Periscoping: Private Key Distribution for Mixnets

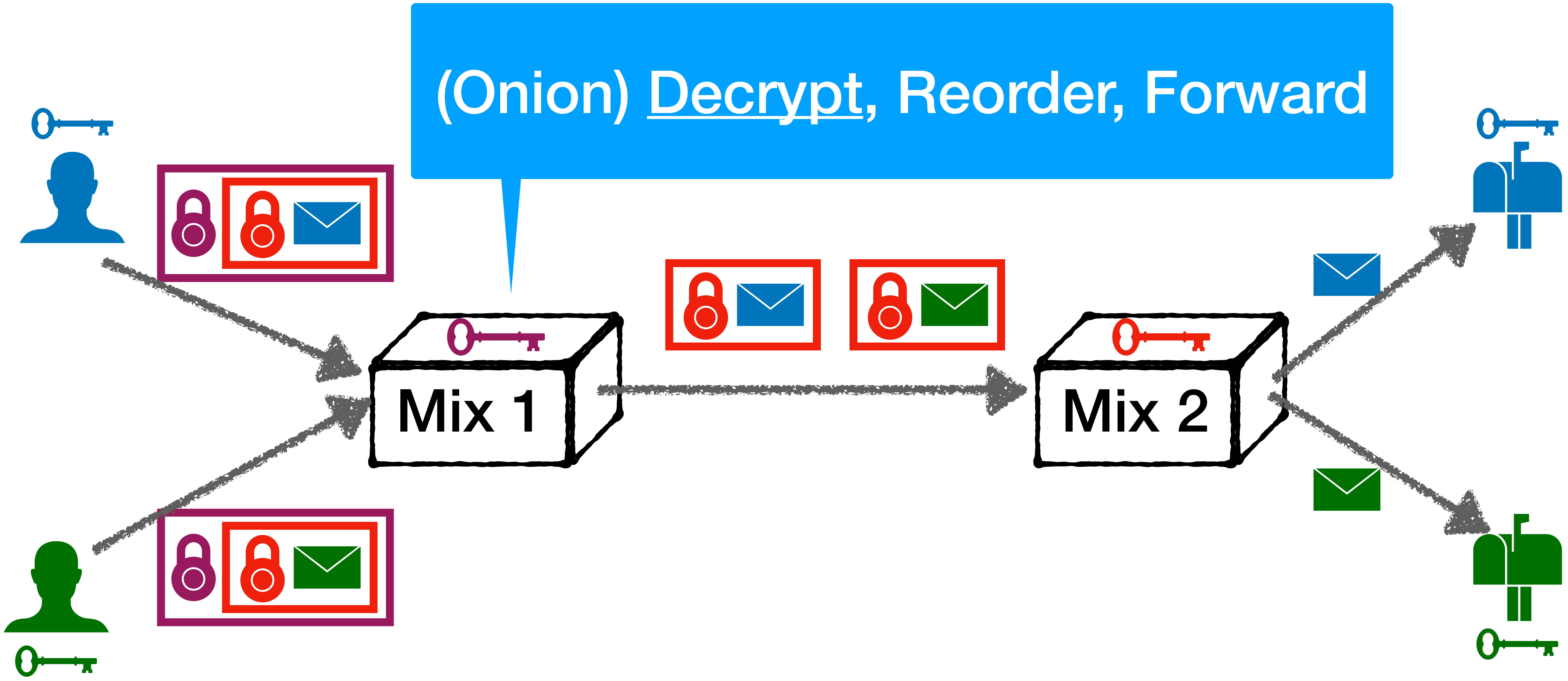
Shuhao Liu
Shenzhen Institute of Computing Sciences
May 15, 2024

A joint work with Li Chen (University of Louisiana at Lafayette) and Yuanzhong Fu

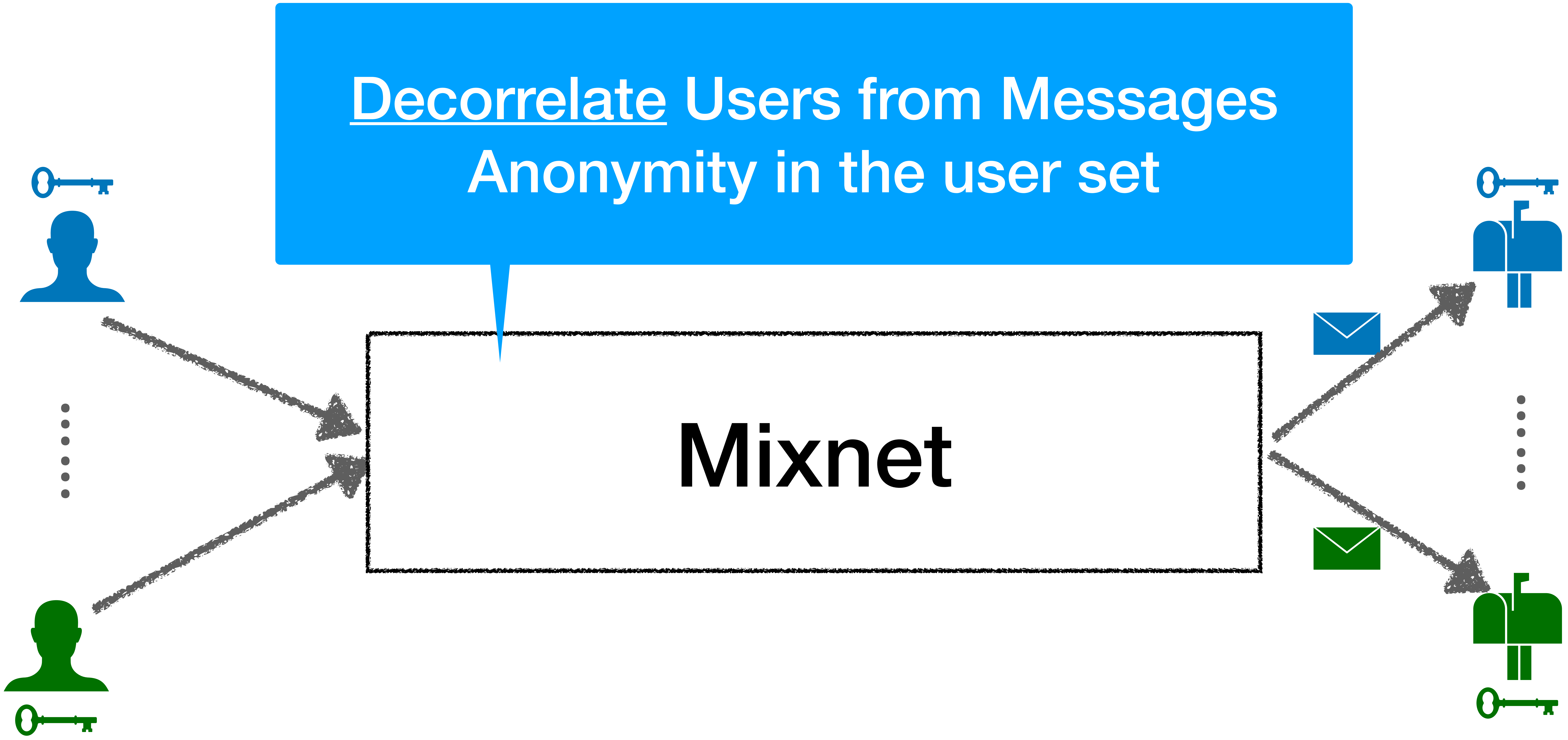
Background: Private Messaging

- Confidentiality
  - Data: the content
  - Solution: end-to-end encryption
- Metadata privacy
  - Data: when, with whom, how many…
  - Solution: Mixnets

Background: Mixnets

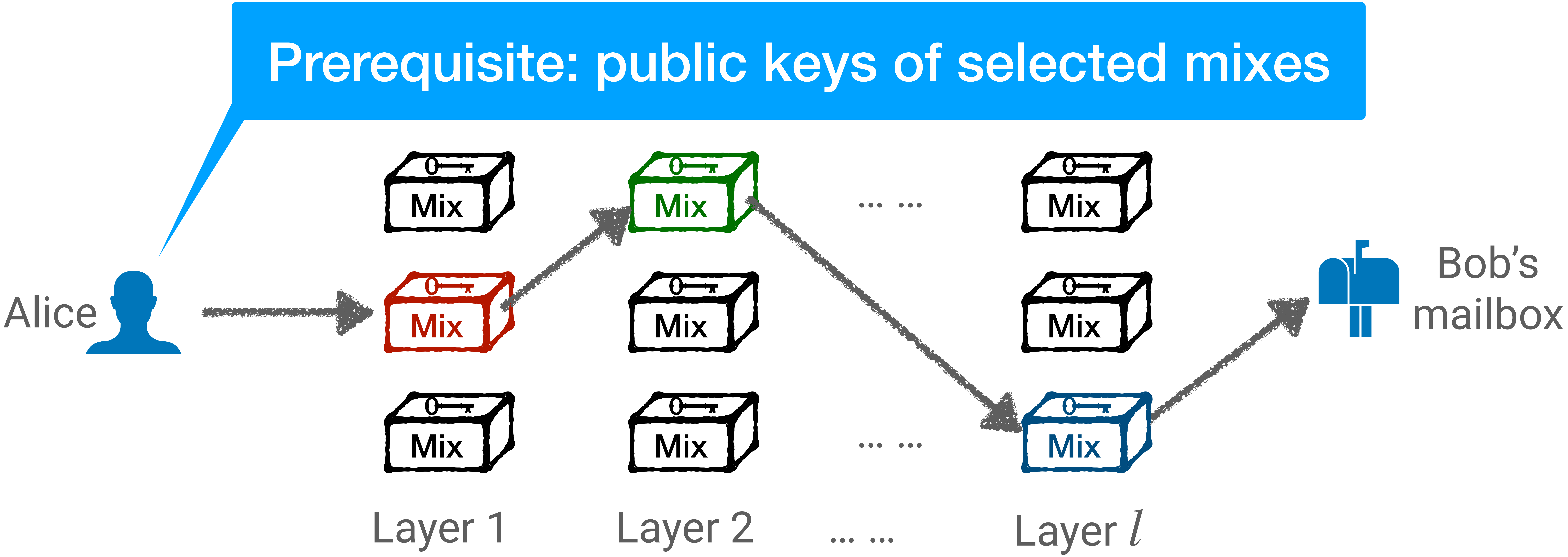


Background: Mixnets



Background: Free-Route Mixnet

- Scalability: a <u>random</u> length-l path in n available mixes <u>each round</u>



Requirements for Alice

- Maintain ALL n public keys UP-to-DATE
- Select mixes independently randomly, by herself

Challenge at Scale

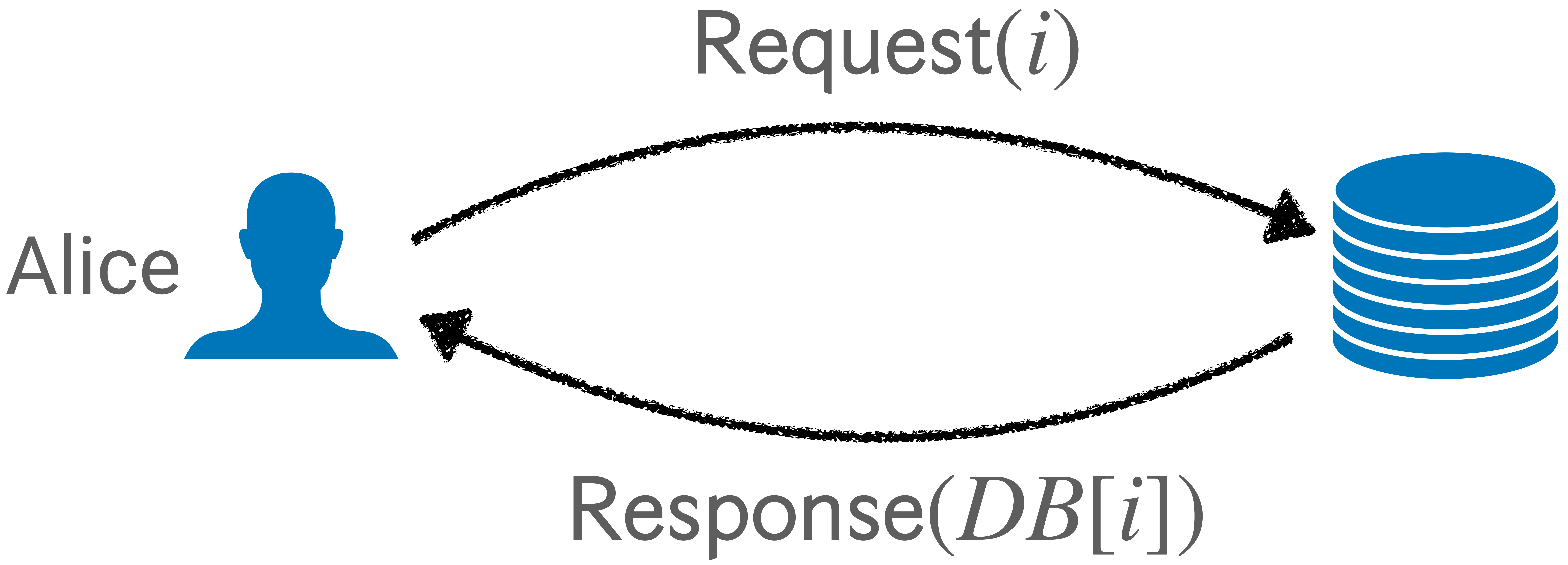
- Distribute Mixnet information directory DB to all users
  - Each mix has a copy of DB
  - Entry: ID, public key, URL, expiration, certificate
- n mixes can support $O(n)$ users: $O(n^2)$ aggregate traffic
  - If $n > 100{,}000$, sync traffic $\geq 10\,\%$ total bandwidth usage

Design Objectives

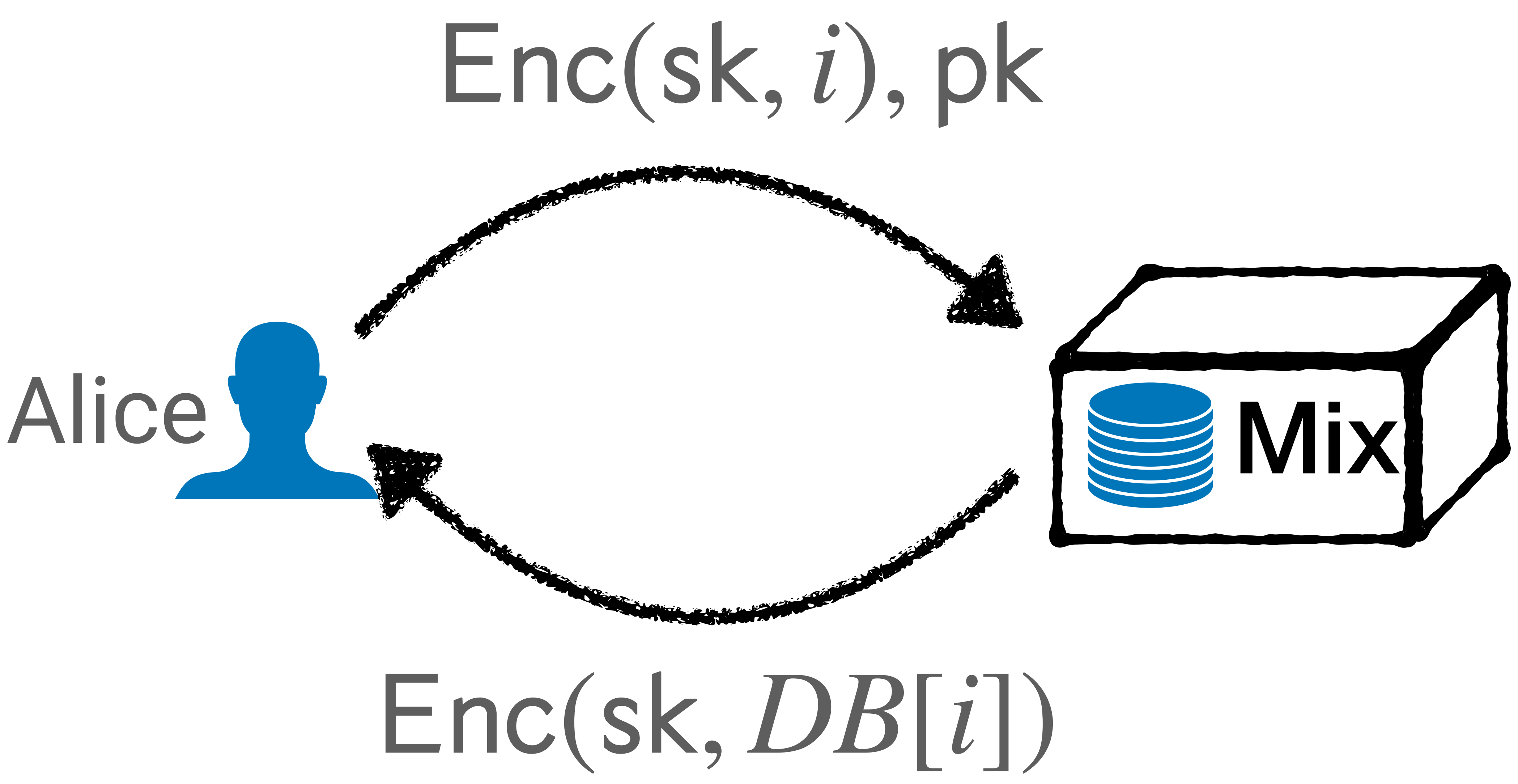
- Scalability: a sublinear download size for each user
  - Users cannot maintain the entire DB
- Threat model
  - Honest mixes are curious
  - At least 1 honest mix out of l (same as all free-route mixnets)
- Security: computational indistinguishability of DB entries

Private Information Retrieval

- Request $DB[i]$ without revealing the value of i

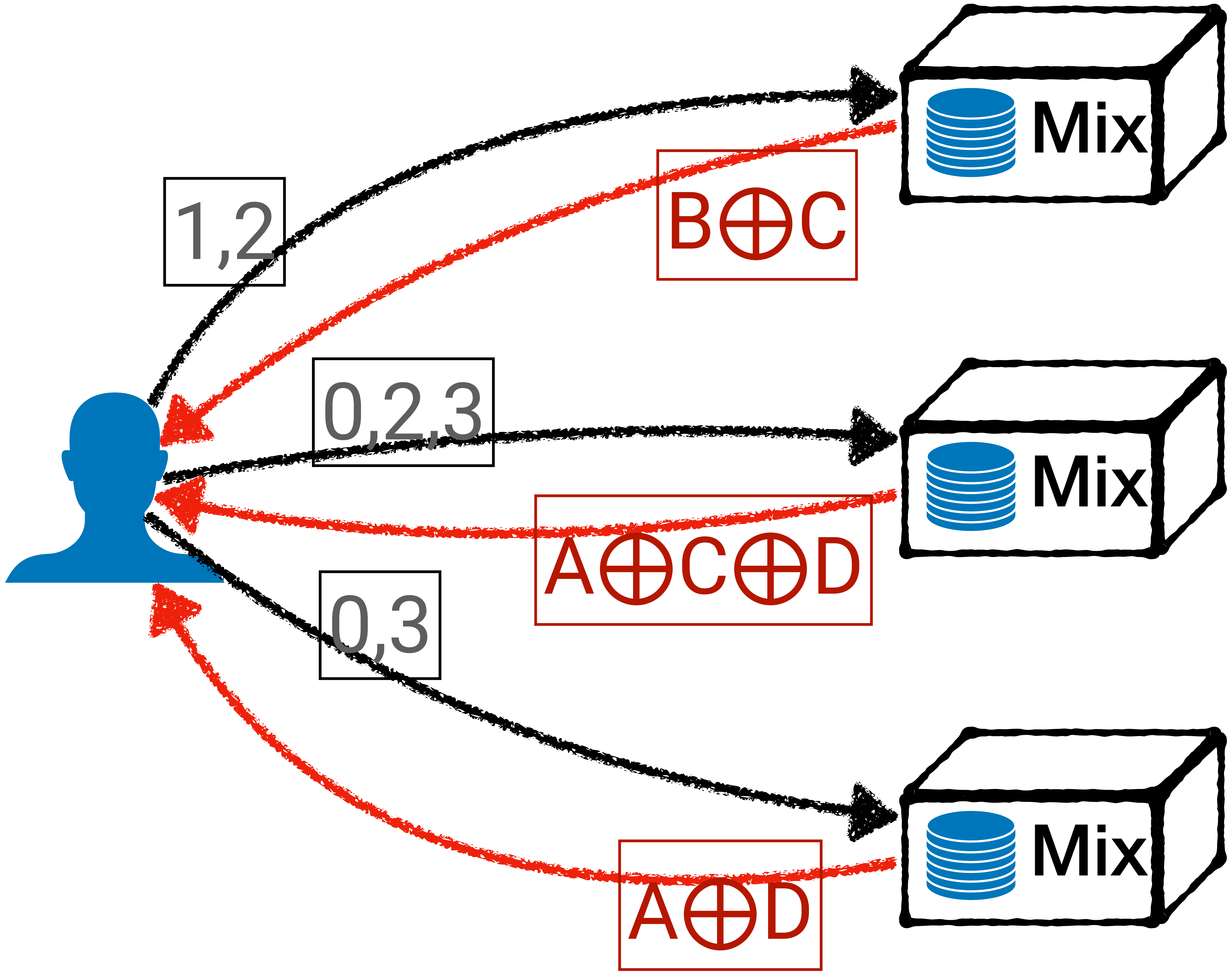


Prior Work: PIRTor (OnionPIR)



- Homomorphic encryption-based
  - 25s for $n = 2^{16}$,
    400s for $n = 2^{20}$
  - Request: 64KB
  - Response: $\geq 4.7 \times$ entry size

Basic Solution: Multi-Server PIR



 = [A, **B**, C, D]

- Proven secure if probability=0.5 in 2 request
- ✅ $O(l)$ download traffic
- ❌ $O(nl)$ upload traffic

How to **compress** a random set?

Random Set Compression

- Review: PseudoRandom Functions (PRFs)
  - A master key can generate a multiset S of pseudorandom numbers
    - $S = \{y = F(\mathsf{mk}, x) \,|\, x \in D\}$, e.g., $D = \{0,1,2,3\}$

🌟 mk is a compressed representation for S.

Random Set Compression

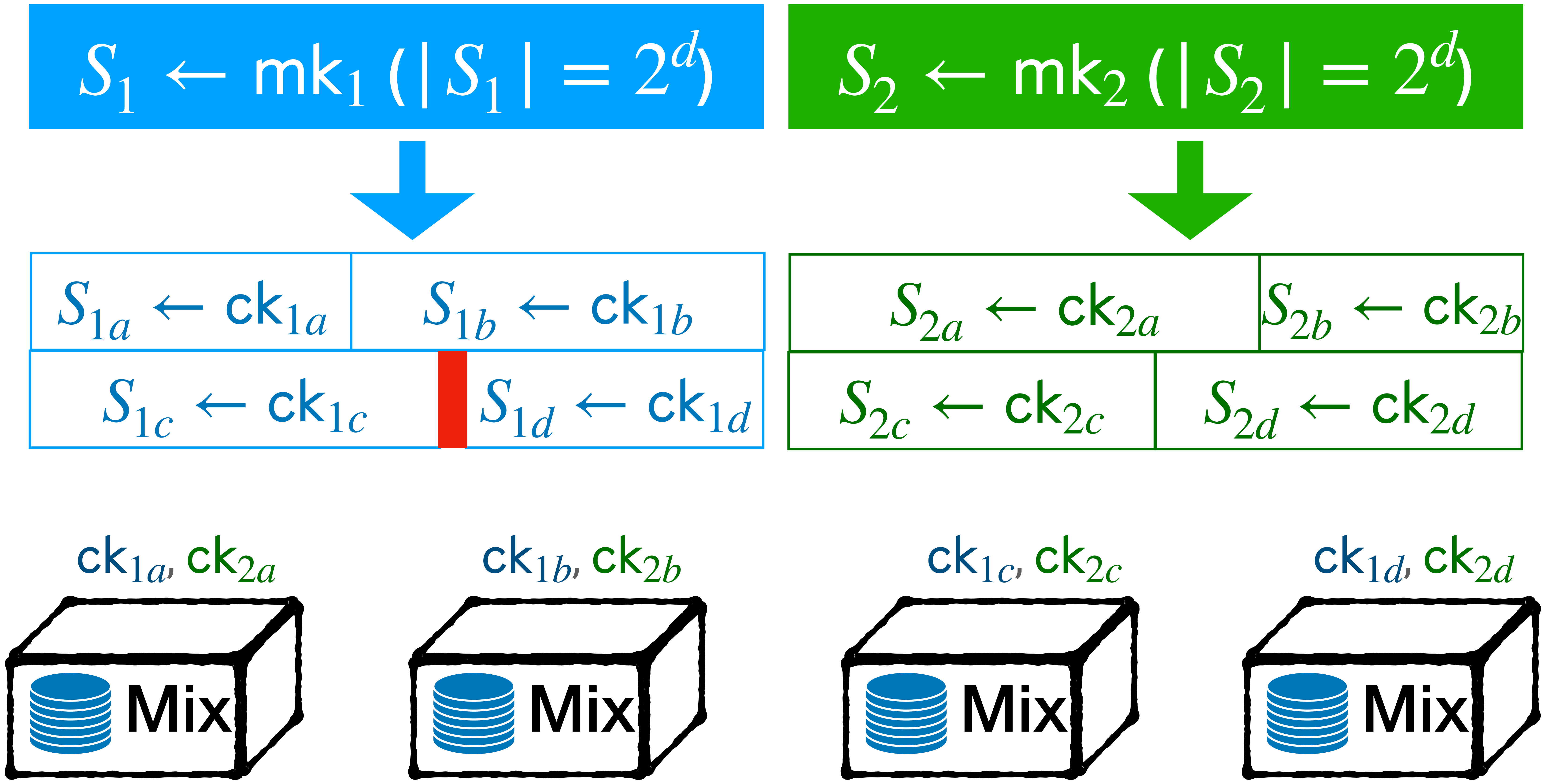
- **Constrained PseudoRandom Function (cPRF)**
  - A master key can generate a multiset S of pseudorandom numbers
    - $S = \{y = F(\mathsf{mk}, x) \mid x \in D\}$, e.g., $D = \{0,1,2,3\}$
  - A constrained key can generate a subset S' of S

🌟 mk is a compressed representation of S.
🌟 ck is a compressed representation of (any) $S' \subseteq S$.

Request Construction

- ✅ $O(l)$ download traffic
- ✅ $O(l \log n)$ upload traffic

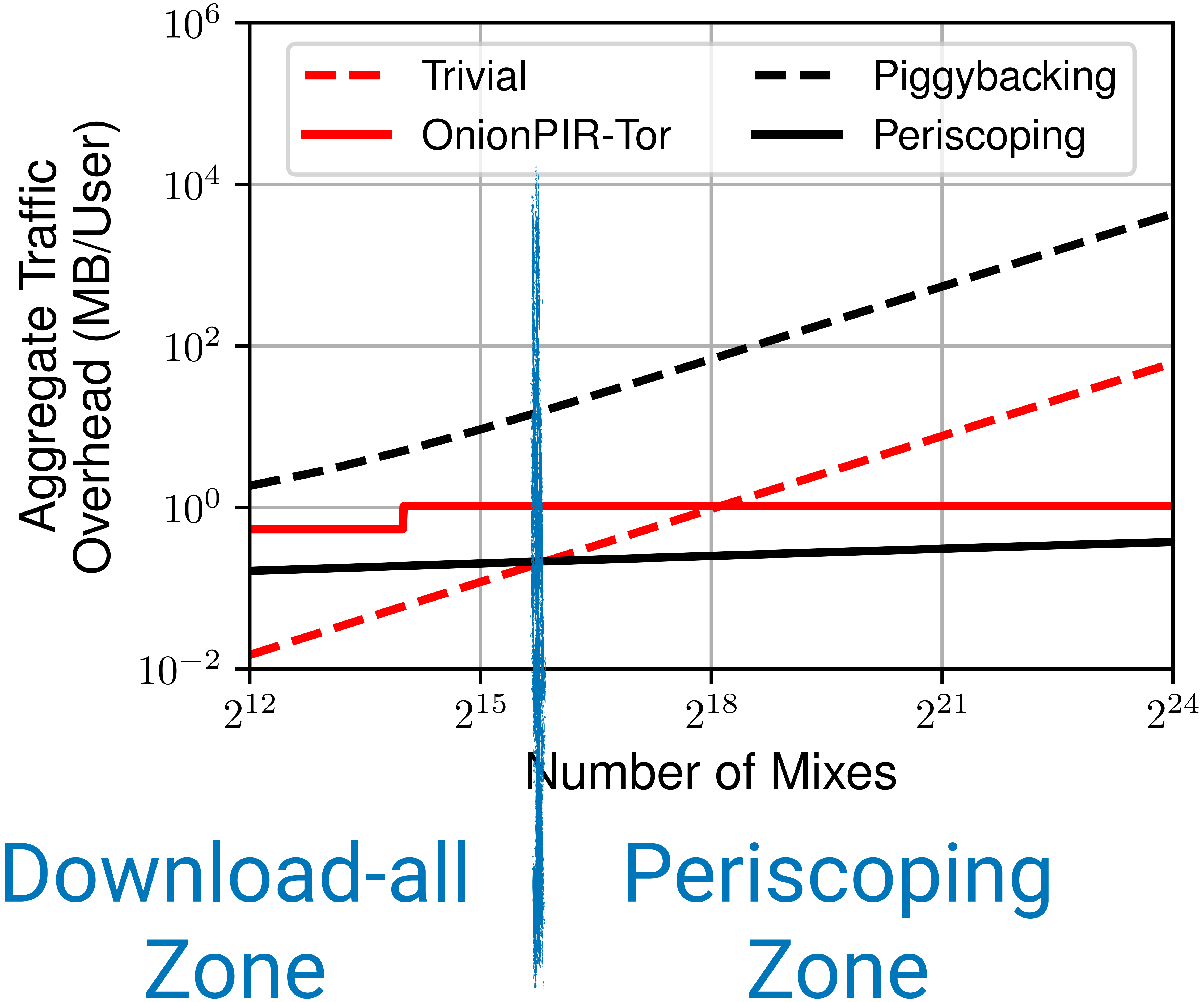
- DB has $n = 2^{2d}$ entries. Request 4 instances.



Request Construction

- Further optimizations: key reuse
  - ✅ $O(l)$ download traffic overhead
  - ✅ $O(l \log n) \rightarrow O(\log n)$ upload traffic overhead

Experiments: Communication Cost



Download-all Zone

Periscoping Zone

Experiments: Computational Cost

Stage	# Mixes	OnionPIR-Tor (s)	Piggybacking (ms)	Periscoping (ms)
Request building	2^{16}	8.8	6.7	15.2
	2^{20}	11.1	8.3	40.8
Response Generation	2^{16}	60.1	1.4	3.6
	2^{20}	973.4	22.5	59.9
Response Recovery	2^{16}	20.3	0.6	0.4
	2^{20}	21.8	0.6	0.3

Take-Aways

- **Problem:** Low-cost key distribution for large-scale mixnets
- **Solution:** A novel multi-server PIR scheme based on constrained Pseudorandom Functions
- **What is interesting:** Constrained Pseudorandom Functions to compress sets of random numbers
- **Application:** privacy-preserving data analytics

Thanks! Q &A