Groove: Flexible Metadata-Private Messaging

Ludovic Barman, Moshe Kol, David Lazar,
Yossi Gilad, Nickolai Zeldovich

EPFL, Hebrew University of Jerusalem, MIT CSAIL

OSDI'22

Metadata-Private Communications

I can't read the contents, but I can see who Alice is contacting !

Categories of Related Work

	Resist traffic-analysis	**Scalability**	**Asynchronous clients**
Tor	no	O(users)	yes
Private Information Retrieval	yes	O(users^2)	yes
Mixnets with circuits	yes	O(users)	**no**

Mixnets: Impractical for smartphones

- Device must be always on and sending cover traffic at a fixed rate
  - Battery concerns
  - Bandwidth usage concerns
  - Messages are lost if the client is offline

- This doesn't match the user experience of standard messaging apps :

Groove (our system)

	Resist traffic-analysis	Scalability	Asynchronous clients
Tor	no	O(users)	yes
Private Information Retrieval	yes	O($users^2$)	yes
Mixnets with circuits	yes	O(users)	**no**
Groove (our system)	yes	O(users)	**yes**

Contributions

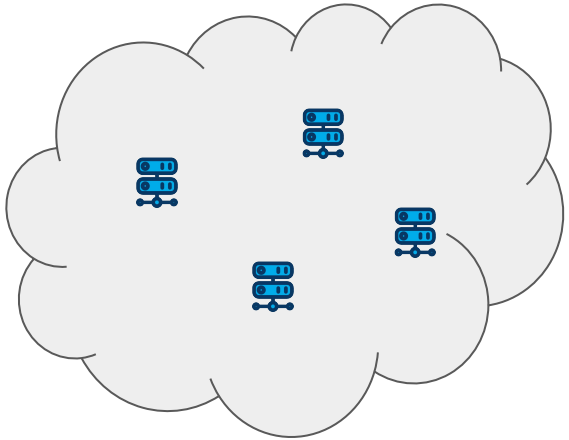
- Groove removes the rigid/synchronous requirement of large-scale mixnets
  - Users don't have to be online at the same time (focus of this talk)
  - Users can have multiple devices

- It supports mobile devices: ~100MB / month, flexible energy usage

- New techniques:
  - Non-interactive circuit setup (focus of this talk) for multiple devices
  - Oblivious protocol to avoid downloading cover traffic
  - New key-rolling mechanism for forward and future secrecy

Model

Honest sender



Groove

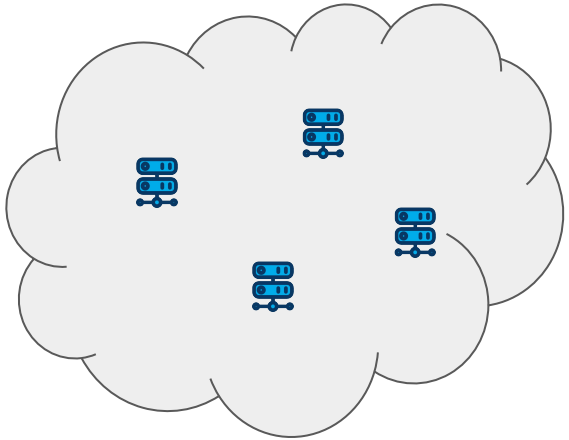
Honest recipient

Global Active Adversary

- observes all network communication
- can edit/drop/inject/delay any message
- controls a fraction of the servers (e.g., 80%)

Model

Honest sender



Groove

Honest recipient

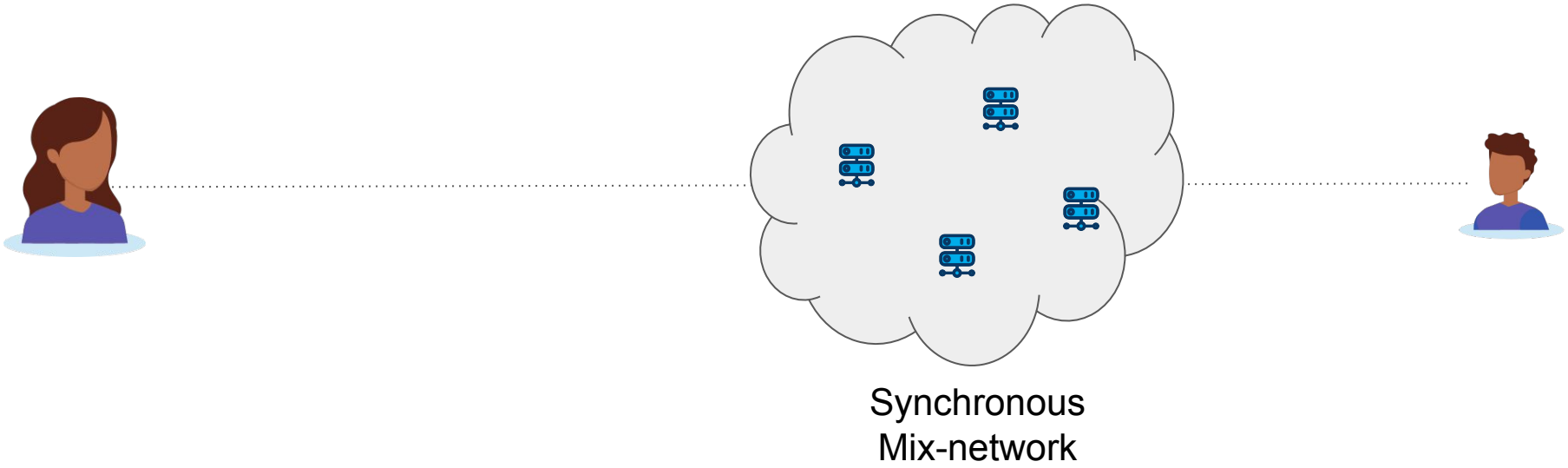
Global Active Adversary

- observes all network communication
- can edit/drop/inject/delay any message
- controls a fraction of the servers (e.g., 80%)

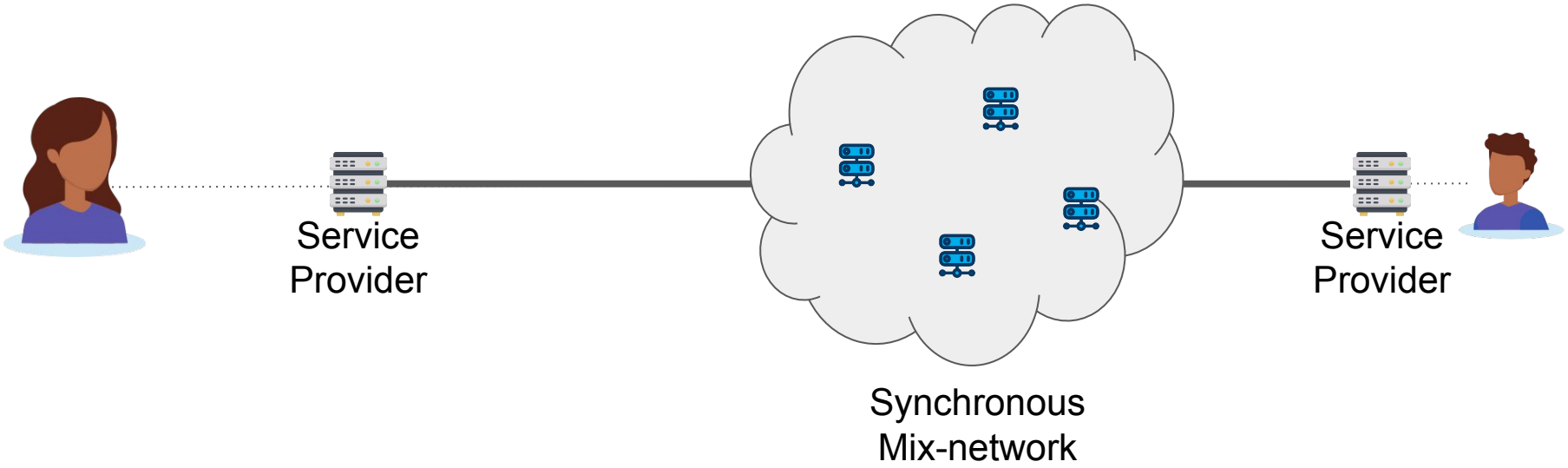
Security notion:

$$obs_A(Alice \leftrightarrow Bob) \cong \; / \; obs_A(Alice \leftrightarrow Bob)$$

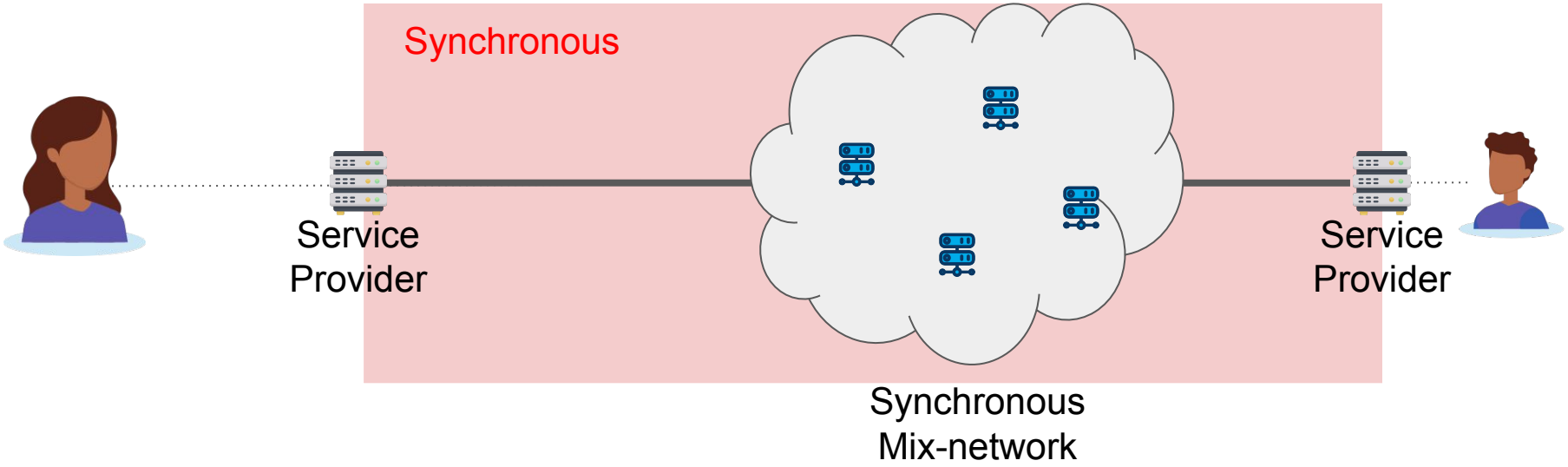
Enabling asynchronous clients



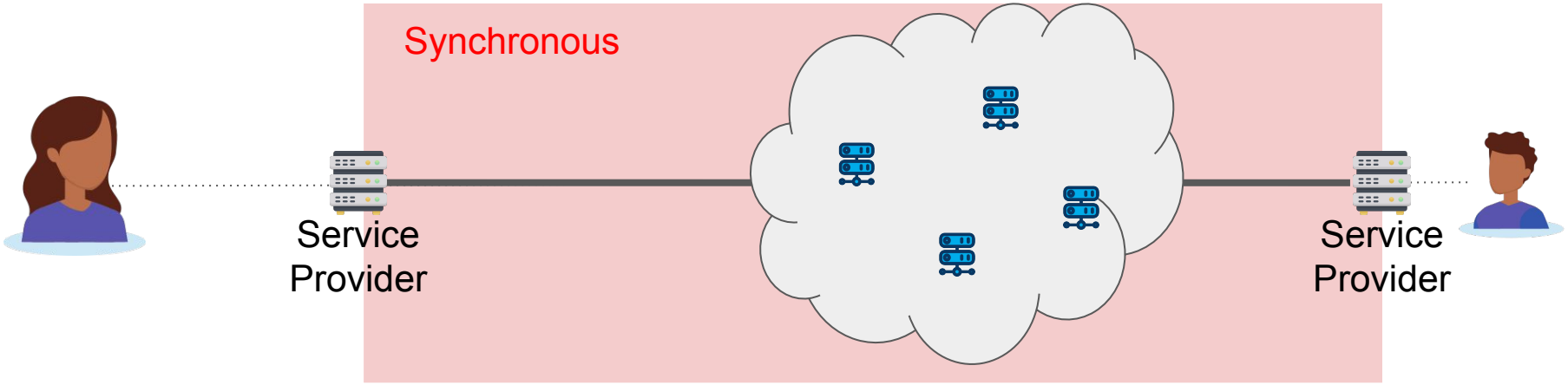
Enabling asynchronous clients



Enabling asynchronous clients

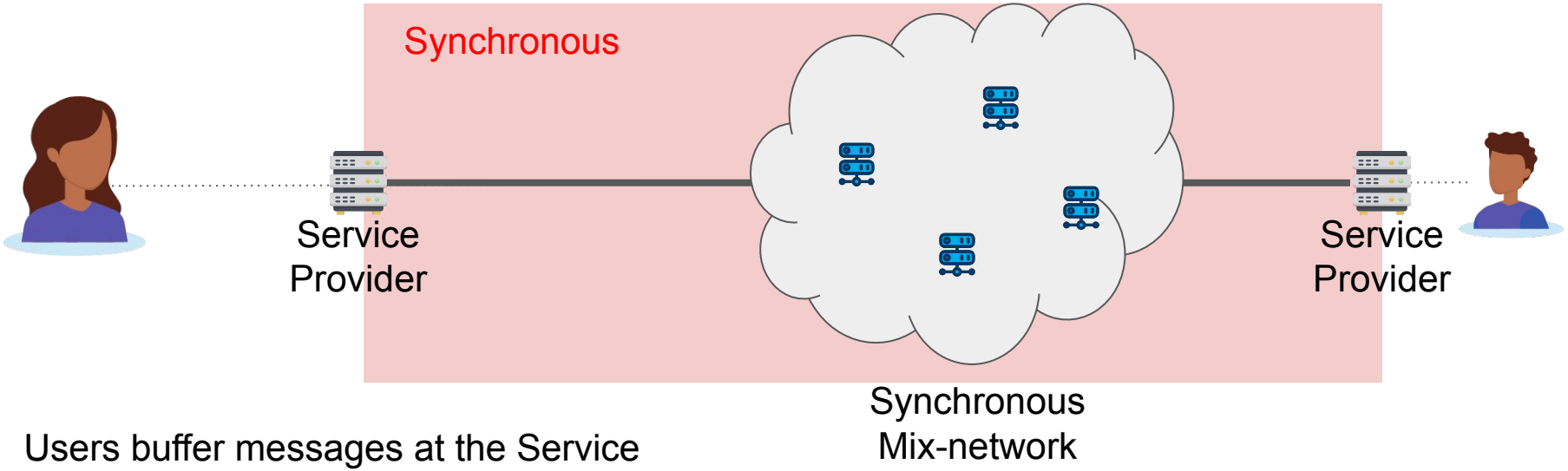


Service providers enable asynchronous clients



Users buffer messages at the Service Provider, then can disconnect.

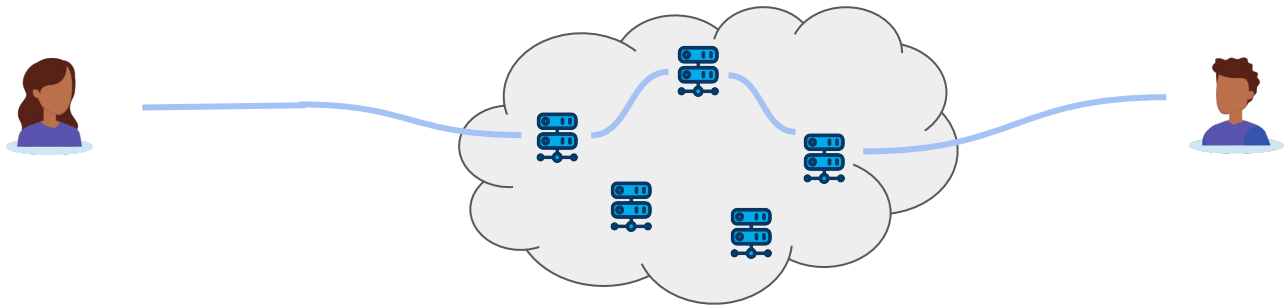
Service providers enable asynchronous clients



Users buffer messages at the Service Provider, then can disconnect.

Challenge: how to make this service provider untrusted

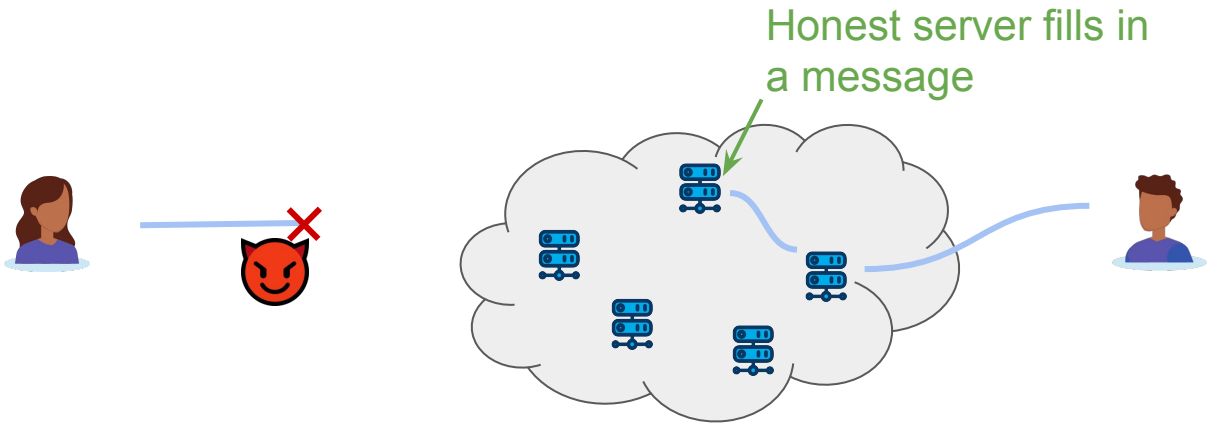
Background: Circuits [1]



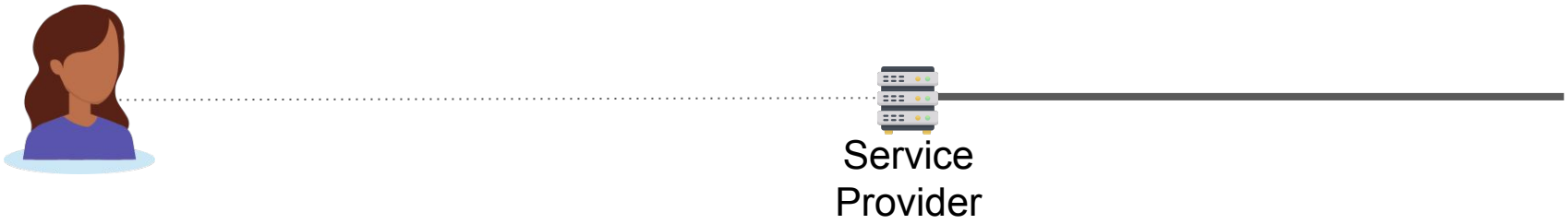
- Long-lived connection between a sender and a recipient
- The path is chosen by the sender
- Once setup, it carries messages at a constant rate

[1] D. Lazar, Y. Gilad, N. Zeldovich. **Yodel: strong metadata security for voice calls.** SOSP'19

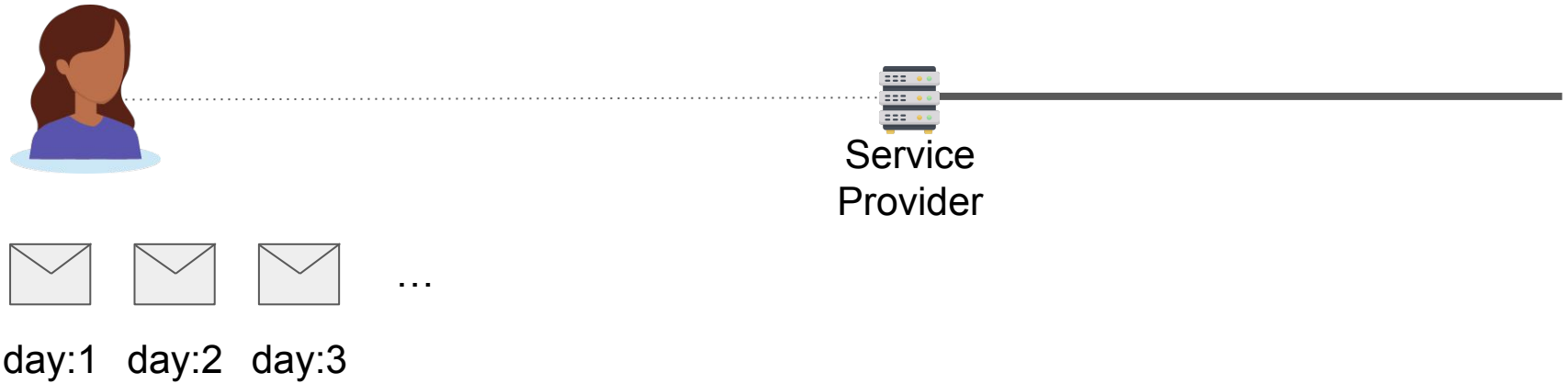
Background: Once setup, circuits resist active attacks



Groove: Pre-computed circuit setup messages

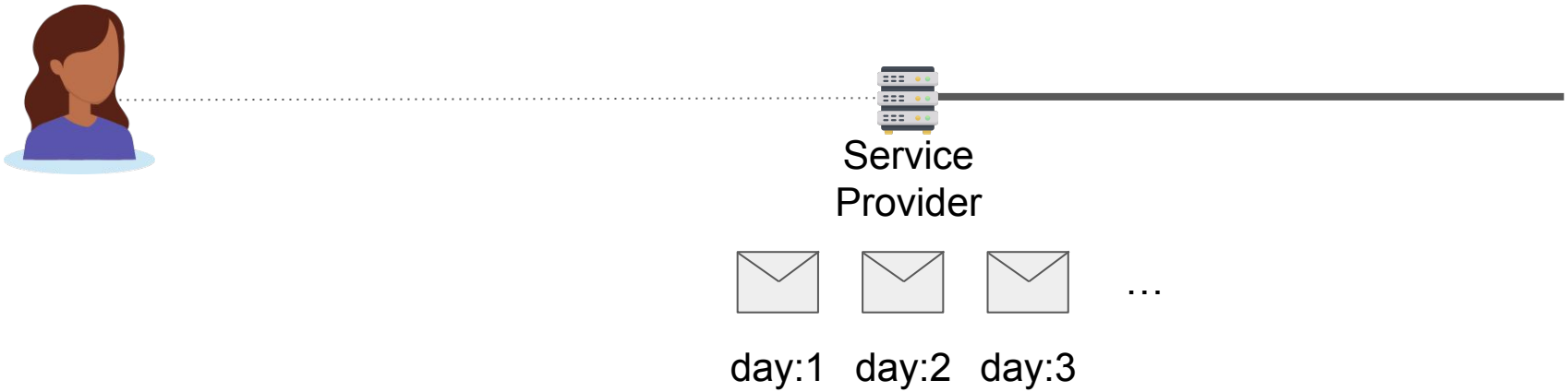


Groove: Pre-computed circuit setup messages



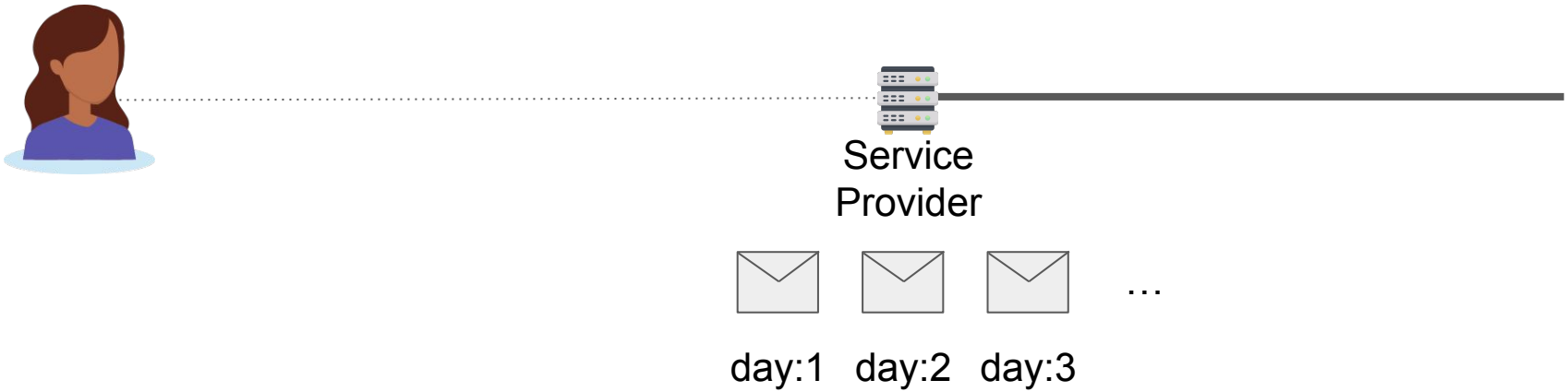
- Alice pre-computes all messages for a month

Groove: Pre-computed circuit setup messages



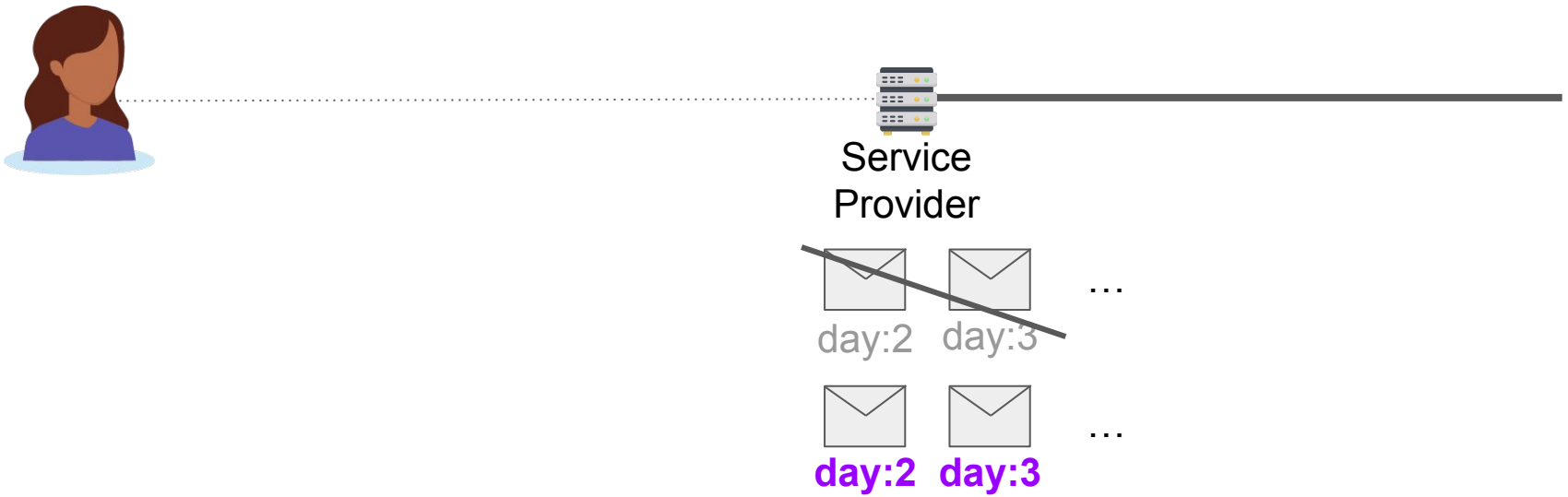
- Alice pre-computes all messages for a month
- She buffers them at the SP, which replays them when needed

Groove: Pre-computed circuit setup messages



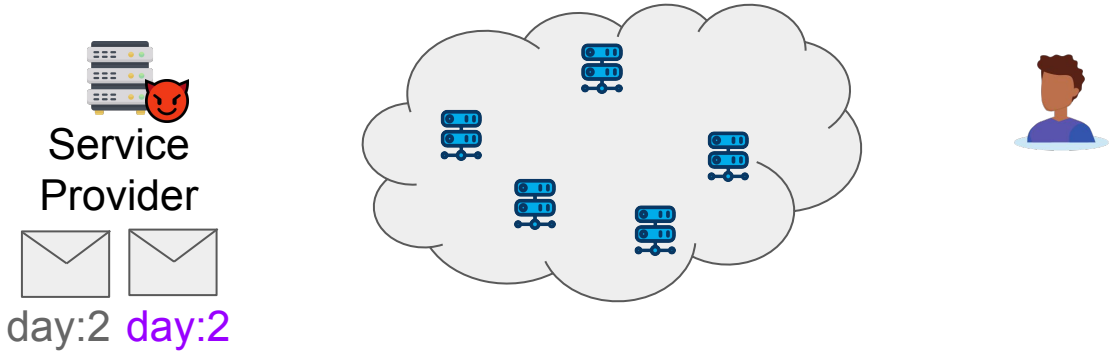
Problem: How can Alice contact a new friend during this month ?

Approach: replacing messages

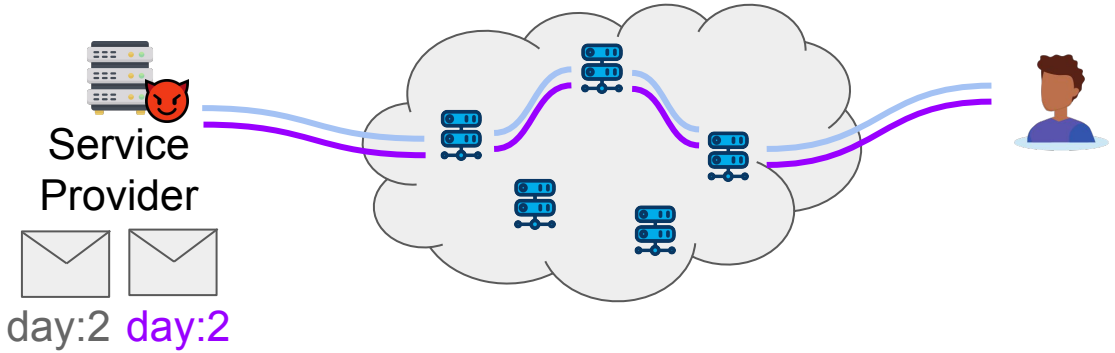


- Every day, Alice uploads **new** setup messages for the next month

Attack on message replacement

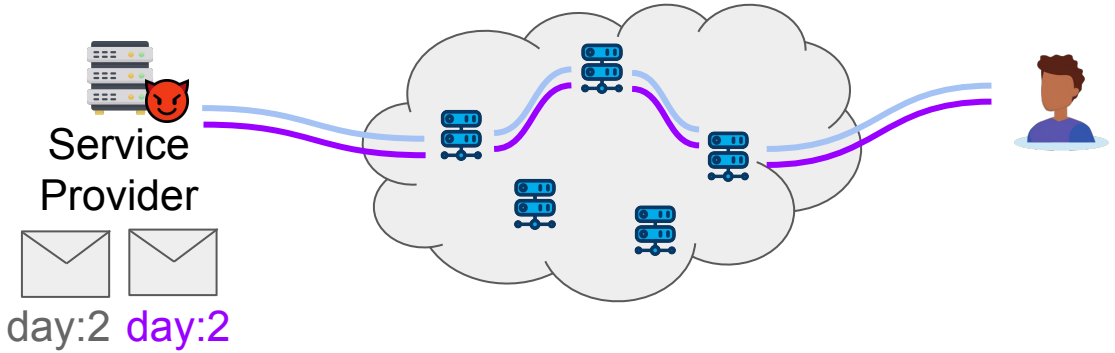


Attack on message replacement



Service provider replays **both** old and new version for day 2

Attack on message replacement

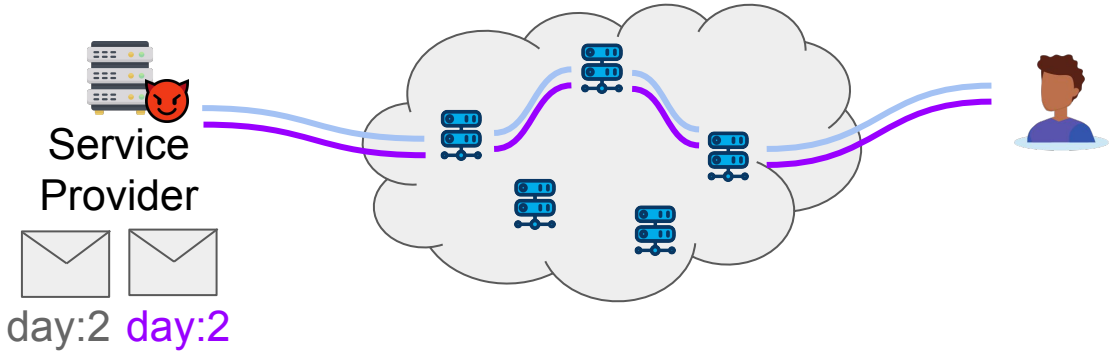


Service provider replays **both** old and new version for day 2

Bob receives two messages !

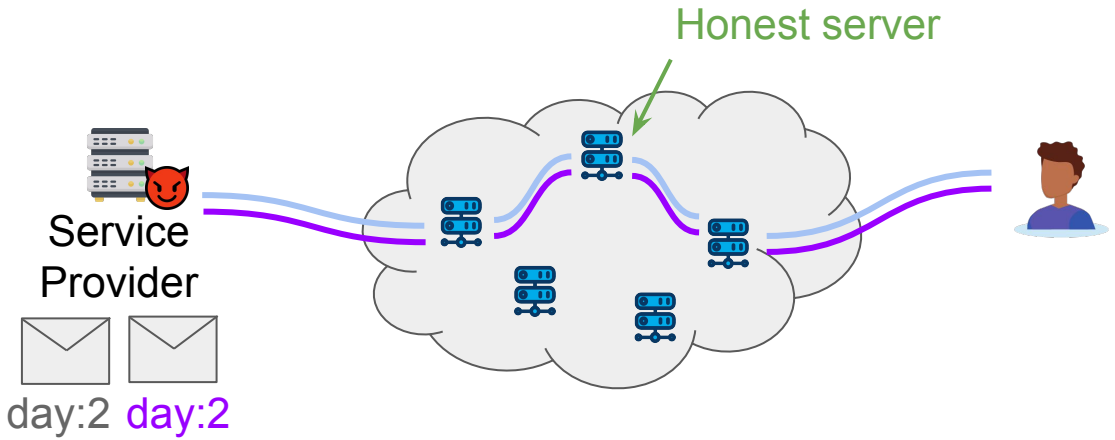
→ Alice is talking with Bob !

Groove: Safe message replacement



Insight 1: Circuit paths are pseudo-random and independent of the correspondent

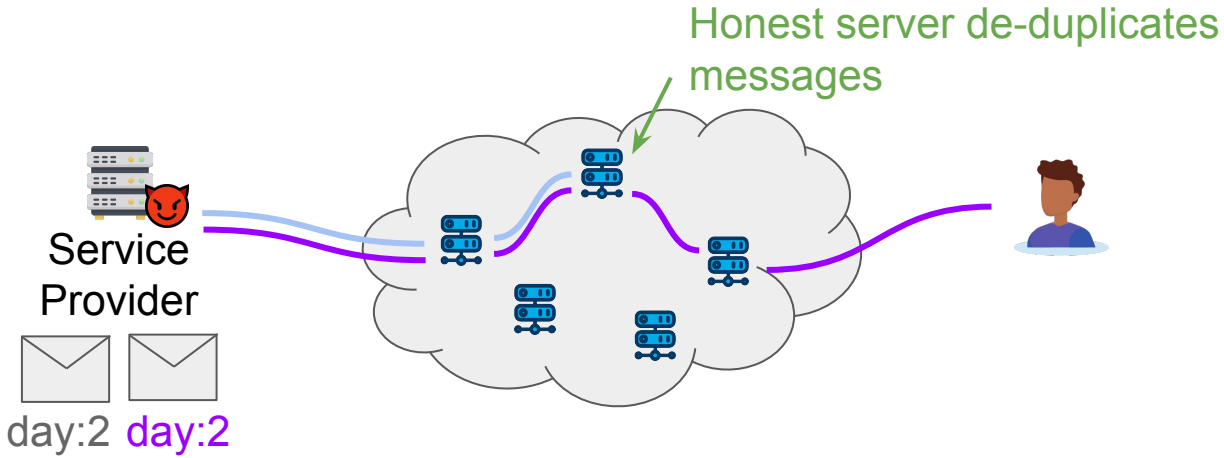
Groove: Safe message replacement



Insight 1: Circuit paths are pseudo-random and independent of the correspondent

Insight 2: SP is malicious, but there is an honest server on the path to Bob !

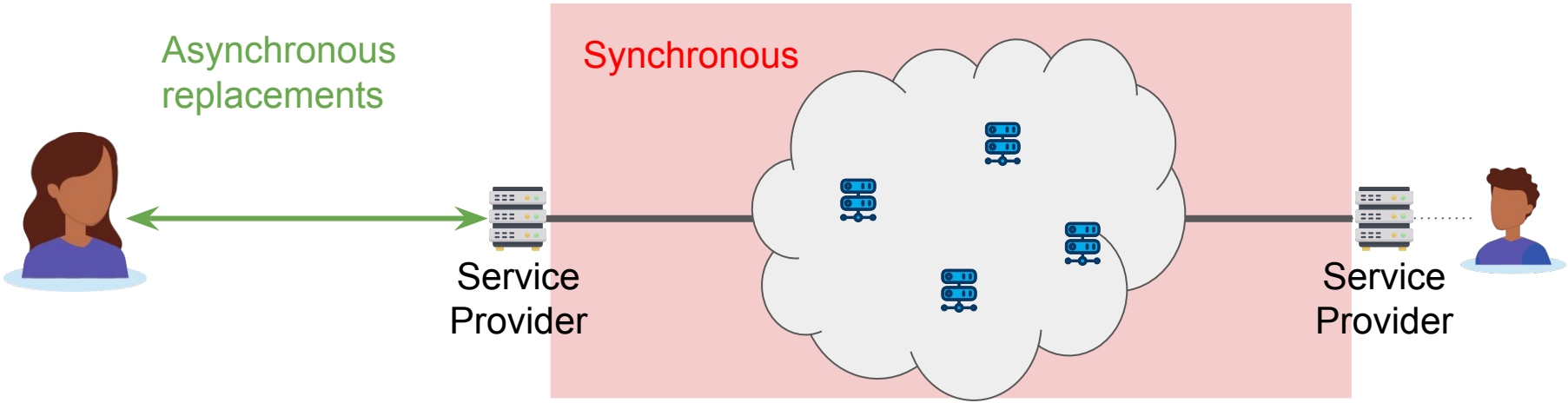
Groove: Safe message replacement



Insight 1: Circuit paths are pseudo-random and independent of the correspondent

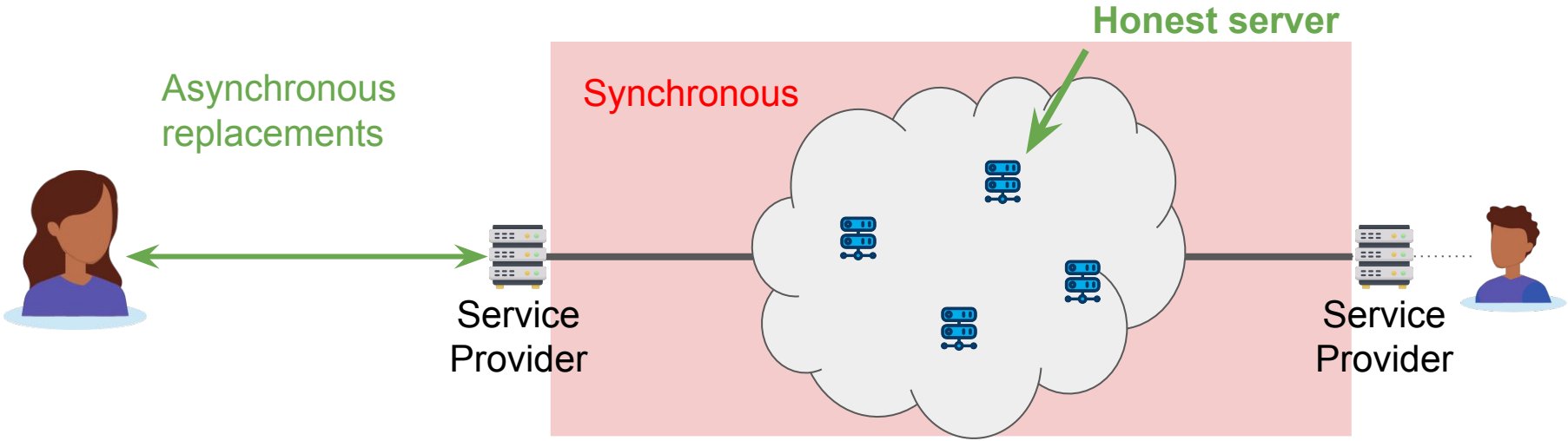
Insight 2: SP is malicious, but there is an honest server on the path to Bob !
It de-duplicates circuit setup messages.

"Primed circuits" =
Non-interactive setup + Safe replacement



1. Alice precomputes messages for a month
2. Alice can replace messages at <u>any time</u>

"Primed circuits" =
Non-interactive setup + Safe replacement



1. Alice precomputes messages for a month
2. Alice can replace messages at <u>any time</u>

<u>Insight:</u> This is made safe by the 1st honest server on the circuit path

See the paper for the other contributions

- Multi-devices for the same user, even if they cannot coordinate
- Clients can retrieve only messages of interest from the Service Provider
- Future secrecy to prevent one old device from compromising others

Implementation slide

- Implemented in Go, on top of Yodel's [1] mixnet
- 20k lines of code
- TLS 1.3 for transport security
- gRPC between client/mixnet/service provider
- Android mobile client built using gomobile

[1] D. Lazar, Y. Gilad, N. Zeldovich. **Yodel: strong metadata security for voice calls.** SOSP'19

Evaluation questions

- What's the latency of the system ?
- What is the phone bandwidth usage ?
- What is the energy consumption of the phone ?

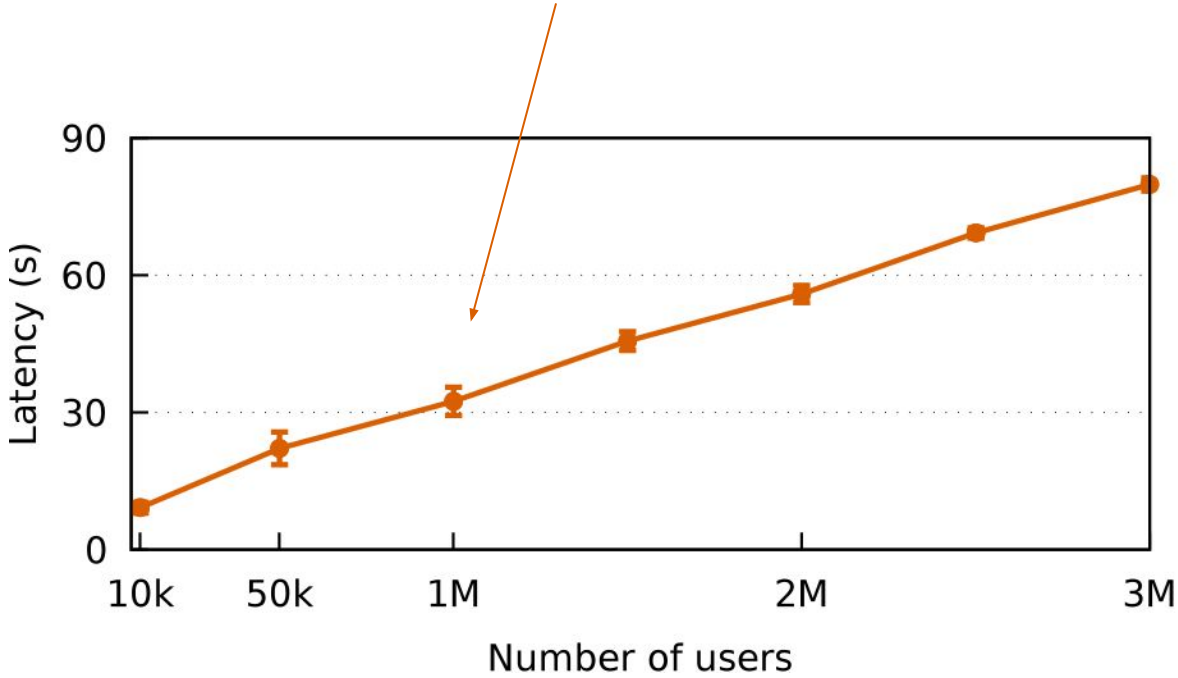
See the paper for more evaluation questions (e.g., on the backend)

Experimental setup

- client: Pixel 4 phone
- 100 servers on AWS in 4 regions (US + EU)
- each server is a 32 core 3.1Ghz CPU, 256 GB RAM, 10 Gbps network
- We simulate up to 3 Mio users with 50 friends each

End-to-end latency is between 32s and 2min

- Client connects every 1min to the service provider (user-chosen)
- Backend processes messages in **32s** for 1mio users with 50 friends each



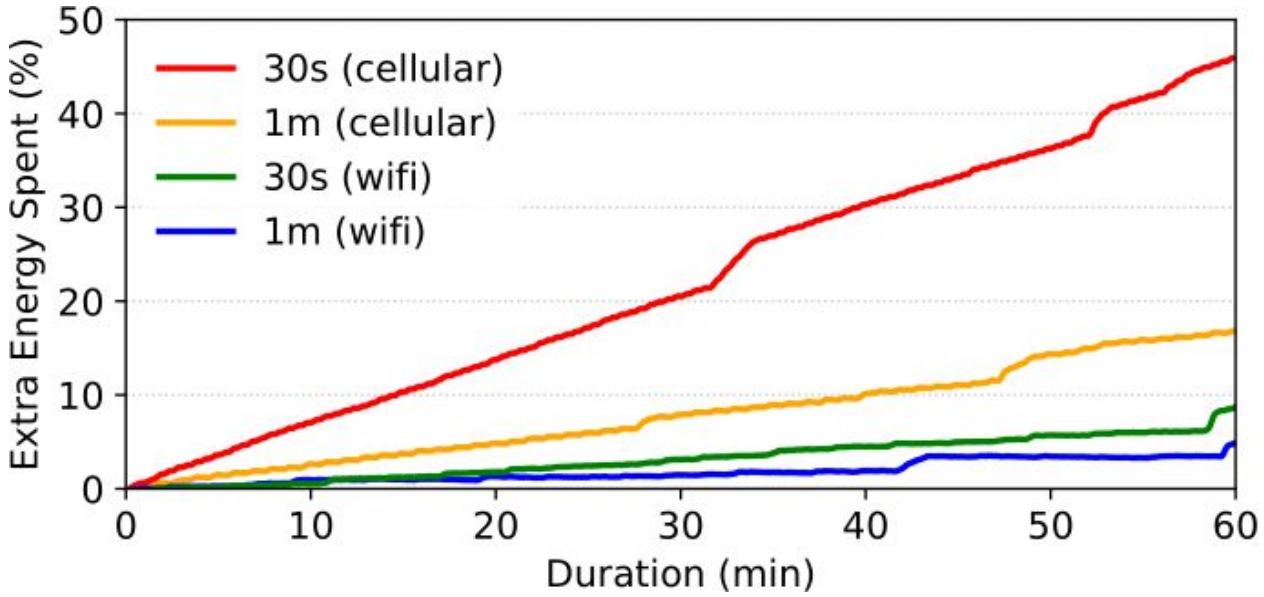
The phone uses ~106MB per month

- Circuit setup messages amount for **3 MB / month**
- For a phone connecting to the SP every minute, messaging is **~103 MB / month**

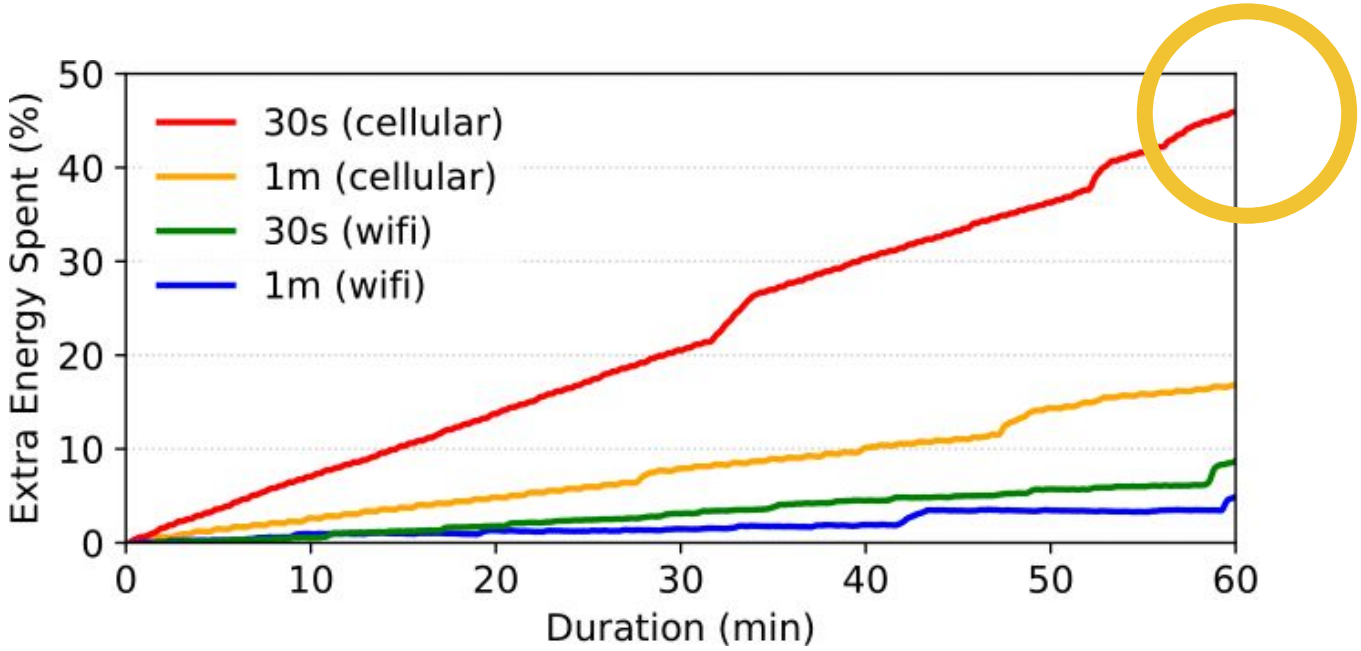
Many related works rely on Alpenhorn's [2], which uses 62 GB/month (600x more)

[2] D. Lazar, N. Zeldovich. **Alpenhorn: Bootstrapping Secure Communication Without Leaking Metadata.** OSDI'16

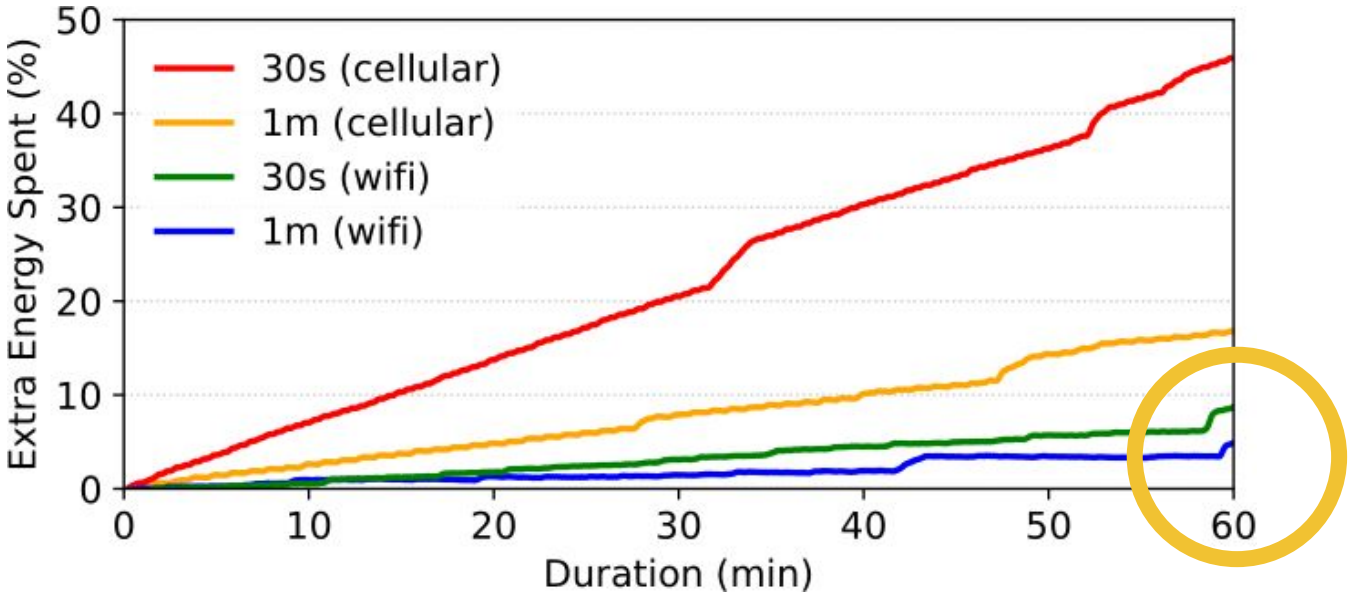
Energy usage compared to an idle phone



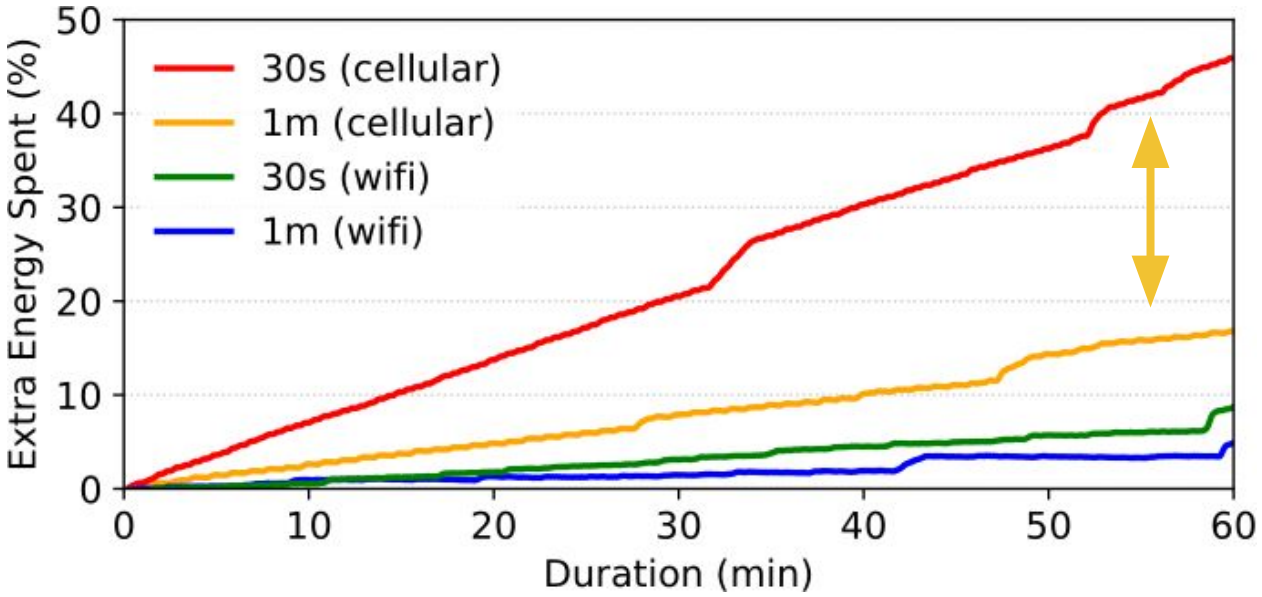
Prior work: fast schedules might drain the battery



On Wi-Fi, extra energy spent is <10%



Clients can dynamically change their schedule



Conclusion

- Groove is a metadata-private communication system
- It removes the rigid/synchronous requirement of large-scale mixnets
- It supports mobile devices:
  - Clients don't need to follow a global schedule
  - Devices can go offline

  } Flexible energy usage

  - ~106 MB / month, 32s-2min latency
- It brings metadata-private communication closer to standard messaging apps

Signal