Express: Lowering the Cost of Metadata-hiding Communication with Cryptographic Privacy

Saba Eskandarian, Stanford
Henry Corrigan-Gibbs, MIT CSAIL
Matei Zaharia, Stanford
Dan Boneh, Stanford

Our Story

Our Story

How to Communicate Privately?

Option 1:

End to end encrypted messaging apps

E.g. Signal, WhatsApp

Problem: **metadata**

How to Communicate Privately?

Option 1:

End to end encrypted messaging apps

E.g. Signal, WhatsApp

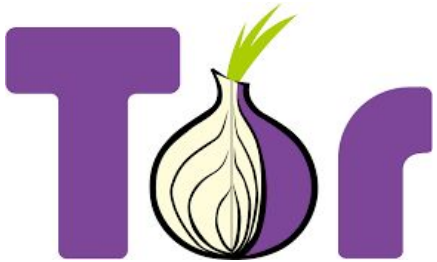
Problem: **metadata**

Option 2:

Anonymizing proxy

E.g. Tor, SecureDrop

Problem: **global adversaries**



How to Communicate Privately?

Option 3: Metadata-hiding communication systems with cryptographic privacy

How to Communicate Privately?

Option 3: Metadata-hiding communication systems with cryptographic privacy

E.g. Riposte, Pung, Talek, Karaoke, Atom, XRD, Verdict, Dissent, ….

How to Communicate Privately?

Option 3: Metadata-hiding communication systems with cryptographic privacy

E.g. Riposte, Pung, Talek, Karaoke, Atom, XRD, Verdict, Dissent, ….

Drawback: **heavy requirements placed on clients**

- Requirement to run in synchronized rounds
- High communication costs

How to Communicate Privately?

Option 3: Metadata-hiding communication systems with cryptographic privacy

E.g. Riposte, Pung, Talek, Karaoke, Atom, XRD, Verdict, Dissent, ….

Drawback: **heavy requirements placed on clients**

- Requirement to run in synchronized rounds
- High communication costs

Can we make metadata-hiding communication work for whistleblowing?

Introducing Express

Communication system designed for practical metadata-hiding whistleblowing

Introducing Express

Communication system designed for practical metadata-hiding whistleblowing

Journalists can register mailboxes for sources to send messages/documents

Introducing Express

Communication system designed for practical metadata-hiding whistleblowing

Journalists can register mailboxes for sources to send messages/documents

Whistleblowers do not need to access the system in synchronized rounds

Introducing Express

Communication system designed for practical metadata-hiding whistleblowing

Journalists can register mailboxes for sources to send messages/documents

Whistleblowers do not need to access the system in synchronized rounds

<u>Asymptotic improvements:</u>
client computation costs O(1)
communication costs O(1)
(both previously O(√N))

Introducing Express

Communication system designed for practical metadata-hiding whistleblowing

Journalists can register mailboxes for sources to send messages/documents

Whistleblowers do not need to access the system in synchronized rounds

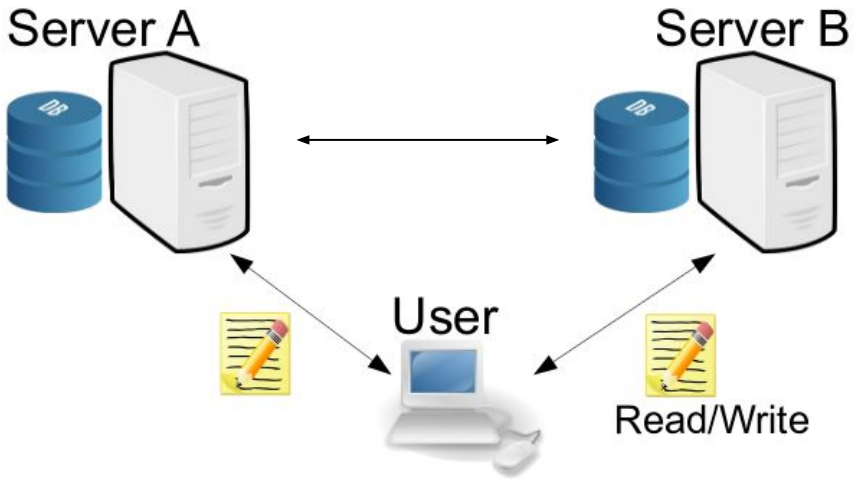
Asymptotic improvements:
client computation costs O(1)
communication costs O(1)
(both previously O(√N))

Practical improvements:
6x improvement in server computation time
8x improvement in client computation time
>10x improvement in communication costs
6x reduction in dollar cost to run system

Express Overview

2 server system, secure against:

- Arbitrarily many corrupt users
- Up to one corrupt server



Express Overview

2 server system, secure against:

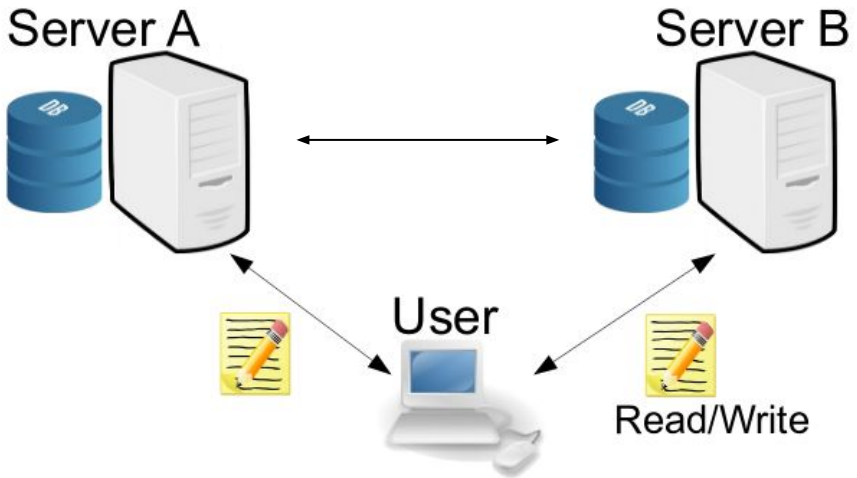
- Arbitrarily many corrupt users
- Up to one corrupt server

Supported operations:

Register mailbox

(Private) write to mailbox

Read from mailbox



Express Overview

2 server system, secure against:

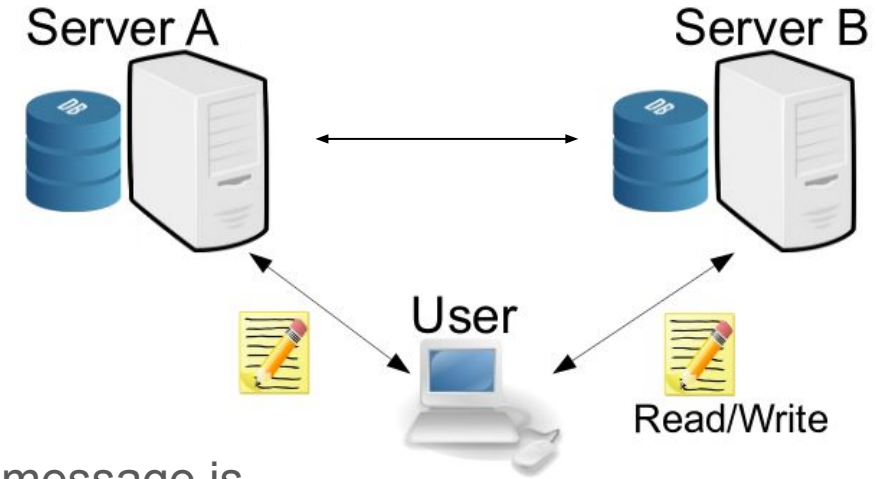
- Arbitrarily many corrupt users
- Up to one corrupt server

Supported operations:

Register mailbox

(Private) write to mailbox

Read from mailbox



Security: can't tell who the *recipient* of a message is

Express Overview

2 server system, secure against:

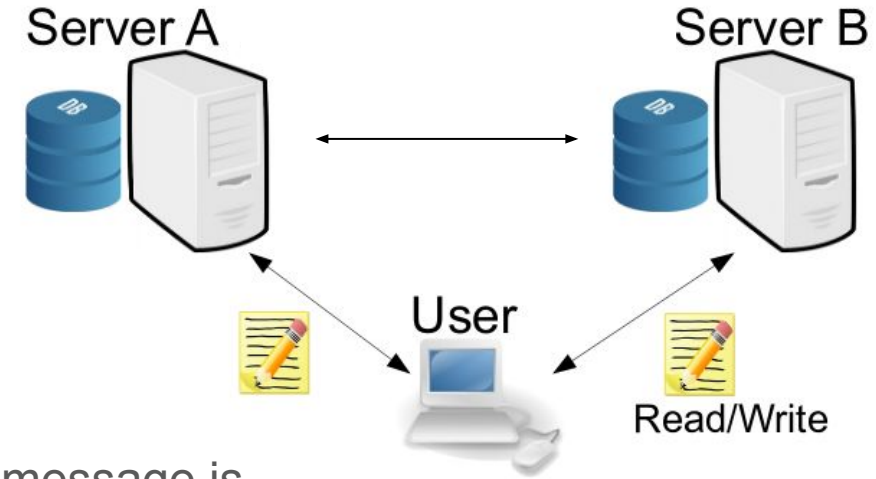
- Arbitrarily many corrupt users
- Up to one corrupt server

Supported operations:

Register mailbox

(Private) write to mailbox

Read from mailbox



Security: can't tell who the *recipient* of a message is

Assumption: user knows "address" of mailbox to which it sends message

Tool: Private Writing with Distributed Point Functions

Point function: a function that is zero everywhere, except at one point

Distributed Point Functions and their Applications, Niv Gilboa, Yuval Ishai, *Eurocrypt'14.*

Tool: Private Writing with Distributed Point Functions

Point function: a function that is zero everywhere, except at one point

x	f(x)
0	0
1	0
2	0
3	“Hi!”
4	0

Distributed Point Functions and their Applications, Niv Gilboa, Yuval Ishai, *Eurocrypt’14.*

Tool: Private Writing with Distributed Point Functions

Point function: a function that is zero everywhere, except at one point

x	$f_1(x)$
0	“abc”
1	“xf$”
2	“^tg”
3	“!7≈”
4	“jhV”

$\oplus$

x	$f_2(x)$
0	“abc”
1	“xf$”
2	“^tg”
3	“‘2!)”
4	“jhV”

=

x	f(x)
0	0
1	0
2	0
3	“Hi!”
4	0

Distributed Point Functions and their Applications, Niv Gilboa, Yuval Ishai, *Eurocrypt’14.*

Tool: Private Writing with Distributed Point Functions

Point function: a function that is zero everywhere, except at one point

Distributed point function: technique for efficiently splitting a point function into two pieces, each a (non-point) function whose XOR is the original point function

x	$f_1(x)$
0	“abc”
1	“xf$”
2	“^tg”
3	“!7≈”
4	“jhV”

⊕

x	$f_2(x)$
0	“abc”
1	“xf$”
2	“^tg”
3	“‘2!)”
4	“jhV”

=

x	f(x)
0	0
1	0
2	0
3	“Hi!”
4	0

Key features:

- concise representation
- fast to generate

Distributed Point Functions and their Applications, Niv Gilboa, Yuval Ishai, *Eurocrypt’14.*

Tool: Private Writing with Distributed Point Functions

Addr	Data
0	0
1	0
2	0
3	0
4	0

Addr	Data
0	0
1	0
2	0
3	0
4	0

Distributed Point Functions and their Applications, Niv Gilboa, Yuval Ishai, *Eurocrypt’14.*
Private Information Storage, Rafail Ostrovsky, Victor Shoup, *STOC’97*

Tool: Private Writing with Distributed Point Functions

x	f(x)
0	0
1	0
2	0
3	“Hi!”
4	0

Addr	Data
0	0
1	0
2	0
3	0
4	0

Addr	Data
0	0
1	0
2	0
3	0
4	0

Distributed Point Functions and their Applications, Niv Gilboa, Yuval Ishai, *Eurocrypt’14.*
Private Information Storage, Rafail Ostrovsky, Victor Shoup, *STOC’97*

Tool: Private Writing with Distributed Point Functions

x	$f_1(x)$
0	“abc”
1	“xf$”
2	“^tg”
3	“!7≈”
4	“jhV”

x	$f_2(x)$
0	“abc”
1	“xf$”
2	“^tg”
3	“‘2!)”
4	“jhV”

Addr	Data
0	0
1	0
2	0
3	0
4	0

Addr	Data
0	0
1	0
2	0
3	0
4	0

Distributed Point Functions and their Applications, Niv Gilboa, Yuval Ishai, *Eurocrypt’14.*
Private Information Storage, Rafail Ostrovsky, Victor Shoup, *STOC’97*

Tool: Private Writing with Distributed Point Functions

f_1

f_2

Addr	Data
0	0
1	0
2	0
3	0
4	0

Addr	Data
0	0
1	0
2	0
3	0
4	0

Distributed Point Functions and their Applications, Niv Gilboa, Yuval Ishai, *Eurocrypt'14.*
Private Information Storage, Rafail Ostrovsky, Victor Shoup, *STOC'97*

Tool: Private Writing with Distributed Point Functions

f_1

f_2

Addr	Data
0	$f_1(0)$
1	$f_1(1)$
2	$f_1(2)$
3	$f_1(3)$
4	$f_1(4)$

Addr	Data
0	$f_2(0)$
1	$f_2(1)$
2	$f_2(2)$
3	$f_2(3)$
4	$f_2(4)$

Distributed Point Functions and their Applications, Niv Gilboa, Yuval Ishai, *Eurocrypt'14.*
Private Information Storage, Rafail Ostrovsky, Victor Shoup, *STOC'97*

Tool: Private Writing with Distributed Point Functions

f_1

f_2

Addr	Data
0	“abc”
1	“xf$”
2	“^tg”
3	“!7≈”
4	“jhV”

Addr	Data
0	“abc”
1	“xf$”
2	“^tg”
3	“‘2!)”
4	“jhV”

Distributed Point Functions and their Applications, Niv Gilboa, Yuval Ishai, *Eurocrypt’14.*
Private Information Storage, Rafail Ostrovsky, Victor Shoup, *STOC’97*

Tool: Private Writing with Distributed Point Functions

f_1

f_2

Addr	Data
0	“abc”
1	“xf$”
2	“^tg”
3	“!7≈”
4	“jhV”

Addr	Data
0	“abc”
1	“xf$”
2	“^tg”
3	“‘2!)”
4	“jhV”

⊕

“Hi!”

Distributed Point Functions and their Applications, Niv Gilboa, Yuval Ishai, *Eurocrypt’14.*
Private Information Storage, Rafail Ostrovsky, Victor Shoup, *STOC’97*

Hiding Data

How to prevent curious clients from reading others' mailboxes?

Addr	Data
0	“abc”
1	“xf$”
2	“^tg”
3	“!7≈”
4	“jhV”

Addr	Data
0	“abc”
1	“xf$”
2	“^tg”
3	“‘2!)”
4	“jhV”

Hiding Data

How to prevent curious clients from reading others' mailboxes?

Encrypt each row with a different key held by the owner of the mailbox

Addr	Data	Key
0	“abc”	k_{NYT}
1	“xf$”	k_{WaPo}
2	“^tg”	k_{WSJ}
3	“!7≈”	$k_{Buzzfeed}$
4	“jhV”	$k_{Inquirer}$

Addr	Data	Key
0	“abc”	k_{NYT}
1	“xf$”	k_{WaPo}
2	“^tg”	k_{WSJ}
3	“‘2!)”	$k_{Buzzfeed}$
4	“jhV”	$k_{Inquirer}$

Hiding Data

How to prevent curious clients from reading others' mailboxes?

Encrypt each row with a different key held by the owner of the mailbox

Different key sent to each server, encrypt in CTR mode to allow adding messages

Addr	Data	Key
0	"abc"	k_{NYT1}
1	"xf$"	k_{WaPo1}
2	"^tg"	k_{WSJ1}
3	"!7≈"	$k_{Buzzfeed1}$
4	"jhV"	$k_{Inquirer1}$

Addr	Data	Key
0	"abc"	k_{NYT2}
1	"xf$"	k_{WaPo2}
2	"^tg"	k_{WSJ2}
3	"'2!)"	$k_{Buzzfeed2}$
4	"jhV"	$k_{Inquirer2}$

Hiding *Meta*data

Construction thus far vulnerable to polling attack:

Attacker reads every row after each write to see which one was changed

Hiding *Meta*data

Construction thus far vulnerable to polling attack:

Attacker reads every row after each write to see which one was changed

Solution: servers non-interactively re-randomize every row after each write

Additional cost is low since they already write to each row

Hiding *Meta*data

Data Server A

Addr.	Key	Data
0	k_{A0}	abc + f(k_{A0}, c)
1	k_{A1}	xf$ + f(k_{A1}, c)
2	k_{A2}	!7≈ + f(k_{A2}, c)
3	k_{A3}	^tg + f(k_{A3}, c)

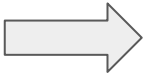
logN bits 128 bits Data size

Hiding *Meta*data

Data Server A

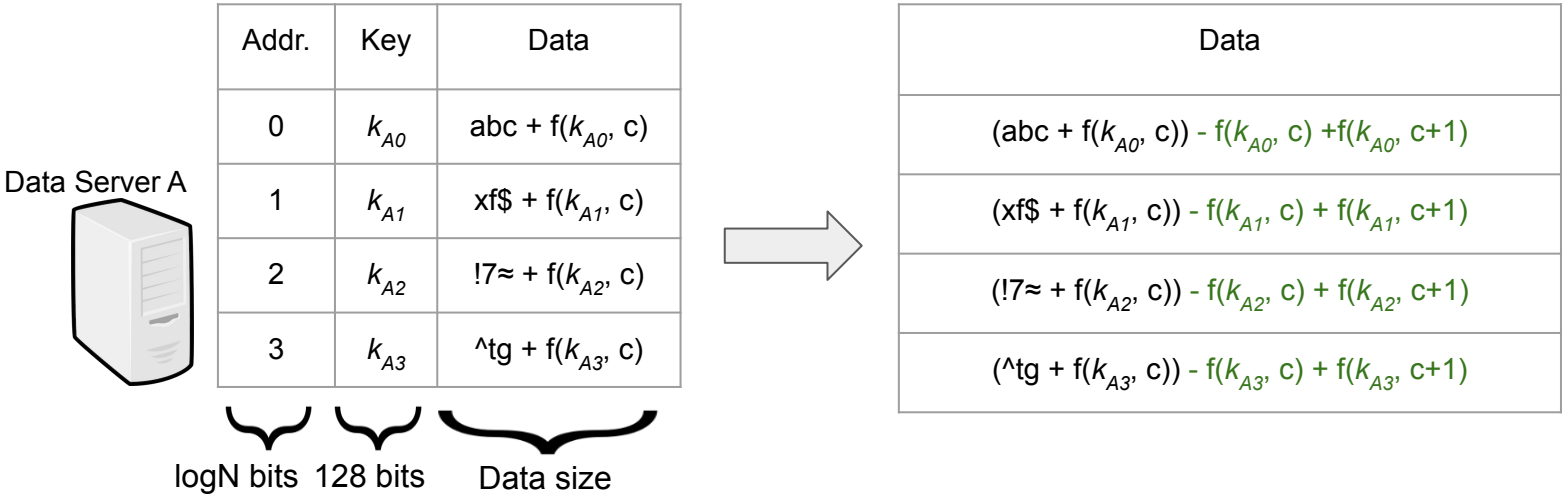
Addr.	Key	Data
0	k_{A0}	abc + f(k_{A0}, c)
1	k_{A1}	xf$ + f(k_{A1}, c)
2	k_{A2}	!7≈ + f(k_{A2}, c)
3	k_{A3}	^tg + f(k_{A3}, c)

logN bits 128 bits Data size



Data
(abc + f(k_{A0}, c)) - f(k_{A0}, c) +f(k_{A0}, c+1)
(xf$ + f(k_{A1}, c)) - f(k_{A1}, c) + f(k_{A1}, c+1)
(!7≈ + f(k_{A2}, c)) - f(k_{A2}, c) + f(k_{A2}, c+1)
(^tg + f(k_{A3}, c)) - f(k_{A3}, c) + f(k_{A3}, c+1)

Hiding *Meta*data



Optimization: only rerandomize just before a read, not after each write

Plausible Deniability

How to protect privacy of whistleblowers if *all users* are whistleblowers?

Conscript your friends into larger anonymity sets with JavaScript, Henry Corrigan-Gibbs, Bryan Ford, *WPES'13*

Plausible Deniability

How to protect privacy of whistleblowers if *all users* are whistleblowers?

Idea: Cooperative web sites embed JS that sends dummy write requests

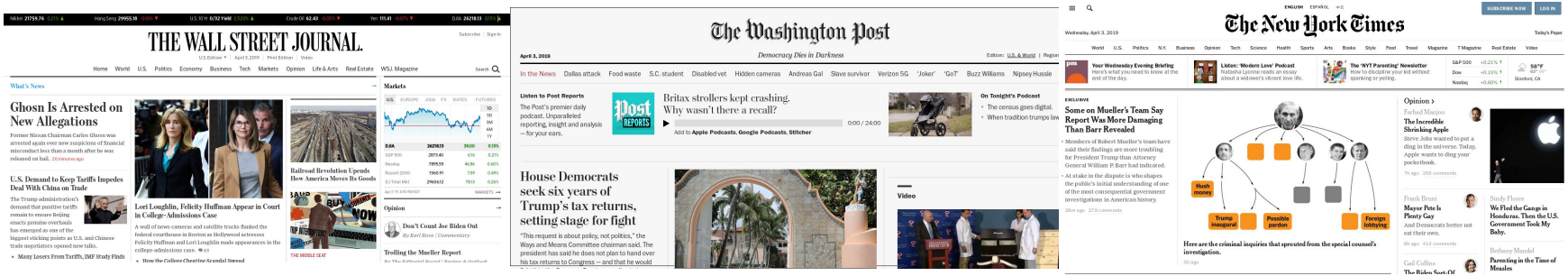
Conscript your friends into larger anonymity sets with JavaScript, Henry Corrigan-Gibbs, Bryan Ford, *WPES'13*

Plausible Deniability

How to protect privacy of whistleblowers if *all users* are whistleblowers?

Idea: Cooperative web sites embed JS that sends dummy write requests

- Incentives properly aligned for news organizations
- Metadata-hiding means we only need 1 recipient mailbox for dummy writes
- Client-side costs low enough to not affect browsing experience



Conscript your friends into larger anonymity sets with JavaScript, Henry Corrigan-Gibbs, Bryan Ford, *WPES'13*

Handling Disruptive Users

Any number of users can act maliciously in arbitrary ways

Handling Disruptive Users

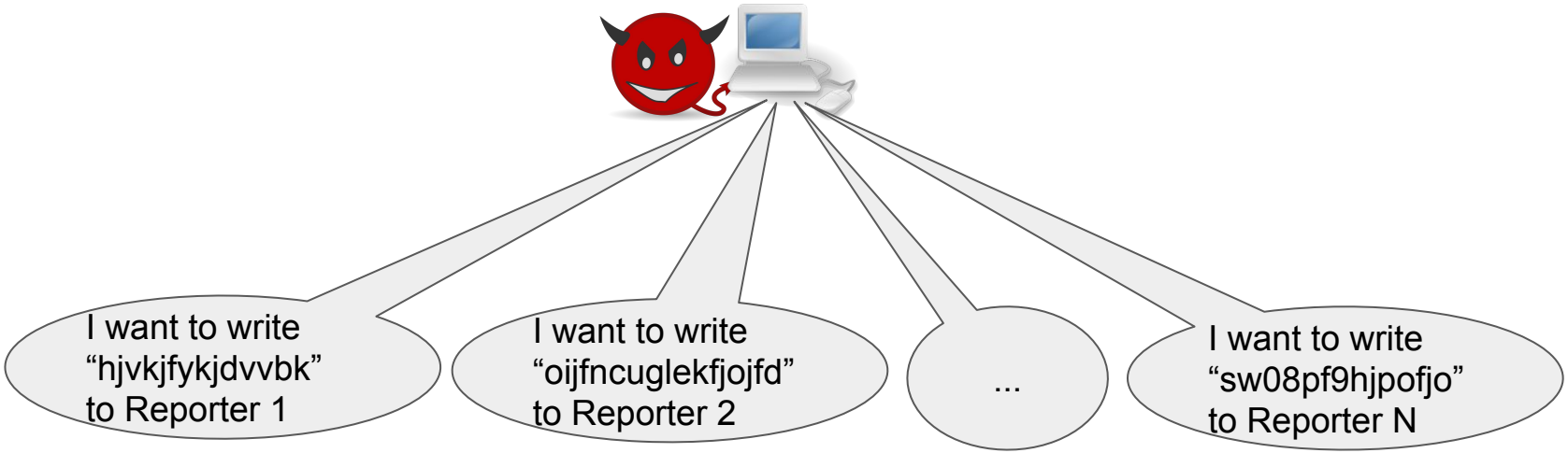
Any number of users can act maliciously in arbitrary ways

Two kinds of attacks:

1. Disruptive user writes to others' mailbox
2. Disruptive user sends malformed DPF to write to many mailboxes

Handling Disruptive Users

Problem: disruptive user writes to others' mailboxes



Virtual Addresses

Problem: disruptive user writes to others' mailboxes

Solution: hide mailboxes in exponentially large address space

Addr	Data
0	“abc”
1	“xf$”
2	“^tg”
...	...
...	...
...	...
$2^{128}-2$	“!7≈”
$2^{128}-1$	“jhV”

Virtual Addresses

Problem: disruptive user writes to others' mailboxes

Solution: hide mailboxes in exponentially large address space

New problem: too many addresses, bad performance

Addr	Data
0	“abc”
1	“xf$”
2	“^tg”
...	...
...	...
...	...
$2^{128}-2$	“!7≈”
$2^{128}-1$	“jhV”

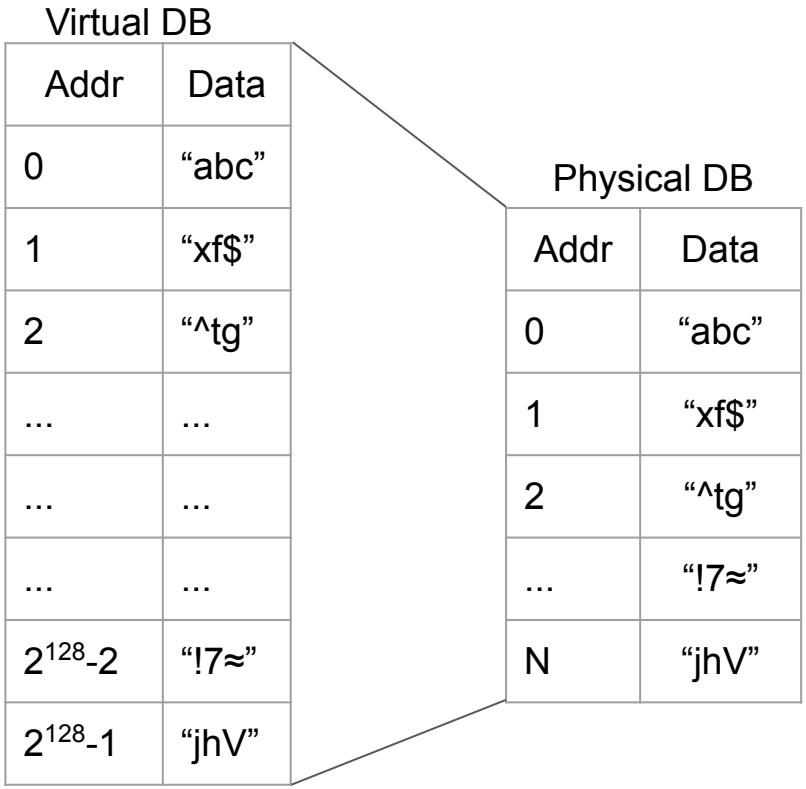
Virtual Addresses

Problem: disruptive user writes to others' mailboxes

Solution: hide mailboxes in exponentially large address space

New problem: too many addresses, bad performance

Solution: virtual addresses



Auditing

Problem: disruptive user sends malformed DPF to write to many mailboxes

x	f(x)
0	989f4
1	dDf73
...	
2^{128}-2	08dji3
2^{128}-1	89hfif

Auditing

Problem: disruptive user sends malformed DPF to write to many mailboxes

Solution: servers blindly *audit* all incoming write requests

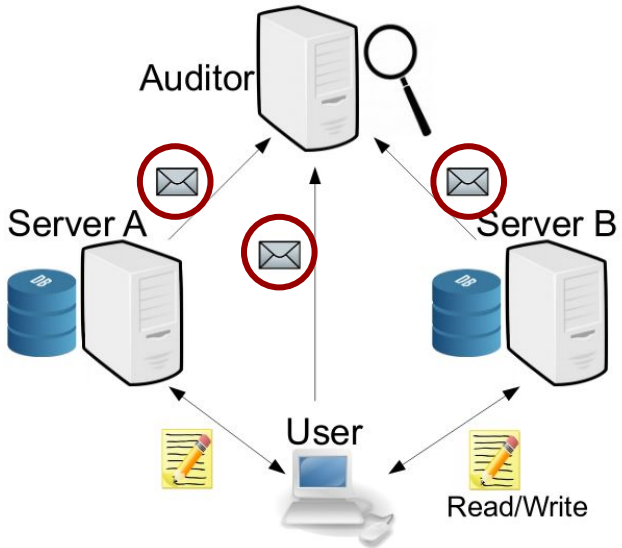
Auditing

Problem: disruptive user sends malformed DPF to write to many mailboxes

Solution: servers blindly *audit* all incoming write requests

Prior work: third server audits requests

- O(√N) communication
- O(√N) client/auditor computation



Riposte: An Anonymous Messaging System Handling Millions of Users, Henry Corrigan-Gibbs, Dan Boneh, David Mazieres, *Oakland'15*.

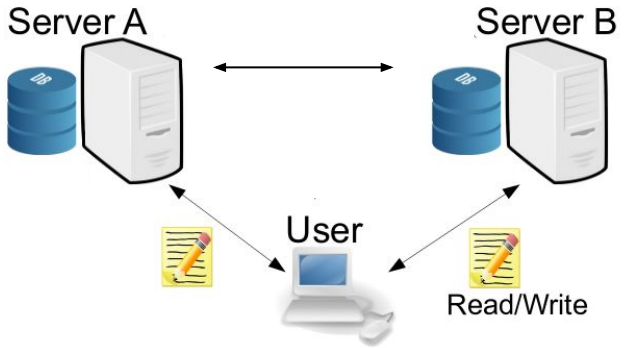
Auditing

Problem: disruptive user sends malformed DPF to write to many mailboxes

Solution: servers blindly *audit* all incoming write requests

<u>New auditing protocol:</u>

- O(1) communication
- O(1) client computation
- No additional server!



Auditing

Goal: prove vectors of DPF evaluations only differ at one point

Function Secret Sharing: Improvements and Extensions, Elette Boyle, Niv Gilboa, Yuval Ishai, *CCS'16.*

Auditing

Goal: prove vectors of DPF evaluations only differ at one point

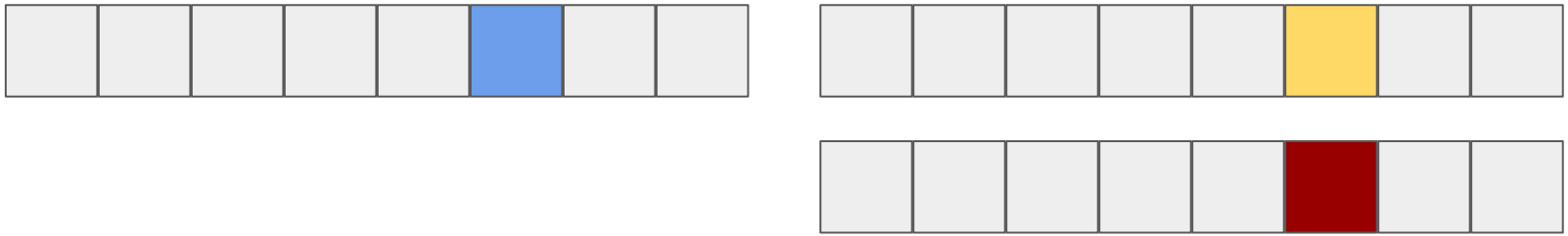
Prior work has a semihonest solution where servers use a cheap MPC (only 2 multiplications) to verify this property.

Function Secret Sharing: Improvements and Extensions, Elette Boyle, Niv Gilboa, Yuval Ishai, *CCS'16.*

Auditing

Goal: prove vectors of DPF evaluations only differ at one point

Prior work has a semihonest solution where servers use a cheap MPC (only 2 multiplications) to verify this property.



Issue: malicious server can guess & check the nonzero entry

Function Secret Sharing: Improvements and Extensions, Elette Boyle, Niv Gilboa, Yuval Ishai, *CCS'16.*

Auditing

Tool: secret-shared non-interactive proofs (SNIPs)

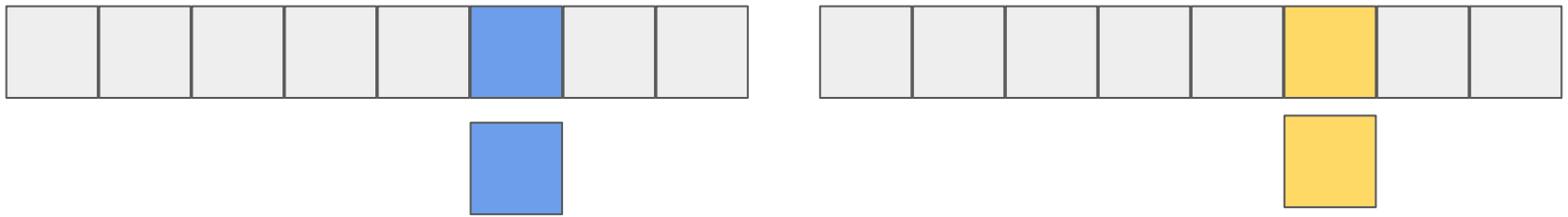
Idea: client sends SNIP proof to servers that honest evaluation of the semihonest protocol accepts the DPF

Auditing

Tool: secret-shared non-interactive proofs (SNIPs)

Idea: client sends SNIP proof to servers that honest evaluation of the semihonest protocol accepts the DPF

Key new trick: client knows the nonzero index & value, only needs O(1) work to prove things about non-zero entry, even though servers did O(N) work.

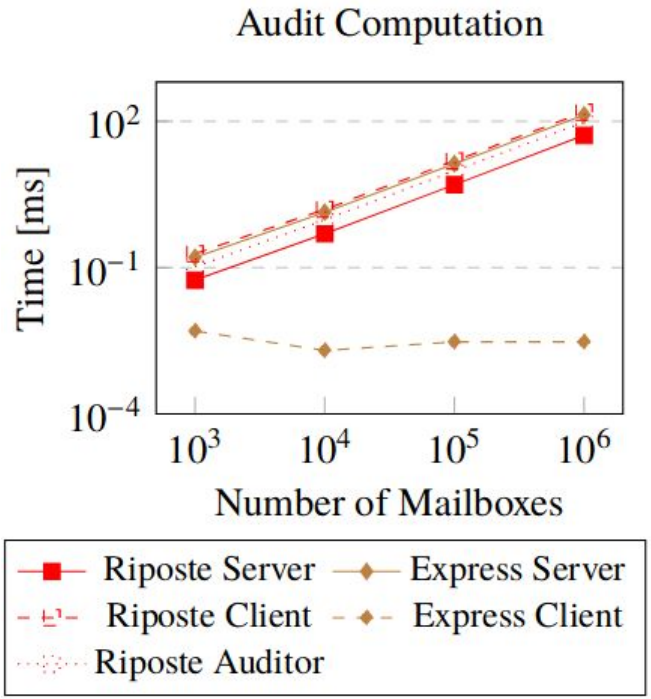


Evaluation

Evaluation

Auditing Protocol

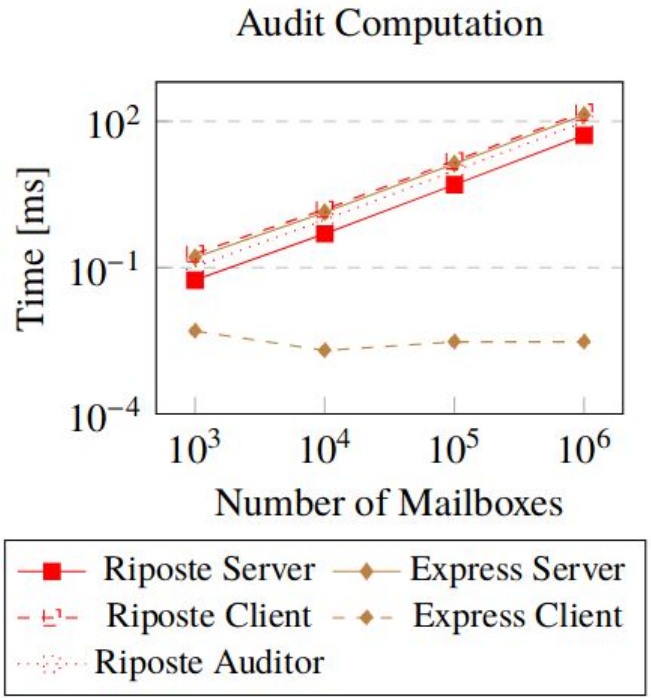
- Client runs in under 5 *micro*seconds always



Evaluation

Auditing Protocol

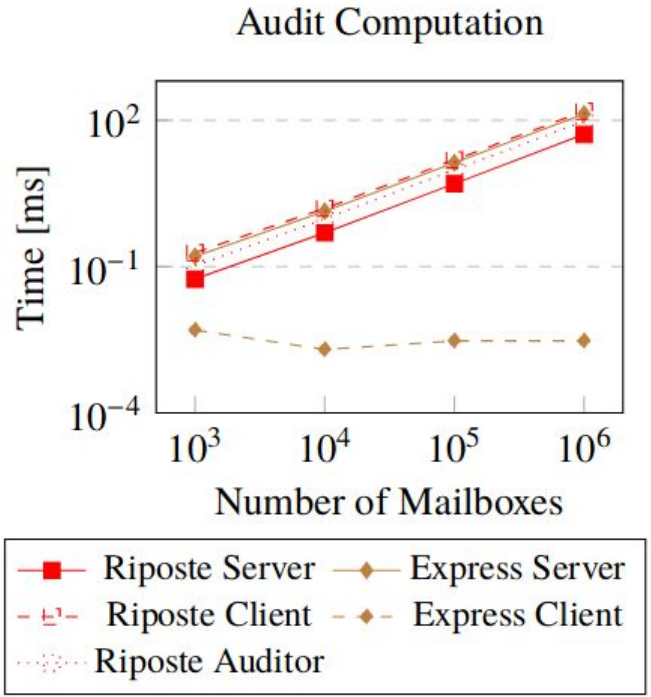
- Client runs in under 5 *micro*seconds always
- 55,000x faster than Riposte for 1m mailboxes



Evaluation

Auditing Protocol

- Client runs in under 5 *micro*seconds always
- 55,000x faster than Riposte for 1m mailboxes
- Enables 8x reduction in overall client computation (now 20ms)

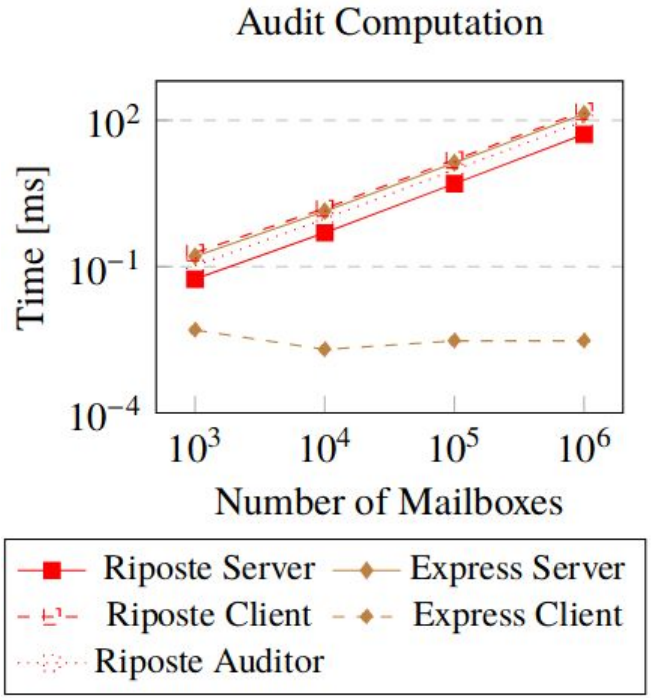


Riposte: An Anonymous Messaging System Handling Millions of Users, Henry Corrigan-Gibbs, Dan Boneh, David Mazieres, *Oakland'15.*

Evaluation

Auditing Protocol

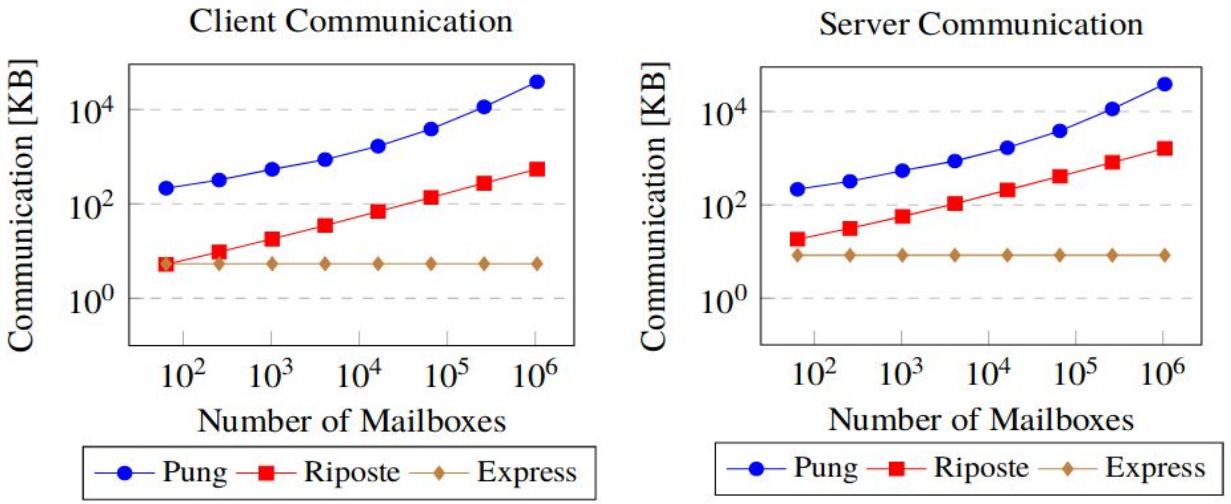
- Client runs in under 5 *micro*seconds always
- 55,000x faster than Riposte for 1m mailboxes
- Enables 8x reduction in overall client computation (now 20ms)
- Comparable on server, where auditing is not the bottleneck



Riposte: An Anonymous Messaging System Handling Millions of Users, Henry Corrigan-Gibbs, Dan Boneh, David Mazieres, *Oakland'15.*

Evaluation

Communication Costs



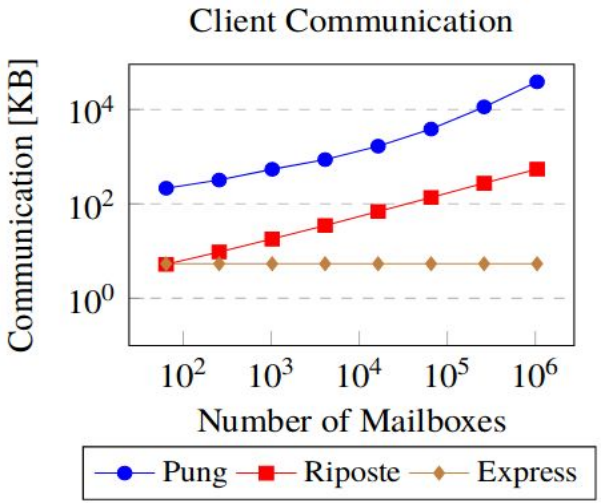
(Sending 160B messages)

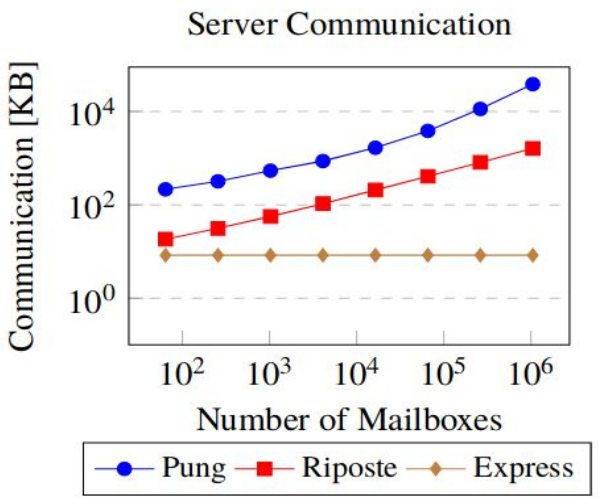
Riposte: An Anonymous Messaging System Handling Millions of Users, Henry Corrigan-Gibbs, Dan Boneh, David Mazieres, *Oakland'15.*
Unobservable Communication over Fully Untrusted Infrastructure, Sebastian Angel, Srinath Setty, *OSDI'16.*

Evaluation

Communication Costs

For 2^{14} mailboxes:
13x improvement on client, 25x on server





(Sending 160B messages)

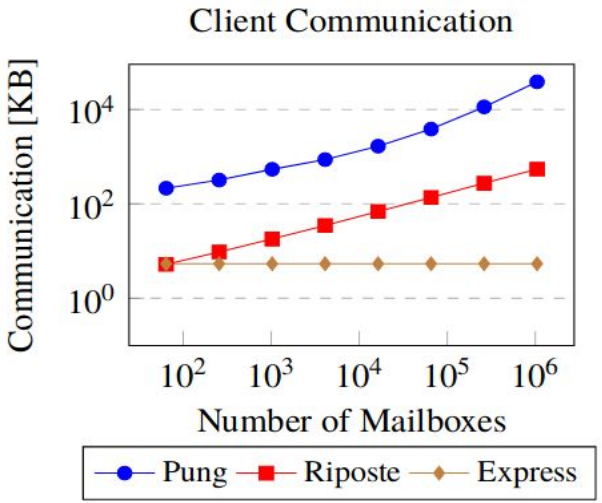
Riposte: An Anonymous Messaging System Handling Millions of Users, Henry Corrigan-Gibbs, Dan Boneh, David Mazieres, *Oakland'15.*
Unobservable Communication over Fully Untrusted Infrastructure, Sebastian Angel, Srinath Setty, *OSDI'16.*

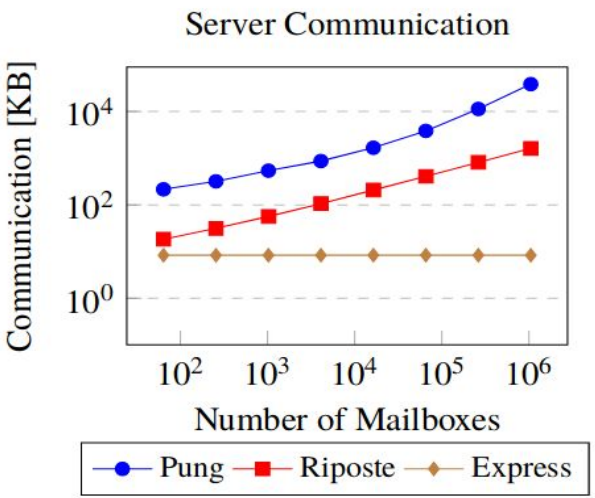
Evaluation

Communication Costs

For 2^{14} mailboxes:
13x improvement on client, 25x on server

For 2^{20} mailboxes:
101x improvement on client, 195x on server





(Sending 160B messages)

Riposte: An Anonymous Messaging System Handling Millions of Users, Henry Corrigan-Gibbs, Dan Boneh, David Mazieres, *Oakland'15.*
Unobservable Communication over Fully Untrusted Infrastructure, Sebastian Angel, Srinath Setty, *OSDI'16.*

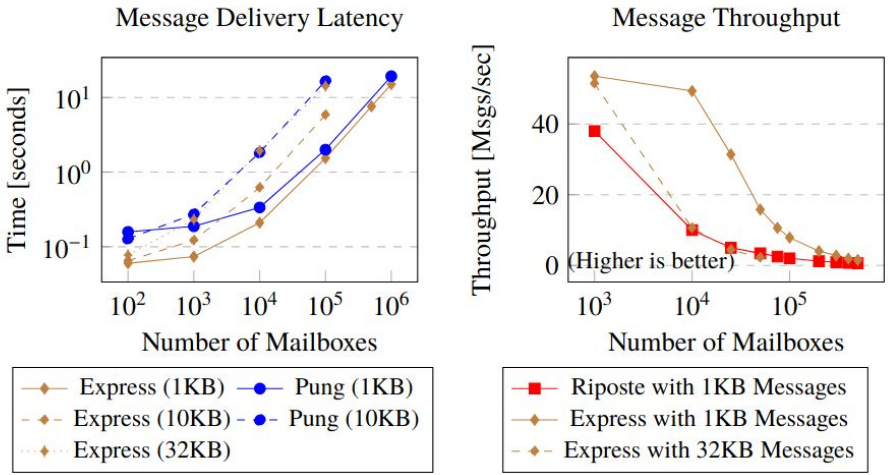
Evaluation

Server-side costs

Modest improvements in server-side performance

- 1.4-6.3x throughput of Riposte (1KB msg)
- 1.3-2.6x faster than Pung (1KB msg)
- 2-2.9x faster than Pung (10KB msg)

32KB message performance still comparable to prior work on smaller sizes



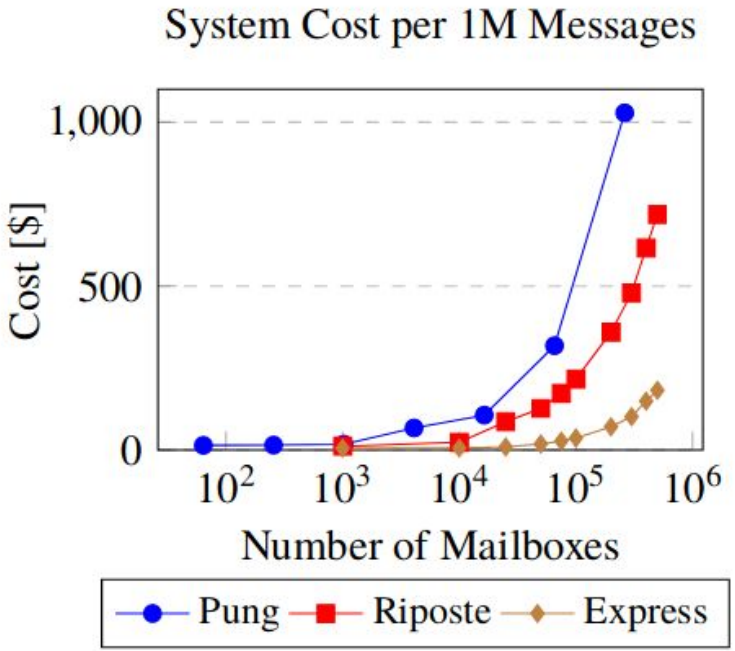
Riposte: An Anonymous Messaging System Handling Millions of Users, Henry Corrigan-Gibbs, Dan Boneh, David Mazieres, *Oakland'15.*
Unobservable Communication over Fully Untrusted Infrastructure, Sebastian Angel, Srinath Setty, *OSDI'16.*

Evaluation

Dollar Cost

Estimate based on GCP prices for servers and data egress

Cost per 1M messages for 100K registered mailboxes 6x less than Riposte



Riposte: An Anonymous Messaging System Handling Millions of Users, Henry Corrigan-Gibbs, Dan Boneh, David Mazieres, *Oakland'15.*
Unobservable Communication over Fully Untrusted Infrastructure, Sebastian Angel, Srinath Setty, *OSDI'16.*

Express

Metadata-hiding communication system with application to private whistleblowing

- Asymptotic speedup from $O(\sqrt{N})$ to $O(1)$ for auditing
- Speedup of 8x on client, up to 6x on server (compared to Riposte)
- 6x lower dollar cost to operate system
- 13-7,000x or more reduction in communication costs

Paper: https://arxiv.org/pdf/1911.09215.pdf
Code: https://github.com/SabaEskandarian/Express
Contact: saba@cs.stanford.edu