EPFL

SPRING
SECURITY AND PRIVACY ENGINEERING LABORATORY

CS-523 Advanced Topics in Privacy Enhancing Technologies

Anonymous Communications

Theresa Stadler
SPRING Lab
theresa.stadler@epfl.ch

- Slides adapted from: Carmela Troncoso, George Danezis, and Matthew Wright

Introduction

Anonymous communications

Course aim: learn **toolbox for privacy engineering**

toolbox
to protect communications from inferences

notions
to express privacy communications

mechanisms
to protect traffic data

attacks
on traffic data

Application Layer

Network Layer

Goals

What should you learn today?

- Understand the **need to protect network data**
- Understand that in electronic communications **anonymity can have many flavours**
- Understanding the **key aspects** of anonymous communications designs
  - **Infrastructure decisions**
  - **Adversarial models**
  - **Protection goals**
- Which are the **most used anonymous communications** design alternatives
- Understand which **attacks** are most relevant when protecting privacy of network data

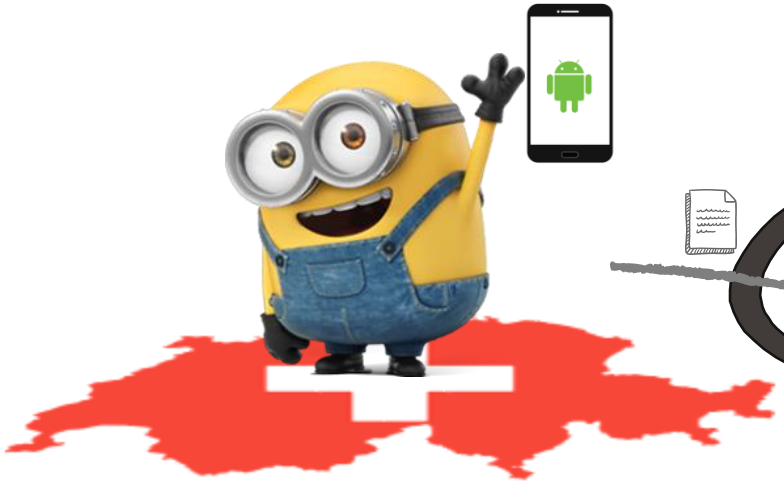
Metadata encodes a lot of information

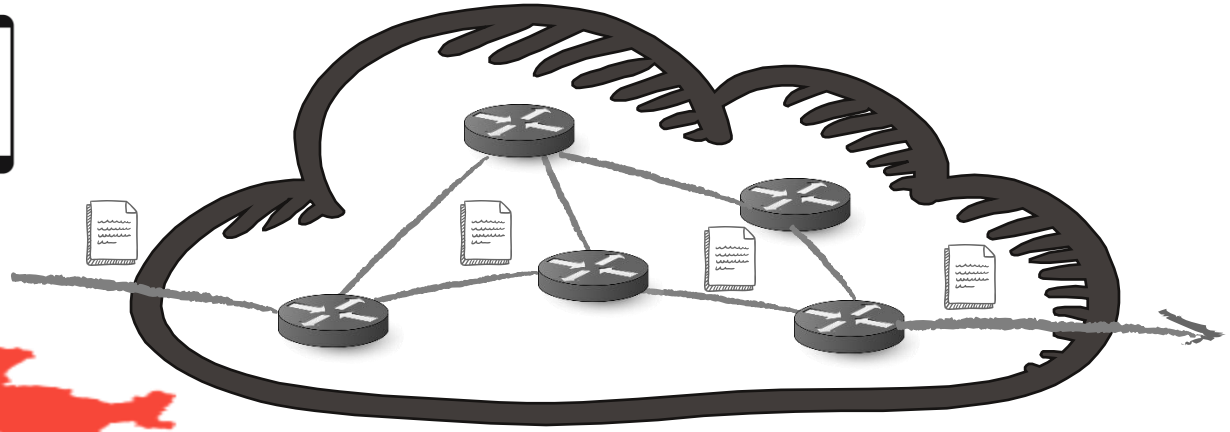
Pseudoidentifier
Pseudoidentifier Sensitive

Device type, OS, applications/software, sensors,….

Pseudoidentifier Location

IP, MAC, routes,…



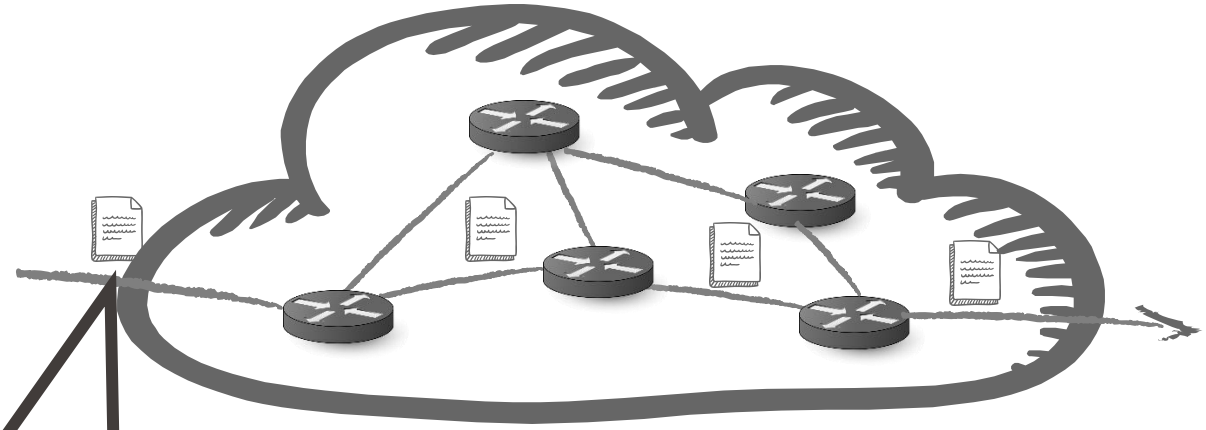


Pseudoidentifier Sensitive

Location

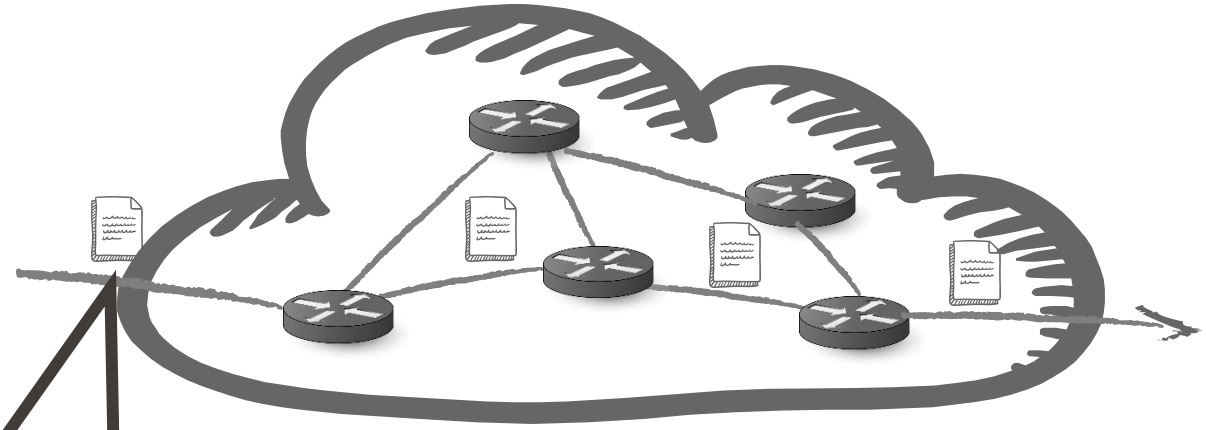
Metadata all around

We have encryption, what is the problem?



%Q}!$#!{}{¨@%%:@}
@$@@¨}{}{@@}{}@{
{@}@#$¨}{%@$%@

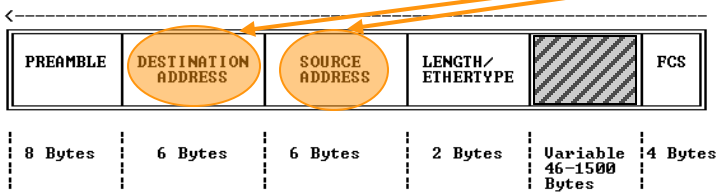
We have encryption, what is the problem?



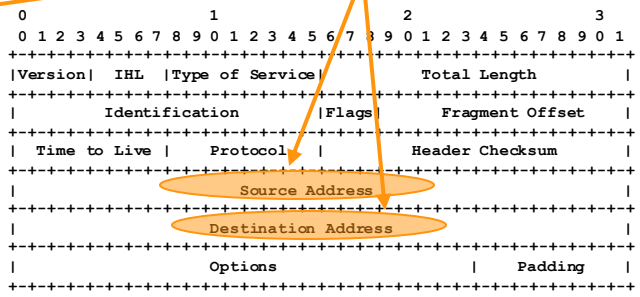
Same for TCP, SMTP, IRC, HTTP, ...

%Q}!$#!{}{¨@%%:@}
@$@@¨}{}{@@}{}@{
{@}@#$¨}{%@$%@

Weak identifier



Ethernet
(IEEE 802.3, 1997)



IPv4 Header
(RFC 791, 1981)

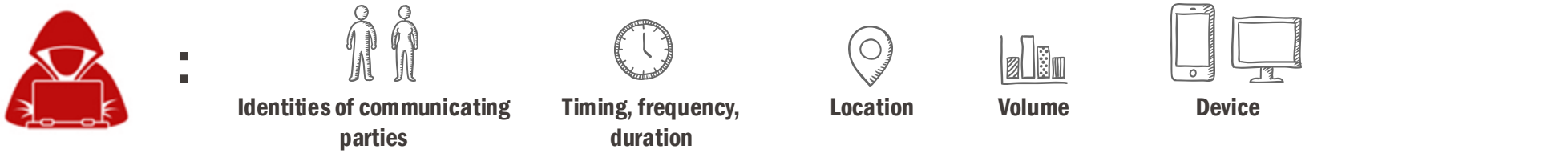
The problem is traffic analysis

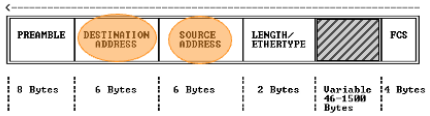
Wikipedia: traffic analysis is the process of intercepting and examining messages in order to deduce information from patterns in communication

Traffic what?

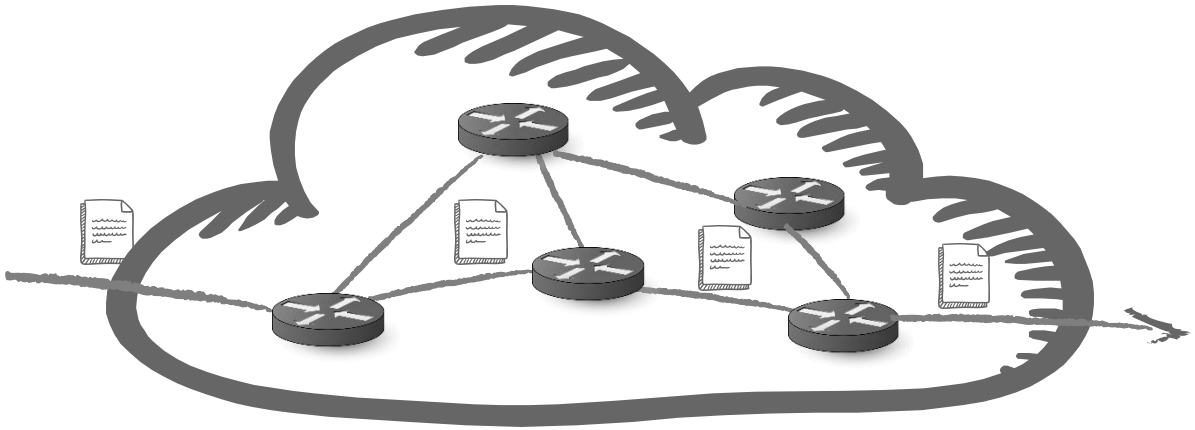
Wikipedia: traffic analysis is the process of intercepting and examining messages in order to deduce information from patterns in communication

Making use of “just” traffic data of a communication (aka **metadata**) to extract information (as opposed to analyzing content or perform cryptanalysis)





Ethernet
(IEEE 802.3, 1997)



Traffic what?

Wikipedia: traffic analysis is the process of intercepting and examining messages in order to deduce information from patterns in communication

Making use of “just” traffic data of a communication (aka **metadata**) to extract information (as opposed to analyzing content or perform cryptanalysis)

Military Roots

M. Herman: “These non-textual techniques can establish **targets' locations**, order-of-battle and **movement**. Even when messages are not being deciphered, traffic analysis of the target's Command, Control, Communications and intelligence system and its patterns of behavior provides indications of his **intentions** and **states of mind**”

WWI: British troops finding German boats.

WWII: assessing size of German Air Force, fingerprinting of transmitters or operators (localization of troops).

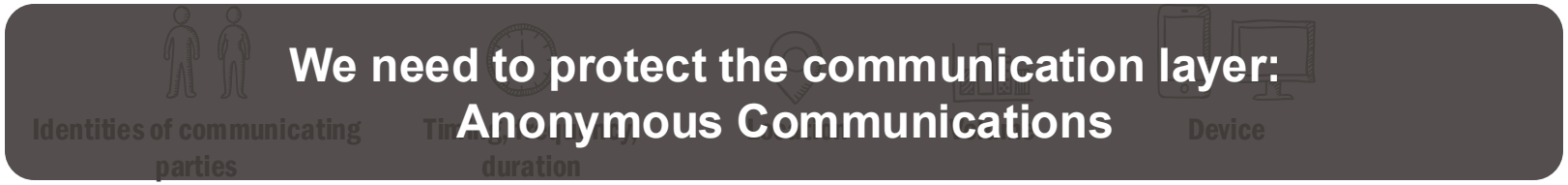
Herman, Michael. Intelligence power in peace and war. Cambridge University Press, 1996.

Traffic what?

Wikipedia: traffic analysis is the process of intercepting and examining messages in order to deduce information from patterns in communication

Making use of “just” traffic data of a communication (aka **metadata**) to extract information (as opposed to analyzing content or perform cryptanalysis)

:



We need to protect the communication layer: Anonymous Communications

MODERN TIMES

Diffie&Landau: ”Traffic analysis, not cryptanalysis, is the backbone of communications intelligence”

Stewart Baker (NSA): “Metadata **absolutely tells you everything about somebody’s life**. If you have enough metadata, you don’t really need content.”

Snowden: XkeyScore, PRISM revelations

Diffie, Whitfield, and Susan Landau. Privacy on the line: The politics of wiretapping and encryption. MIT press, 2010.
http://www.theguardian.com/world/interactive/2013/nov/01/snowden-nsa-files-surveillance-revelations-decoded

Why anonymous communications? Isn't this only for cybercriminals?

Who needs anonymous communications?

Cyber-criminals: DRM infringement, hacker, spammer, terrorist, etc.

https://www.eff.org/deeplinks/2013/10/online-anonymity-not-only-trolls-and-political-dissidents
http://geekfeminism.wikia.com/wiki/Who_is_harmed_by_a_%22Real_Names%22_policy%3F

Why anonymous communications? Isn't this only for cybercriminals?

Who needs anonymous communications?

Cyber-criminals: DRM infringement, hacker, spammer, terrorist, etc.

But even more importantly:

People who need special protections

- Journalists
- Whistleblowers
- Human rights activists
- Military/intelligence personnel
- Abuse victims

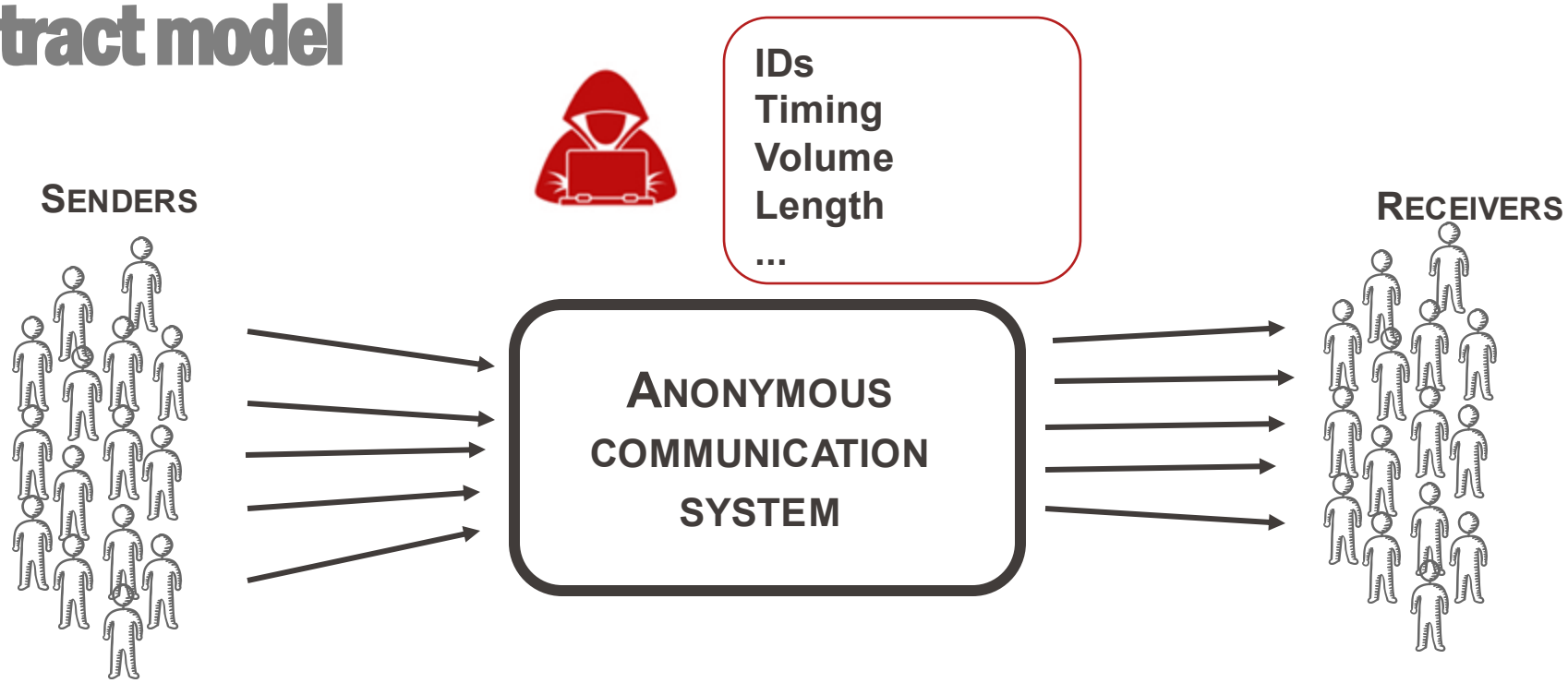
Average users

- Avoid tracking by advertising companies
- Protect sensitive personal information from businesses, like insurance companies, banks, etc.
- Express unpopular or controversial opinions
- Have a dual life
- Try uncommon things
- Specialised applications
  - eVoting
  - Auctions
- …

https://www.eff.org/deeplinks/2013/10/online-anonymity-not-only-trolls-and-political-dissidents
http://geekfeminism.wikia.com/wiki/Who_is_harmed_by_a_%22Real_Names%22_policy%3F

Anonymous communications

Abstract model



Bitwise unlinkability

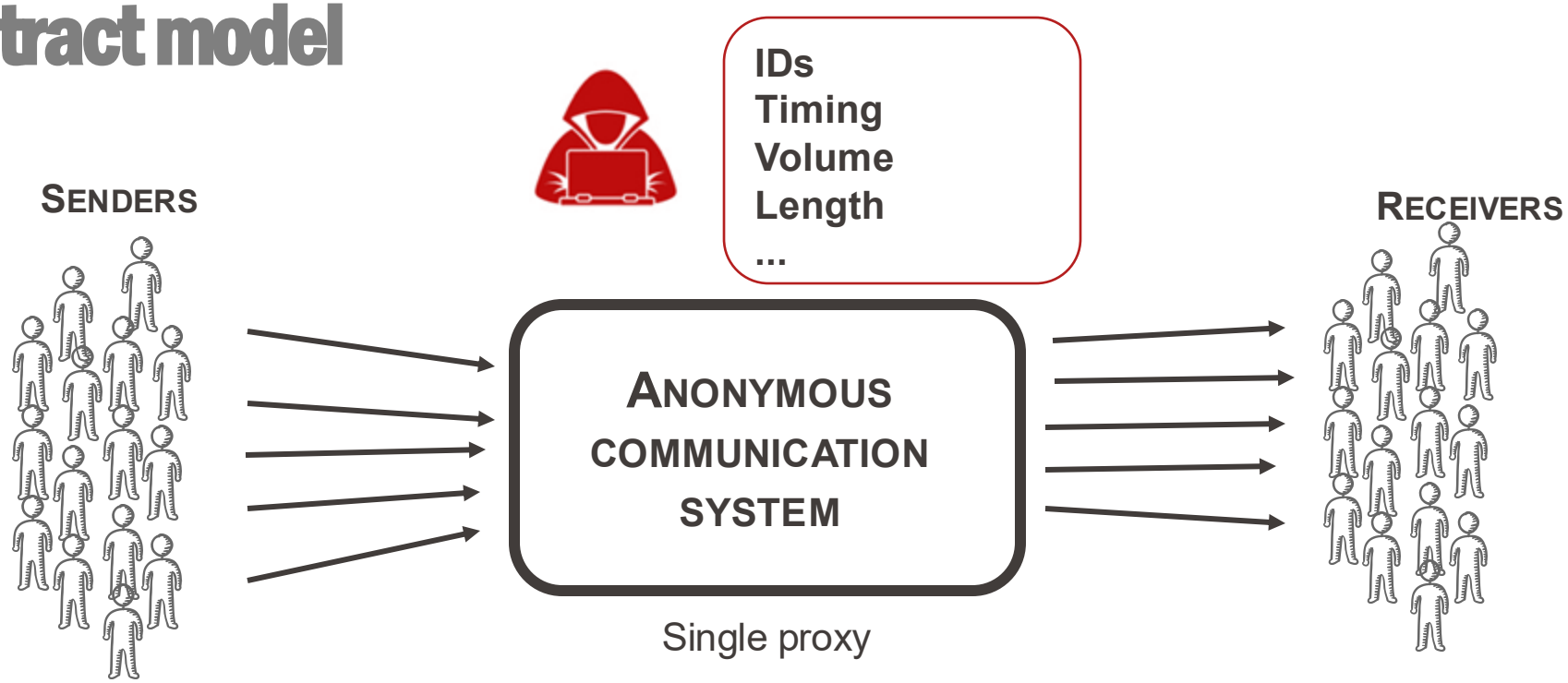
Crypto to make inputs and outputs bit patterns different

(re)packetizing + (re)schedule

Destroy patterns (traffic analysis resistance)

Anonymous communications

Abstract model



Bitwise unlinkability

Crypto to make inputs and outputs bit patterns different

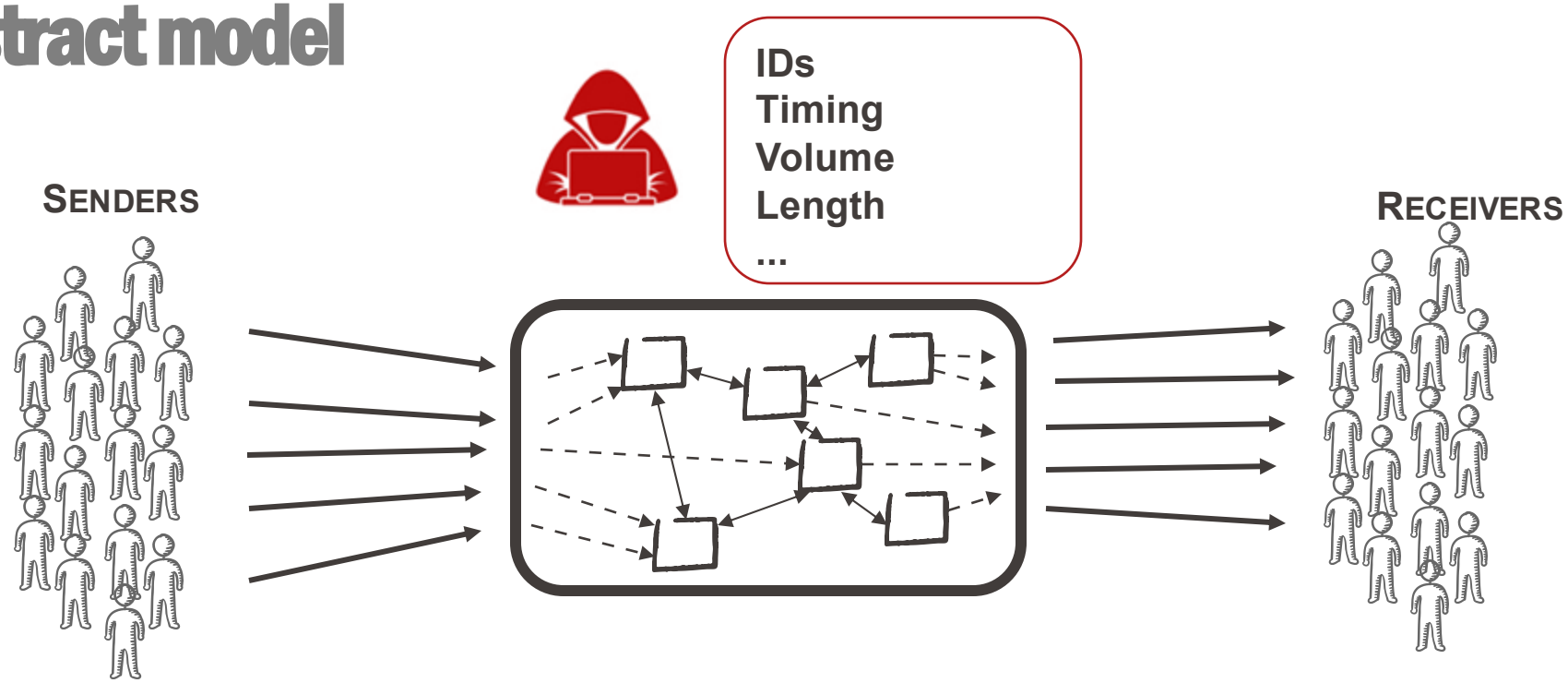
(re)packetizing + (re)schedule

Destroy patterns (traffic analysis resistance)

Performance problem: Low throughput
Security problem: Corrupt proxy or proxy hacked / coerced
Real case: Penet.fi vs the church of scientology (1996)

Anonymous communications

Abstract model



Bitwise unlinkability

Crypto to make inputs and outputs bit patterns different

(re)packetizing + (re)schedule + (re)routing

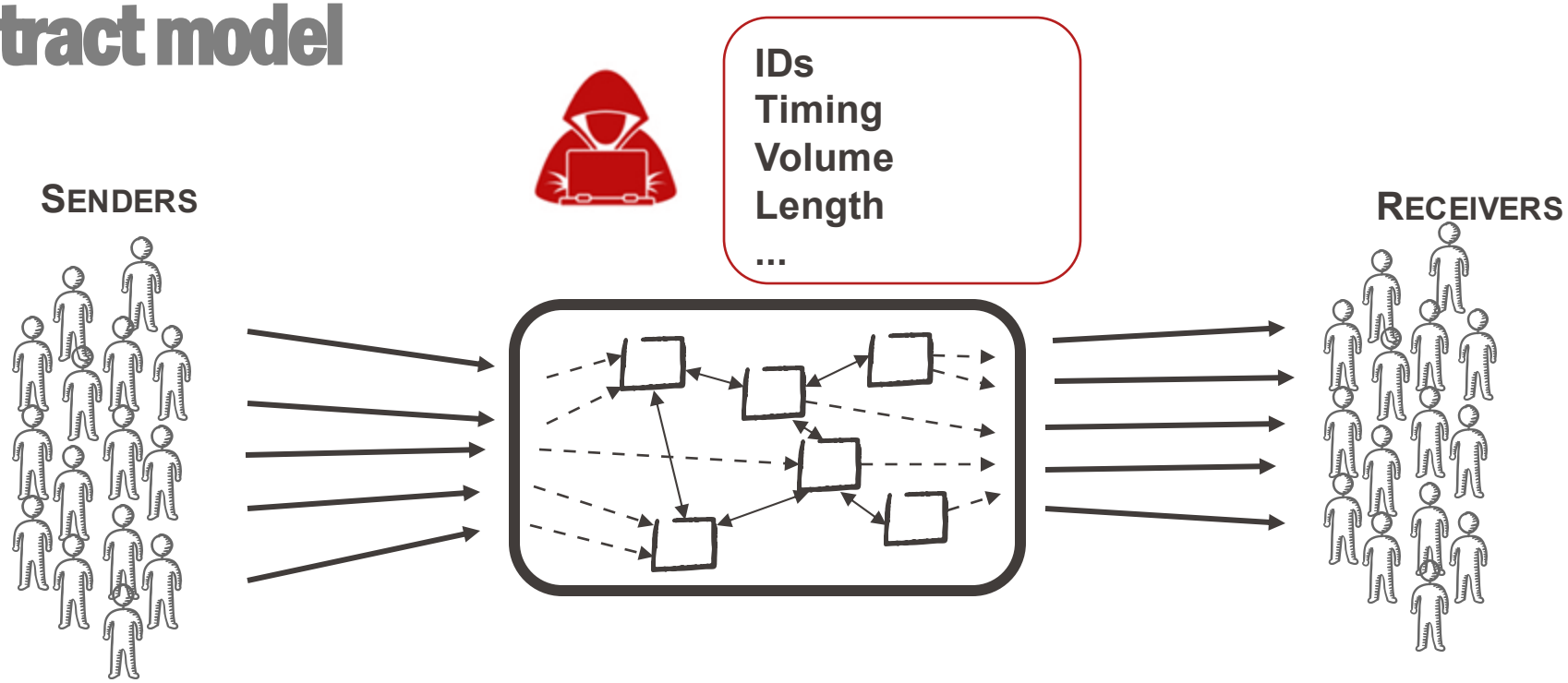
Destroy patterns (traffic analysis resistance)

Load balancing

Distribute trust

Anonymous communications

Abstract model



Bitwise unlinkability ✅

Crypto to make inputs and outputs bit patterns different

(re)packetizing + (re)schedule + (re)routing ❌

Destroy patterns (traffic analysis resistance)

Load balancing

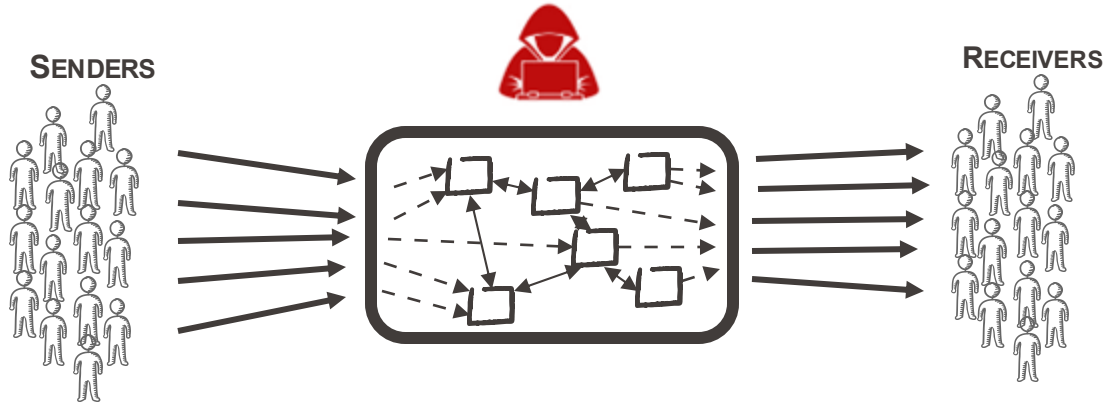
Distribute trust

Deployment challenges

- Bandwidth and delay
- Churn
- Intrinsic network differences
- Trust?

Anonymous communications design

Key aspects



Infrastructure (who routes messages):

User-based:
nodes = users (peer to peer)

User-independent:
nodes = others, not (necessarily) trusted

Hybrid:
nodes = mix users and others

Adversarial capabilities:

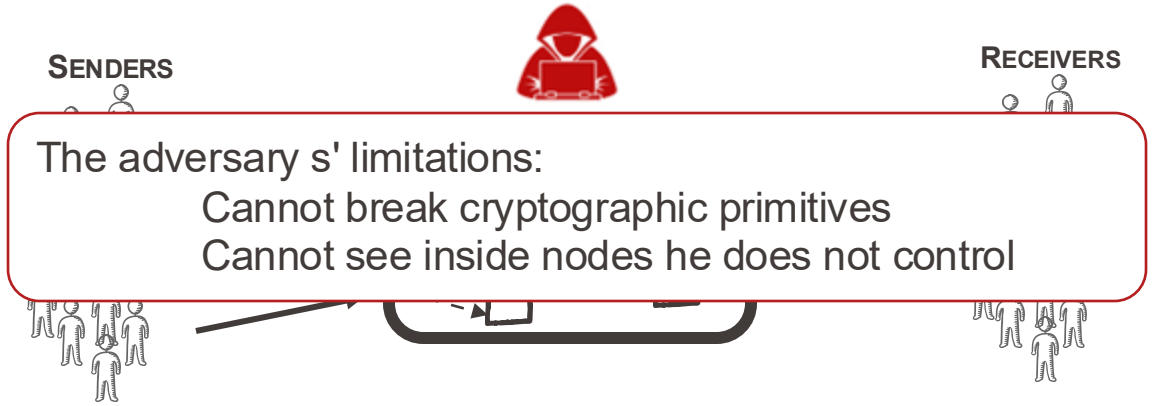
Global vs. partial:
What can the adversary see?

Active vs. passive:
What can the adversary do?

Internal vs. external:
Can the adversary see inside?

Anonymous communications design

Key aspects



Infrastructure (who routes messages):

User-based:
nodes = users (peer to peer)

User-independent:
nodes = others, not (necessarily) trusted

Hybrid:
nodes = mix users and others

Adversarial capabilities:

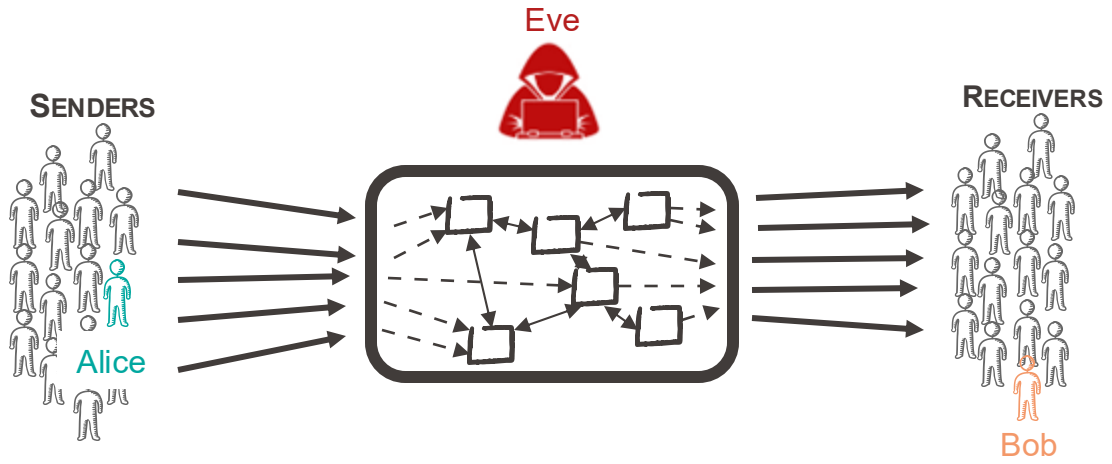
Global vs. partial:
What can the adversary see?

Active vs. passive:
What can the adversary do?

Internal vs. external:
Can the adversary see inside?

Anonymous communications design

Key aspects



Goals (What do we want to protect?):

Sender anonymity:
Bob (or Eve) cannot know that Alice sent the message

Receiver anonymity:
Alice (or Eve) cannot know that Bob received the message

Bidirectional anonymity:
Alice and Bob cannot know each others' identity

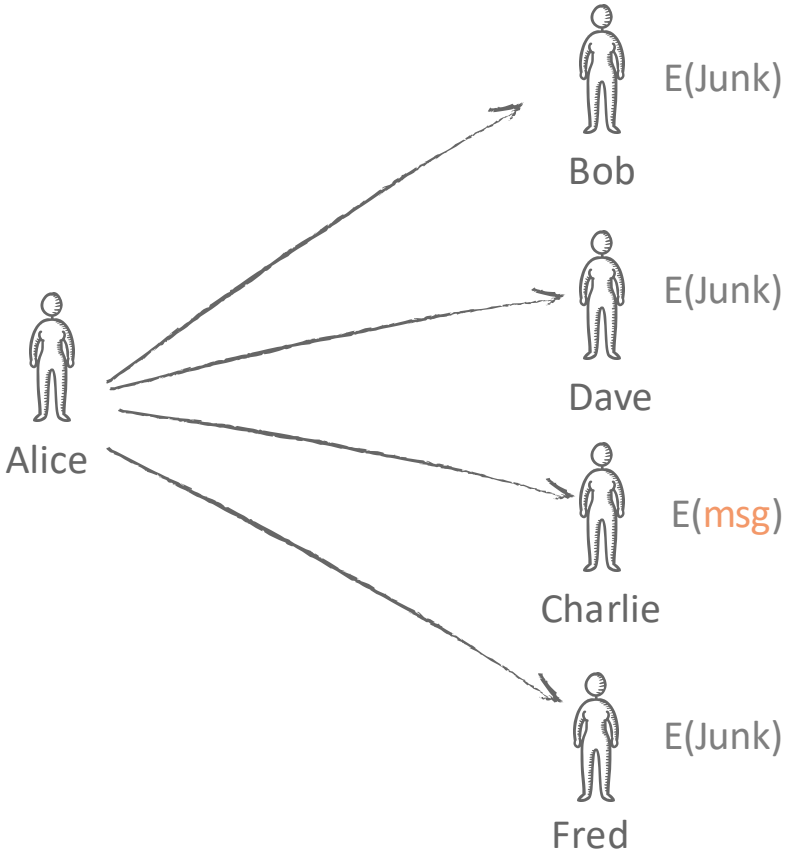
3rd party anonymity:
Bob and Alice know each other, Eve doesn't

Unobservability:
Eve cannot tell if Alice or Bob send/receive anything

Unlinkability:
Messages from Alice or Bob cannot be linked

Anonymous Communications

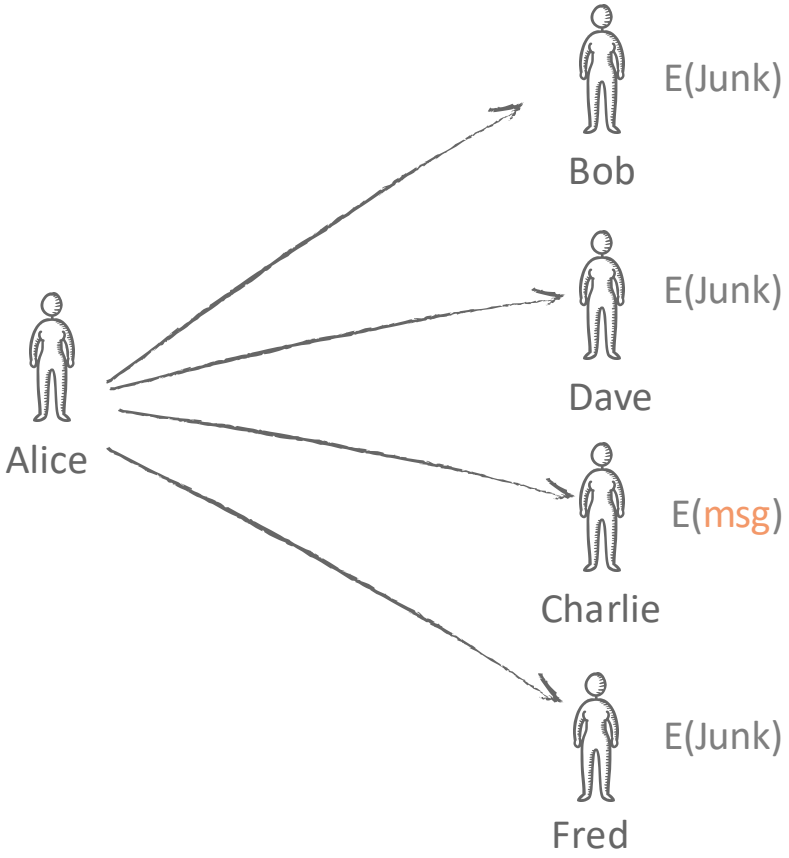
A naïve approach



Simple receiver anonymity!

Anonymous Communications

A naïve approach



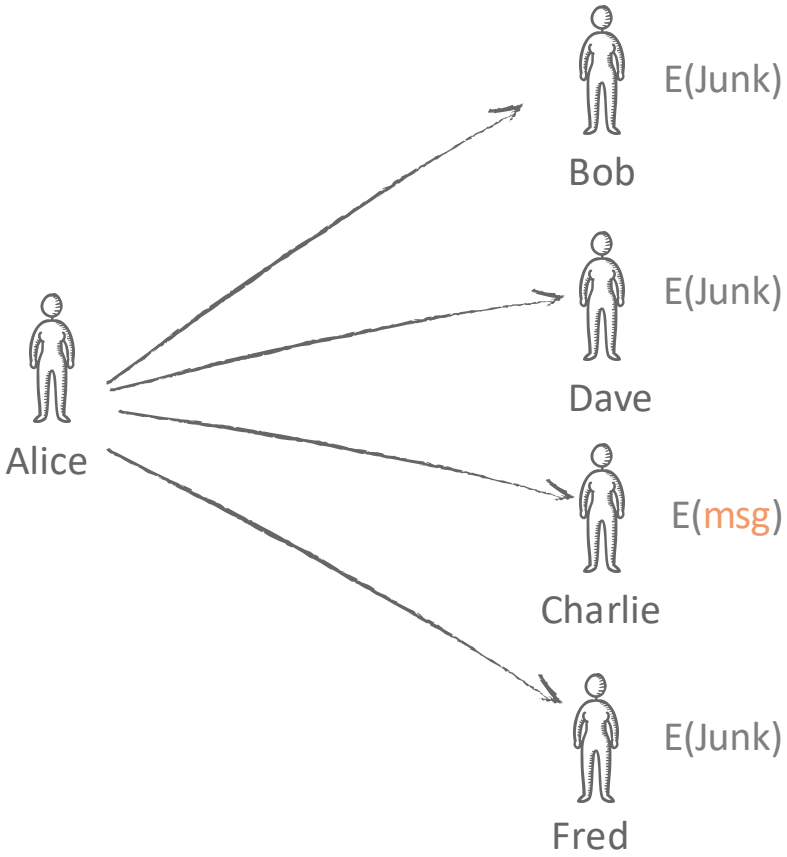
Simple receiver anonymity!

Why look at such a simple scheme?

- So you do not try re-invent this…
- …**in any form**
- Because it provides a baseline

Anonymous Communications

A naïve approach



Simple receiver anonymity!

Why look at such a simple scheme?

- So you do not try re-invent this…
- …**in any form**
- Because it provides a baseline
- …also in terms of drawbacks
  - Coordination
  - Sender anonymity
  - Bandwidth
  - Latency

Anonymous Communications

Designs

4 paradigmatic examples (basic designs → many follow ups!)

- Different goals
- Different infrastructure
- Different adversarial model
- Different attacks

DC Networks

DC Networks
Dining cryptographers

THE PROBLEM

How to send messages in a closed group network with

Sender anonymity: no actor can identify the sender of a message

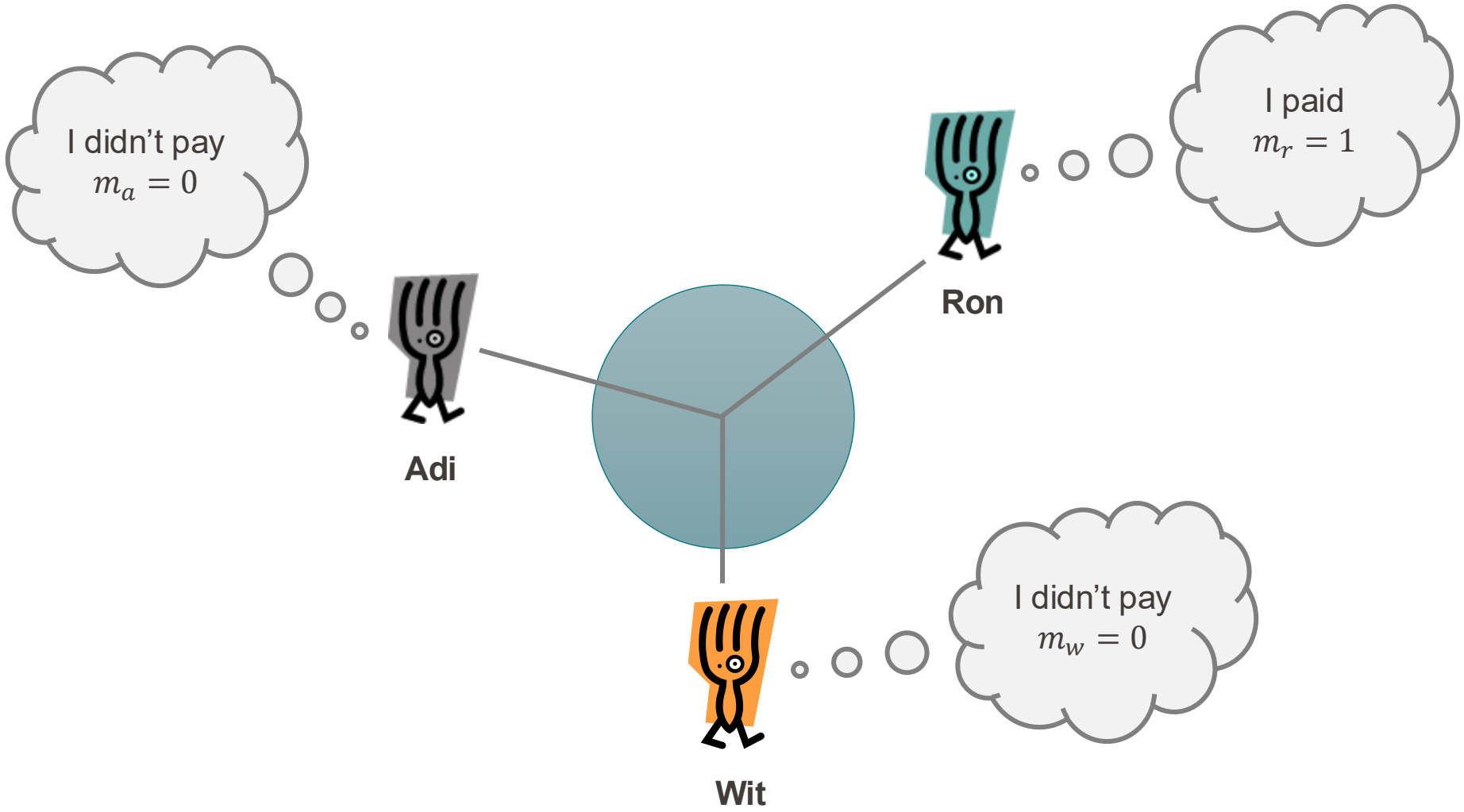
Receiver anonymity: no actor can identify the receiver of a message

"Three cryptographers are having a dinner at a restaurant. At the end the waiter informs them that the dinner has already been paid. Either this was one of them, or the National Security Agency (NSA). The cryptographers want to know whether the NSA paid for the dinner, without divulging the identity of their colleague if it's not the case."

David Chaum.
The dining cryptographers problem: Unconditional sender and recipient untraceability.
Journal of cryptology, 1988, vol. 1, no 1, p. 65-75.

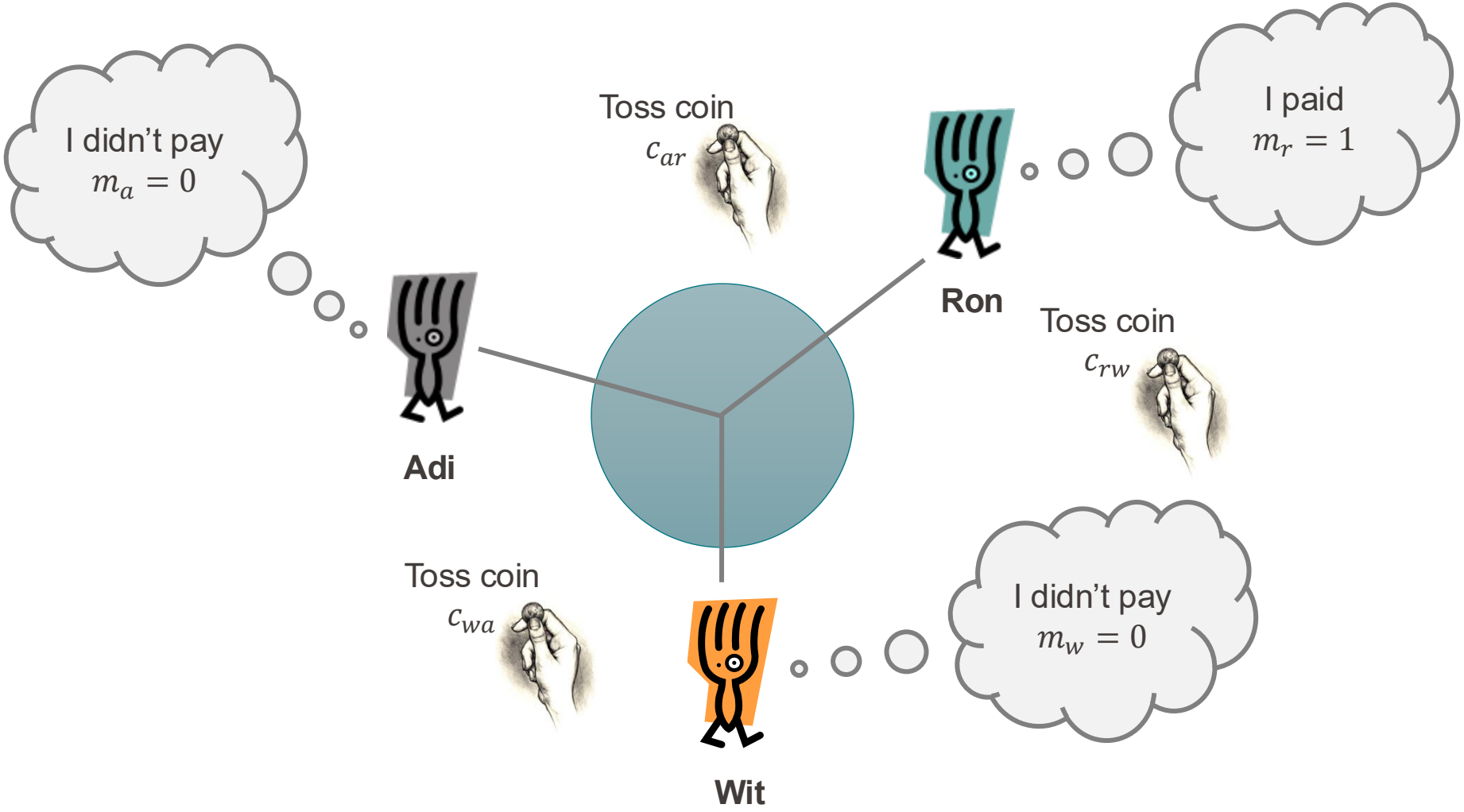
DC Networks

Dining cryptographers



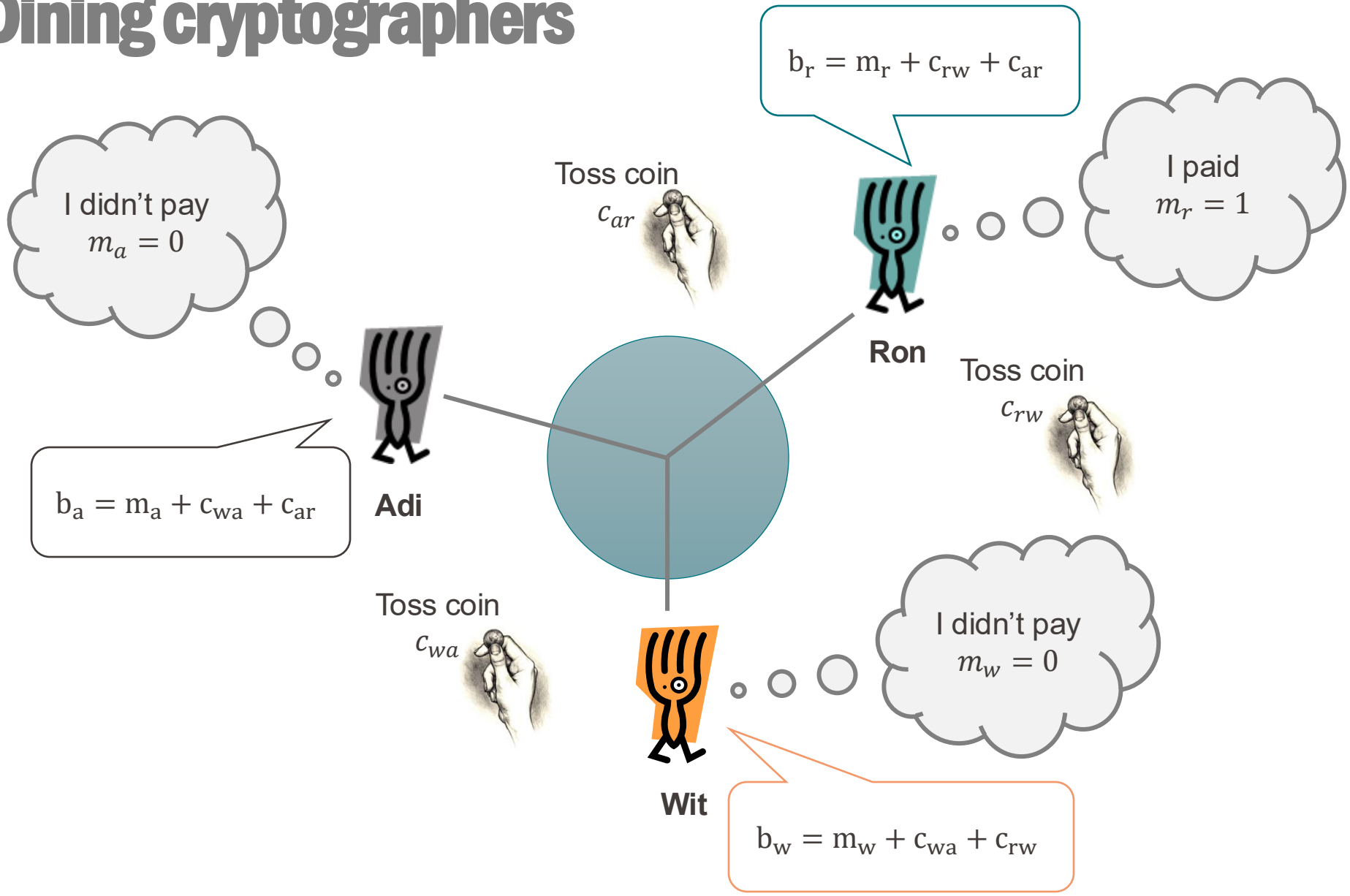
DC Networks

Dining cryptographers



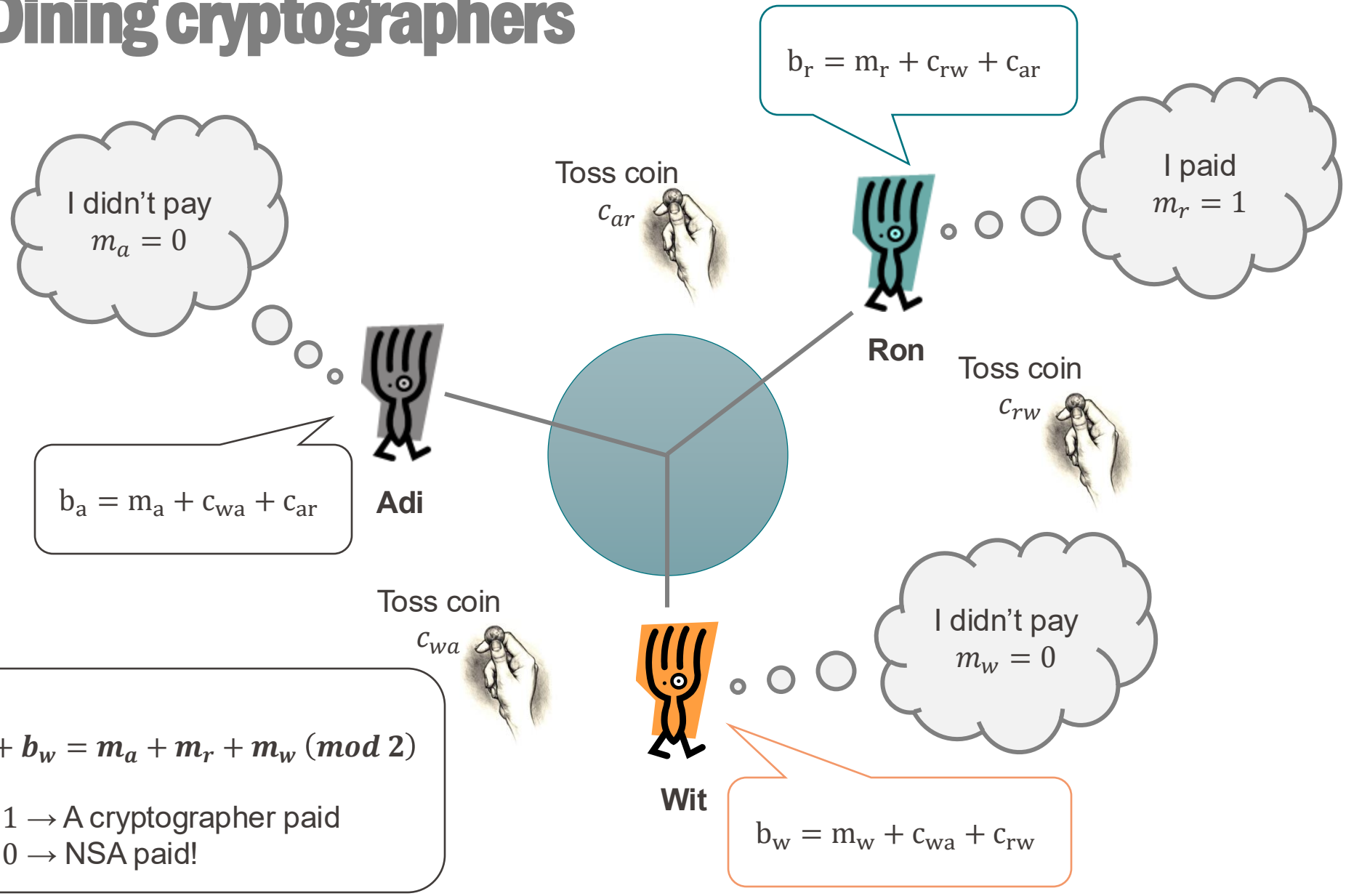
DC Networks

Dining cryptographers



DC Networks

Dining cryptographers



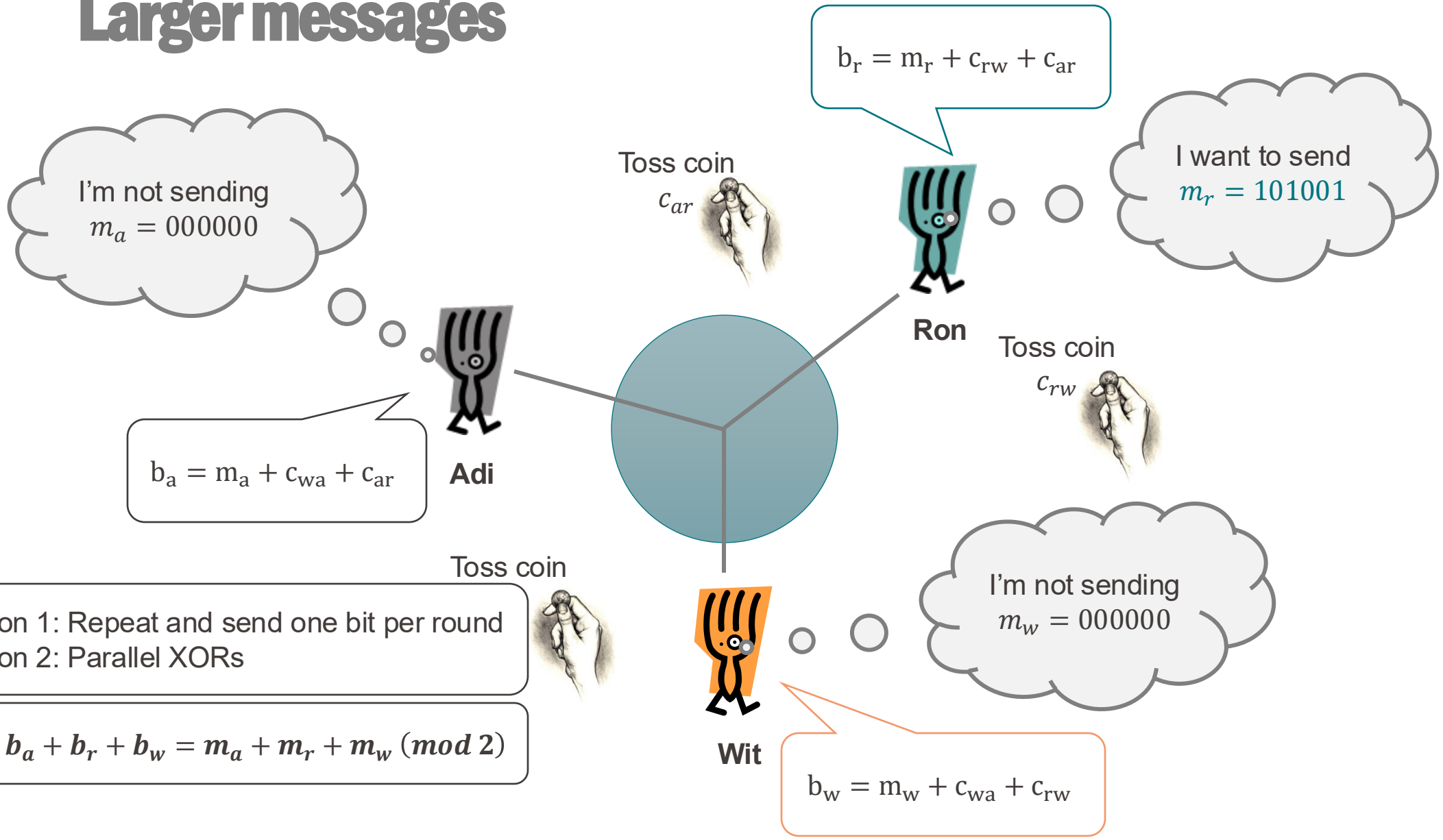
Answer

$\boldsymbol{B = b_a + b_r + b_w = m_a + m_r + m_w \ (mod\ 2)}$

$B = 1 \rightarrow$ A cryptographer paid

$B = 0 \rightarrow$ NSA paid!

DC Networks
Larger messages



Option 1: Repeat and send one bit per round
Option 2: Parallel XORs

$$B = b_a + b_r + b_w = m_a + m_r + m_w \ (mod\ 2)$$

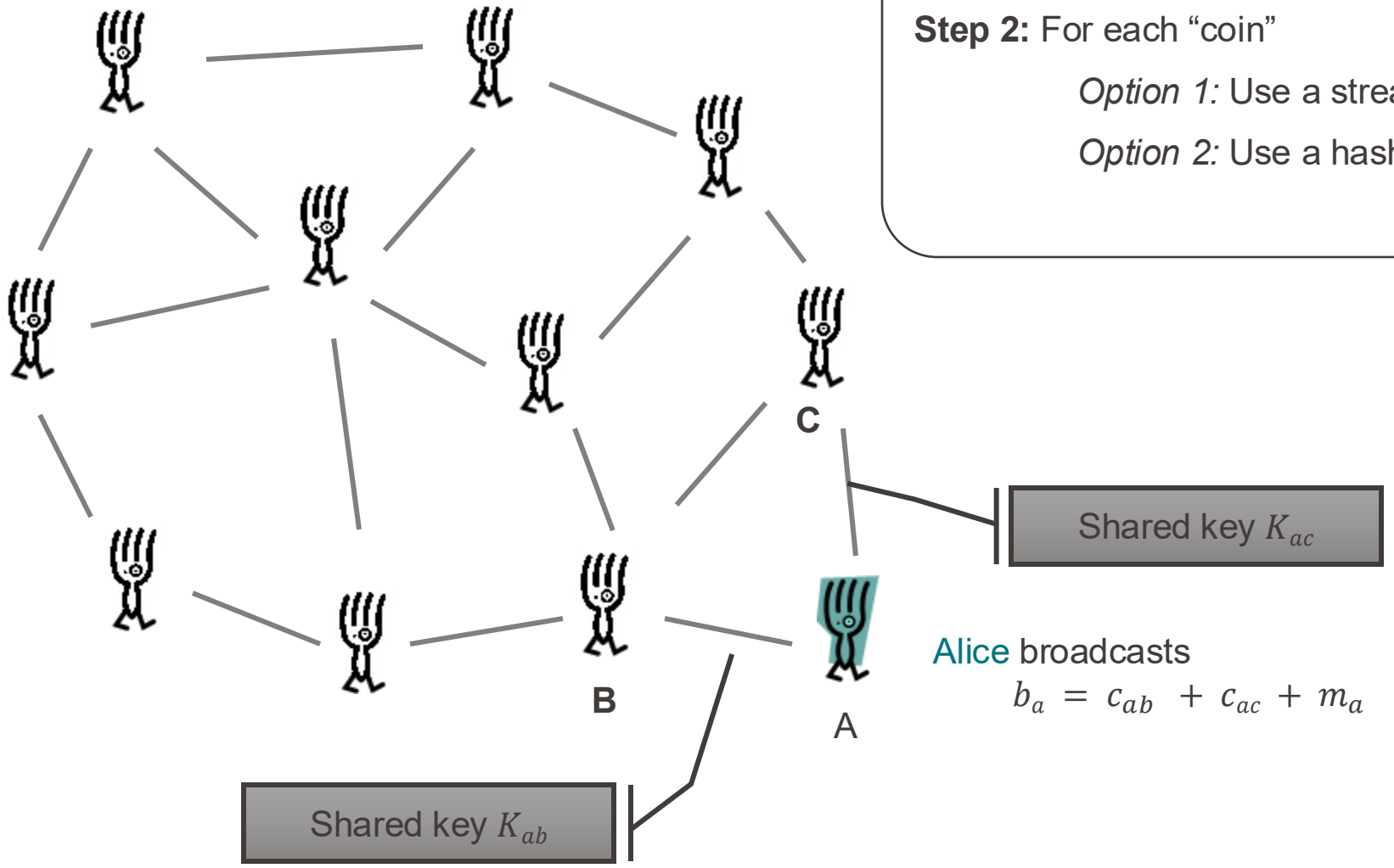
DC Networks
Larger messages

Step 1: Establish a shared key (offline, Diffie Hellman,…)

Step 2: For each “coin”

Option 1: Use a stream cipher to produce bits

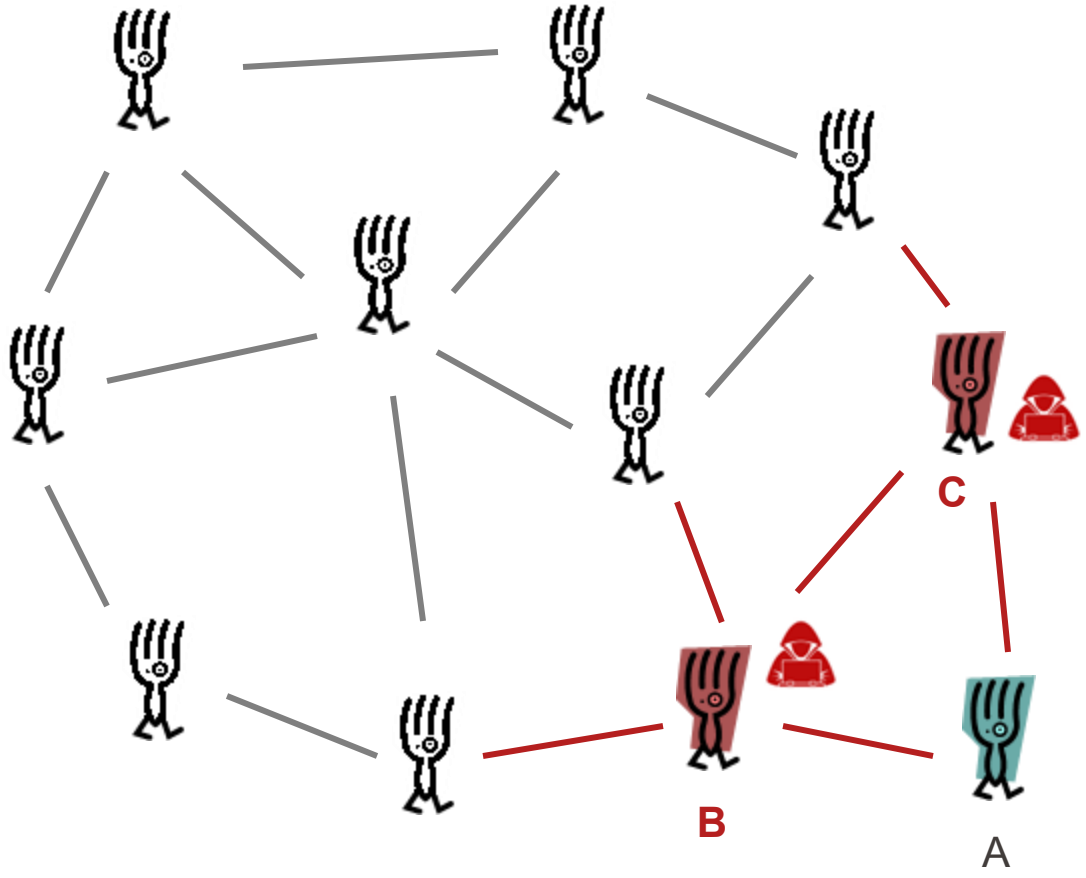
Option 2: Use a hash function: $c_i = h(K, i)$



Alice broadcasts

$$b_a = c_{ab} + c_{ac} + m_a$$

DC Networks
Security



If **B** and **C** corrupt, they know c_{ab} and c_{ac}

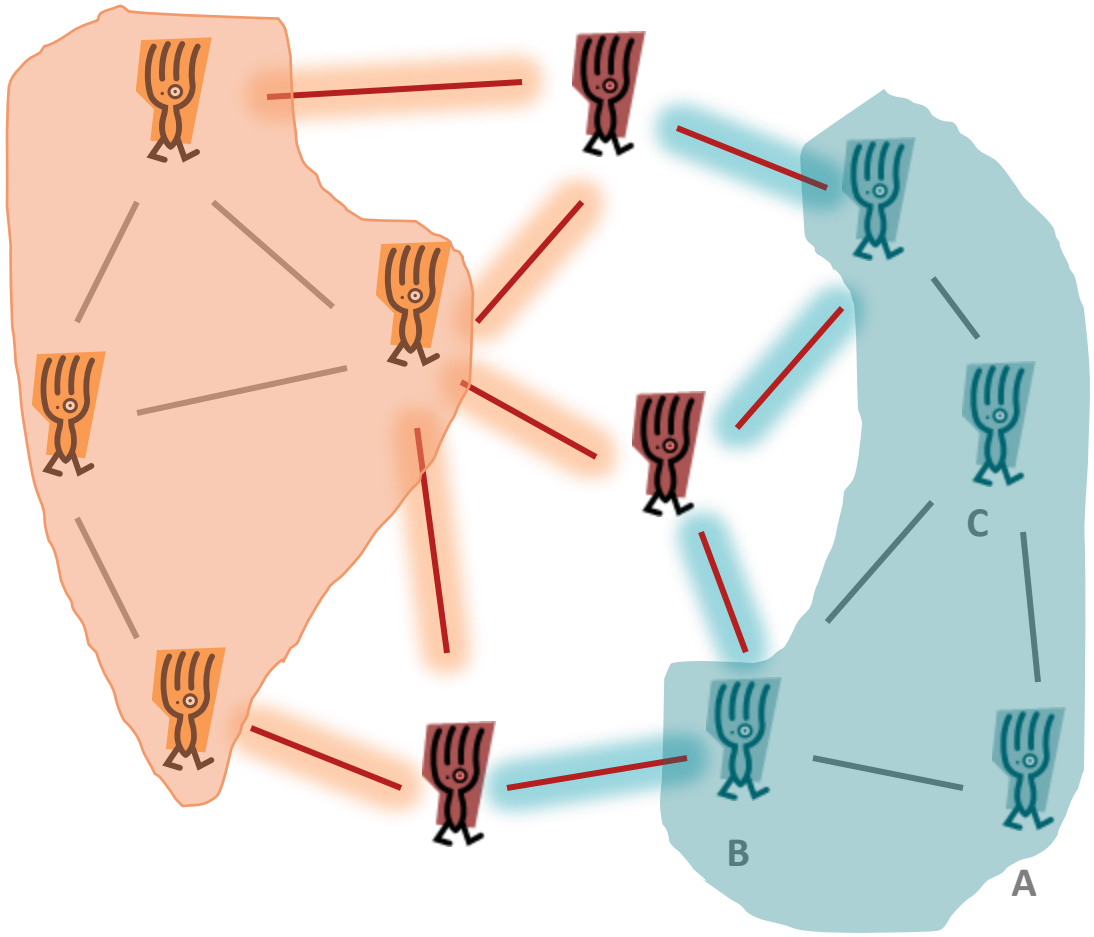
$$b_a = c_{ab} + c_{ac} + m_a + c_{ab} + c_{ac}$$

→ Reveal Alice's secret bit m_a
No anonymity

Alice broadcasts

$$b_a = c_{ab} + c_{ac} + m_a$$

DC Networks
Security



Adversaries can establish a partition in the graph!

Red partitions into yellow and blue

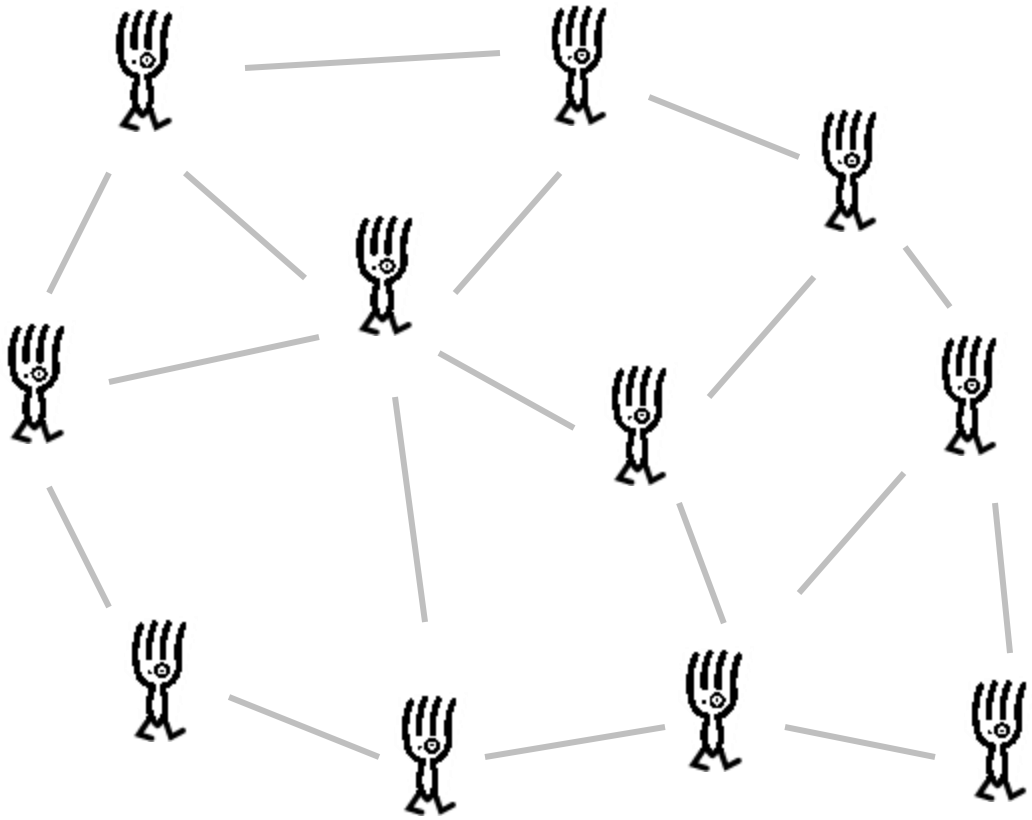
Sum = 0 | Sum = 1

Anonymity set reduced from 8 to 4!!

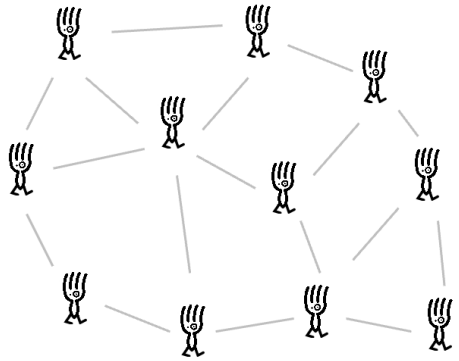
DC Networks
Implementation choices

How to distribute messages (efficiently?)

Option 1: Trusted third party
Option 2: Ring
Option 3: Tree



DC Networks
Summary



Goal: Sender/receiver anonymity

Infrastructure: who routes messages

User-based:
nodes = users (peer to peer) = 

User-independent:
nodes = others, not (necessarily) trusted

Hybrid:
nodes = mix users and others

Adversarial capabilities:

Global *vs. partial*:
What can the adversary see?

Active *vs. passive*:
What can the adversary do?

Internal *vs. external:*
Can the adversary see inside?

Waidner, M., & Pfitzmann, B. The dining cryptographers in the disco: Unconditional sender and recipient untraceability with computationally secure serviceability. *EUROCRYPT*, *89*.

DC Networks
Summary

Security is great!

- Full key sharing graph ⇔ perfect anonymity
  - In a given setting: all users available all the time

Communication cost – BAD (N broadcasts for each message!)

- Naive: $O(N^2)$ cost, O(1) Latency
- Not so naïve
  - Ring: O(N) messages, O(N) latency
  - Tree: O(N) messages, O(logN) latency
  - Centralized: O(N) messages, O(1) latency [but trust!]

Not practical for large(r) N! ☹

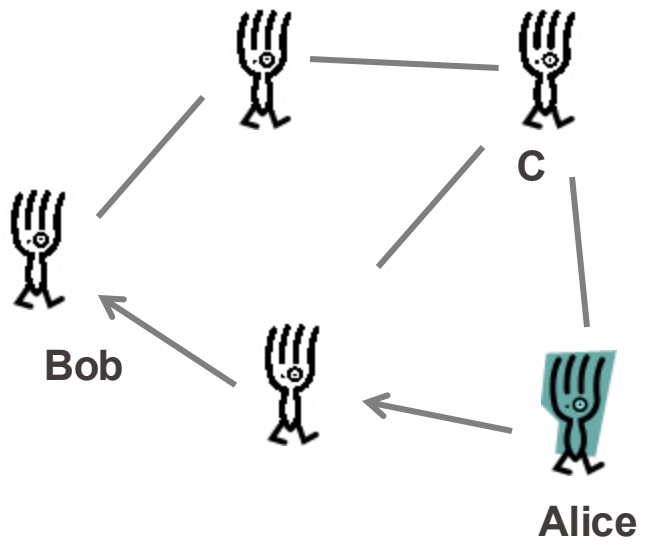
- Only for small settings, e.g., local wireless communications

https://arxiv.org/abs/1710.10237

DC Networks

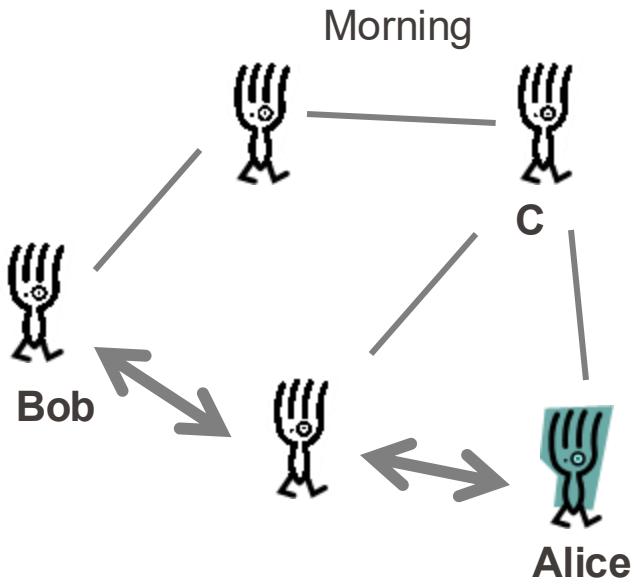
Are all users online all the time?

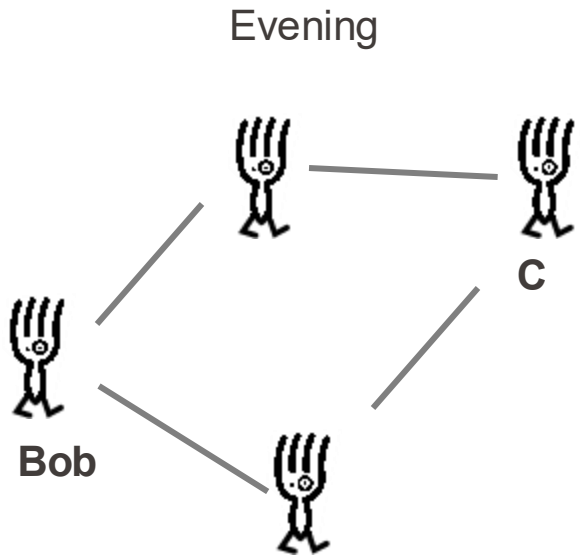
Assumption



- Single message from Alice to Bob or between a closed group
- All always online and coordinated

Reality

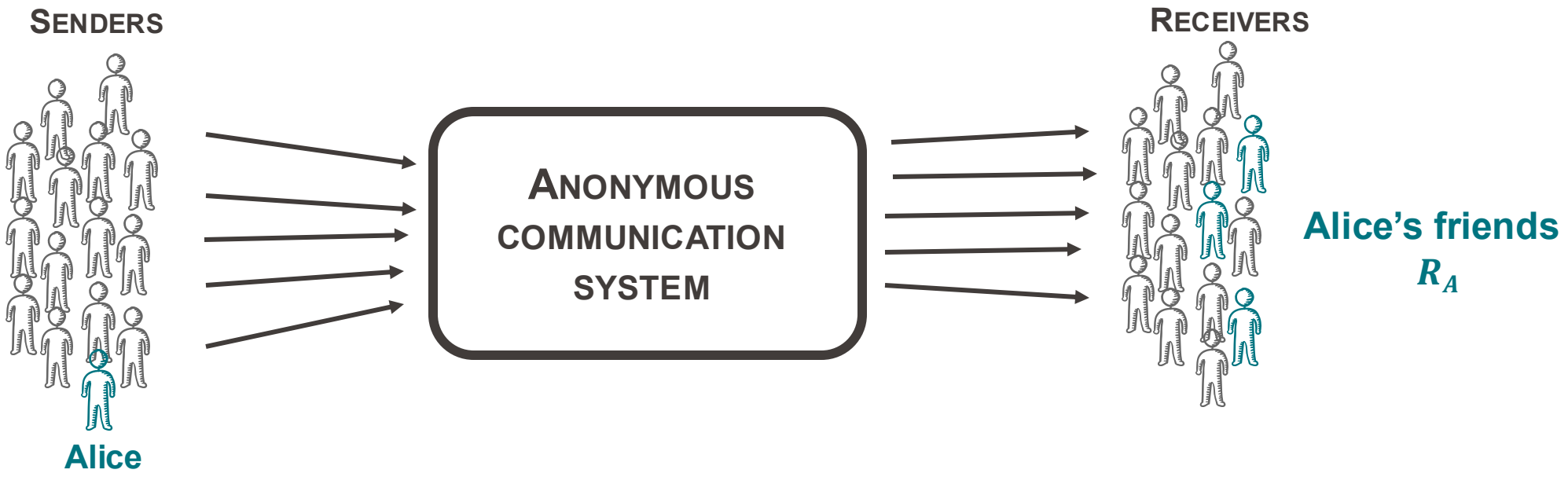




- Alice has a few friends that she messages often
  - Interactive stream between Alice and Bob (TCP)
  - Emails from Alice to Bob (SMTP)
- Alice **is not always on-line** (network churn)

Repetition → Patterns → Attacks

Fundamental limits

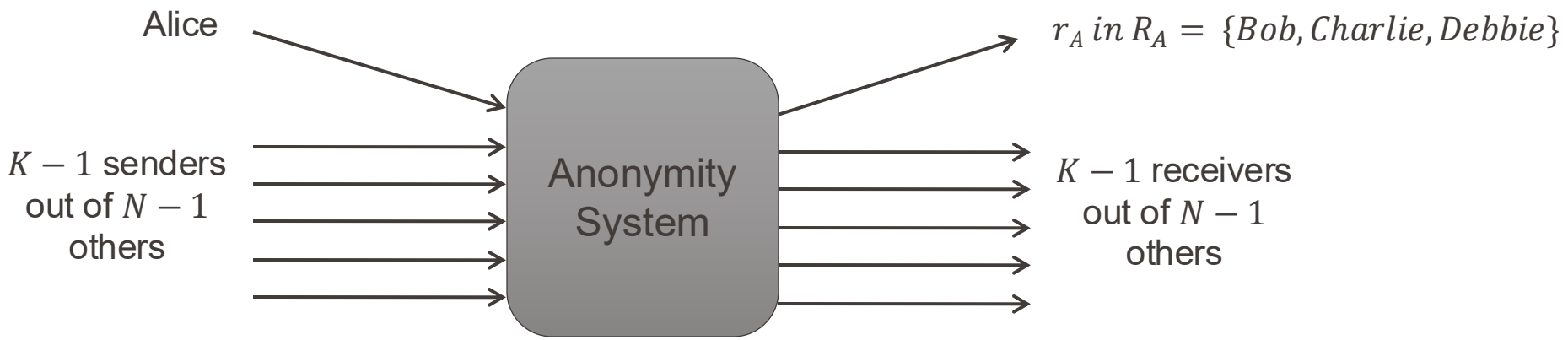


Setting:

- N senders / receivers – Alice is one of them
- Alice messages a small number of friends (set of size M): $r_A \; in \; R_A = \{Bob, Charlie, Debbie\}$
- Adversary's goal: Infer who out of the N receivers are Alice's friends?

Kesdogan, D., Agrawal, D., & Penz, S. Limits of anonymity in open environments. In *International Workshop on Information Hiding* (pp. 53-69). 2002

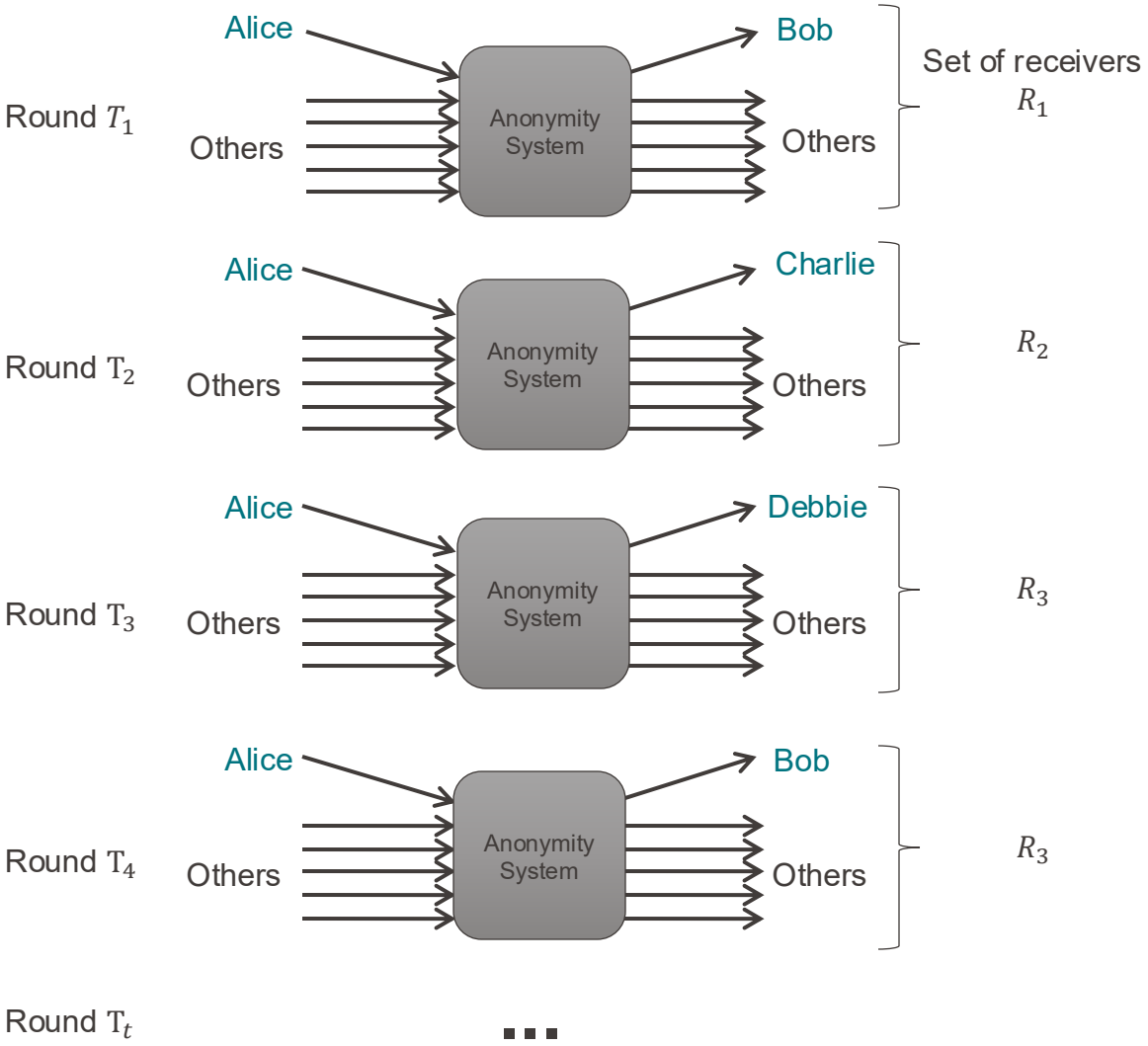
A single round
Perfect anonymity



In each communication round:

- K messages from senders to anonymous receivers
- Alice (1 out of K) sends a single message to one of her friends
- Anonymity set size K
  - Entropy metric $E_A = \log K$
- Perfect anonymity

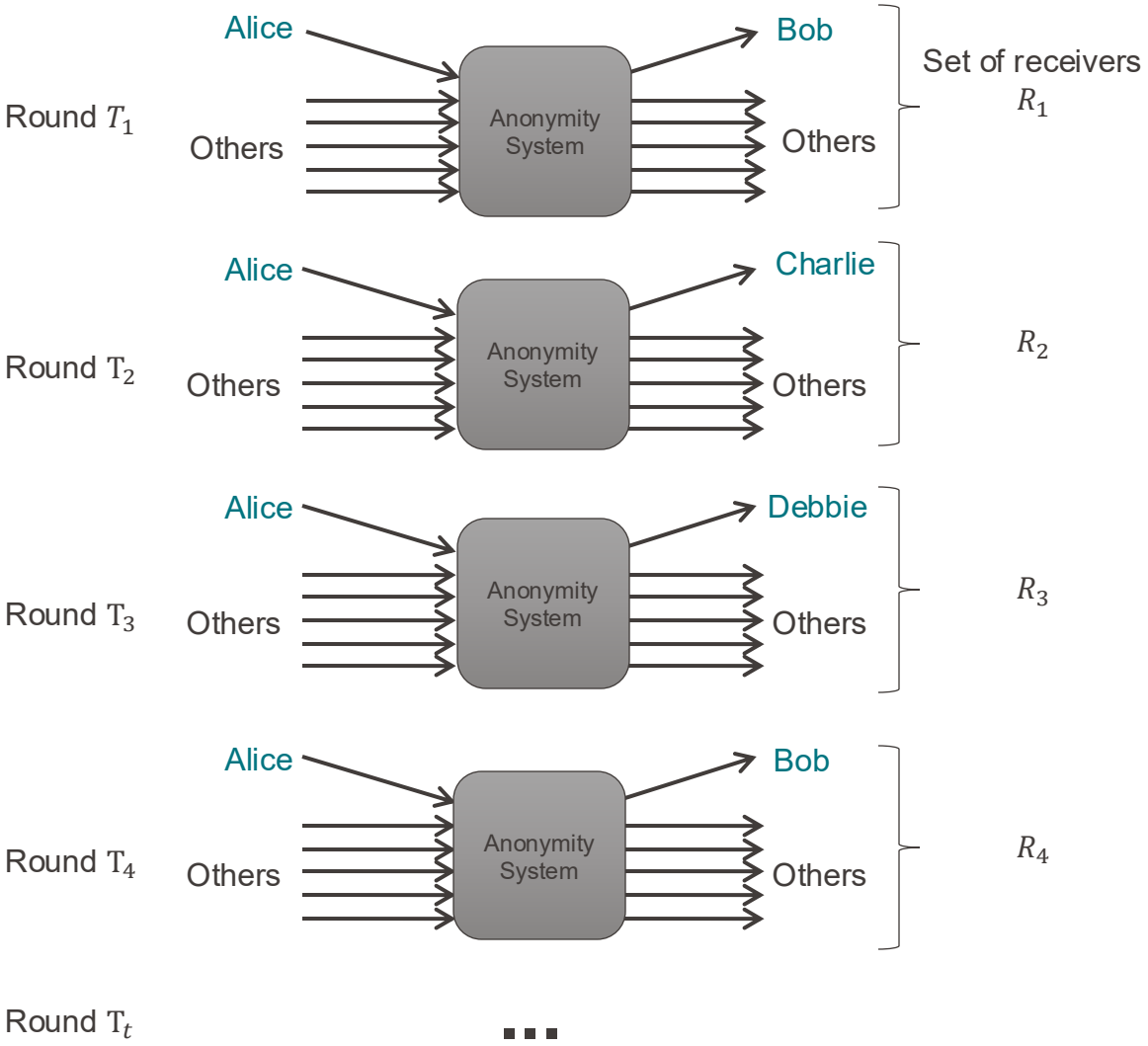
With many rounds…



- Adversary observes many rounds in which Alice participates while other senders vary
- In each round in which Alice participates, one of the K receivers is one of her friends

Agrawal, D., & Kesdogan, D. Measuring anonymity: The disclosure attack. *IEEE Security & privacy*, *99*(6), 27-34. 2003

With many rounds…
Disclosure attacks



1) Learning phase

Find M mutually disjoint receiver sets - that is, each set has only one of Alice's friends - by observing Alice's incoming and outgoing messages.

→ Get basis sets $\{R_1, R_2, R_3\}$ such that $R_i \cap R_j = \emptyset$ for $i \neq j$

2) Excluding phase

Observe new receiver sets R' until all of Alice's **non-friends** are excluded from the basis sets

→ Refine each basis set by keeping only intersection $\mathrm{R}' \cap R_i \neq \emptyset$ until each basis set contains only single element $\mathrm{R_i} = \{\mathrm{r}_A\}$

Agrawal, D., & Kesdogan, D. Measuring anonymity: The disclosure attack. *IEEE Security & privacy*, *99*(6), 27-34. 2003

Disclosure attacks
Statistical disclosure attacks

- Note that the friends of Alice will be in the sets more often than random receivers
- How often? Expected number of messages per receiver after t rounds:
  - $\mu_{other} = (1 / N) \cdot (K - 1) \cdot t$
  - $\mu_{Alice} = (1 / m) \cdot t + \mu_{other}$
- Just count the number of messages per receiver when Alice is sending!
  - $\mu_{Alice} > \mu_{other}$

Disclosure attacks
Take aways

- Disclosure attacks are extremely powerful. They can only be countered at a high cost.
- Counter-intuitive: The larger the observed number of recipients N the easier the attack

- Original attack NP-hard but… Statistical disclosure attacks
  - Very efficient to implement (vectorised)
  - Gives faster partial results
  - Can be extended to complex anonymity system

Kesdogan, D., & Pimenidis, L. The hitting set attack on anonymity protocols. In International Workshop on Information Hiding (pp. 326-339). 2004

Danezis, G., & Troncoso, C. Vida: How to use bayesian inference to de-anonymize persistent communications. In *International Symposium on Privacy Enhancing Technologies Symposium* (pp. 56-72). 2009

Crowds

EPFL

Crowds

THE PROBLEM

How to anonymously **browse the internet**

Sender anonymity: no actor can identify the sender of a message

*"Crowds, named for the notion of "blending into a crowd", operates by grouping users into a **large** and **geographically diverse group** (crowd) that collectively issues requests on behalf of its members. Web servers are **unable to learn the true source of a request** because it is **equally likely to have originated from any member of the crowd**, and even collaborating crowd members cannot distinguish the originator of a request from a member who is merely forwarding the request on behalf of another."*

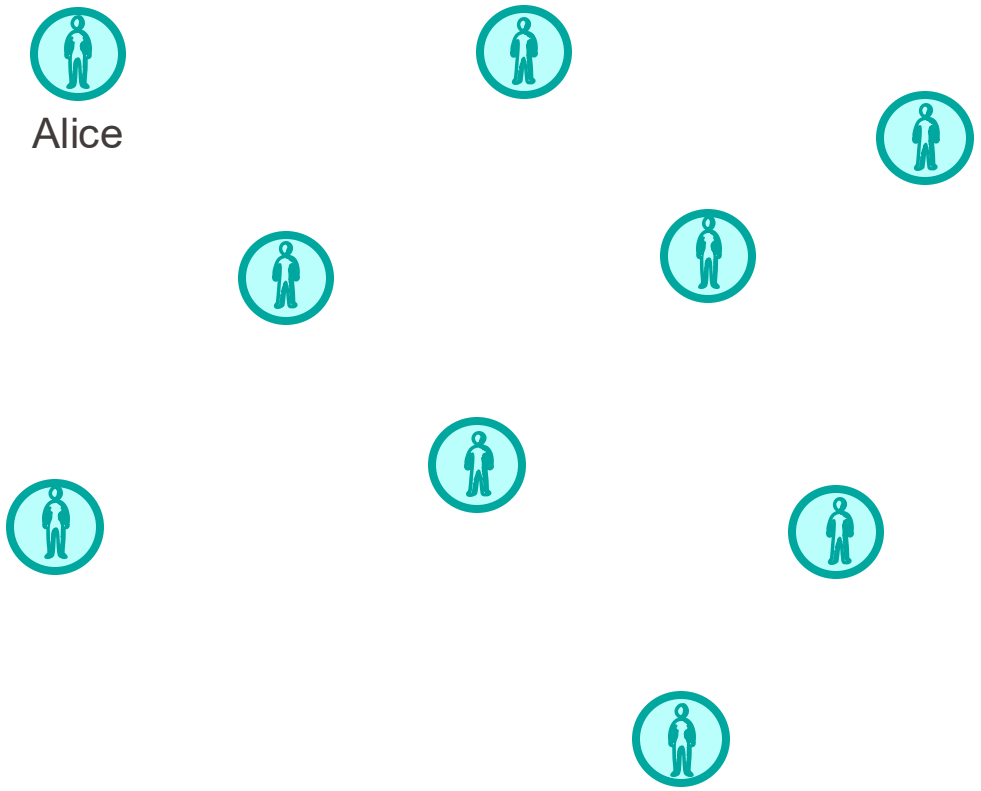
Reiter, M. K., & Rubin, A. D.

Crowds: Anonymity for web transactions.

ACM Transactions on Information and System Security, **1998**, 1(1), 66-92.

Crowds
Basic operation

N: Population size
p_f: Probability of forwarding

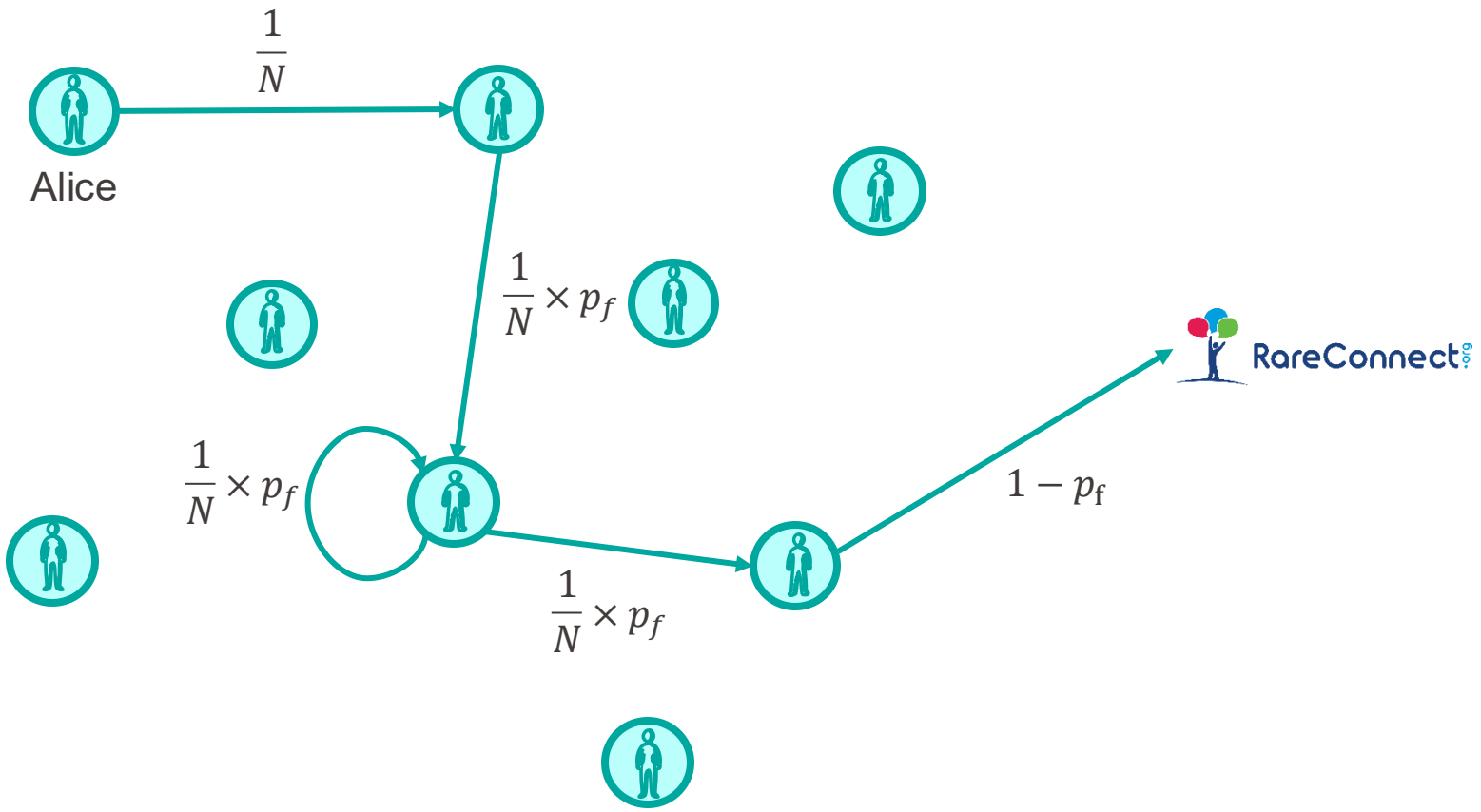


RareConnect

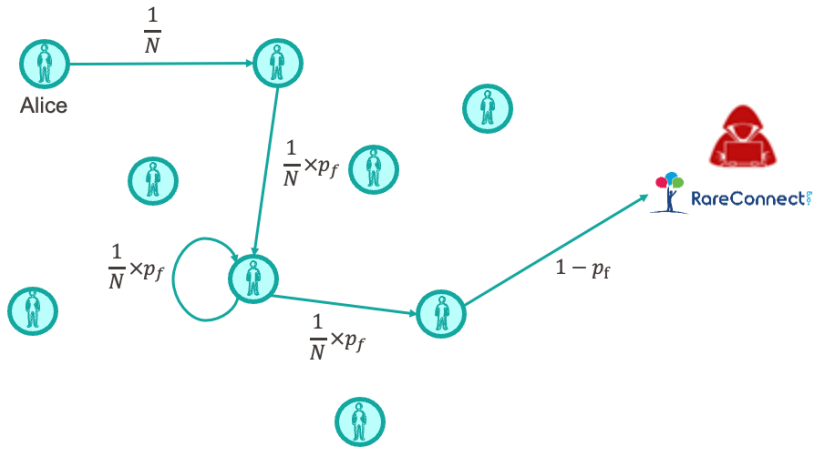
Crowds

Basic operation

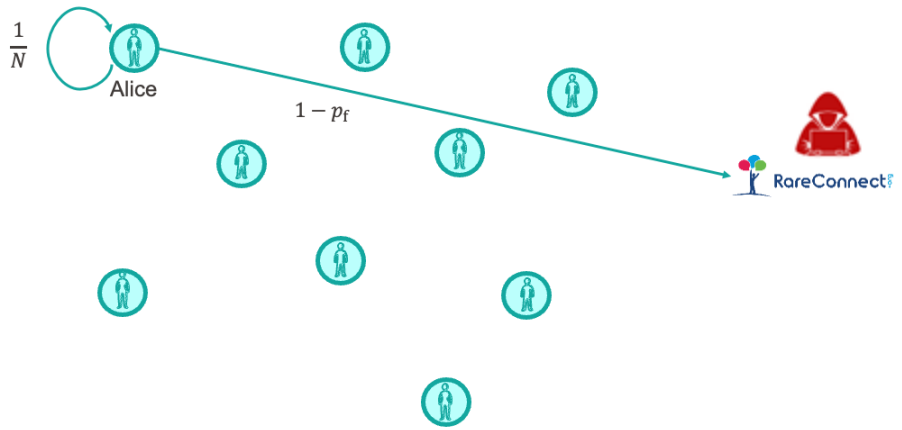
N: Population size
p_f: Probability of forwarding



Crowds
Anonymity analysis - Server



For RareConnect:
Indistinguishable!



MAX ANONYMITY!

$$\Pr[sender = \text{🧍}] = \frac{1}{N}, \forall \text{🧍}$$

$$H = -\sum \frac{1}{N} \log_2 \frac{1}{N} = \log_2 N$$

Crowds

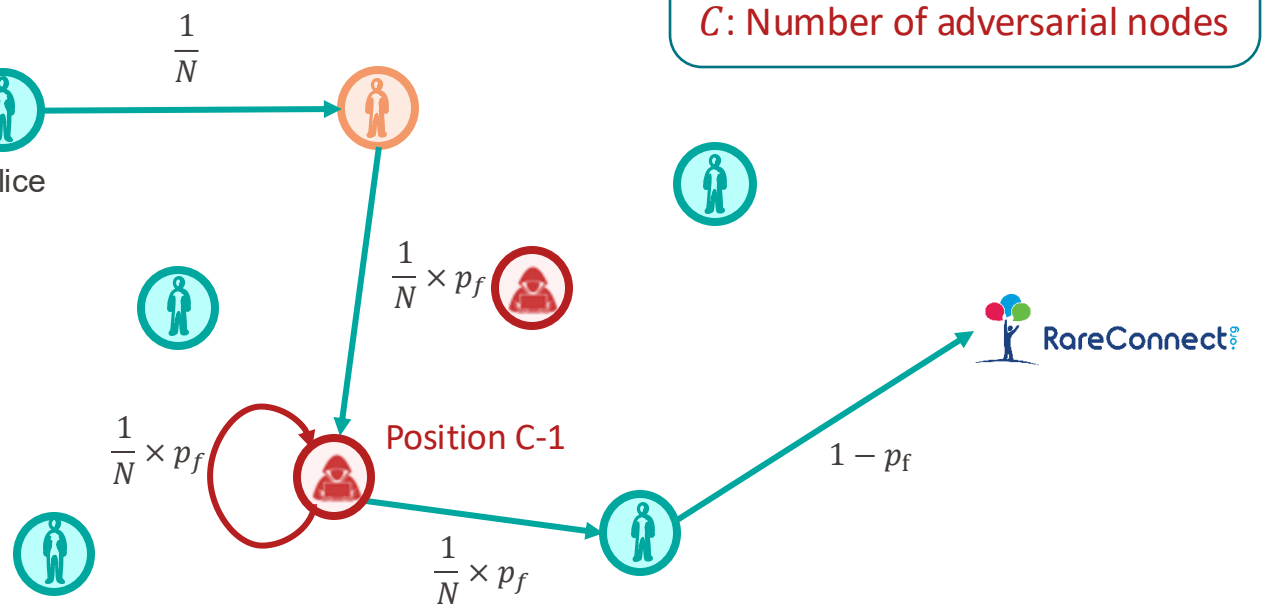
Anonymity analysis – Other users

N: Population size
p_f: Probability of forwarding
C: Number of adversarial nodes

$$Pr[sender = \text{honest}] = 1 - \frac{p_f(N - C - 1)}{N}$$

$$\Pr[sender = \text{adversary}] = 0$$

$$\Pr[sender = \text{honest}] = \frac{1 - \Pr[sender = \text{adversary}]}{N - C - 1} = \frac{p_f}{N}$$



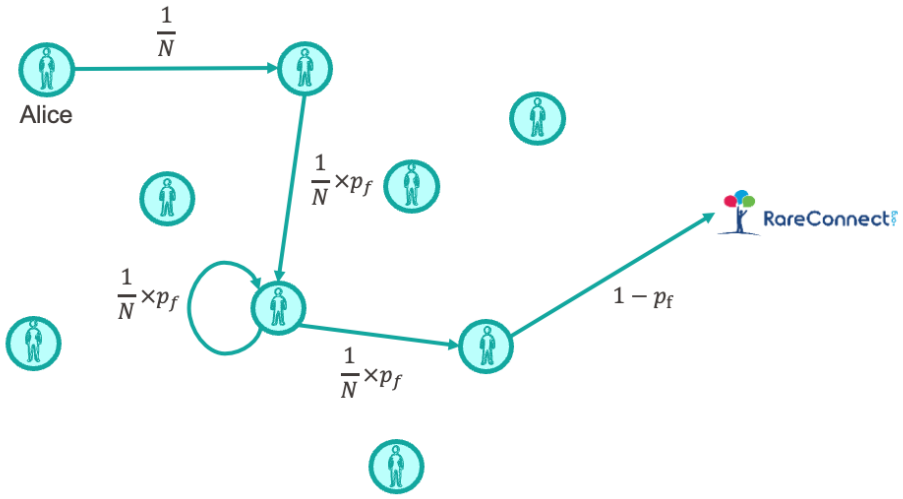
Measure of anonymity

$$H(X) = \frac{N - p_f(N - C - 1)}{N} \log_2 \left[\frac{N}{N - p_f(N - C - 1)} \right] + p_f \frac{N - C - 1}{N} \log_2 \left[\frac{N}{p_f} \right]$$

- *Towards measuring anonymity*. Claudia Diaz, Stefaan Seys, Joris Claessens and Bart Preneel. Designing Privacy Enhancing Technologies (PETS'02)

Crowds
Performance analysis

Original crowds: If good anonymity…
→ Long paths
→ High variance!



The Wisdom of Crowds. Attacks and Optimal Constructions. George Danezis, Claudia Díaz, Emilia Käsper, Carmela Troncoso. ESORICS 2009
J. P. Muñoz-Gea, J. Malgosa-Sanahuja, P. Manzanares-Lopez, J. C. Sanchez-Aarnoutse, and J. Garcia-Haro. 2008. A Low-Variance Random-Walk Procedure to Provide Anonymity in Overlay Networks. ESORICS 2008

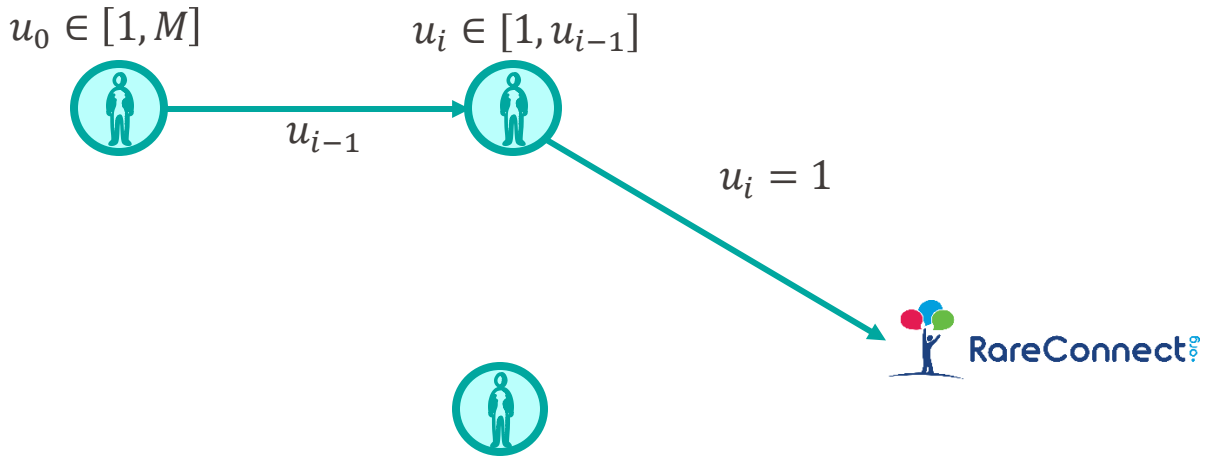
Crowds
Performance analysis

Original crowds: If good anonymity…
→ Long paths
→ High variance!

Always Down (AD): Upper bound
→ Reduced path length
→ Reduced variance

Always Up (AU): Lower bound
→ Reduced path length
→ Reduced variance

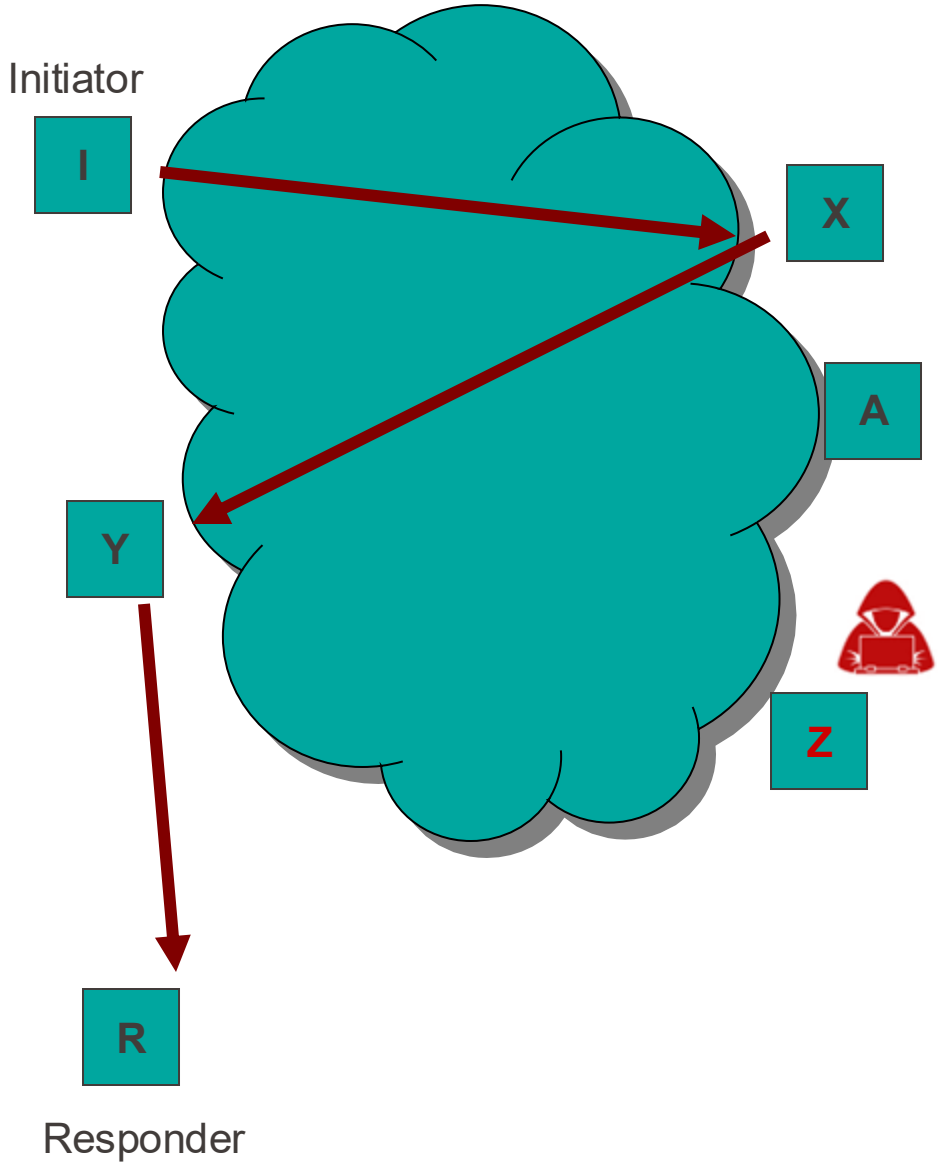
N: Population size
M: Interval size
$u_i \in [1, u_{i-1}]$: Chosen uniformly at random



The Wisdom of Crowds. Attacks and Optimal Constructions. George Danezis, Claudia Díaz, Emilia Käsper, Carmela Troncoso. ESORICS 2009
J. P. Muñoz-Gea, J. Malgosa-Sanahuja, P. Manzanares-Lopez, J. C. Sanchez-Aarnoutse, and J. Garcia-Haro. 2008. A Low-Variance Random-Walk Procedure to Provide Anonymity in Overlay Networks. ESORICS 2008

The predecessor attack

A general attack against sender anonymity for any communication system in which:



- *An Analysis of the Degradation of Anonymous Protocols*. Matthew Wright, Micah Adler, Brian Neil Levine, and Clay Shields. NDSS 2002

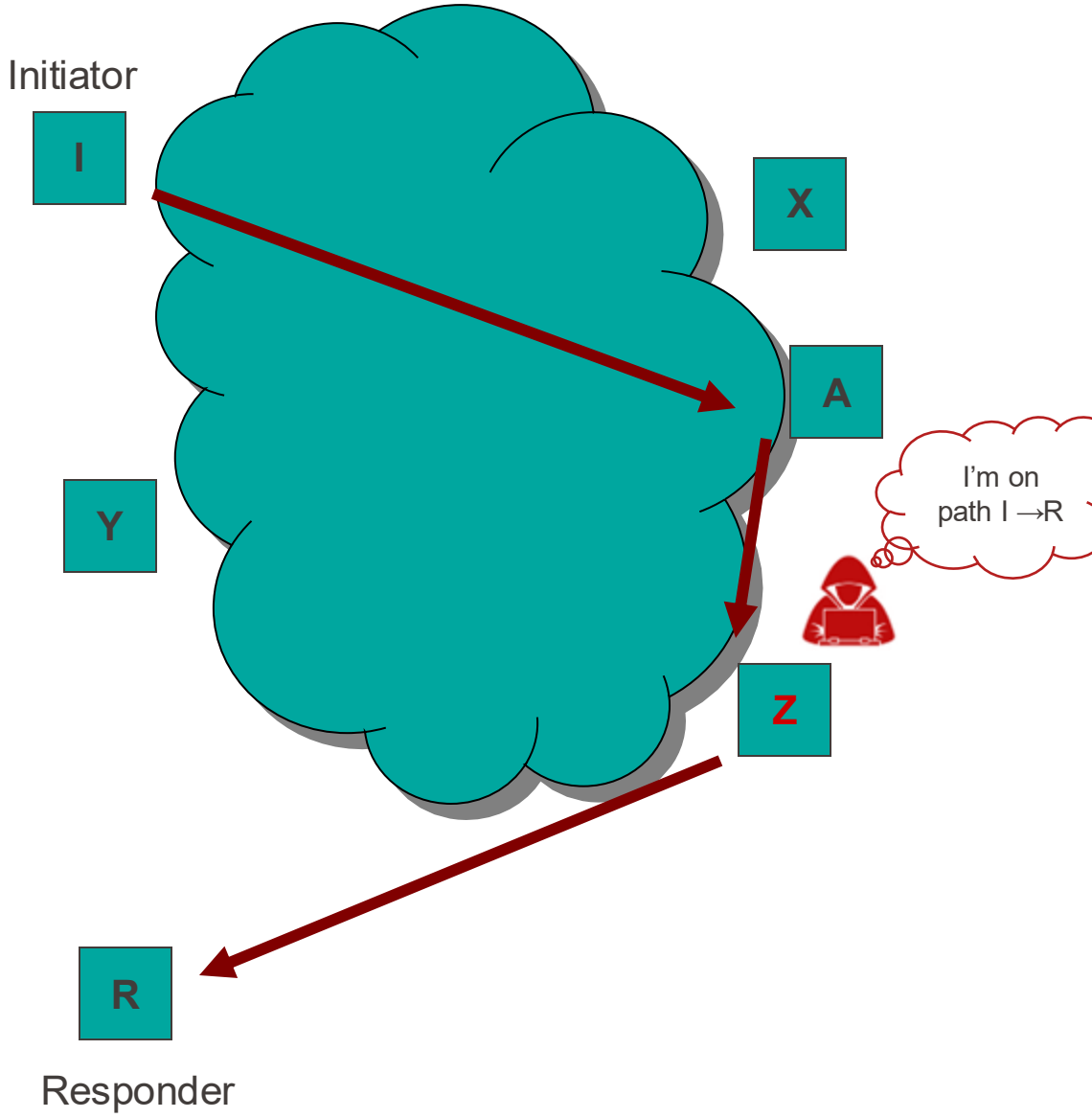
Slides by Matthew Wright

The predecessor attack

A general attack against sender anonymity for any communication system in which:

- Paths vary

- Attacker can observe session-identifying info
  - Responder's IP address
  - Cookie, login name, specific content
  - …

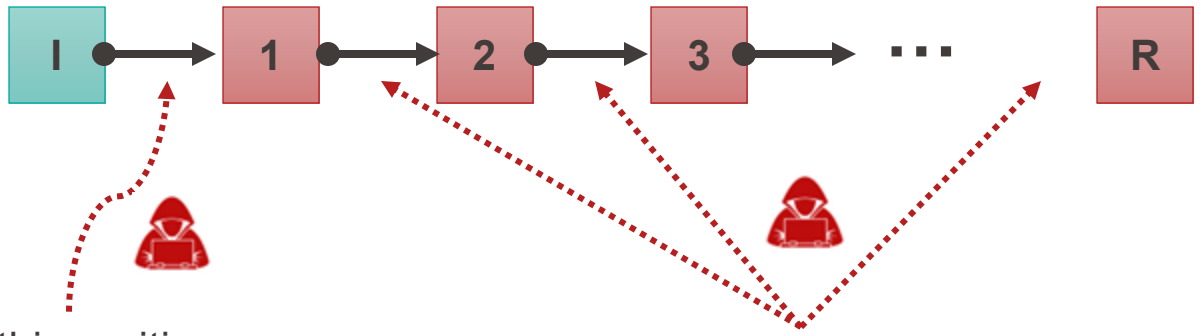


- *An Analysis of the Degradation of Anonymous Protocols*. Matthew Wright, Micah Adler, Brian Neil Levine, and Clay Shields. NDSS 2002

Slides by Matthew Wright

The predecessor attack

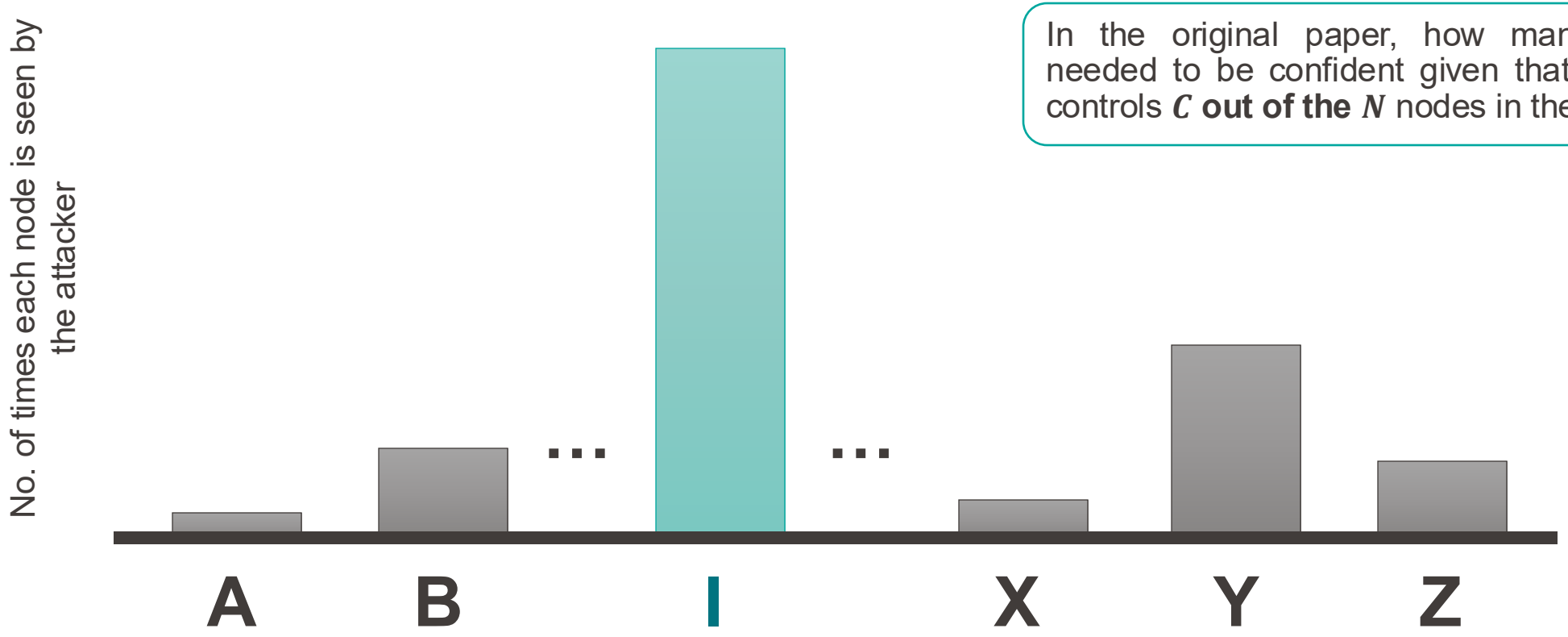
ATTACK IDEA: Log the node before the attacker



node	count
I	41
X	18
Y	24
Z	17

The predecessor attack

Attackers view after many rounds (path reformations)

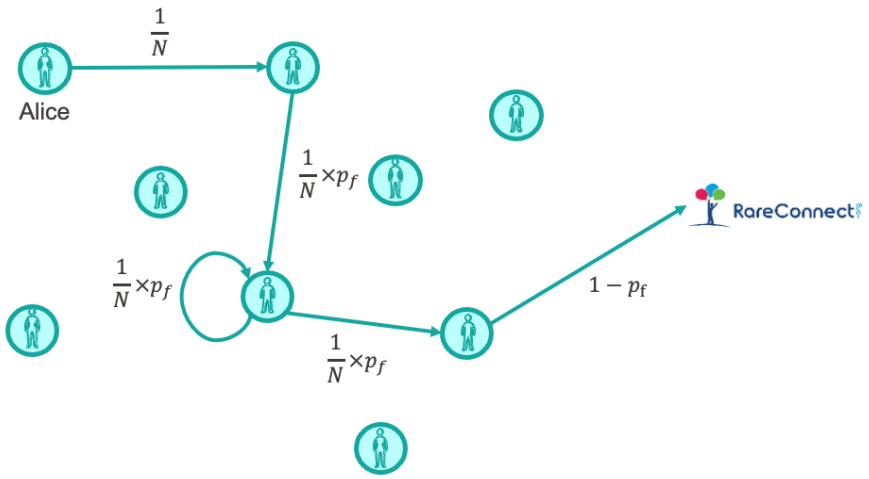


In the original paper, how many rounds are needed to be confident given that the adversary controls C **out of the** N nodes in the Crowd

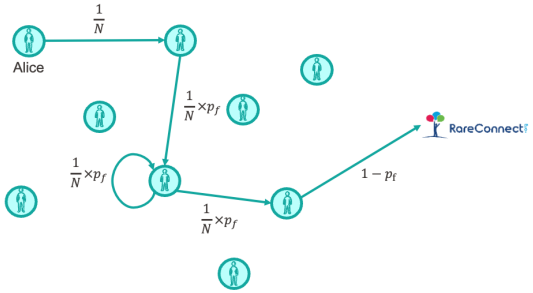
The predecessor attack
Importance

The attack applies to **any protocol** for anonymity, provided that:

- Paths of proxies change
- (Uniformly) random selection of paths
- There exists a position where attackers can:
  - See the initiator send messages in the session
  - Determine some session information



Crowds
Summary



Goal: Sender anonymity

Infrastructure: who routes messages

User-based:
nodes = users (peer to peer) =

User-independent:
nodes = others, not (necessarily) trusted

Hybrid:
nodes = mix users and others

Adversarial capabilities:

Global *vs.* *partial*:
What can the adversary see?

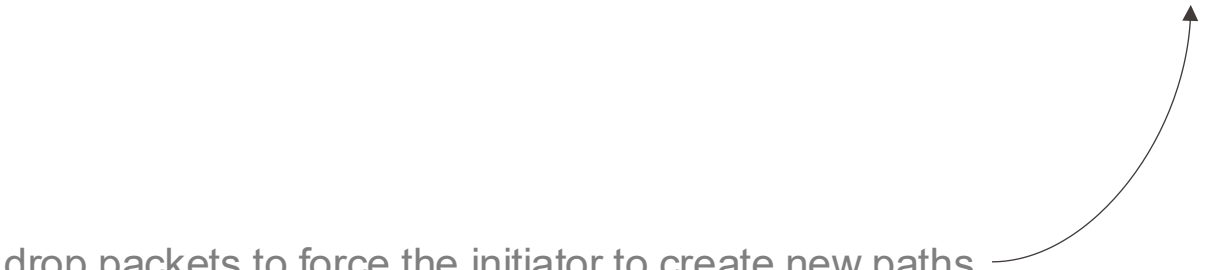
Active *vs. passive*:
What can the adversary do?

Internal *vs. external:*
Can the adversary see inside?

Crowds
Summary

Simple scheme, good performance but…

- Only provides sender anonymity
- Information is sent around in the clear!
- Predecessor attack is a killer in presence of active adversaries



- Borisov et al., Denial of Service or Denial of Security? How Attacks on Reliability can Compromise Anonymity, CCS'07

Mix networks

THE PROBLEM

How to **send messages** anonymously

Sender/receiver anonymity (depends on the configuration)

"A technique based on public key cryptography is presented that allows an electronic mail system to hide who a participant communicates with as well as the content of the communication - in spite of an unsecured underlying telecommunication system."

David Chaum

Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms.

Communications of the ACM, **1981**, 24(2), 84-90.

Mix networks

THE PROBLEM **Not streams!**

How to **send** **messages** anonymously

Sender/receiver anonymity (depends on the configuration)

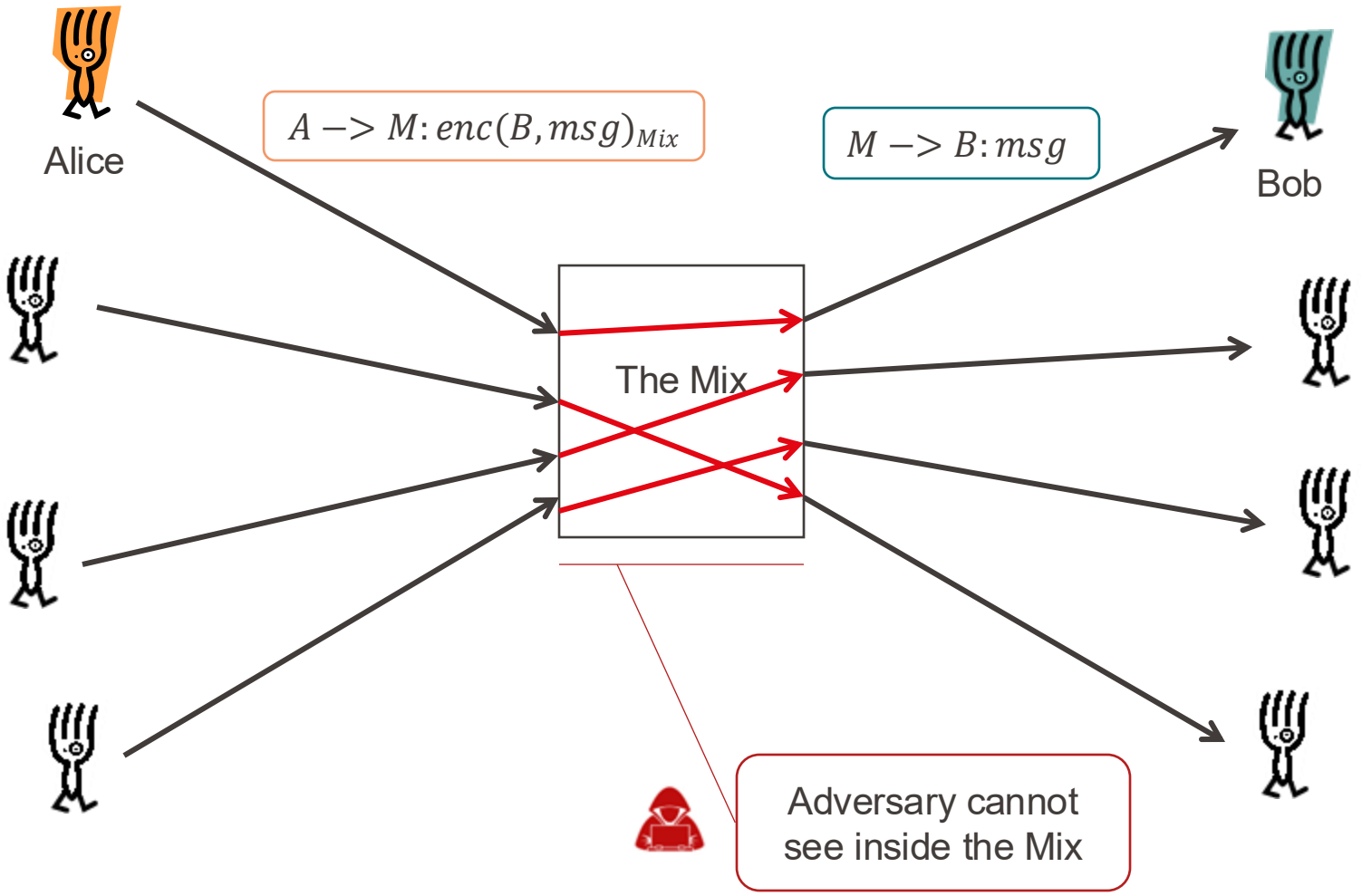
"A technique based on public key cryptography is presented that allows an electronic mail system to hide who a participant communicates with as well as the content of the communication - in spite of an unsecured underlying telecommunication system."

David Chaum

Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms.

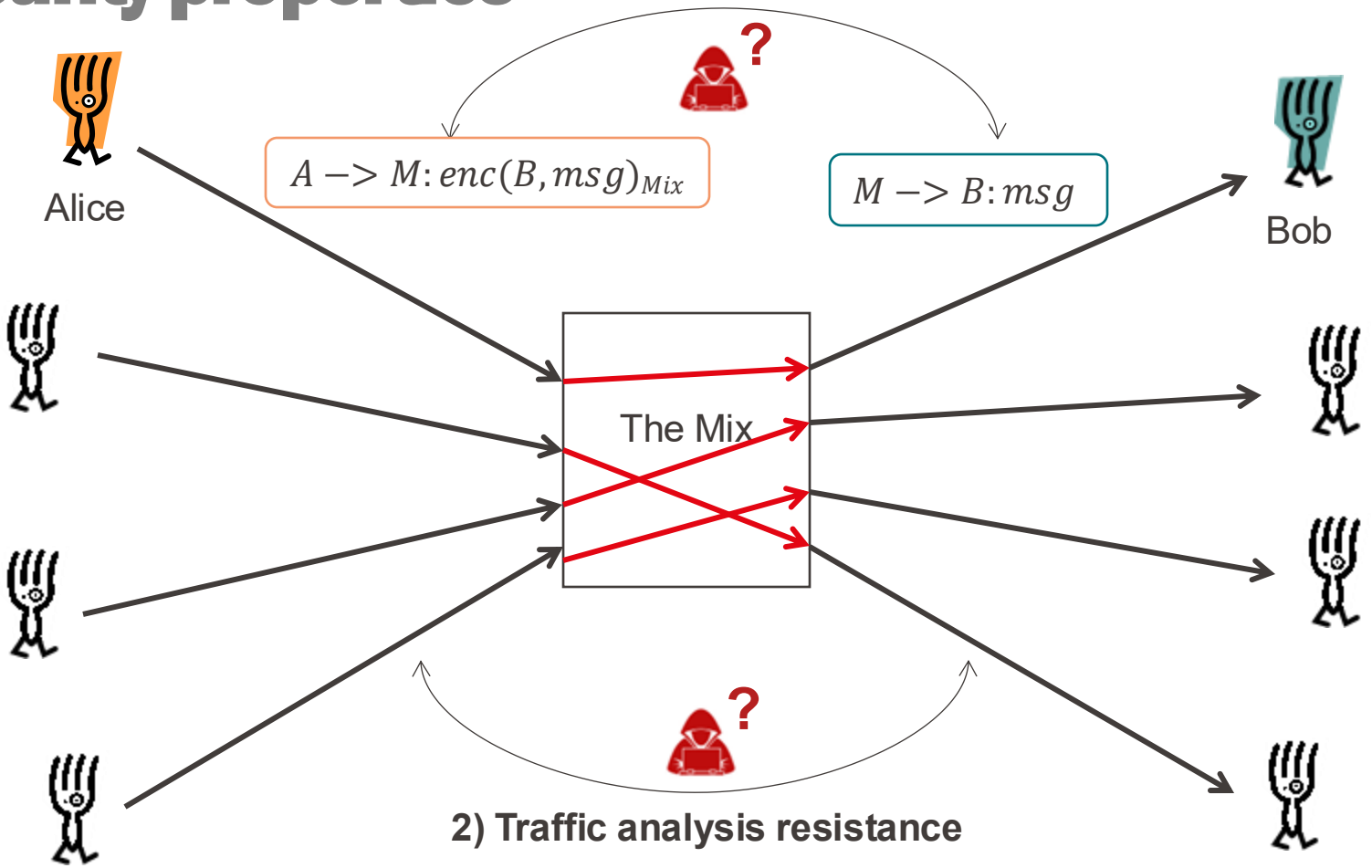
Communications of the ACM, **1981**, 24(2), 84-90.

The mix Illustrated



The mix
Security properties

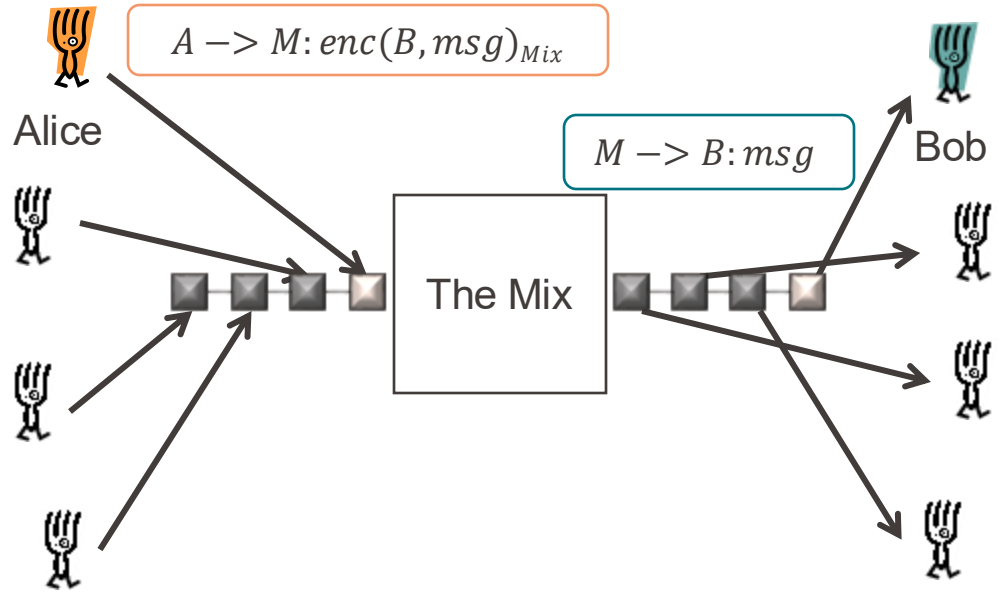
Ensure adversary cannot link messages in and out of the mix from their bit pattern
→Cryptographic problem



Ensure the messages in and out of the mix cannot be linked using any metadata (timing, ...)
→ Two tools: **delay** or **inject** traffic – both add cost

A broken mix design
The 'FIFO' mix

Broken traffic analysis resistance: Mix sends messages out in the order they came in!



Passive Attack

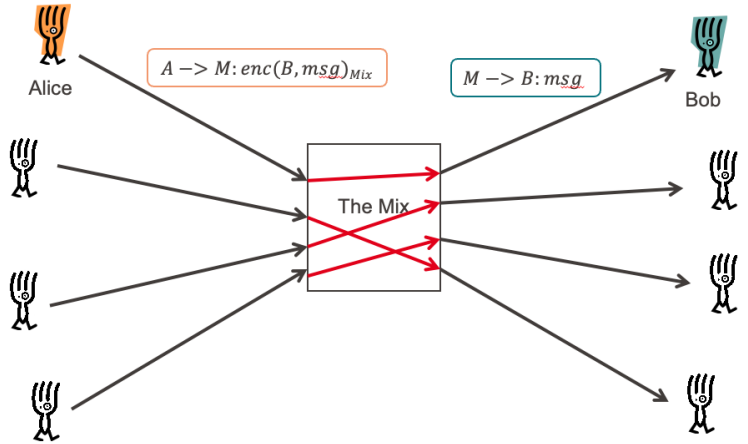
The adversary simply logs the number of messages, and assigns to each input the corresponding output.

Lessons from the FIFO mix

Mix strategies must actually ‘mix’ messages together

Parameters

- **FIRING CONDITION**: when does the mix fire (e.g., base on number of messages, based on time)
- **BATCHING STRATEGY**: which establish how the messages are released. Typically, the storage of messages is called pool. Messages can be released all at the same time, or some can stay in the mix. The batching strategy determines which message will leave.



Lessons from the FIFO mix

Mix strategies must actually 'mix' messages together

Parameters

- **FIRING CONDITION**: when does the mix fire (e.g., base on number of messages, based on time)
- **BATCHING STRATEGY**: which establish how the messages are released. Typically, the storage of messages is called pool. Messages can be released all at the same time, or some can stay in the mix. The batching strategy determines which message will leave.

"Hell is other people" – J.P. Sartre

Anonymity security relies on *others*

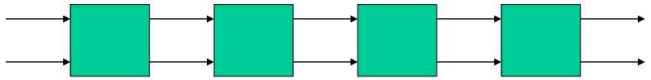
Problem 1: Mix must be honest

Problem 2: Other sender-receiver pairs to hide amongst

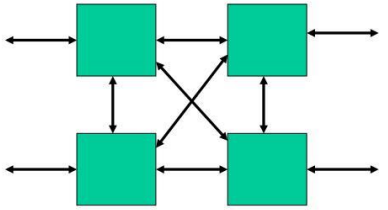
Solve problem 1: Distribute mixing!

- Rely on more mixes – good idea
  - Distribute trust – some could be dishonest
  - Distribute load – fewer messages per mix
- Two extremes
  - Mix Cascades
    - All messages are routed through a preset mix sequence
    - Good for anonymity – poor load balancing
  - Free routing
    - Each message is routed through a random sequence of mixes
    - Security parameter L: length of the sequence

Cascade topology



Free route topology



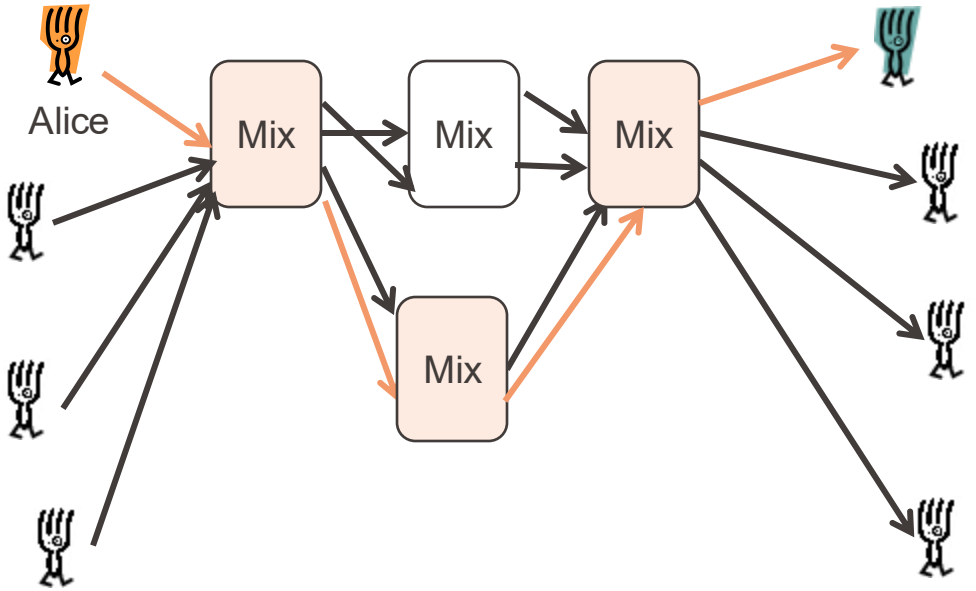
Panel discussion: Mix Cascades vs. P2P: Is one concept superior?. PET 2004

Solve problem 1: Distribute mixing!
… but distributing is hard! Epistemic attacks

Problem: all clients need to use the same information to construct paths through relays.

Otherwise: attacks based on clients' knowledge (epistemic)

- Consider a user only knows a random subset of mix nodes …
- If paths identify clients: then anonymity is not protected.



George Danezis, Paul F. Syverson: Bridging and Fingerprinting: Epistemic Attacks on Route Selection. Privacy Enhancing Technologies 2008: 151-166
George Danezis, Richard Clayton: Route Fingerprinting in Anonymous Communications. Peer-to-Peer Computing 2006: 69-72

Solve problem 1: Distribute mixing!
… but distributing is hard! Epistemic attacks

Problem: all clients need to use the same information to construct paths through relays.

Otherwise: attacks based on clients' knowledge (epistemic)

- Consider a user only knows a random subset of mix nodes …
- If paths identify clients: then anonymity is not protected.

Defences against epistemic attacks

Option 1: Download the whole database of routers and routing information
(Bandwidth cost)

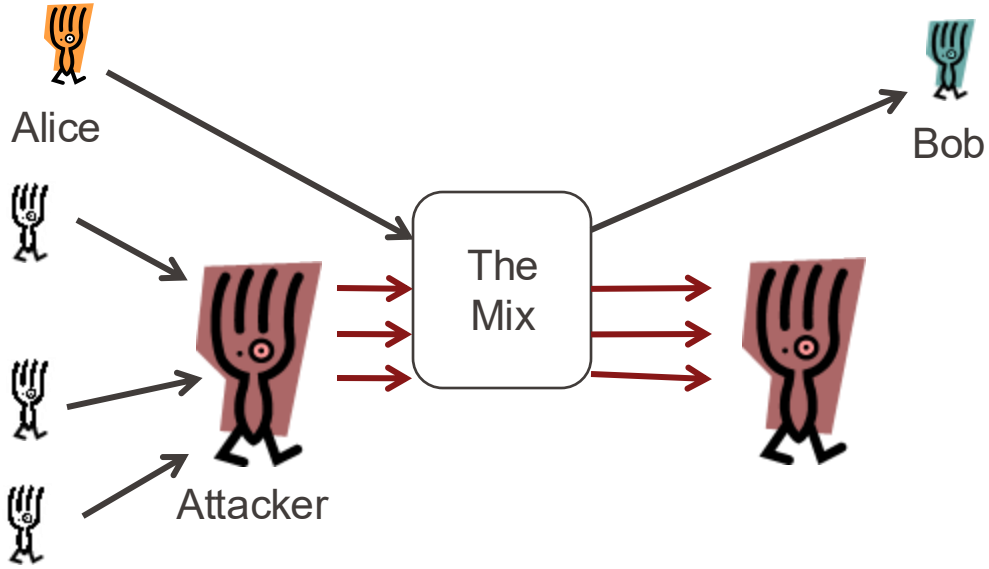
Option 2: Privately download parts of it (Private Information Retrieval)
(Computationally expensive!)

George Danezis, Paul F. Syverson: Bridging and Fingerprinting: Epistemic Attacks on Route Selection. Privacy Enhancing Technologies 2008: 151-166
George Danezis, Richard Clayton: Route Fingerprinting in Anonymous Communications. Peer-to-Peer Computing 2006: 69-72

Solve problem 2: Are there others to mix with?

The (n-1) attack (active adversary)

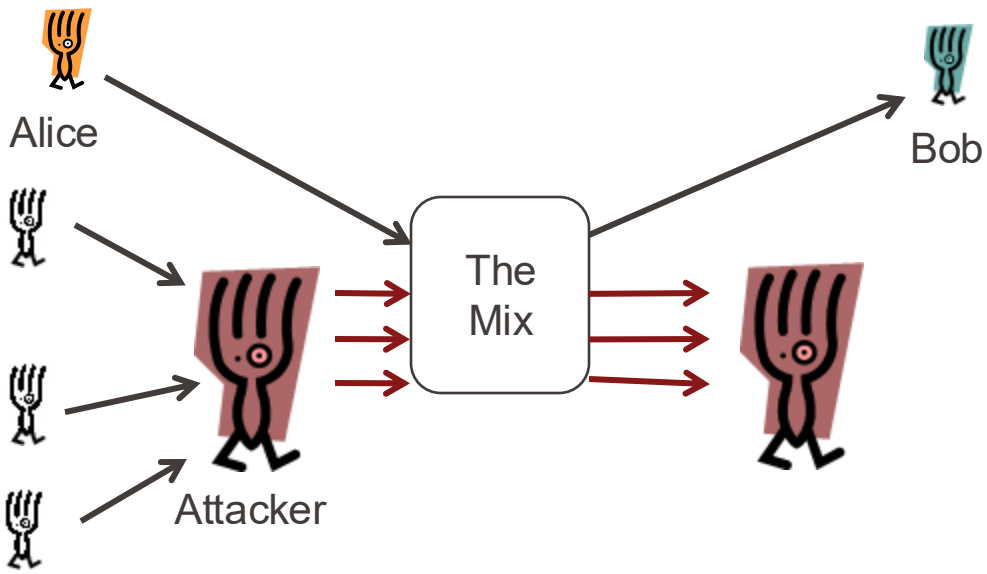
- Wait or flush the mix.
- Block all incoming messages (trickle) and injects own messages (flood) until Alice's message is out.



Solve problem 2: Are there others to mix with?

The (n-1) attack (active adversary)

- Wait or flush the mix.
- Block all incoming messages (trickle) and injects own messages (flood) until Alice's message is out.



Defences against (n-1) attacks

Strong identification to ensure distinct identities

Difficult to adopt

Message expiry

Messages are discarded after a deadline

Prevents the adversary from flushing / injecting

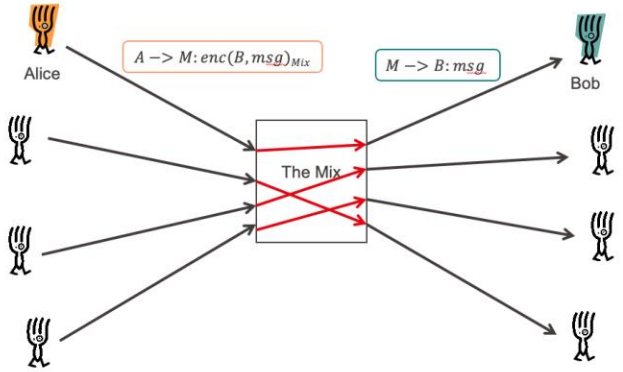
Heartbeat traffic

Mixes route messages in a loop back to themselves

Detect whether an adversary is blocking messages

Forces adversary to subvert everyone, all the time

Mix nets
Summary



Goal: Sender/receiver/3rd party anonymity

Infrastructure: who routes messages

User-based:
nodes = users (peer to peer)

User-independent:
nodes = others, not (necessarily) trusted

Hybrid:
nodes = mix users and others

Adversarial capabilities:

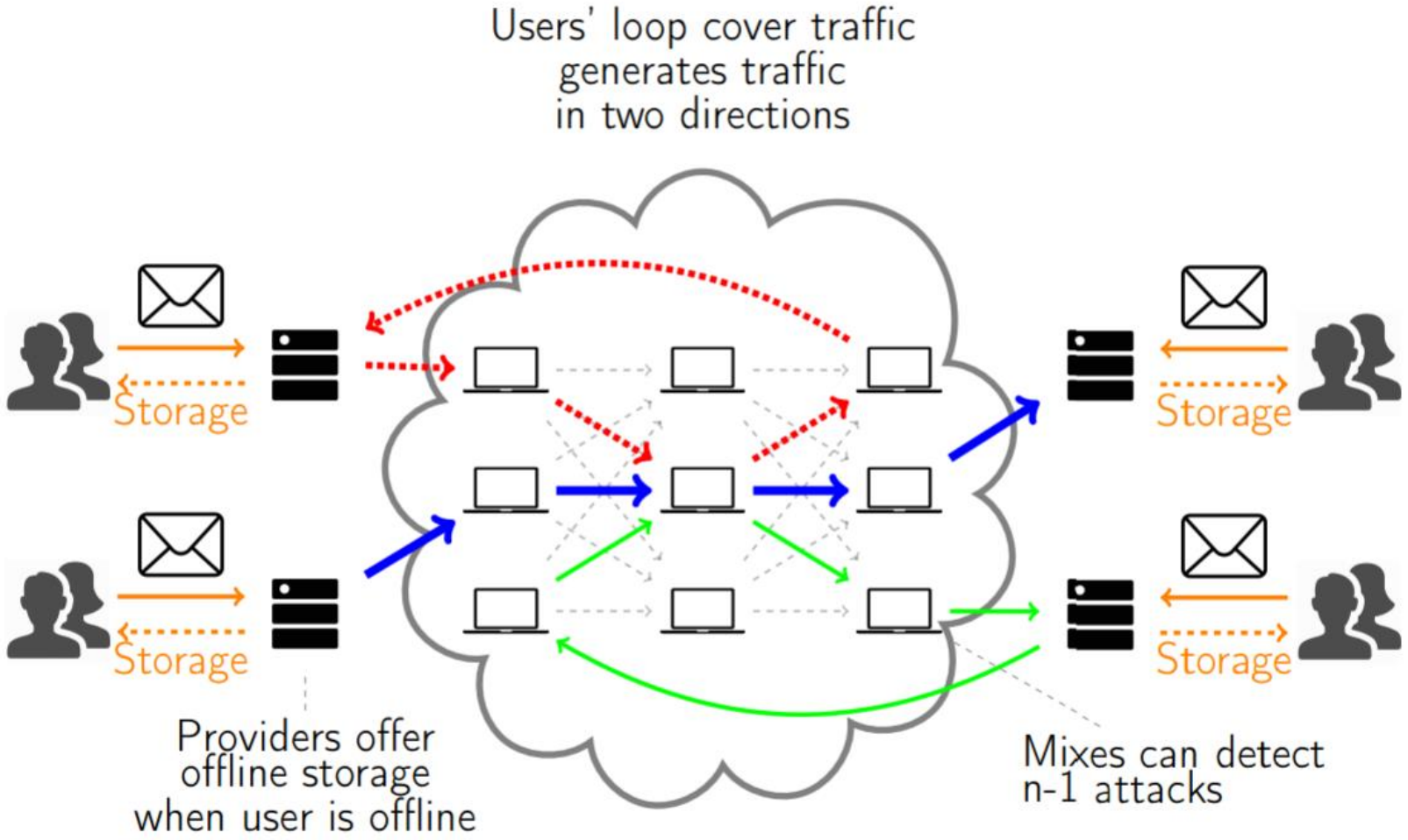
Global *vs. partial*:
What can the adversary see?

Active *vs. passive*:
What can the adversary do?

Internal *vs. external:*
Can the adversary see inside?

A modern mix net: Loopix

https://nymtech.net/



- Ania Piotrowska, Jamie Hayes, Tariq Elahi, Sebastian Meiser, **George Danezis**: The Loopix Anonymity System. Usenix Security 2017

Onion routing

THE PROBLEM

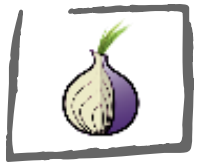
How to **send streams** anonymously

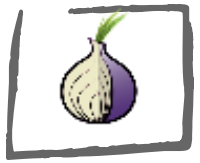
Sender/receiver anonymity (depends on the configuration)

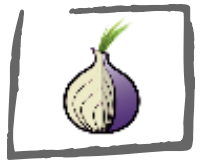
"Tor works on the real-world Internet, requires no special privileges or kernel modifications, requires little synchronization or coordination between nodes, and provides a reasonable tradeoff between anonymity, usability, and efficiency."

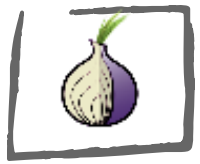
Syverson, P., Dingledine, R., & Mathewson, N.
Tor: The second generation onion router
Usenix Security, 2004

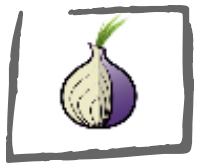
Basic operation

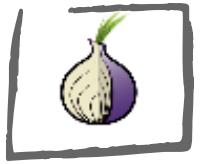




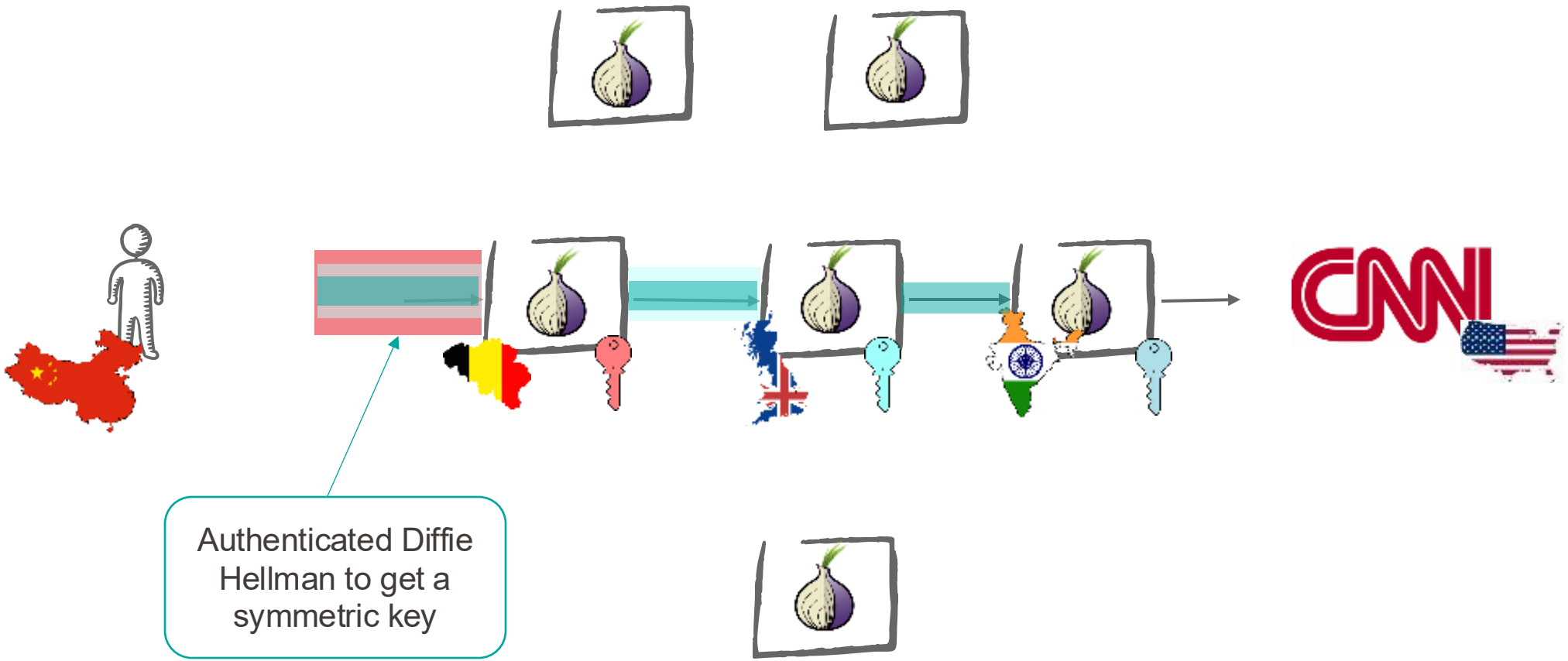




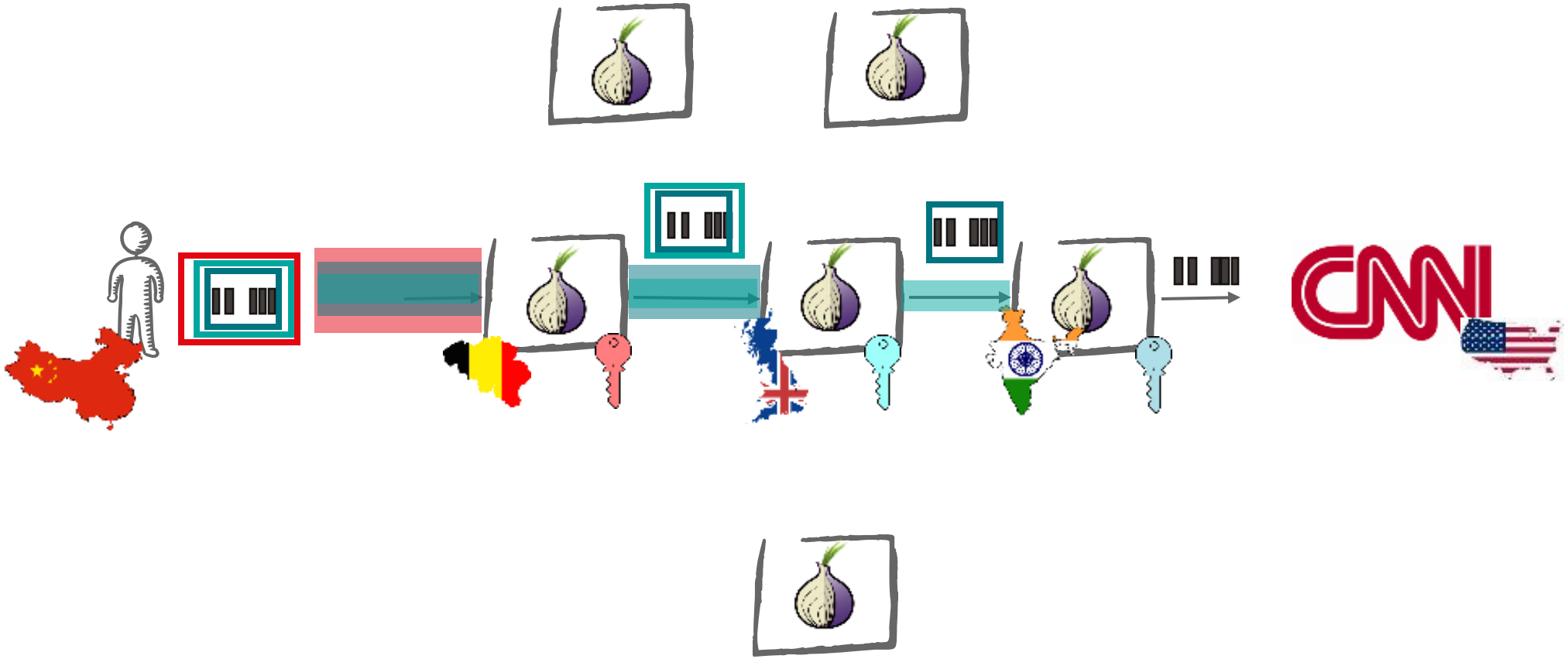




Basic operation

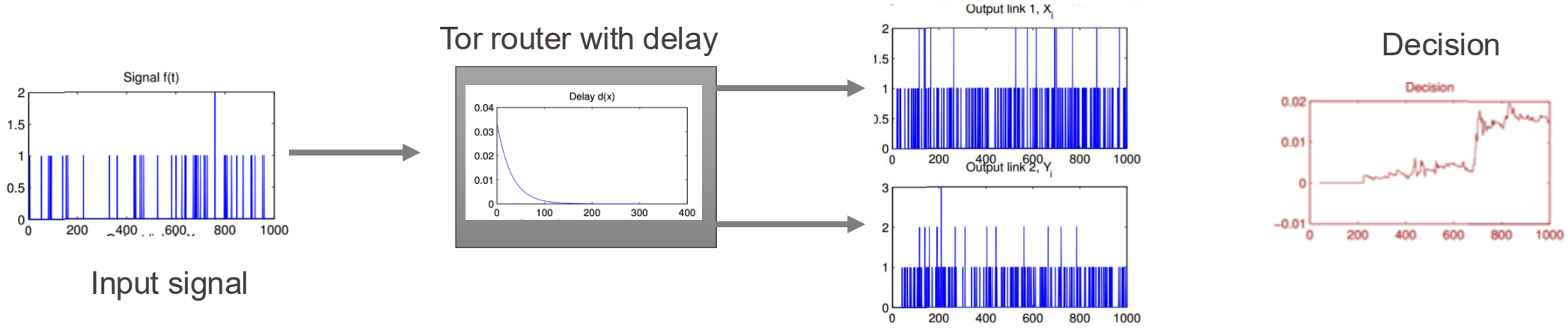


Basic operation



Tor problems
Stream tracing attacks

- Stream tracing attacks allow an adversary to link two points of an anonymous circuit.
- How? Make a model template of output from input, and match.



George Danezis: The Traffic Analysis of Continuous-Time Mixes. Privacy Enhancing Technologies 2004: 35-50

Tor problems
Website fingerprinting

- Tor does not significantly disrupt the timing, volume and dynamics of web browsing streams.
- Website fingerprinting uses machine learning to guess which web page is being loaded through Tor.
- It works well, **even against delaying, cover and other defences**.

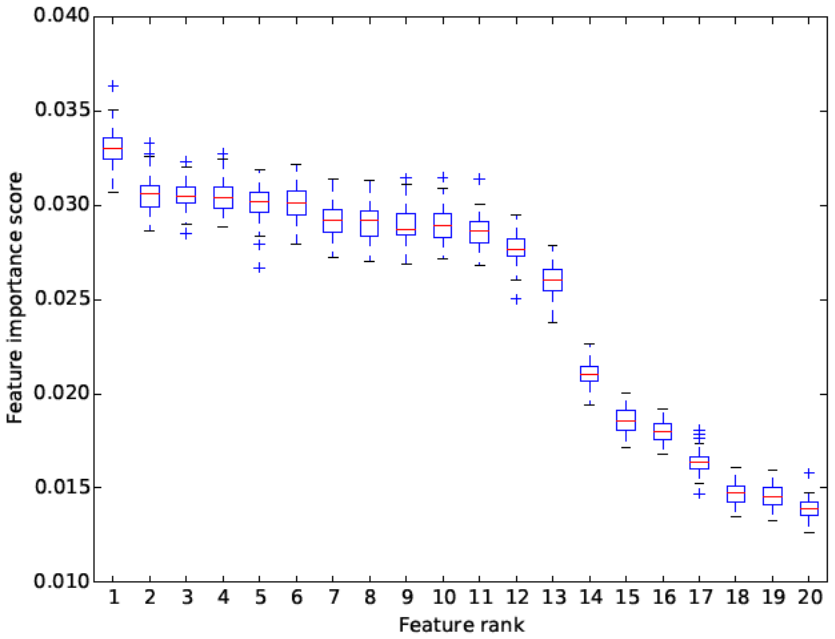
Defenses	This work	*k*-NN [39]	CUMUL [28]	Bandwidth overhead (%)
No defense	0.91 ± 0.01	0.91 ± 0.03	0.91 ± 0.04	0
Morphing [40]	**0.90** ± 0.03	0.82 ± 0.06	0.75 ± 0.07	50 ± 10
Decoy pages [27]	**0.37** ± 0.01	0.30 ± 0.06	0.21 ± 0.02	130 ± 20
Adaptive Padding [31]	**0.30** ± 0.04	0.19 ± 0.03	0.16 ± 0.03	54
BuFLO [12]	**0.21** ± 0.02	0.10 ± 0.03	0.08 ± 0.03	190 ± 20
Tamaraw [35]	**0.10** ± 0.01	0.09 ± 0.02	0.08 ± 0.03	96 ± 9

- Note: same attacks also work great against TLS/SSL!

Tor problems

Why is fingerprinting possible?

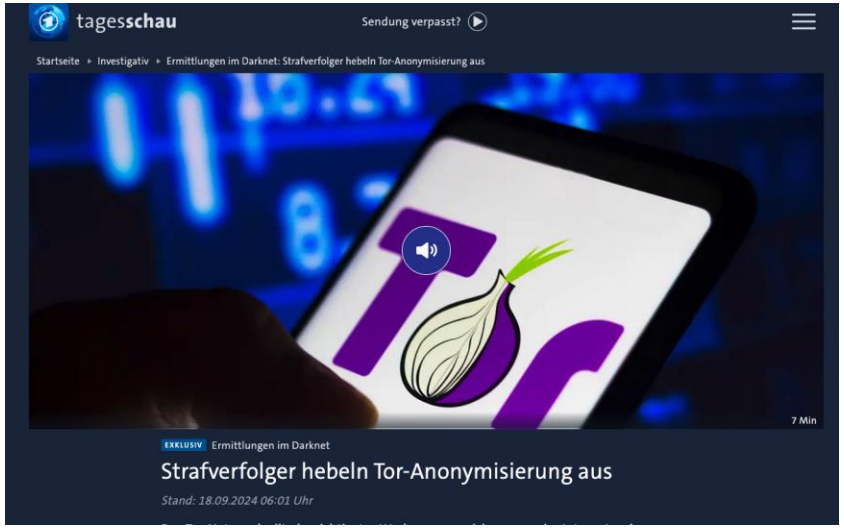
Random forest classifier allows for feature importance analysis.



№	Feature Description
1.	Number of incoming packets.
2.	Number of outgoing packets as a fraction of the total number of packets.
3.	Number of incoming packets as a fraction of the total number of packets.
4.	Standard deviation of the outgoing packet ordering list.
5.	Number of outgoing packets.
6.	Sum of all items in the alternative concentration feature list.
7.	Average of the outgoing packet ordering list.
8.	Sum of incoming, outgoing and total number of packets.
9.	Sum of alternative number packets per second.
10.	Total number of packets.
11-18.	Packet concentration and ordering features list.
19.	The total number of incoming packets stats in first 30 packets.
20.	The total number of outgoing packets stats in first 30 packets.

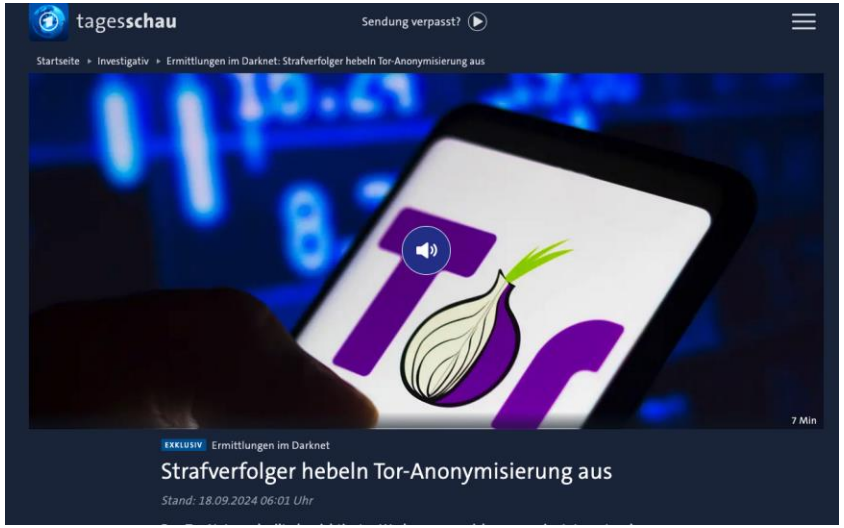
More Tor problems…

- Traffic analysis:
  - Sampling some traffic can suffice to conduct traffic analysis
    - Internet exchanges or autonomous systems may see enough
  - BGP rerouting attacks enable further attacks
  - + Many mix net attacks: DoS & epistemic attacks



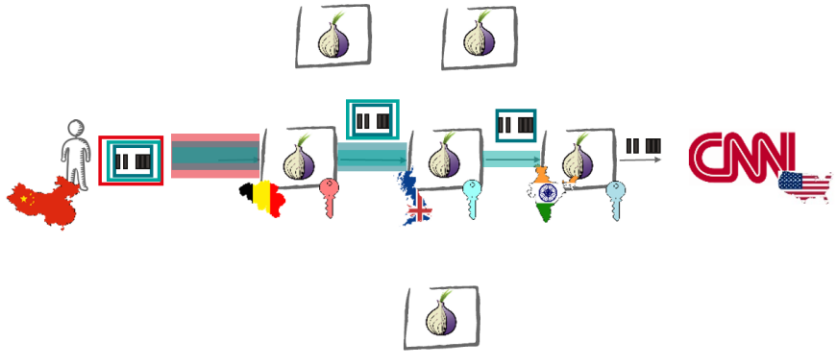
Too much and too little at the same time

- Tor is both too much and too little:
  - **Too little:** real adversaries can gain near GPA capabilities, or enough to break Tor. The Snowden revelations confirm this.
  - **Too much**: if it is trivial to link two points simpler design is possible:
    (1) No need for multiple layers of encryption.
    (2) A single hop security is all you get after a long time.



→ Tor is great if you want to hide from a relatively weak adversary.
Not so great against more powerful adversaries…

Onion routing
Summary



Goal: Sender/receiver/3rd party anonymity

Infrastructure: who routes messages

User-based:
nodes = users (peer to peer) =

User-independent:
nodes = others, not (necessarily) trusted

Hybrid:
nodes = mix users and others

Adversarial capabilities:

Global *vs. partial*:
What can the adversary see?

Active *vs. passive*:
What can the adversary do?

Internal *vs. external:*
Can the adversary see inside?

Take aways

Take aways

- Privacy is much more than confidentiality of contents
- Electronic communications do not happen in a void
  - Adversaries can observe a lot of information and infer more through traffic analysis
- Four paradigmatic examples of anonymous communication systems with varying goals/properties
  - But all use **re-routing** and **encryption** (except Crowds – and that didn't go so well…)
- Attacks on anonymity networks: think about all of them when you design the next one!