Stadium

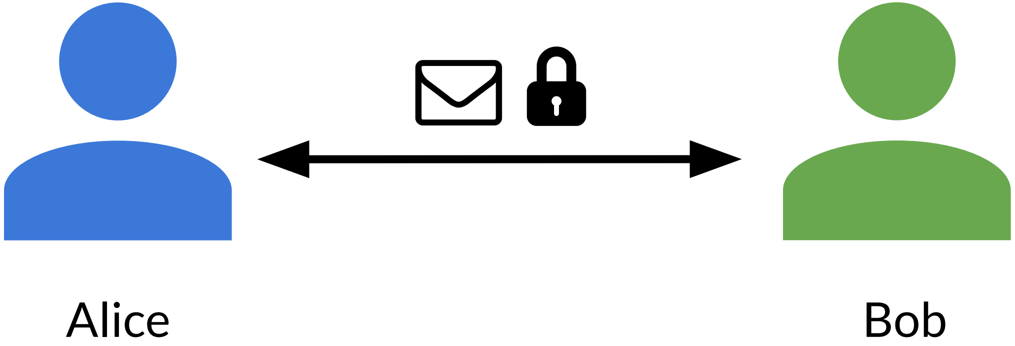
A Distributed Metadata-private Messaging System

Nirvan Tyagi Yossi Gilad Derek Leung
Matei Zaharia Nickolai Zeldovich

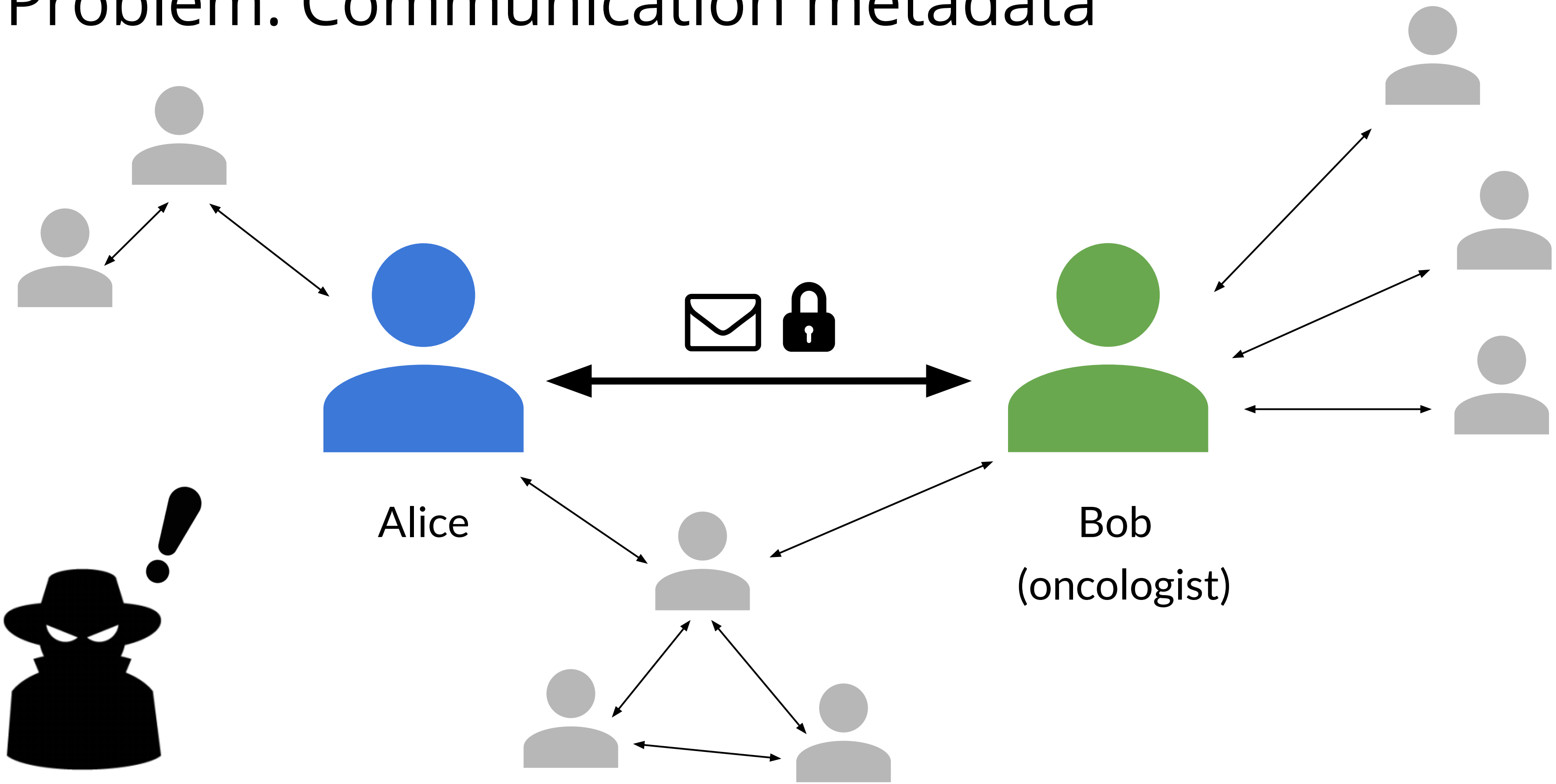
SOSP 2017

Previous talk: Anonymous broadcast

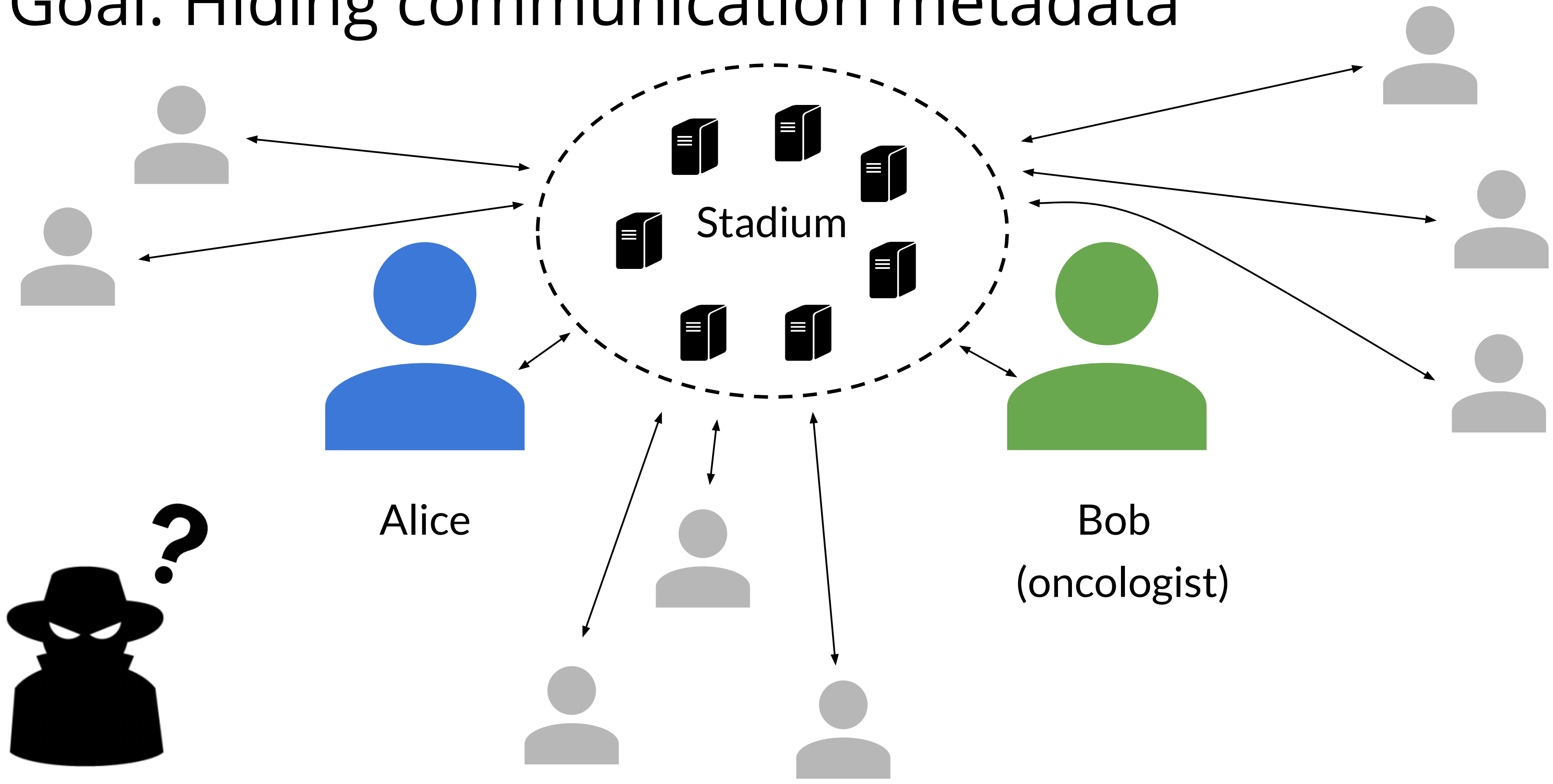
This talk: Private messaging



Problem: Communication metadata



Goal: Hiding communication metadata



Related work

Metadata-private systems with **cryptographic security** limited in throughput.

Dissent [OSDI'12] , Riposte [S&P'15]

~ 1.5 - 65 K messages / min

Pung [OSDI'16] , Atom [SOSP'17]

Related work

Metadata-private systems with **cryptographic security** limited in throughput.

Dissent [OSDI'12] , Riposte [S&P'15]

~ 1.5 - 65 K messages / min

Pung [OSDI'16] , Atom [SOSP'17]

Throughput increased by relaxing guarantees to **differential privacy**.

Vuvuzela [SOSP'15] ~ 2 M messages / min

Related work

Metadata-private systems with **cryptographic security** limited in throughput.

Dissent [OSDI'12] , Riposte [S&P'15]

Pung [OSDI'16] , Atom [SOSP'17]

~ 1.5 - 65 K messages / min

Throughput increased by relaxing guarantees to **differential privacy**.

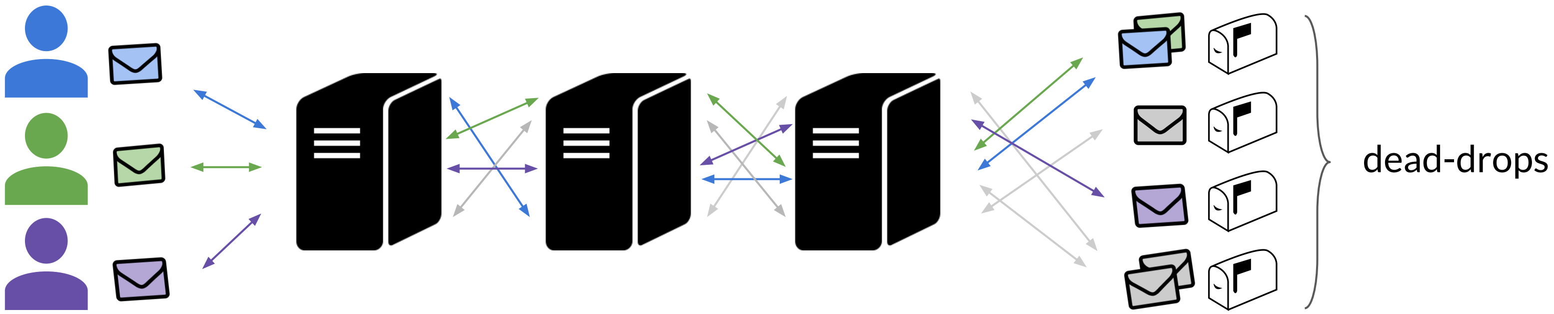
Vuvuzela [SOSP'15] ~ 2 M messages / min

Stadium [SOSP'17] **> 10 M messages / min**

First metadata-private messaging system to scale horizontally

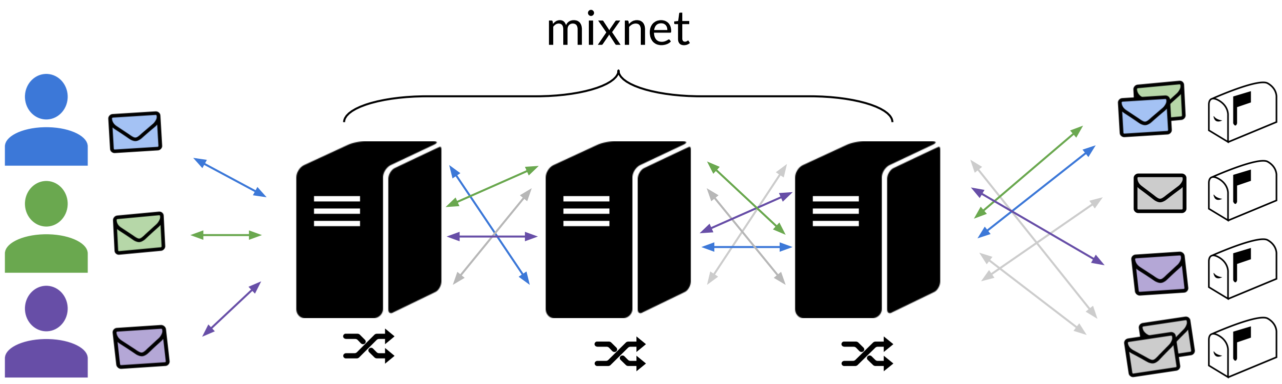
Vuvuzela: Differentially private messaging

- Dead-drops: virtually hosted addresses at which user messages are exchanged



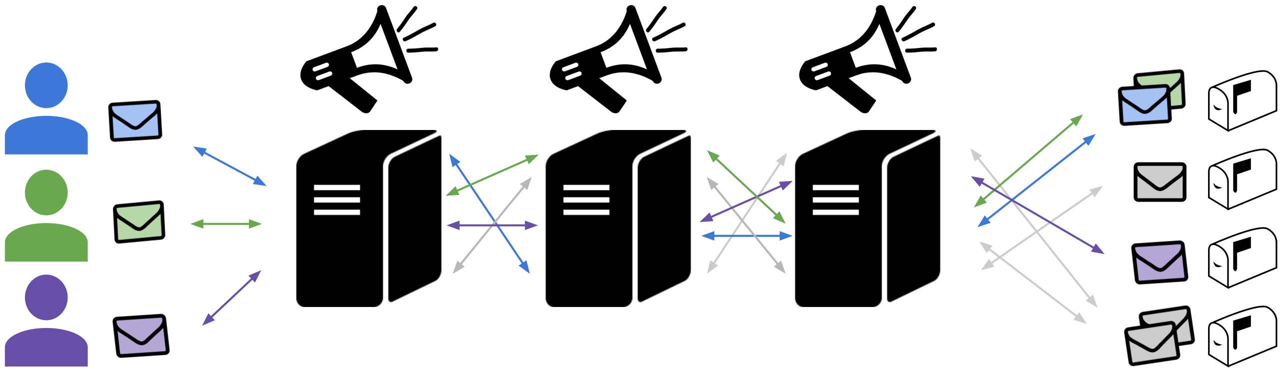
Vuvuzela: Differentially private messaging

- Dead-drops: virtually hosted addresses at which user messages are exchanged
- Mixnet: servers re-randomize and permute messages



Vuvuzela: Differentially private messaging

- Dead-drops: virtually hosted addresses at which user messages are exchanged
- Mixnet: servers re-randomize and permute messages
- Noise: servers add fake messages to obscure adversary observations

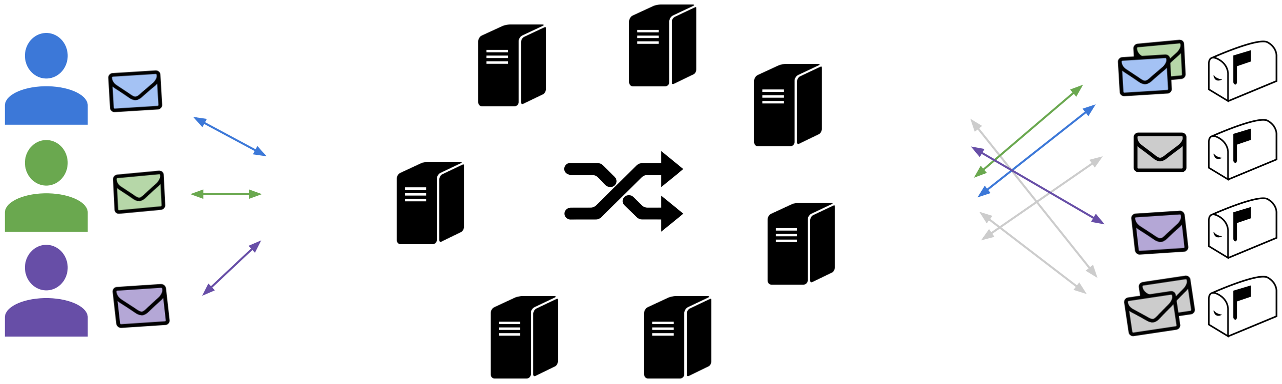


Scaling limitations

- Every server handles all messages
- Running a server is expensive (e.g. 2M users / minute = 1.3 Gbps)

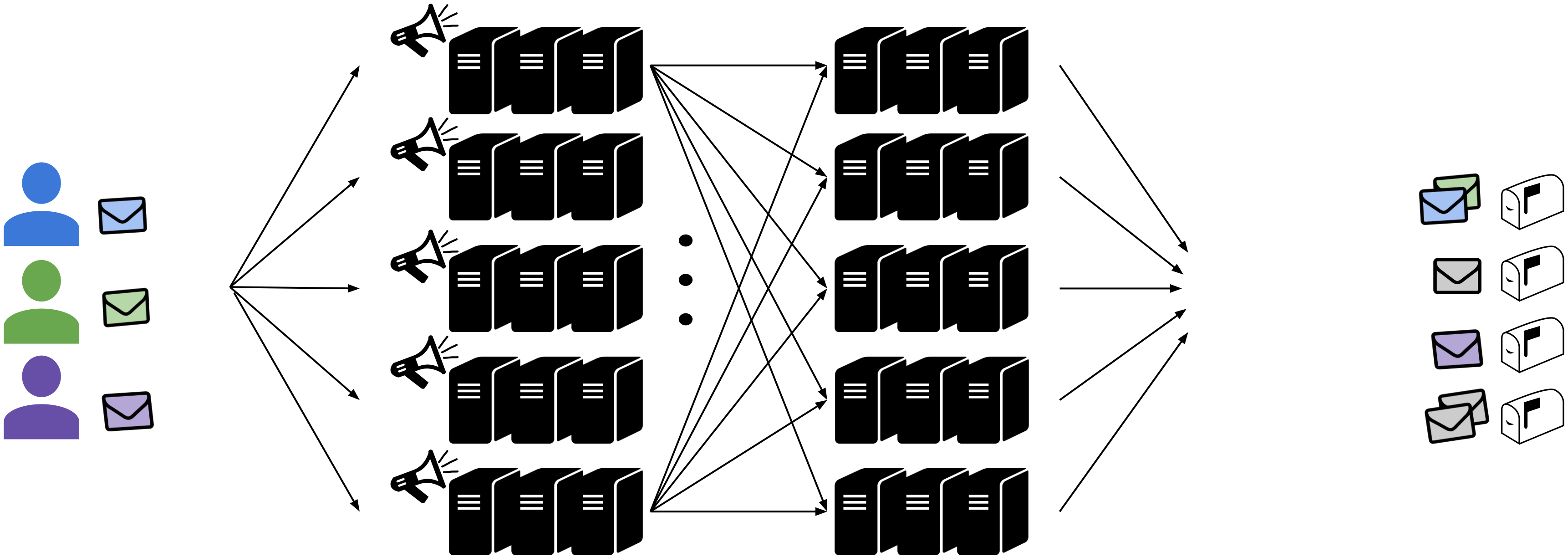
Challenge: How to distribute workload across untrustworthy servers?

1. How to mix messages?
2. How to add noise?



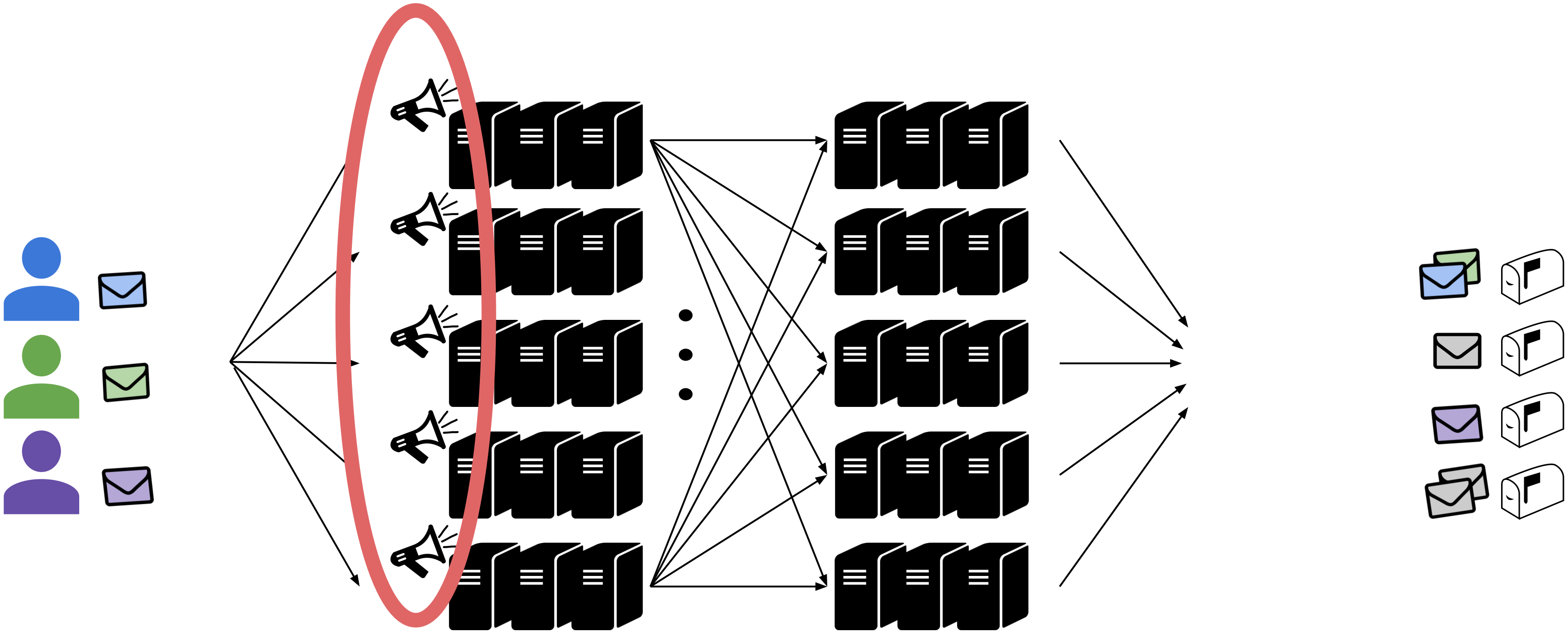
Stadium design

Collaborative noise generation + verifiable parallel mixnet



Stadium design

Collaborative noise generation + verifiable parallel mixnet



Stadium design

Collaborative noise generation + verifiable parallel mixnet

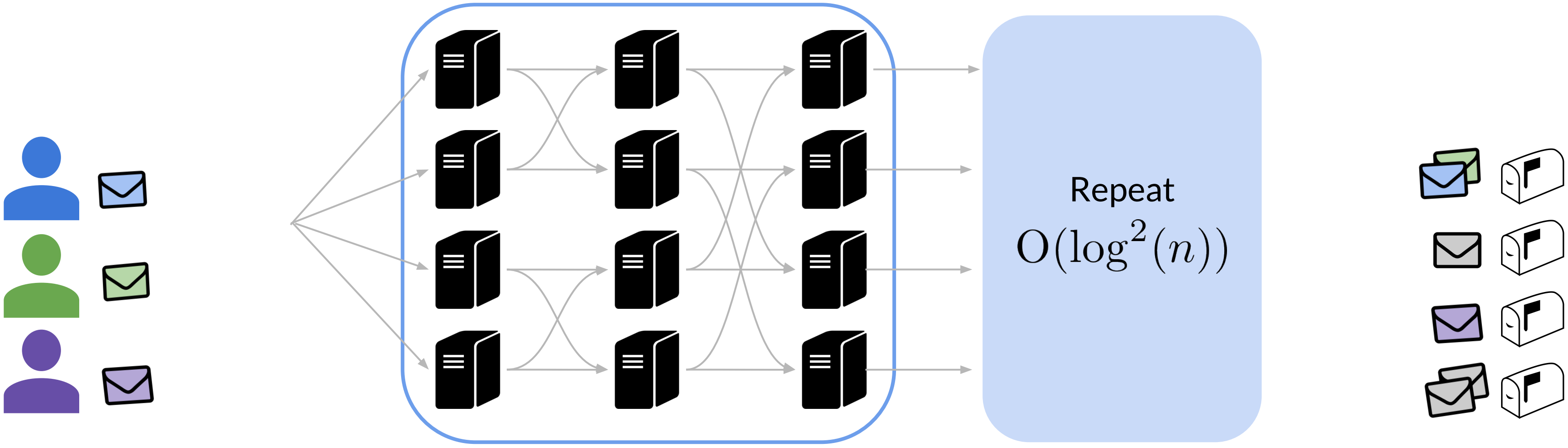
Contributions

- Stadium design
  - Parallel mixnet
  - Collaborative noise generation
  - Verifiable processing including fast zero-knowledge proofs of shuffle
- Multidimensional differential privacy analysis
- Implementation and evaluation of prototype

10 M messages/min with per-server costs of ~100 Mbps

Parallel mixnets with cryptographic security of mixing have large depth.

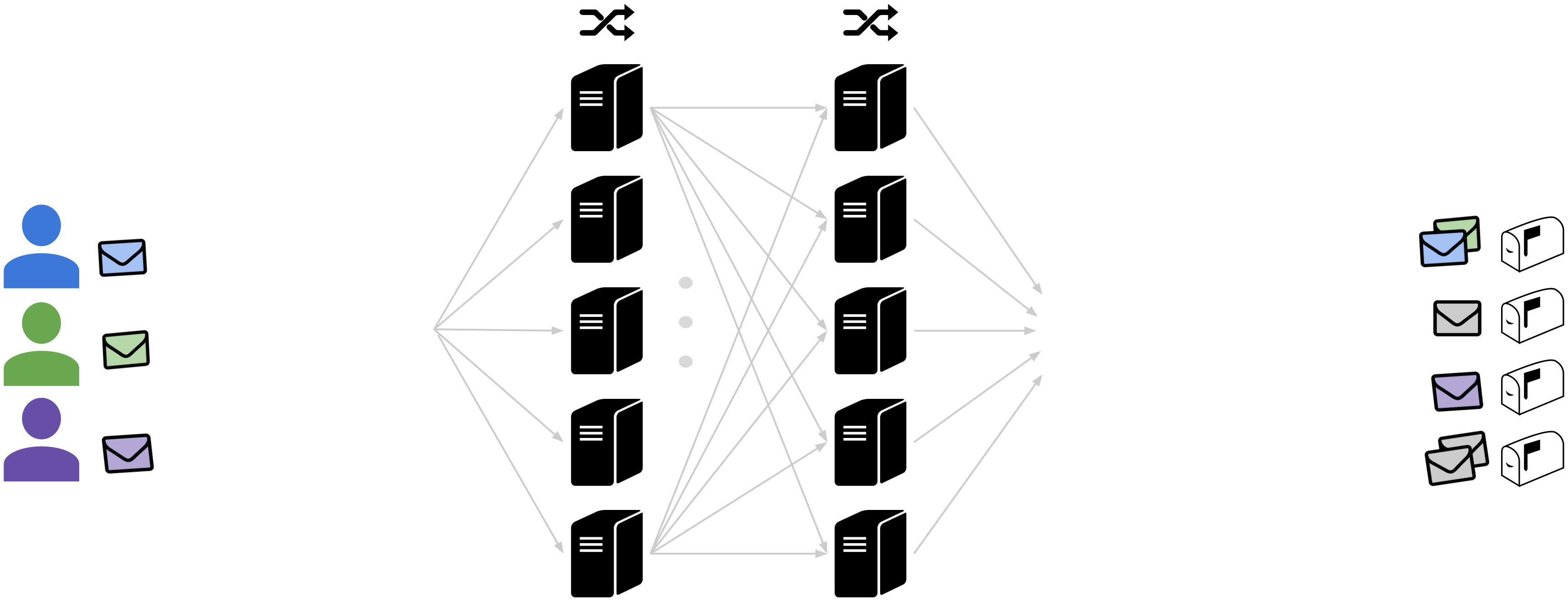
- Iterated butterfly topology [ICALP ‘14] as used by Atom [SOSP ‘17]
- Large depth not good for low latency applications



One butterfly iteration

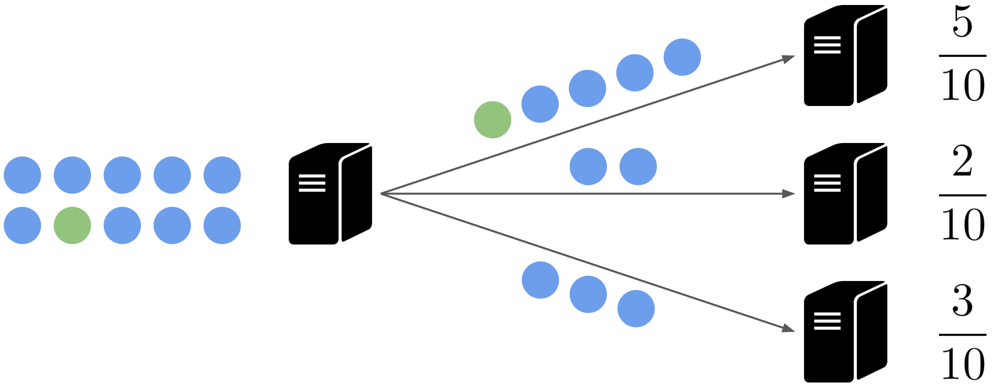
$n =$ # of servers

Stadium uses 2-layer mixnet with differential privacy analysis.



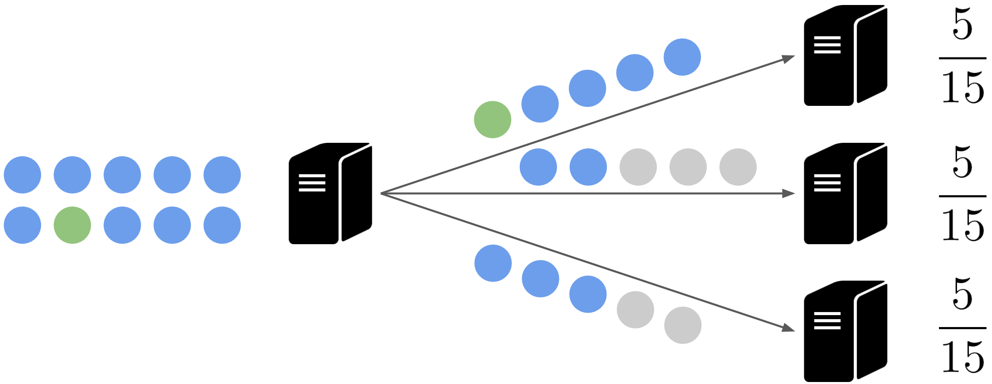
Traffic analysis attacks take advantage of uneven routings.

- Trace messages by modeling likely paths through mixnet (Borisov [PET ‘05])



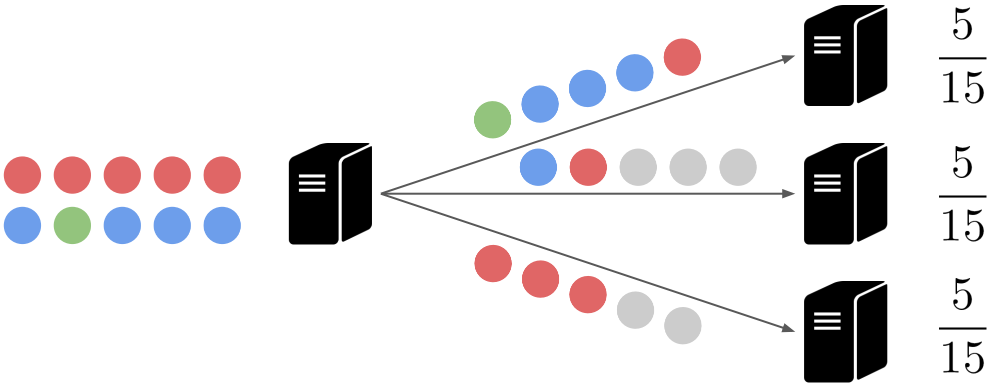
Traffic analysis attacks take advantage of uneven routings.

- Trace messages by modeling likely paths through mixnet (Borisov [PET '05])
- Even if links are padded with dummy messages, adversary can incorporate adversary-known inputs and outputs to infer uneven routing



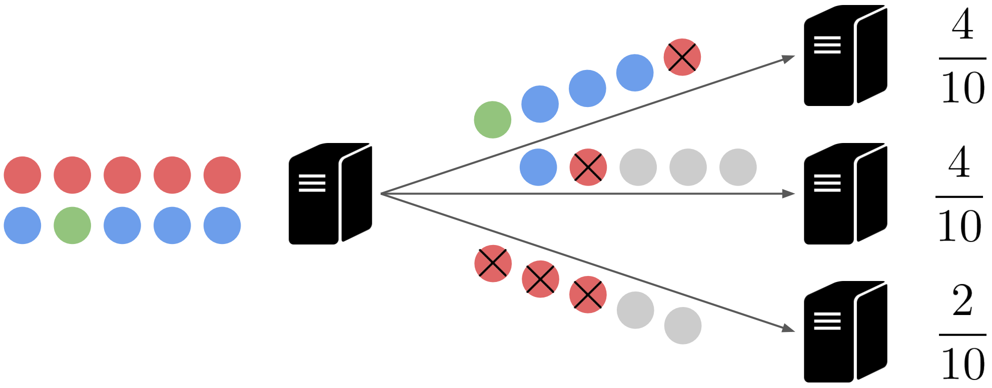
Traffic analysis attacks take advantage of uneven routings.

- Trace messages by modeling likely paths through mixnet (Borisov [PET '05])
- Even if links are padded with dummy messages, adversary can incorporate adversary-known inputs and outputs to infer uneven routing



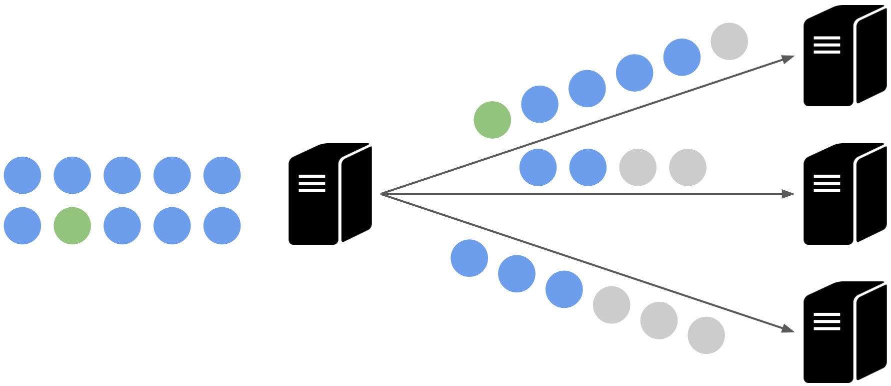
Traffic analysis attacks take advantage of uneven routings.

- Trace messages by modeling likely paths through mixnet (Borisov [PET ‘05])
- Even if links are padded with dummy messages, adversary can incorporate adversary-known inputs and outputs to infer uneven routing



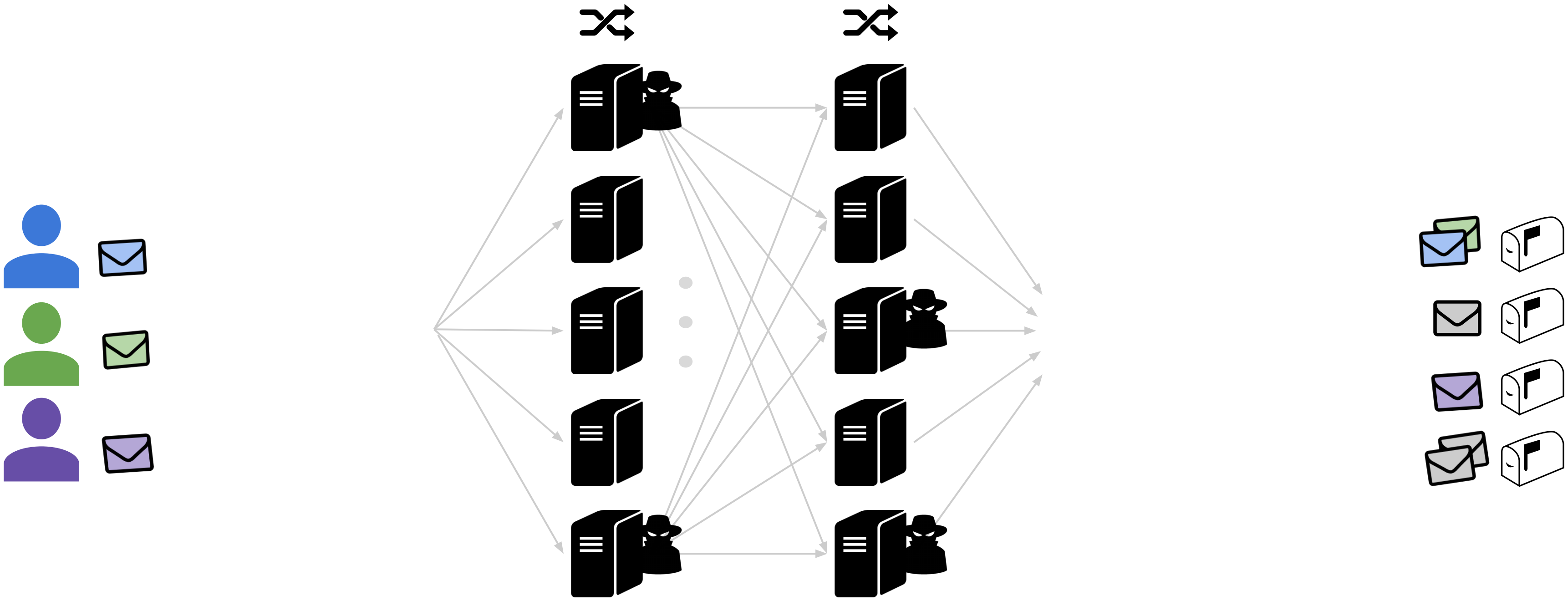
Add noise messages to provide differential privacy for uneven routings.

- Adversary manipulates padding through known message injection
- Unlike padding, noise messages are independent of adversary action



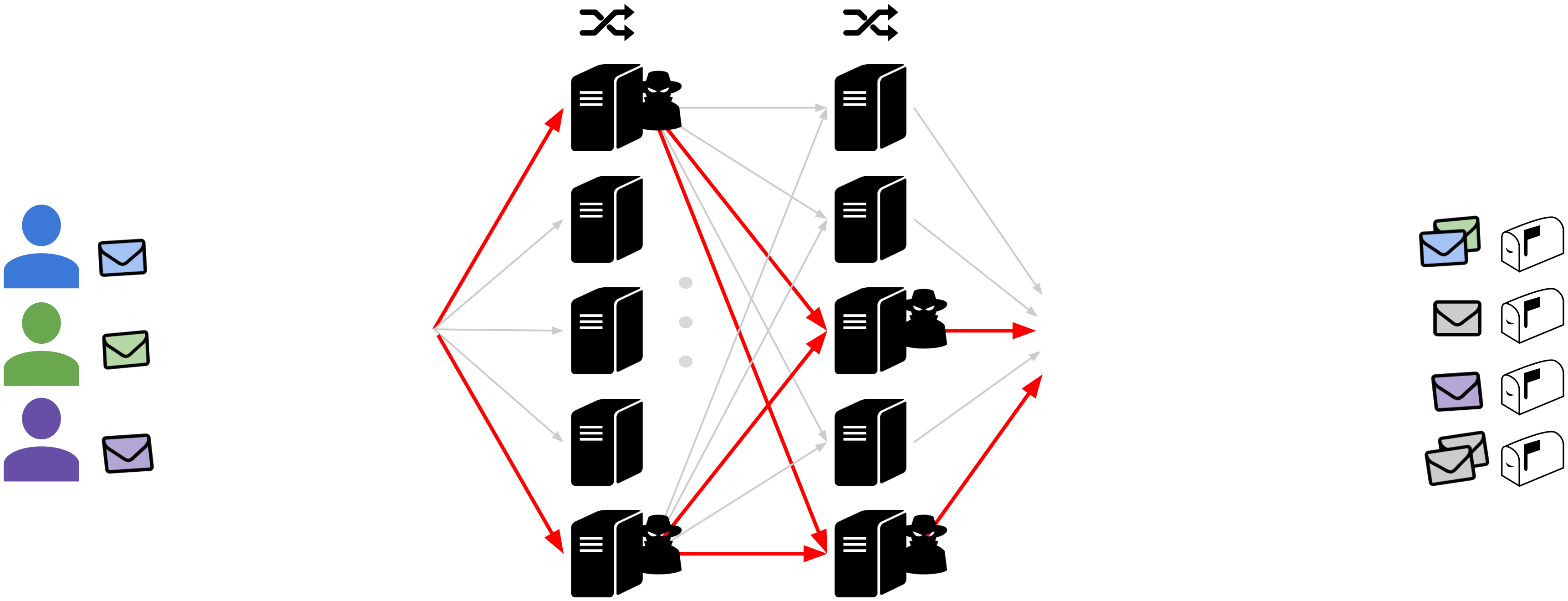
Noising internal links not helpful if messages aren't mixed.

- Adversary learns path of all messages through compromised servers



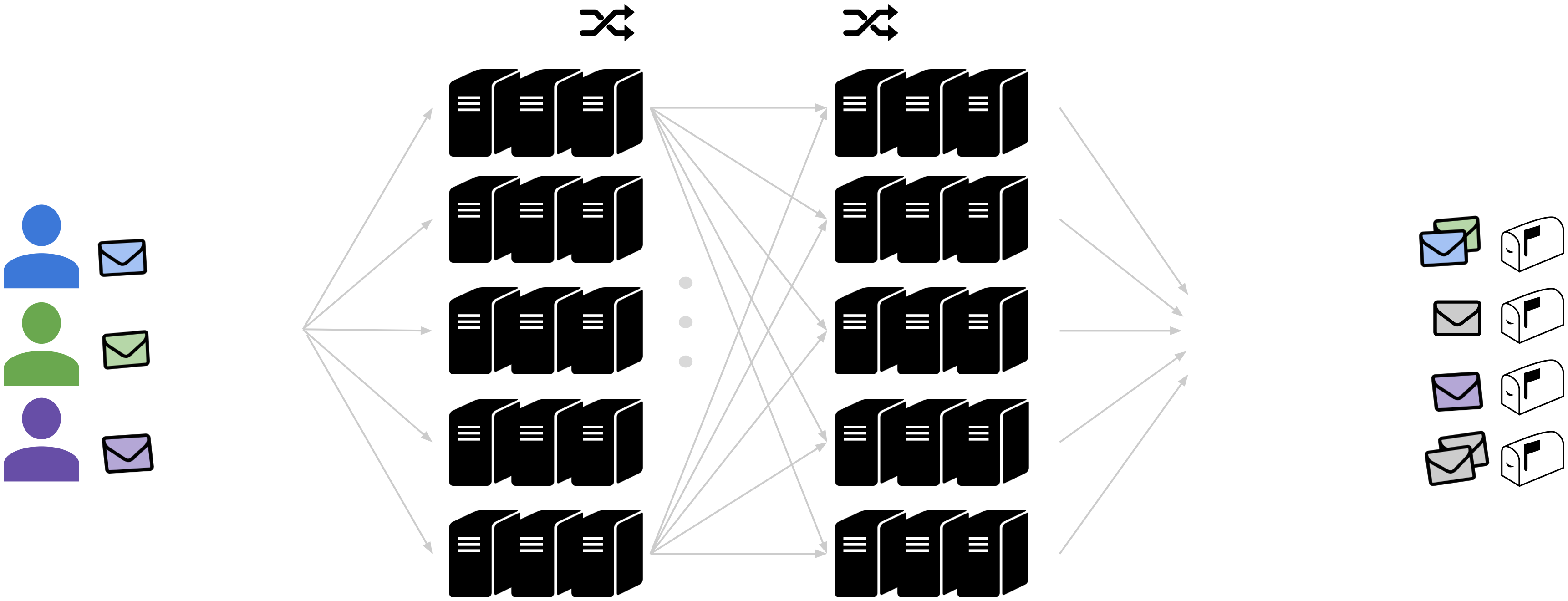
Noising internal links not helpful if messages aren’t mixed.

- Adversary learns path of all messages through compromised servers

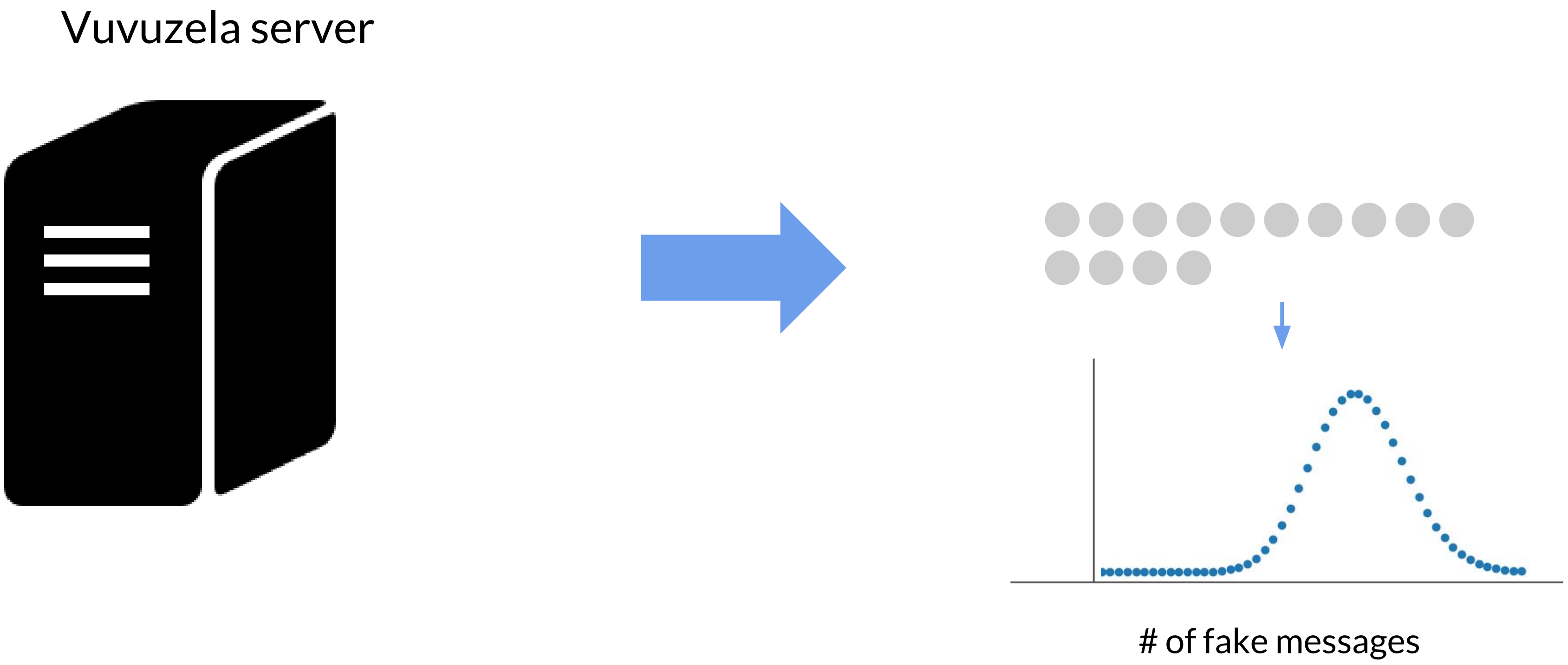


Ensure mixing by organizing providers into small groups of servers.

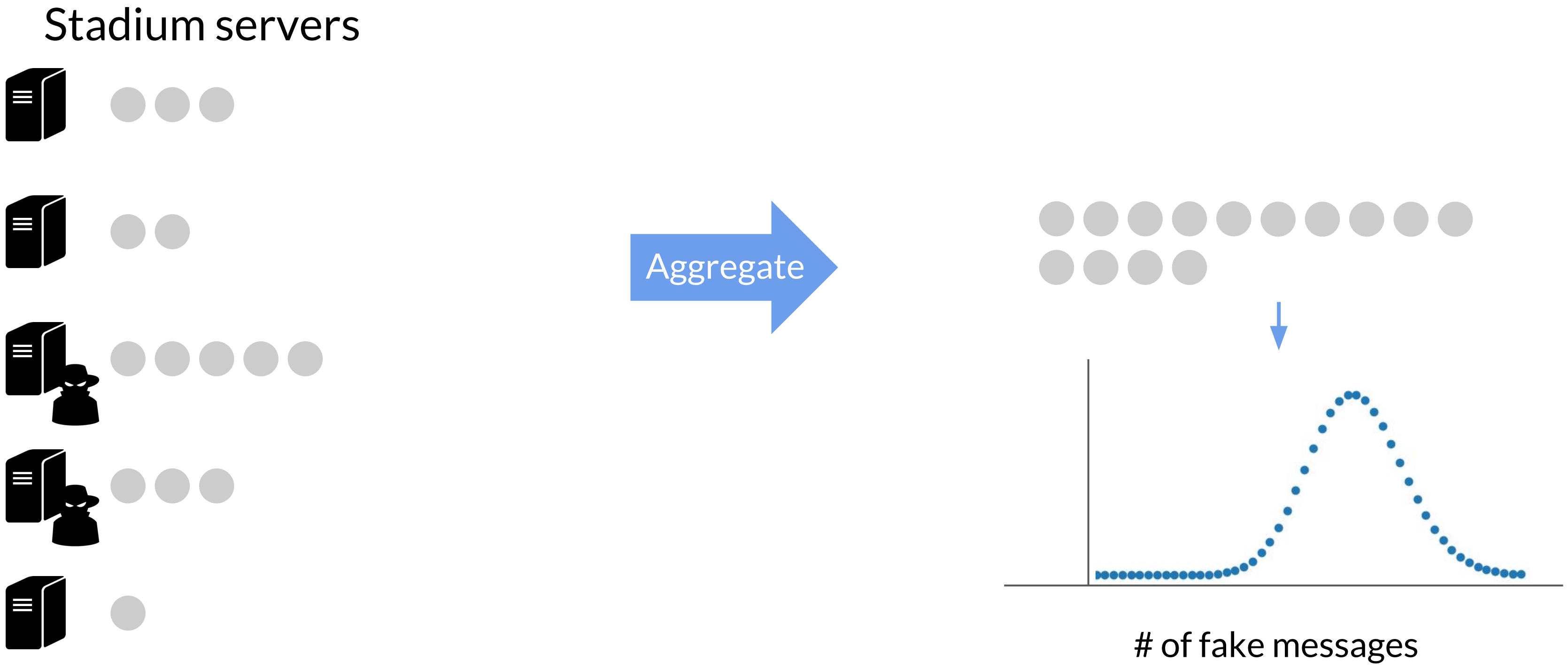
- Probability of compromise with random assignment falls exponentially with group size



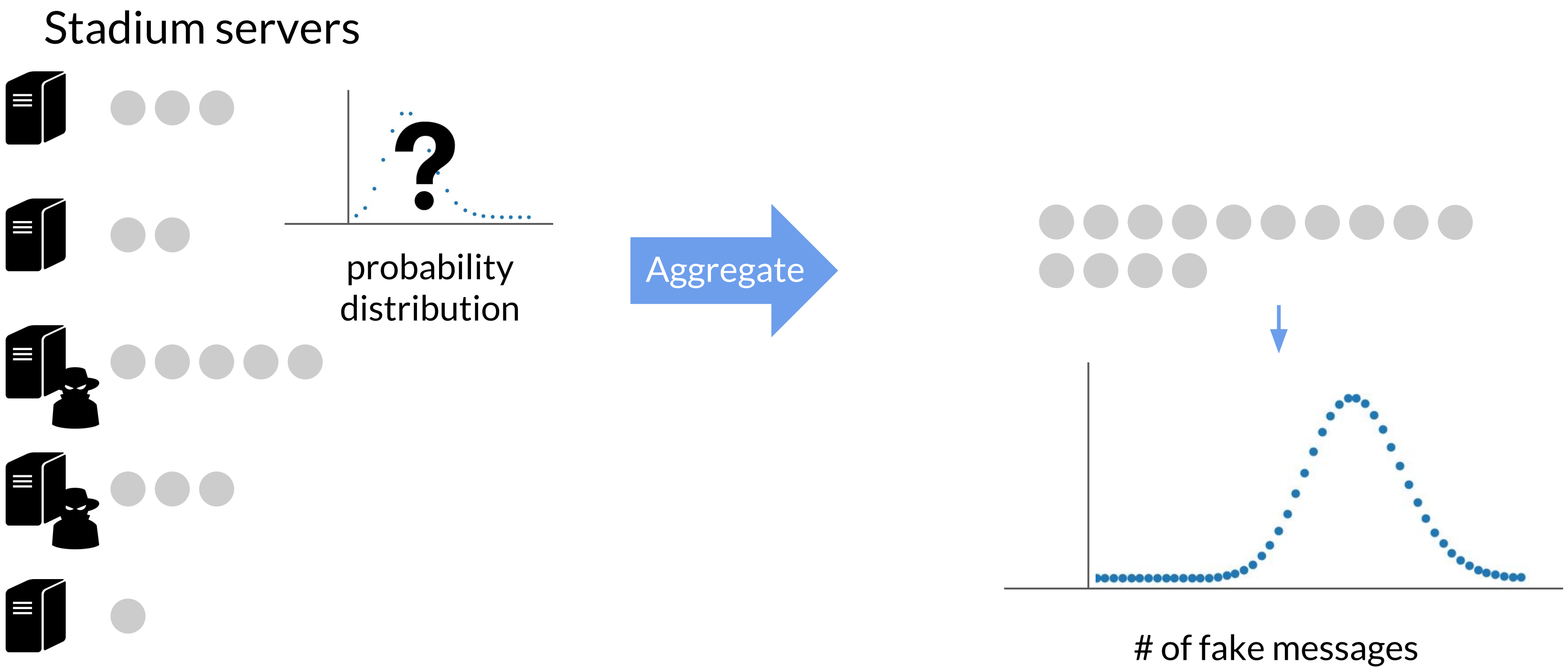
Problem: Scaling noise generation



Problem: Distributed noise generation



Problem: Distributed noise generation

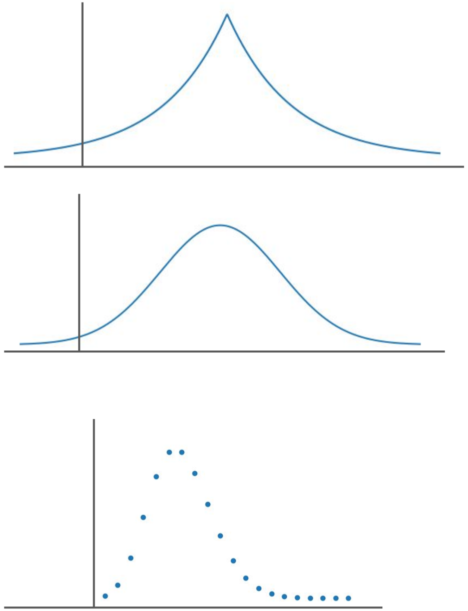


Poisson distribution for distributed noise generation

Noise mechanism

	Additive	Discrete	Non-negative
Laplace	✗	✗	✗
Gaussian	✓	✗	✗
Poisson	✓	✓	✓

- Poisson provides all properties nicely



Multidimensional analysis for reducing noise requirements

- When a user changes communication pattern, only a few links are affected
- Reduce noise by a factor of $\frac{1}{\sqrt{n}}$ where $\frac{1}{n}$ is probability link is affected

Verifiable processing pipeline

- Ensure noise messages stay in system
- Utilize various cryptographic zero knowledge proofs of integrity
- Hybrid verification scheme
- Zero knowledge proof of shuffle is bottleneck processing cost
  - Multicore Bayer-Groth verifiable shuffle on Curve25519
  - **~ 20X** performance speedup over state of the art
  - E.g. 100K ciphertext shuffle speedup from 128 seconds to ~7 seconds

Implementation

- Prototype
  - Control and networking logic in Go (2500 lines of code)
  - Verifiable processing protocols in C++ (9000 lines of code)
    - Highly optimized Bayer-Groth verifiable shuffle implementation
  - Available at github.com/nirvantyagi/stadium

Evaluation

- Recall goal: horizontal scalability with inexpensive servers
- What is the cost of operating a Stadium server?
- Does Stadium horizontally scale?

Evaluation methodology

- Deploy Stadium on up to 100 Amazon c4.8xlarge EC2 VMs
  - 36 virtual cores, 60 GB memory
  - US East region
  - Message size: 144 B

- Extrapolate scaling patterns to larger deployment sizes

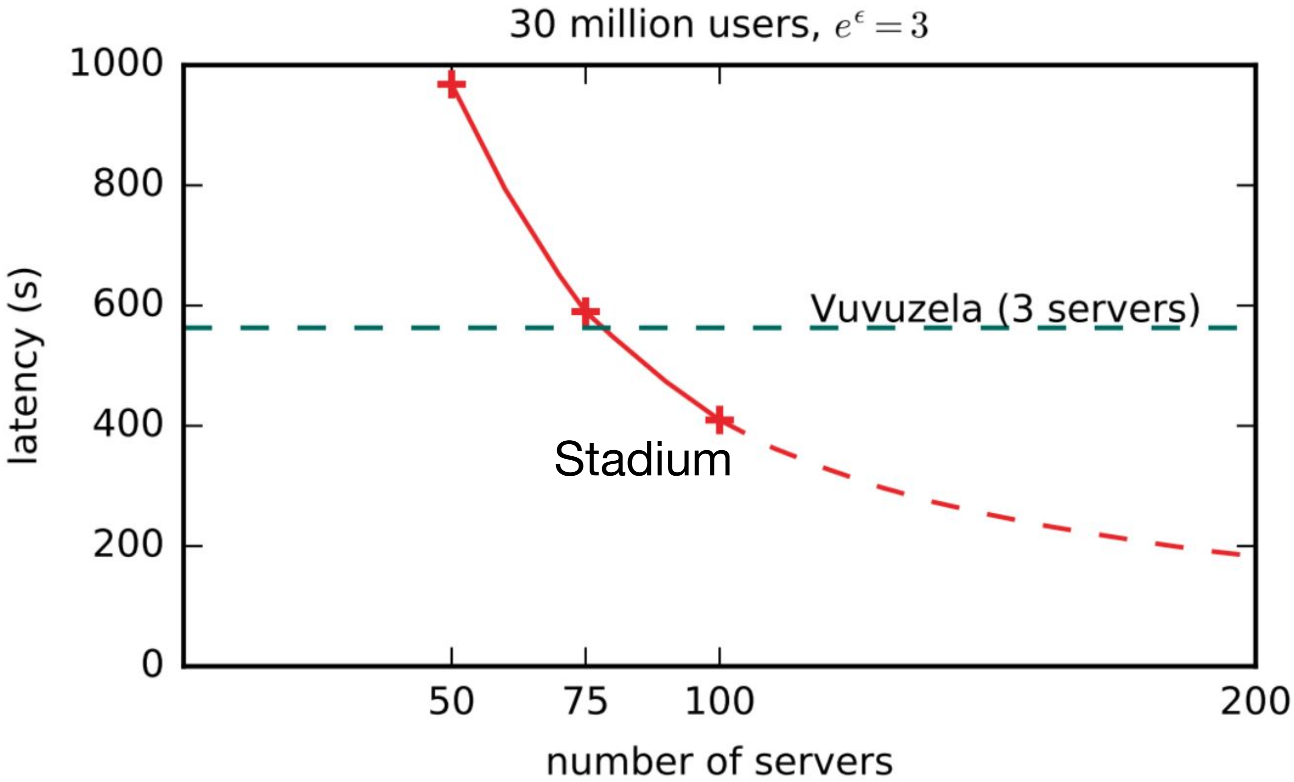
Operating costs of a Stadium server are relatively small

88 - 173 Mbps   6-13% of Vuvuzela’s 1.3Gbps

- Bandwidth is dominant cost
- Operating costs ~ $110 / month*
- Top 300 of relays in Tor offer > 140 Mbps

*W. Norton. 2010. Internet Transit Prices - Historical and Projected. Technical Report. http://drpeering.net/white-papers/Internet-Transit-Pricing-Historical-And-Projected.php

Messages are effectively distributed across servers to reduce latency



Conclusion

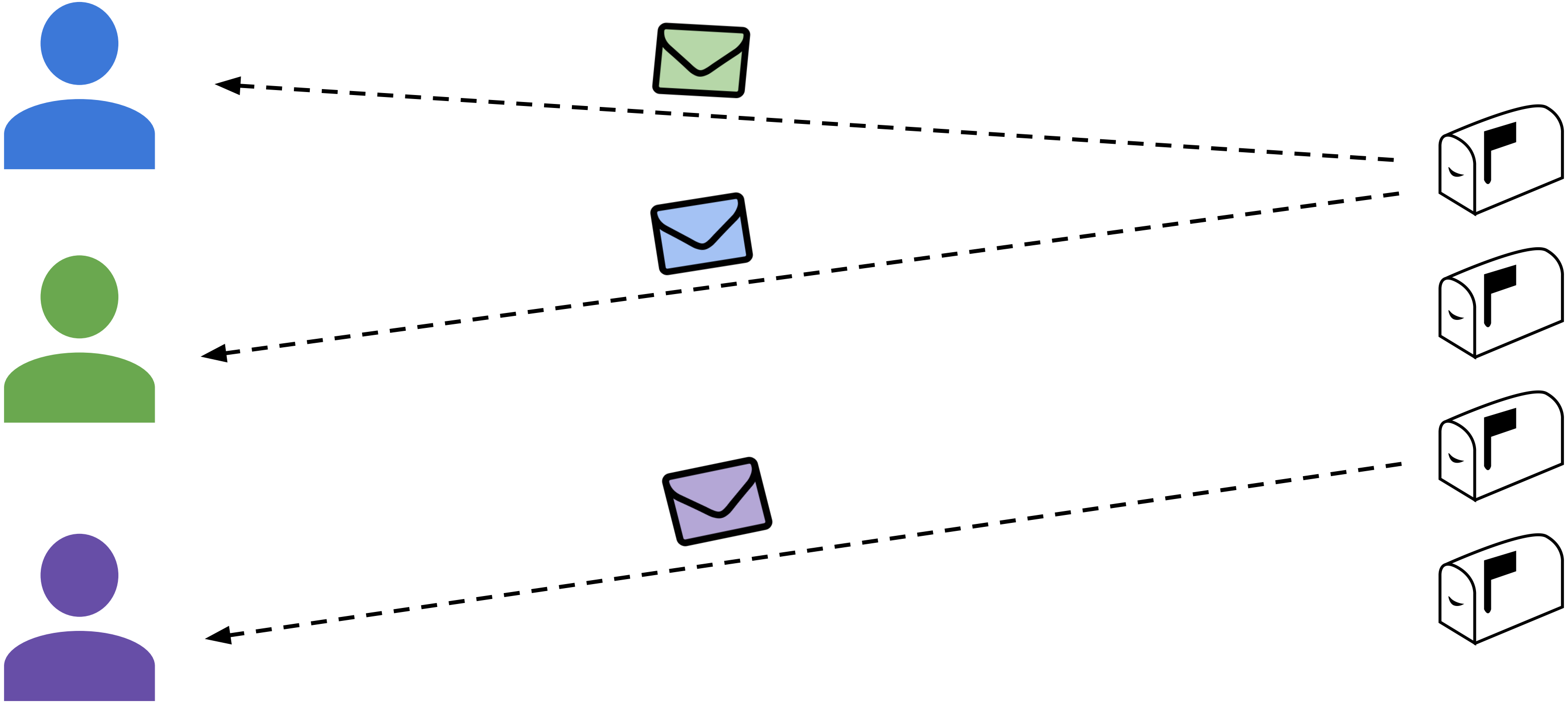
- Stadium: high-throughput, horizontally-scaling, metadata-private system
  - Verifiable parallel mixnet resistant to traffic analysis
  - Fast zero-knowledge proofs of shuffle
  - Collaborative noise generation with Poisson distribution
- Multidimensional differential privacy analysis
- Implementation and evaluation of prototype

Prototype at github.com/nirvantyagi/stadium

Reserve Slides

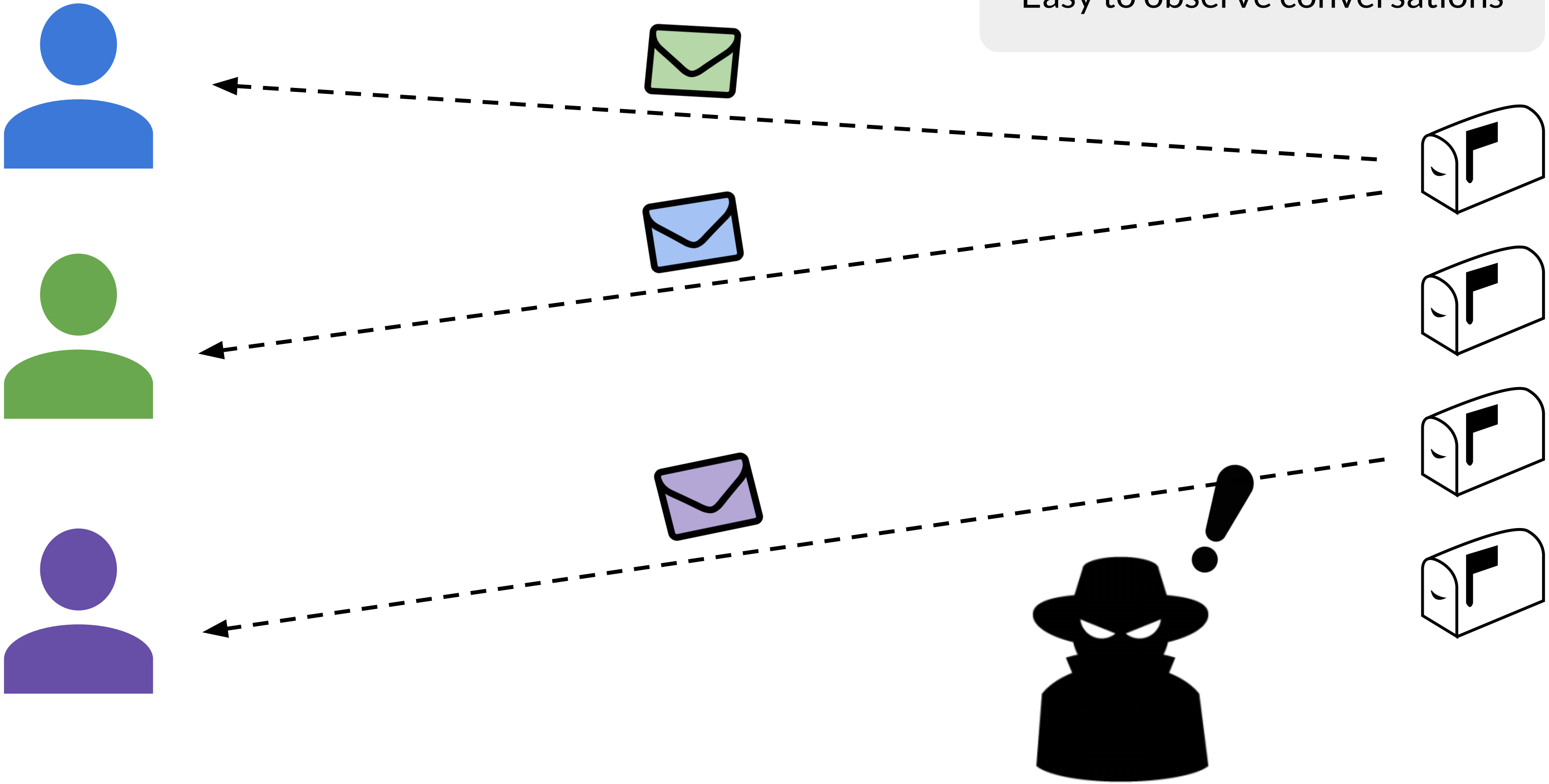
Dead-drop message exchange

Dead-drop message exchange

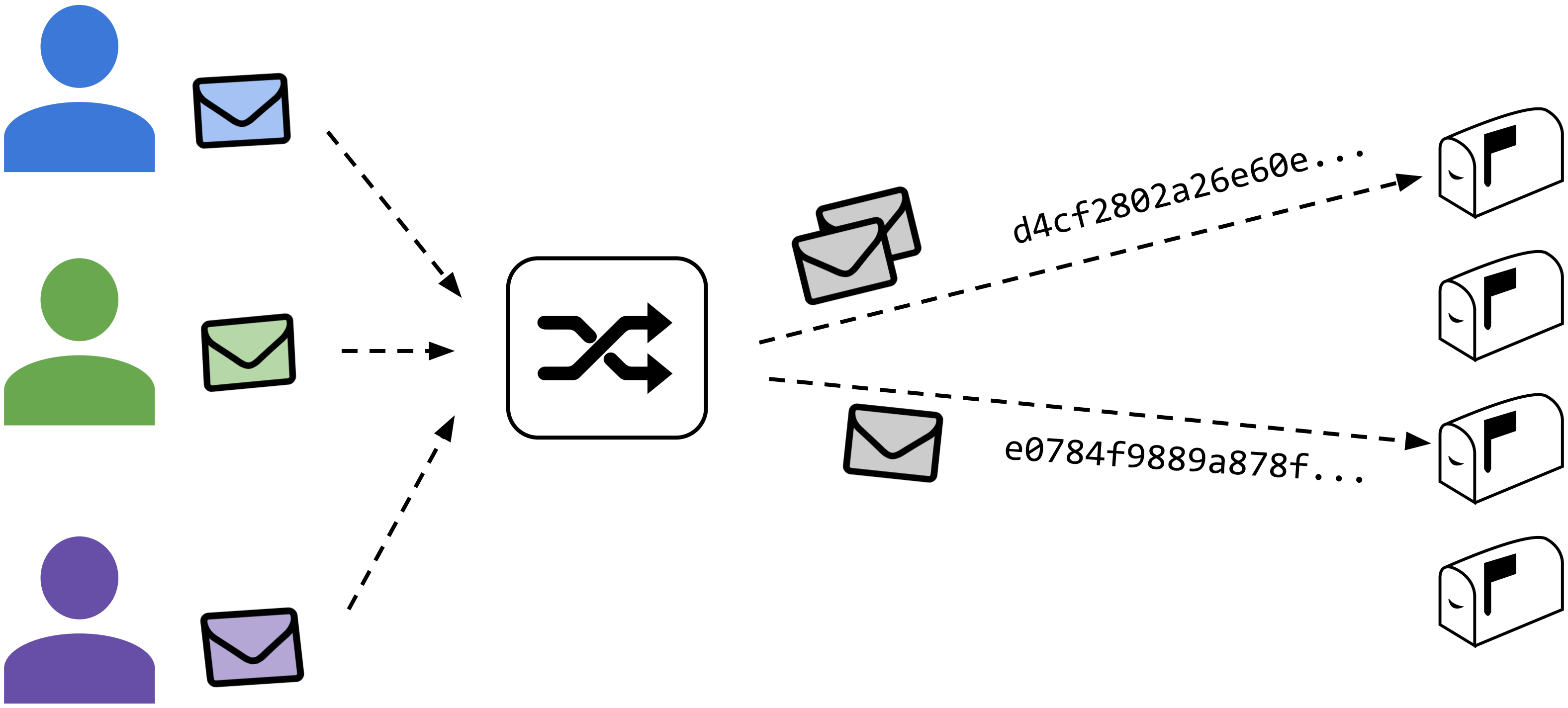


Dead-drop message exchange

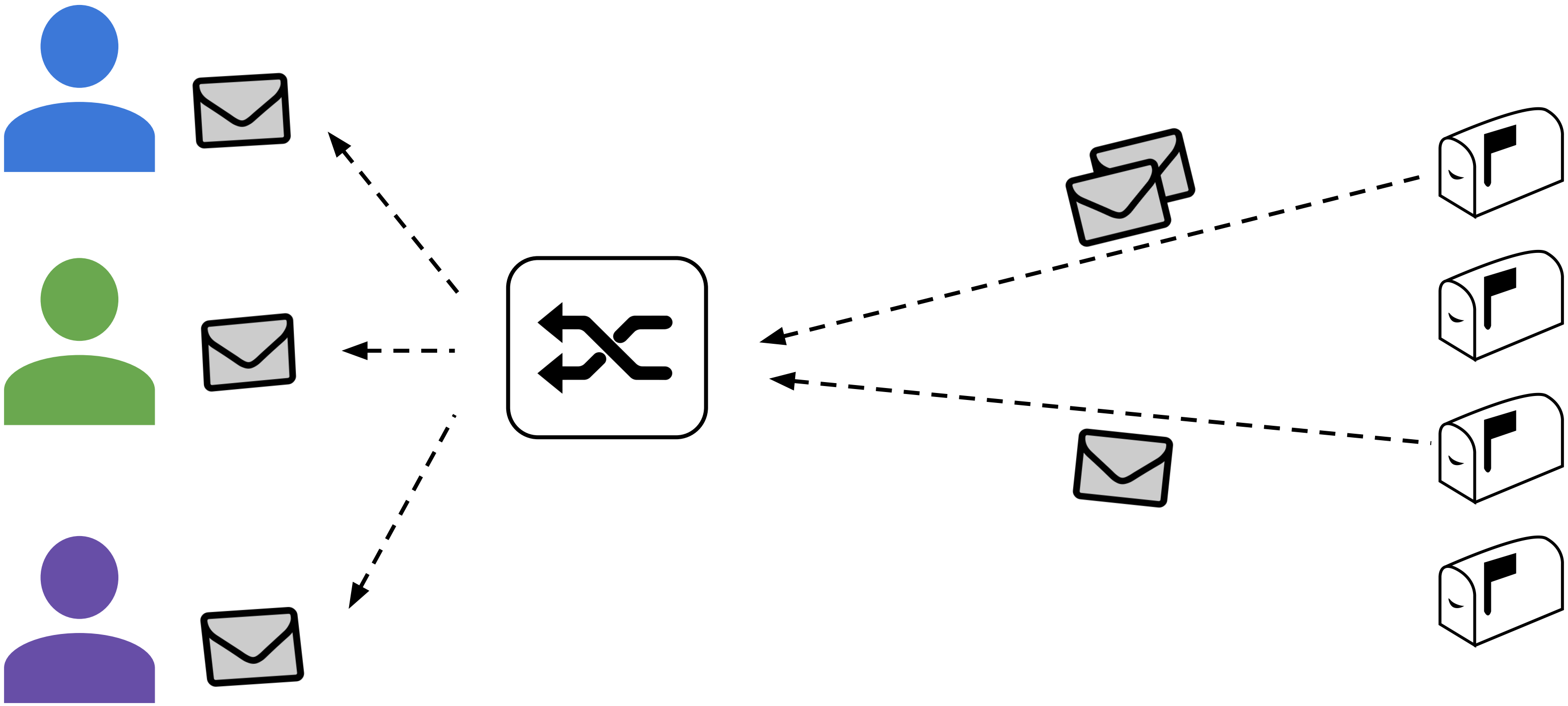
Easy to observe conversations



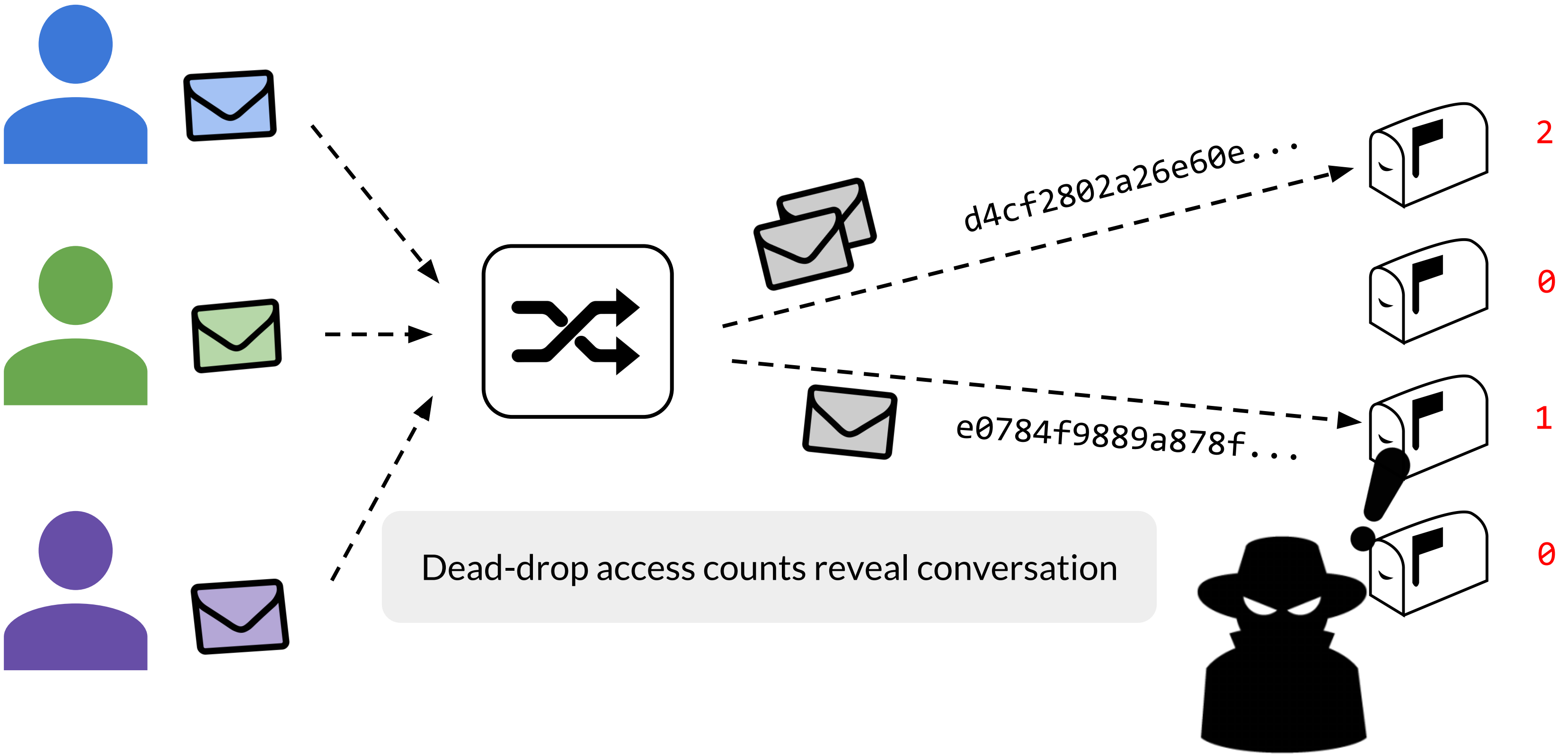
Dead-drop message exchange



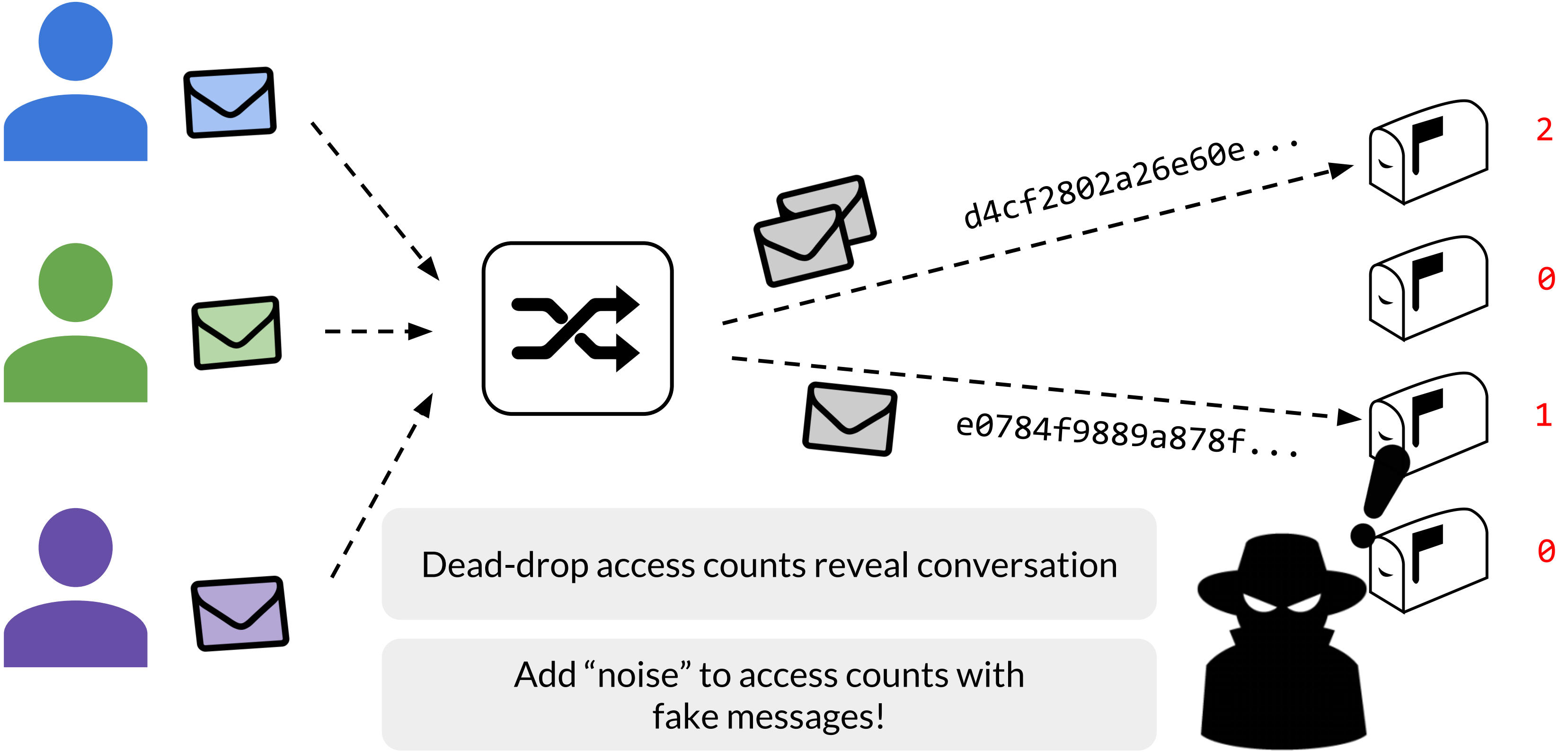
Dead-drop message exchange



Dead-drop message exchange



Dead-drop message exchange



Dead-drop access counts reveal conversation

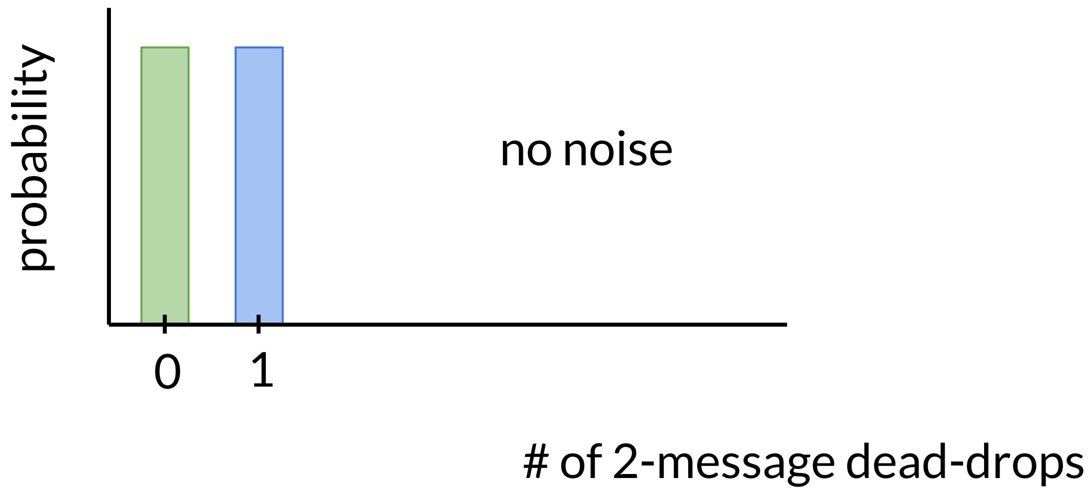
Add “noise” to access counts with fake messages!

Differential Privacy

$$\text{Pr[Alice talking to Bob]} \leq \epsilon \times \text{Pr[Alice not talking to Bob]} + \delta$$

Differential Privacy

$$\text{Pr[Alice talking to Bob]} \leq \epsilon \times \text{Pr[Alice not talking to Bob]} + \delta$$



Differential Privacy

$$\Pr[\text{Alice talking to Bob}] \leq \epsilon \times \Pr[\text{Alice not talking to Bob}] + \delta$$