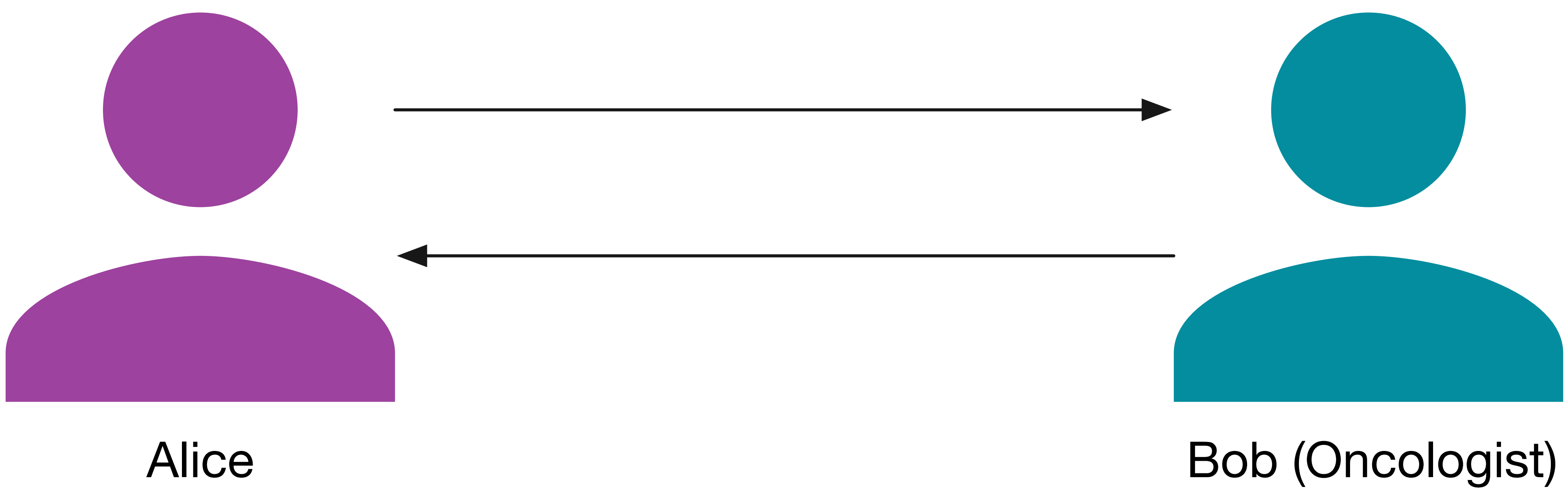
Vuvuzela

a scalable private messaging system

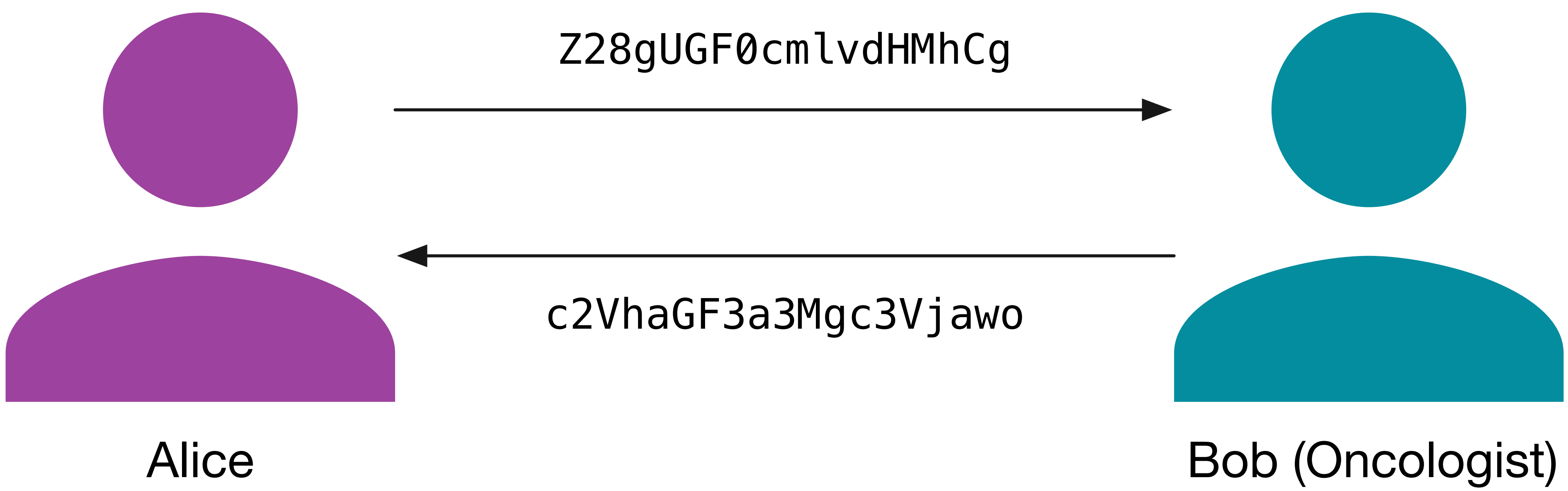
David Lazar

Jelle van den Hooff, Matei Zaharia, Nickolai Zeldovich

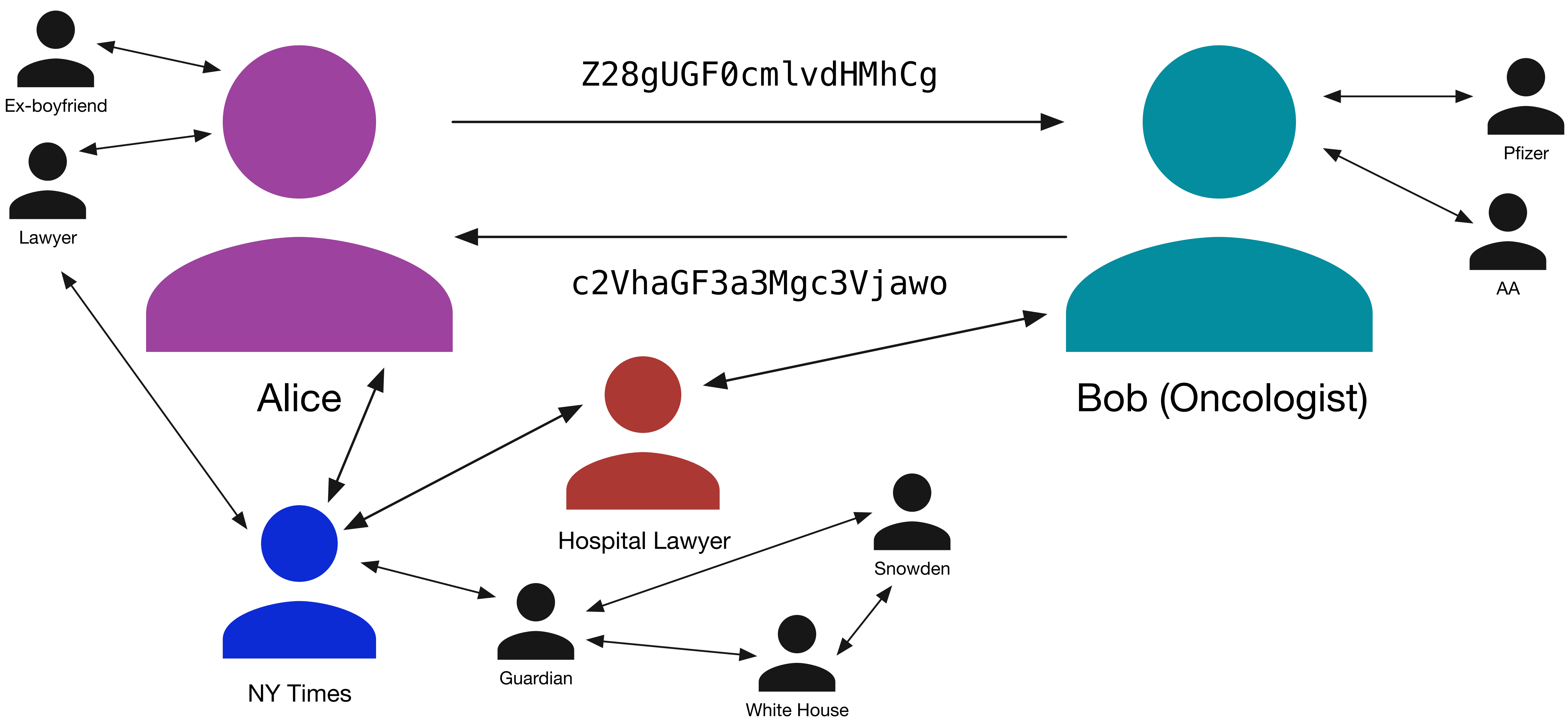
Motivation



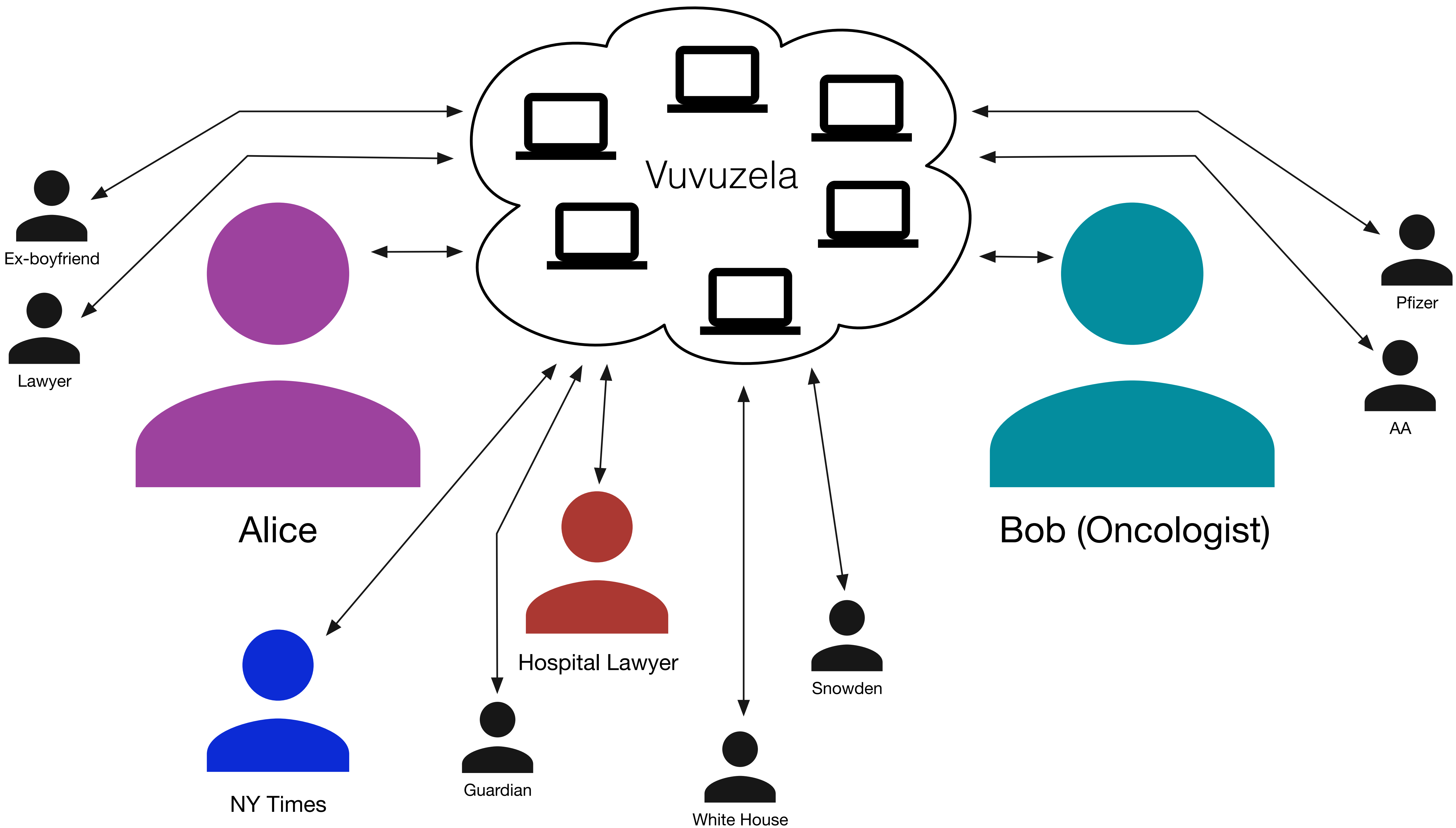
Encryption


Z28gUGF0cmlvdHMhCg
c2VhaGF3a3Mgc3Vjawo
Alice
Bob (Oncologist)

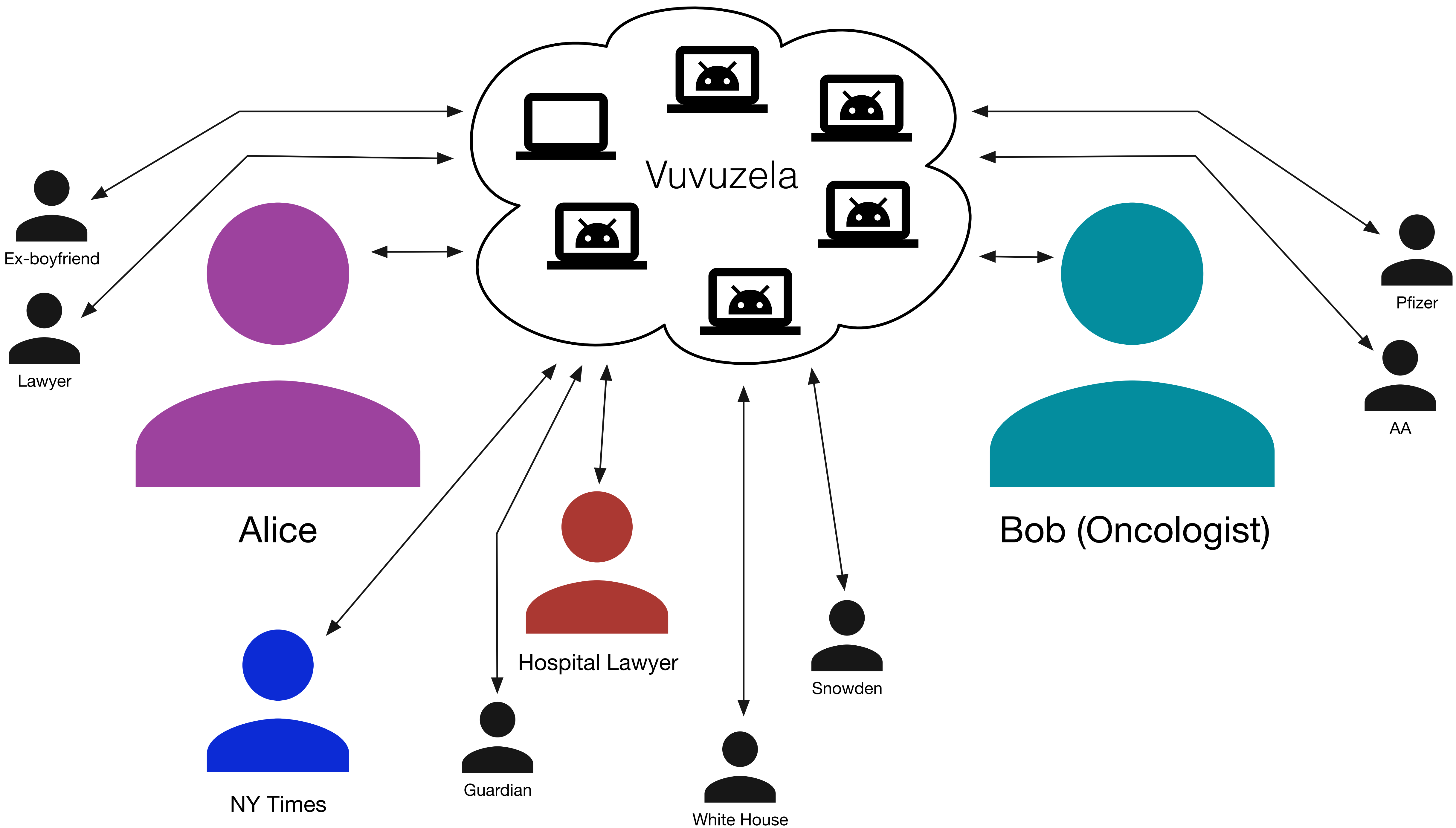
Problem: metadata



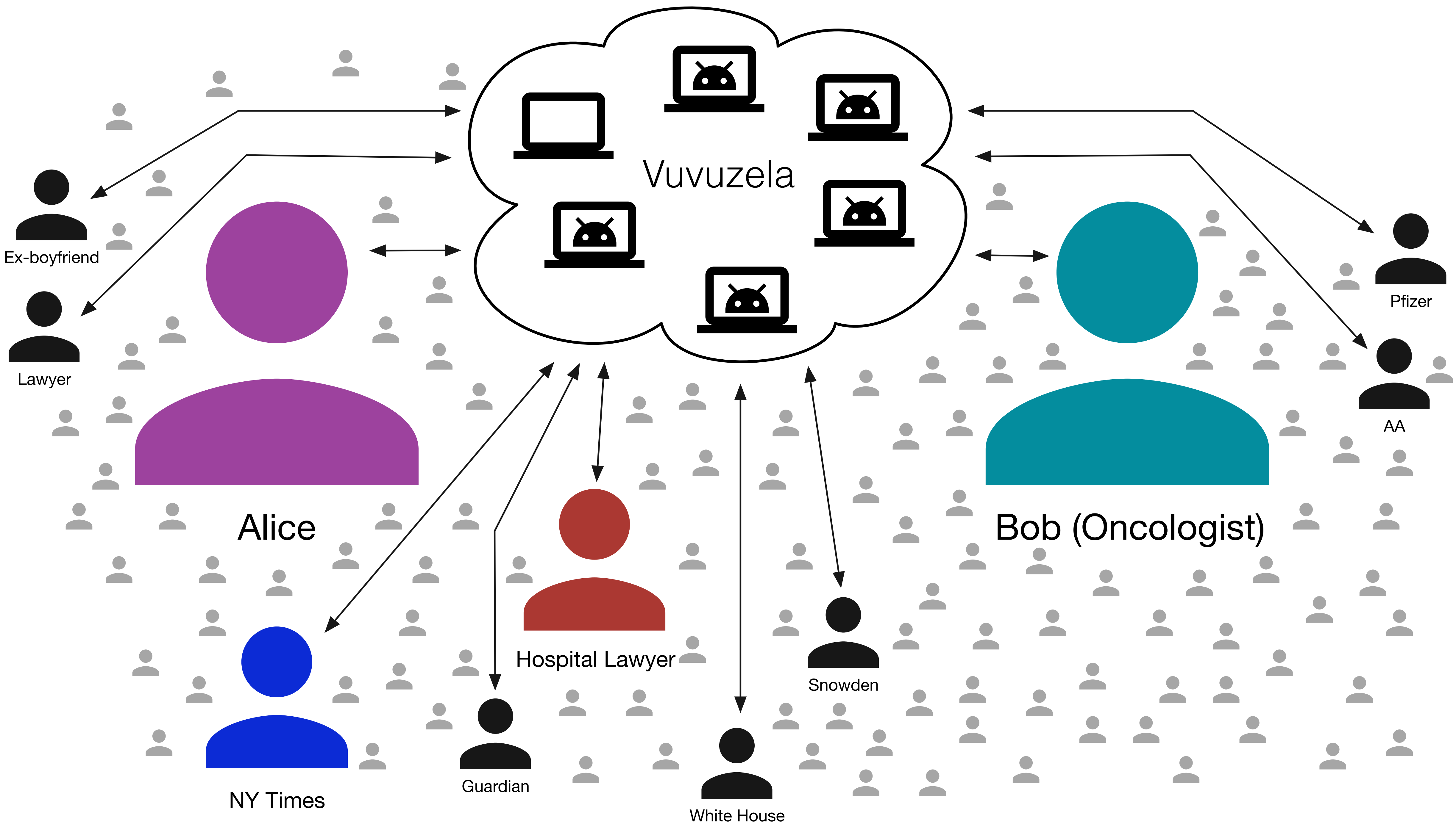
Goal: hide metadata



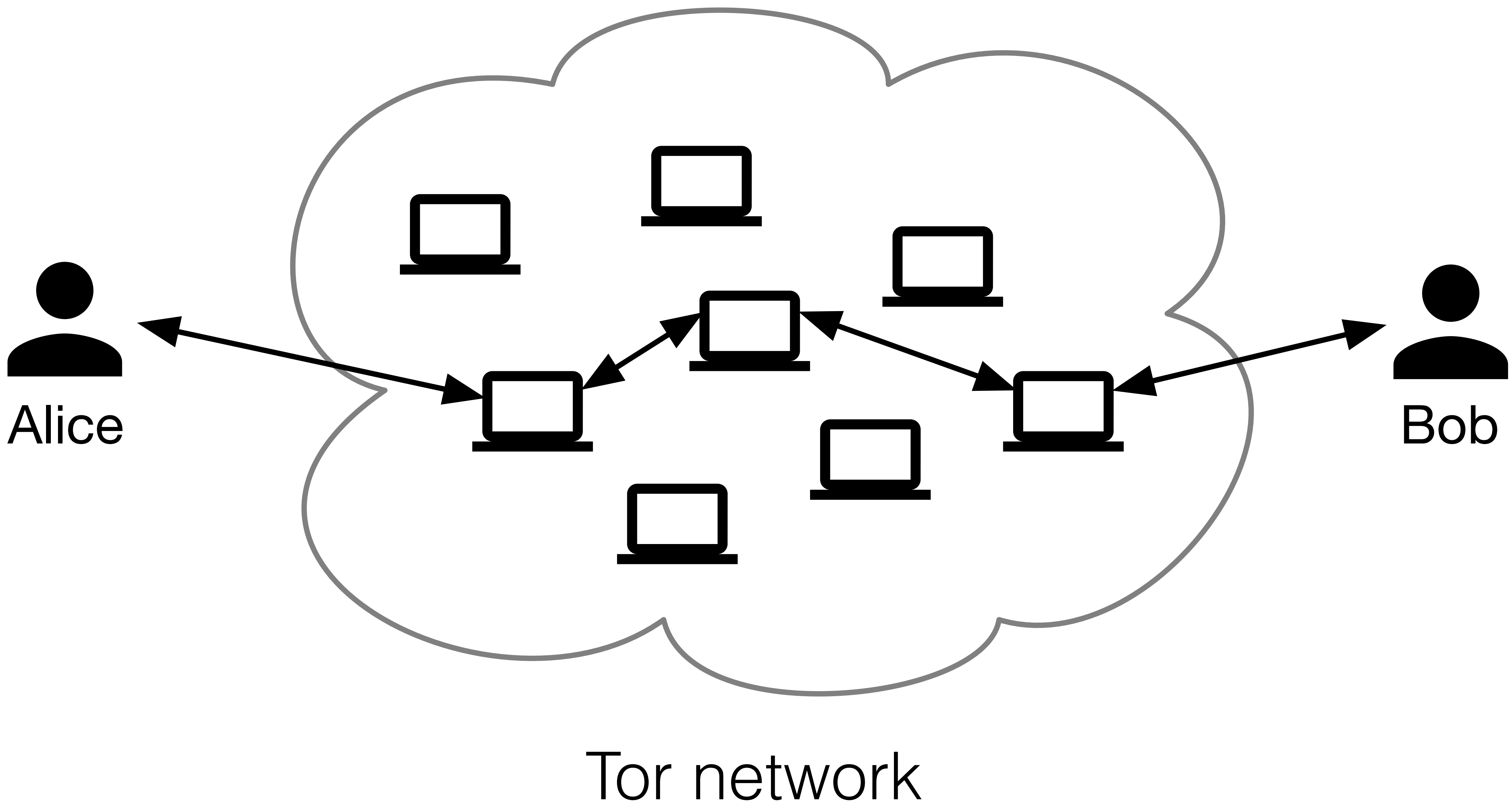
Goal: hide metadata



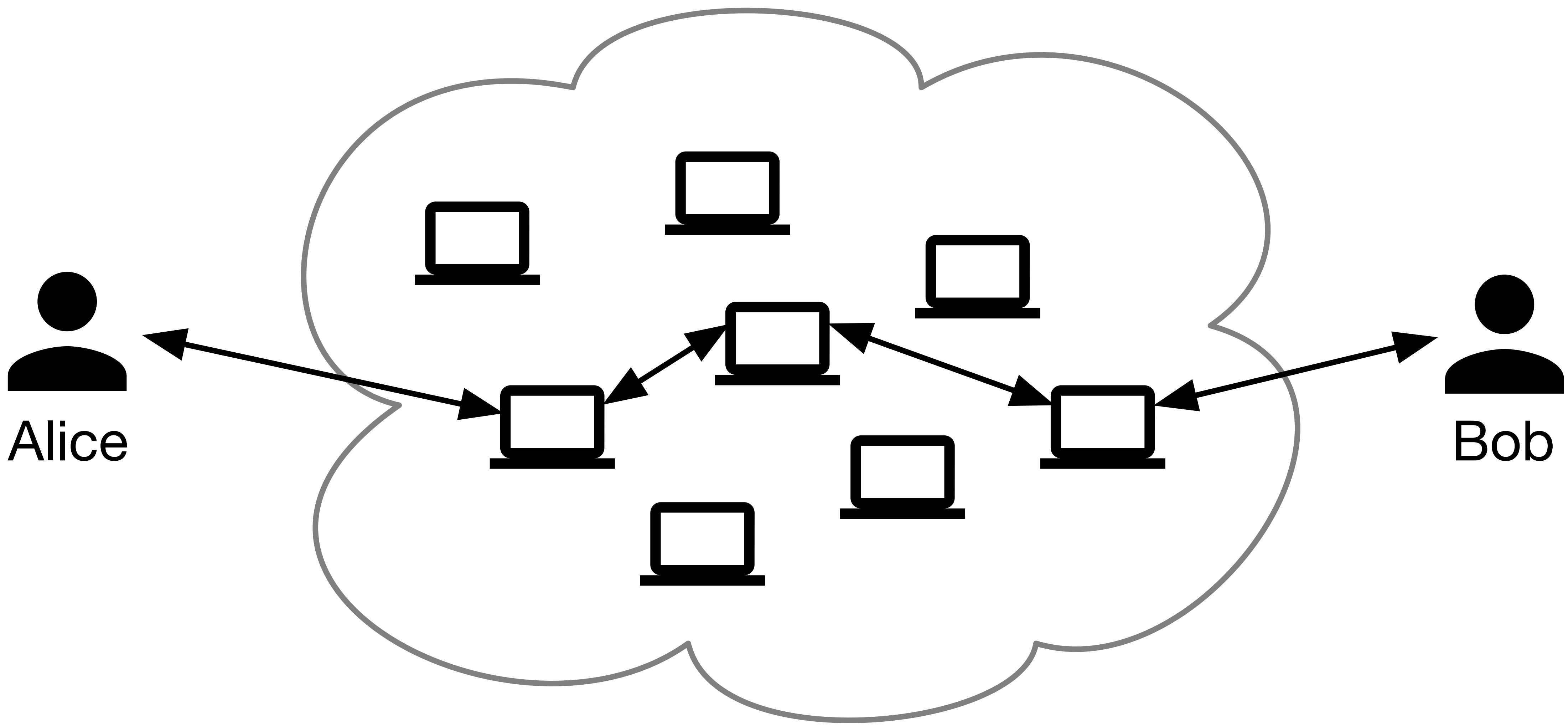
Goal: scalability



Tor is scalable

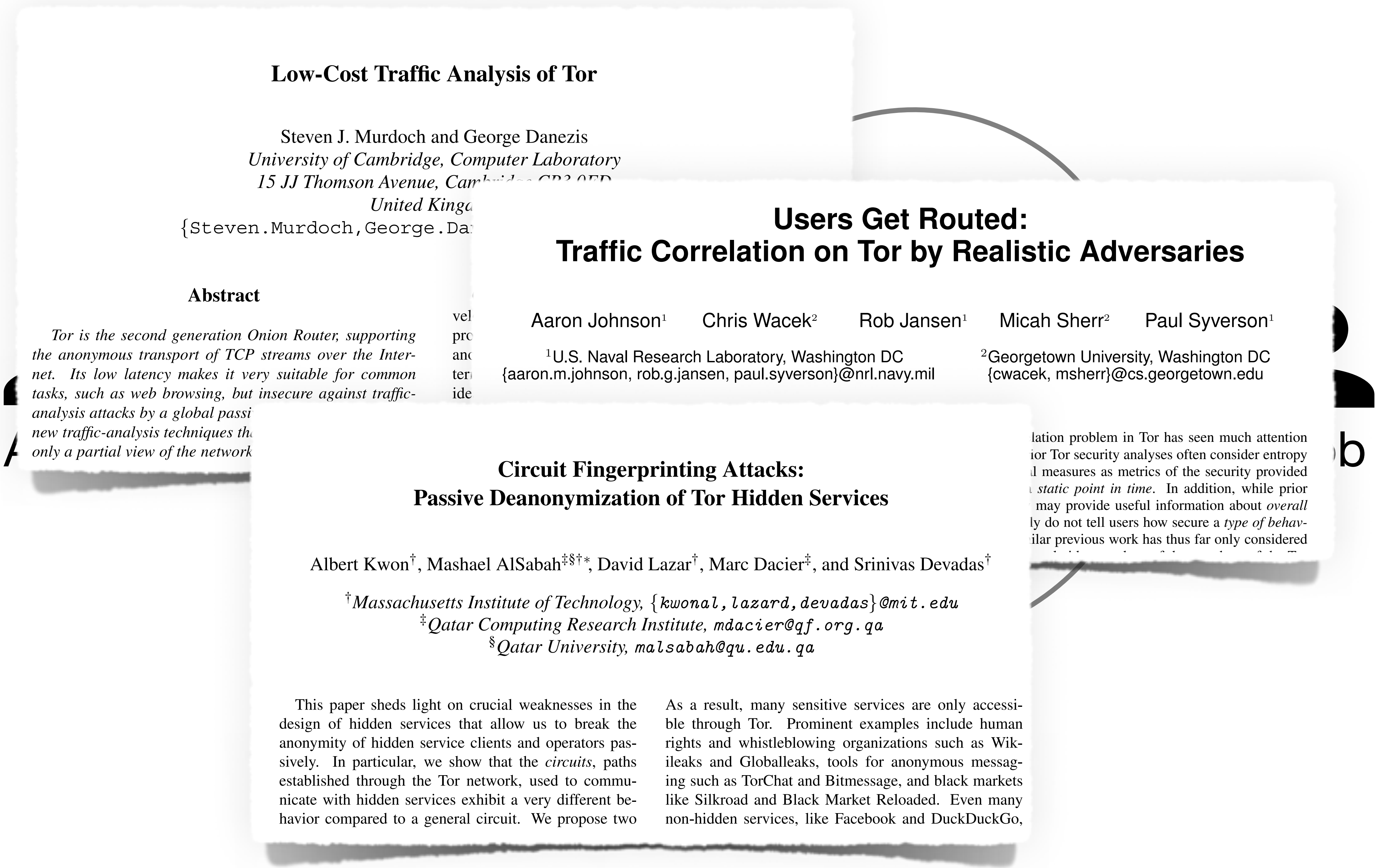


Tor is insecure



Tor network

Tor is insecure



Low-Cost Traffic Analysis of Tor

Steven J. Murdoch and George Danezis
University of Cambridge, Computer Laboratory
15 JJ Thomson Avenue, Cam
United King
{Steven.Murdoch,George.Da

Abstract

Tor is the second generation Onion Router, supporting the anonymous transport of TCP streams over the Internet. Its low latency makes it very suitable for common tasks, such as web browsing, but insecure against traffic-analysis attacks by a global passi
new traffic-analysis techniques th
only a partial view of the network

vel
pro
anc
ter
ide

Users Get Routed:
Traffic Correlation on Tor by Realistic Adversaries

Aaron Johnson[1] Chris Wacek[2] Rob Jansen[1] Micah Sherr[2] Paul Syverson[1]

[1]U.S. Naval Research Laboratory, Washington DC
{aaron.m.johnson, rob.g.jansen, paul.syverson}@nrl.navy.mil

[2]Georgetown University, Washington DC
{cwacek, msherr}@cs.georgetown.edu

lation problem in Tor has seen much attention
ior Tor security analyses often consider entropy
l measures as metrics of the security provided
static point in time. In addition, while prior
may provide useful information about *overall*
ly do not tell users how secure a *type of behav-*
ilar previous work has thus far only considered

Circuit Fingerprinting Attacks:
Passive Deanonymization of Tor Hidden Services

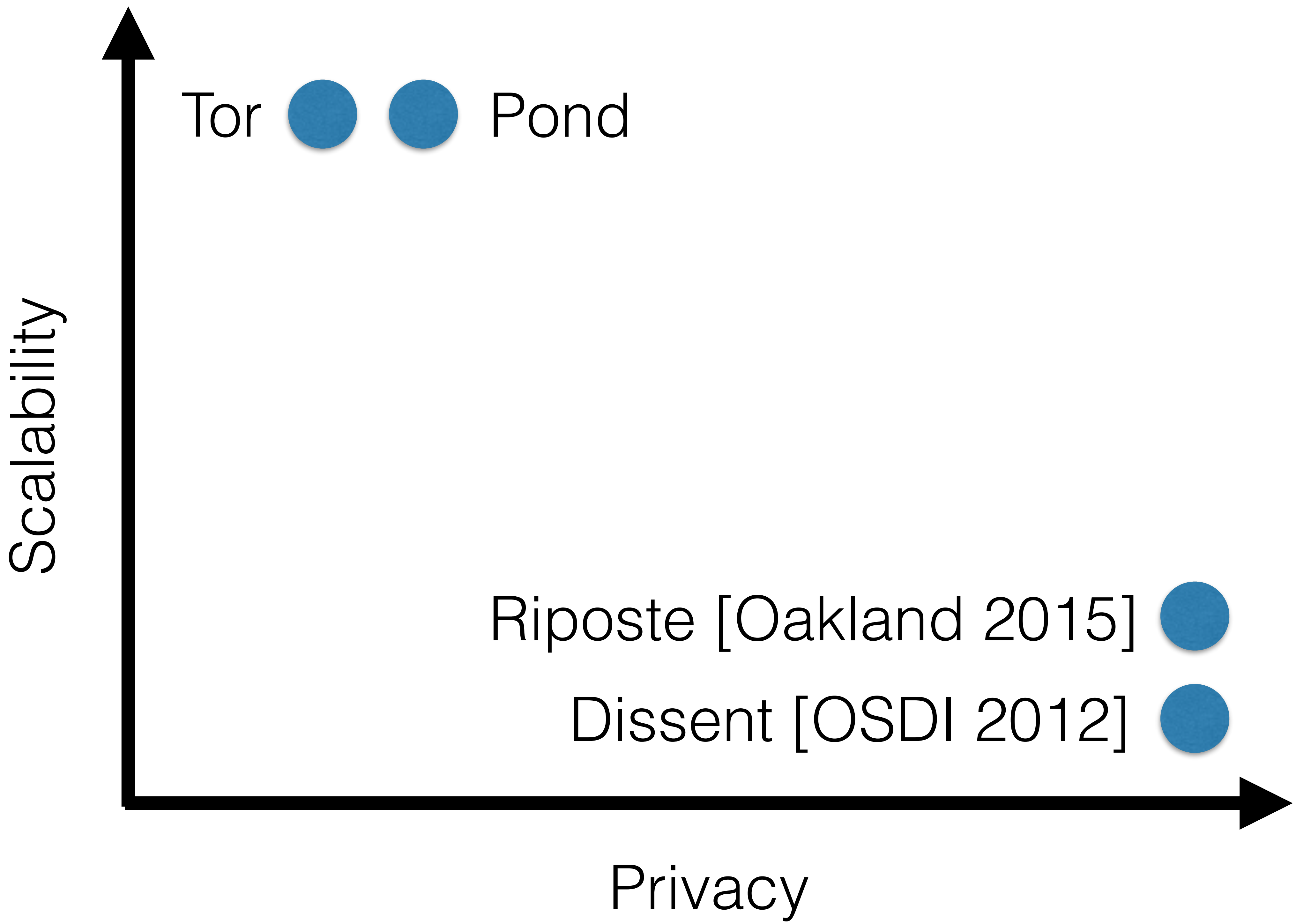
Albert Kwon[†], Mashael AlSabah[‡§†]*, David Lazar[†], Marc Dacier[‡], and Srinivas Devadas[†]

[†]*Massachusetts Institute of Technology,* {kwonal, lazard, devadas}@mit.edu
[‡]*Qatar Computing Research Institute,* mdacier@qf.org.qa
[§]*Qatar University,* malsabah@qu.edu.qa

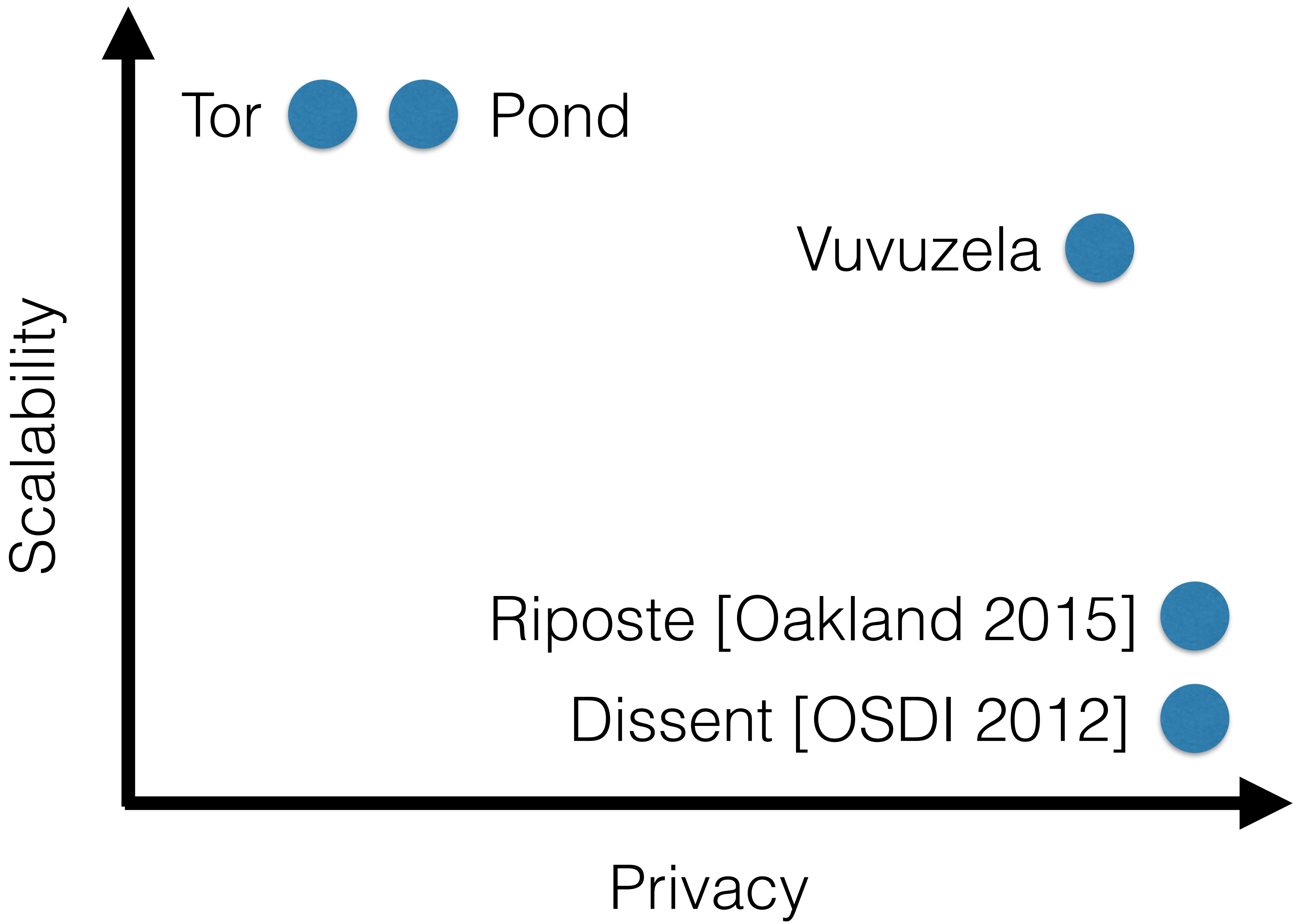
This paper sheds light on crucial weaknesses in the design of hidden services that allow us to break the anonymity of hidden service clients and operators passively. In particular, we show that the *circuits*, paths established through the Tor network, used to communicate with hidden services exhibit a very different behavior compared to a general circuit. We propose two

As a result, many sensitive services are only accessible through Tor. Prominent examples include human rights and whistleblowing organizations such as Wikileaks and Globalleaks, tools for anonymous messaging such as TorChat and Bitmessage, and black markets like Silkroad and Black Market Reloaded. Even many non-hidden services, like Facebook and DuckDuckGo,

Related work



Contribution

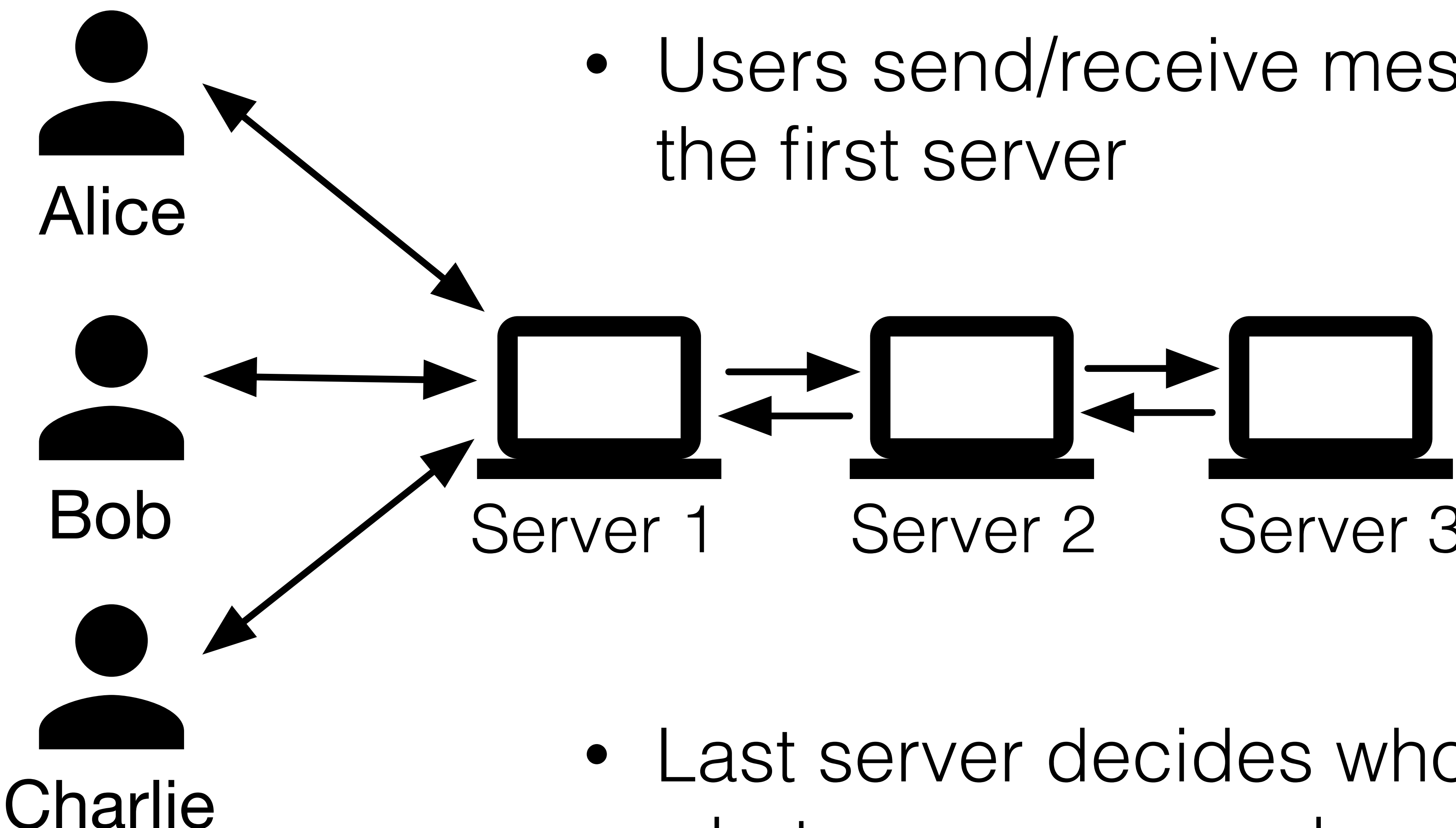


Contribution

- **Vuvuzela**: the first private messaging system that hides metadata from powerful adversaries for millions of users
    - Vuvuzela scales linearly with the number of users
    - Differential privacy for millions of messages per user for one million users
    - 37s end-to-end message latency
    - 60,000 messages / second throughput
    - Good match for private text-based messaging

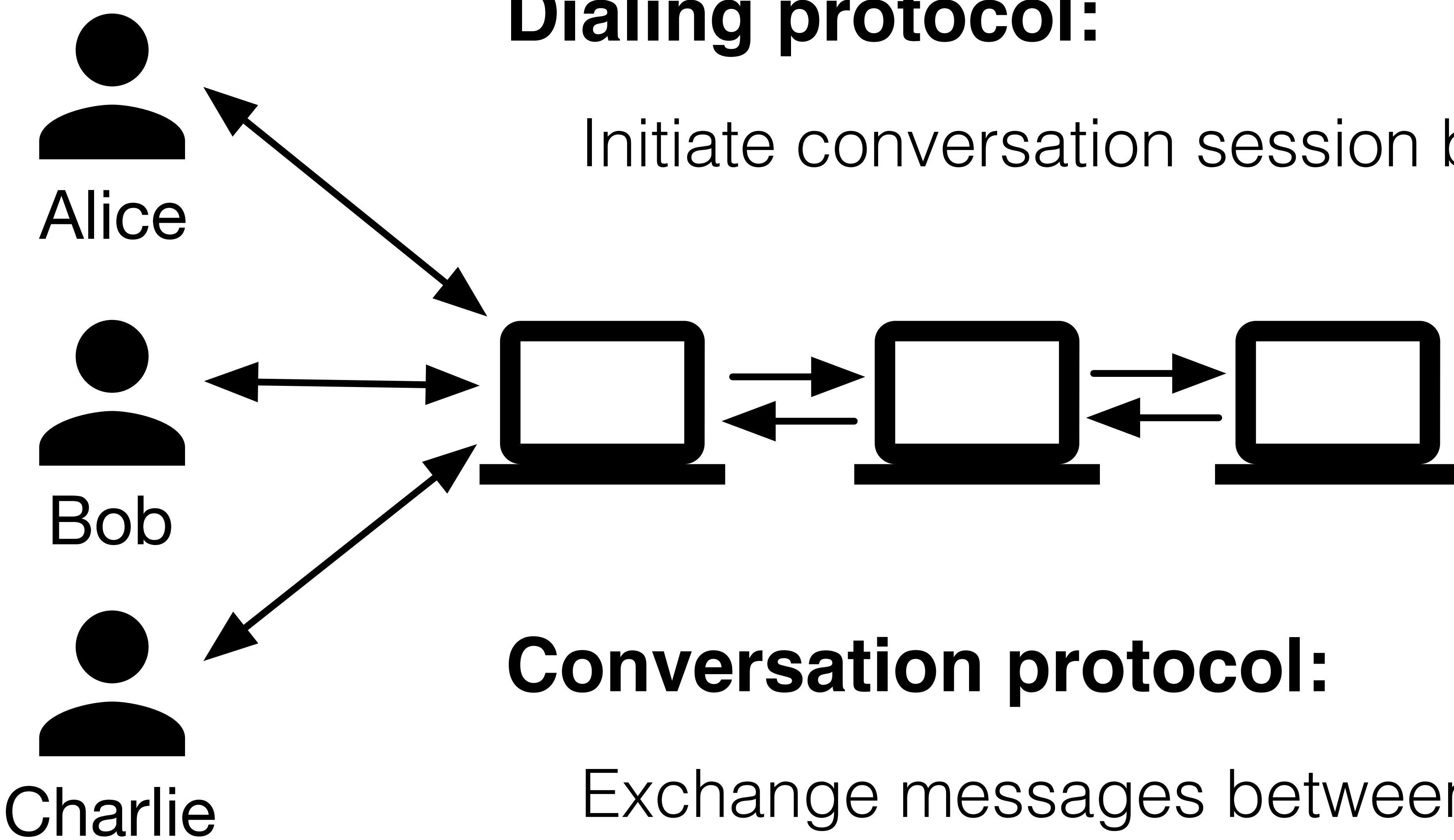
Vuvuzela overview

- Handful of servers arranged in a chain
- Users send/receive messages through the first server



- Last server decides who gets what messages and sends them back down the chain

Vuvuzela's two protocols



Dialing protocol:

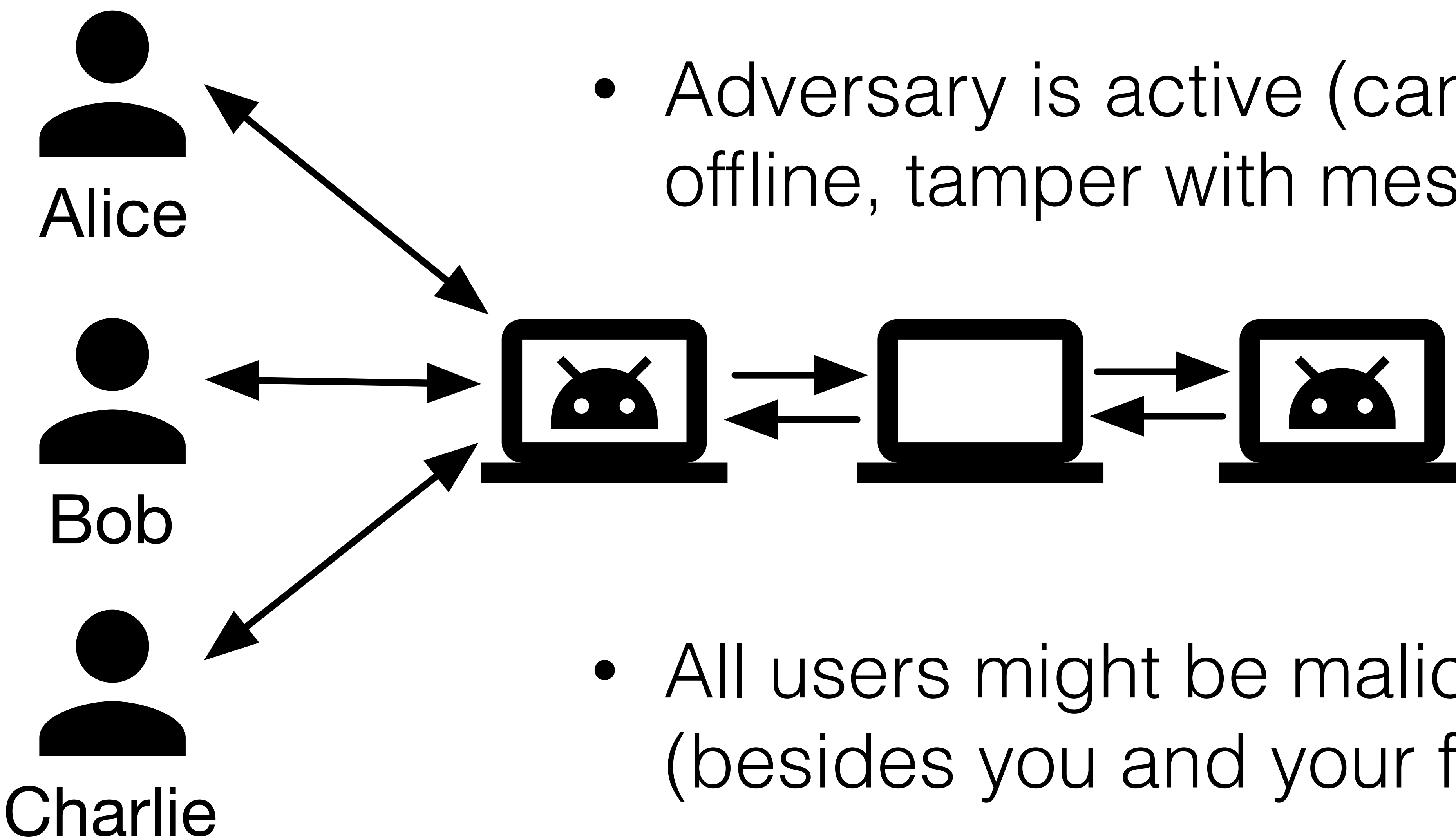
Initiate conversation session between two users

Conversation protocol:

Exchange messages between two users

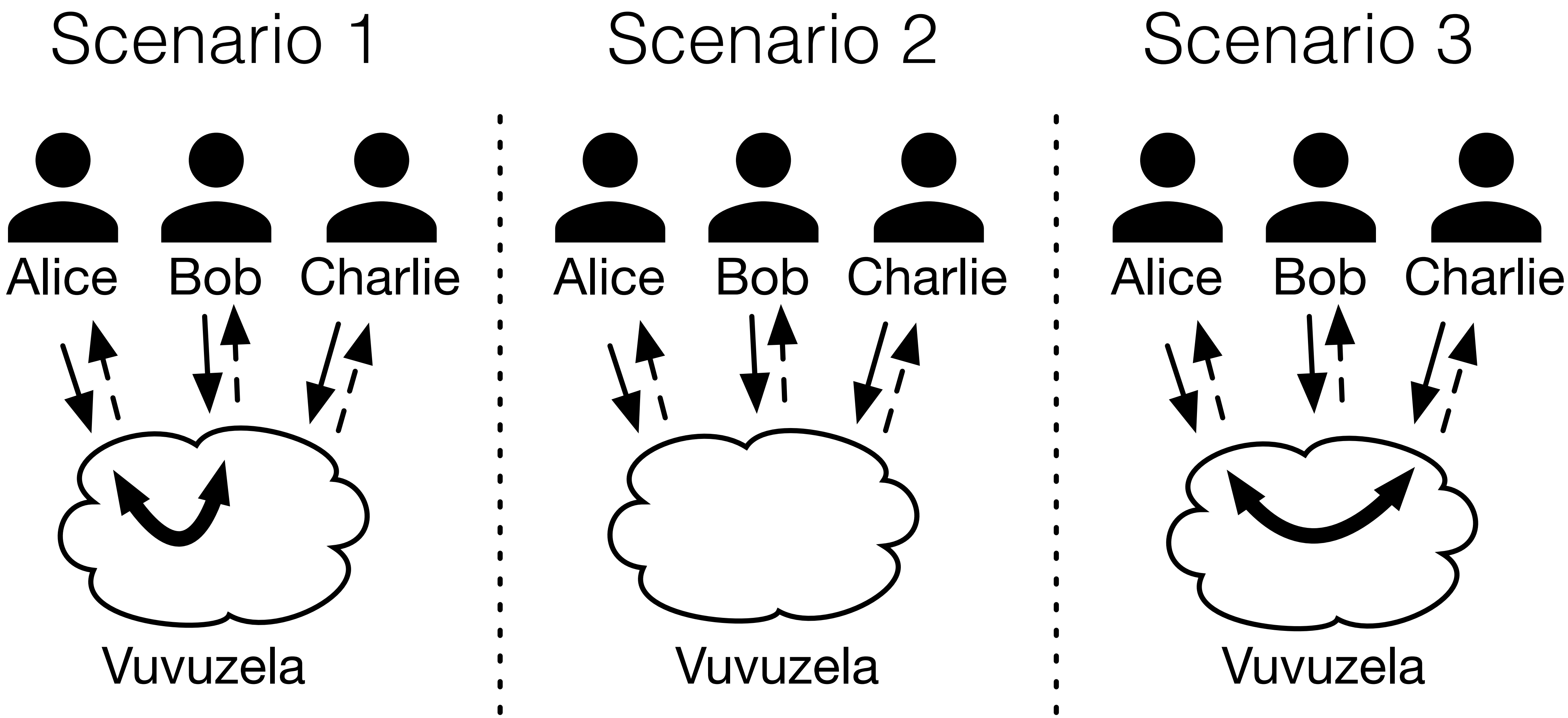
Threat model

- All but one server are compromised
- Adversary is active (can knock users offline, tamper with messages, etc)

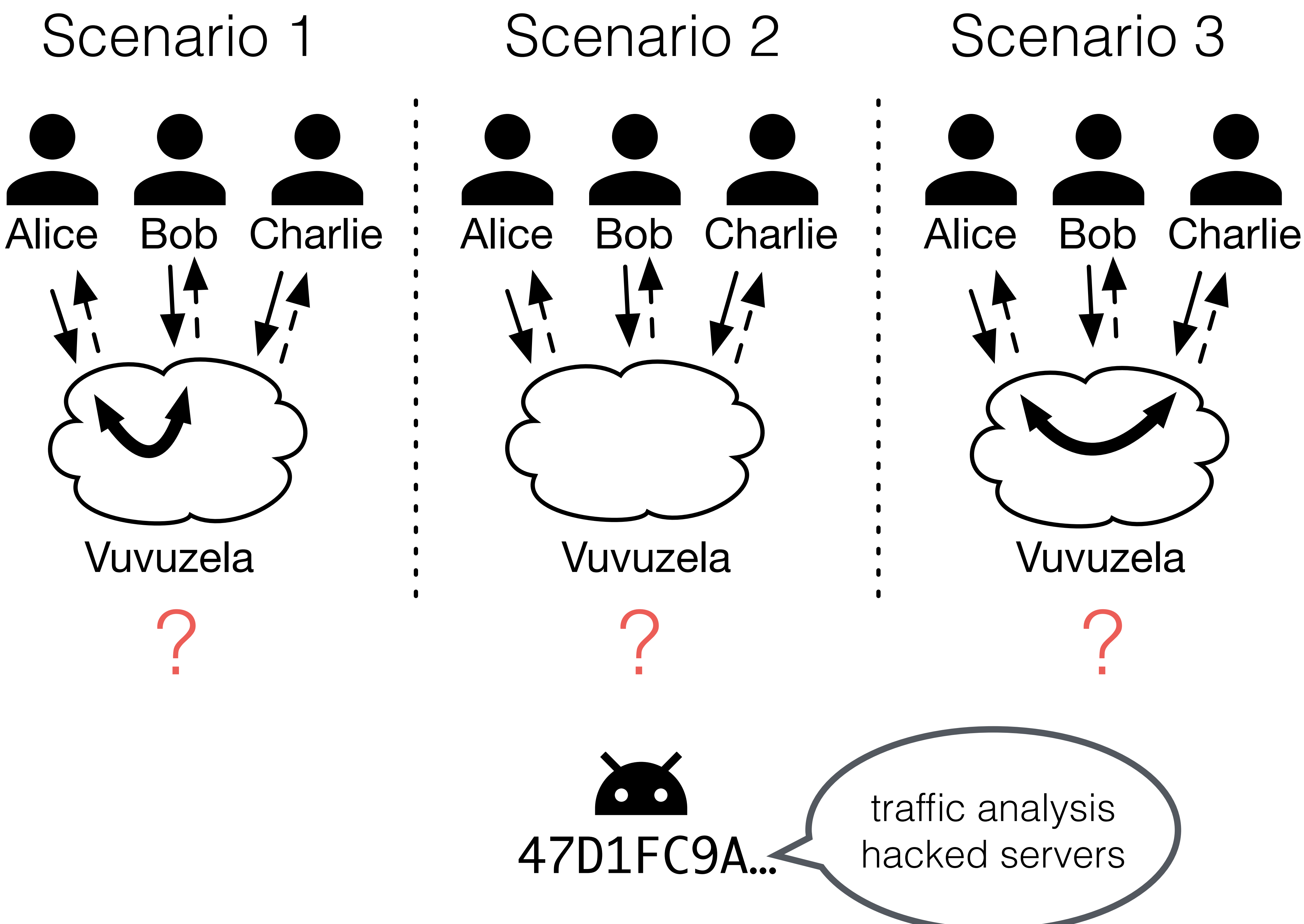


- All users might be malicious (besides you and your friends)
- PKI: users know each other's keys

Metadata privacy



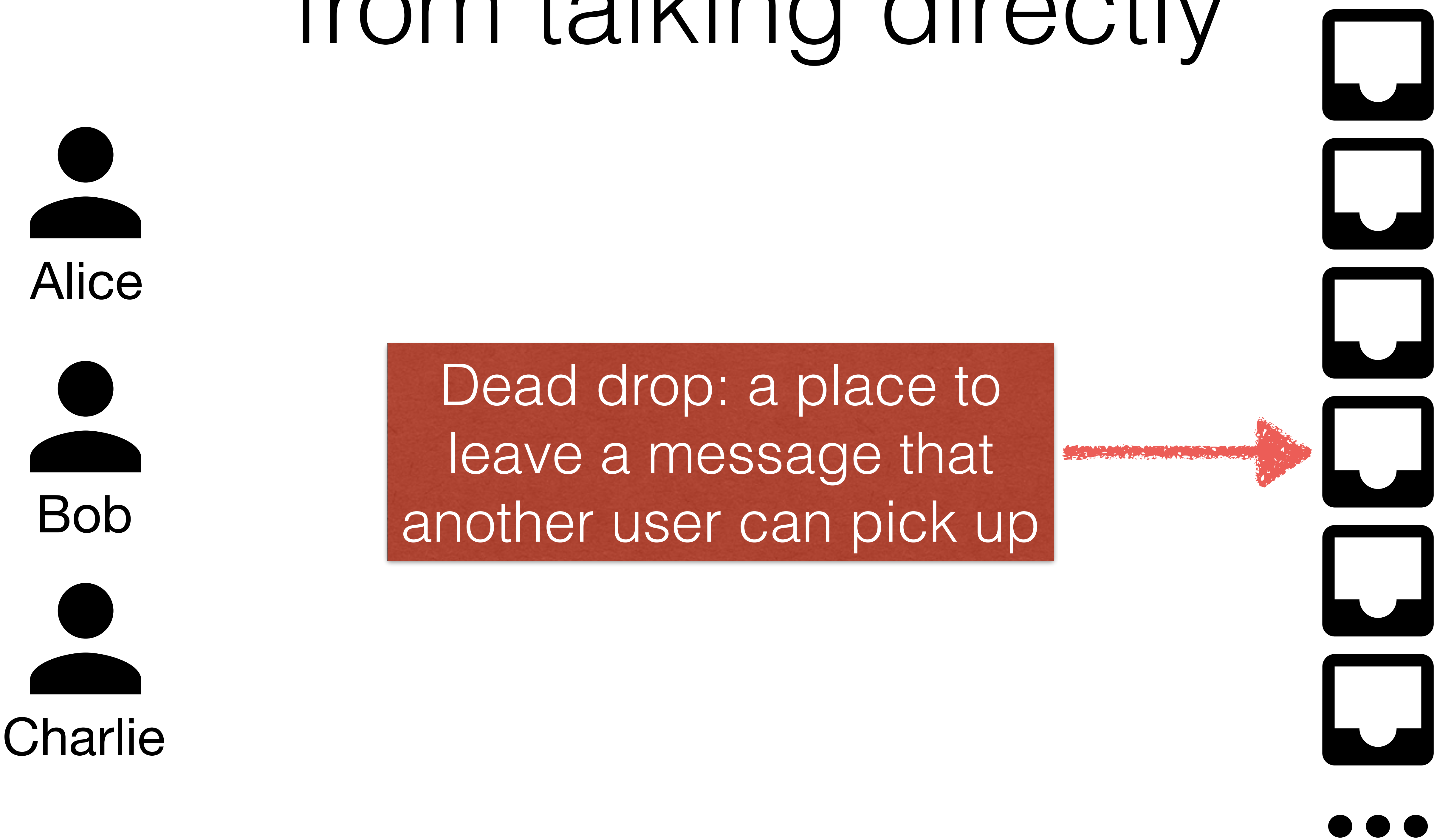
Metadata privacy



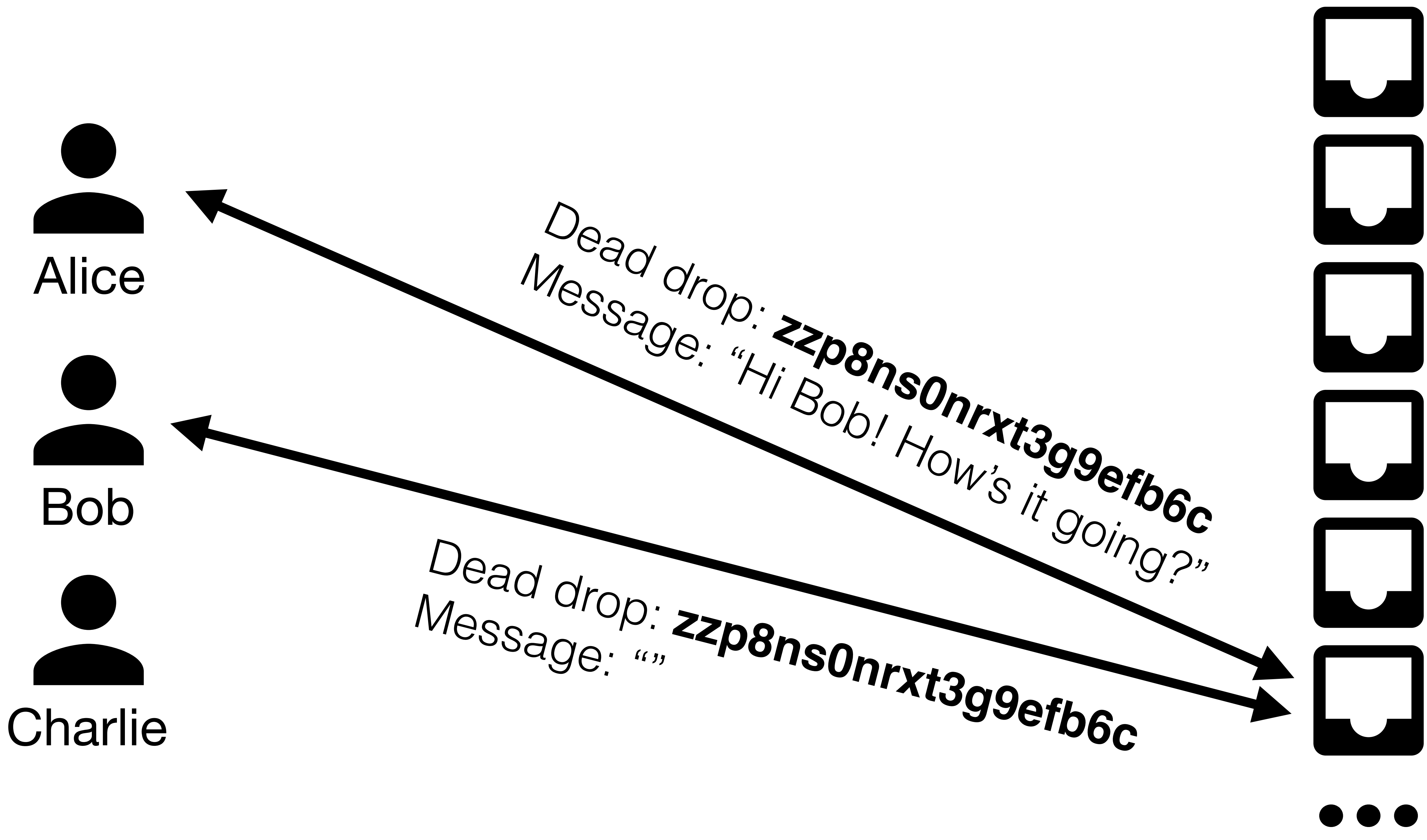
Approach to scalable privacy

- Use efficient cryptography to encrypt as much metadata as possible.
- Add noise to metadata that we can't "encrypt."
- Use differential privacy to reason about how much privacy the noise gives us.

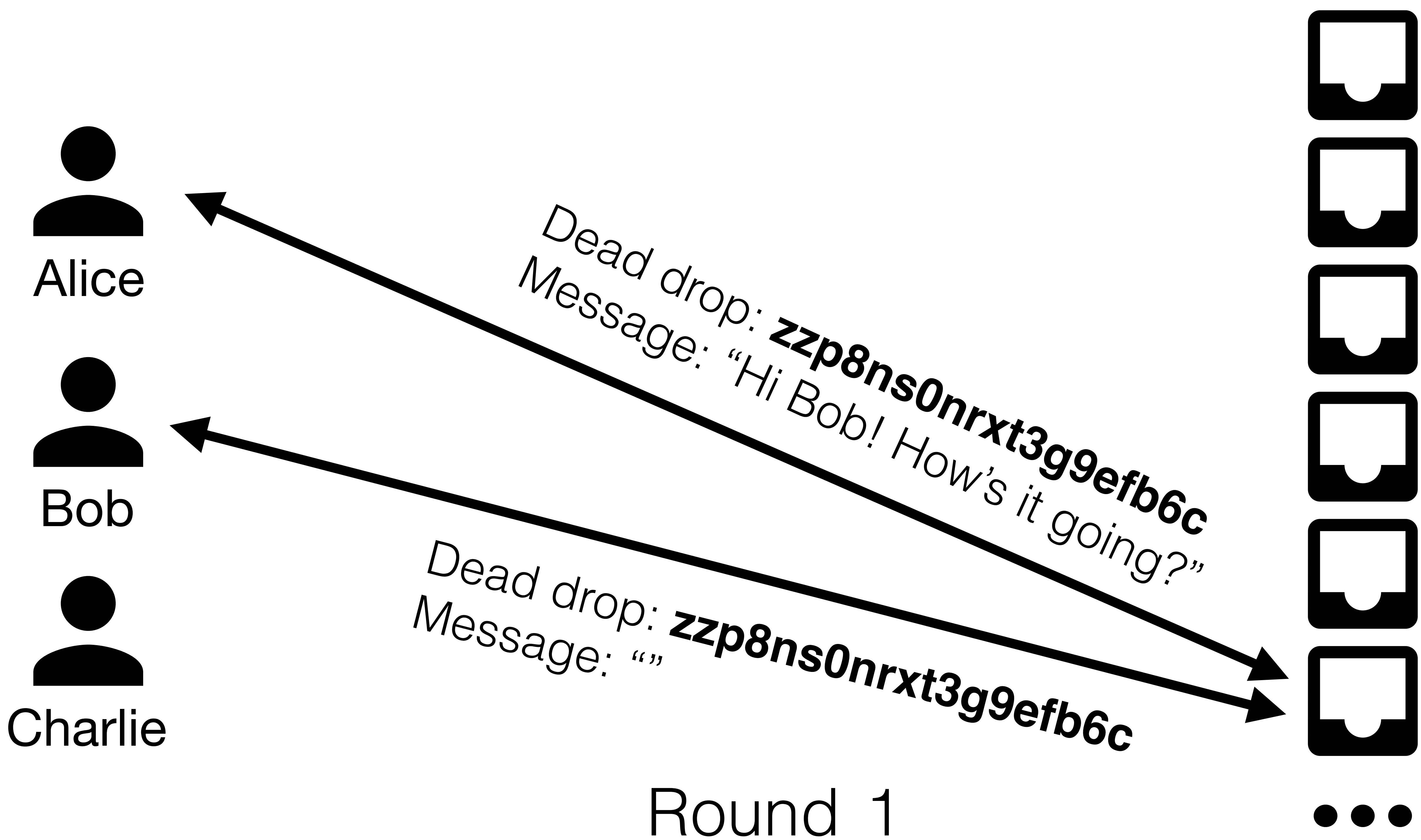
Dead drops prevent users from talking directly



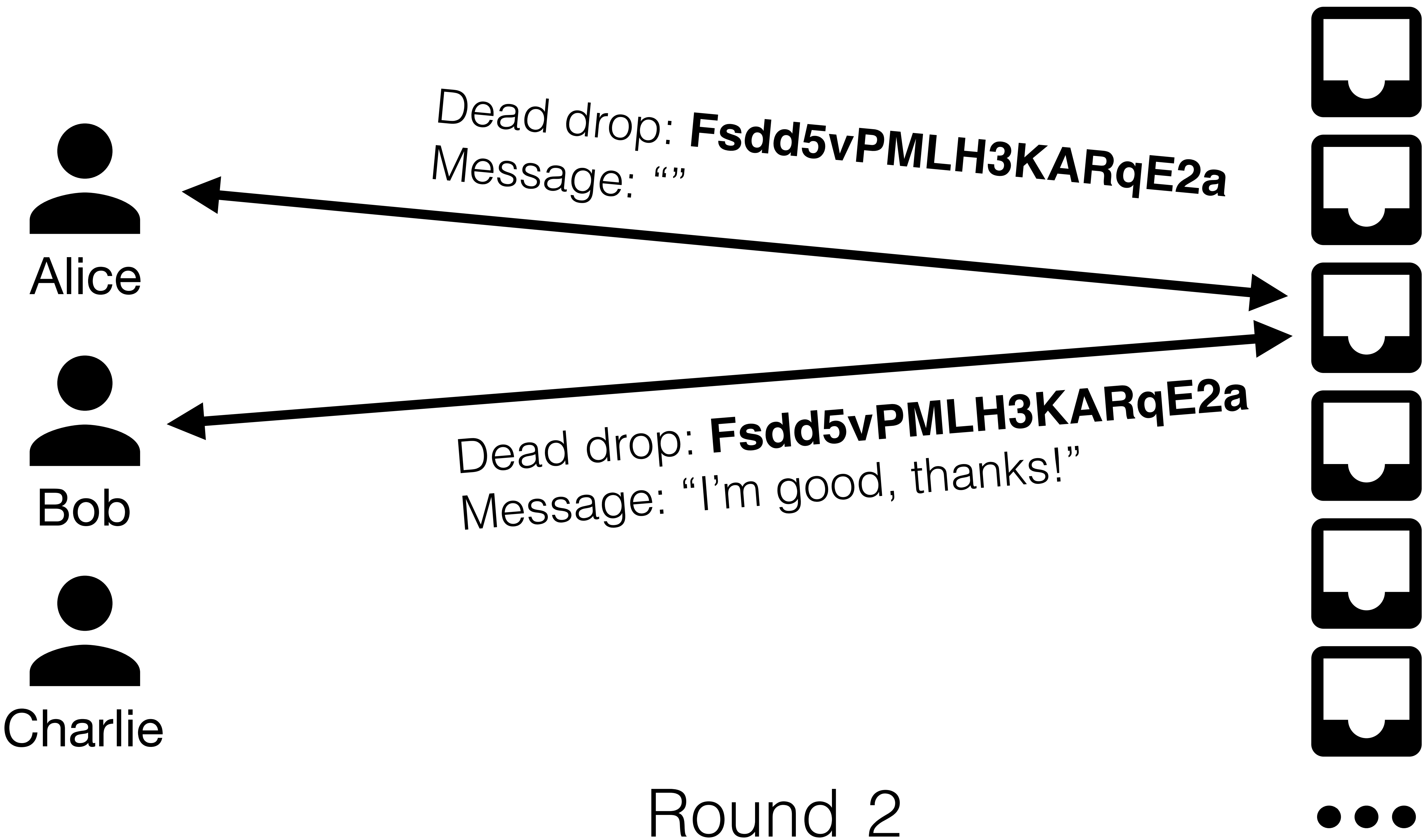
Talking via dead drops



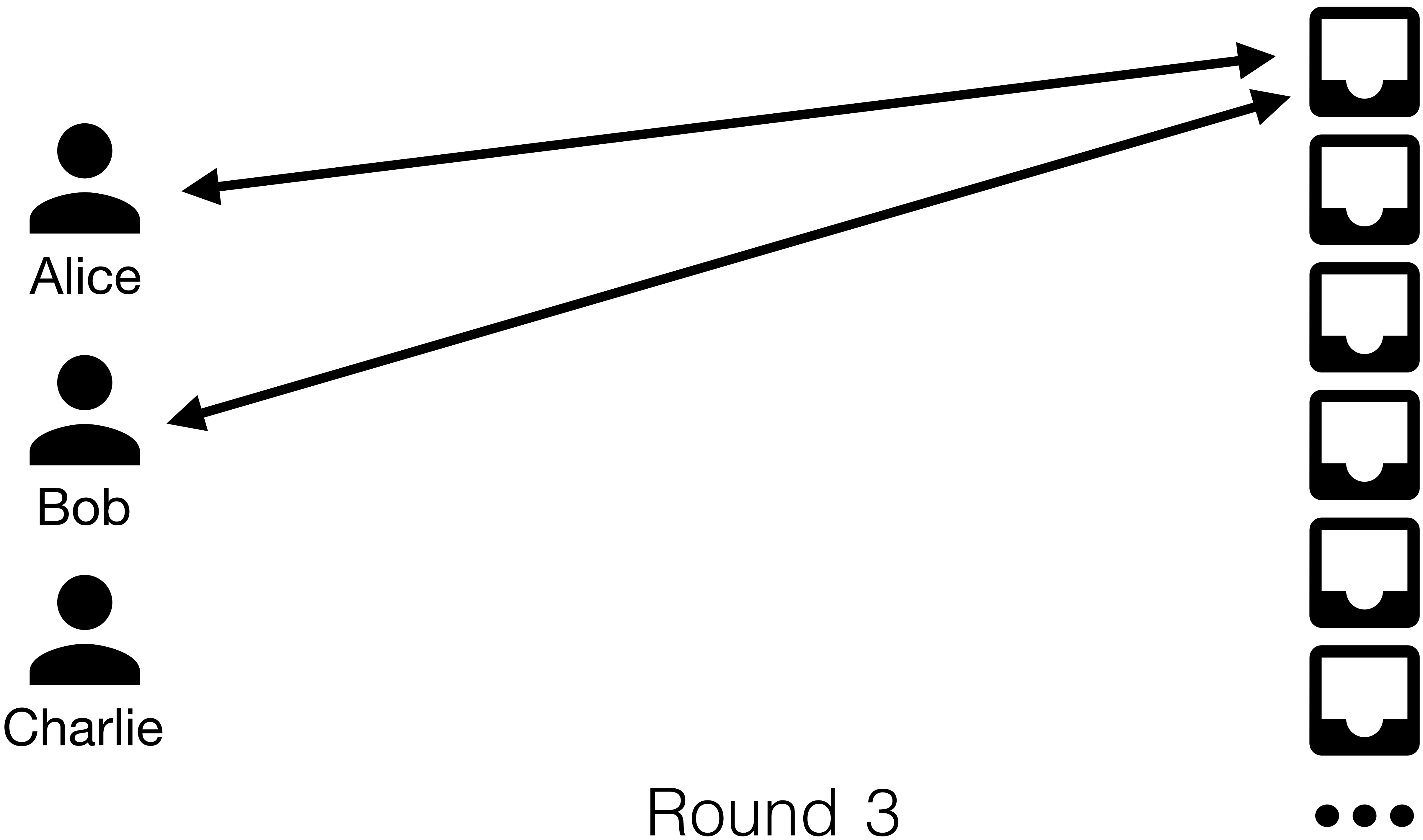
Conversation protocol



Conversation protocol

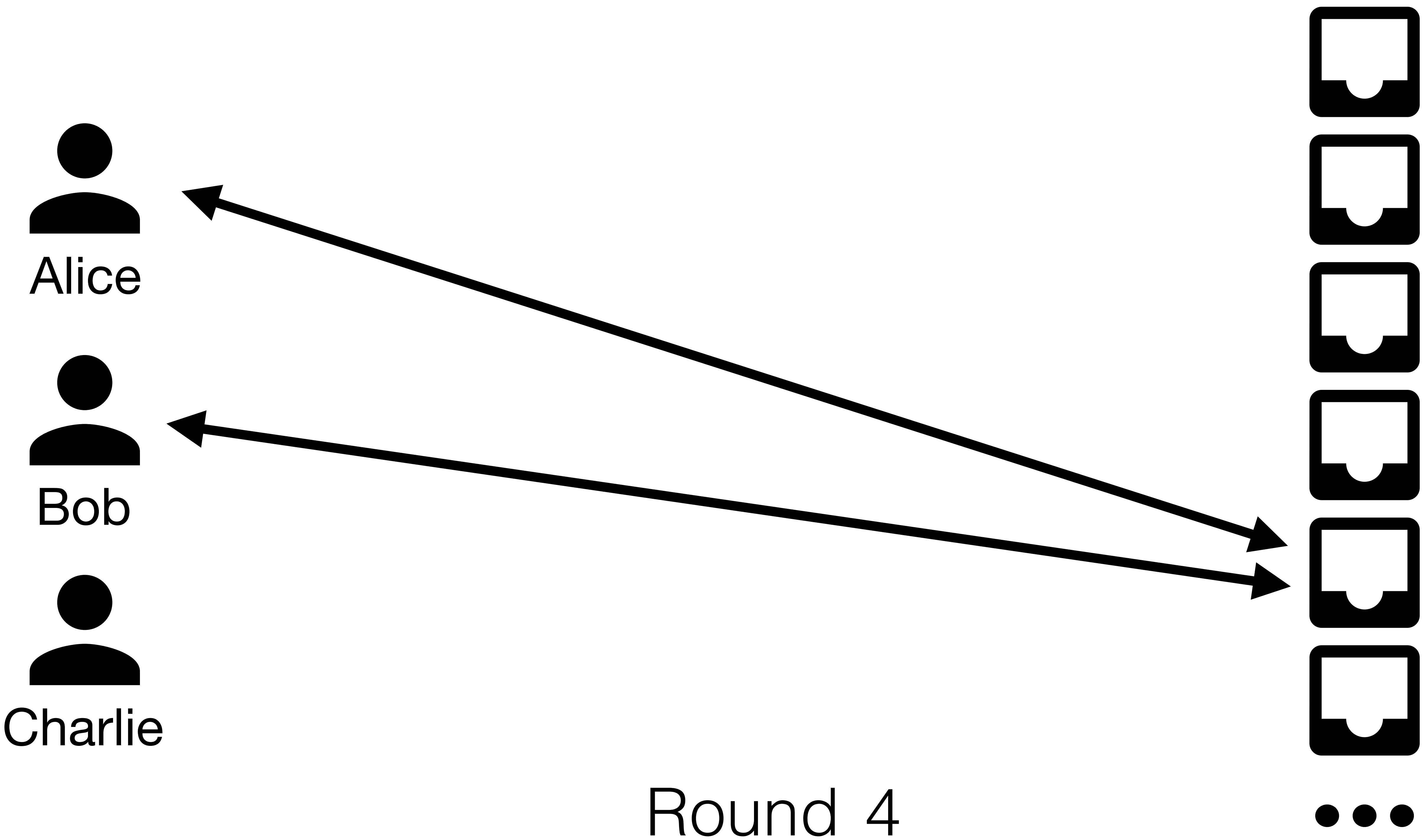


Conversation protocol



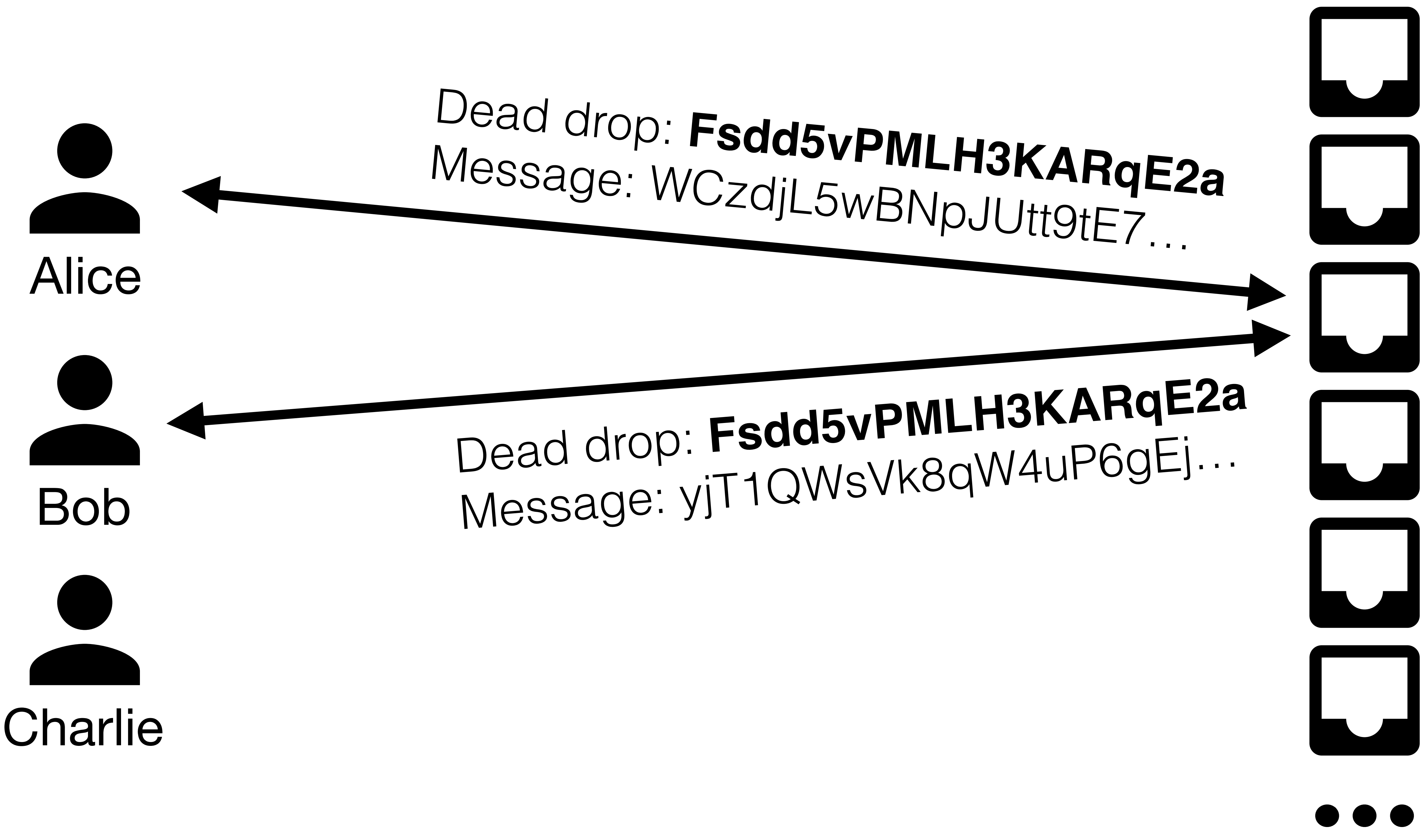
Round 3

Conversation protocol

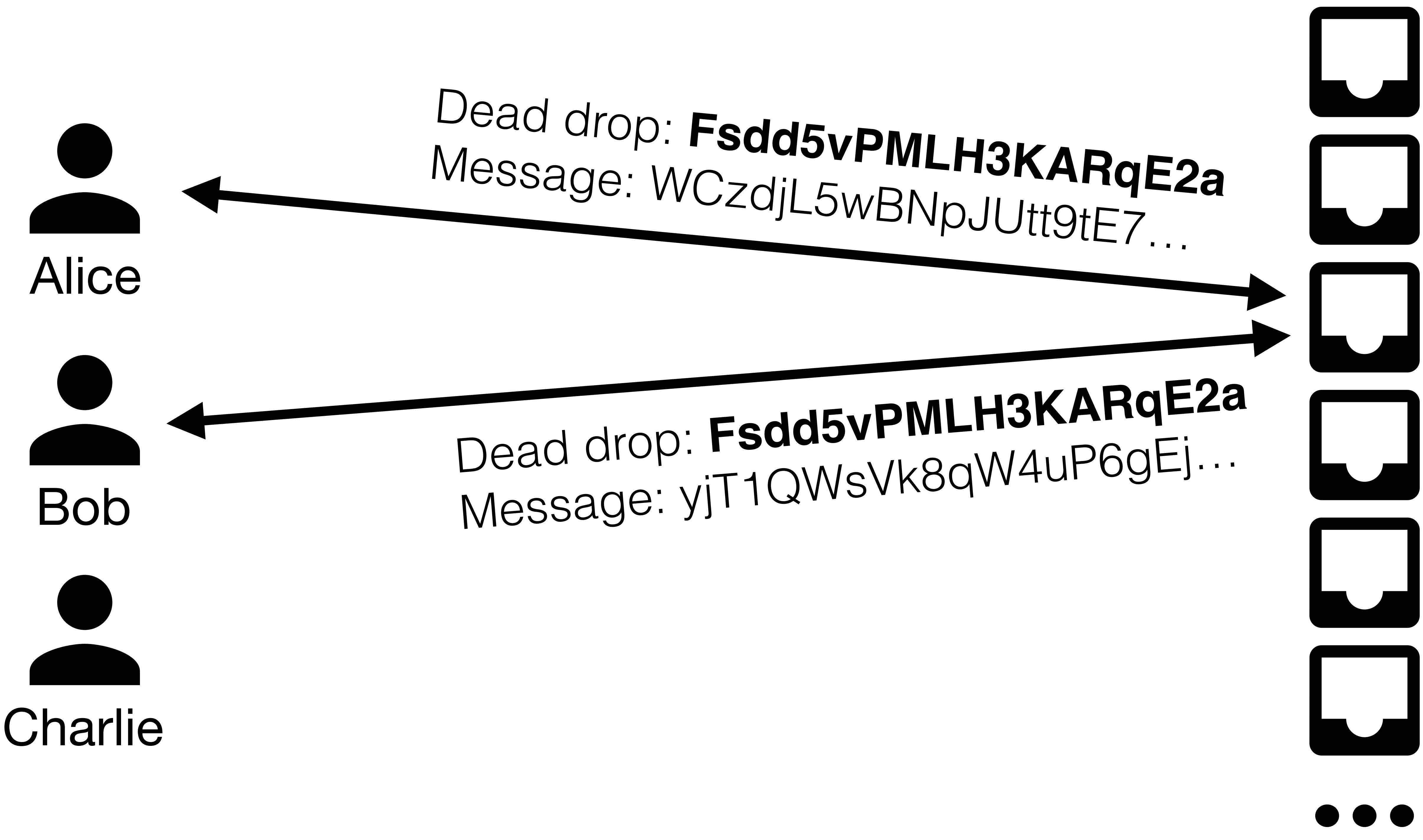


Round 4

Messages are encrypted



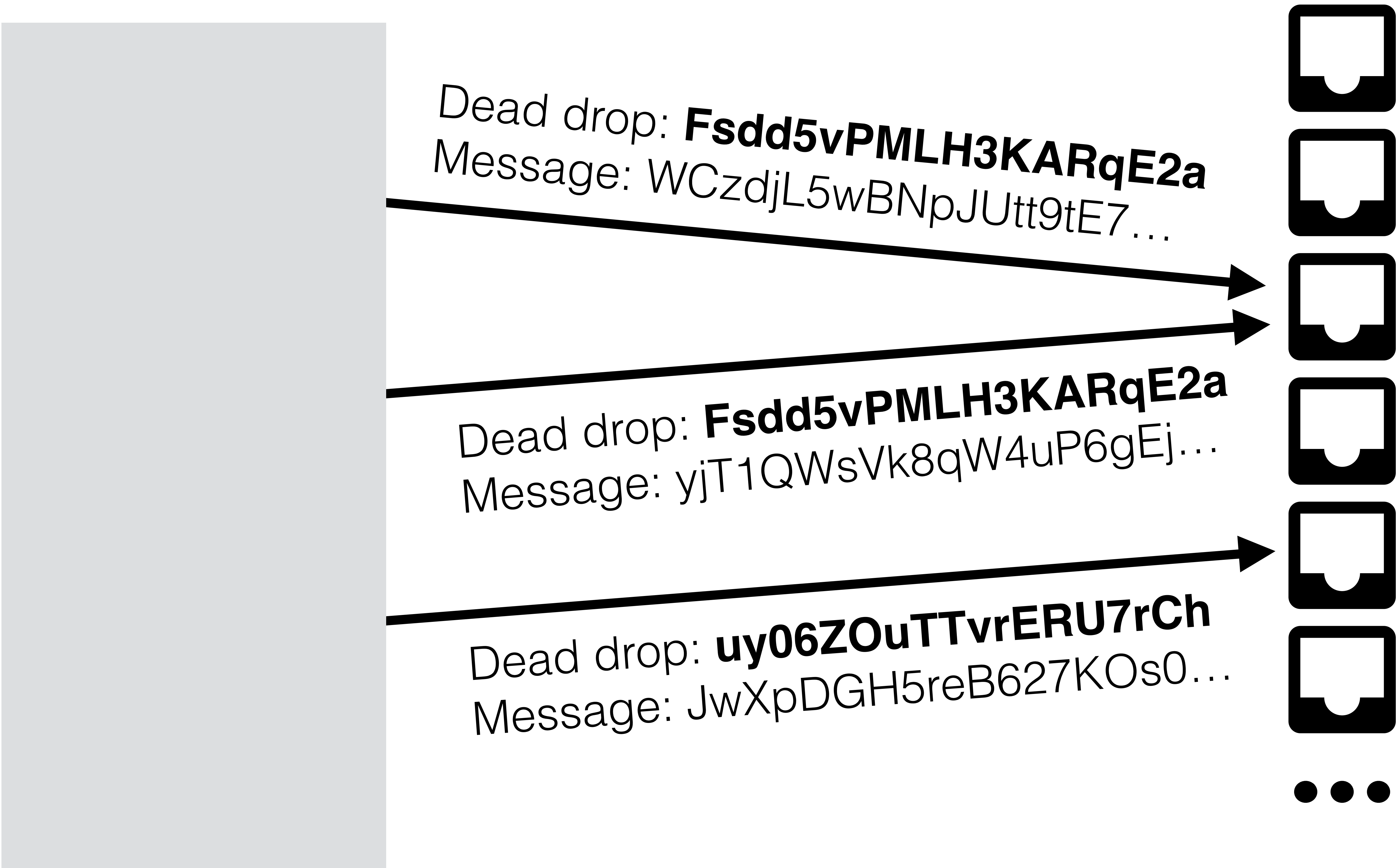
Idle clients send cover traffic



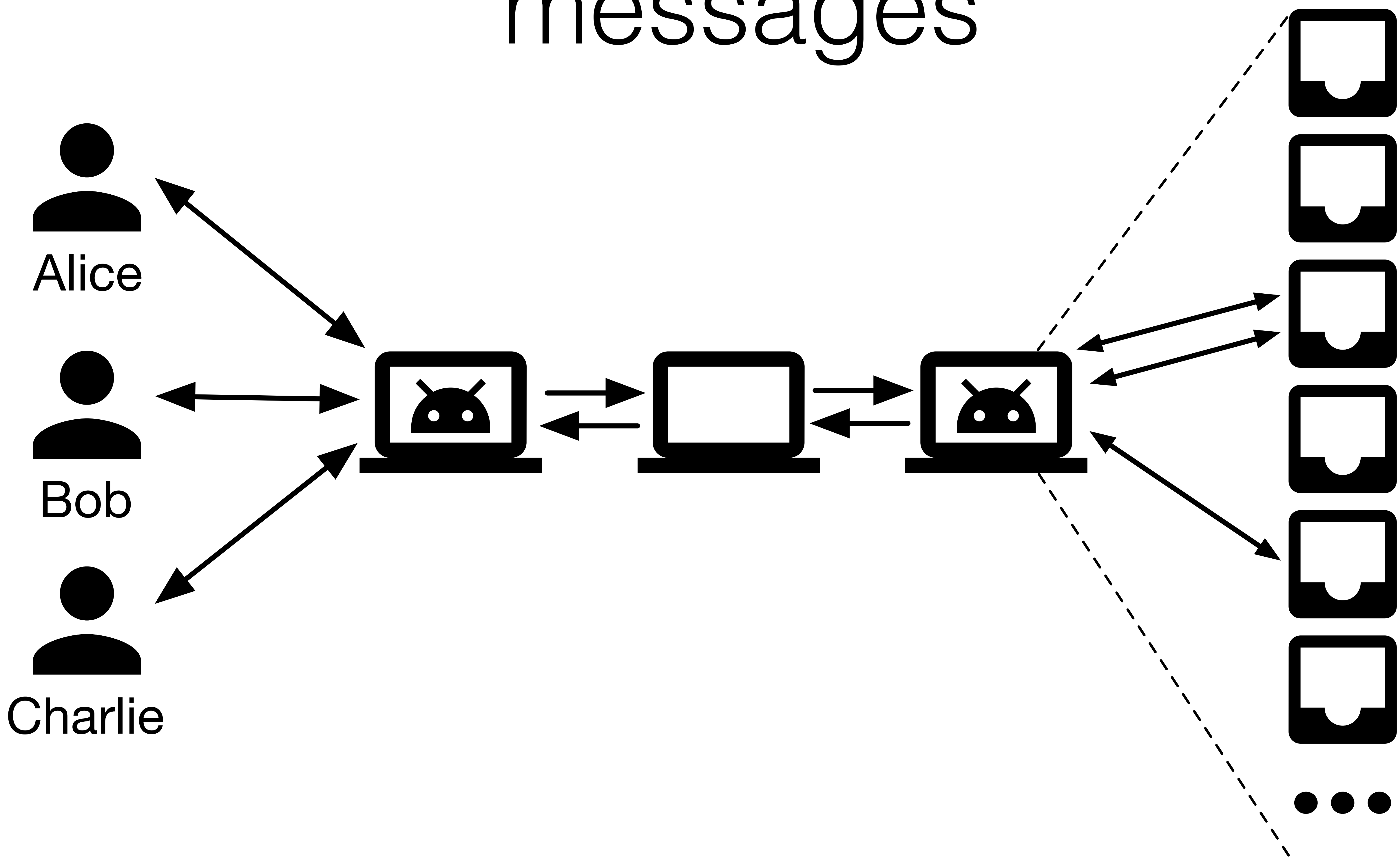
Idle clients send cover traffic

Dead drops give privacy

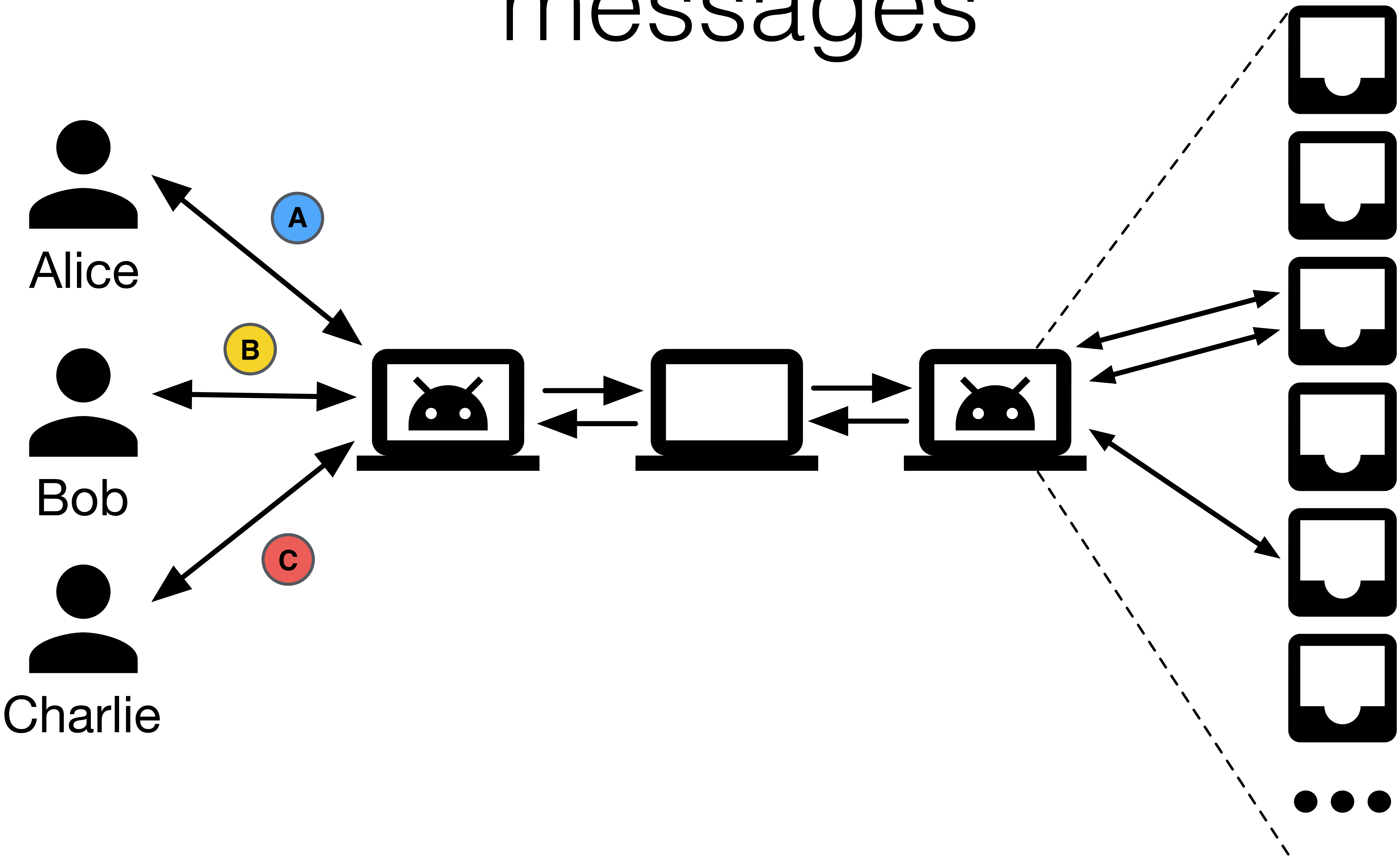
Dead drops give privacy



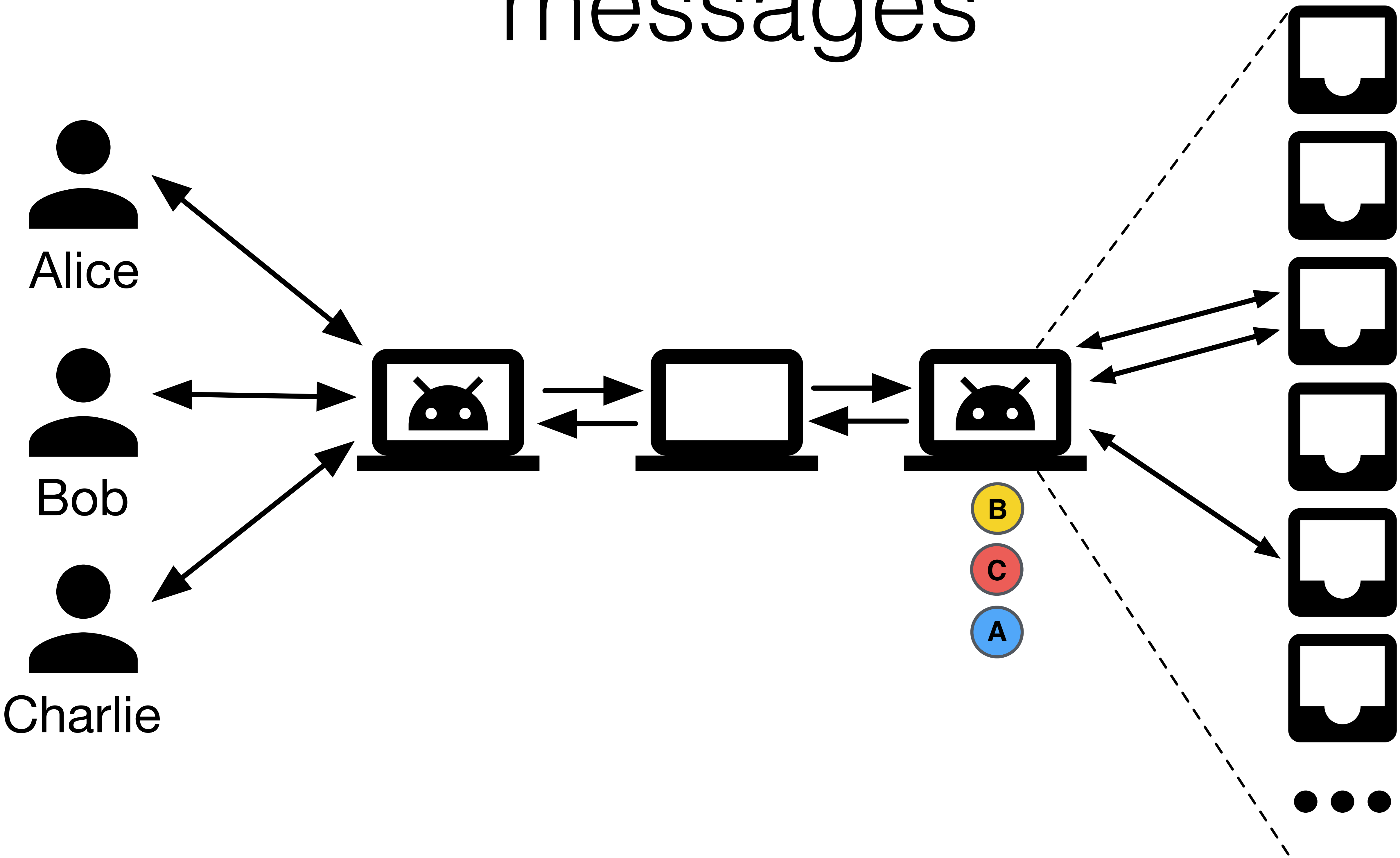
Mixnet hides origin of messages



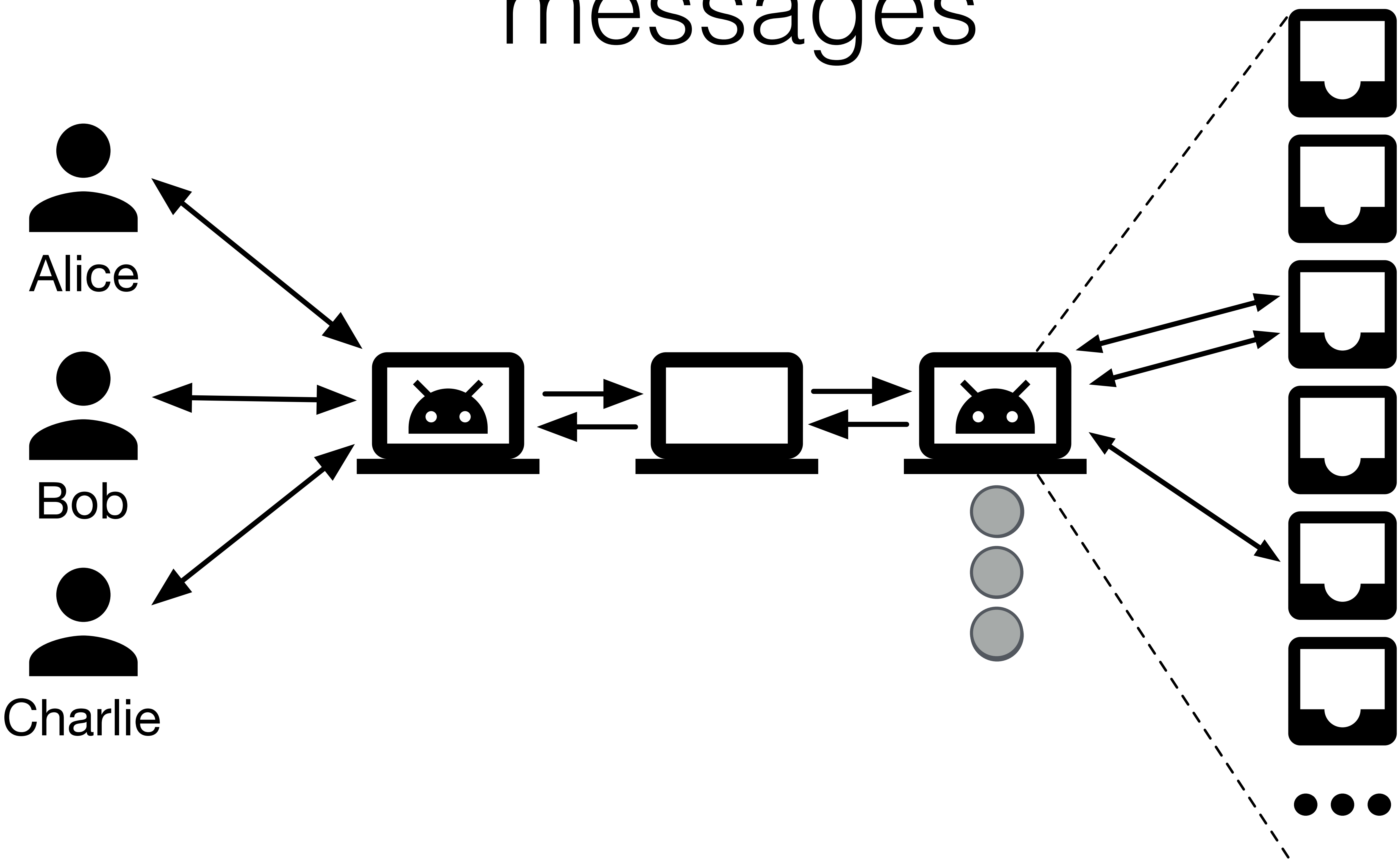
Mixnet hides origin of messages



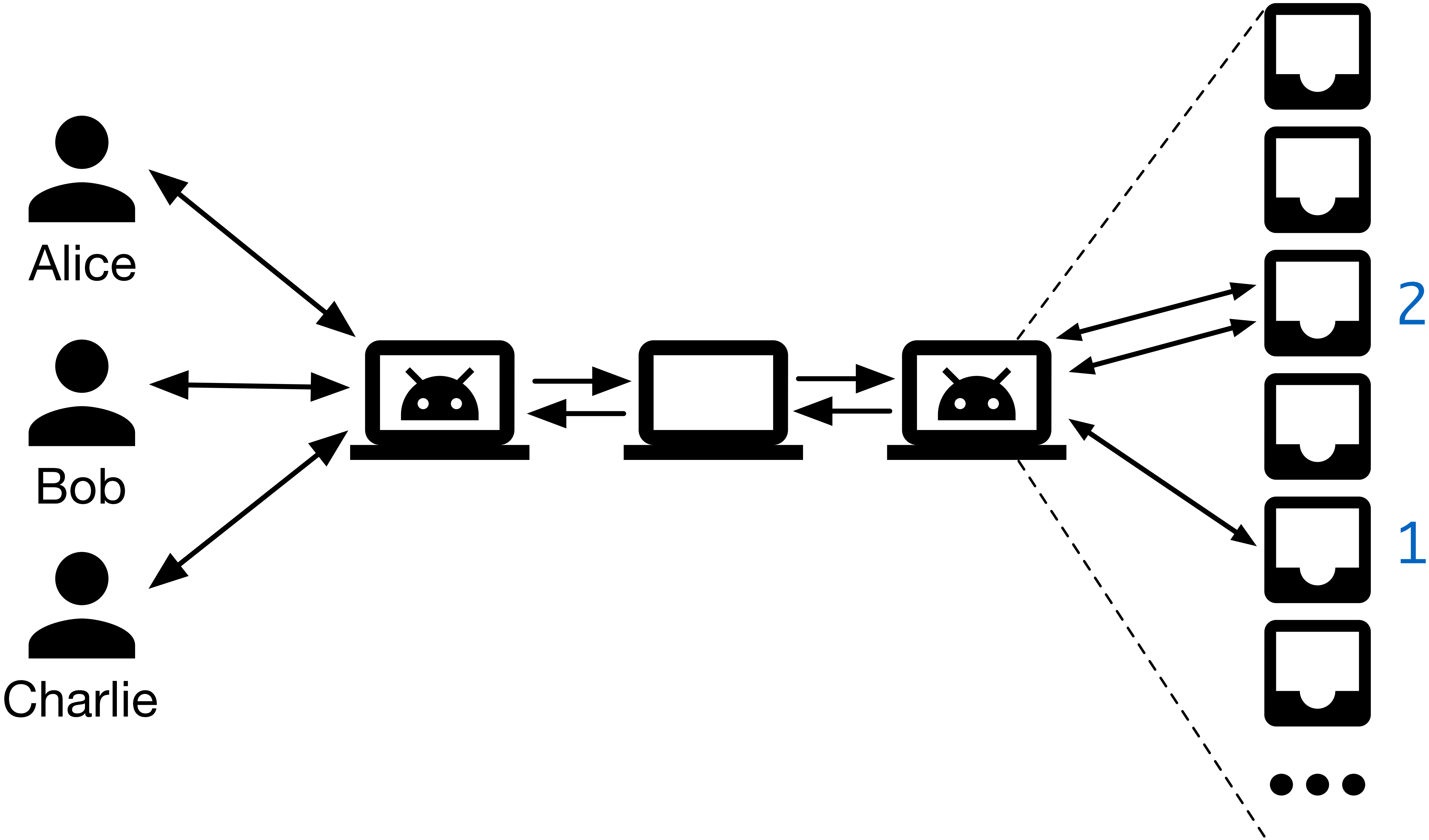
Mixnet hides origin of messages



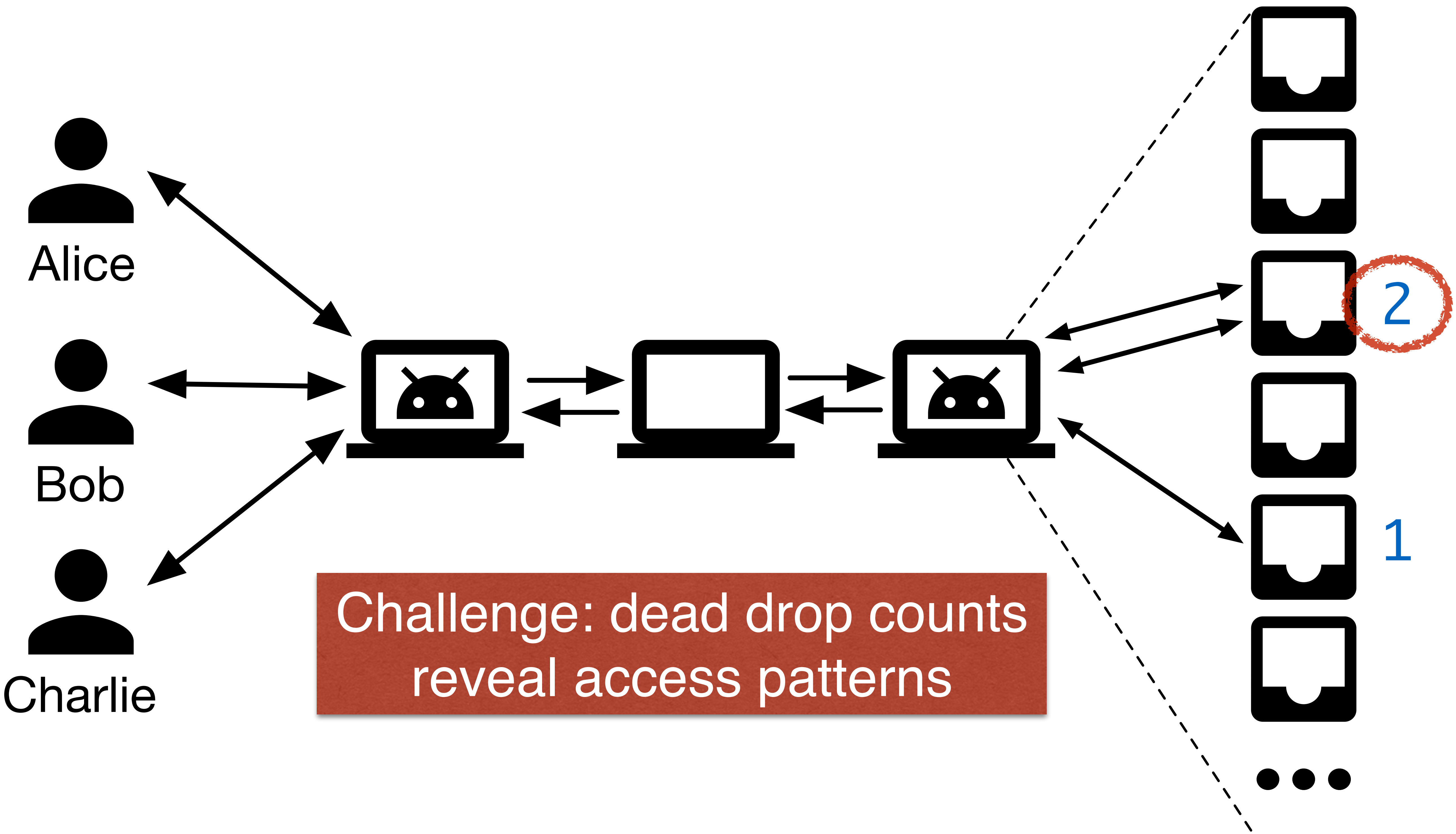
Mixnet hides origin of messages



Are we done yet?



Are we done yet?

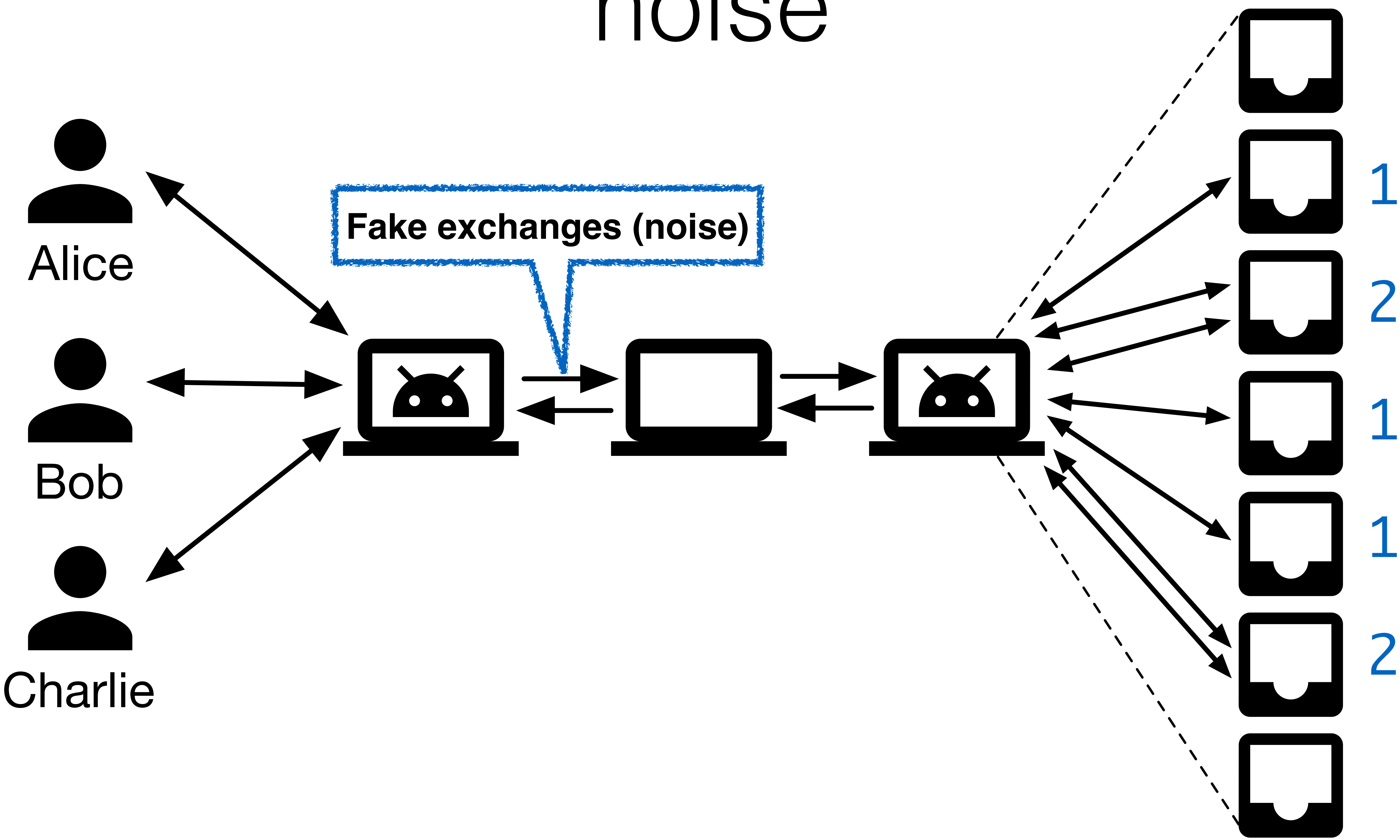


Challenge: dead drop counts reveal access patterns

Demo!

Let’s see why access counts are a problem.

Solution: Each server adds noise



What is noise?

Fake singles

Dead drop: **RY9VjW4XROtTcbnZPaJ**
Message: Bzizd2loCleXdIfHU33mds…

Dead drop: **t53c81TtFdmBCzFLQ7Q**
Message: rCCnMCttJ8C8JMthLxN8…

Dead drop: **pavnHQmuegSmvXz6Y5**
Message: IuA94shFx7okpZdBacjBg…

Fake doubles

Dead drop: **3nPki8GbZWfXRyw61wk**
Message: nE7yvLJLeiCvcD1Cu62…

Dead drop: **3nPki8GbZWfXRyw61wk**
Message: 4QjdRfoB7GoEEb0vtMjf…

Dead drop: **kt2JnceRb7ieU3M1k5Oj**
Message: mb4ZgDABTLTtm9rUZzV…

Dead drop: **kt2JnceRb7ieU3M1k5Oj**
Message: wYNxuyoOiP9Ffjr4LKtv38…

Dead drop: **LWnyE3AB2TTmUcCGL**
Message: k1bVsoTVlJQTEy92Vxd1o…

Dead drop: **LWnyE3AB2TTmUcCGL**
Message: mTLa2cdkKgzADt0oJm8s…

Demo!

Vuvuzela with noise is effective!

Formalizing privacy guarantee

Pr[**i** | Alice **talked** to Bob] $\approx$ **Pr**[**i** | **not** Alice talked to Bob]

(ε,δ) differential privacy, simplified

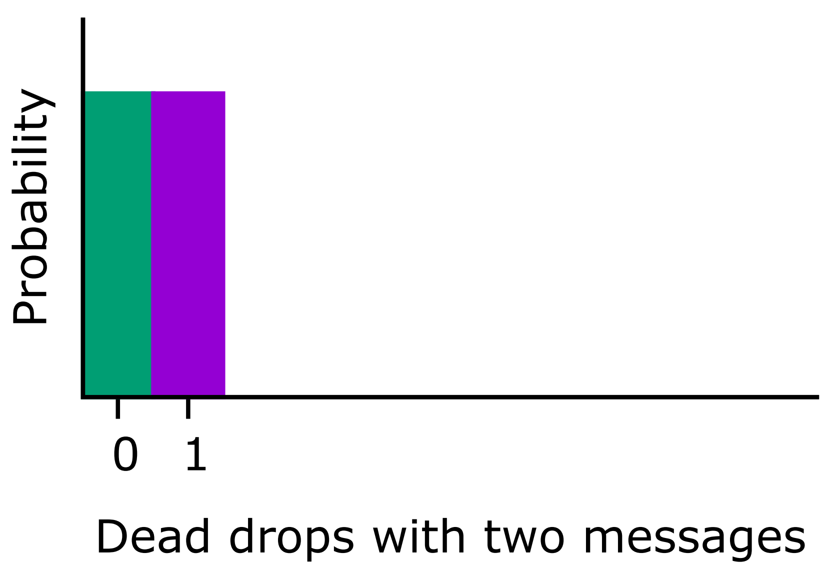
Pr[**i** | Alice **talked** to Bob] $\leq$ $\boldsymbol{\varepsilon} \times$ **Pr**[**i** | **not** Alice talked to Bob]

Noise achieves DP

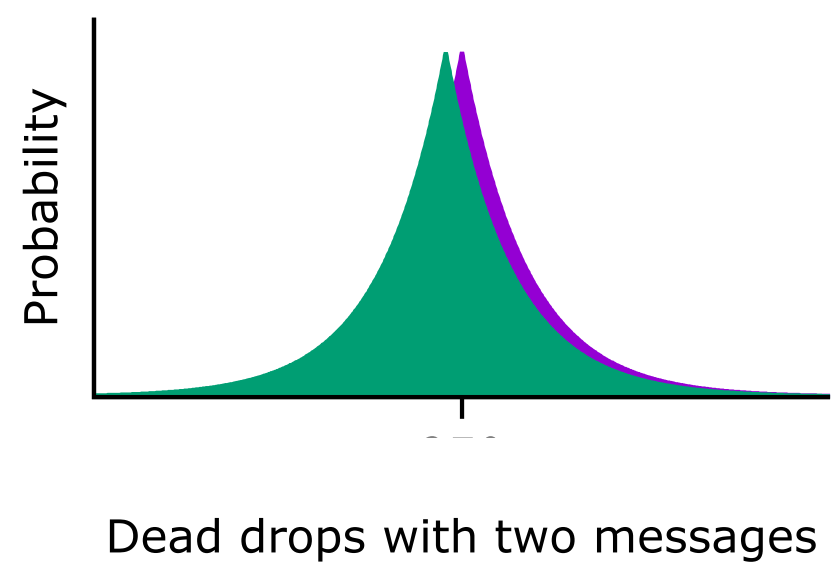
- Let **d** be the number of dead drops with two accesses in a single round.

- To make **d** differentially private, we need to make these distributions very close (indistinguishable):

Pr[**d**=*x* | Alice **talked** to Bob]



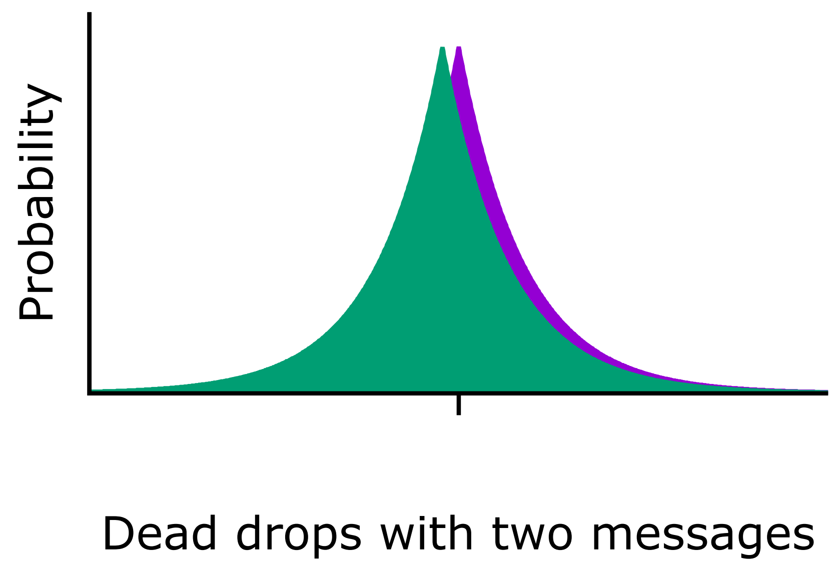
Pr[**d**=*x* | **not** Alice talked to Bob]



Generating this distribution

Pr[**d**=*x* | Alice **talked** to Bob]

Pr[**d**=*x* | **not** Alice talked to Bob]



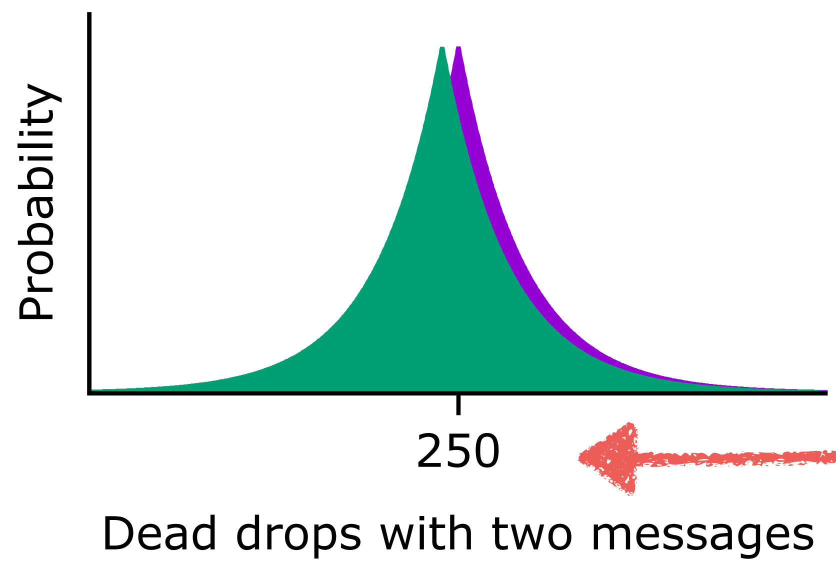
Constraints:

- Can't have negative dead drops
- Distributions have to be close enough for differential privacy

Generating this distribution

Pr[**d**=*x* | Alice **talked** to Bob]

Pr[**d**=*x* | **not** Alice talked to Bob]



Average noise is hundreds of fake messages

Constraints:

- Can't have negative dead drops
- Distributions have to be close enough for differential privacy

Privacy degrades every round

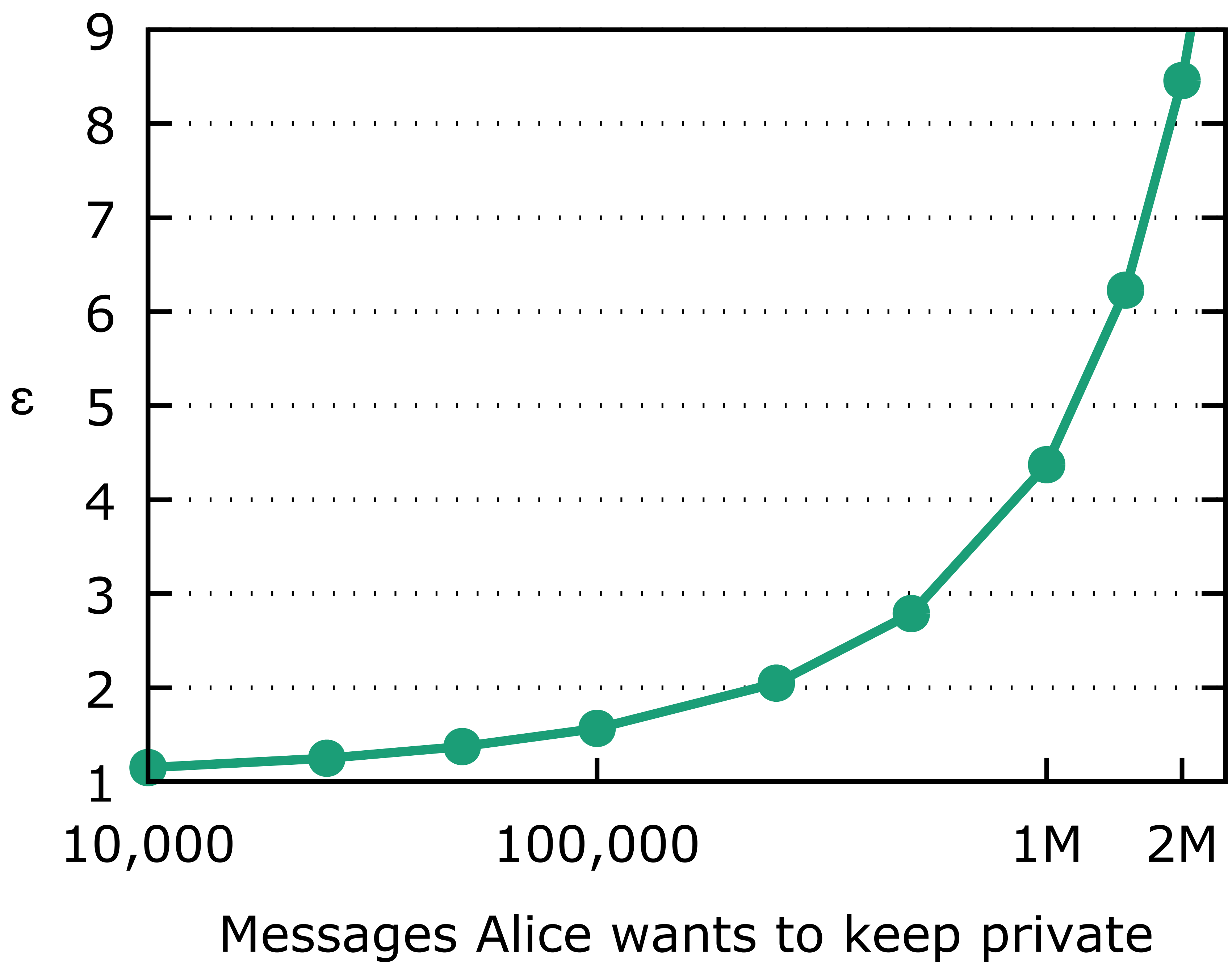
- Each round leaks metadata
- We want differential privacy after sending many messages
- This means adding more noise to support more messages.

Vuvuzela’s approach to noise

- More noise means privacy for more messages.
- Add as much noise as possible, while still keeping the system practical.
- Use differential privacy to compute how much privacy users get.
    - Using composition theorem [Dwork & Roth 2014]
- We picked: 300,000 fake singles and 300,000 fake doubles per server per round.

Privacy with 300,000 noise

Pr[**i** | Alice **talked** to Bob] $\leq$ ***ε*** × **Pr**[**i** | **not** Alice talked to Bob]



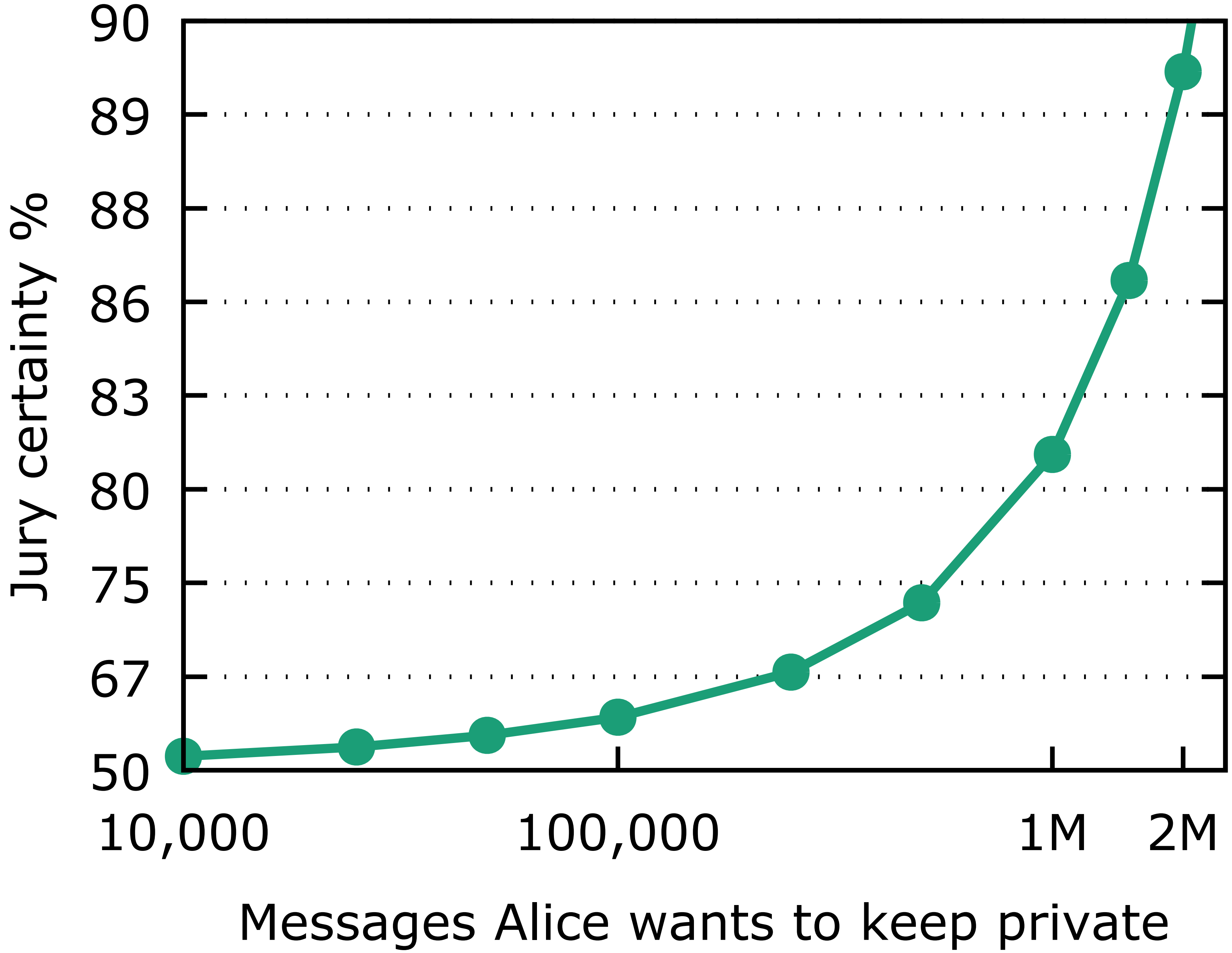
Eve is very evil

- Alice sees previous graph and sends Eve many messages through Vuvuzela.
- Will NSA arrest Alice for talking to Eve?
  - Probably: using Vuvuzela is already suspicious
- Will a fair jury convict Alice of talking to Eve?
  - Unlikely: Vuvuzela observations are not damning evidence!

Alice gets a fair trial

- Jury is already 50% certain Alice did the crime (NSA is intimidating, other evidence, etc)
- Beyond unreasonable doubt = 90% certainty

Alice is innocent for millions of messages



Implementation

- 3,000 lines of Go
- Untrusted entry server manages user connections
- Entry server notifies clients when a new round starts
- Available soon on Github:
  - github.com/davidlazar/vuvuzela

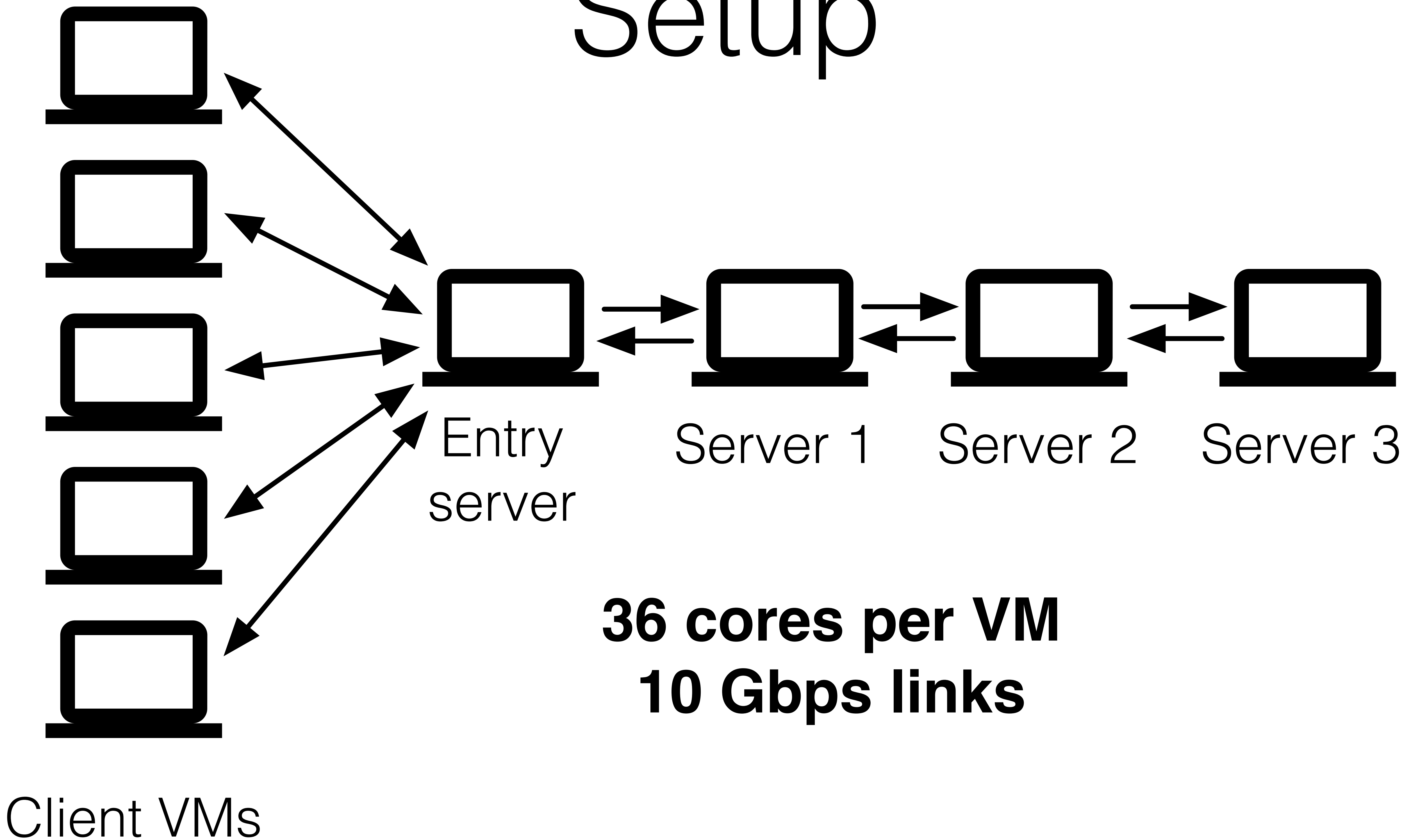
Evaluation

- Can Vuvuzela servers support a large number of users and messages?
- Does Vuvuzela provide acceptable performance?

Asymptotic performance

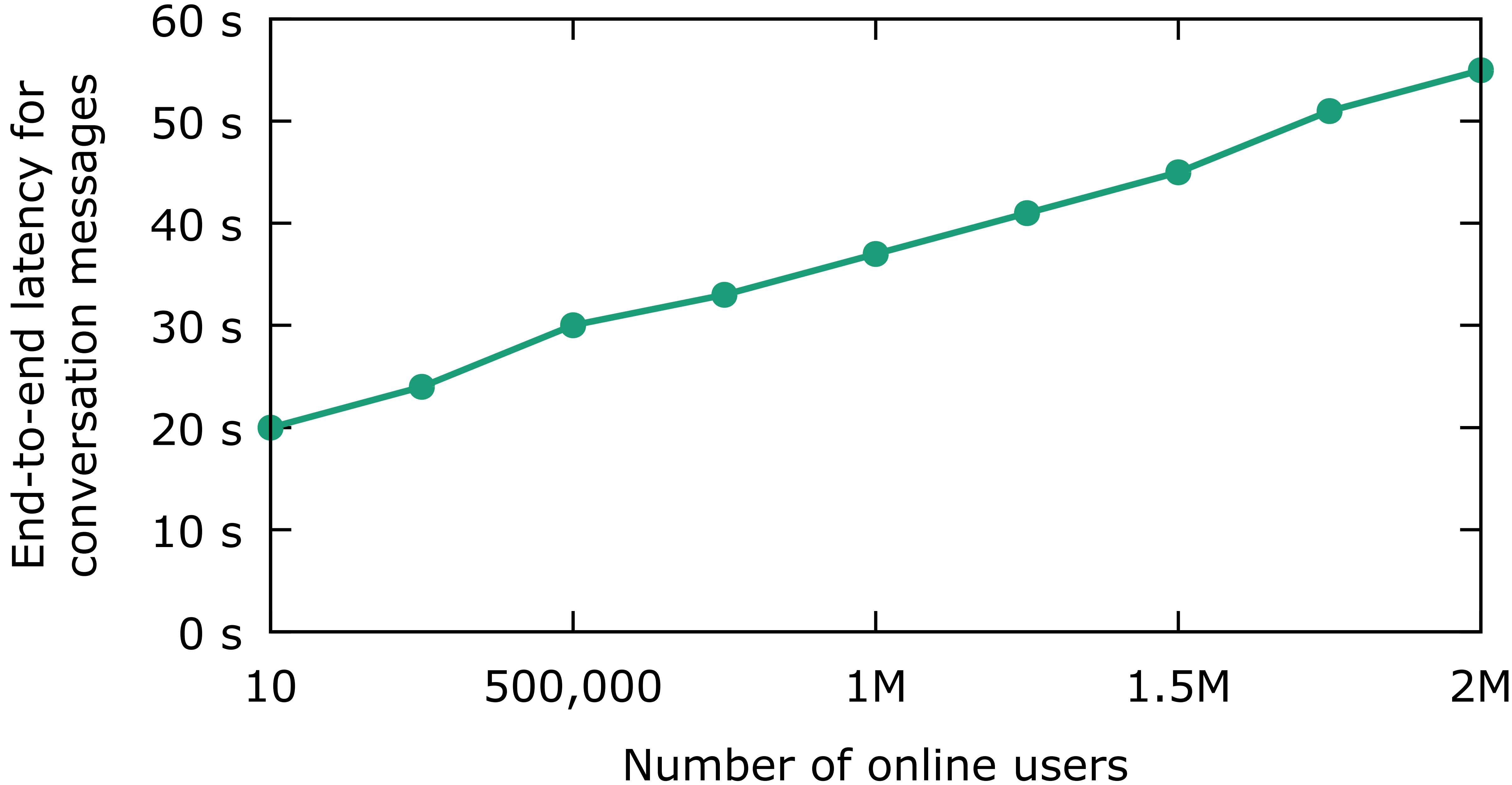
- Noise is independent of number of users.
- Performance is linear in number of users
  - Bandwidth, latency, CPU

Setup



36 cores per VM
10 Gbps links

Acceptable end-to-end latency for text messaging



Performance bottlenecks

- CPU bound
  - Dominated by mixnet operations
- High bandwidth cost
  - 166 MB/s for servers, 12 KB/s for clients
  - Can lower bandwidth by increasing latency linearly

Conclusion

- **Problem:** hide metadata in a secure and scalable way.
- **Approach:**
  - Encrypt as much metadata as possible.
  - Add noise to obscure remaining metadata.
  - Formalized privacy guarantee with differential privacy
- **Vuvuzela:** scalable private messaging without metadata
  - Scales linearly with number of users
  - Privacy for millions of messages per user → 37s latency
  - 60,000 messages / second of throughput

What happens after 2M?

- Privacy for lifetime of messages is unrealistic under this configuration
- User’s should change their expectation to just expect privacy for a subset of messages
  - Example: privacy just for important messages.
  - Example: privacy just for recent messages.
- User does not need to specify which subset of messages to keep private
  - Vuvuzela’s guarantee holds for any (small) subset of messages that the adversary cares about