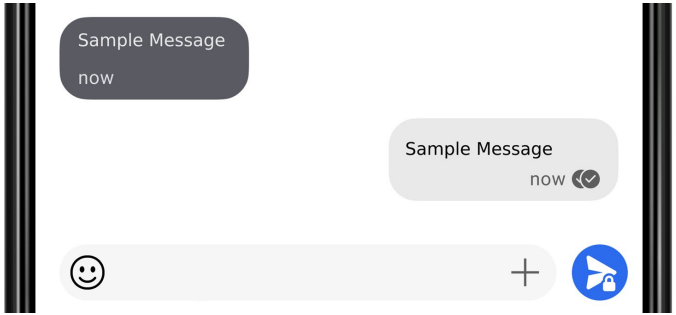
Truly Anonymous Sealed Sender in Signal

By Eric Chen and Boyan Litchev
Mentored by Kyle Hogan and Simon Langowski

What is Signal?

- Privacy-conscious messaging app
  - End-to-end encrypted
- 40 million monthly active users



Motivation

Confidentiality vs Anonymity

- Confidentiality → people don’t know the *contents* of a conversation
  - Message is encrypted

- Anonymity → don’t know the participants of a conversation
  - (Or the social graph of a network)

A Case for Anonymity

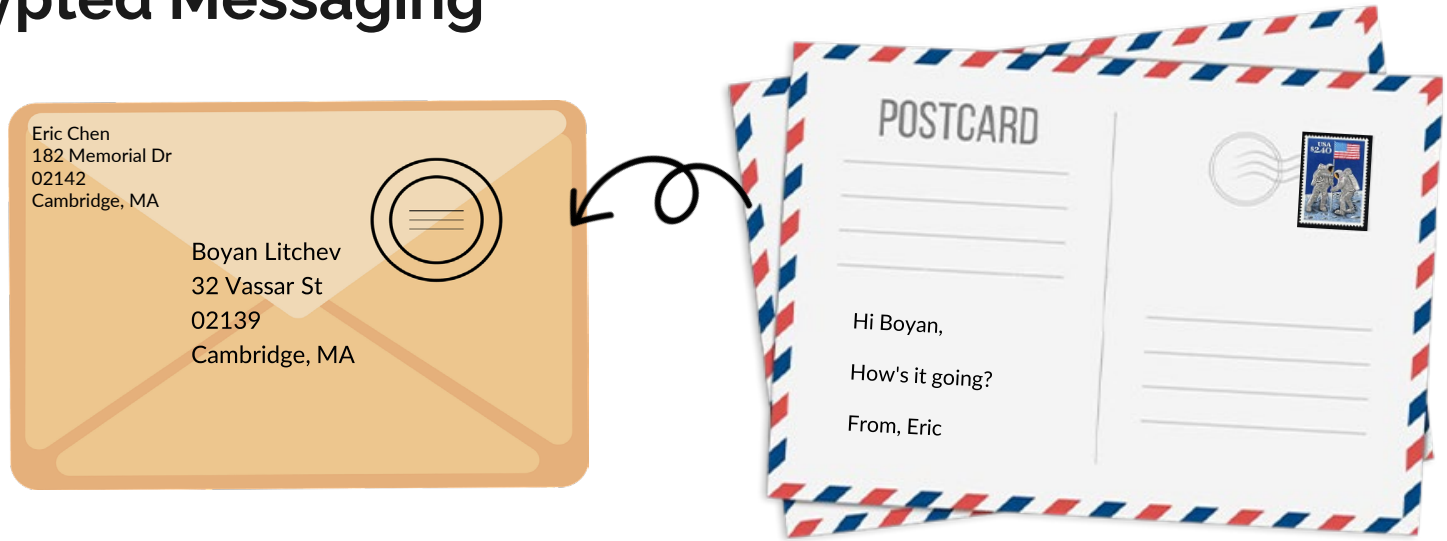
- Subpoenas
- Protest organization
- Whistleblowers
- Accuracy for research and surveys

Signal & Anonymous Communication

Normal (SMS/MMS) Messaging

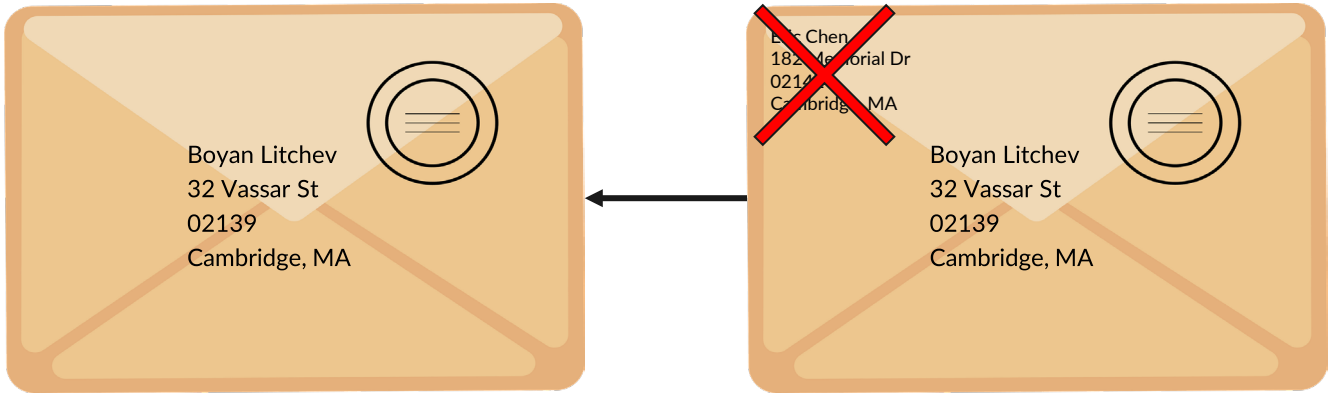
- Post Office knows message contents
- Post Office knows who Boyan and Eric are

Encrypted Messaging



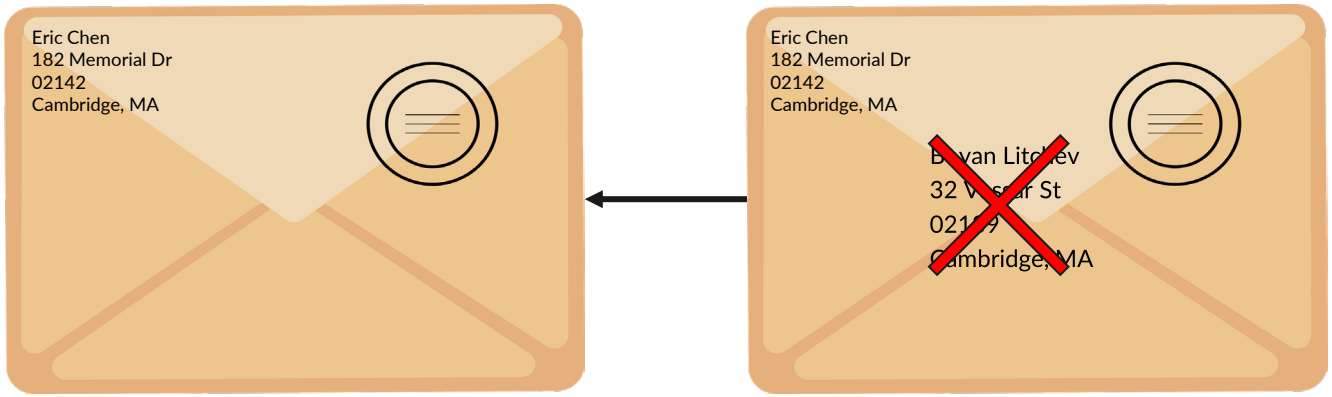
- Post Office knows message contents
- Post Office knows who Boyan and Eric are

Sealed Sender Messaging



- Post Office doesn't know message
- Post Office doesn't know who sent the message

Sealed Recipient



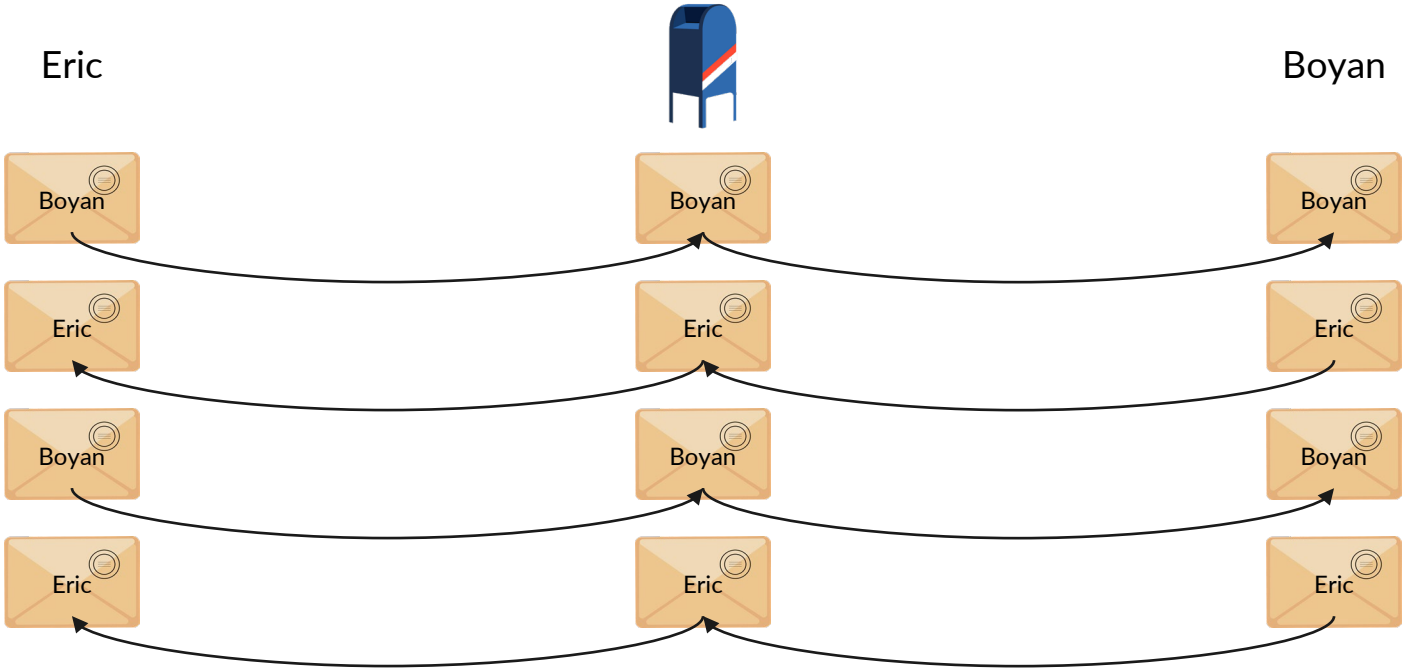
- Post Office can’t deliver the message

Sealed Sender's Anonymity Guarantees

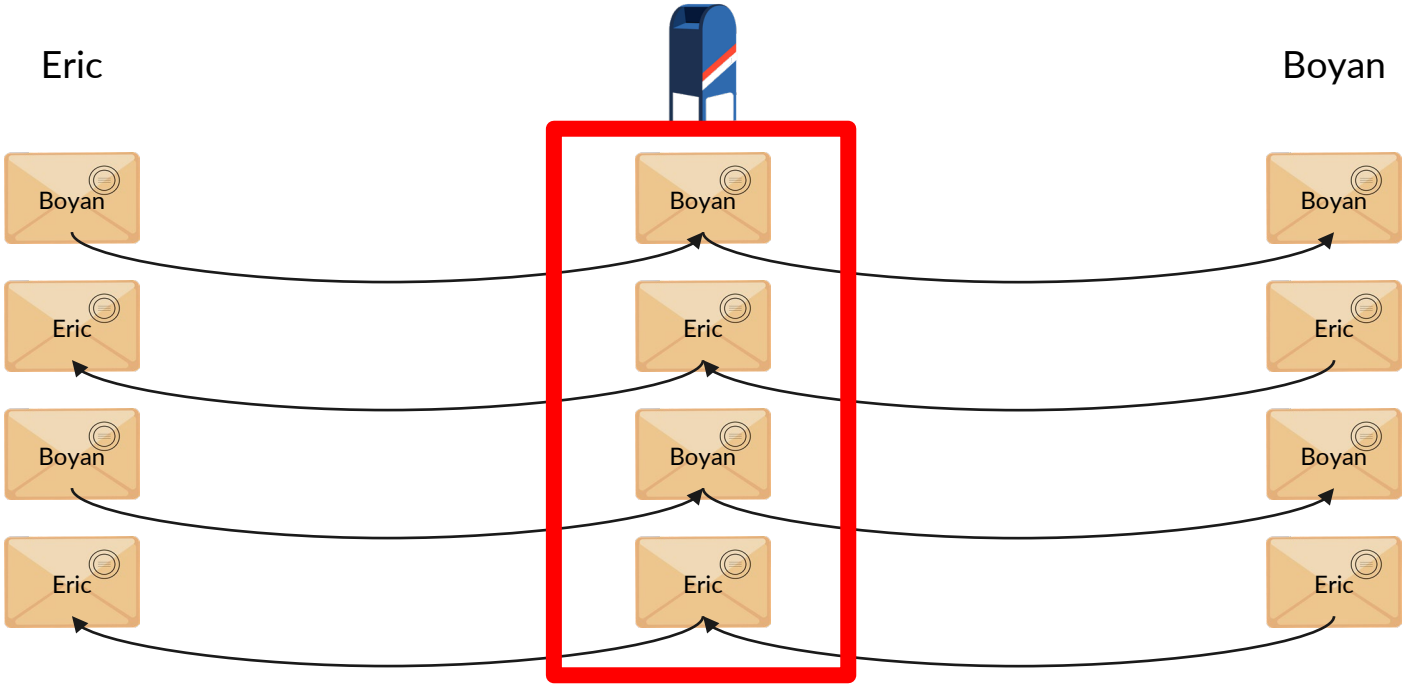
The Long Term

- Over time, Sealed Sender doesn’t prevent the post office from knowing that Boyan and Eric are talking

A Standard Conversation



A Standard Conversation



The Digital World

- Boyan and Eric text back rapidly
  - Delivery receipts are sent within ~2 seconds
- Signal can see messages to Eric are consistently close to messages to Boyan
  - Over time, knows they are talking

Signal's View:

Recipient	Time
⋮	
To: Carol	0 s
To: David	1 s
To: Boyan	1.5 s
To: Eric	3 s
To: David	6 s
To: Alice	7 s

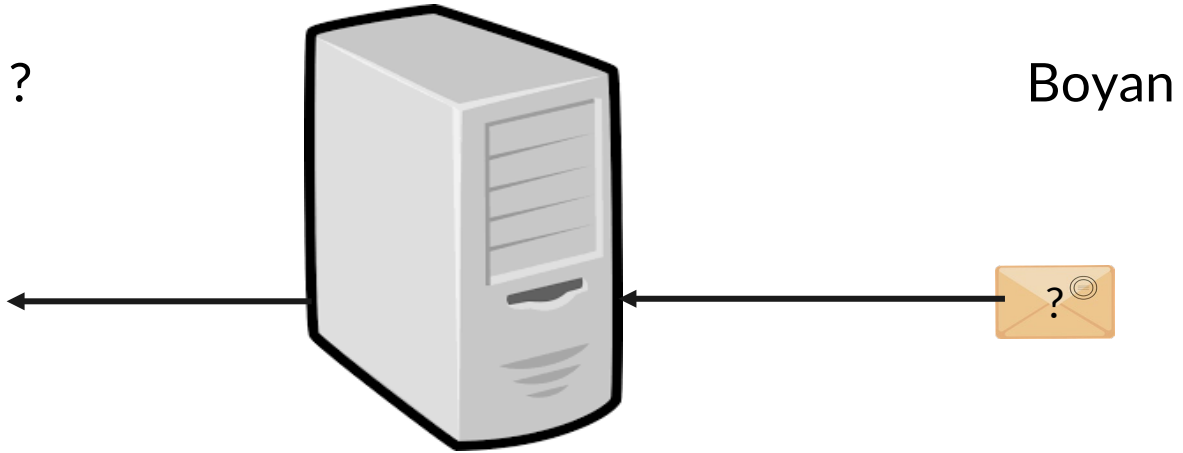
Our Goal

- Server shouldn't directly know that Eric and Boyan are talking
- Avoid timing-based attacks, by either
  - Hiding timing for messages
  - Hiding at least one of the participants

Recipient Anonymity & PIR

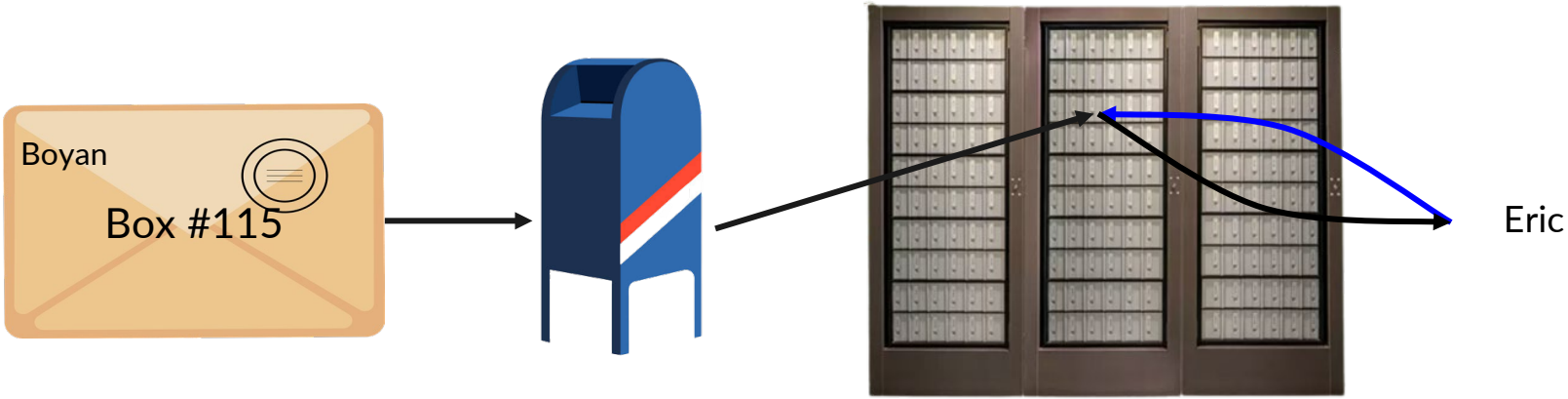
Anonymous Receiving

- If receiver is anonymous, server can't directly deliver it there



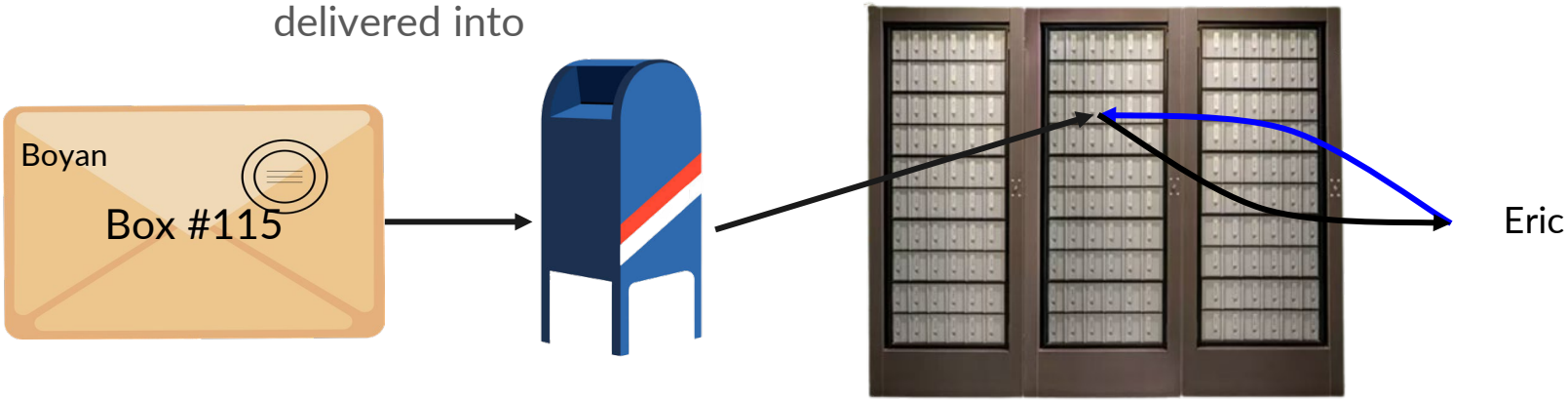
P.O. Boxes

- Can rent without revealing your identity
- Boyan and Eric agree on a box beforehand, then Boyan delivers it to that box



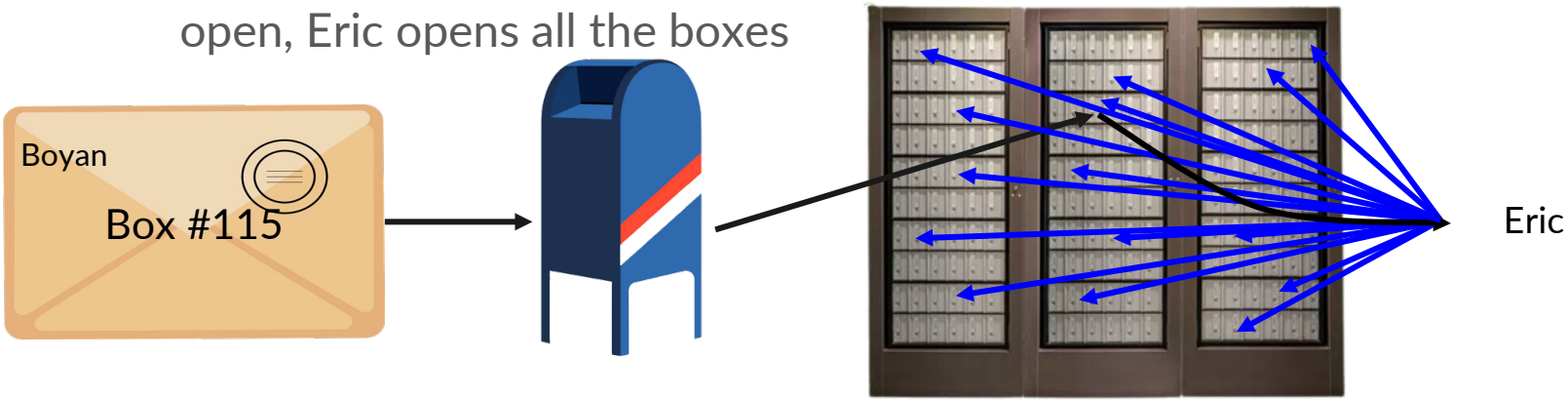
P.O. Boxes

- Can rent without revealing your identity
- Boyan and Eric agree on a box beforehand, then Boyan delivers it to that box
  - Post Office figures out who Eric is when he opens the box Boyan delivered into



Private Letter Retrieval

- Need to break linkage between box Eric accesses and who Eric is talking to
- To prevent the Post Office from knowing which box he needed to open, Eric opens all the boxes

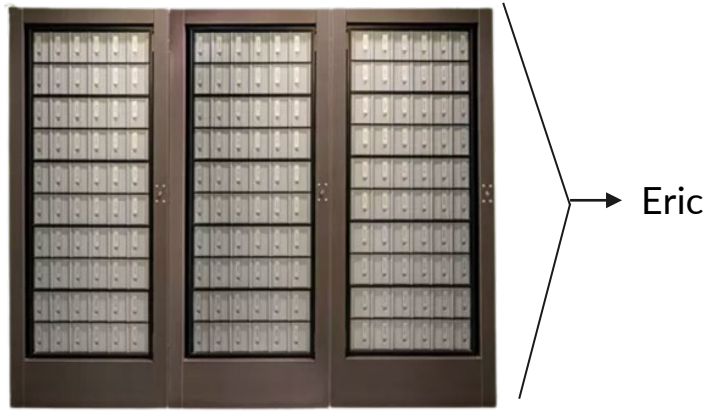


No Delivery

- Eric has to go to the Post Office repeatedly, can't have the message delivered
  - Even if Eric received no mail

Digital Private Letter Retrieval

- Trivial implementation is to just ship everyone a copy of the database
  - Doesn’t violate confidentiality due to encryption
- Huge network costs



Private Information Retrieval

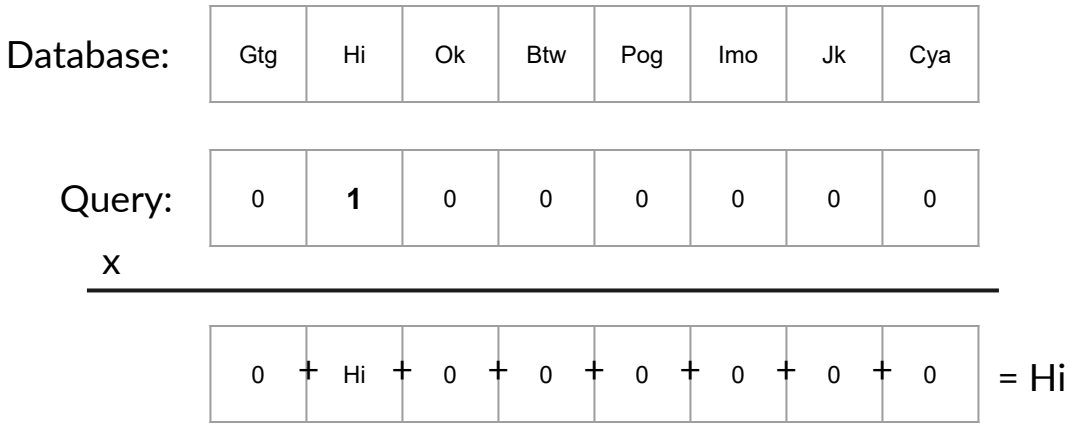
- Messages pushed into digital mailboxes
- Eric sends a query which will operate on every database element indistinguishably
  - Signal can’t tell which element was accessed

Database:

Gtg	Hi	Ok	Btw	Pog	Imo	Jk	Cya

Private Information Retrieval (PIR) Basics

- Query is all 0s and one 1
- Multiply each database element by query
- Add those up to get result
- Response is only the size of a single element



PIR + Sealed Sender = Anonymity

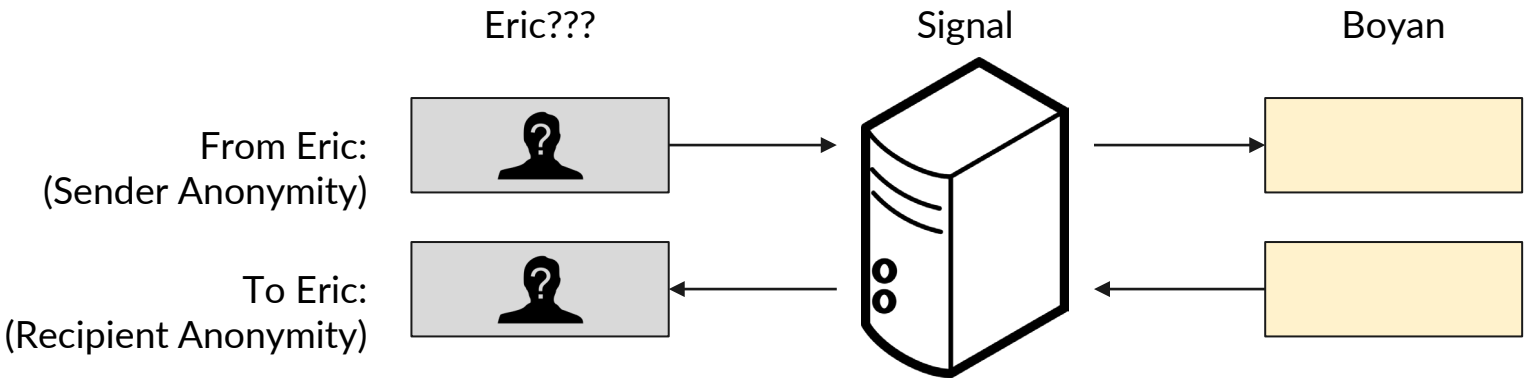
- Sealed Sender provides sender anonymity
- PIR provides recipient anonymity

Our Scheme

= Anonymous
= Not Anonymous

Overall Goal

- One person is always anonymous, and gets both sender and recipient anonymity
  - Signal can't tell Boyan and Eric are talking to each other



Protocol Comparisons

Our protocol:

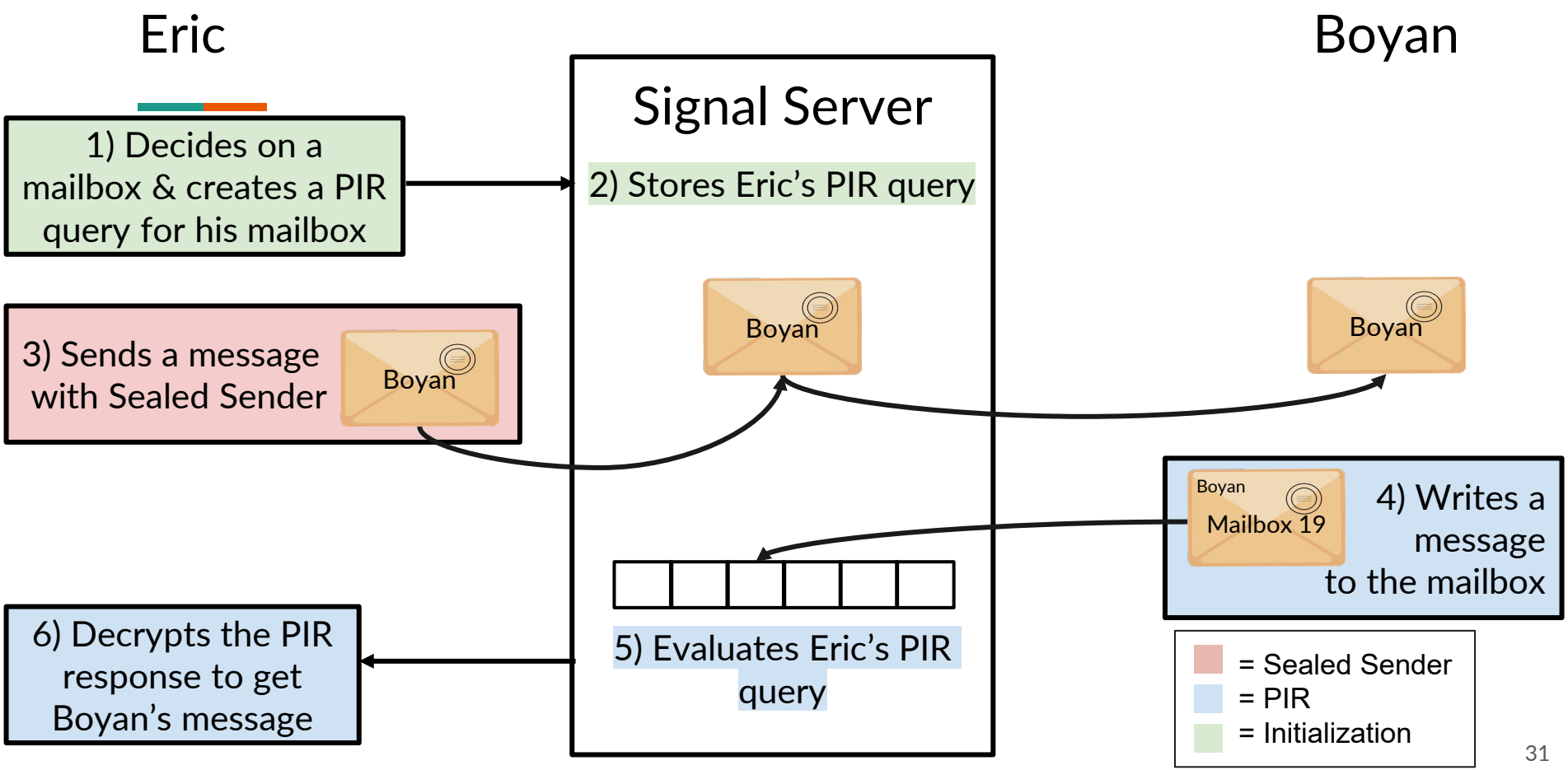
Recipient	Time
⋮	
To: Carols	0
From: Alices	1
To: Boyan	1.5 s
From: Boyan	3 s
To: Davids	6
From: Alice	7

Sealed Sender only:

Recipient	Time
⋮	
To: Carols	0
To: Davids	1
To: Boyan	1.5 s
To: Eric	3 s
To: Davids	6
To: Alices	7

An Asymmetric Protocol

- Only Eric is anonymous
  - Sending and Receiving use different protocols
- This is sufficient to hide that Boyan and Eric are talking
- One user sends through sealed sender and the other writes to a PIR mailbox

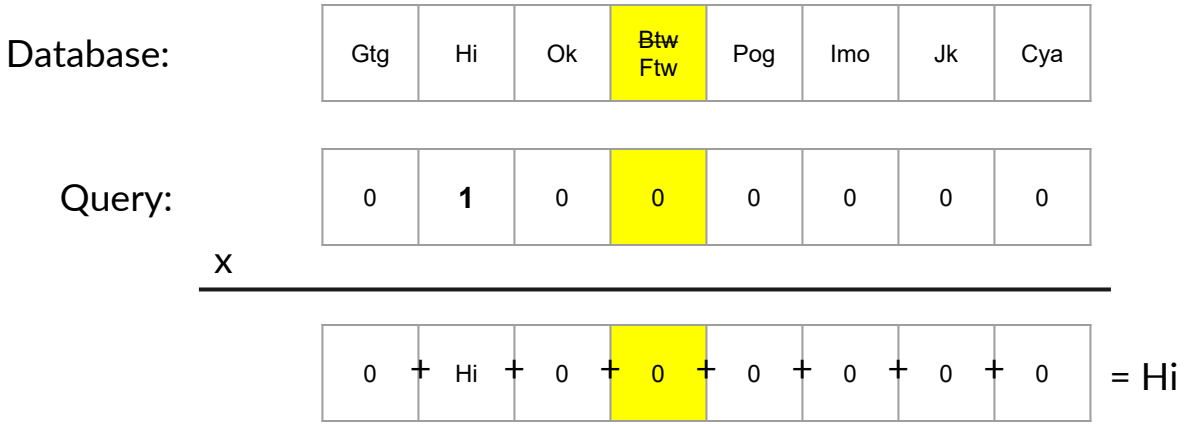

Eric
Boyan
Signal Server
1) Decides on a mailbox & creates a PIR query for his mailbox
2) Stores Eric's PIR query
3) Sends a message with Sealed Sender
Boyan
Boyan
Boyan
Boyan
Mailbox 19
4) Writes a message to the mailbox
5) Evaluates Eric's PIR query
6) Decrypts the PIR response to get Boyan's message
= Sealed Sender
= PIR
= Initialization

Pushing Responses

- Any conversations stay in the same mailbox
  - Queries stay the same
- Since query is always the same, we can have Signal store it
- Signal can re-evaluate periodically and push out the update
  - Even if someone isn't online, they don't lose anonymity

PIR Optimizations

- Can update queries instead of recomputing
  - Store PIR results



Other PIR Details

- Queries can be compressed to only encode for one index, instead of having a ciphertext for each index
  - Query sizes are ~14 KB in state of the art schemes
- High network costs (~2.5 times larger responses)
  - On 2KB elements, we send 200 GB per push
  - Sending un-needed responses
    - Much more bandwidth needed

Takeaways

- Our protocol doesn’t require constant activity
- Hides that Boyan and Eric are talking to each other
- Less computationally expensive than similar protocols

Acknowledgments

- Thanks to MIT PRIMES for making this project possible
- Thanks to our mentors Kyle Hogan and Simon Langowski

Any Questions?