

# Express: Private Communication without Synchronization

Saba Eskandarian, Henry Corrigan-Gibbs, Matei Zaharia, Dan Boneh

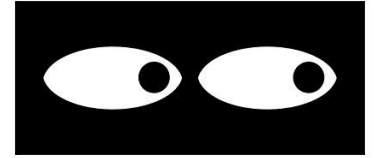
# Our Story



# 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**

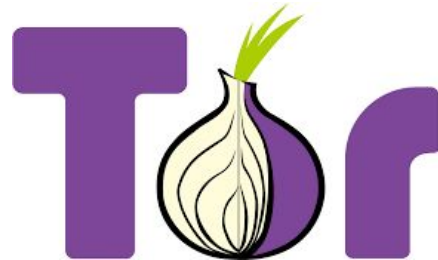


Option 2:

Anonymizing proxy

E.g. Tor, SecureDrop

Problem: **global adversaries**



# How to Communicate Privately?

Option 3: Metadata-hiding communication systems

# How to Communicate Privately?

Option 3: Metadata-hiding communication systems

E.g. Riposte, Pung, Vuvuzela, Talek, Alpenhorn, Stadium, Karaoke, Atom, XRD, Verdict, Dissent, Herbivore, ....



# How to Communicate Privately?

Option 3: Metadata-hiding communication systems

E.g. Riposte, Pung, Vuvuzela, Talek, Alpenhorn, Stadium, Karaoke, Atom, XRD, Verdict, Dissent, Herbivore, ....

Drawback: **Require running in rounds/synchronization**

# How to Communicate Privately?

Option 3: Metadata-hiding communication systems

E.g. Riposte, Pung, Vuvuzela, Talek, Alpenhorn, Stadium, Karaoke, Atom, XRD, Verdict, Dissent, Herbivore, ....

Drawback: **Require running in rounds/synchronization**

Can we get any metadata-hiding system that does not require running in rounds?

# Introducing Express

First metadata-hiding communication system with no requirement for users to contact server at regular intervals

# Introducing Express

First metadata-hiding communication system with no requirement for users to contact server at regular intervals

Journalists can register mailboxes for sources to send messages/documents

# Introducing Express

First metadata-hiding communication system with no requirement for users to contact server at regular intervals

Journalists can register mailboxes for sources to send messages/documents

## Asymptotic improvements:

client computation costs  $O(\log N)$

communication costs  $O(\log N)$

(both previously  $O(\sqrt{N})$ )

# Introducing Express

First metadata-hiding communication system with no requirement for users to contact server at regular intervals

Journalists can register mailboxes for sources to send messages/documents

## Asymptotic improvements:

client computation costs  $O(\log N)$

communication costs  $O(\log N)$

(both previously  $O(\sqrt{N})$ )

## Practical improvements:

5x improvement in server computation time

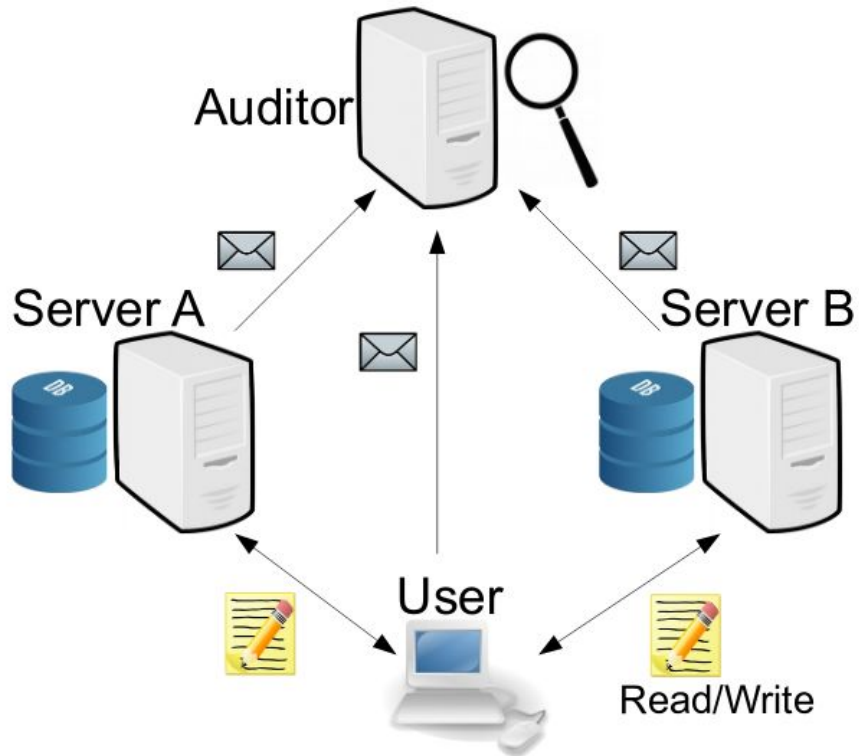
8x improvement in client computation time

>10x improvement in communication costs

# Express Overview

3 server system, secure against:

- Arbitrarily many corrupt users
- Up to one corrupt server



# Express Overview

3 server system, secure against:

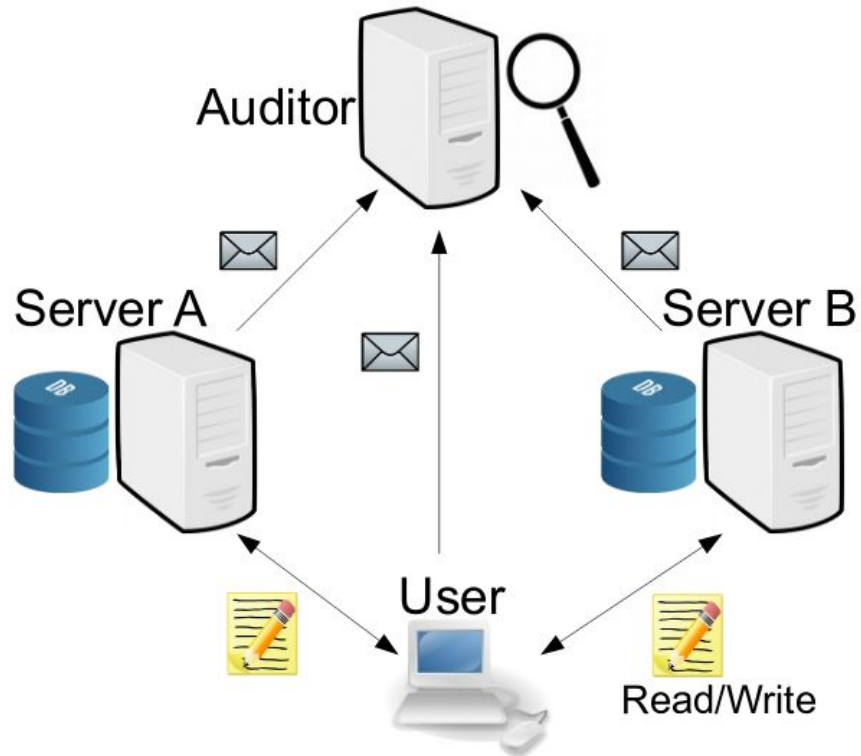
- Arbitrarily many corrupt users
- Up to one corrupt server

Supported operations:

Register mailbox

(Private) write to mailbox

Read from mailbox





# Express Overview

3 server system, secure against:

- Arbitrarily many corrupt users
- Up to one corrupt server

Supported operations:

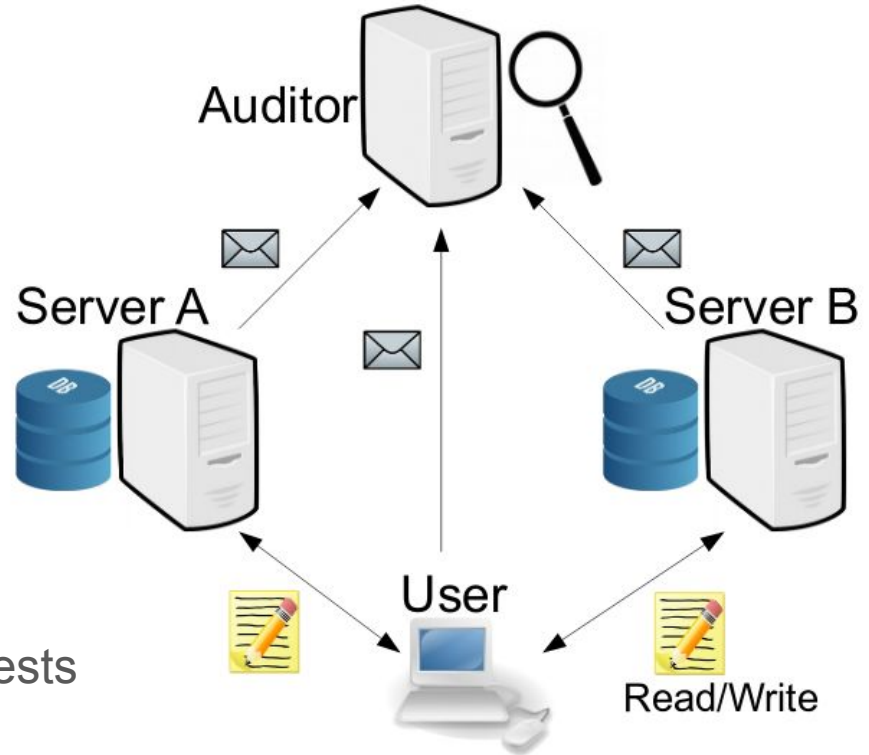
Register mailbox

(Private) write to mailbox

Read from mailbox

Servers A/B store DB, handle requests

Auditor filters malformed/malicious requests



# Express Overview

3 server system, secure against:

- Arbitrarily many corrupt users
- Up to one corrupt server

Supported operations:

Register mailbox

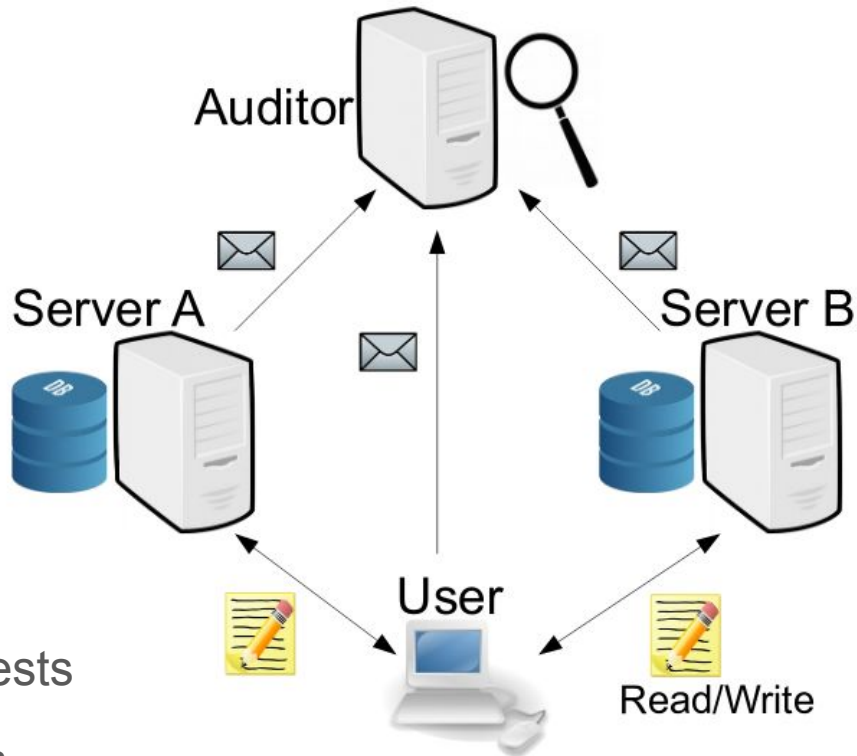
(Private) write to mailbox

Read from mailbox

Servers A/B store DB, handle requests

Auditor filters malformed/malicious requests

**Security:** can't tell who the *recipient* of a message is (unless you are the recipient)



# Outline

Introduction/Overview

Hiding metadata without rounds

Handling disruptive users

Metadata-hiding “web browsing”

Evaluation

# Tool: Private Writing with Distributed Point Functions

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

# 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

# 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

# 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”

$\oplus$

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

Key features:

- concise representation
- fast to generate

# Tool: Private Writing with Distributed Point Functions



I want to write  
"Hi!" to address 3

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



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



# 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

# Tool: Private Writing with Distributed Point Functions



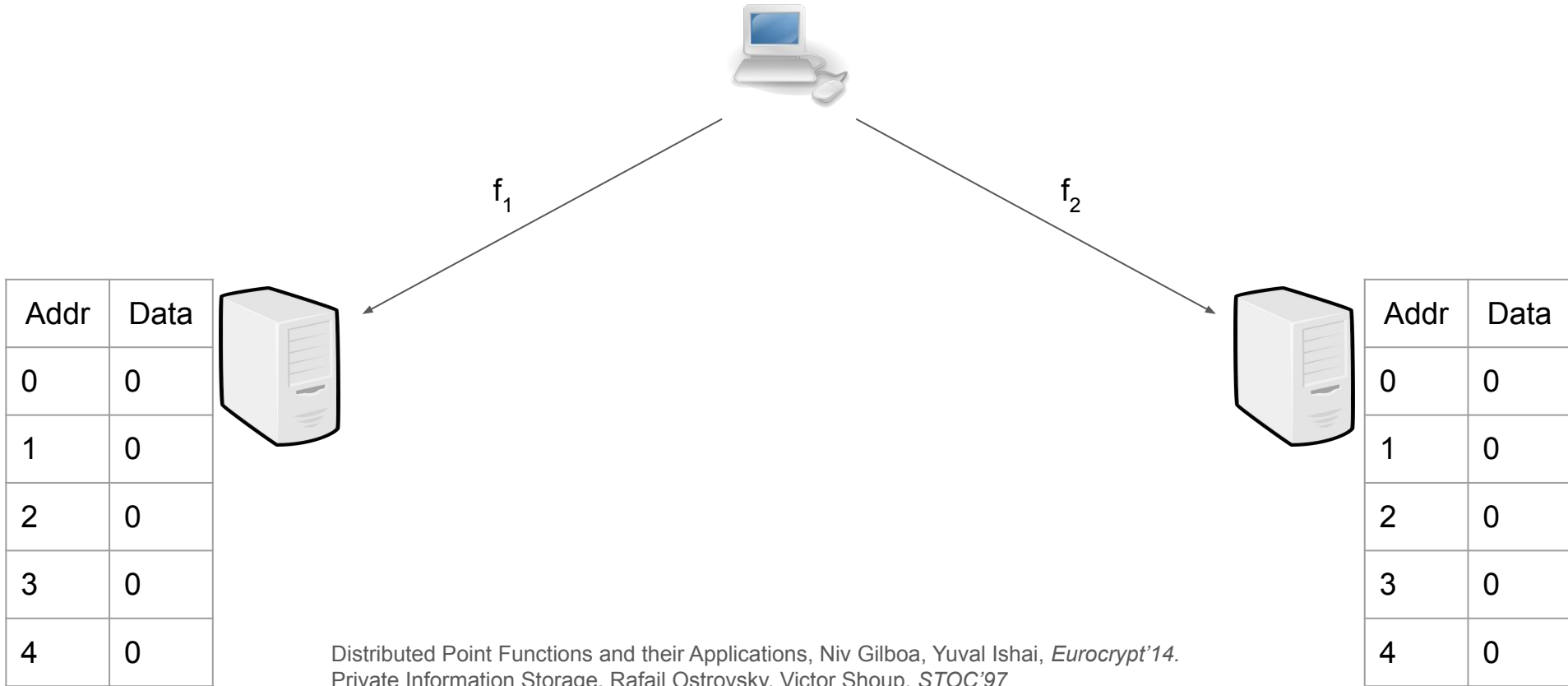
x	$f_1(x)$	x	$f_2(x)$
0	"abc"	0	"abc"
1	"xf\$"	1	"xf\$"
2	"^tg"	2	"^tg"
3	"!7≈"	3	"!2!)"
4	"jhV"	4	"jhV"

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

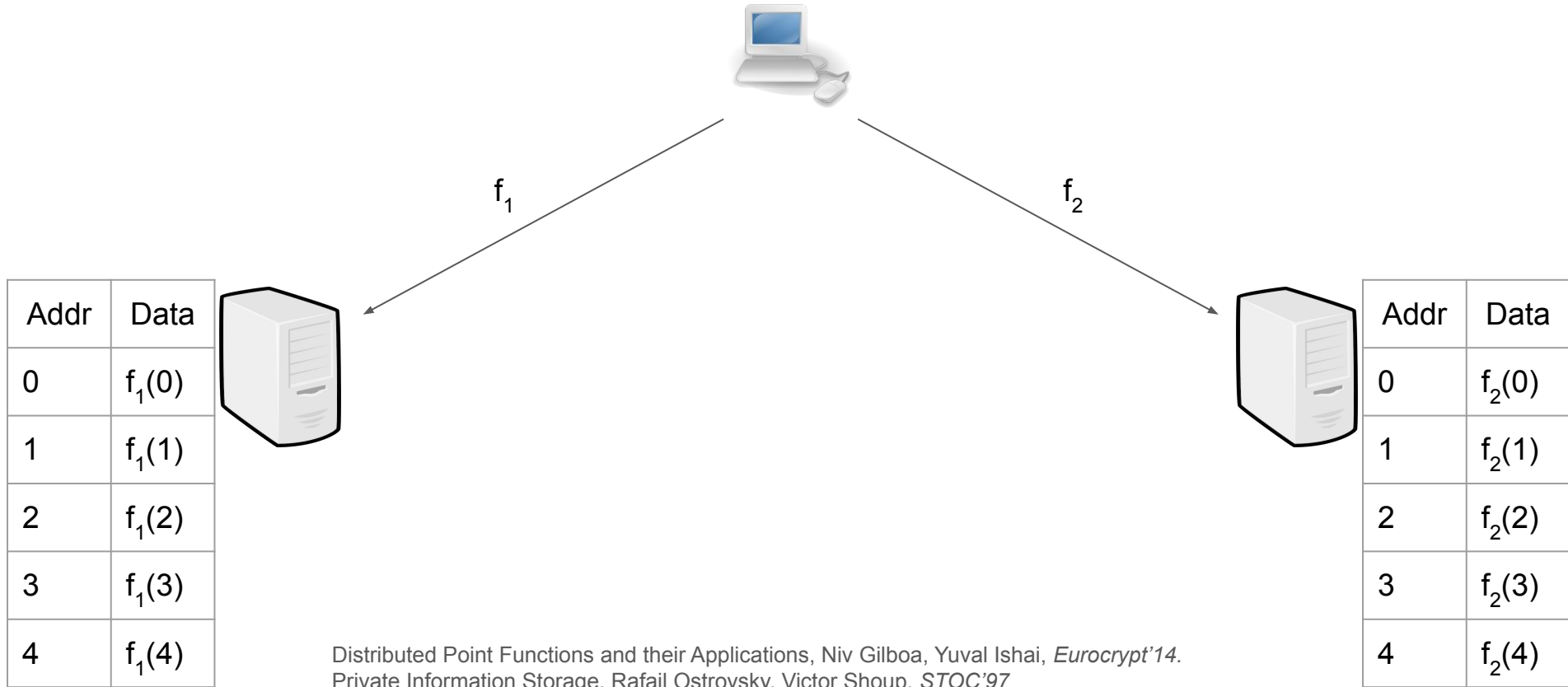


Addr	Data
0	0
1	0
2	0
3	0
4	0

# Tool: Private Writing with Distributed Point Functions



# Tool: Private Writing with Distributed Point Functions



# 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	"!7~"
4	"jhV"

# Tool: Private Writing with Distributed Point Functions



$f_1$

$f_2$



$\oplus$

“Hi!”

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?

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"	$k_{\text{NYT}}$
1	"xf\$"	$k_{\text{WaPo}}$
2	"^tg"	$k_{\text{WSJ}}$
3	"!7≈"	$k_{\text{Buzzfeed}}$
4	"jhV"	$k_{\text{Inquirer}}$



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



# 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

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



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

# Hiding *Metadata*

Construction thus far vulnerable to polling attack:

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

# Hiding *Metadata*

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 *Metadata*

Data Server A



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

$\underbrace{\hspace{1.5cm}}$   $\underbrace{\hspace{1.5cm}}$   $\underbrace{\hspace{3.5cm}}$   
logN bits   128 bits   Data size

# Hiding Metadata

Data Server A



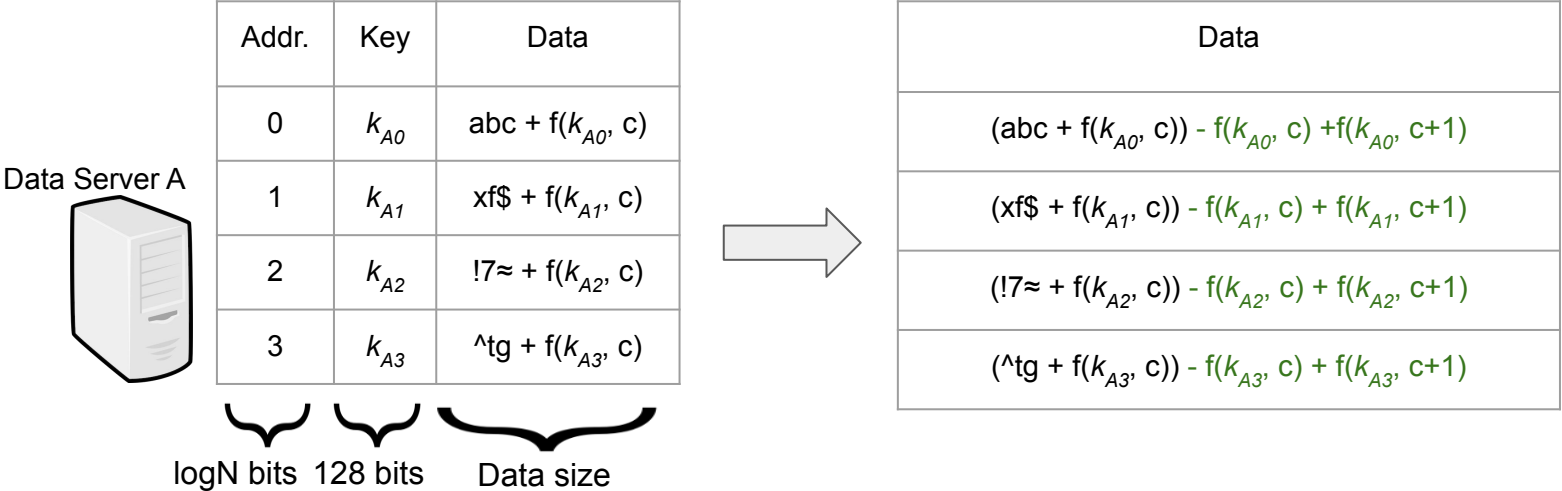
Addr.	Key	Data
0	$k_{A0}$	$abc + f(k_{A0}, c)$
1	$k_{A1}$	$xf\$ + f(k_{A1}, c)$
2	$k_{A2}$	$!7\approx + f(k_{A2}, c)$
3	$k_{A3}$	$\wedge tg + f(k_{A3}, c)$

$\underbrace{\hspace{1.5cm}}$   $\underbrace{\hspace{1.5cm}}$   $\underbrace{\hspace{3.5cm}}$   
 logN bits    128 bits    Data size



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

# Hiding Metadata



Cost to re-randomize a row: (msg length/16) AES blocks

Cost to compute DPF for a row: (256 + msg length/16) AES blocks

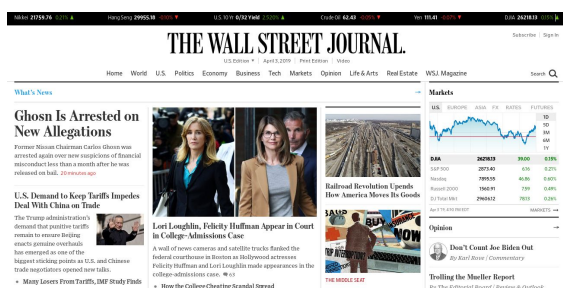
# Plausible Deniability

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

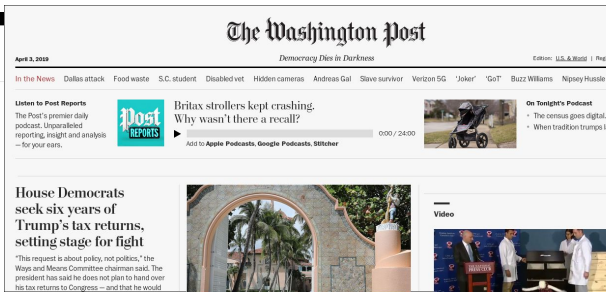
# Plausible Deniability

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

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



The screenshot shows the homepage of The Wall Street Journal. At the top, there is a navigation bar with the site's name and various utility links. The main headline is "Ghosh Is Arrested on New Allegations" with a sub-headline "Former Nuveen Chairman Carlos Ghosn was arrested again over new allegations of financial misconduct from three months after he was released on bail." Below this, there are several other news items, including "U.S. Demand to Keep Tariffs Impedes Deal With China on Trade" and "Lord Loughlin, Felicity Huffman Appear in Court in College Admissions Case". A "Markets" section is visible on the right side of the page.



The screenshot shows the homepage of The Washington Post. The main headline is "House Democrats seek six years of Trump's tax returns, setting stage for fight". Below this, there are several other news items, including "Britain strollers kept crashing. Why wasn't there a recall?" and "On Tonight's Podcast: The census goes digital". A "Post Reports" section is also visible, featuring a "Post Reports" logo and a link to "Add to Apple Podcasts, Google Podcasts, Stitcher".



The screenshot shows the homepage of The New York Times. The main headline is "Some on Mueller's Team Say Report Was More Damaging Than Barr Revealed". Below this, there are several other news items, including "Listen: 'Modern Love' Podcast" and "The 'NYT Parenting' Newsletter". A prominent feature is a diagram titled "Here are the criminal inquiries that sprouted from the special counsel's investigation." which shows a central figure (Donald Trump) connected to three boxes: "Russia", "Position", and "Foreign Influence".



# Plausible Deniability

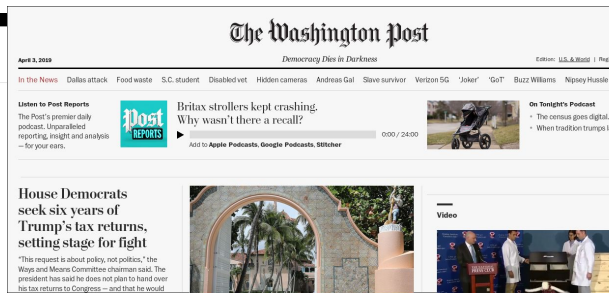
How to protect privacy of whistleblowers if *all users* are whistleblowers?

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



The Wall Street Journal homepage features a top navigation bar with market data (Dow Jones, S&P 500, Nasdaq, etc.) and a main headline: "Ghosh is Arrested on New Allegations". Below the headline is a sub-headline: "Former Monoc Chairman Carvin Ghosh was arrested again over new allegations of financial misconduct less than a month after he was released on bail." Other visible headlines include "U.S. Demand to Keep Tariffs Impedes Deal With China on Trade" and "Lord Loughlin, Felicity Huffman Appear in Court in College Admissions Case".



The Washington Post homepage features a top navigation bar with the site name and a main headline: "Democracy Dies in Darkness". Below the headline is a sub-headline: "Britain strollers kept crashing. Why wasn't there a recall?". Other visible headlines include "House Democrats seek six years of Trump's tax returns, setting stage for fight" and "Listen to Post Reports: The Post's premier daily podcast. Unparalleled reporting, insight and analysis -- for your ears."



The New York Times homepage features a top navigation bar with the site name and a main headline: "Some on Mueller's Team Say Report Was More Damaging Than Barr Revealed". Below the headline is a sub-headline: "A number of former Mueller's team have said their findings are more troubling than President Trump's attorney General William Barr had indicated." Other visible headlines include "The 'NYT Parenting' Newsletter" and "The Incredible Shrinking Apple". A diagram titled "Here are the criminal inquiries that sprouted from the special counsel's investigations." shows a flow from "Trump (criminal)" to "Position (political)" and "Foreign (foreign)".

# 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

Any number of users can act maliciously in arbitrary ways

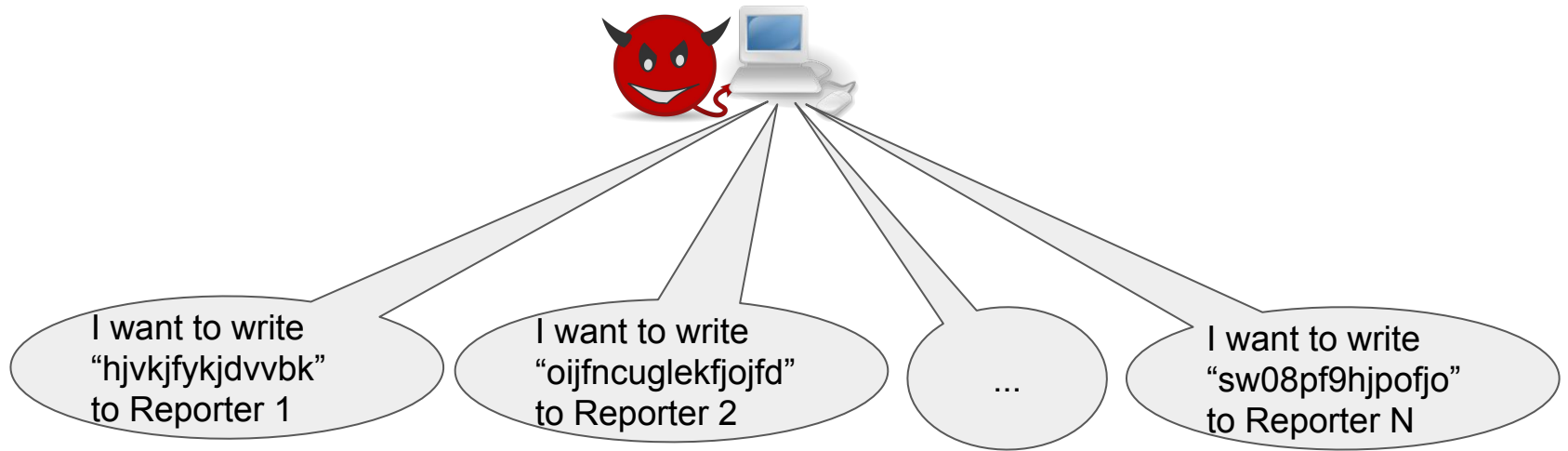
Two kinds of attacks:

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

Mechanism for preventing disruption can't compromise privacy

# 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"

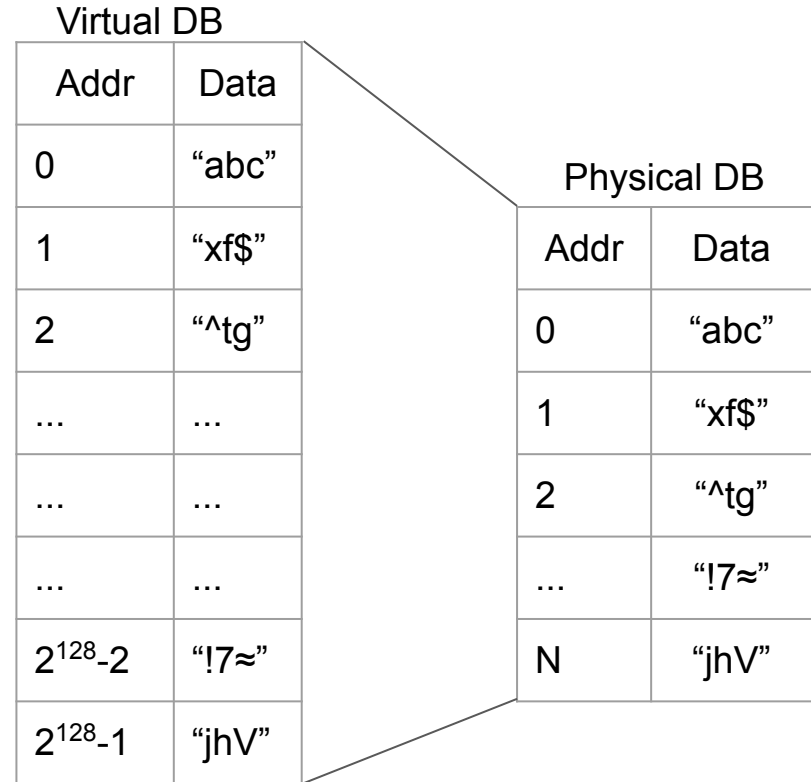
# Virtual Addresses

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

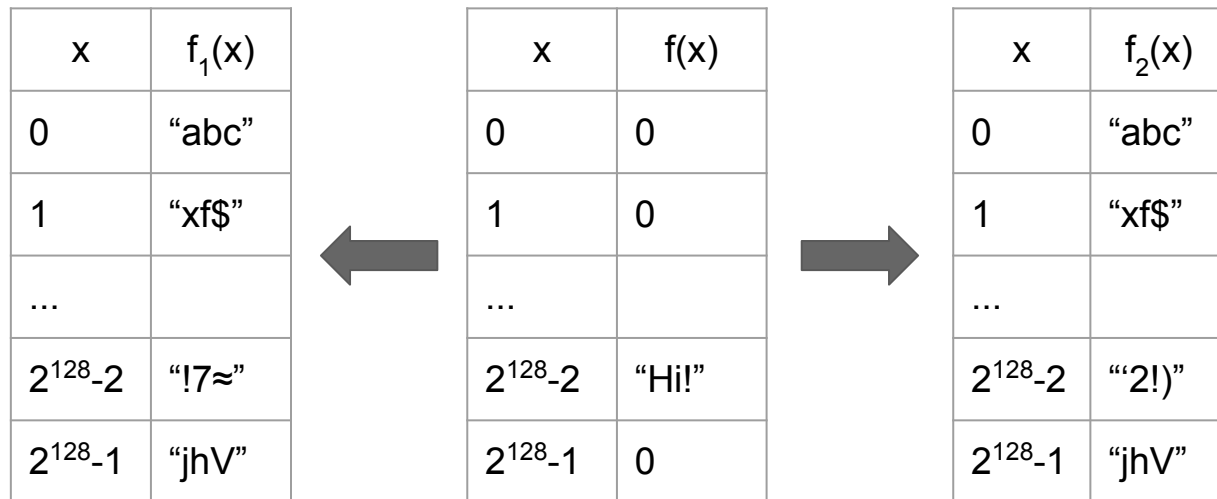
Problem: disruptive user sends malformed DPF to write to many mailboxes



x	f(x)
0	0
1	0
...	
$2^{128}-2$	"Hi!"
$2^{128}-1$	0

# Auditing

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



# Auditing

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

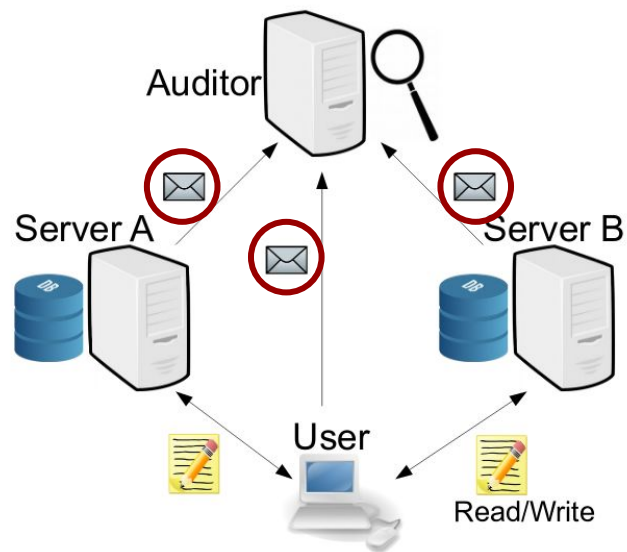


x	f(x)
0	989f4
1	dDf73
...	
$2^{128}-2$	08dji3
$2^{128}-1$	89hfif

# Auditing

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

Solution: third server *audits* all incoming write requests



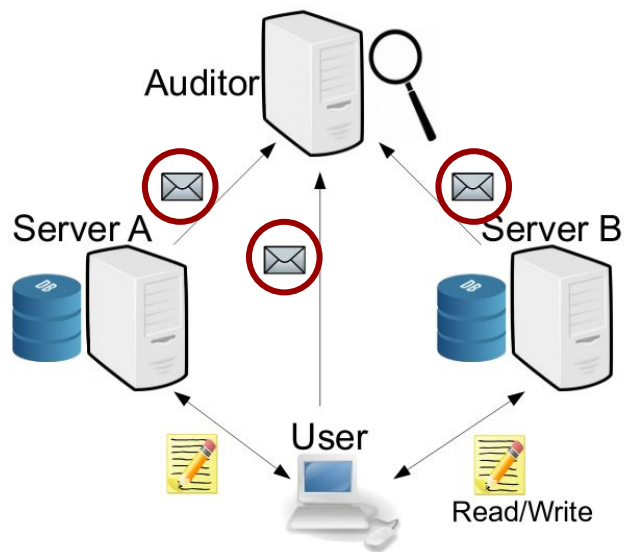
# Auditing

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

Solution: third server *audits* all incoming write requests

New auditing protocol:

- $O(\log N)$  communication
- $O(\log N)$  client/auditor computation
- Prior work: all  $O(\sqrt{N})$



# Auditing

Our problem: proving DPF write only modifies one entry in DB

x	$f_1(x)$	x	$f_2(x)$
0	“abc”	0	“abc”
1	“xf\$”	1	“xf\$”
2	“^tg”	2	“^tg”
3	“!7≈”	3	““2!””
4	“jhV”	4	“jhV”

# Auditing

Our problem: proving DPF write only modifies one entry in DB

More general problem: proving two vectors differ at one point



# Auditing

Our problem: proving DPF write only modifies one entry in DB

More general problem: proving two vectors differ at one point



$\oplus$



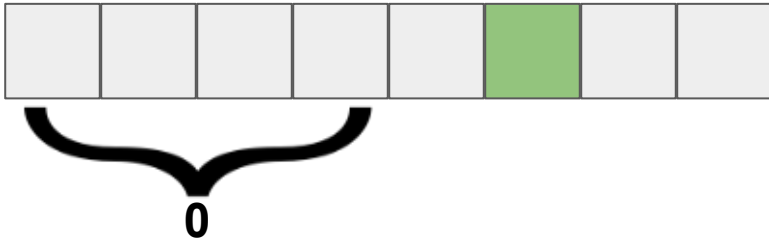
=





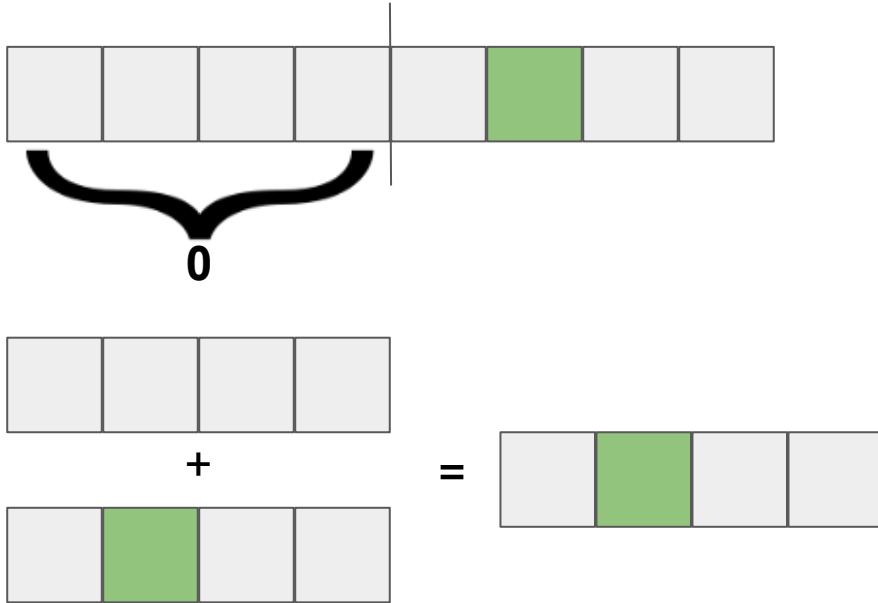
# Auditing

Idea: Recursively prove that one half is zero



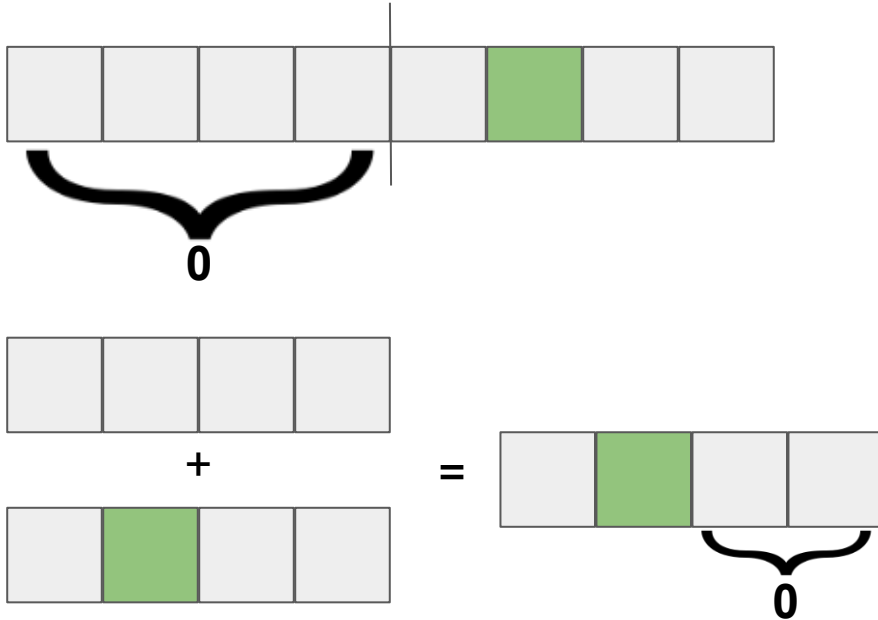
# Auditing

Idea: Recursively prove that one half is zero



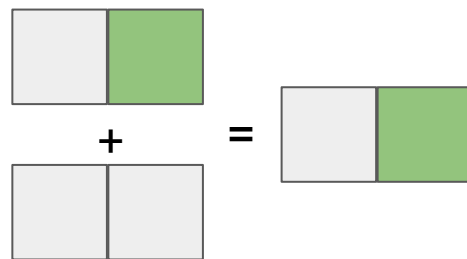
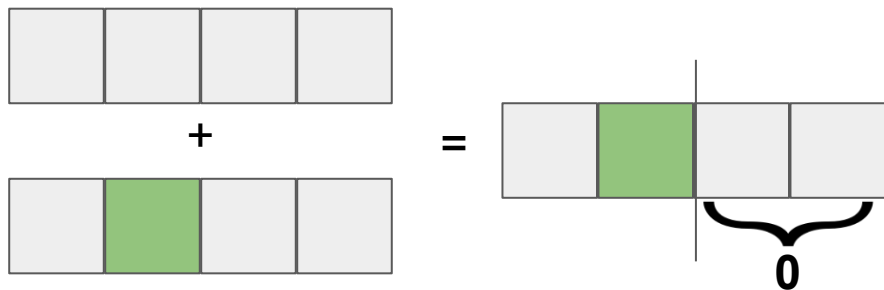
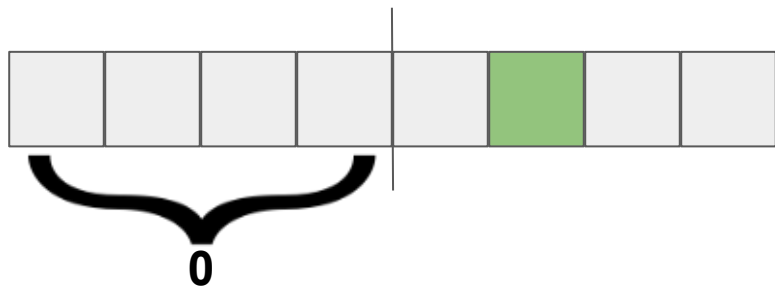
# Auditing

Idea: Recursively prove that one half is zero



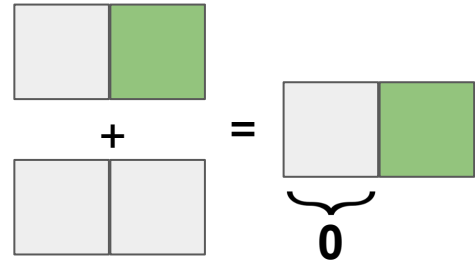
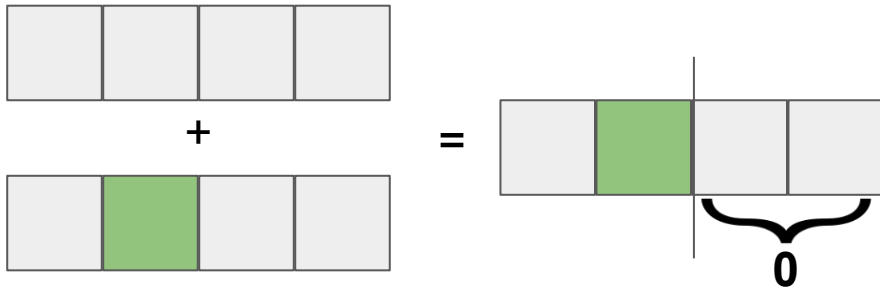
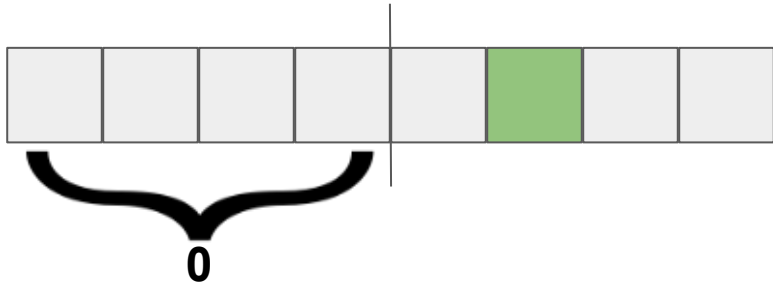
# Auditing

Idea: Recursively prove that one half is zero



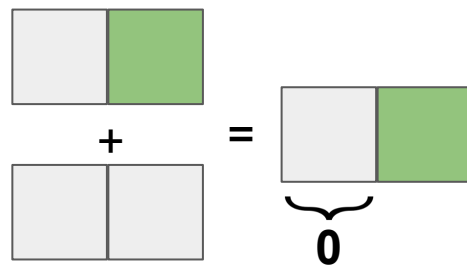
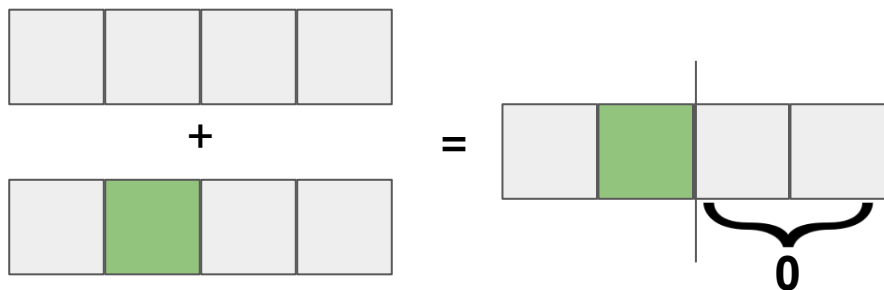
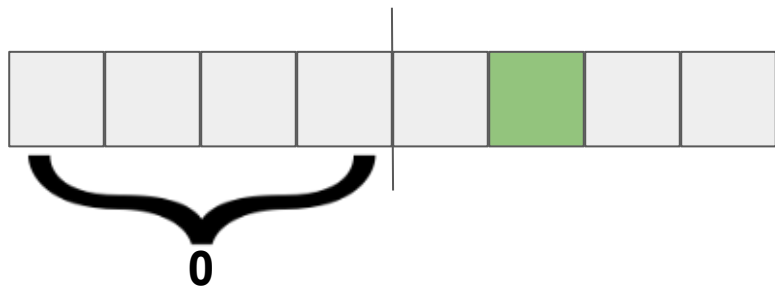
# Auditing

Idea: Recursively prove that one half is zero



# Auditing

Idea: Recursively prove that one half is zero



Claim: If there is more than one nonzero entry, the proof will fail on at least one level of recursion

# Auditing

Claim: If there is more than one nonzero entry, the proof will fail on at least one level of recursion

Proof:

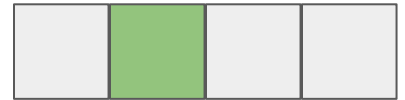
- 1.
- 2.
- 3.

# Auditing

Claim: If there is more than one nonzero entry, the proof will fail on at least one level of recursion

Proof:

1. Consider the first recursive step where there is only one nonzero entry
- 2.
- 3.



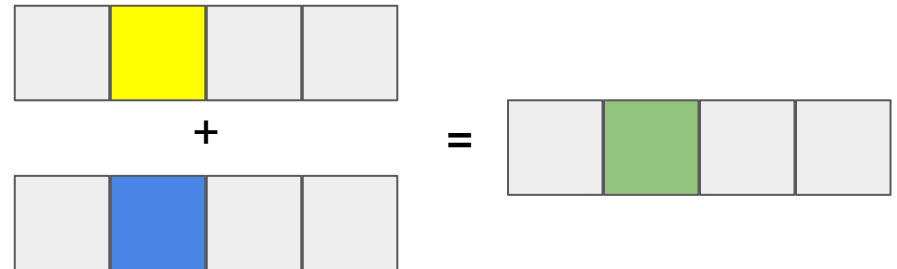


# Auditing

Claim: If there is more than one nonzero entry, the proof will fail on at least one level of recursion

Proof:

1. Consider the first recursive step where there is only one nonzero entry
2. The preceding step must have had two nonzero entries on opposite sides
- 3.

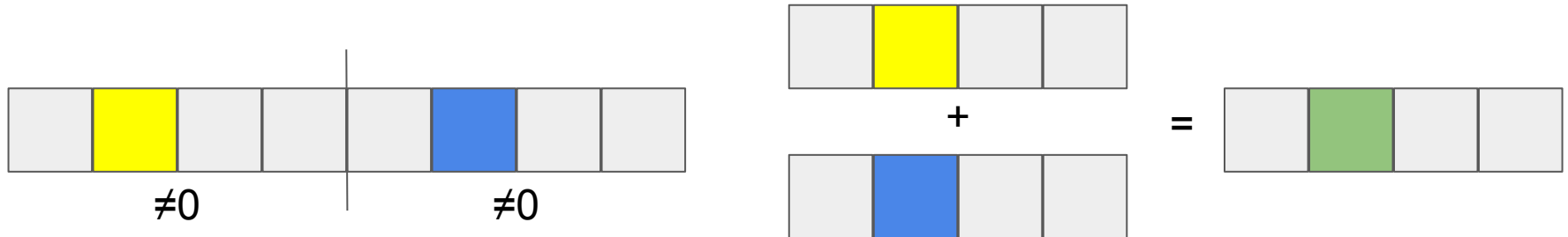


# Auditing

Claim: If there is more than one nonzero entry, the proof will fail on at least one level of recursion

Proof:

1. Consider the first recursive step where there is only one nonzero entry
2. The preceding step must have had two nonzero entries on opposite sides
3. Proof must then fail because neither half is zero



# Auditing

How to prove a vector is all zeros?

# Auditing

How to prove a vector is all zeros?

Interpret each DPF output as an element in a prime-order field

Multiply each element by a random value and sum

# Auditing

How to prove a vector is all zeros?

Interpret each DPF output as an element in a prime-order field

Multiply each element by a random value and sum

Servers do this separately on their shares of the vector and send to auditor

# Auditing

How to prove a vector is all zeros?

Interpret each DPF output as an element in a prime-order field

Multiply each element by a random value and sum

Servers do this separately on their shares of the vector and send to auditor

Server doesn't know which half is zero, sends sum for each half (in random order)

# Auditing

How to prove a vector is all zeros?

Interpret each DPF output as an element in a prime-order field

Multiply each element by a random value and sum

Servers do this separately on their shares of the vector and send to auditor

Server doesn't know which half is zero, sends sum for each half (in random order)

Auditor accepts if one pair of sums are equal

# Auditing with Malicious Servers

A malicious data server can violate privacy in the protocol so far, e.g.:

Corrupt content of one half; If auditor still accepts, that half was non-zero



# Auditing with Malicious Servers

A malicious data server can violate privacy in the protocol so far, e.g.:

Corrupt content of one half; If auditor still accepts, that half was non-zero

Mitigation: client helps police data servers

# Auditing with Malicious Servers

A malicious data server can violate privacy in the protocol so far, e.g.:

Corrupt content of one half; If auditor still accepts, that half was non-zero

Mitigation: client helps police data servers

Client gets random seed from data servers

Client tells auditor which pair should sum to zero

Client tells auditor what the non-zero sum should be

# Another Application: Web Browsing

Goal: browse the web without ISP or surveillance learning what sites you access

# Another Application: Web Browsing

Goal: browse the web without ISP or surveillance learning what sites you access

Non-goals:

Hide your identity from the sites you visit  
(not an anonymity system)

Backwards compatibility  
(sites run custom protocol to deliver pages)

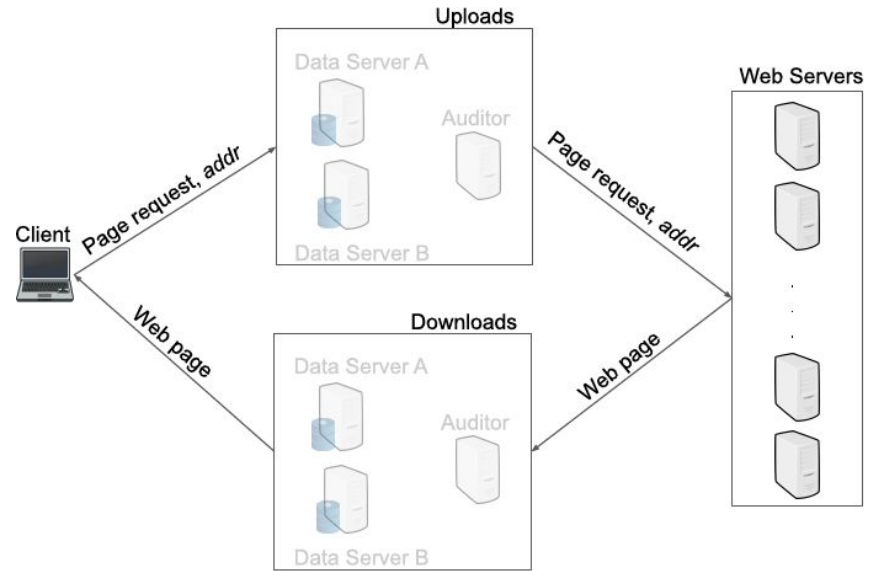
# Another Application: Web Browsing

Goal: browse the web without ISP or surveillance learning what sites you access

Non-goals:

Hide your identity from the sites you visit  
(not an anonymity system)

Backwards compatibility  
(sites run custom protocol to deliver pages)



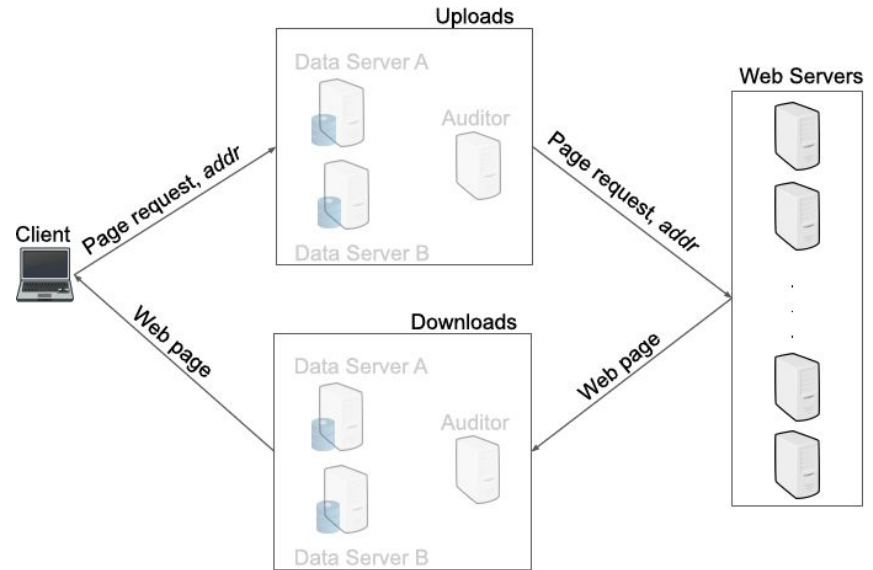
Idea: Use 2 instance of Express in parallel to upload requests and download pages

# Web Browsing with Express

## Express instance 1: Uploads

Web sites have public addresses to receive page requests

## Express instance 2: Downloads



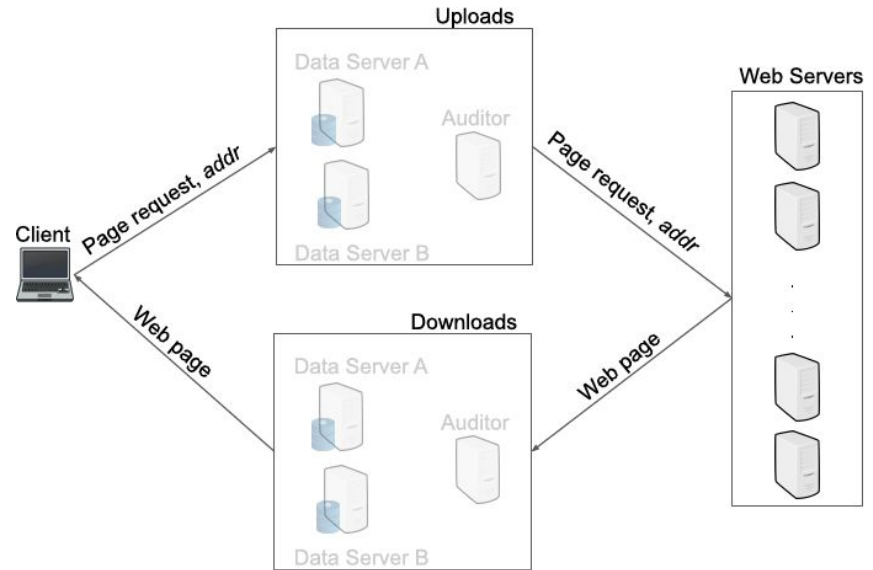
# Web Browsing with Express

## Express instance 1: Uploads

Web sites have public addresses to receive page requests

## Express instance 2: Downloads

Clients register *short-lived* addresses to receive pages, include their short-lived address in page request to instance 1



# Web Browsing with Express

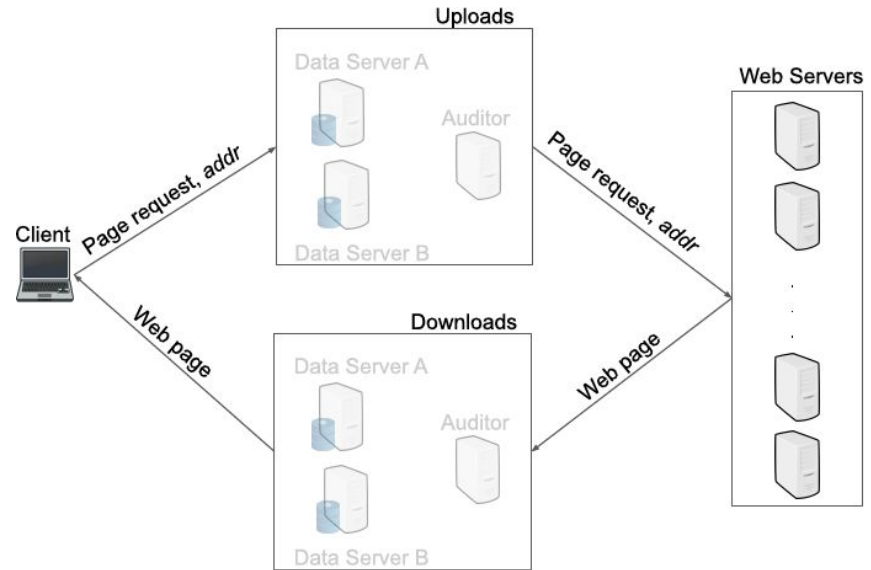
## Express instance 1: Uploads

Web sites have public addresses to receive page requests

## Express instance 2: Downloads

Clients register *short-lived* addresses to receive pages, include their short-lived address in page request to instance 1

Web servers need to contact Express at regular intervals, but clients do not

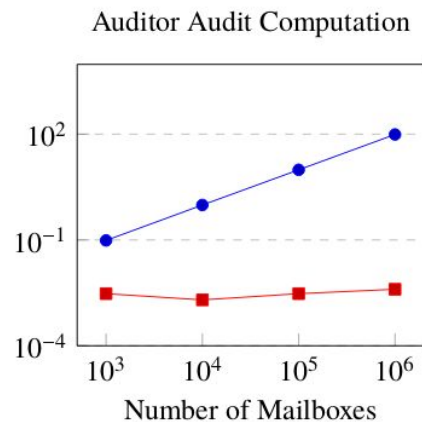
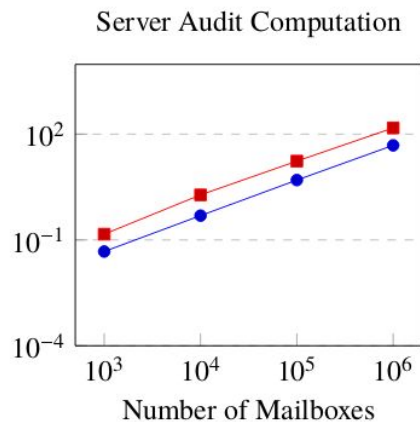
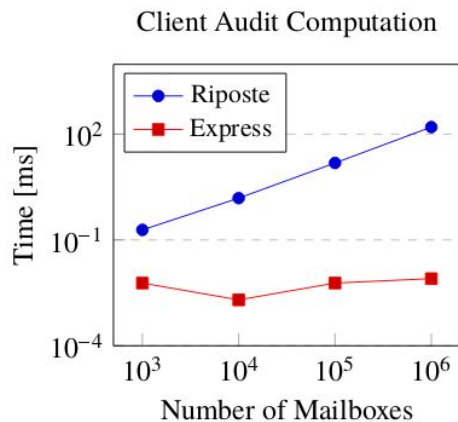




# Evaluation

# Evaluation

## Auditing Microbenchmarks



Under 10 *microseconds* for 1m mailboxes (compare to 159, 98 microseconds)

Enables 8x improvement in client computation time

# Evaluation

## Client Costs

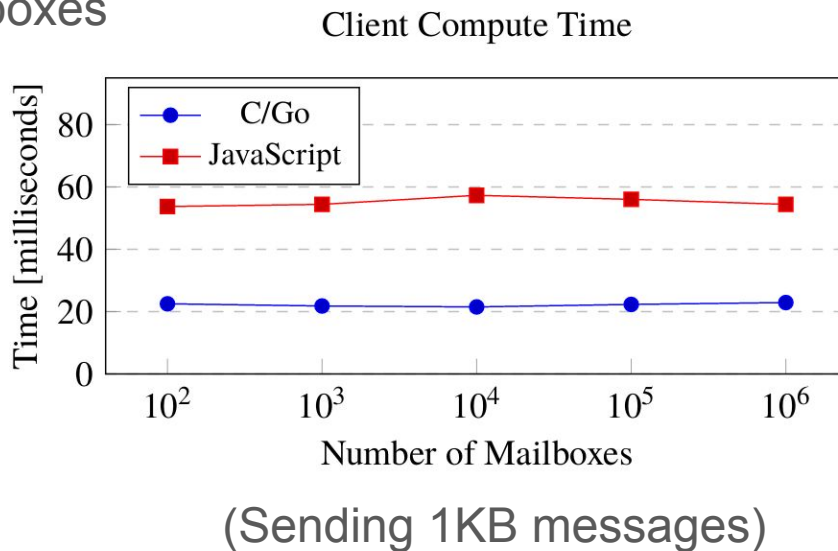
Asymptotically  $O(\log N)$  in number of mailboxes

In practice, almost independent

Less than 1ms increase from 100 to 1m

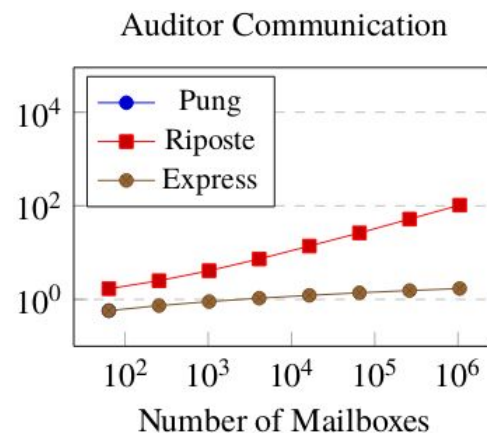
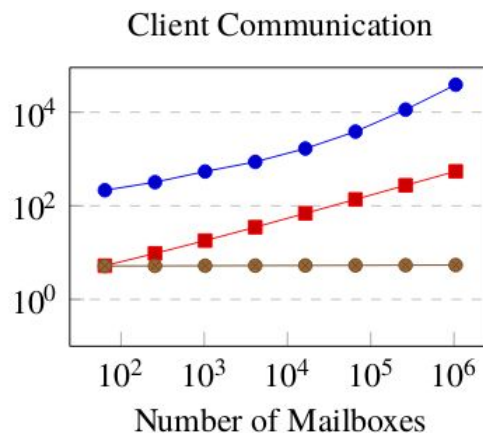
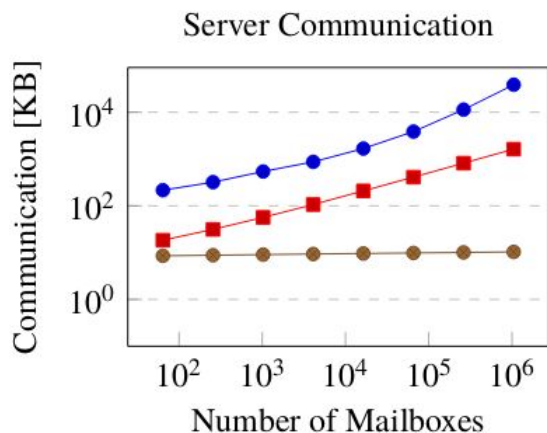
JS code size: 71KB

Less than 2% of major news sites' sizes



# Evaluation

## Communication Costs



(Sending 160B messages)

For  $2^{14}$  mailboxes: 10x improvement

For  $2^{20}$  mailboxes: 100x improvement (client/server), 50x improvement (auditor)

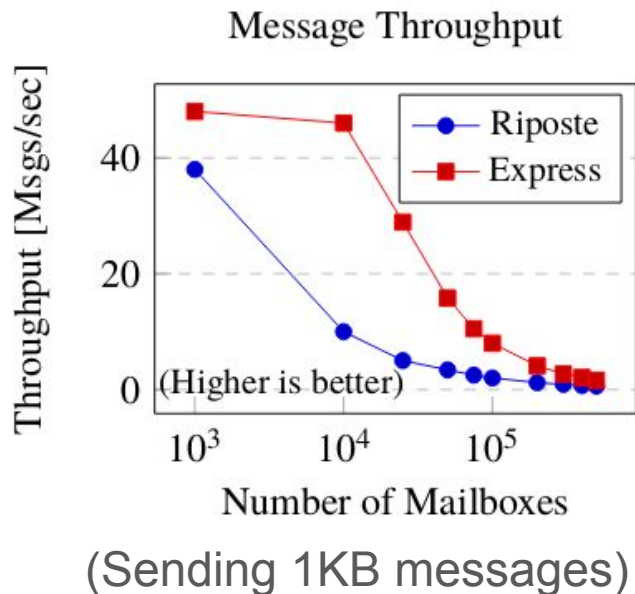
# Evaluation

## Comparison to Riposte

Riposte supports anonymous broadcast,  
Express supports broadcast and private  
messages

1.3-5.8x throughput improvement

Performance becomes similar as both  
systems become compute-bound on  
server side



# Express

First metadata-hiding communication system with no synchronization requirement

Asymptotic speedup from  $O(\sqrt{N})$  to  $O(\log N)$

Practical speedup up to 5x on server, 8x on client

10x or more reduction in communication costs

Applications to private whistleblowing and metadata-hiding web browsing

Contact: [saba@cs.stanford.edu](mailto:saba@cs.stanford.edu)