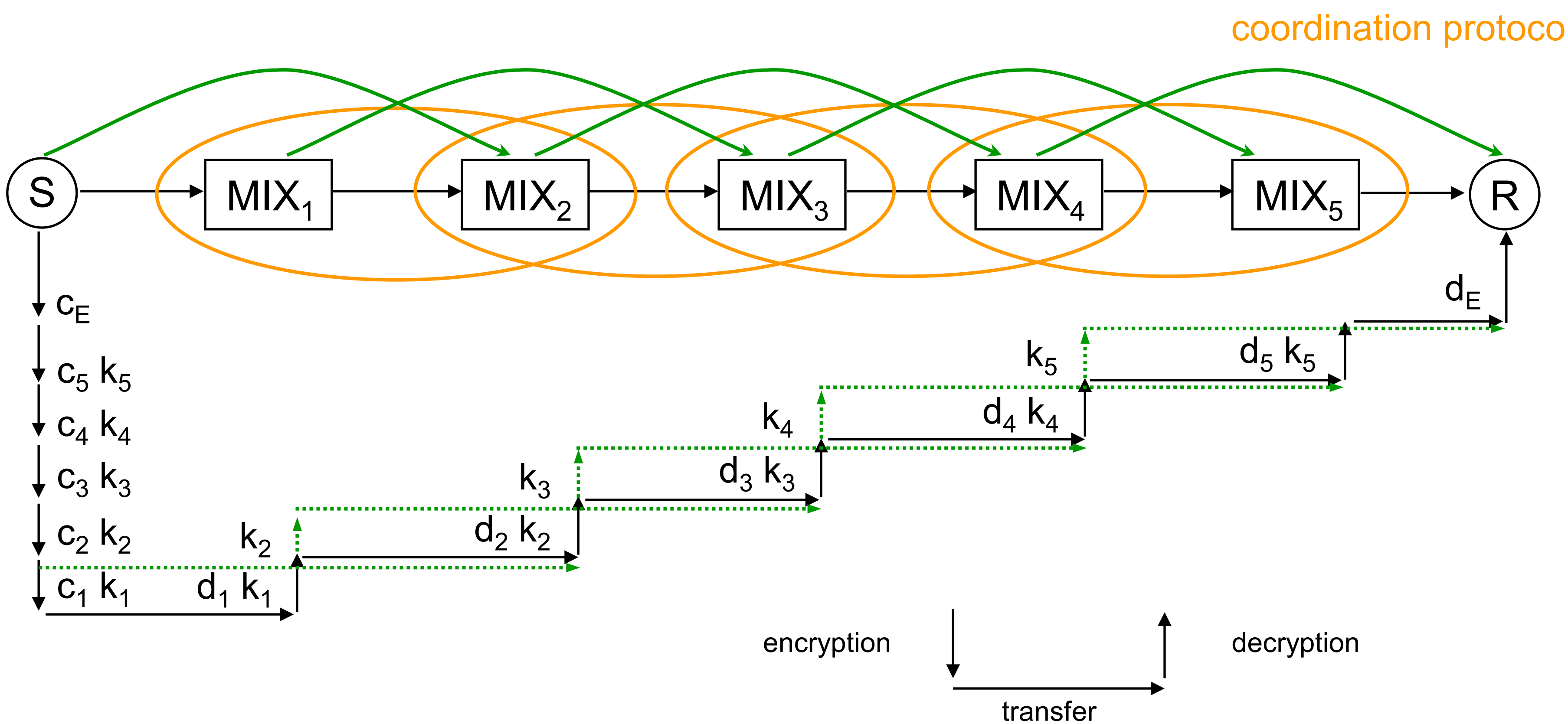


2 alternative routes via disjoint MIXes



$MIX_{i'}$ or $MIX_{i''}$ can substitute MIX_i

Fault tolerance in MIX-networks (2)



In each step, one MIX can be skipped

Complexity of the basic methods

	unobservability of neighboring lines and stations as well as digital signal regeneration RING-network	DC-network	MIX-network
attacker model	physically limited	computationally restricted w.r.t. service delivery – computationally restricted • cryptographically strong • well analyzed	computationally restricted not even well analyzed asymmetric encryption systems are known which are secure against adaptive active attacks
expense per user	$O(n)$ $(\geq \frac{n}{2})$ transmission	$O(n)$ $(\geq \frac{n}{2})$ transmission $O(k \cdot n)$ key	$O(k)$, practically: ≈ 1 transmission on the last mile ... in the core network $O(k^2)$, practically: $\approx k$

n = number of users

k = connectedness key graph of DC-networks respectively number of MIXes

Encryption in layer models

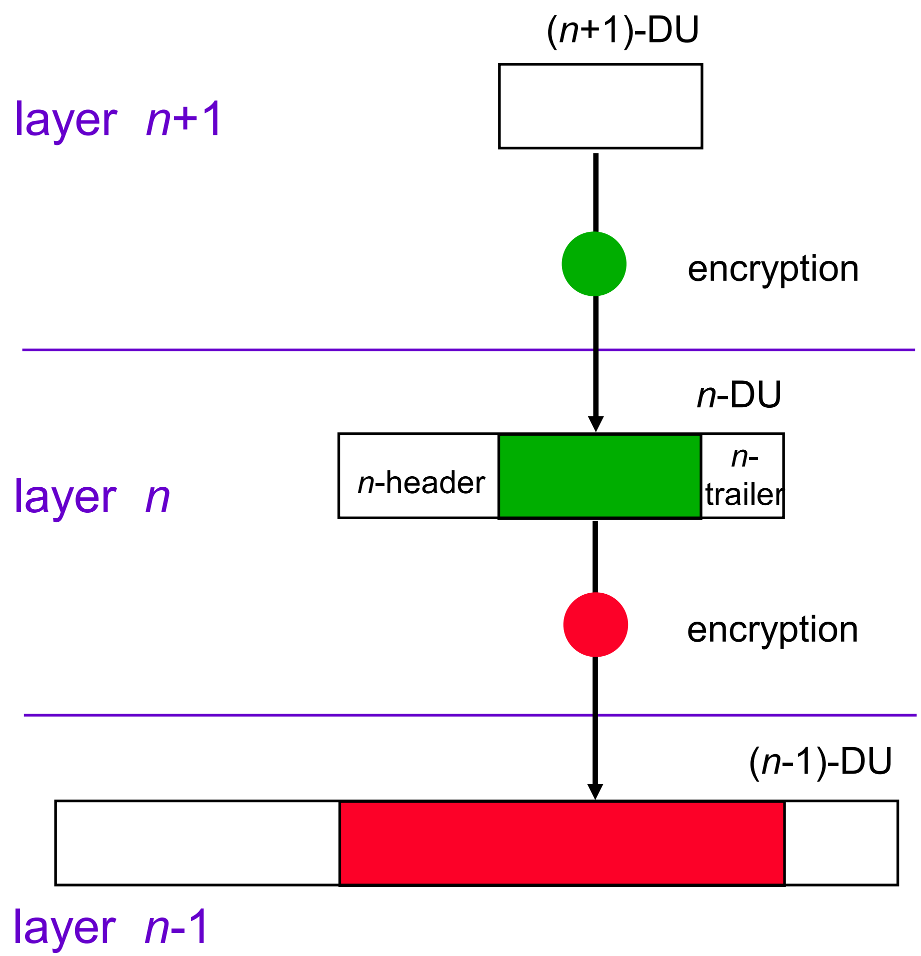
In the OSI model it holds:

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*.

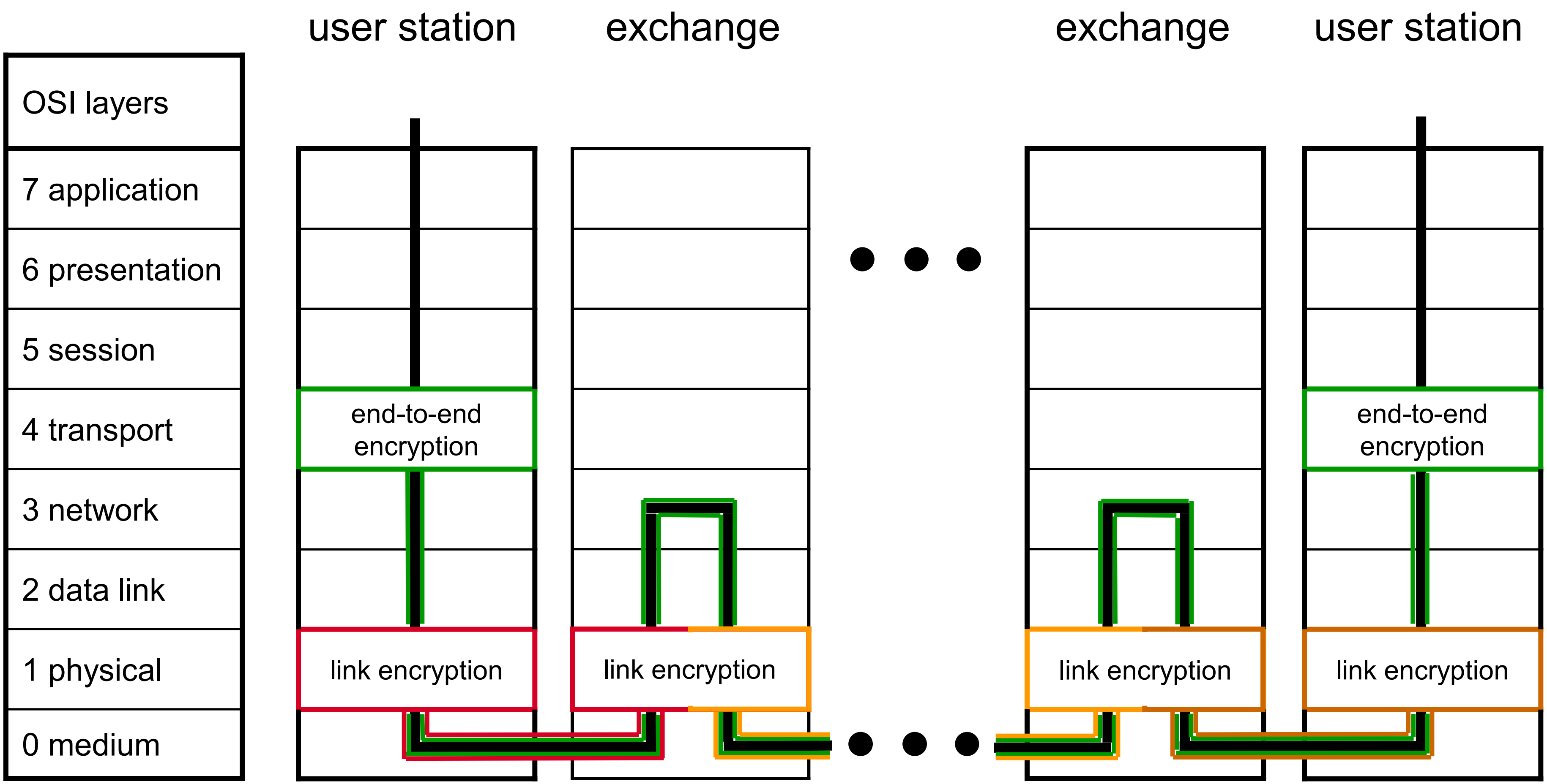
For packet-oriented services, the layer *n* typically furnishes the (*n*+1)-DUs with a *n*-header and possibly with an *n*-trailer, too, and delivers this as *n*-DU to layer *n*-1. This can also be done encrypted again.

and so on.

All encryptions are independent with respect to both the encryption systems and the keys.



Arranging it into the OSI layers (1)



Arranging it into the OSI layers (2)

OSI layers	broadcast	query	MIX-network	DC-network	RING-network
7 application					
6 presentation					
5 session					
4 transport	implicit addressing	implicit addressing			
3 network	broad-cast	query and superpose	buffer and re-encrypt		
2 data link				anonymous access	anonymous access
1 physical	channel selection			superpose keys and messages	digital signal regeneration
0 medium					ring

has to preserve anonymity against the communication partner

end-to-end encryption

has to preserve anonymity

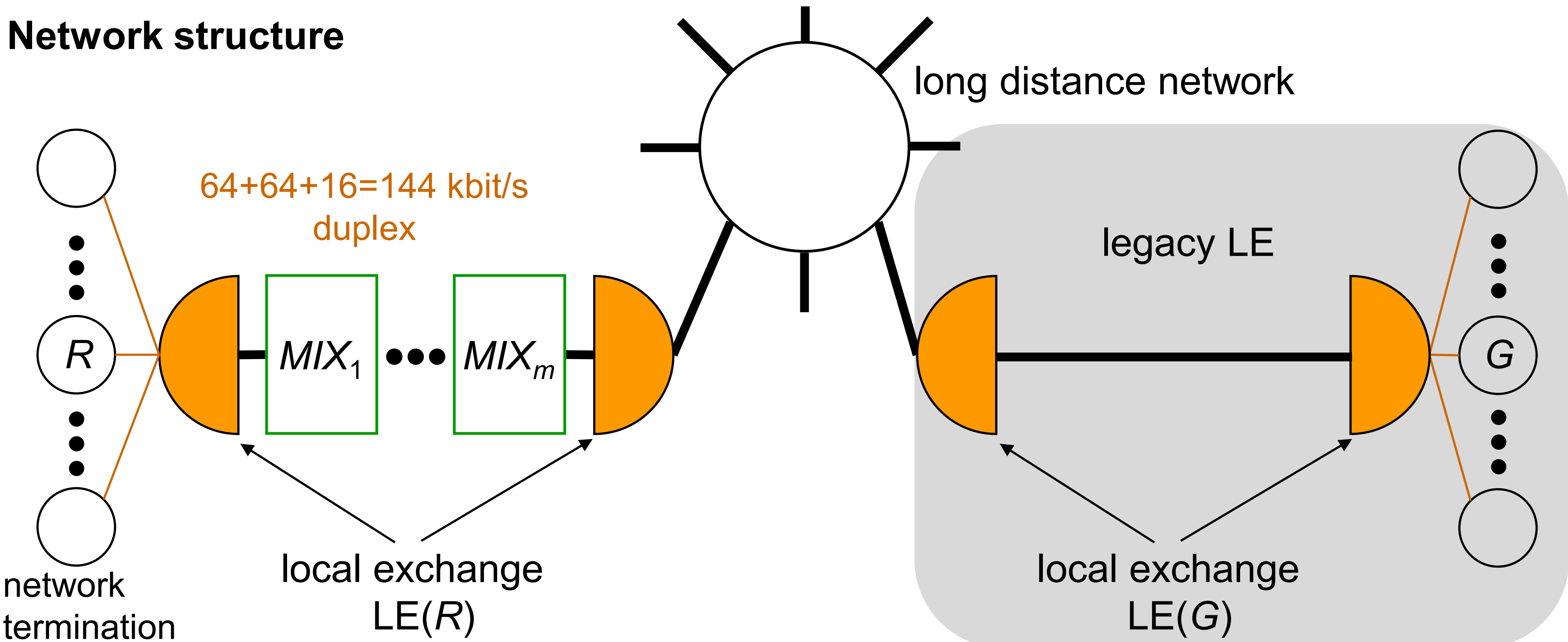
realizable without consideration of anonymity

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

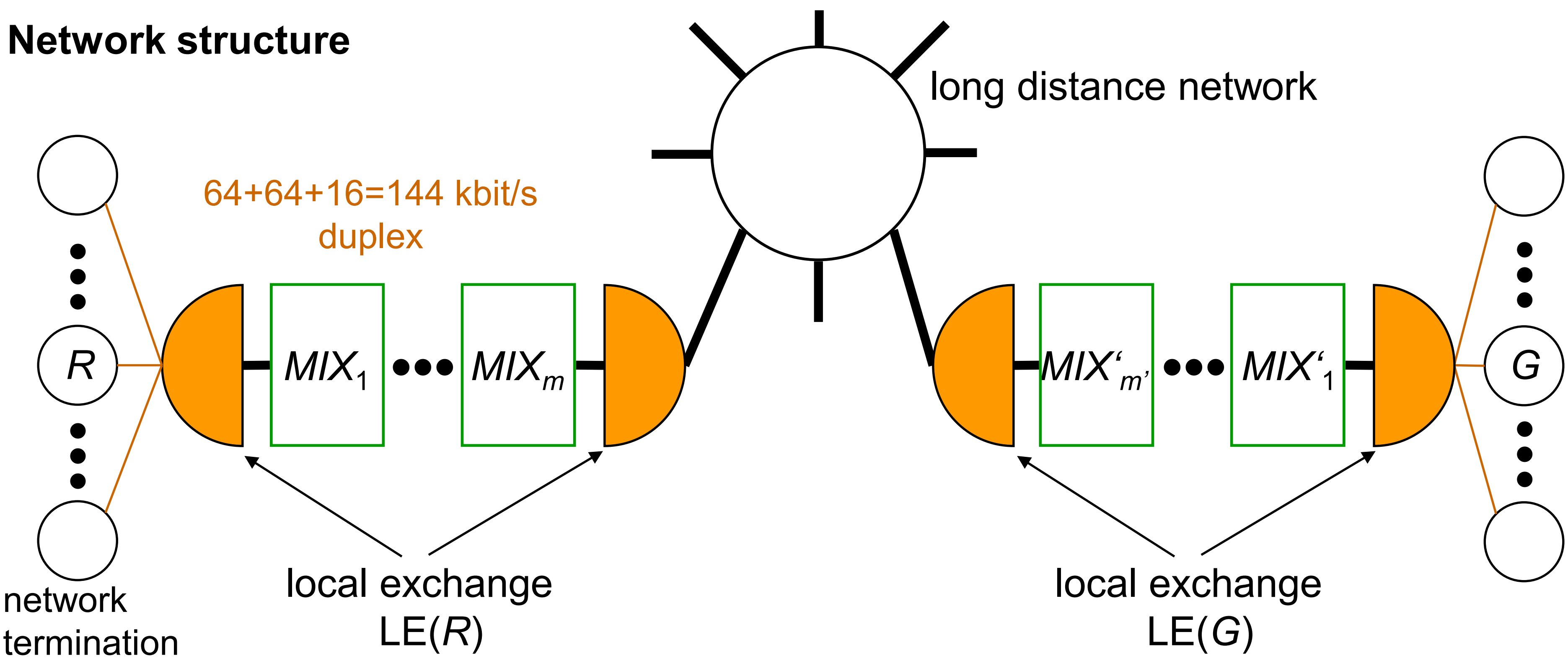
2 independent 64-kbit/s duplex channels on a 144-kbit/s subscriber line

hardly any additional delay on established channels

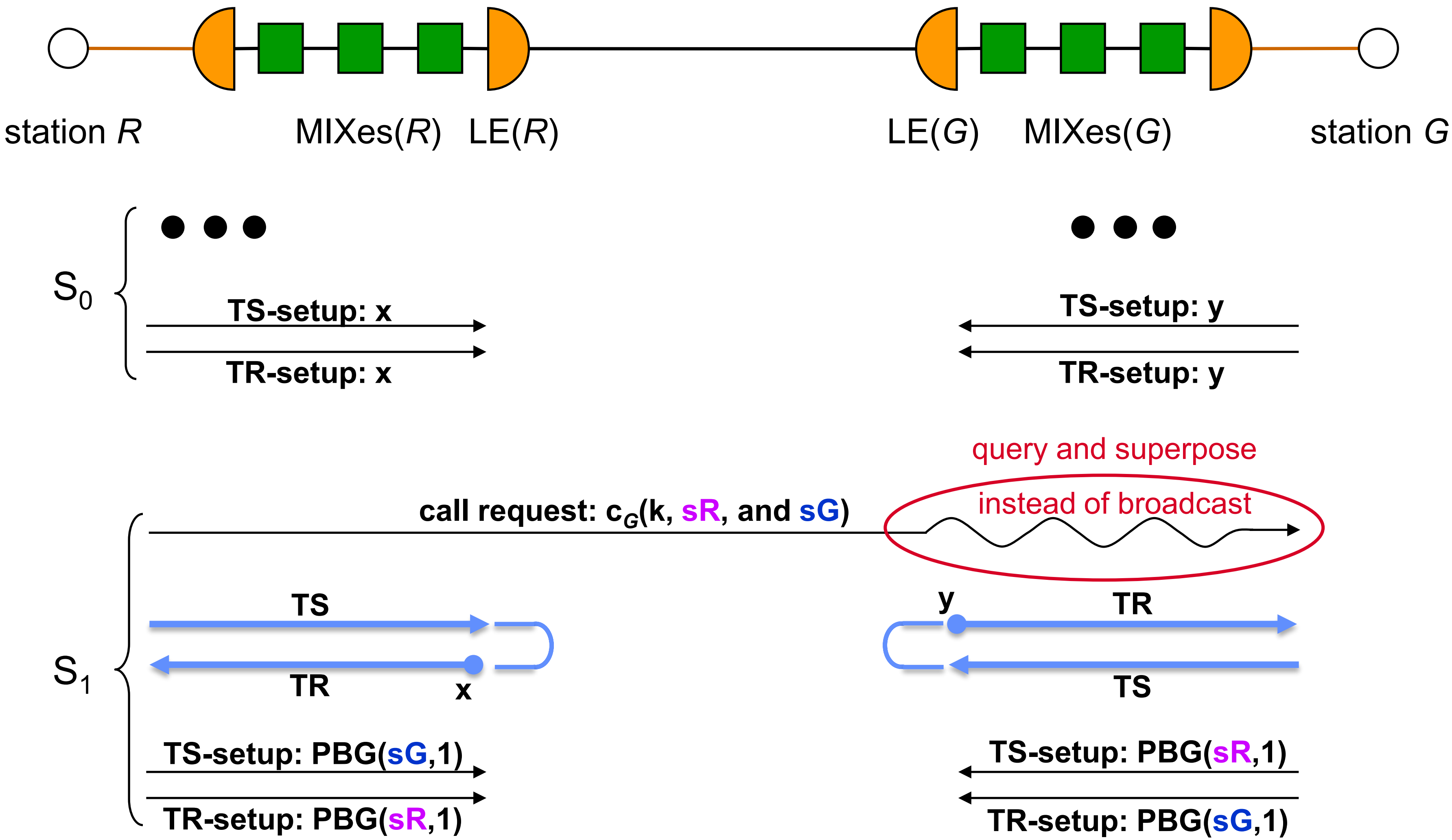
establish a channel within 3 s

no additional traffic on the long distance network

Network structure

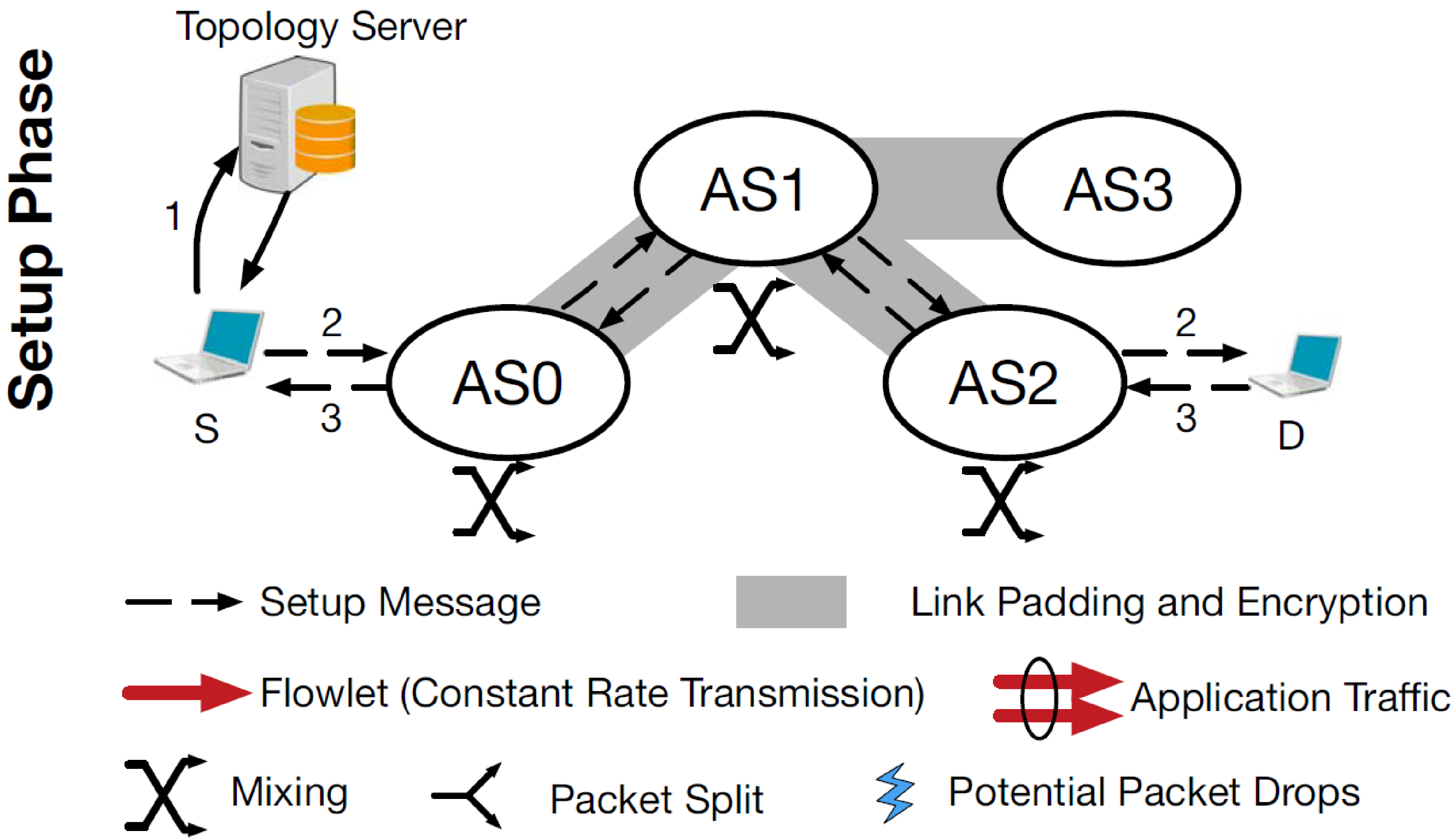


Time-slice channels (1)



Taranet

Chen et. al:: “TARANET: Traffic-Analysis Resistant Anonymity at the Network Layer”, 2018



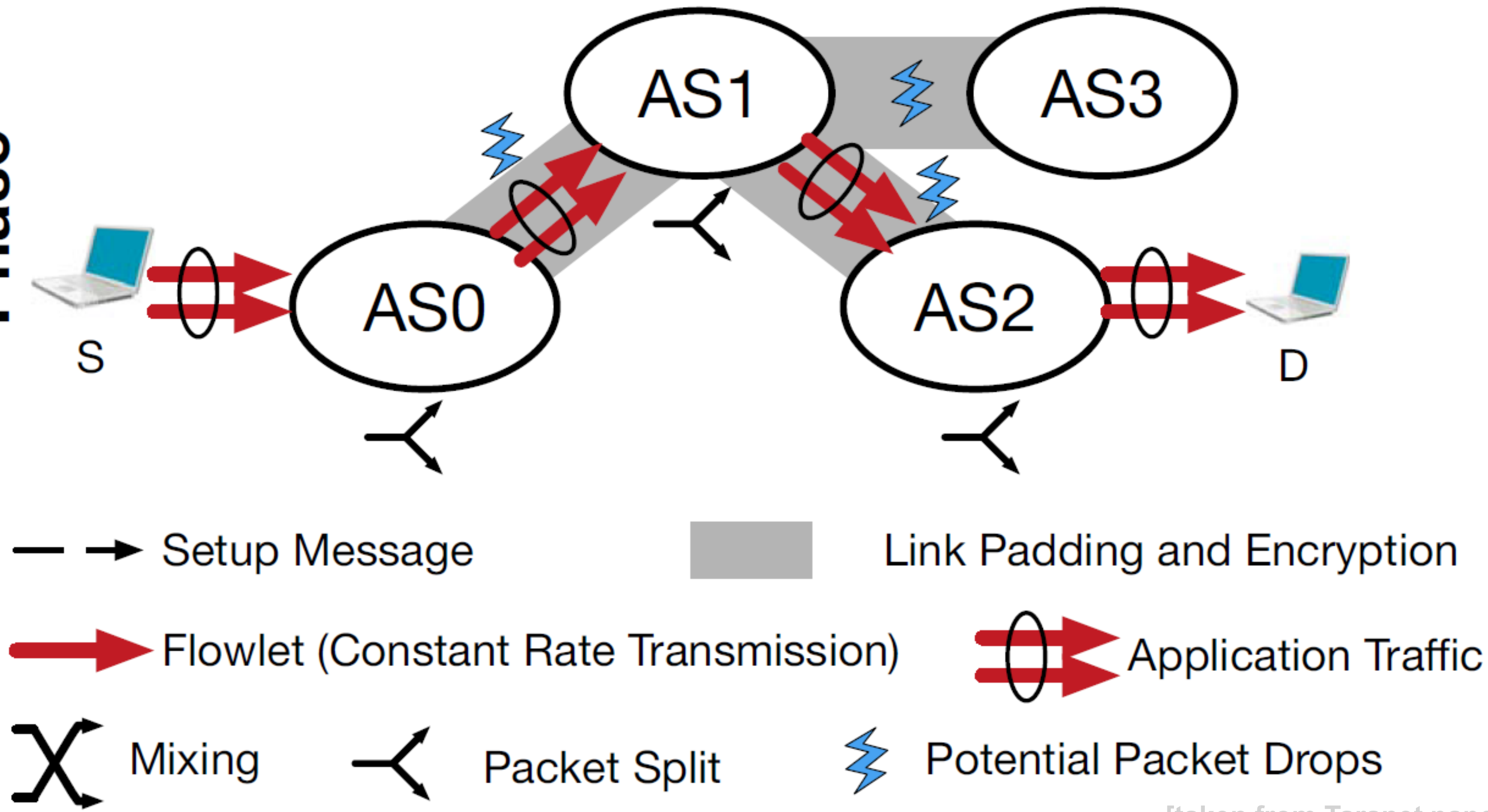
[taken from Taranet paper]

Taranet

Chen et. al:: “TARANET: Traffic-Analysis Resistant Anonymity at the Network Layer”, 2018

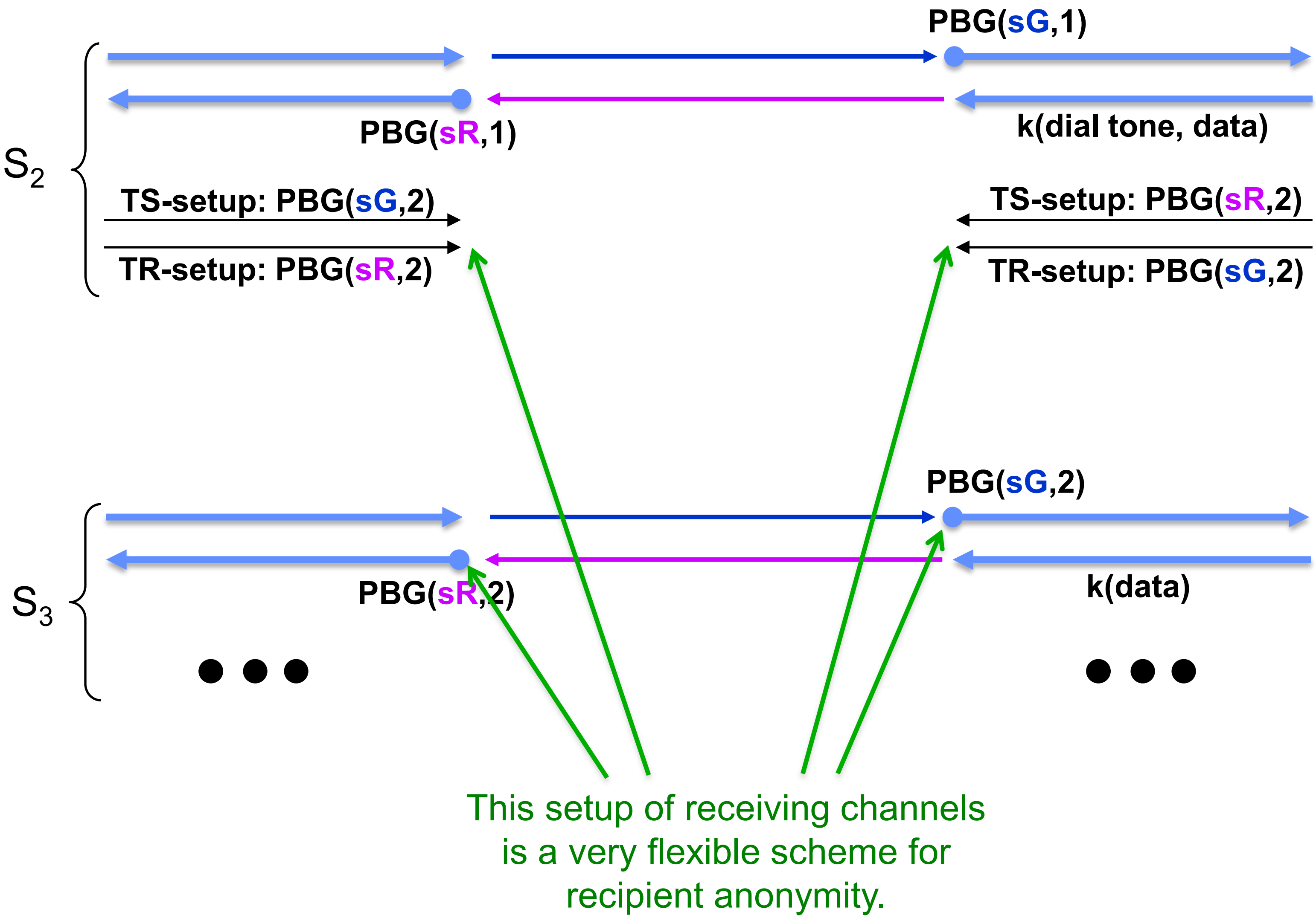
- main idea: splitting traffic into time slice channels (***flowlet***)
- ***Mix packet splitting*** for maintaining constant rate (dummy) traffic

Data Transmission Phase



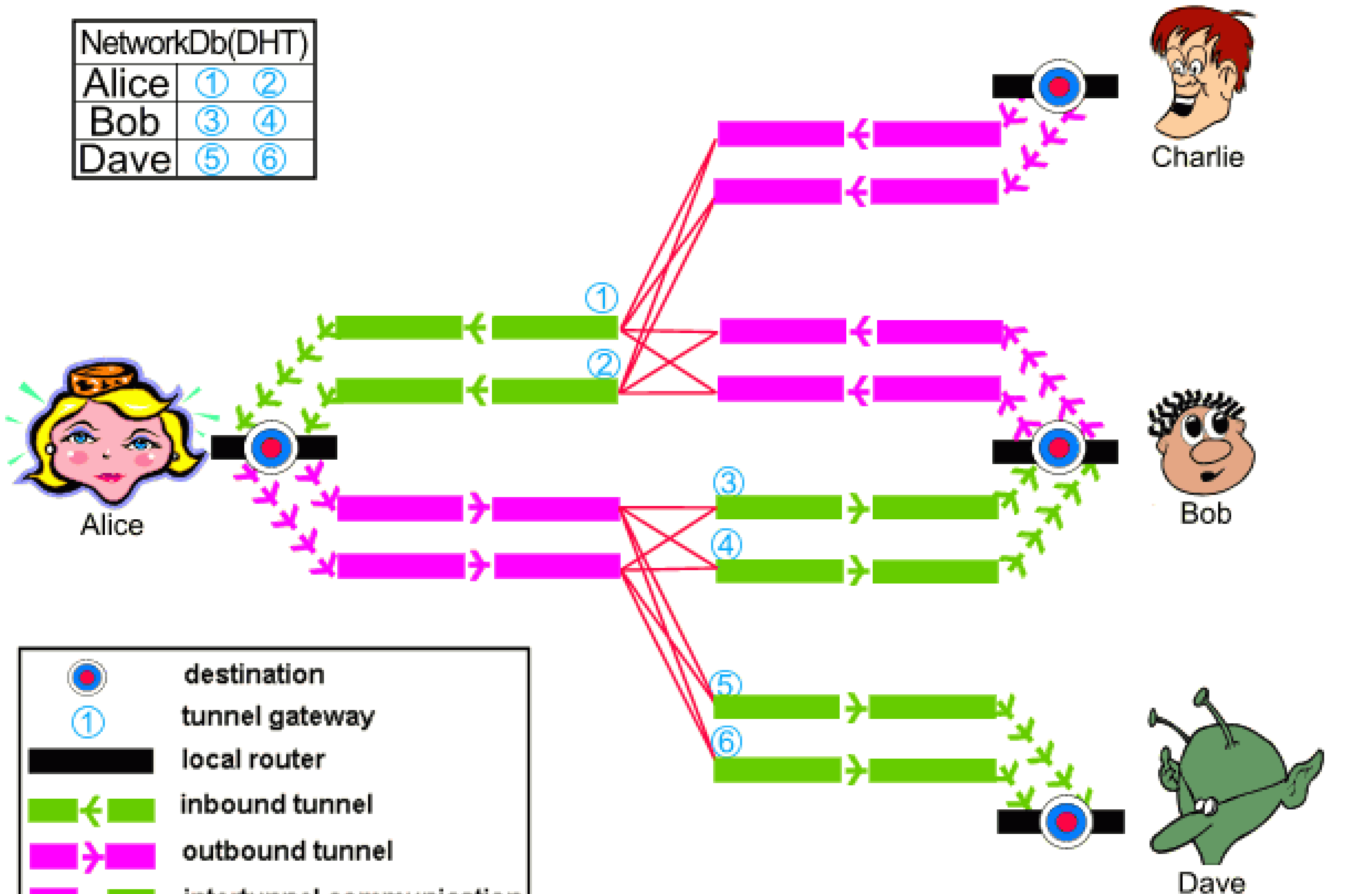
[taken from Taranet paper]

Time-slice channels (2)



I2P — Invisible Internet Project

geti2p.net



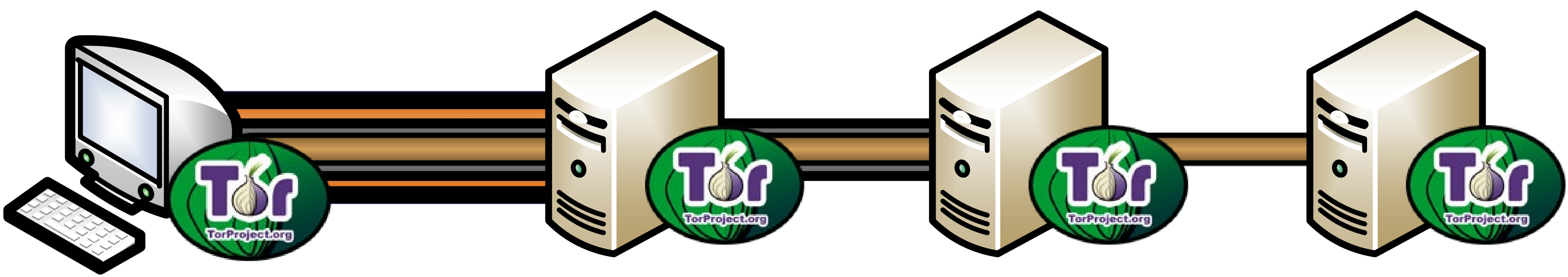
[https://geti2p.net/en/docs/how/intro]

Tor

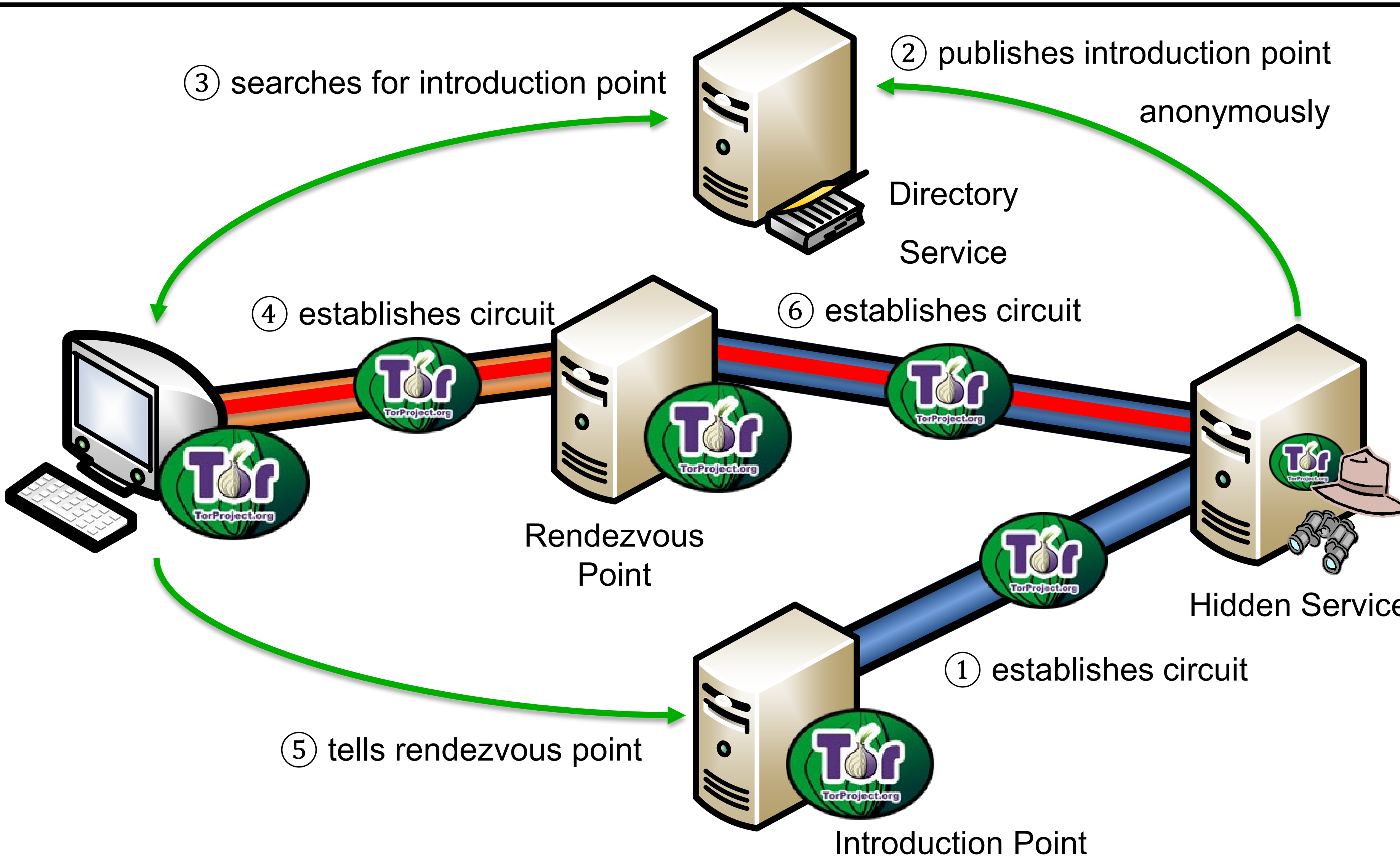
- basic building block:
  - symmetric encrypted channels → called: circuits
  - multiple streams multiplexed over one circuit

- Mix packet: cells
  - 512 bytes

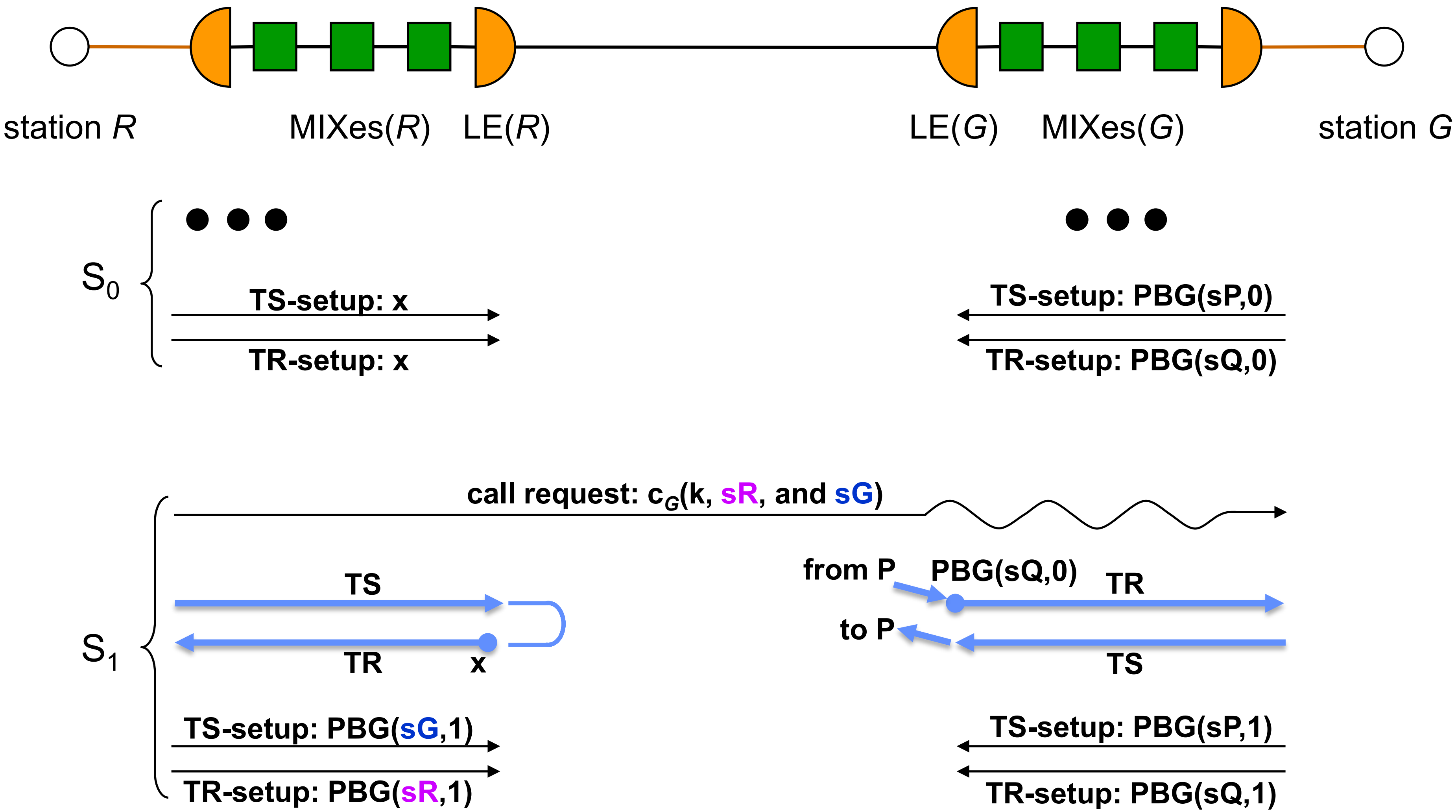
- asymmetric crypto for key exchange: Diffie-Hellman
  - telescopically
    - CREATE-Cell sent to next Tor node over already established circuit



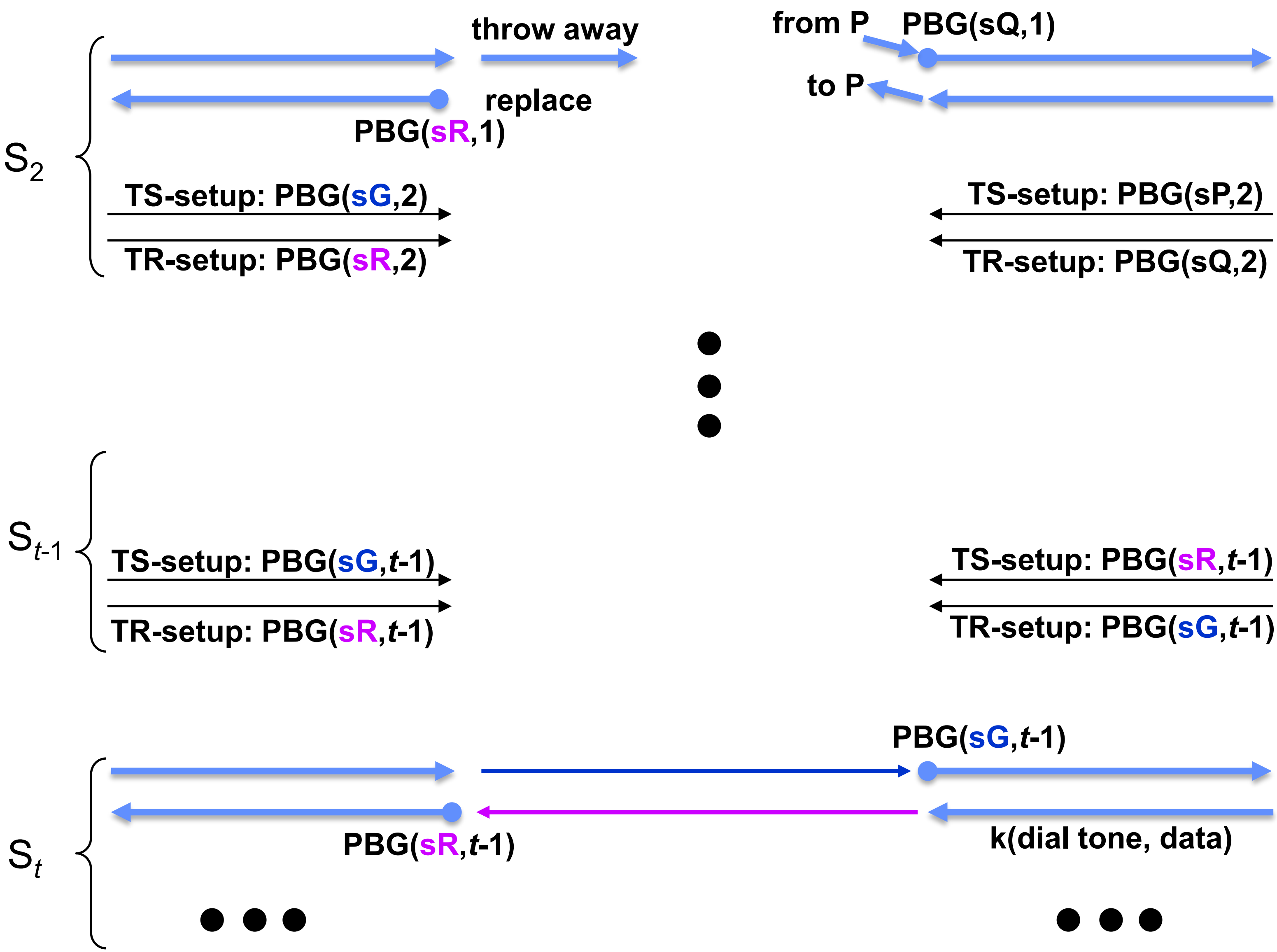
Tor: Hidden Services



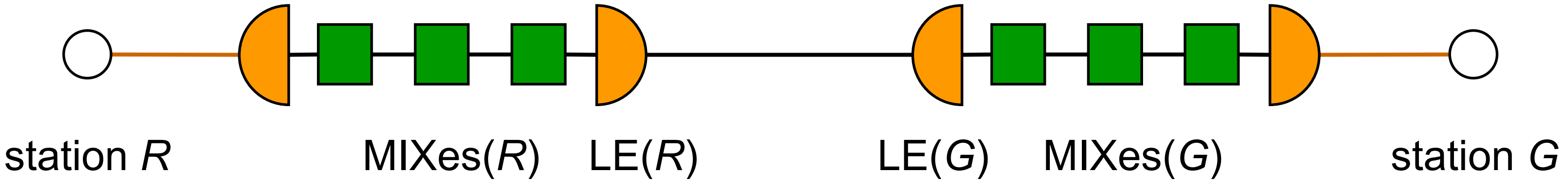
Connection configuration later (1)

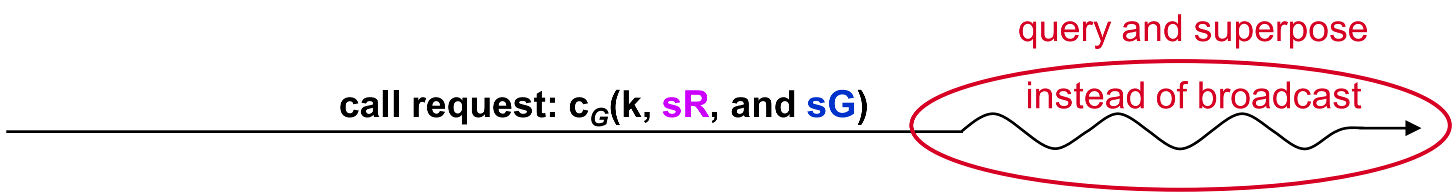


Connection configuration later (2)



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

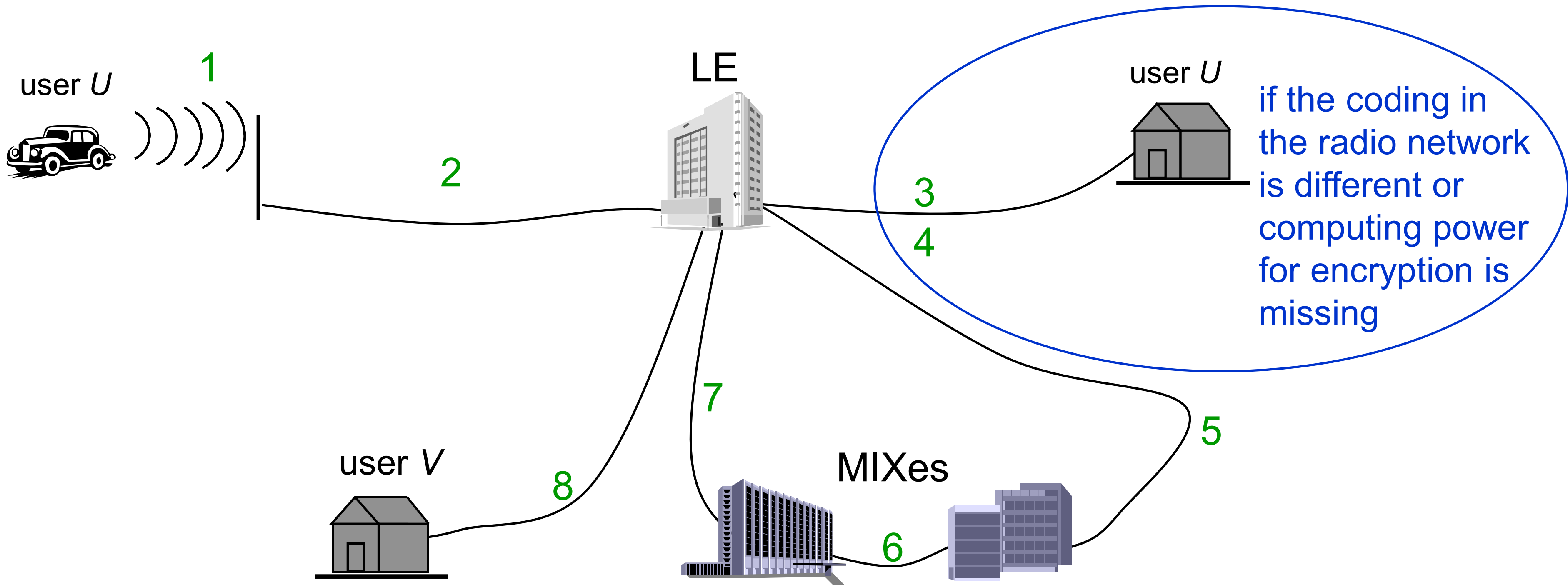
not commend-able

not applic-able

➔ 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



+ 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.

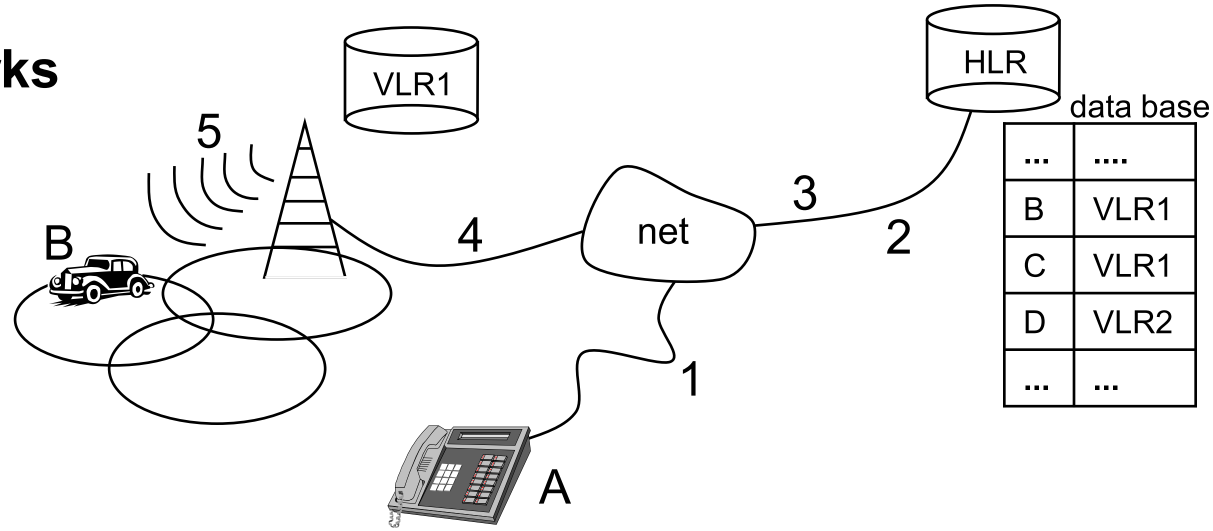
+ Filter + Generation of visible implicit addresses + Restrict the region

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

No movement profiles in radio networks

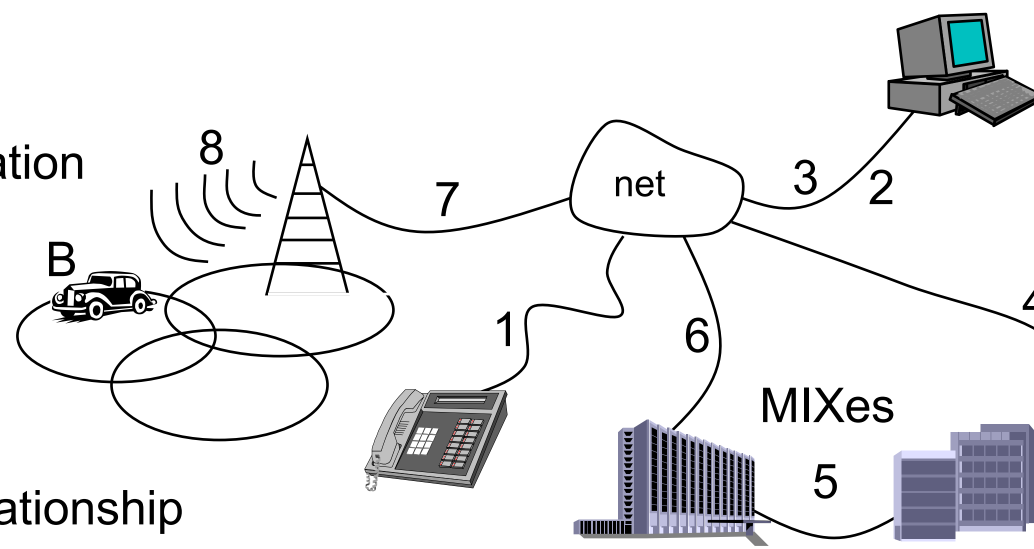
Cellular mobile networks

- 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



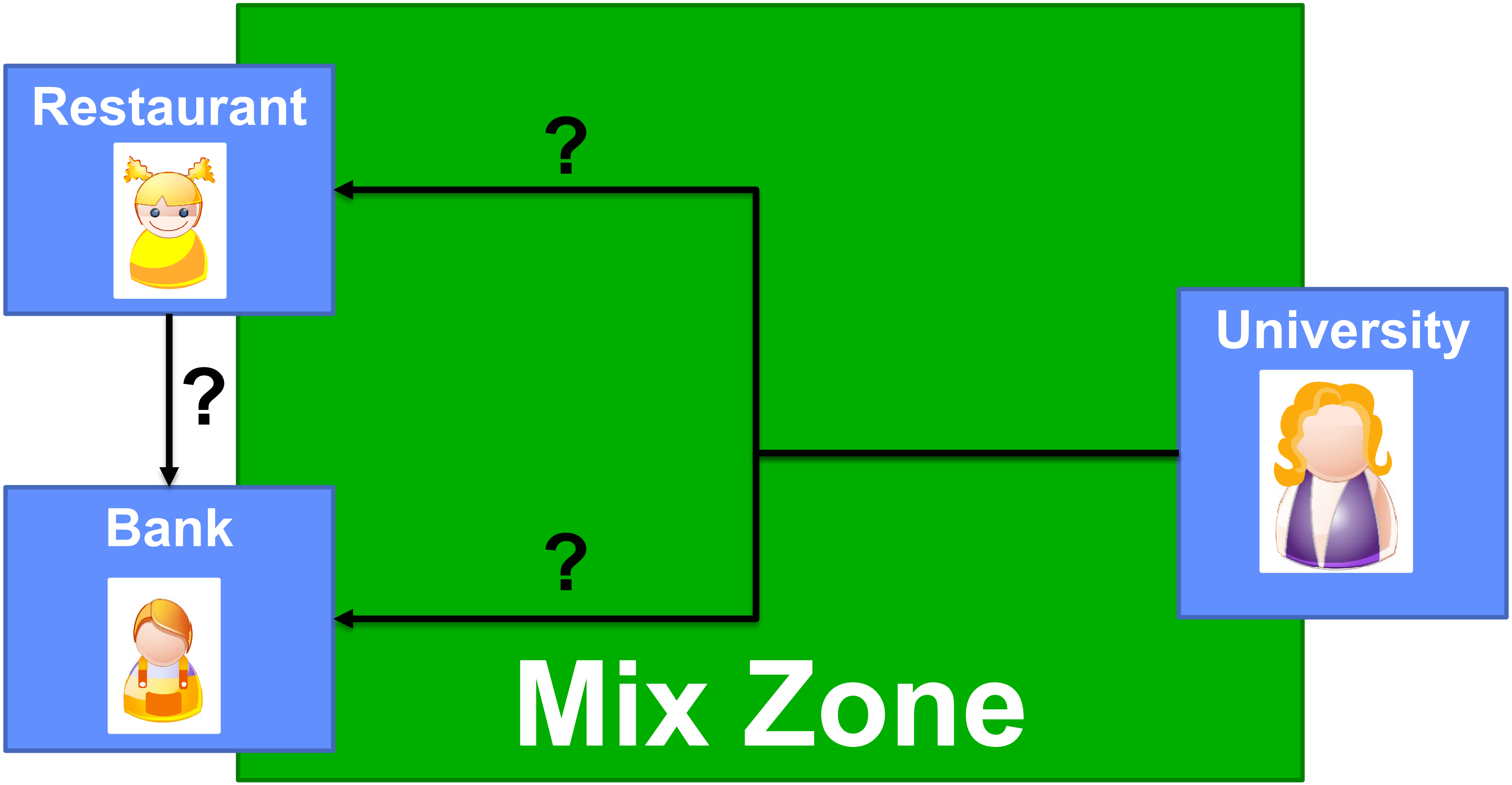
Mix Zones: User Privacy in Location-aware Services

[Alastair R. Beresford, Frank Stajano, 2004]

- 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!

Conclusions & Outlook (2)

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/

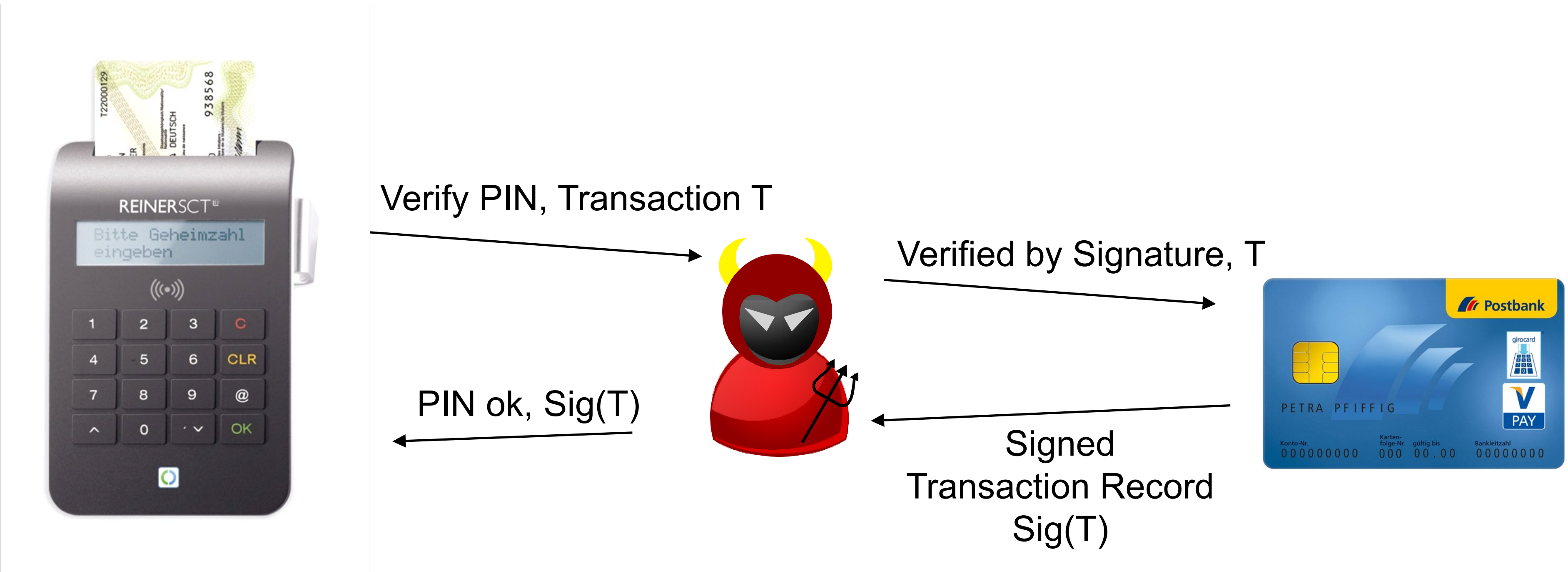
Chip & PIN Problem

Verify PIN, Transaction T

PIN ok,

Signed Transaction Sig (T)

Chip & PIN Problem



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.

  https://www.cl.cam.ac.uk/research/security/banking/

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)

Some Problems regarding Banking Cards

- **PIN** = **HEAD** (**DEC** (**DES** (*AccountNumber*)))
- **DEC** (***x***) = *x* mod 10
  - {0123456789ABCDEF} ➔ {0123456789012345}
- **HEAD** (***x***): `if (x < 1000) x = x + 1000`
  - **0**… → **1**…
- **HSM** (*PIN*, *AccountNumber* , **DEC**) ➔ { true, false }
  - Attack:
    - **DEC**: {0123456789ABCDEF} → {0000000100000000}
    - **if** (**HSM** (‚0000‘, *AccountNumber*, **DEC**)) == True ➔ no ‚7‘ in PIN

Security properties of digital payment systems

digital (integrity, availability)

Payment system is **secure** if

via communication network
immaterial, digital

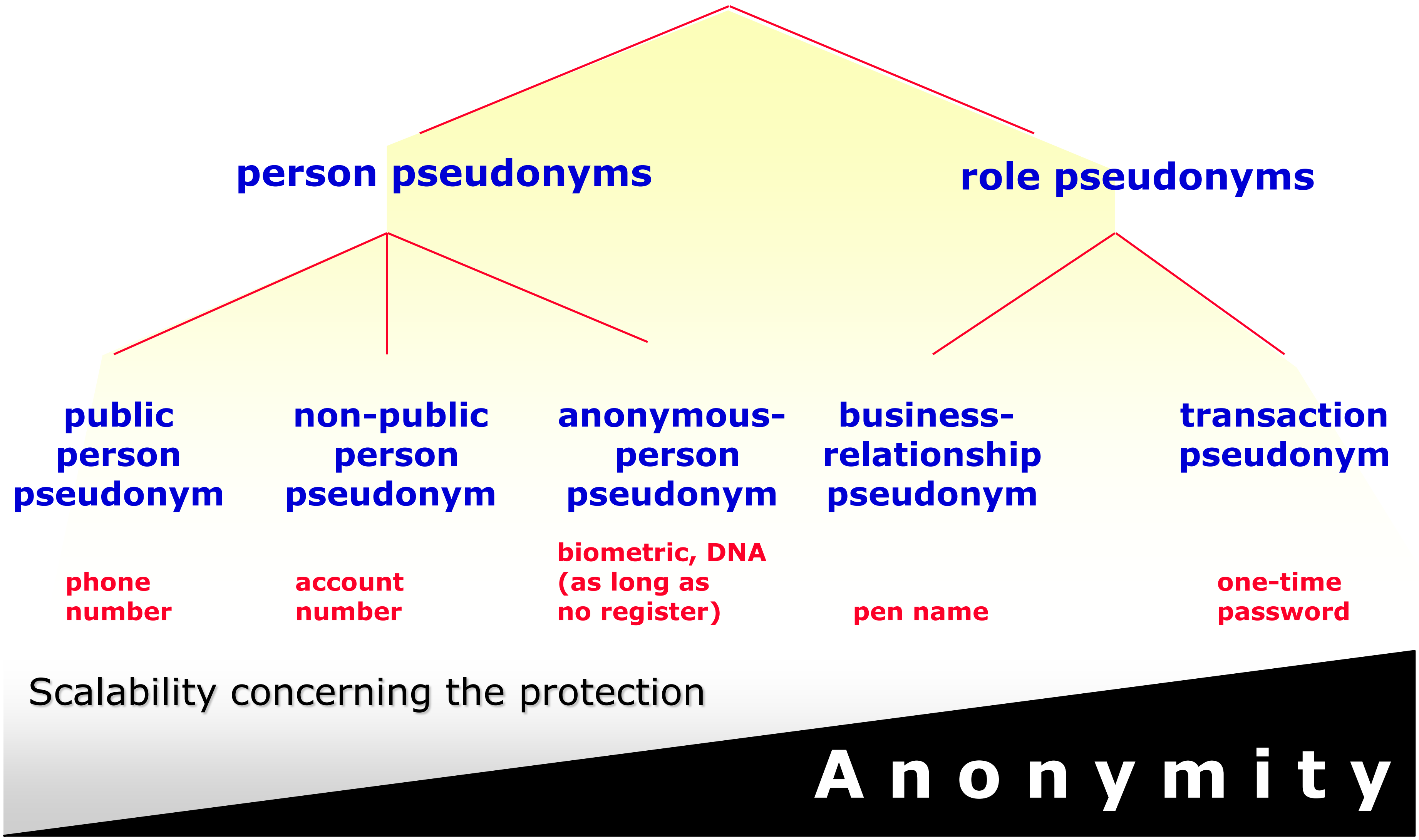
- user can transfer the rights received,
- user can loose a right only if he is willing to,
- if a user who is willing to pay uniquely denotes another user as recipient, only this entity receives the right,
- user can prove transfers of rights to a third party if necessary (receipt problem), and
- the 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

Pseudonyms examples



Pseudonyms: Linkability in detail

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.

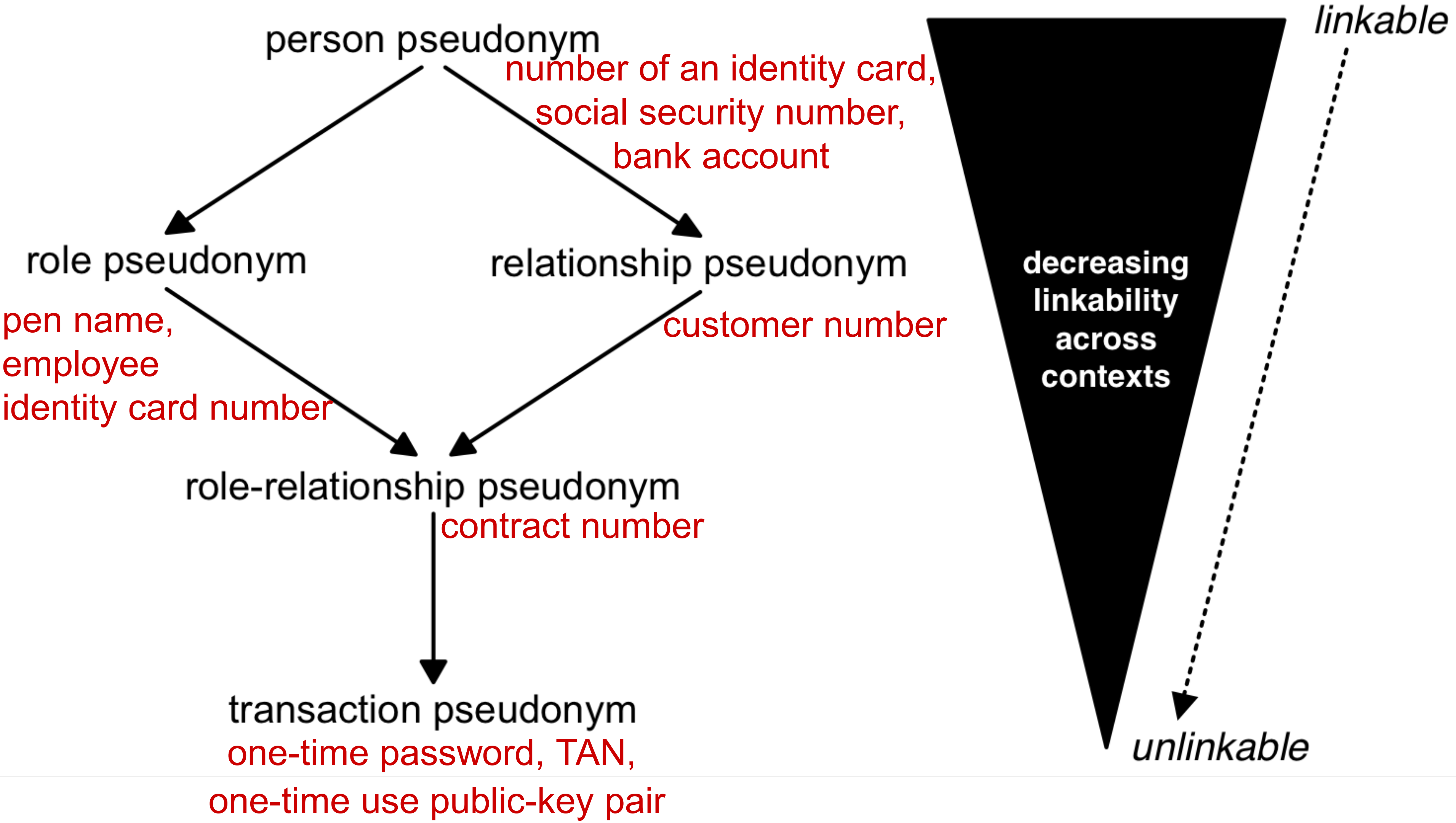
Bank account with bank as trustee for identity,
Credit card number ...

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:

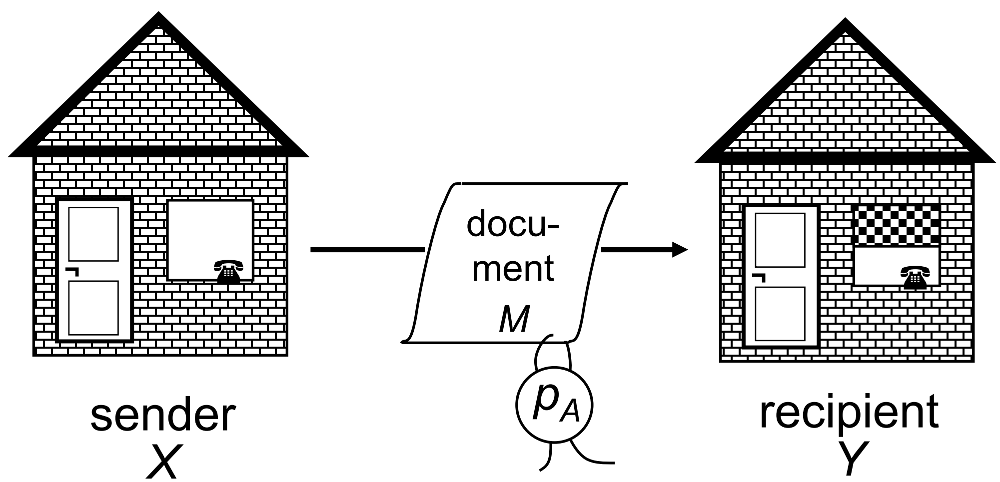
$s_A(M)$

X —— $M, s_A(M)$ ——→ Y

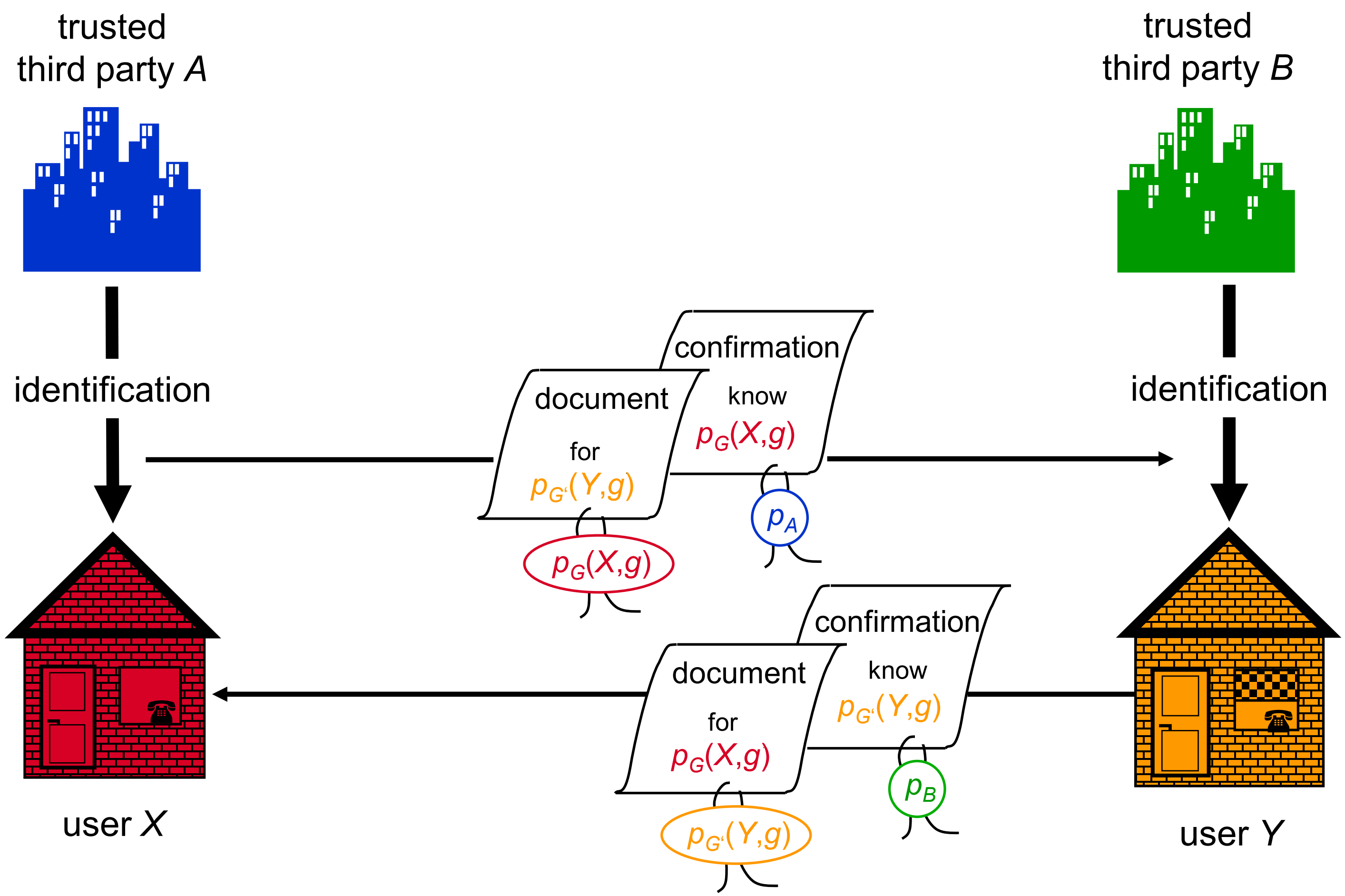
test the
signature:

$t_A(M, s_A(M))$?

graphical notation



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



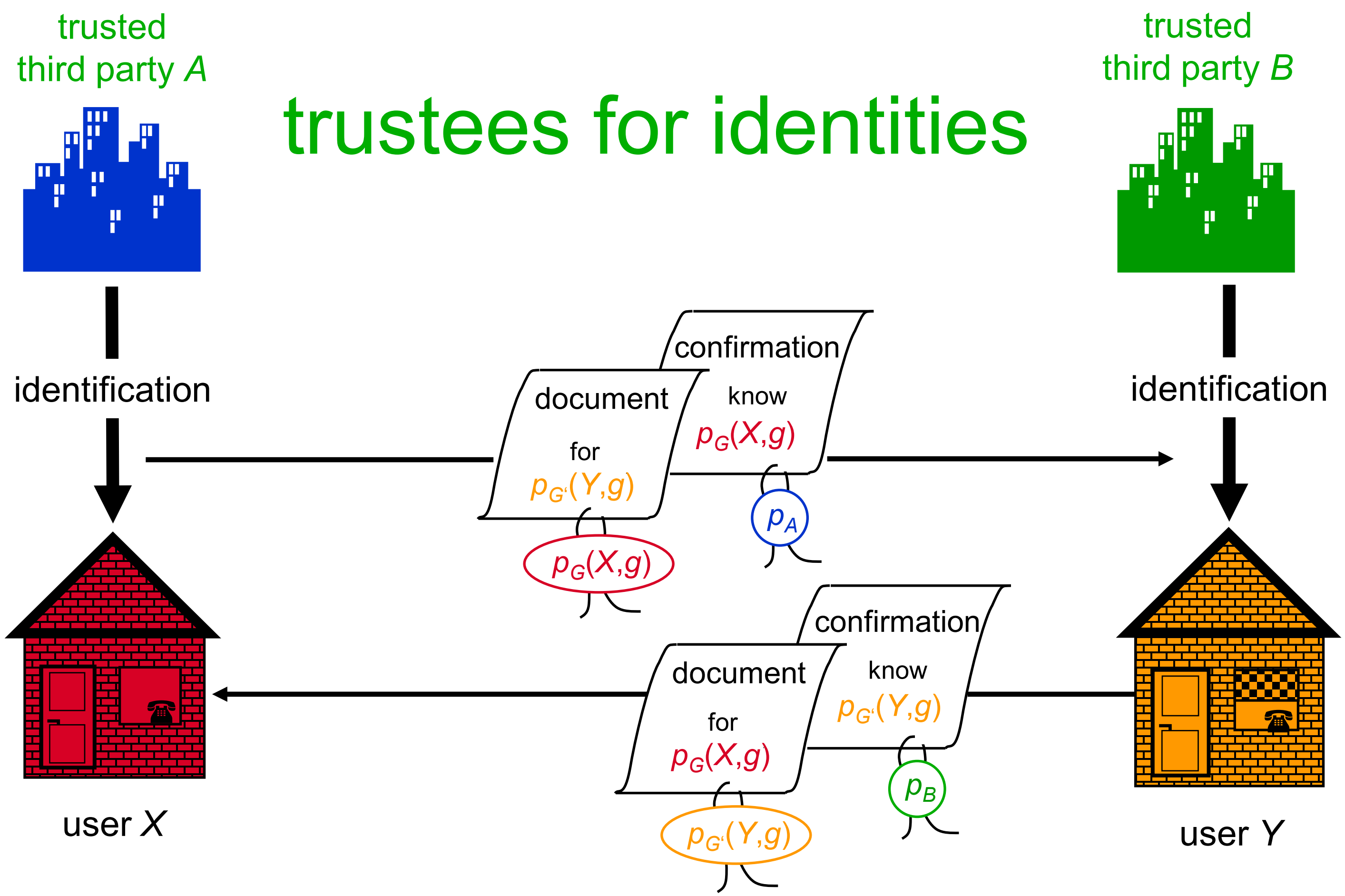
Generalization:

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

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

error / attack tolerance (cf. MIXes)

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



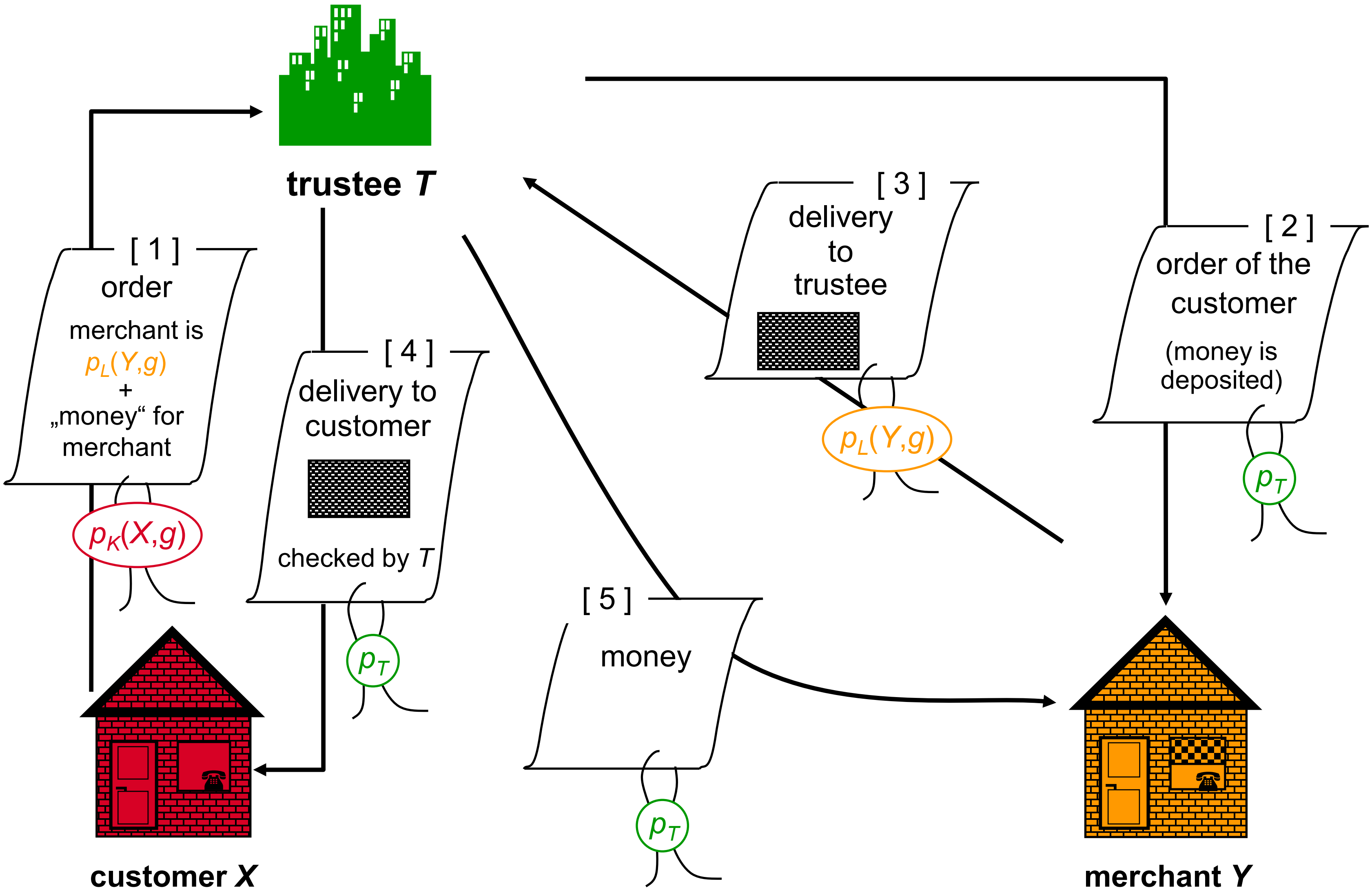
Generalization:

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

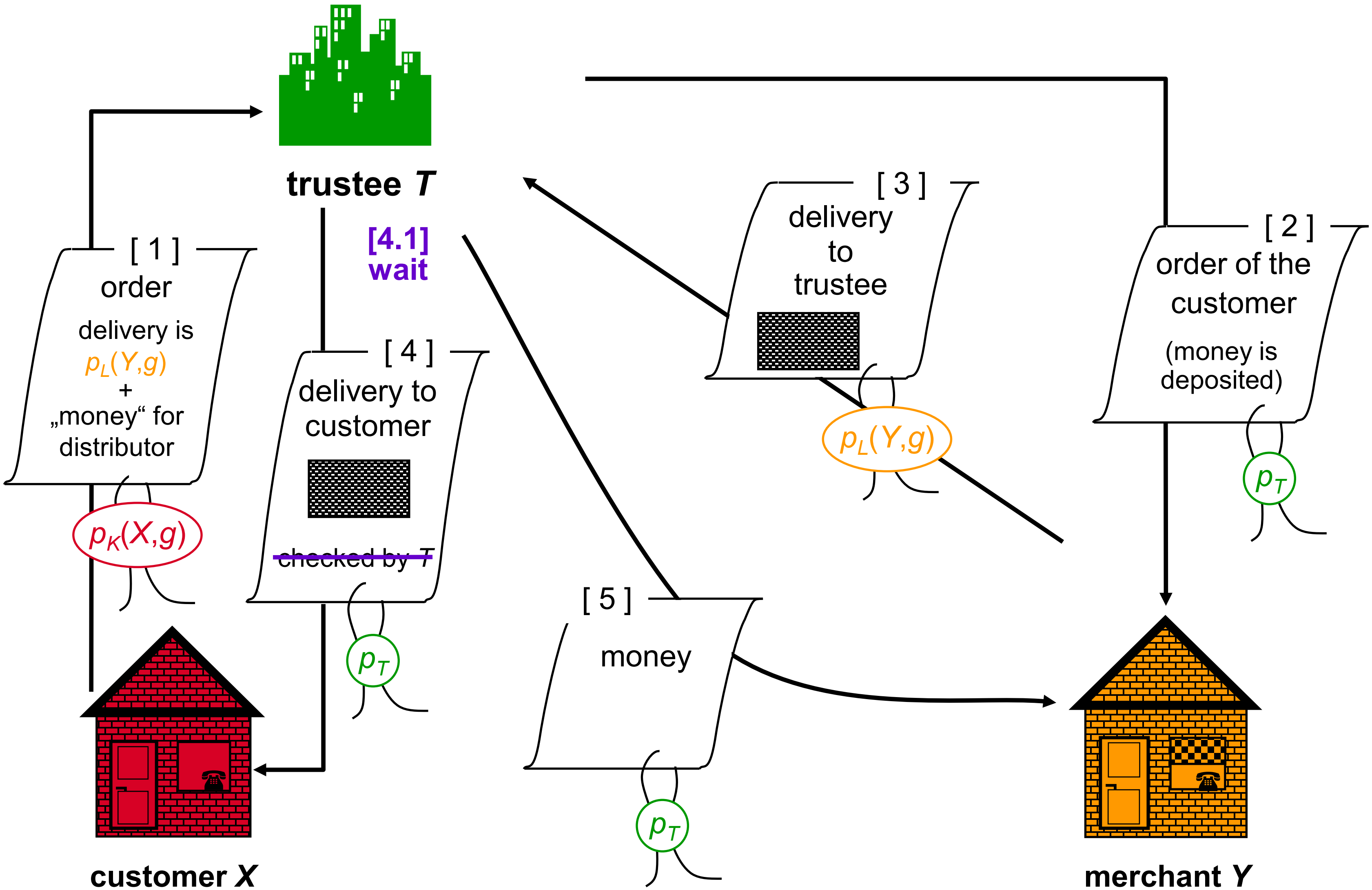
$$B'_1 \rightarrow B'_2 \rightarrow \ldots \rightarrow B'_m$$

error / attack tolerance (cf. MIXes)

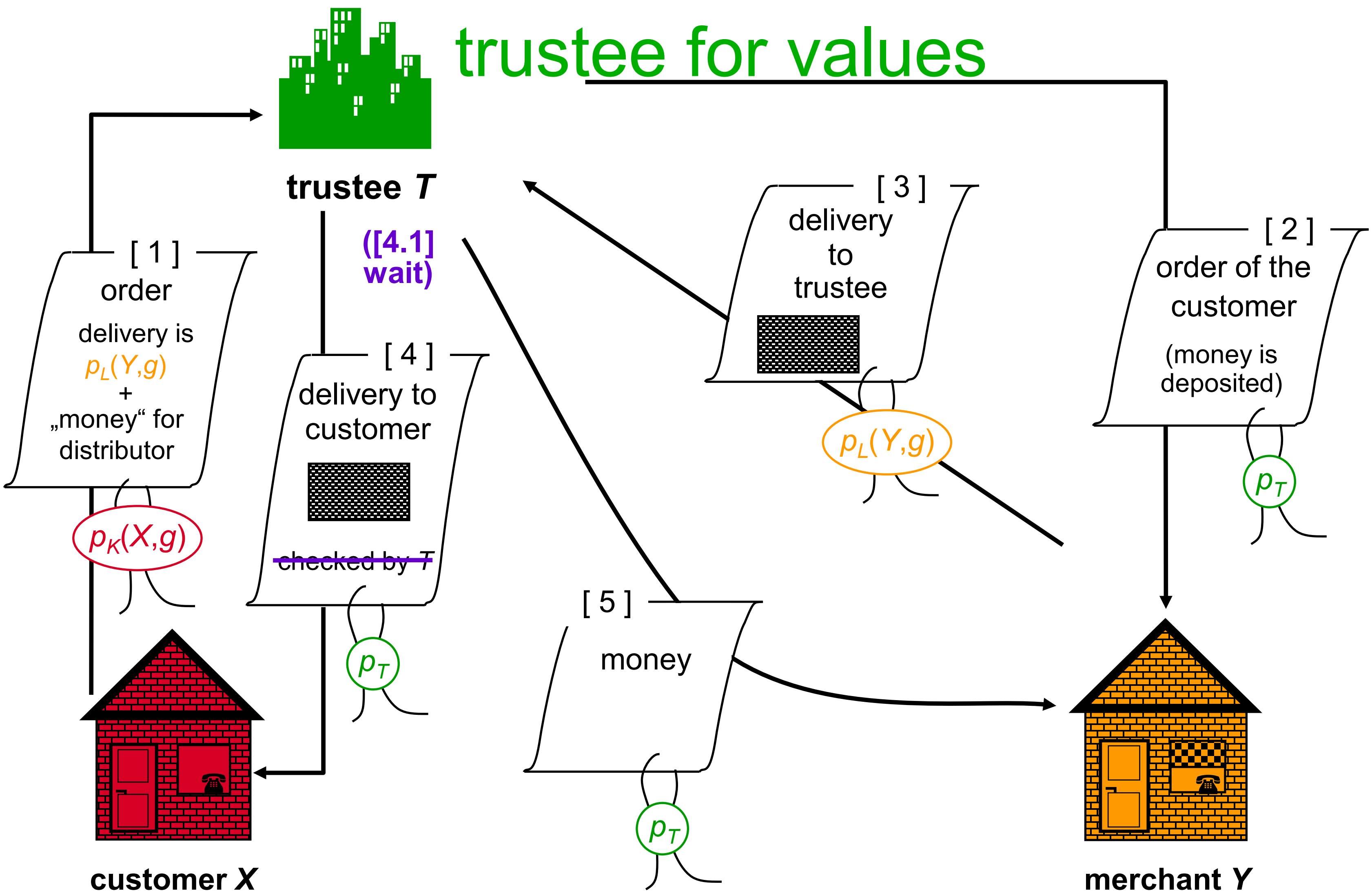
Security for completely anonymous business partners using active trustee who can check the goods



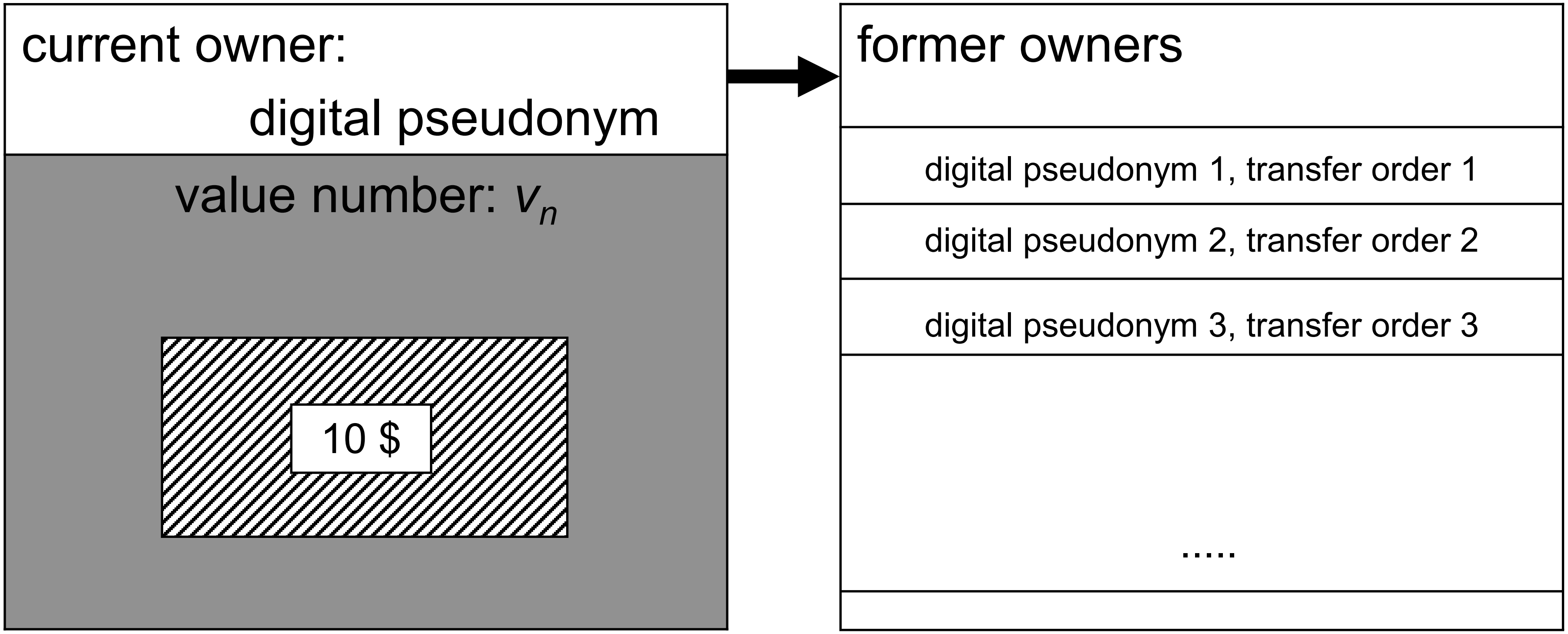
Security for completely anonymous business partners using active trustee who can not check the goods



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



Anonymously transferable standard values



Anonymously transferable standard value

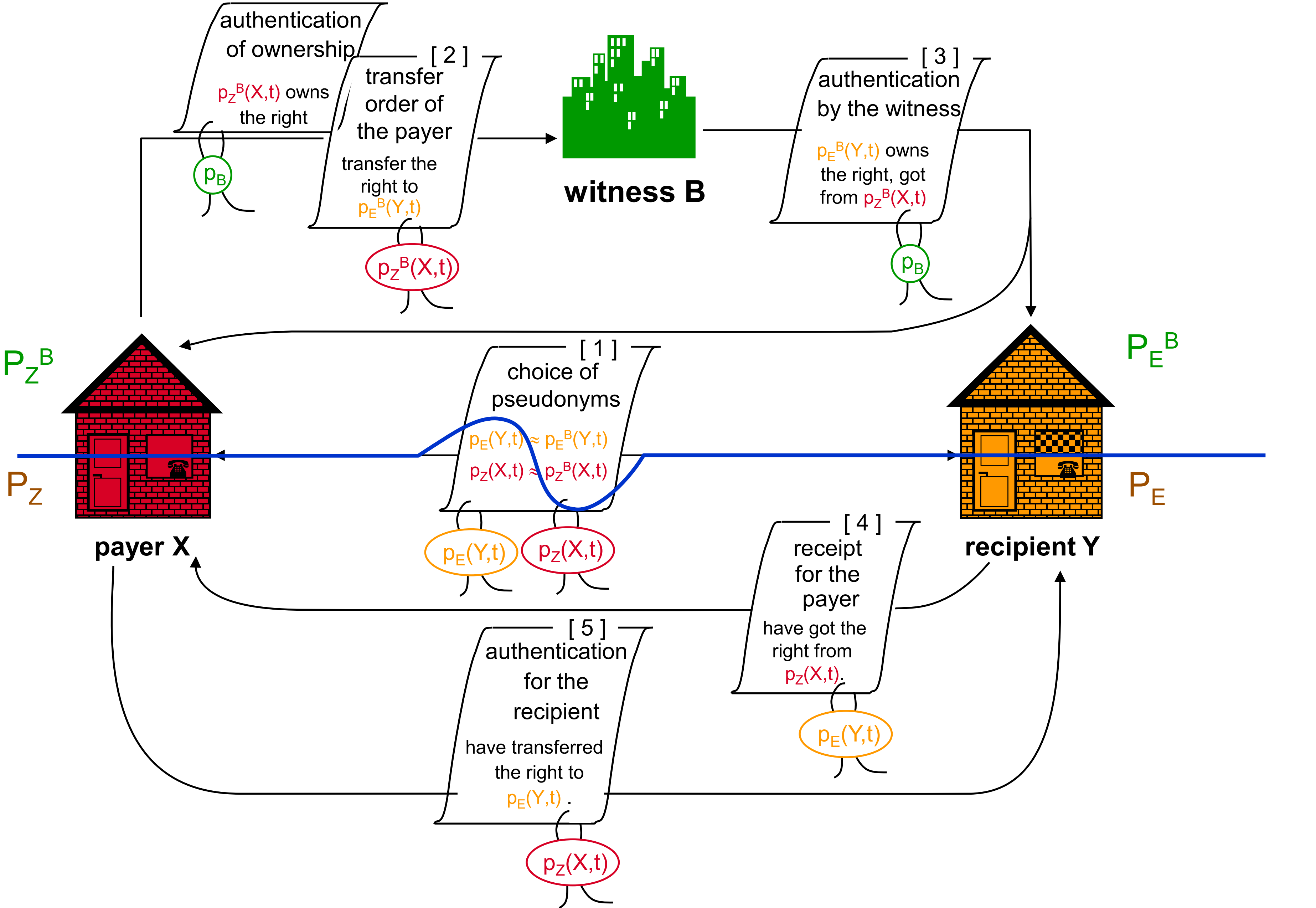
Bitcoin – a decentral payment system

[Satoshi Nakamoto: Bitcoin: A Peer-to-Peer Electronic Cash System. 2008]

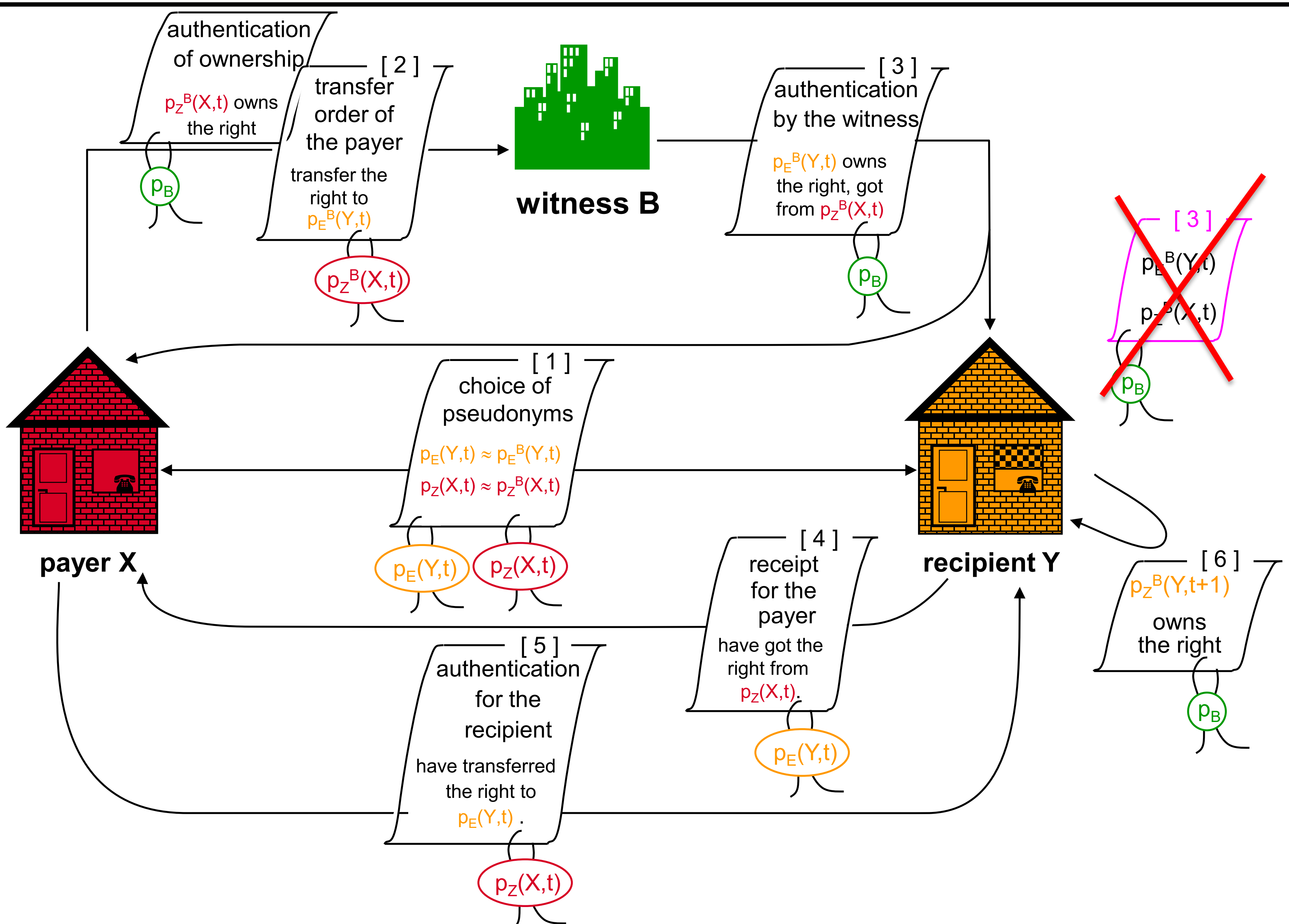
- Key feature: Bitcoin transfer between pseudonyms (Bitcoin addresses)
- Bitcoin pseudonym ≡ public key of ECDSA
- Sender signs transfer
- Double spending protection:
  - Bitcoin network keeps history of all transactions
  - Transactions have timestamps → only oldest is valid
    - Bitcoin network works as "distributed time server"
  - Binding of transaction and timestamp: „proof-of-work“[1]:
    - search for **z**: Hash(*Transaction, Timestamp,* **z**) = 00000… (0|1)* < *w*
    - *w* adjusted over timer
- https://www.blockchain.info

[1]Cynthia Dwork, Moni Naor: „Pricing via Processing or Combatting Junk Mail “, CRYPTO 1992

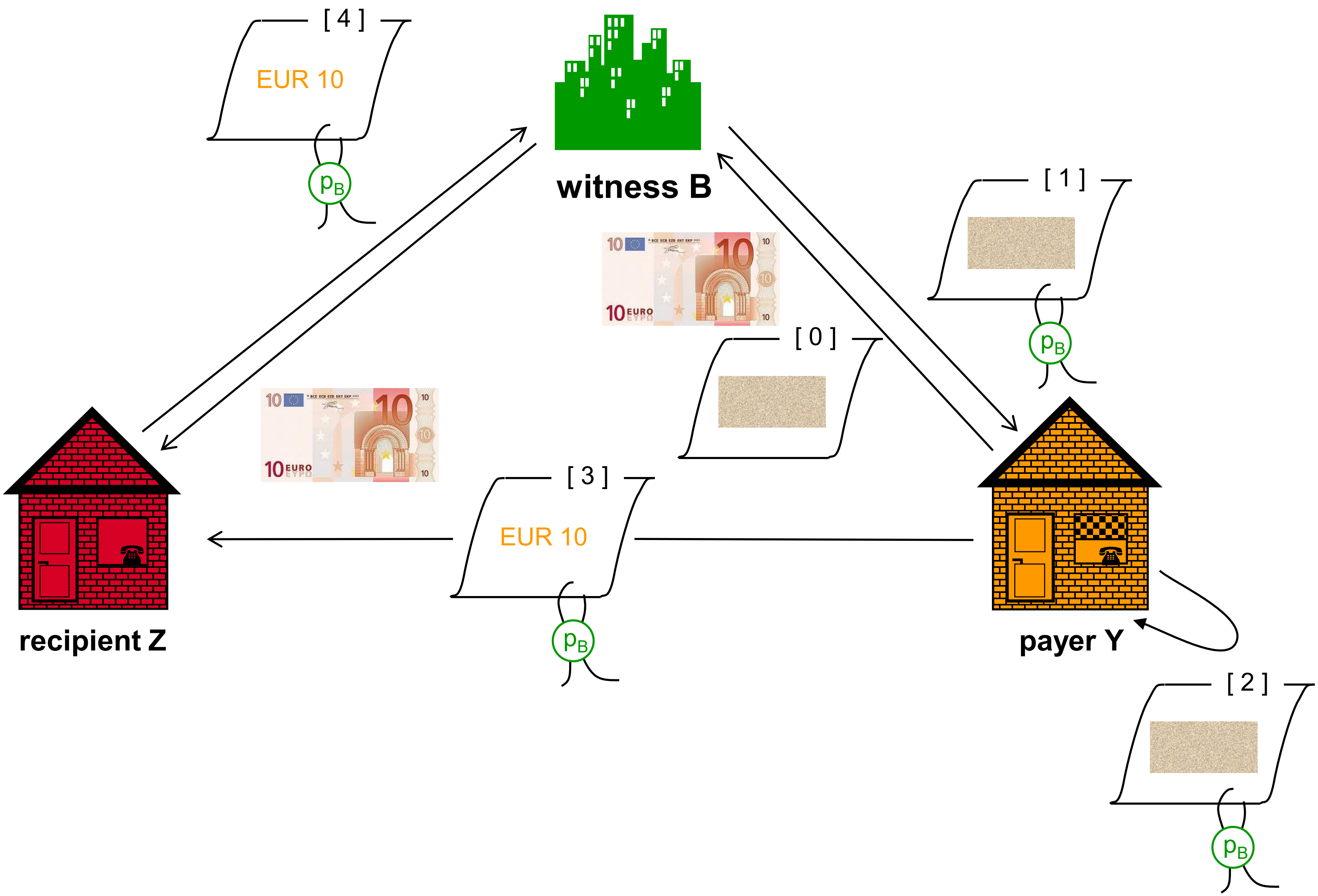
Basic scheme of a secure and anonymous digital payment system



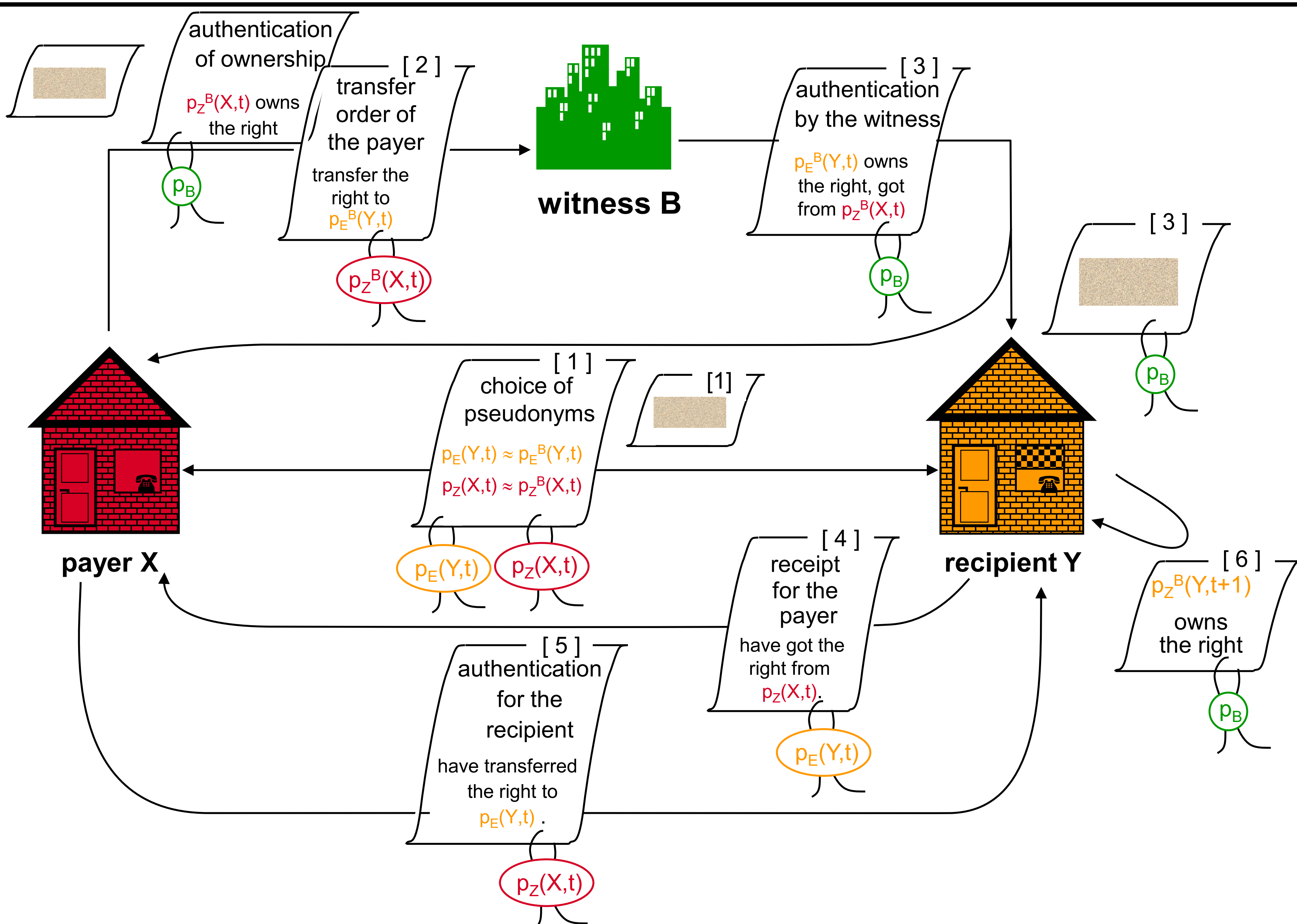
Transformation of the authentication by the witness



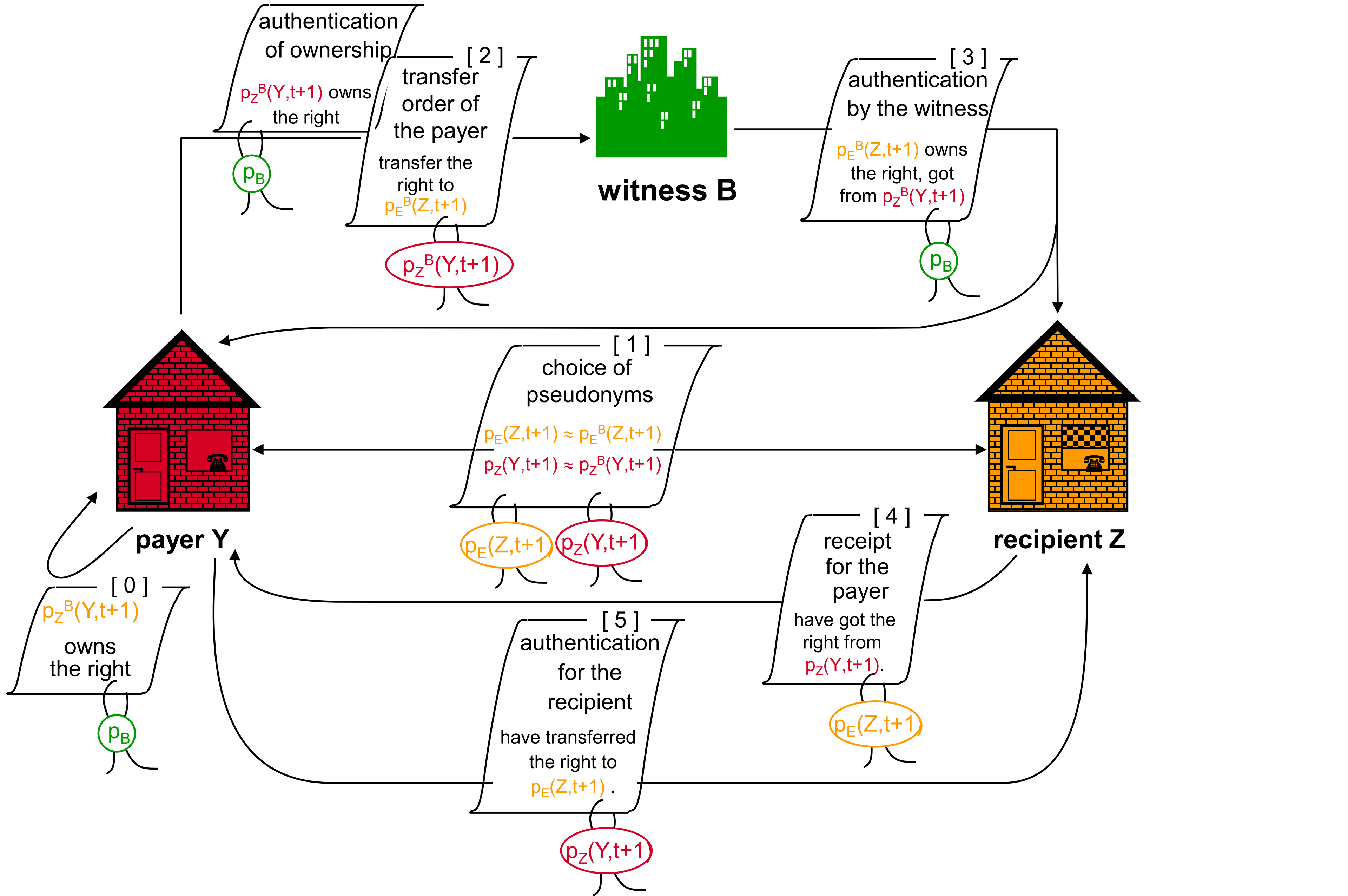
Transformation of the authentication by the witness: Simplified Steps



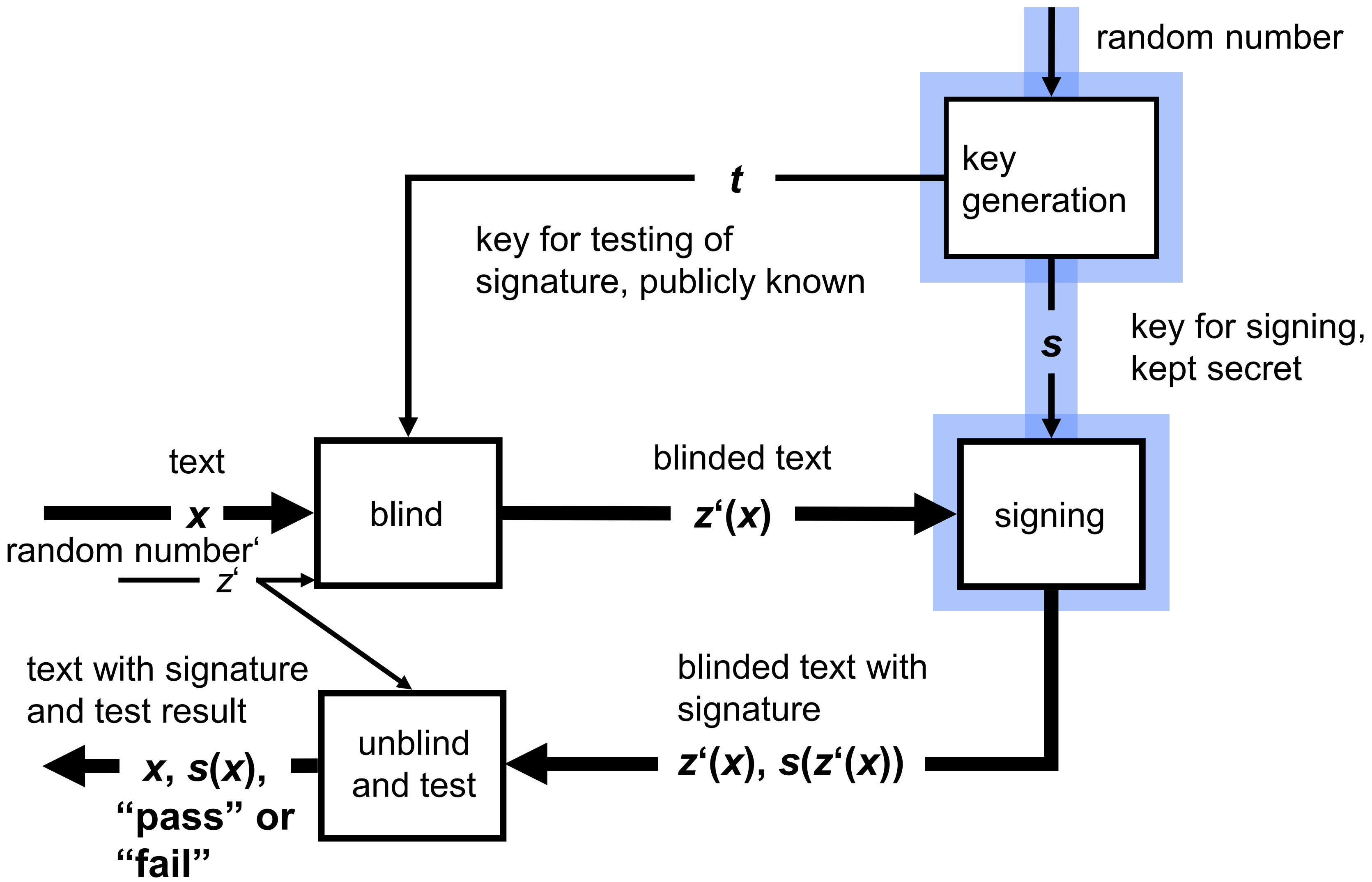
Transformation of the authentication by the witness



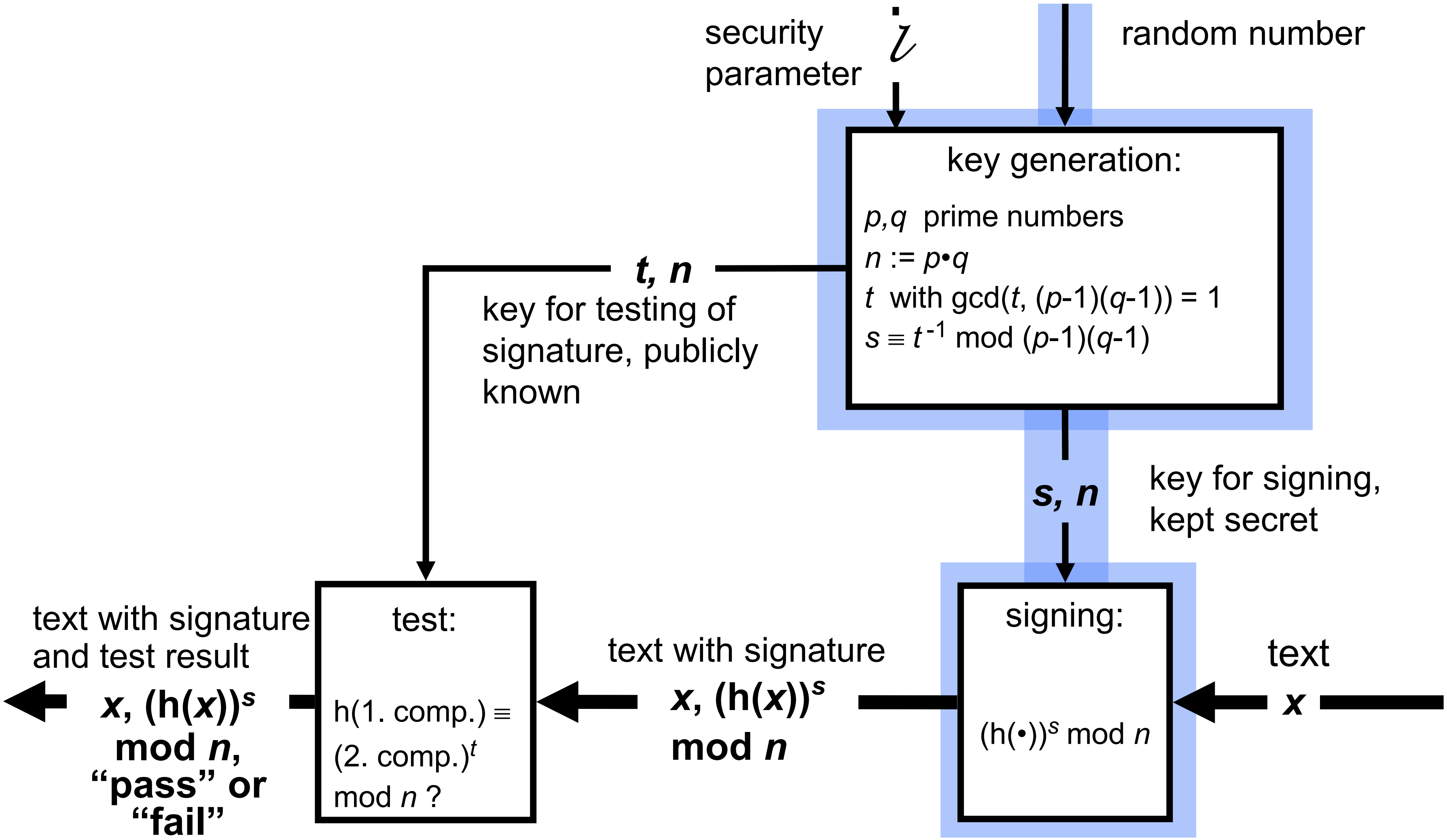
The next round: Y in the role payer to recipient Z



Signature system for signing blindly



RSA as digital signature system with collision-resistant hash function h



One time convertible authentication

Recipient

choose pseudonym

p

(test key of arbitrary sign. system)

Collision-resistant hash function **h**

$p, h(p)$

choose $r \in_R Z_n^*$

$(p, h(p)) \cdot r^t$ → $((p, h(p)) \cdot r^t)^s$

$(p, h(p))^s \cdot r$ ←

multiply with

r^{-1}

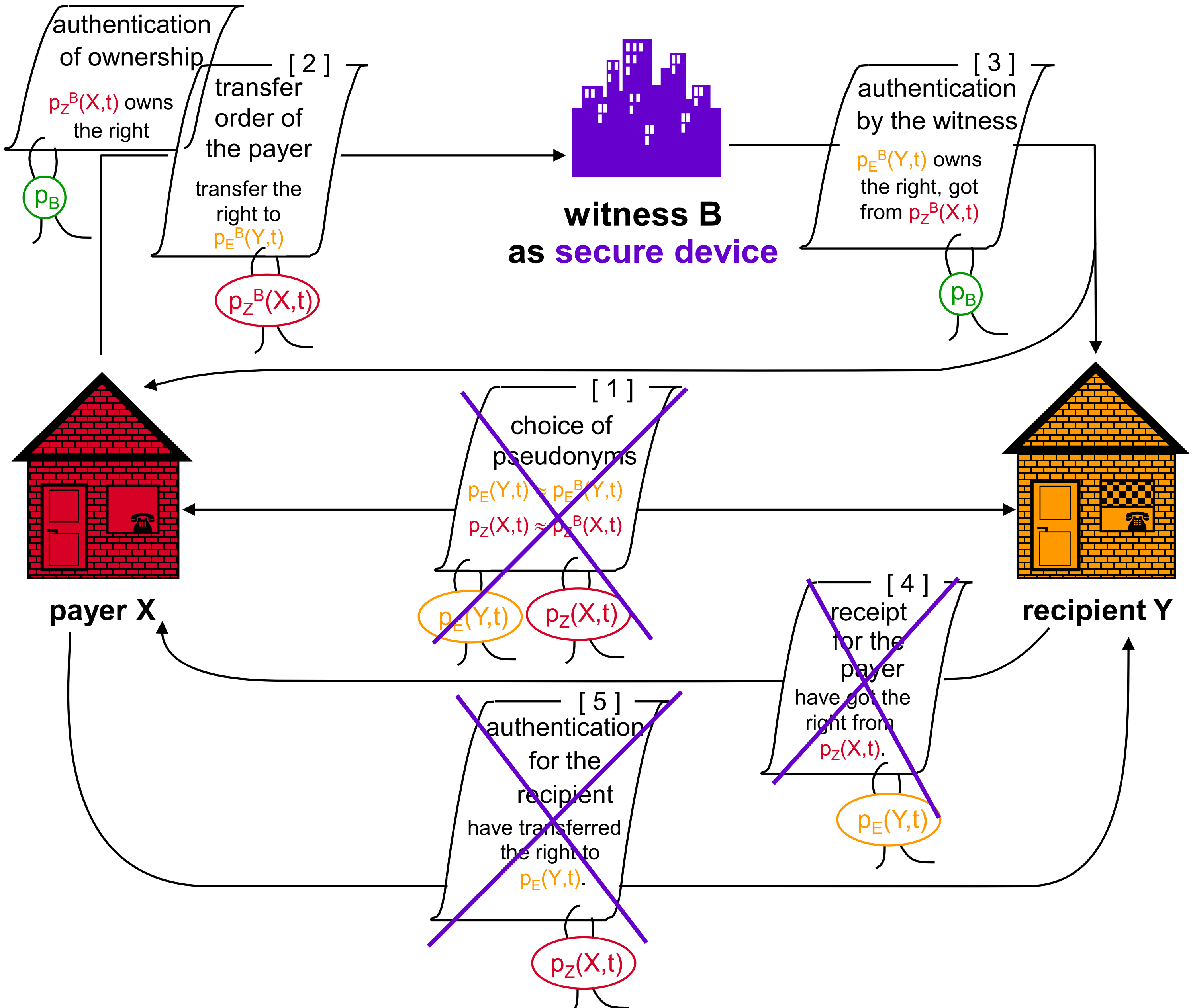
get

$(p, h(p))^s$

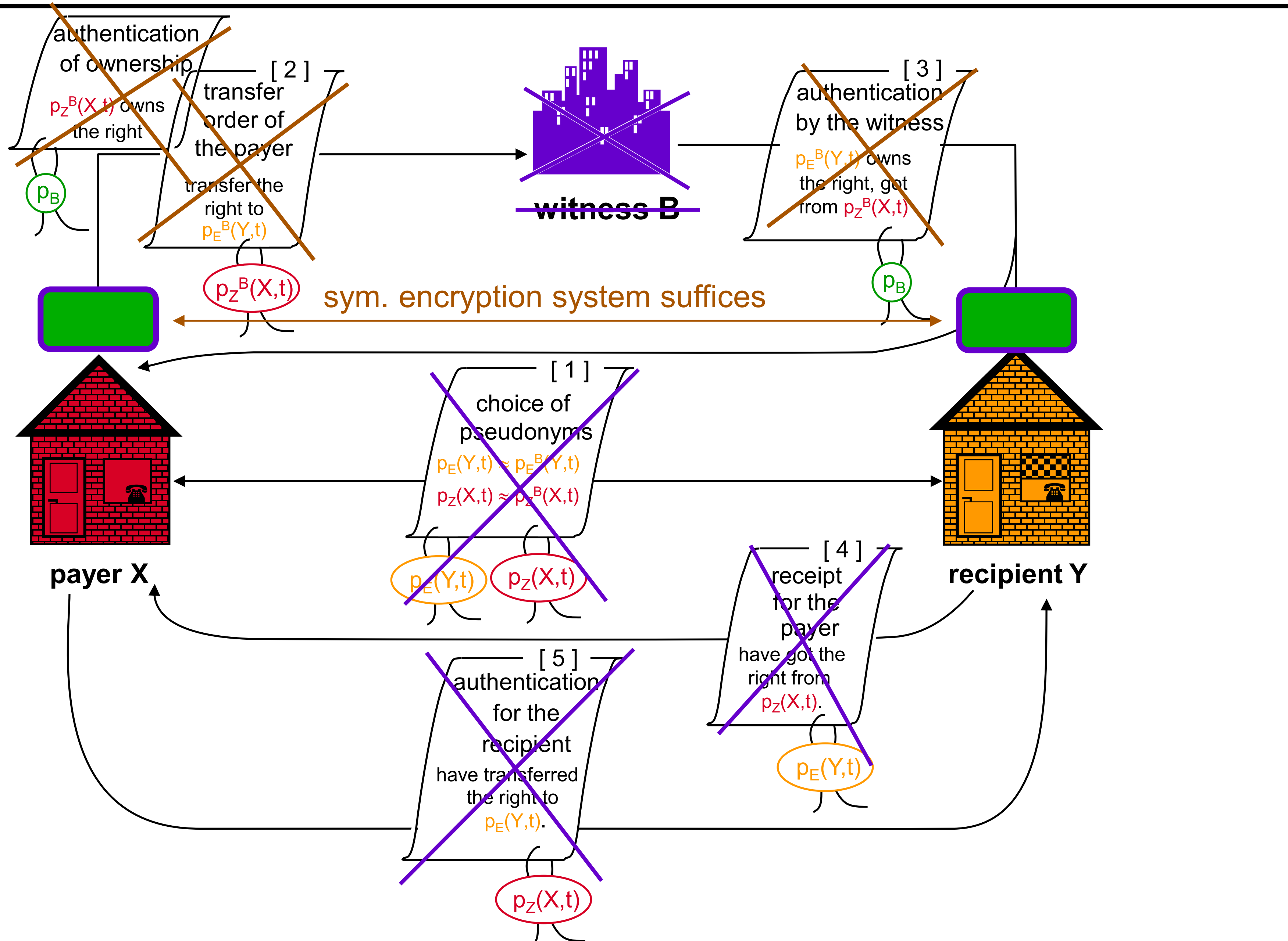
Issuer (i.e. witness)

RSA test key t, n, publicly known

Secure device: 1st possibility



Secure device: 2nd possibility



Offline payment system

Payment systems with security by Deanonymizability

k	security parameter
I	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:

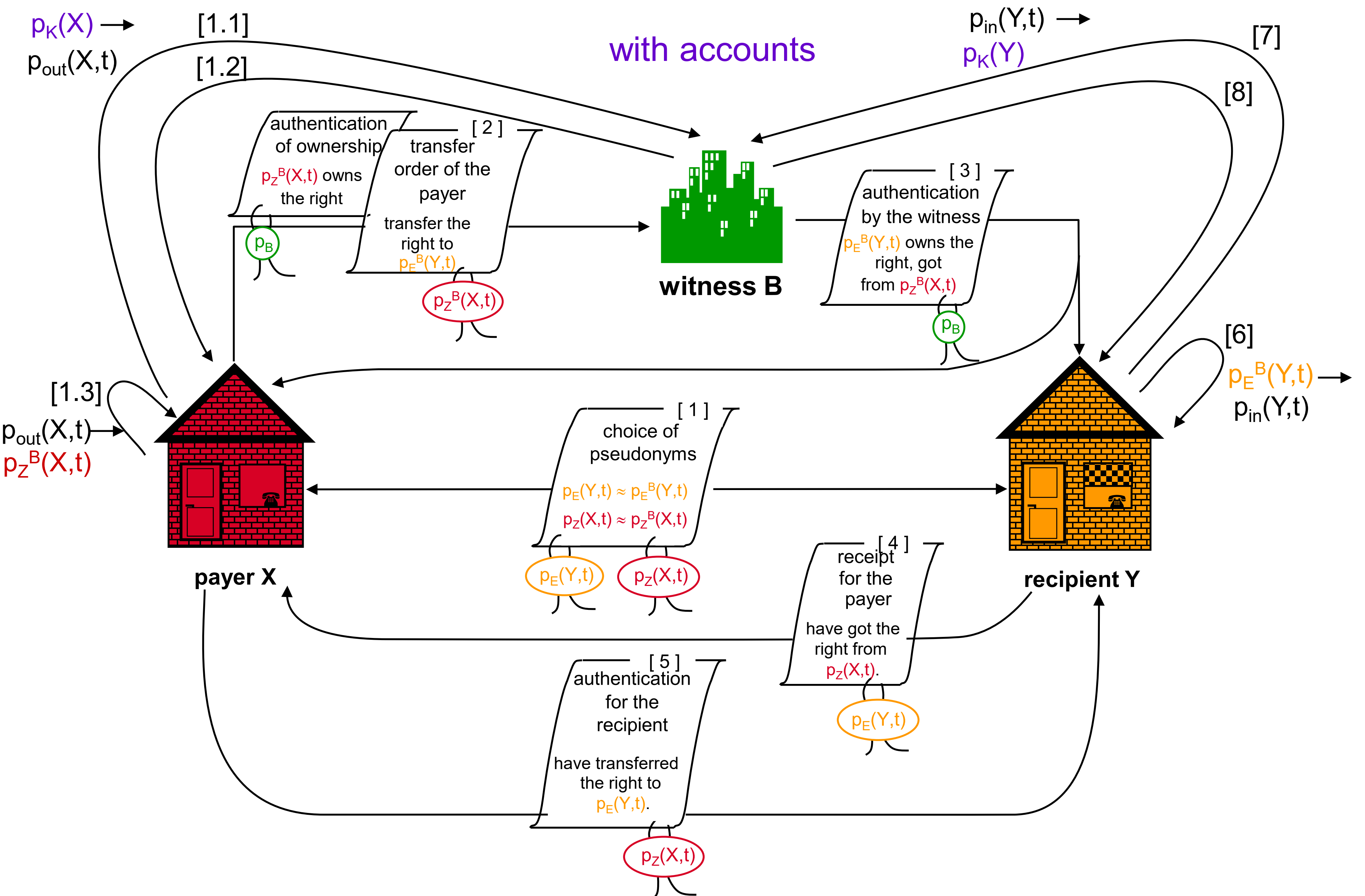
$$s_{Bank}(C(r_1), C(r_1 \oplus I), C(r_2), C(r_2 \oplus I), \ldots, C(r_k), C(r_k \oplus I)),$$

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

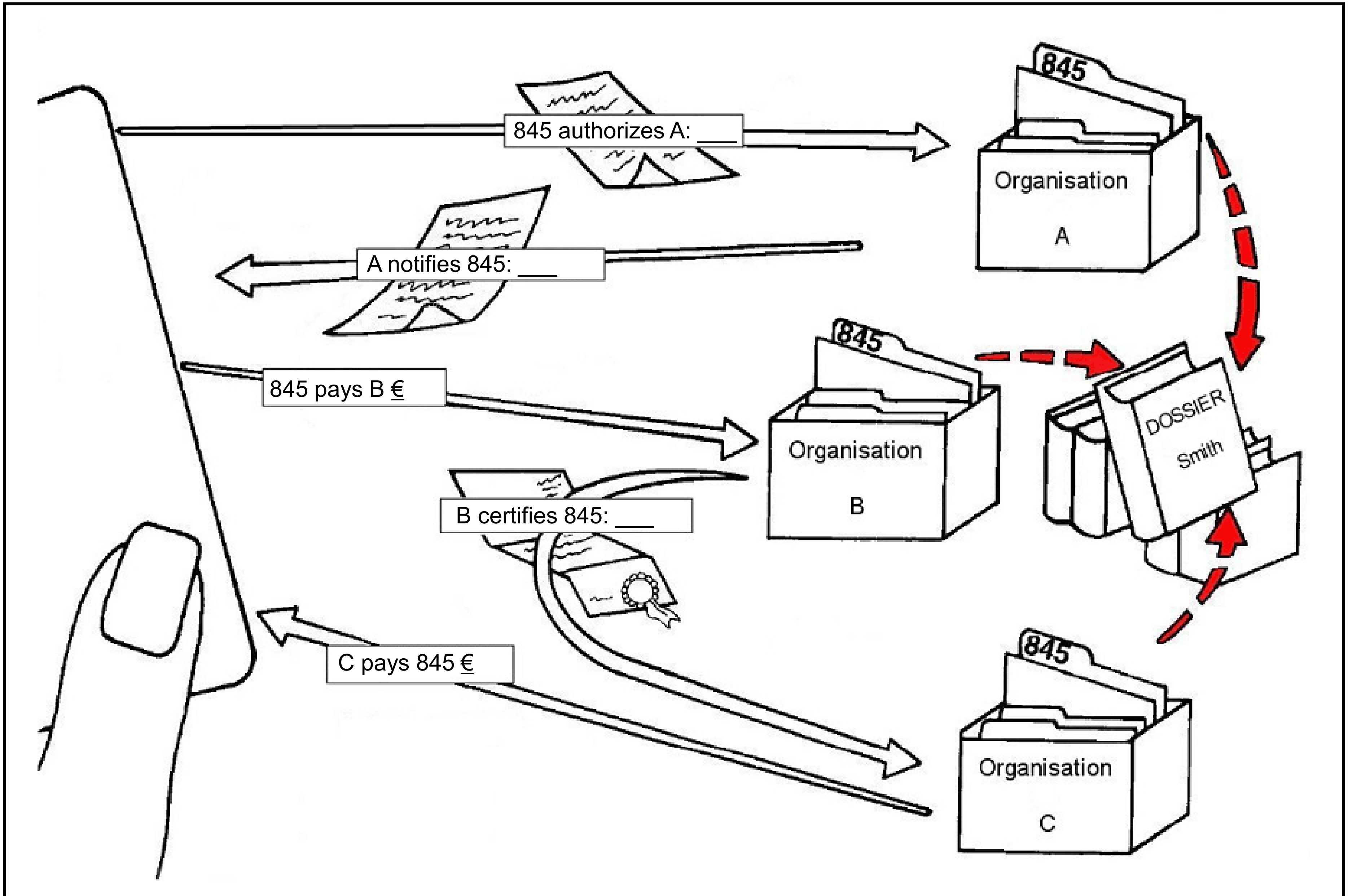
Hand-over to two honest recipients:
probability ($\exists i$: bank gets to know r_i and $r_i \oplus i$) $\geq 1\text{-}e^{-c \cdot k}$

(original owner identifiable)

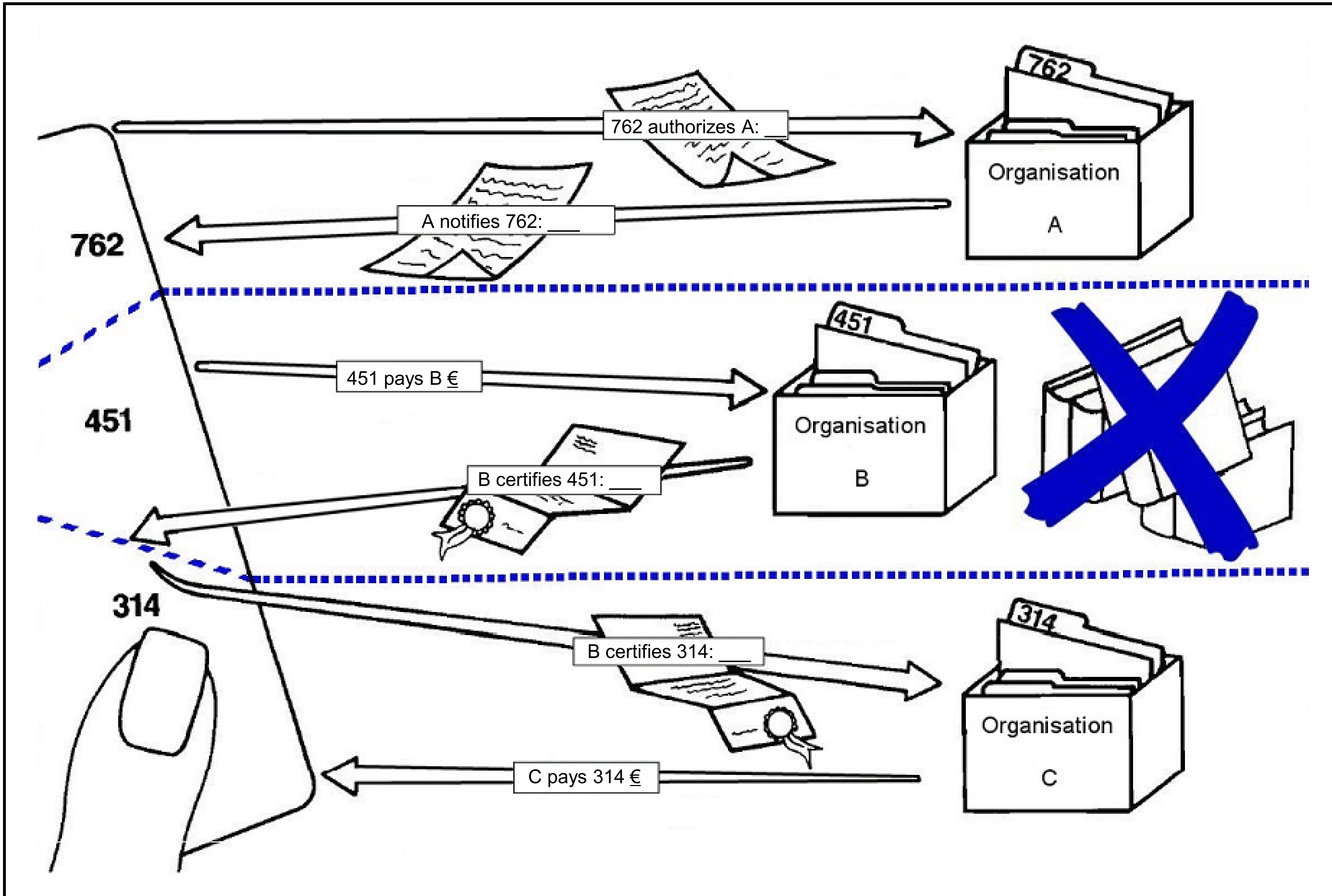
Secure and anonymous digit. payment system with accounts



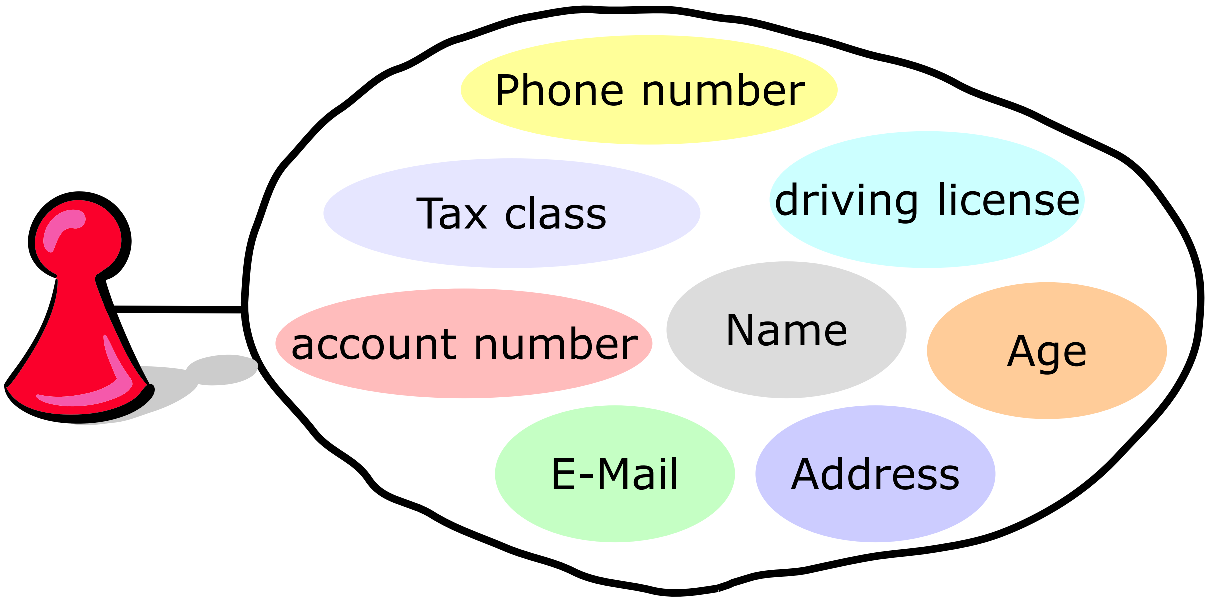
Personal identifier



Role pseudonyms (business-relationship and transaction pseudonyms)

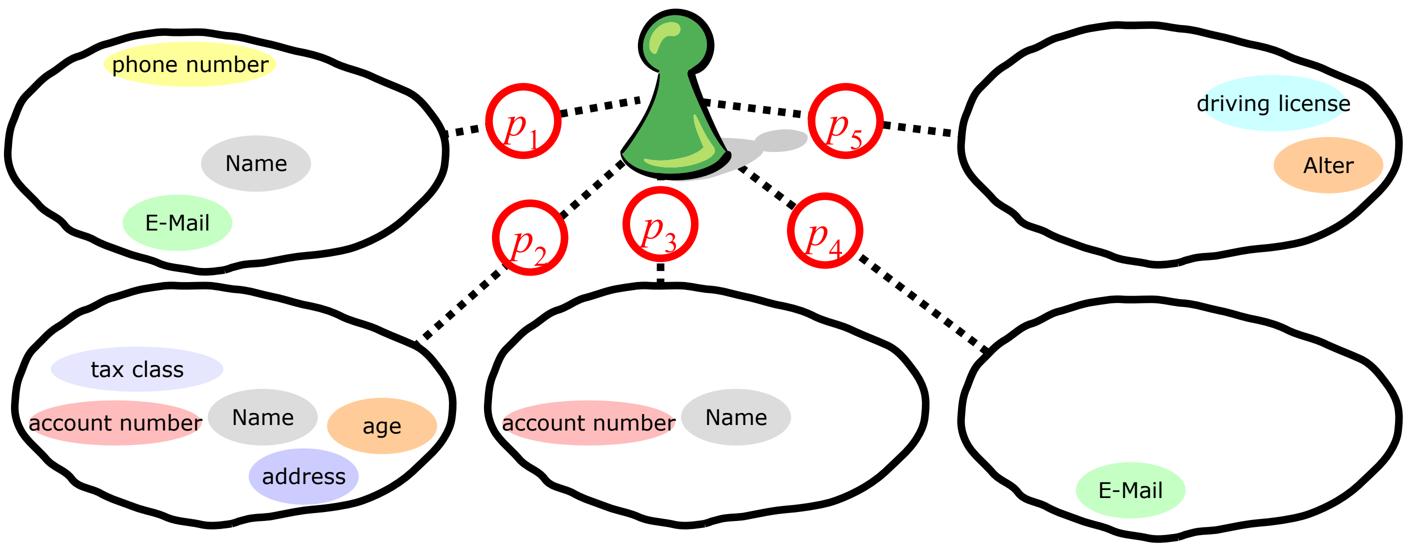


- Usually: one identity per user



Problem: Linkability of records

⌘ Many Partial-Identities per user



→ **Management / disclosure / linkability under the control of the user**

Implementation: Pseudonyms

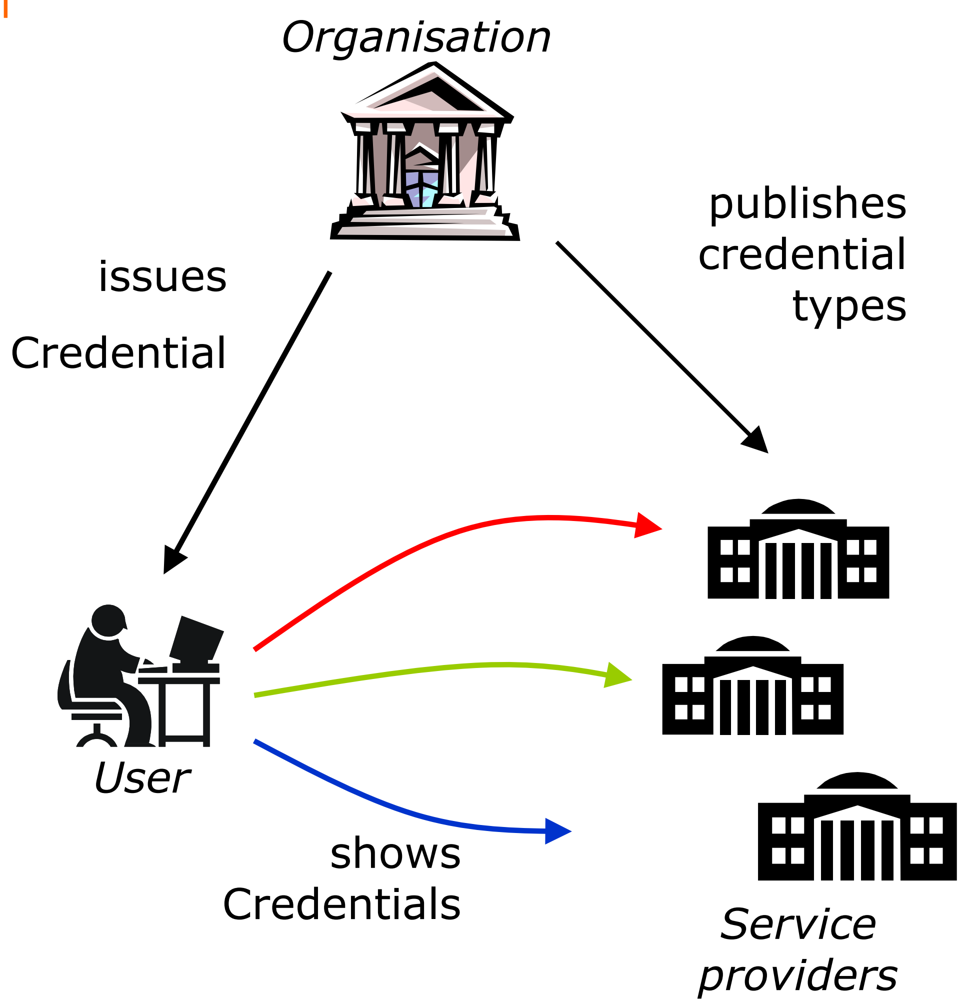
- many services need only a **few data**
- revealing that data under a **Pseudonym** prevents unnecessary linkability with other data of the user
- **different actions / data** are initially unlinkable if one uses different pseudonyms

Example: Car Rental

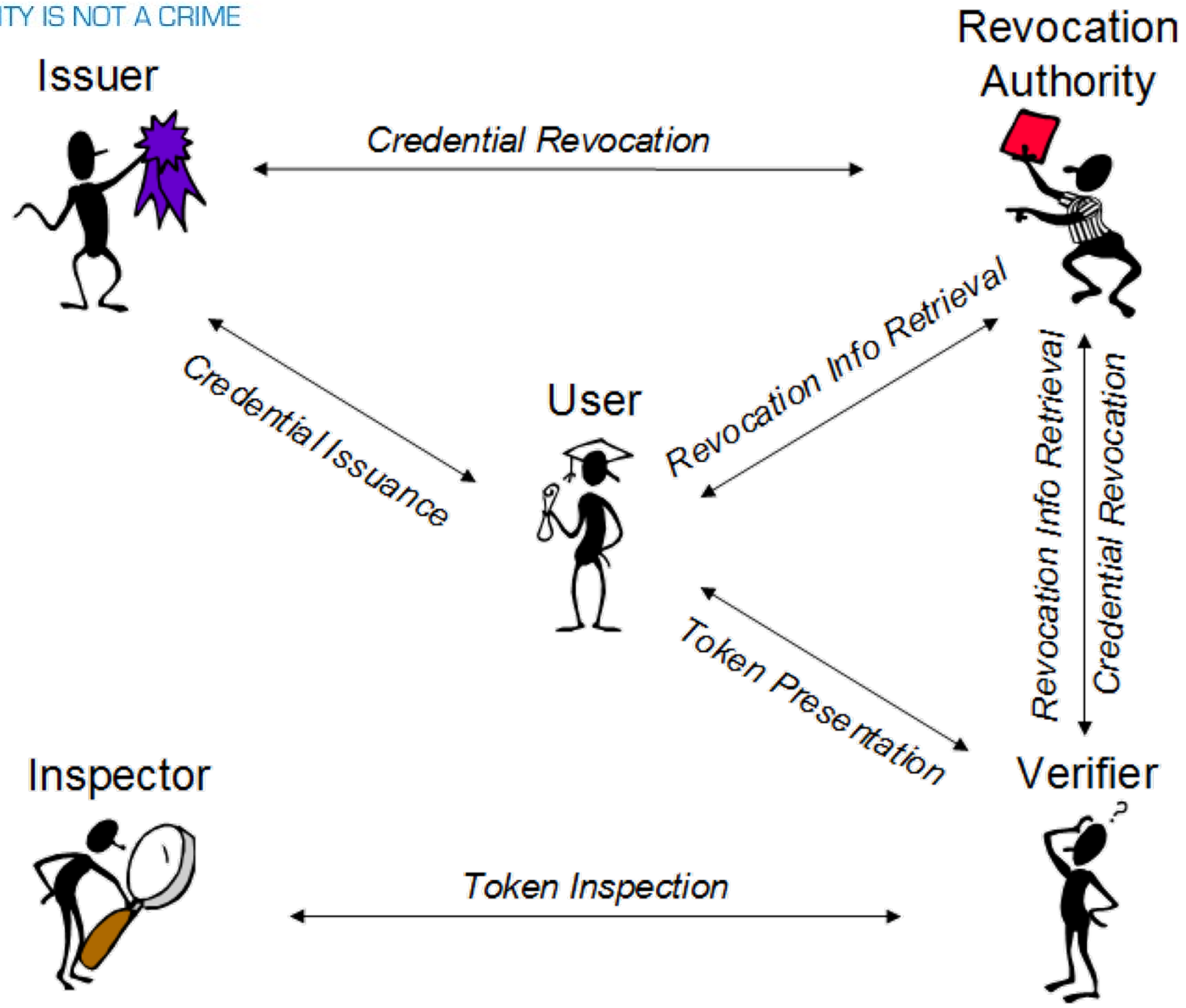
necessary data:
• Possession of a driving license valid for the car wanted

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.



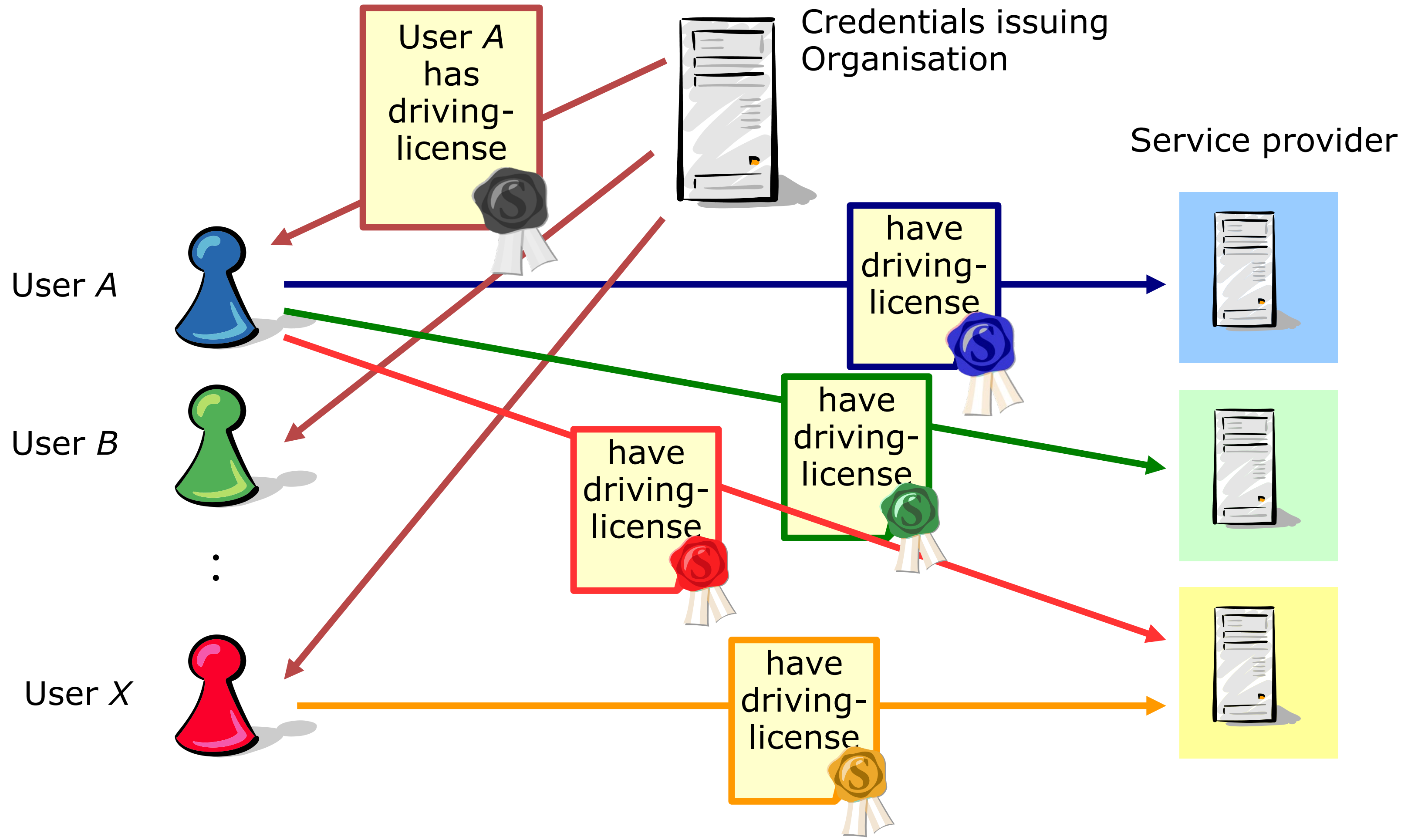
Anonymous Credentials More complete View



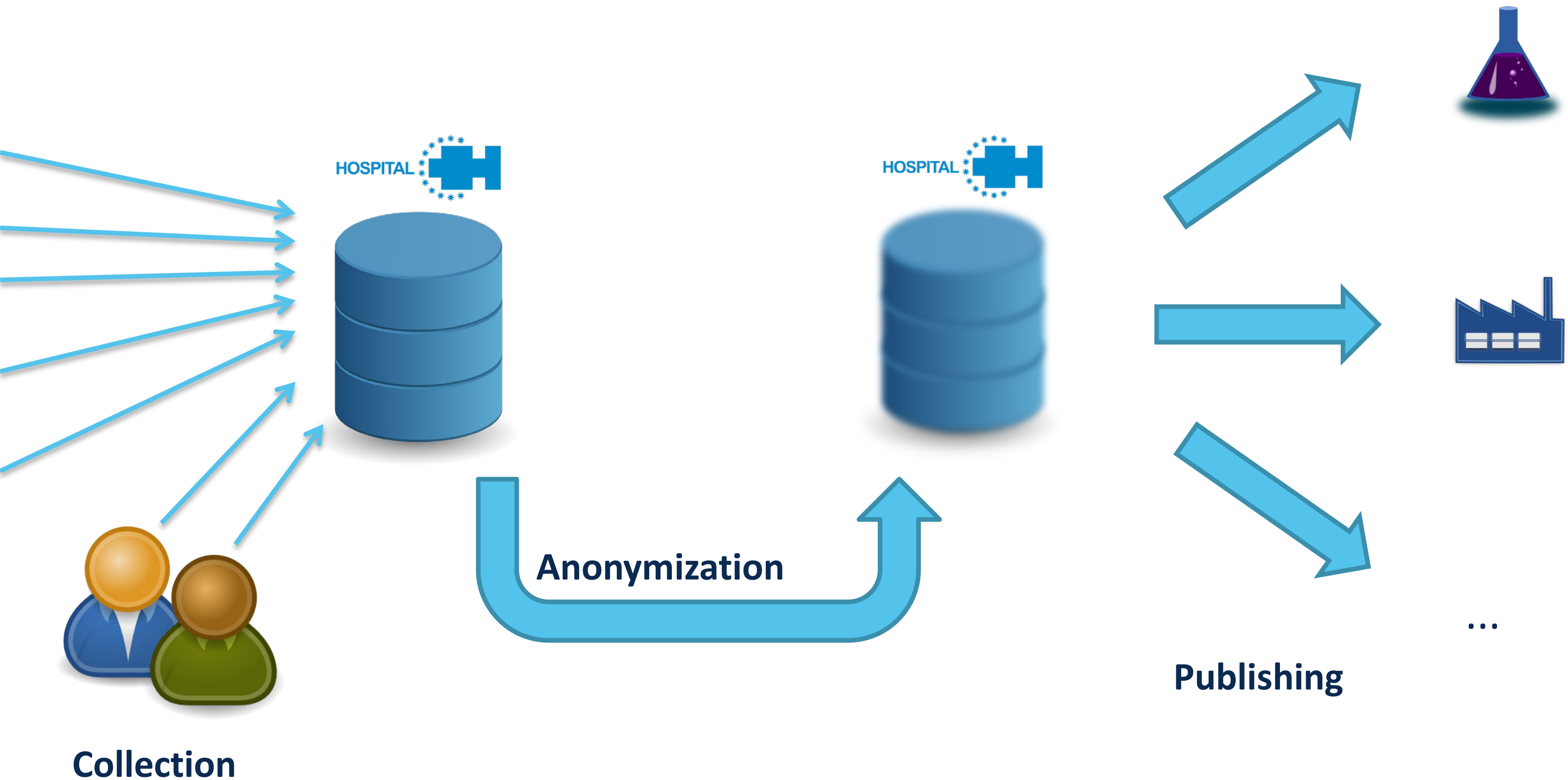
- Inspector can deanonymise

- Taken from EU project ABC4Trust [https://abc4trust.eu/download/Deliverable_D2.2.pdf]

Usage of Anonymous Credentials



Data Publishing – Use-Case

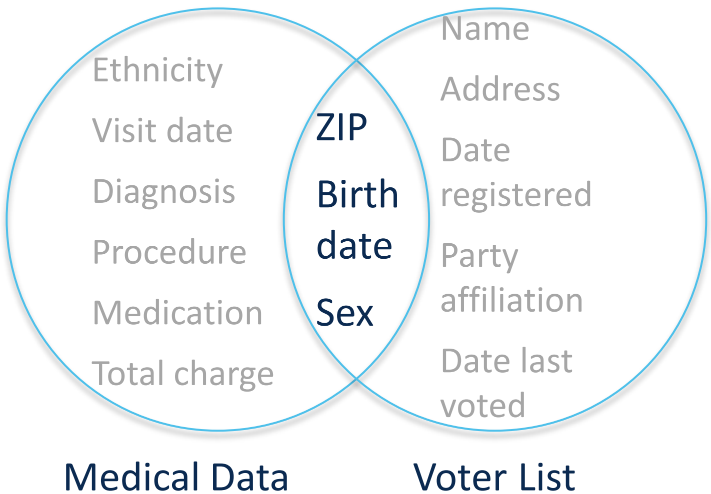


Data Publishing – Classification of Data

	Quasi ID			Sensitive		Non-sensitive	
	ZIP	Age	Sex	Disease	Salary	Q1	Q2
	47677	43	Male	Heart	3.000	a1	13
	47602	22	Female	Flu	5.000	a5	4
	47678	45	Female	Hepatitis	6.000	a4	22
	47905	31	Male	HIV	4.000	a1	12
	47909	36	Male	Flu	10.000	a2	8

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

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)

L. Sweeney: *k-anonymity: a model for protecting privacy*, Int. J. Uncertain. Fuzziness Knowl.-Based Syst., October 2002

Data Publishing – Classification of Data

	Quasi ID			Sensitive		Non-sensitive	
	ZIP	Age	Sex	Disease	Salary	Q1	Q2
	47677	43	Male	Heart	3.000	a1	13
	47602	22	Female	Flu	5.000	a5	4
	47678	45	Female	Hepatitis	6.000	a4	22
	47905	31	Male	HIV	4.000	a1	12
	47909	36	Male	Flu	10.000	a2	8

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

Data Publishing – Anonymization (*k*-Anonymity)

	ZIP Code	Age	Disease
1	47677	29	Heart Disease
2	47602	22	Heart Disease
3	47678	27	Heart Disease
4	47905	43	Flu
5	47909	52	Heart Disease
6	47906	47	Cancer

k=3 →

	ZIP Code	Age	Disease
1	476**	2*	Heart Disease
2	476**	2*	Heart Disease
3	476**	2*	Heart Disease
4	4790*	≥40	Flu
5	4790*	≥40	Heart Disease
6	4790*	≥40	Cancer

- 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** → at least *l* different values for each sensitive attribute in each equivalence class
  - **Problem:** meaning of “different”: different kinds of cancer → cancer
    – Solution: ***t*-closeness**

Semantic Security

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

Privacy Equivalent to Semantic Security

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)

Absolut Privacy Problem

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 ϵ-differential privacy if for all data sets D_1 and D_2 differing on at most one element, and all $S \subseteq Image(K)$,

$$Pr[K(D_1) \in S] \leq e^{\epsilon} \cdot Pr[K(D_2) \in S]$$

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 ϵ the output of $K(\,)$ can vary a lot
- For small ϵ the output of $K(\,)$ can only vary slightly

Small ϵ:

- Higher privacy, lower utility

Large ϵ:

- Lower privacy, higher utility

Differential Privacy – Context

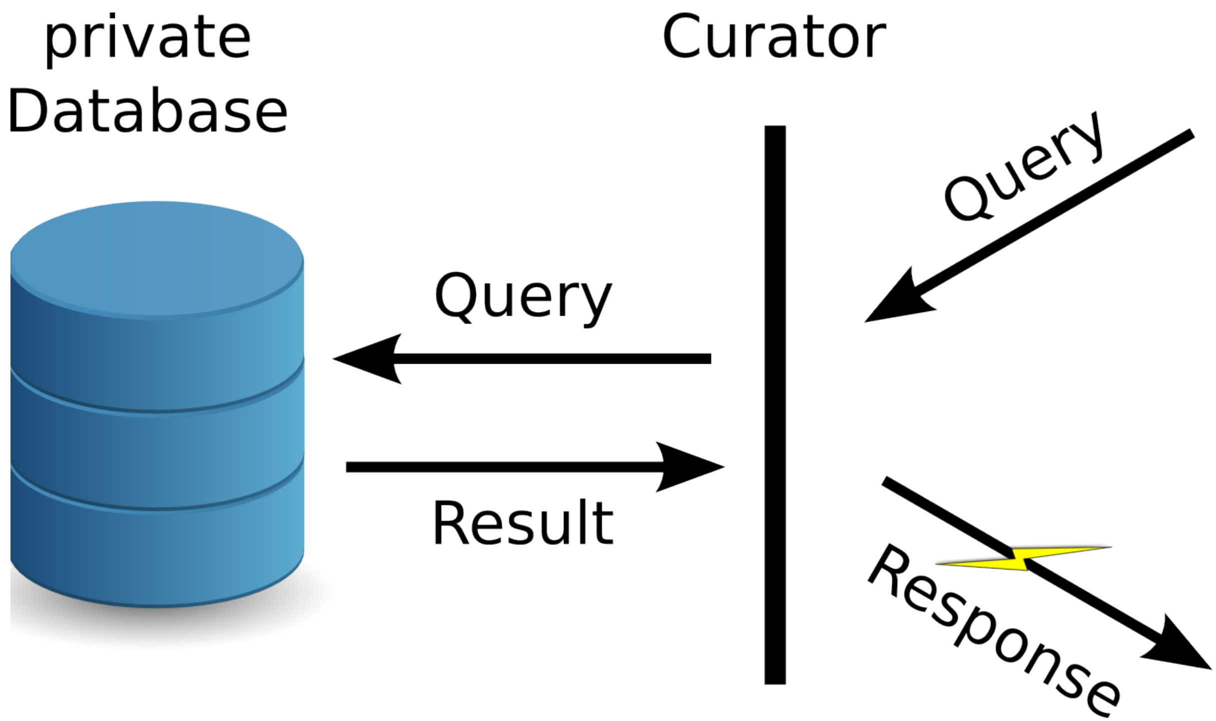
$$Pr[K(D_1) \in S] \leq exp(\epsilon) \cdot Pr[K(D_2) \in S]$$

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:

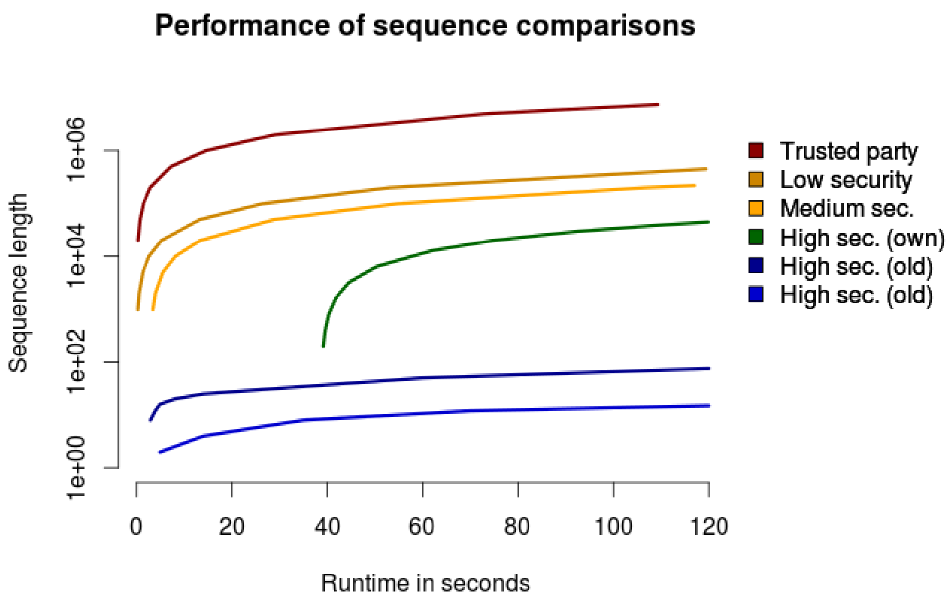
- Perturbed Histogram
- In general: statistics about database

Typical approach:

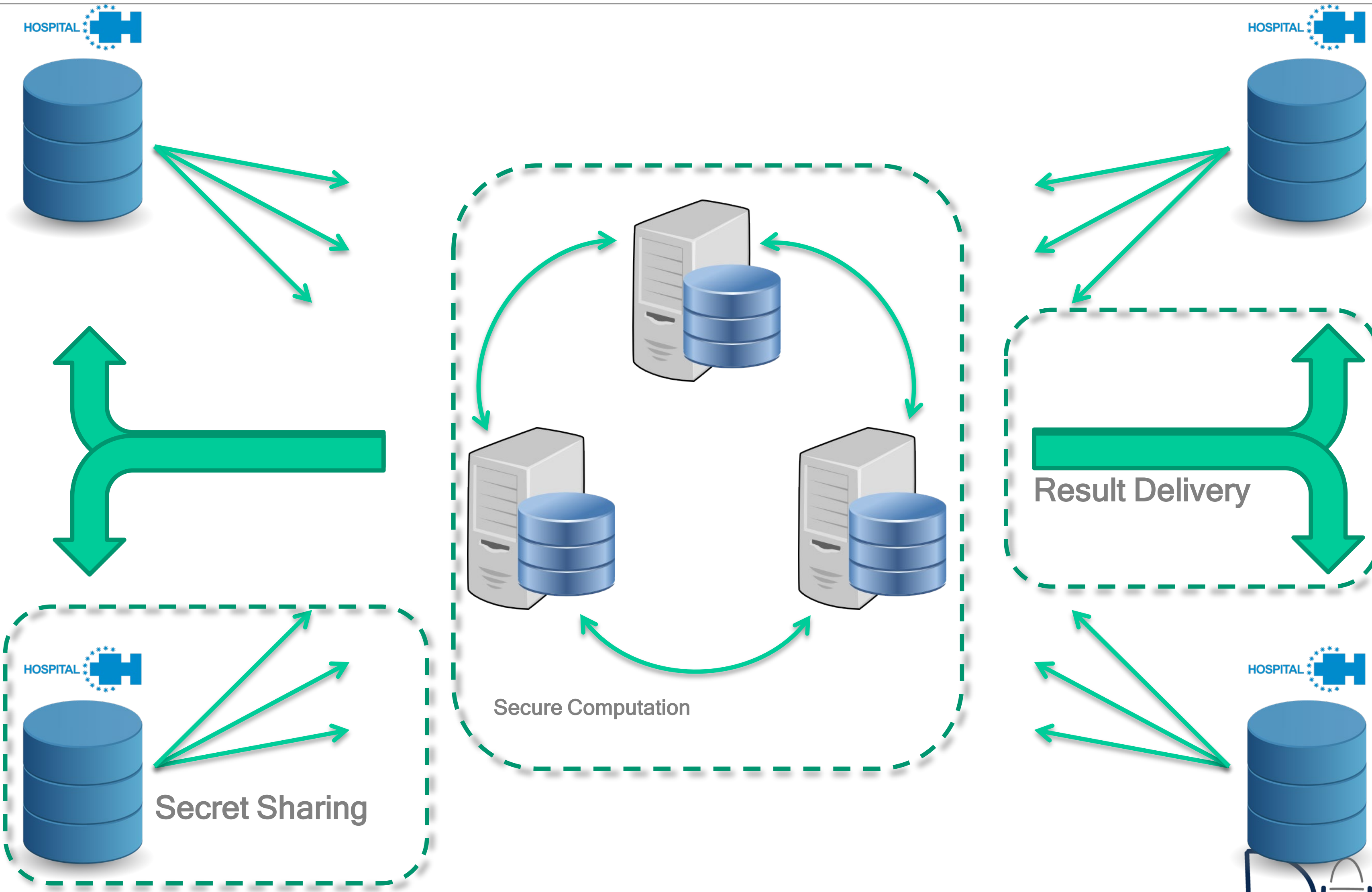
Calculate statistic then add noise.

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

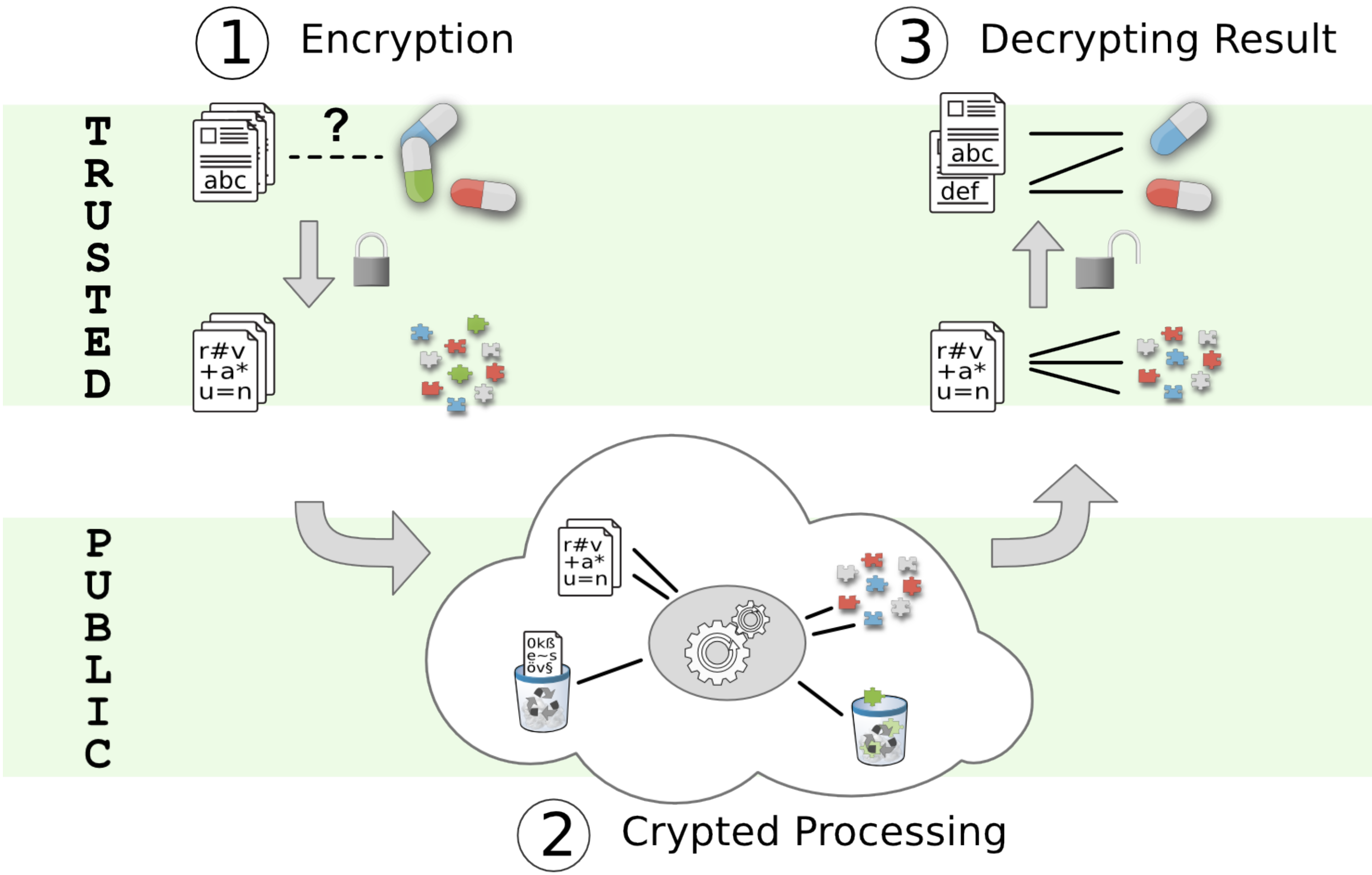
- **Secure string matching**
  - sequence comparisons
  - similarity between strings
  - fuzzy text search
  - basis for text mining

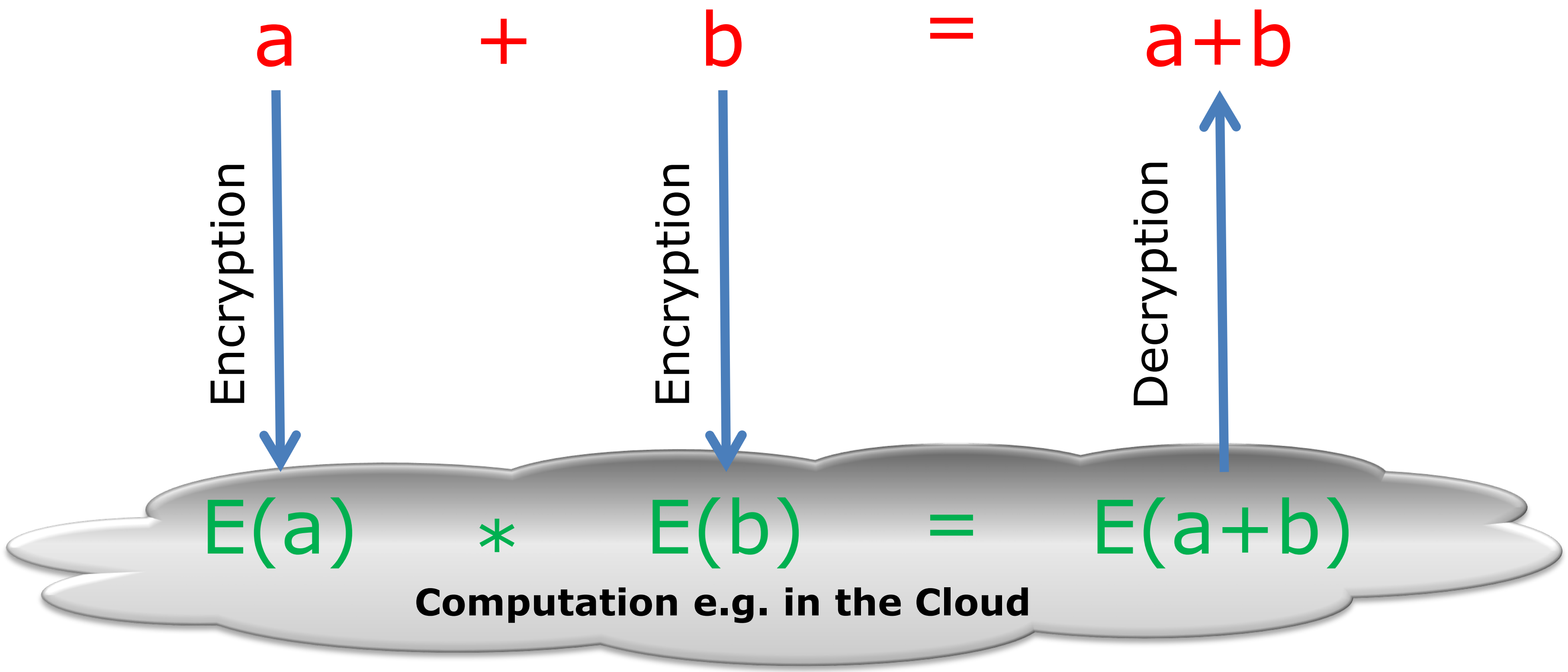


Secure Multi-Party Computations



Homomorphic Encryption



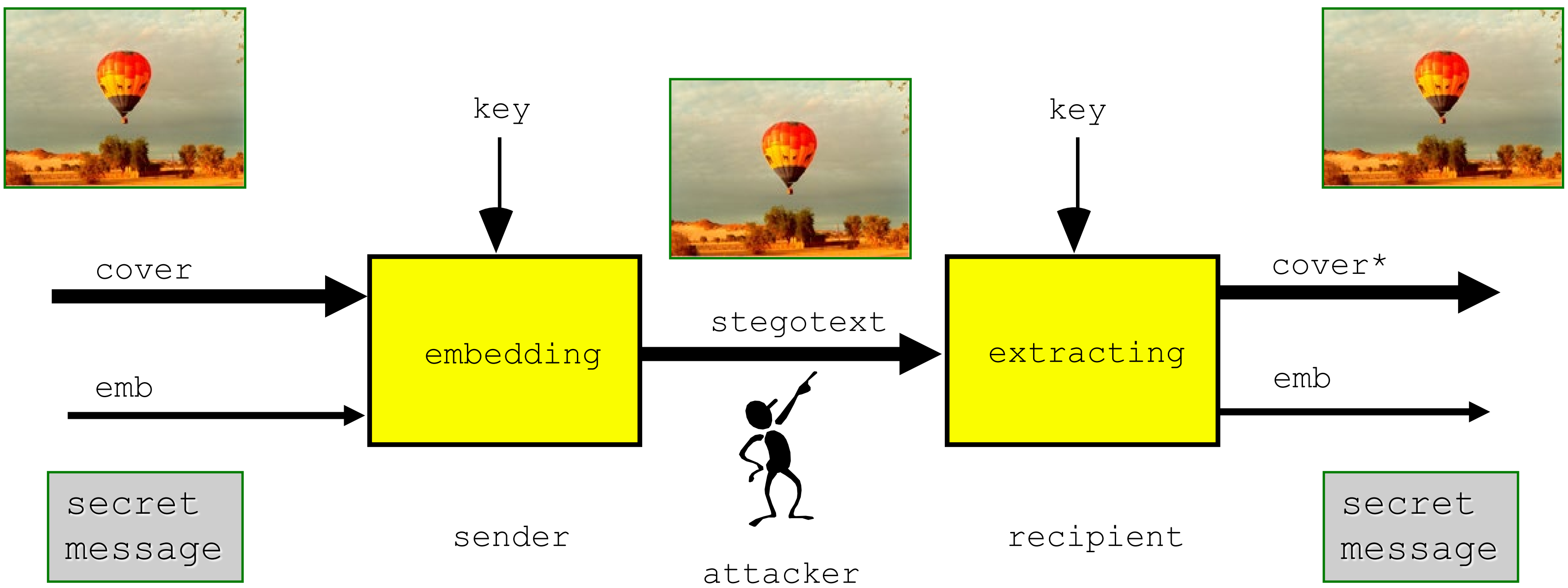


- Computation with secret inputs
  - inputs could be from different parties
- Based on the properties of a Homomorphism:
  - $f(a) \circ f(b) = f(a+b)$
- in principle: arbitrary „circuits" / algorithms computable
  - huge overhead!

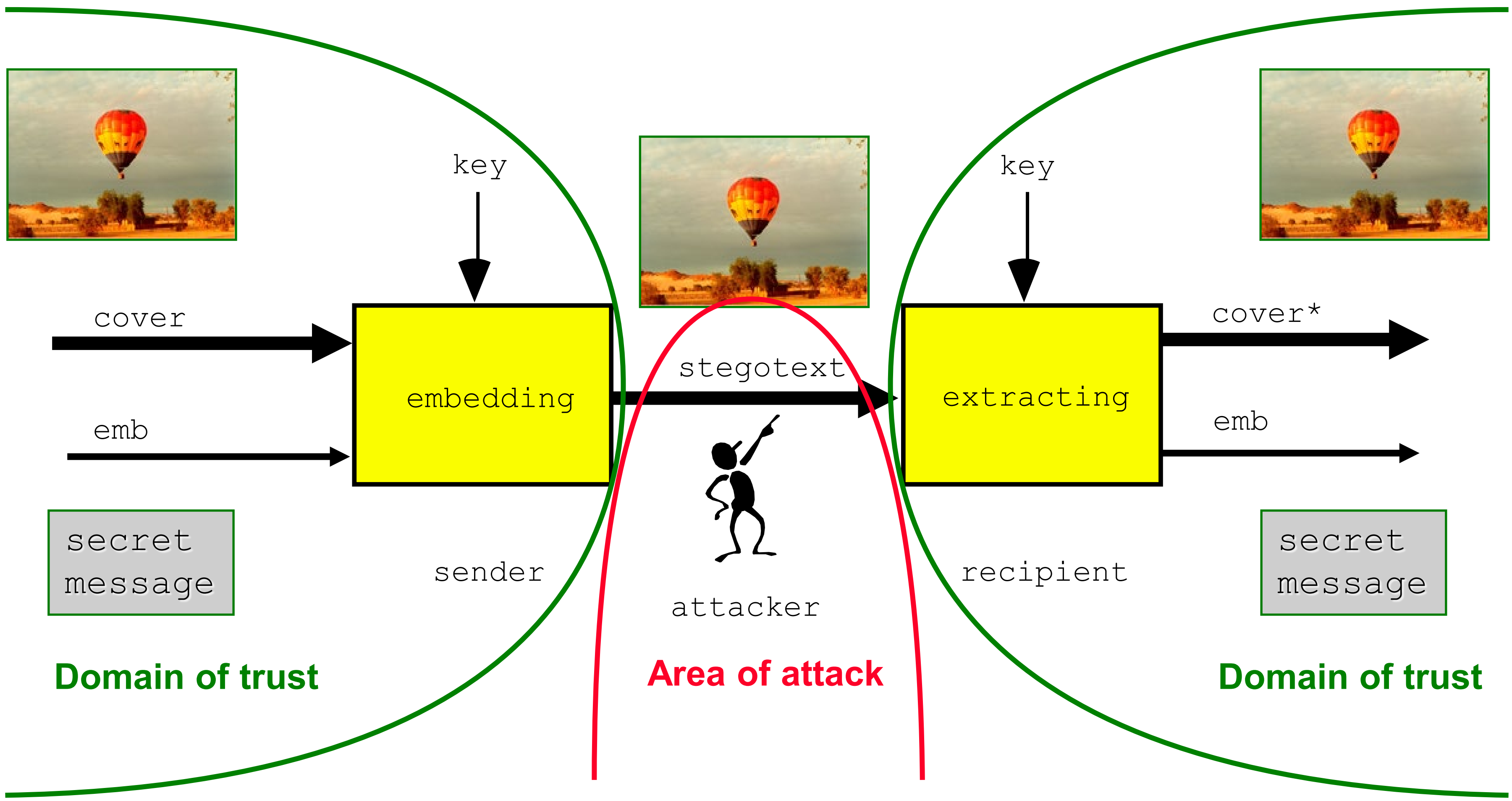
Cryptography and the impossibility of its legal regulation

- 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

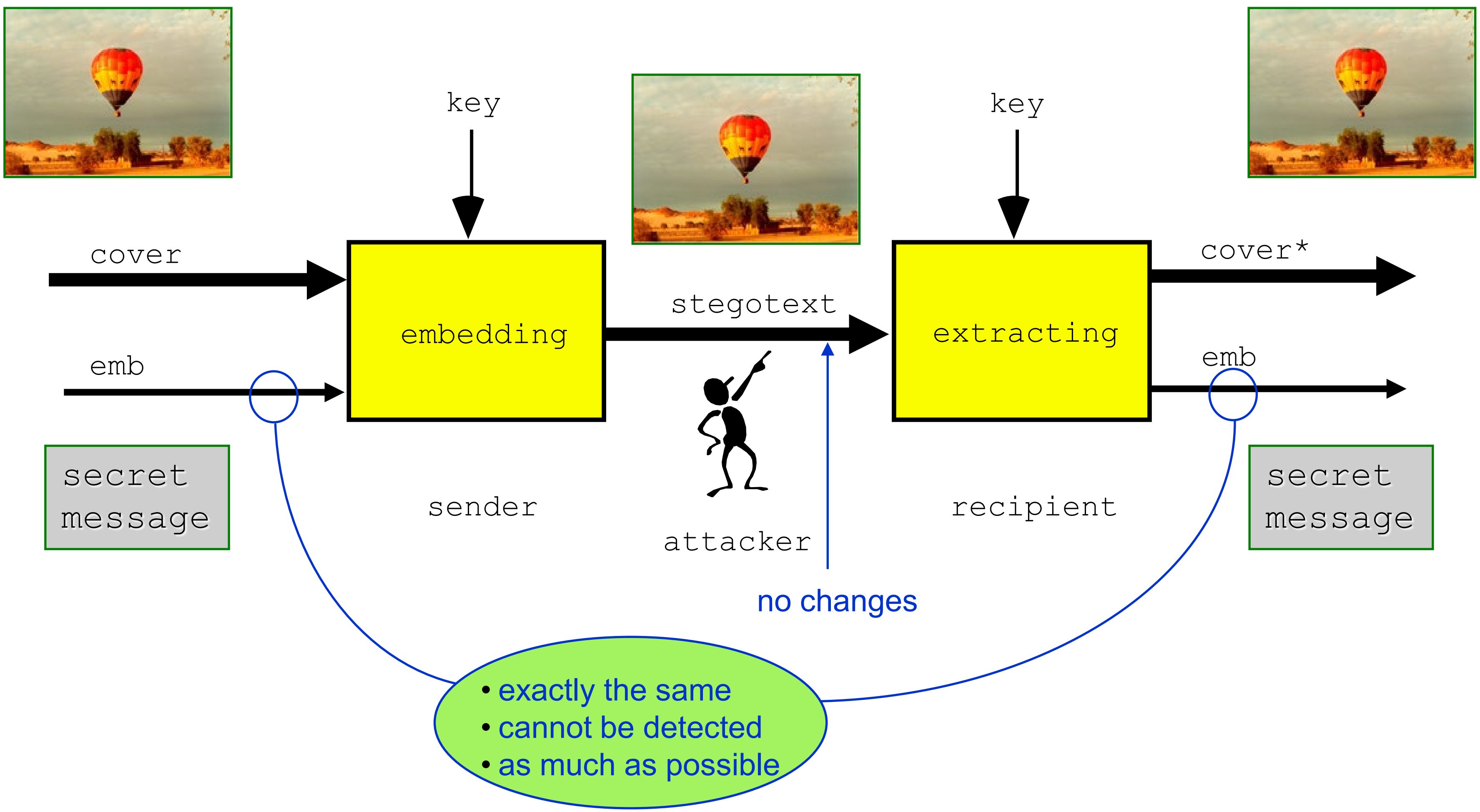
Steganography



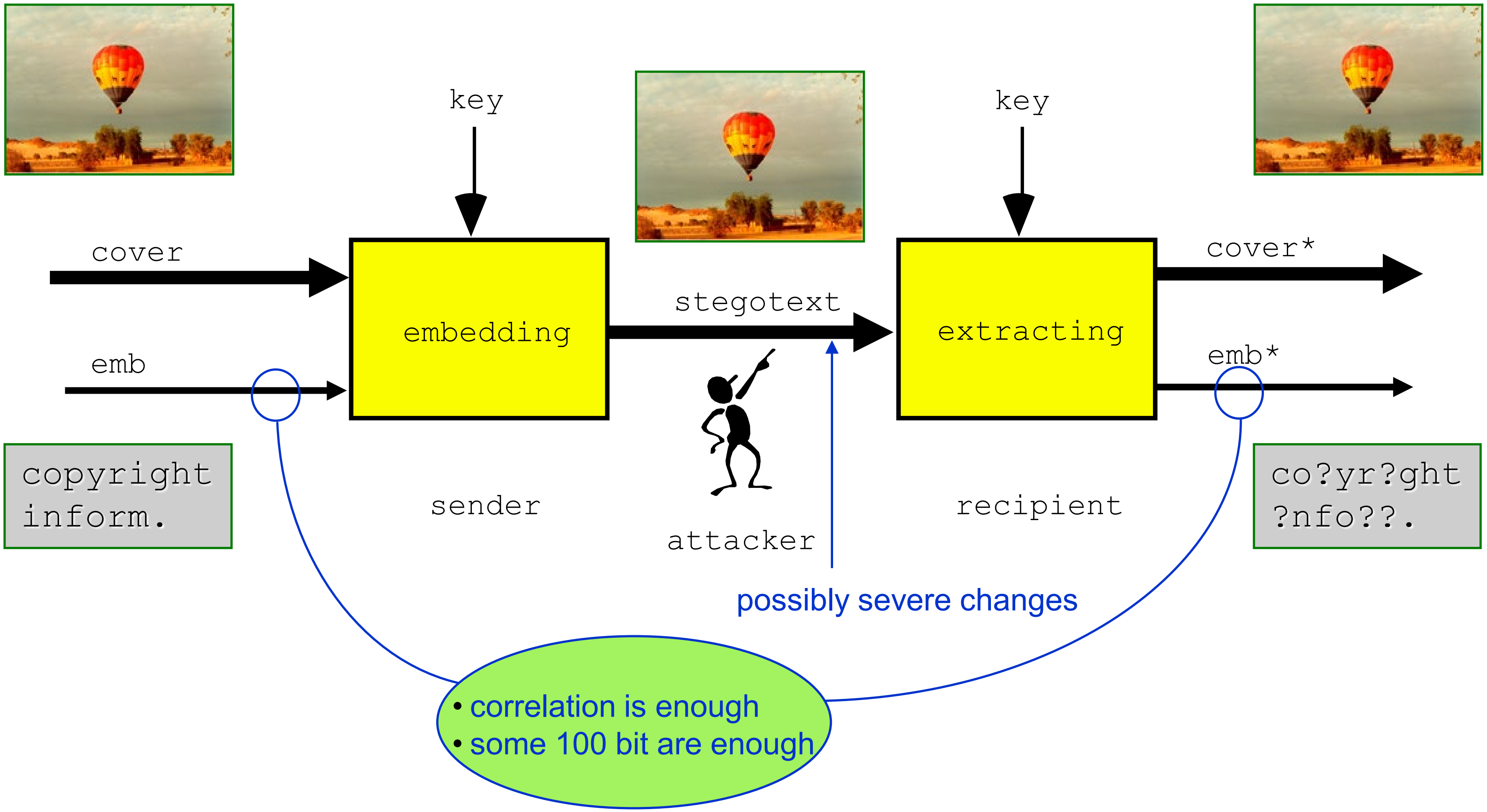
Steganography



Steganography: Secrecy of secrecy



Steganography: Watermarking and Fingerprinting

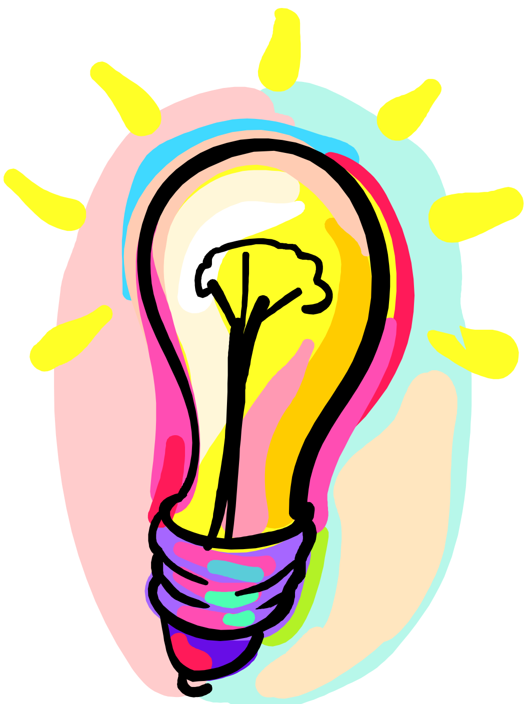


Proposals to regulate cryptography ?

?

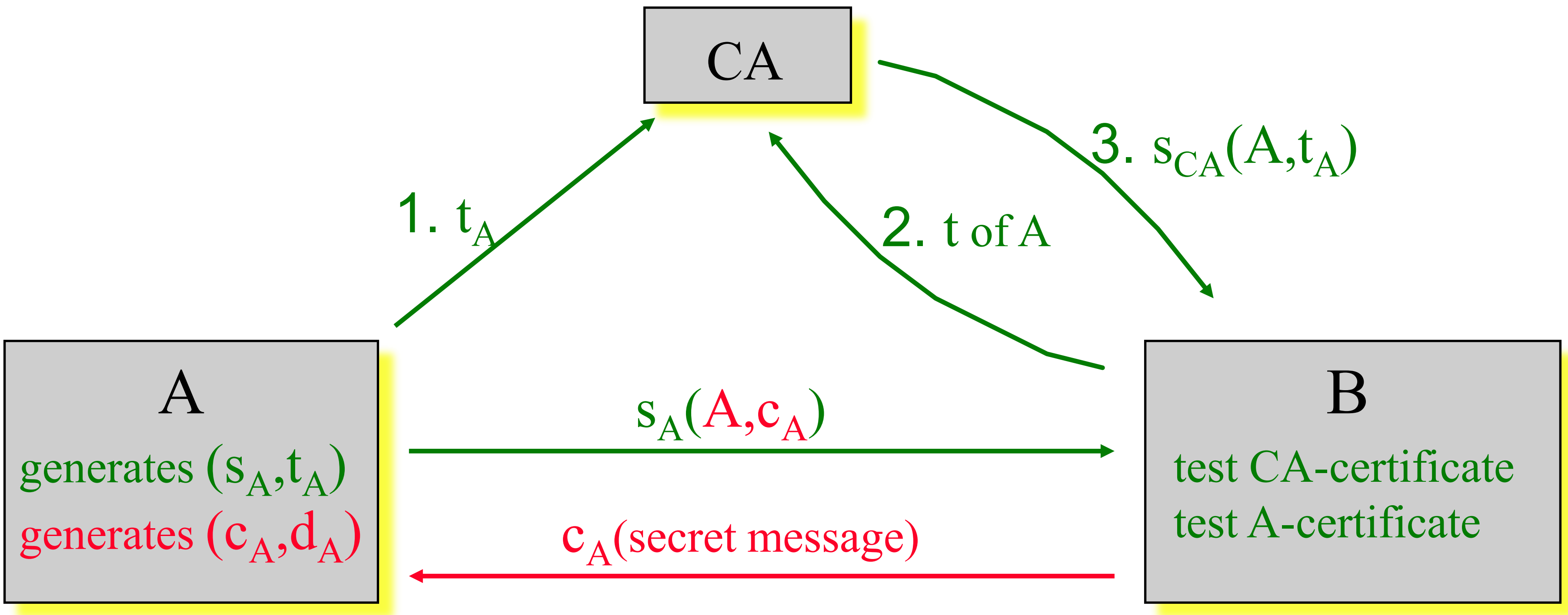
- Would you regulate cryptography to help fight crime ?
- If so: How ?

Proposals to regulate cryptography !



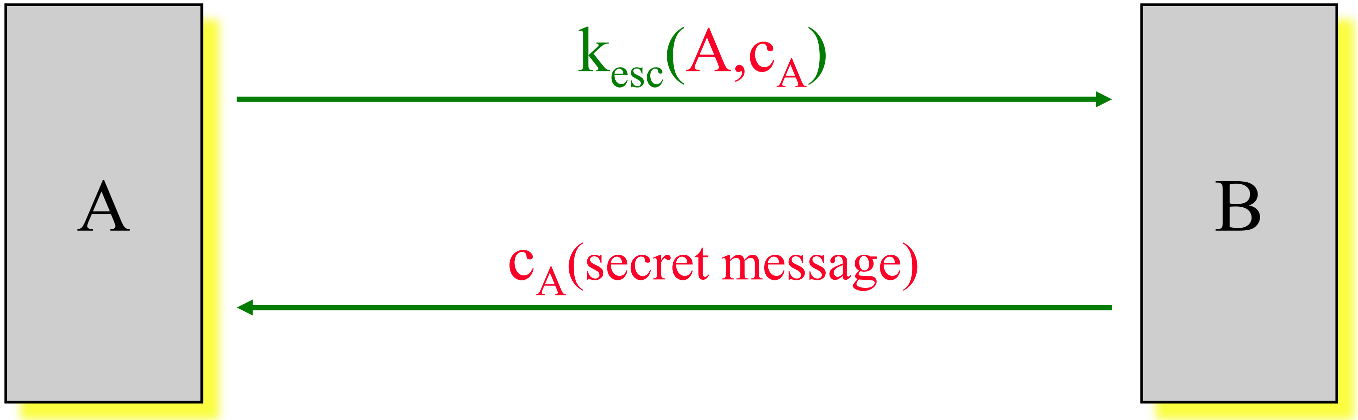
- Outlaw encryption
- Outlaw encryption – with the exception of small key lengths
- Outlaw encryption – with the exception of Key Escrow or Key Recovery systems
- Publish public encryption keys only within PKI if corresponding secret key is escrowed
- Obligation to hand over decryption key to law enforcement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

Secure digital signatures —> Secure encryption



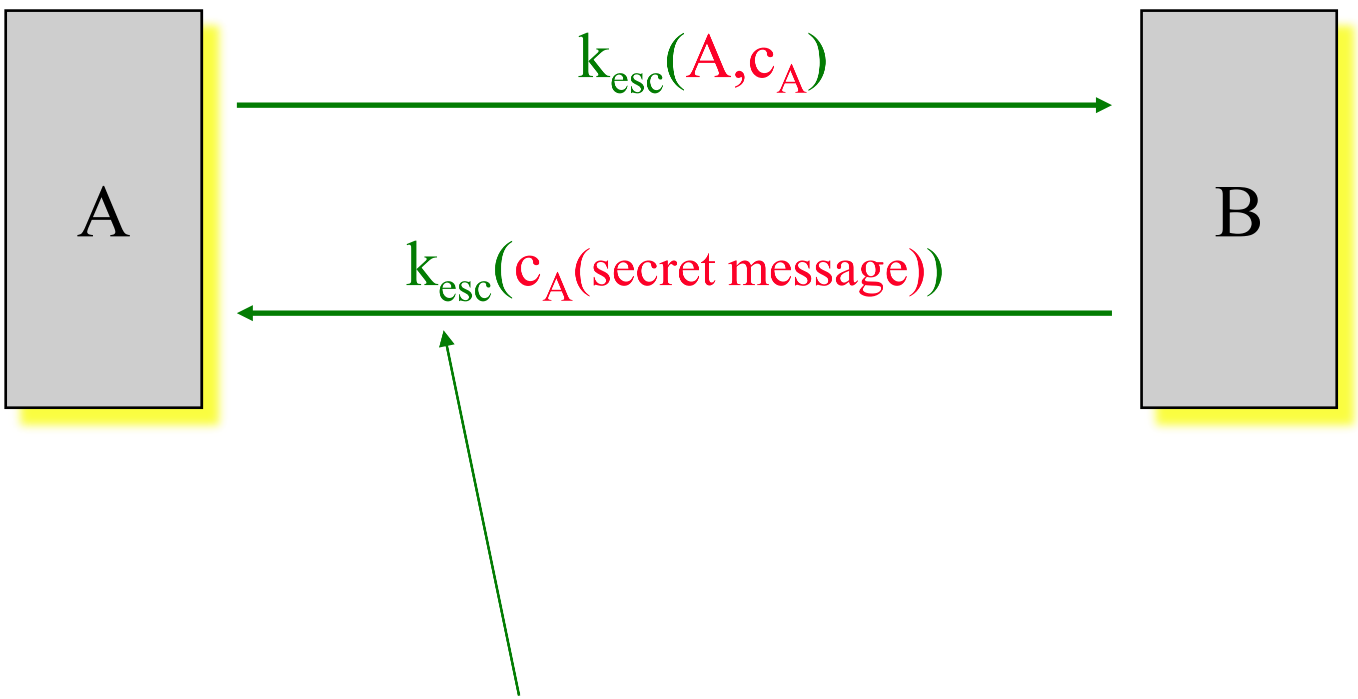
A does not need a certificate for c_A issues by CA

Key Escrow encryption without permanent surveillance



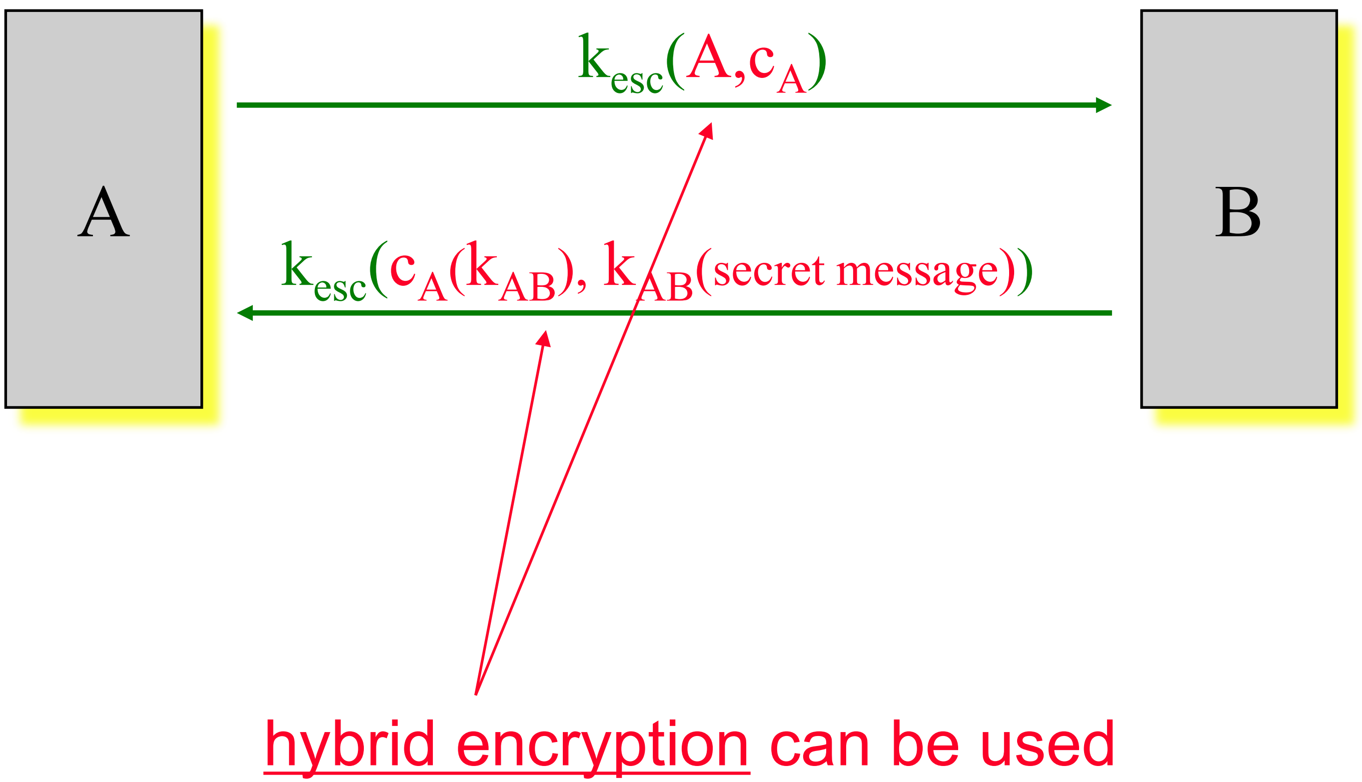
—> Encryption without Key Escrow

Key Escrow encryption without permanent surveillance

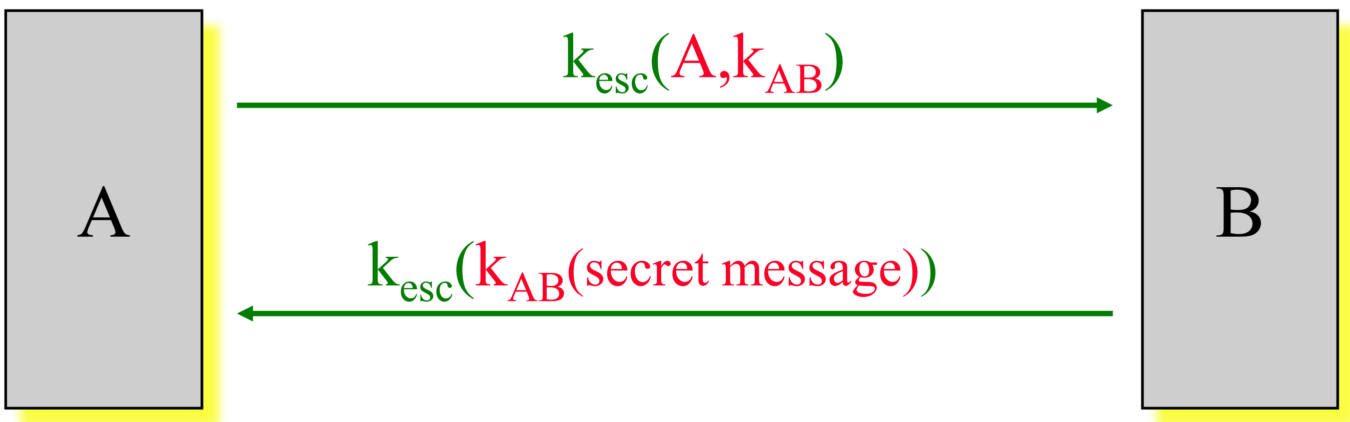


employ Key Escrow additionally
to keep your encryption without Key Escrow secret

Key Escrow encryption without permanent surveillance



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

$b_1, \dots b_n$ mit $b_i \in \{0, 1\}$

Berechnet

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

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

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

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

form

Überträgt (die Mengenklammern bedeuten „zufällige Reihenfolge")

$\{(b_1, MAC_1), (a_1, MAC'_1)\} \dots$

$\{(b_n, MAC_n), (a_n, MAC'_n)\}$ —>

intermingle

Empfänger *B*

Kennt k_{AB}

falsely authenticated messages

Probiert, ob

$\{MAC_1 = code(k_{AB}, b_1)$ oder

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

und empfängt den passenden Wert b_1

...

probiert, ob

$\{MAC_n = code(k_{AB}, b_n)$ oder

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

und empfängt den passenden Wert b_n

separate

Ronald L. Rivest: Chaffing and Winnowing: Confidentiality without Encryption; MIT Lab for Computer Science, March 22, 1998; http://theory.lcs.mit.edu/~rivest/chaffing.txt

Symmetric authentication → Encryption

Sender A

Kennt k_{AB}

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

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

Überträgt
$(1, b_1, MAC_1), \ldots (n, b_n, MAC_n)$

Empfänger B

Kennt k_{AB}

Komplementgenerierer

falsely authenticated messages

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

Bildet $a_1, \ldots a_n$, die bitweise invertierte Nachricht.
Wählt zufällig $MAC'_1 \ldots MAC'_n$ und mischt in den Nachrichtenstrom von Sender A an die passenden Stellen
$(1, a_1, MAC'_1), \ldots (n, a_n, MAC'_n)$

Überträgt die Mischung ——o——————>

form and intermingle without knowing the key

separate

normales Authentikationsprotokoll
Ignoriert Nachrichten mit falscher Seque
Ignoriert Nachrichten mit falscher Authe
gibt die übrigbleibenden weiter
empfangen wird mit größter Wahrschein
$b_1, \ldots b_n$

Abhörer
kann a_i und b_i nicht unterscheiden

Key exchange for steganography ?

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 !

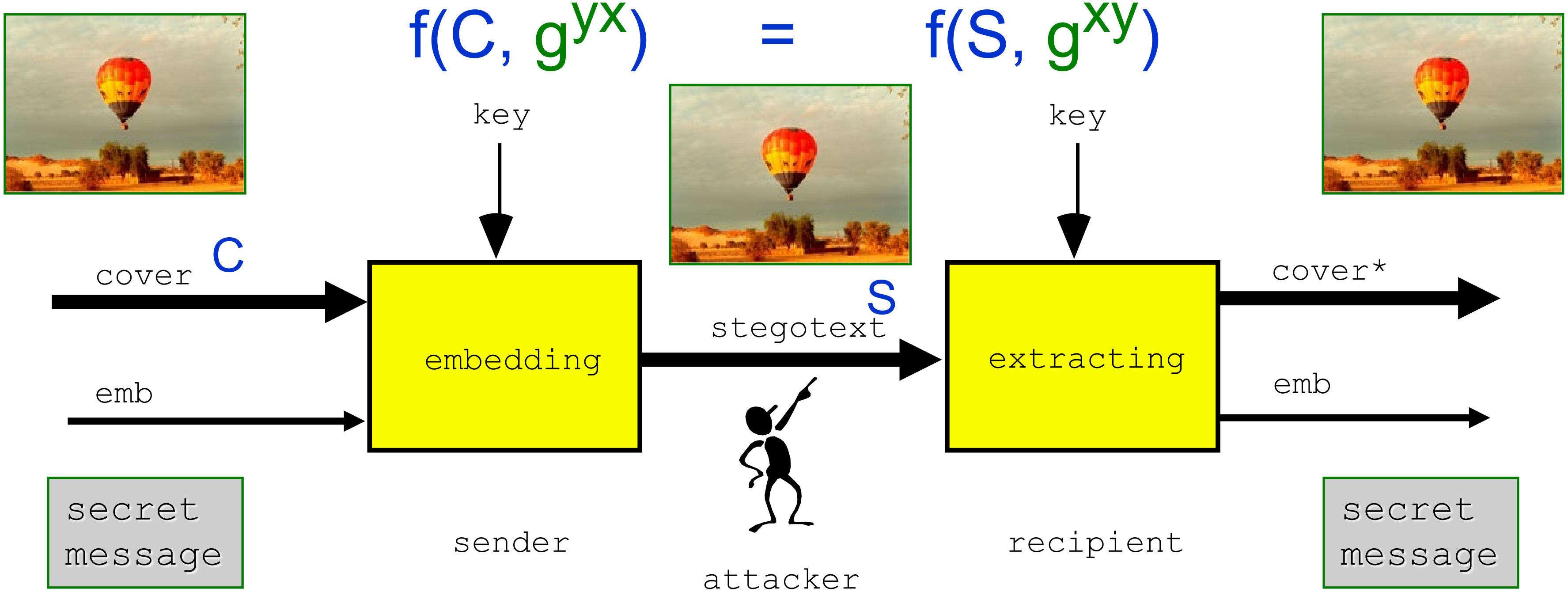
Diffie-Hellman Public-Key Agreement

secret: x y

public: g^x g^y

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

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

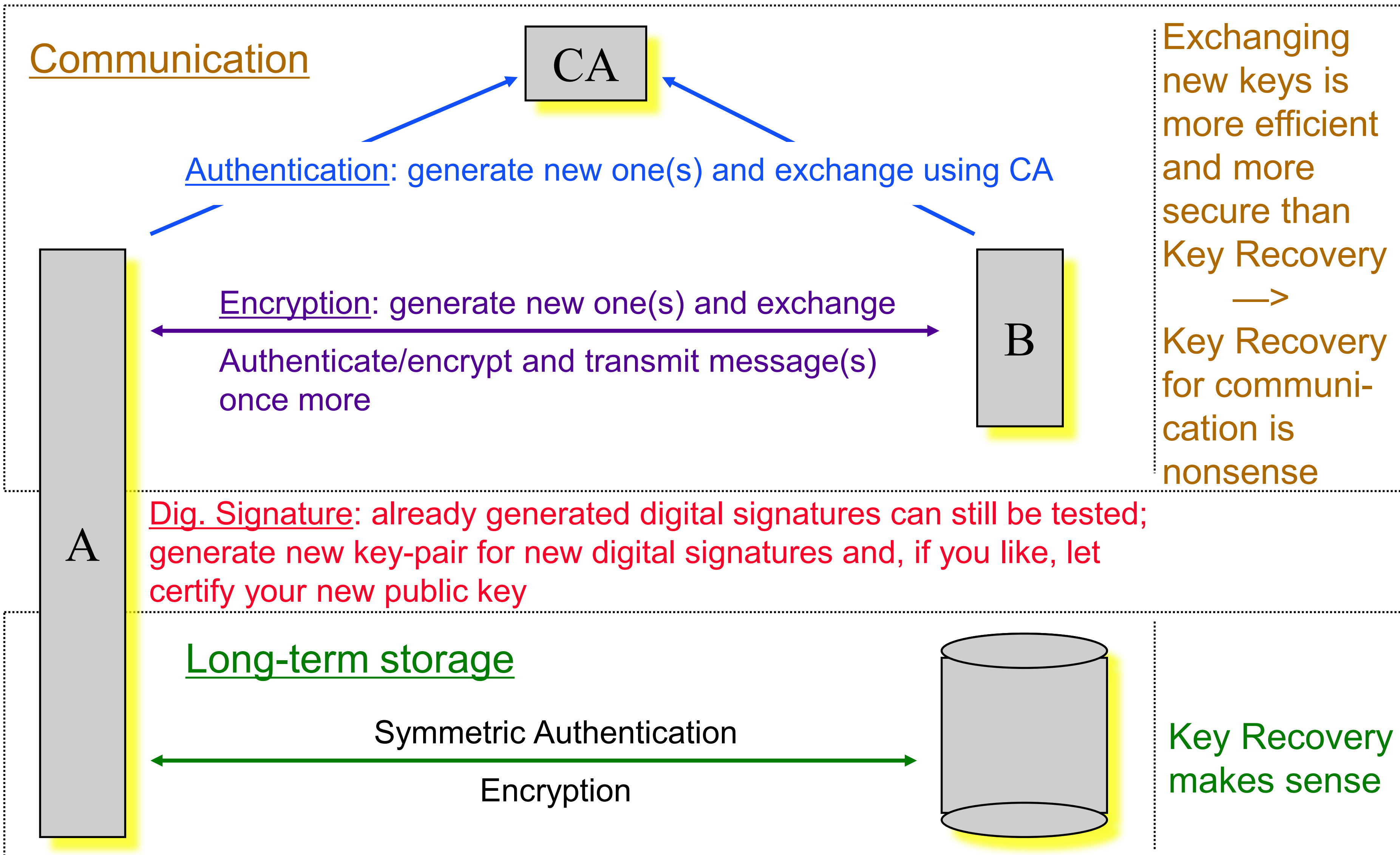


Summary

Digital Signatures	→	Encryption
Key Escrow without permanent surveillance	→	Key exchange, multiple encryption
Multimedia communication	→	Steganography

Cryptoregulation ignores technical constraints

Loosing secret keys



Key Recovery – for which keys ?

<table>
<tr><th colspan="2"></th><th colspan="2">protecting</th></tr>
<tr><th colspan="2"></th><th>communication</th><th>long-term storage</th></tr>
<tr><td colspan="2">Encryption</td><td rowspan="3">Key Recovery functionally unnecessary, but additional security risk</td><td>Key Recovery useful</td></tr>
<tr><td rowspan="2">Authen-tication</td><td>symmetric (MACs)</td><td></td></tr>
<tr><td>asymmetric (dig. signature)</td><td></td></tr>
</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-Proxies

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-presering 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

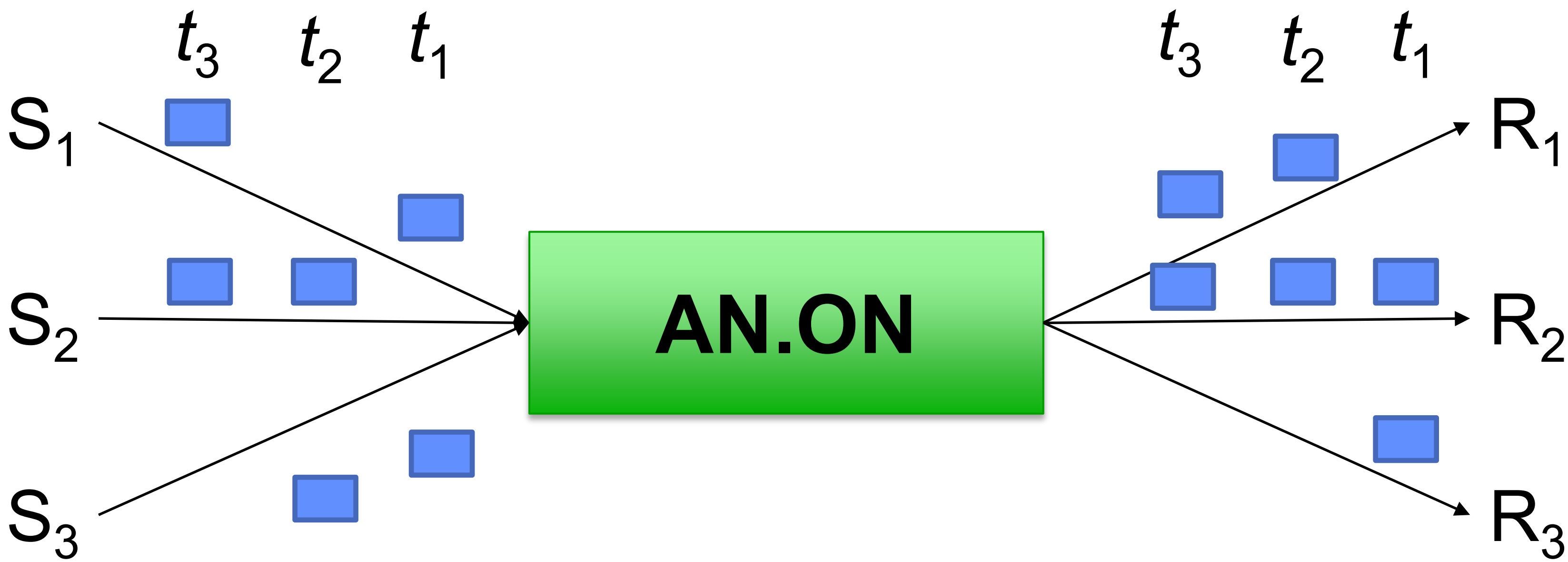
- Soundness and Completeness
  - valid signatures always verify correctly
  - invalid signatures always fail verification.
- Unforgeable
  - only group members can create valid signatures
- Anonymity
  - given a message and its signature, the signing group member cannot be determined without the group 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

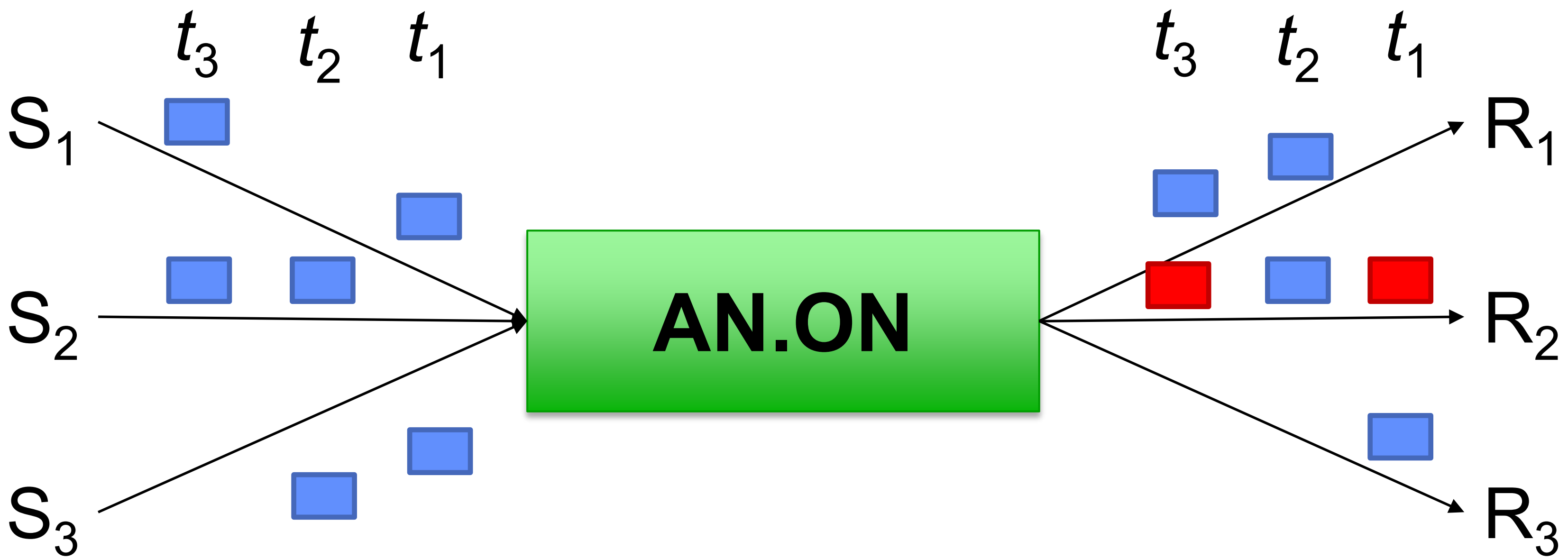
Zero Knowledge Proof of Knowledge (ZKP)

Long Term Intersection Attacks



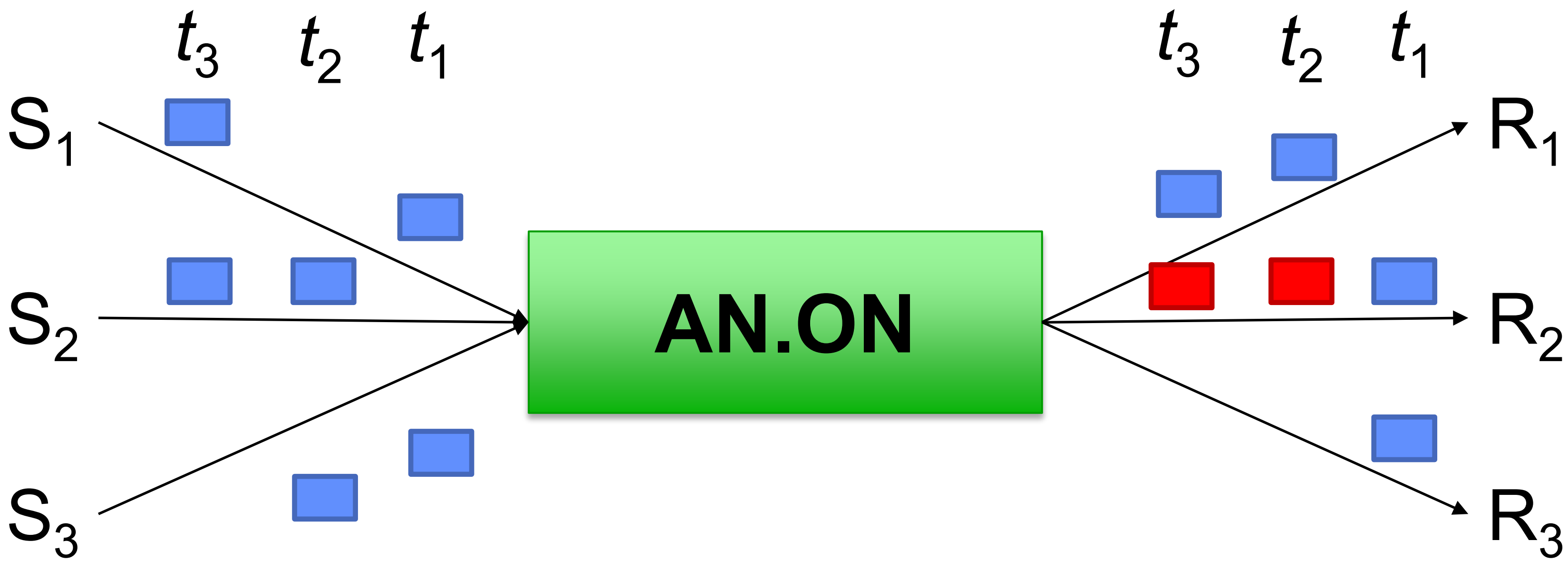
- Deanonymisation by Linkability of Messages

Long Term Intersection Attacks



- Deanonymisation by Linkability of Messages

Long Term Intersection Attacks



- Deanonymisation by Linkability of Messages