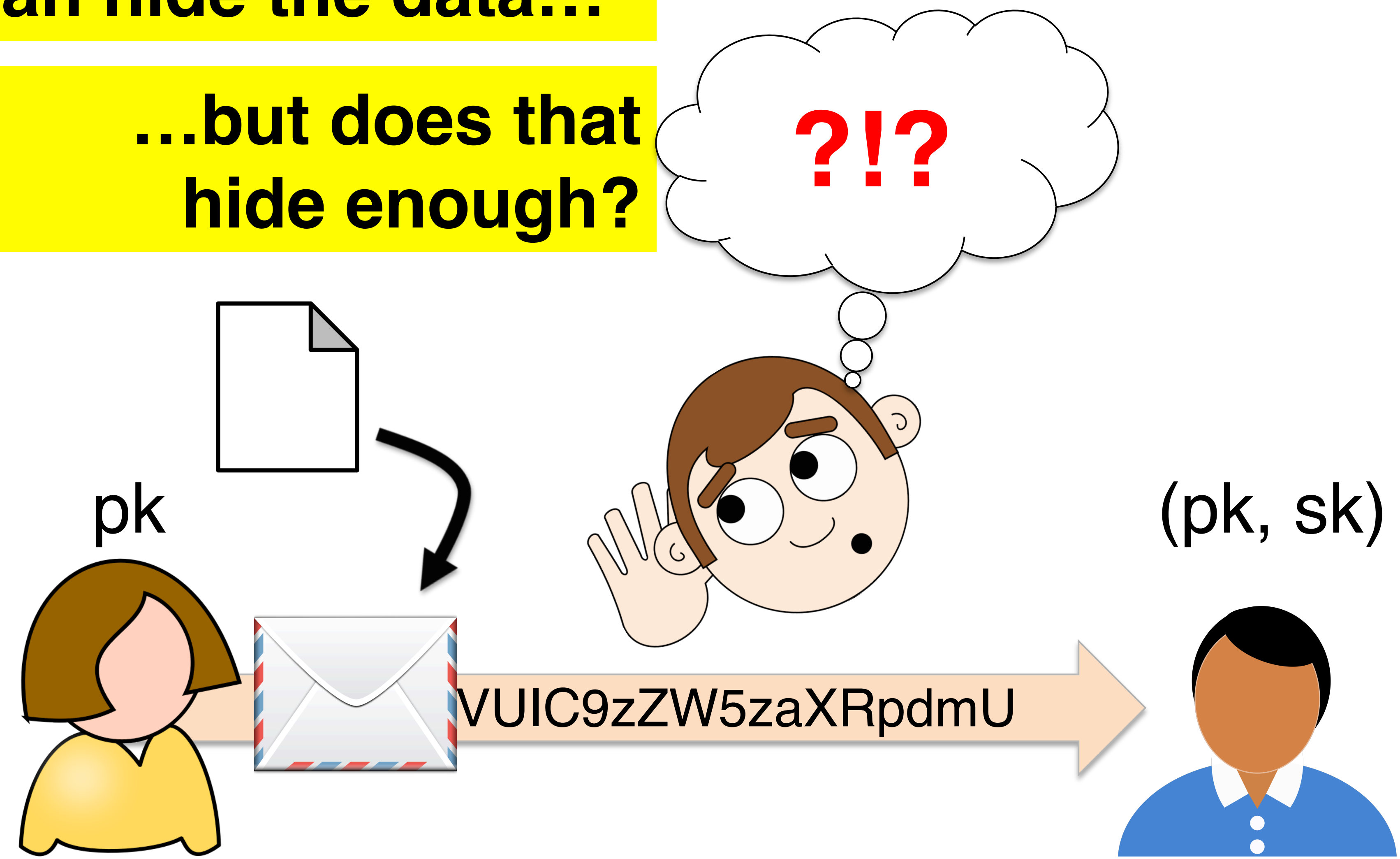
Riposte: An Anonymous Messaging System Handling Millions of Users

Henry Corrigan-Gibbs,
Dan Boneh, and David Mazières
Stanford University

IEEE Security and Privacy
18 May 2015

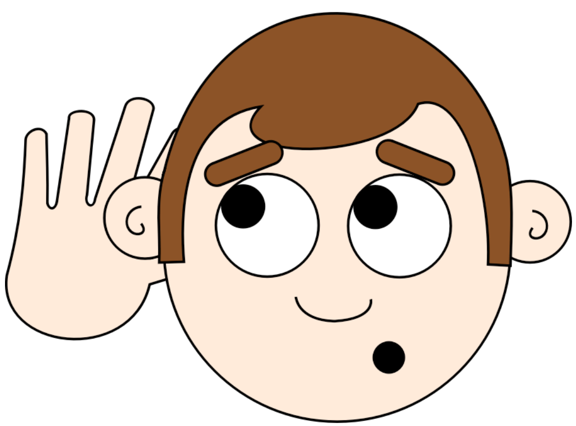
With encryption, we can hide the data…

…but does that hide enough?



Time	From	To	Size
10:12	Alice	Bob	2543 B
10:27	Carol	Alice	567 B
10:32	Alice	Bob	450 B
10:35	Bob	Alice	9382 B

⋮

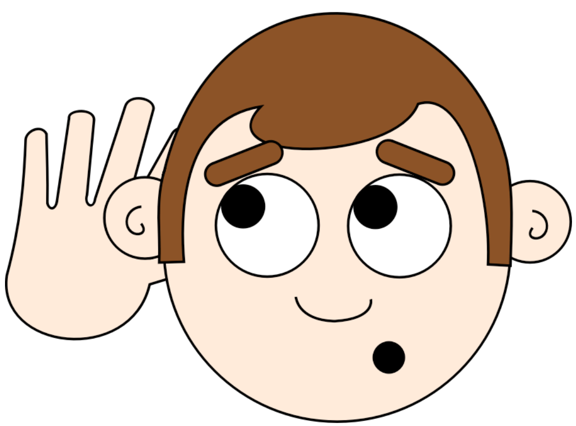


[cf. Ed Felten's testimony before the House Judiciary Committee, 2 Oct 2013]

Time	From	To	Size
10:12	Alice	**taxfraud@stanford.edu**	2543 B
10:27	Carol	Alice	567 B
10:32	Alice	Bob	450 B
10:35	Bob	Alice	B

⋮

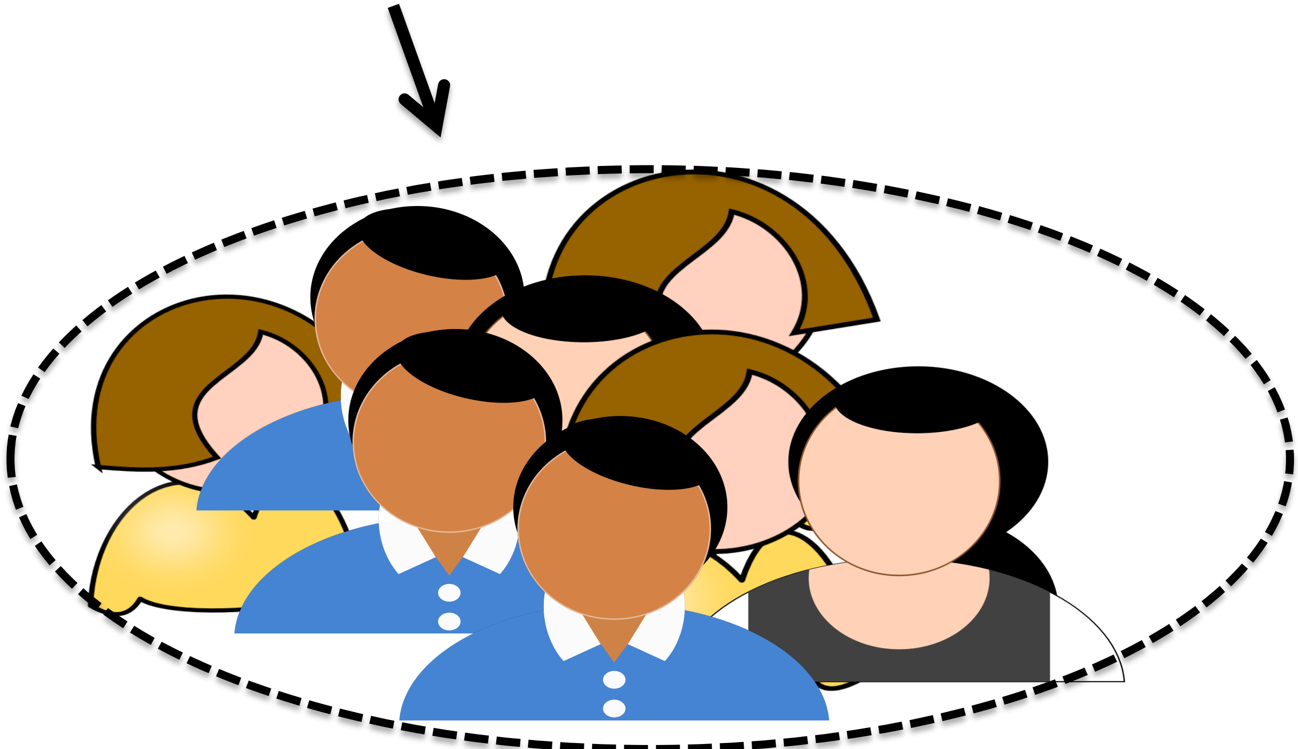
Hiding the data is necessary, but not sufficient



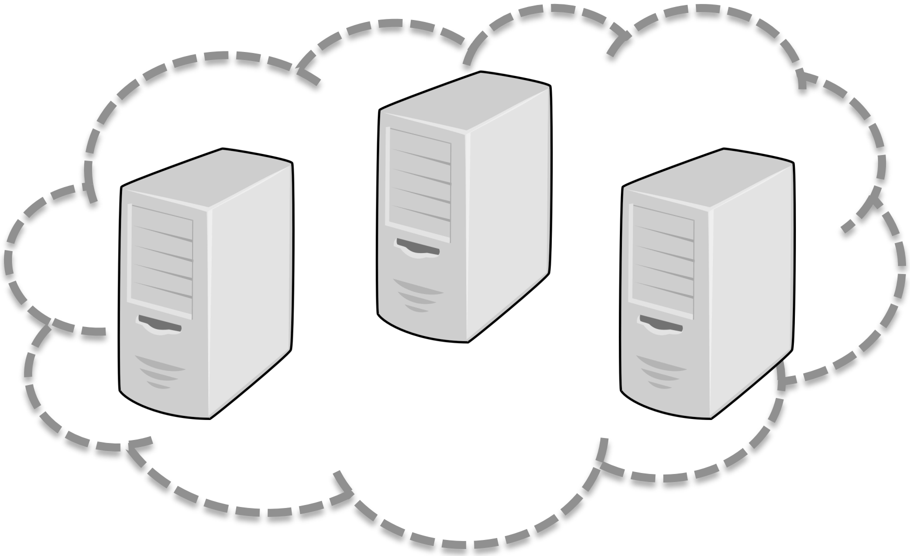
[cf. Ed Felten's testimony before the House Judiciary Committee, 2 Oct 2013]

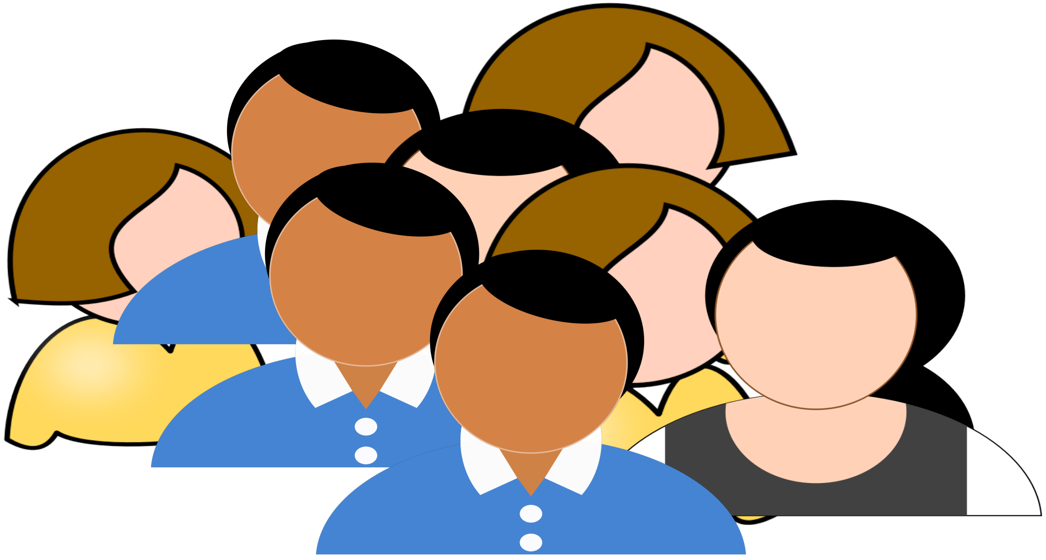
Goal

The “Anonymity Set”

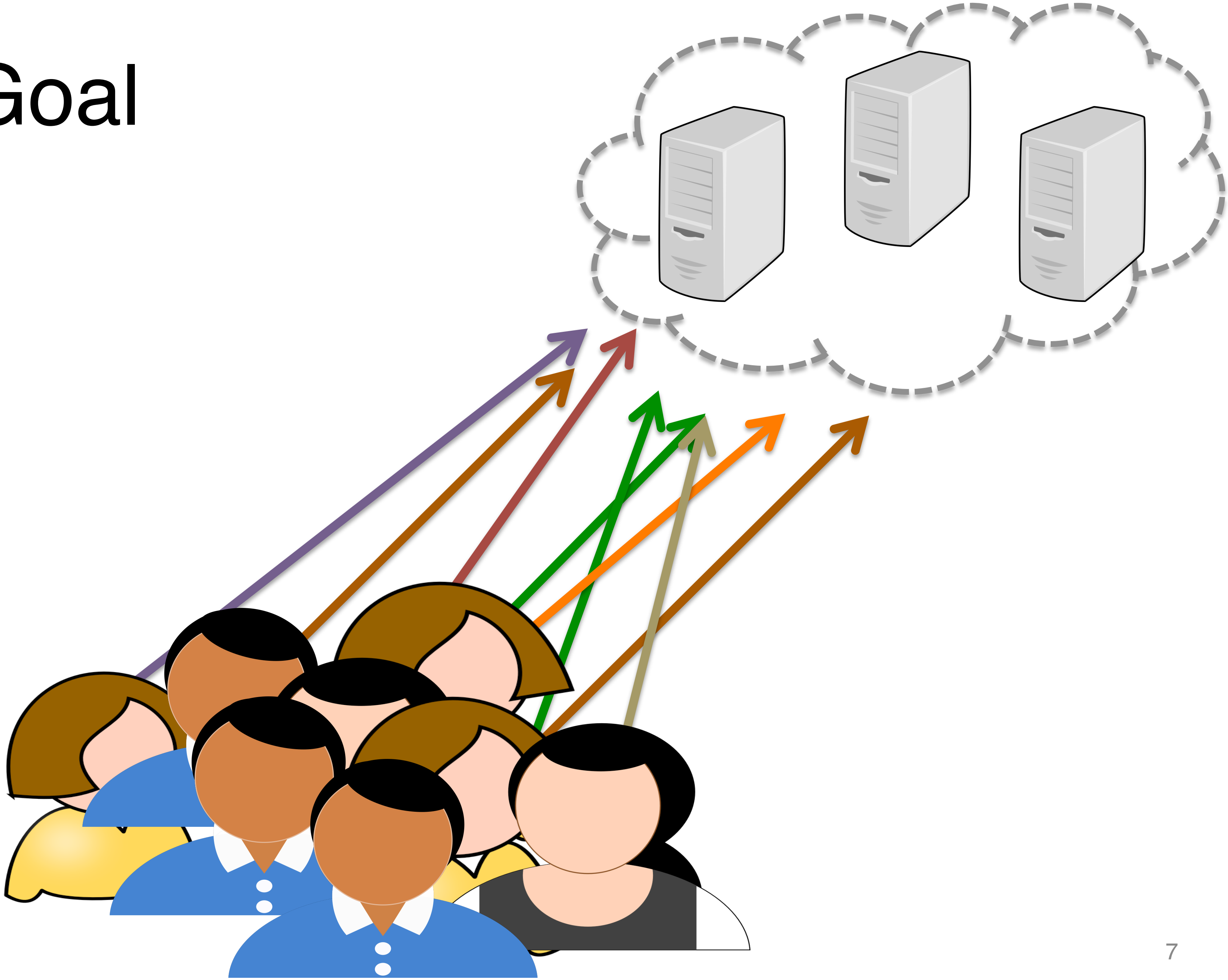


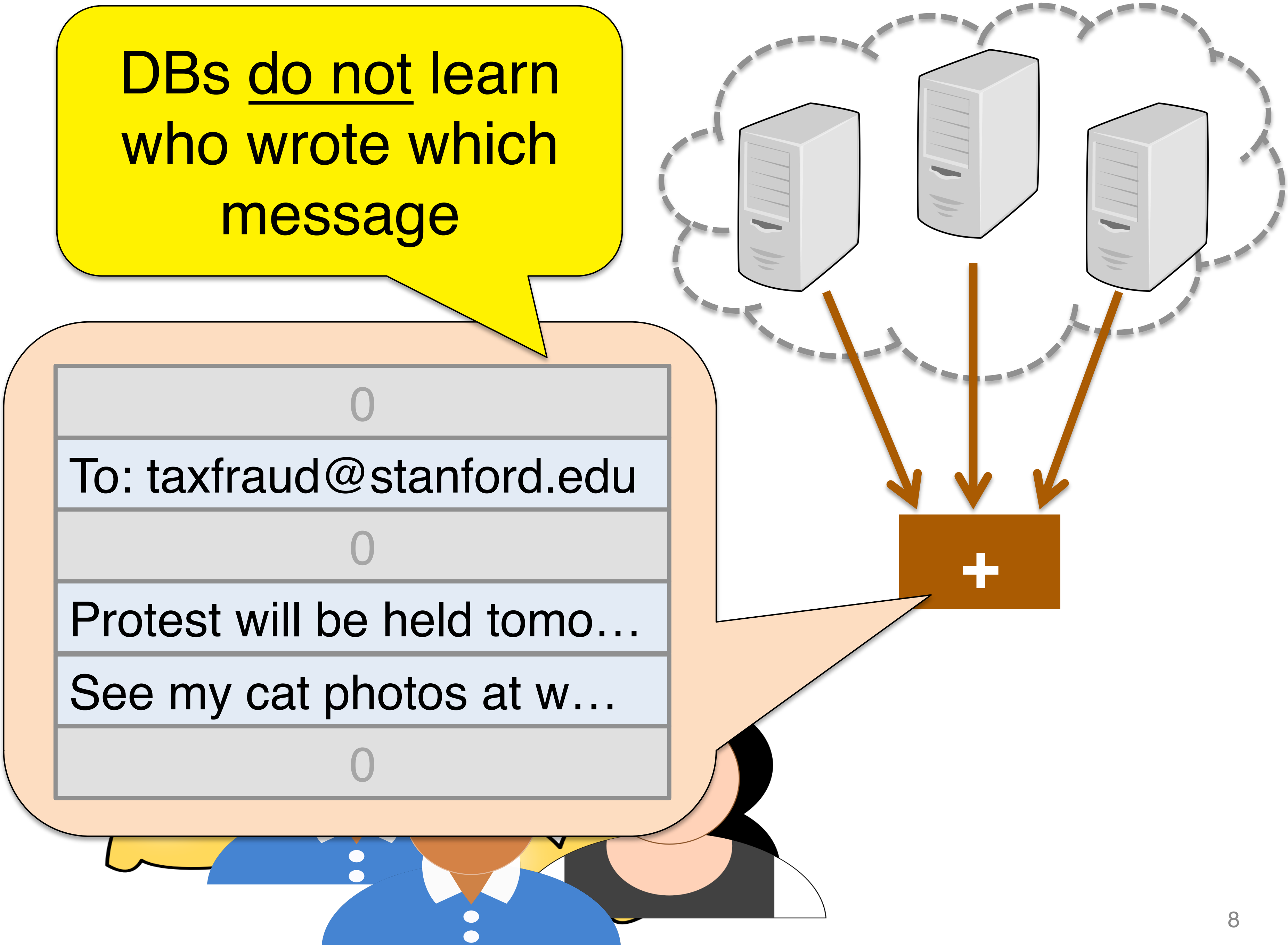
Goal





Goal



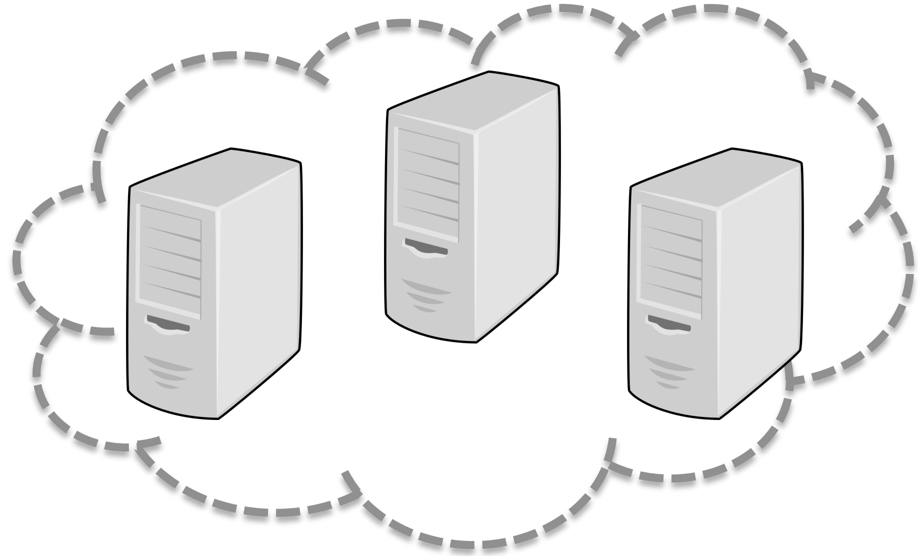

DBs do not learn who wrote which message
0
To: taxfraud@stanford.edu
0
Protest will be held tomo…
See my cat photos at w…
0
+

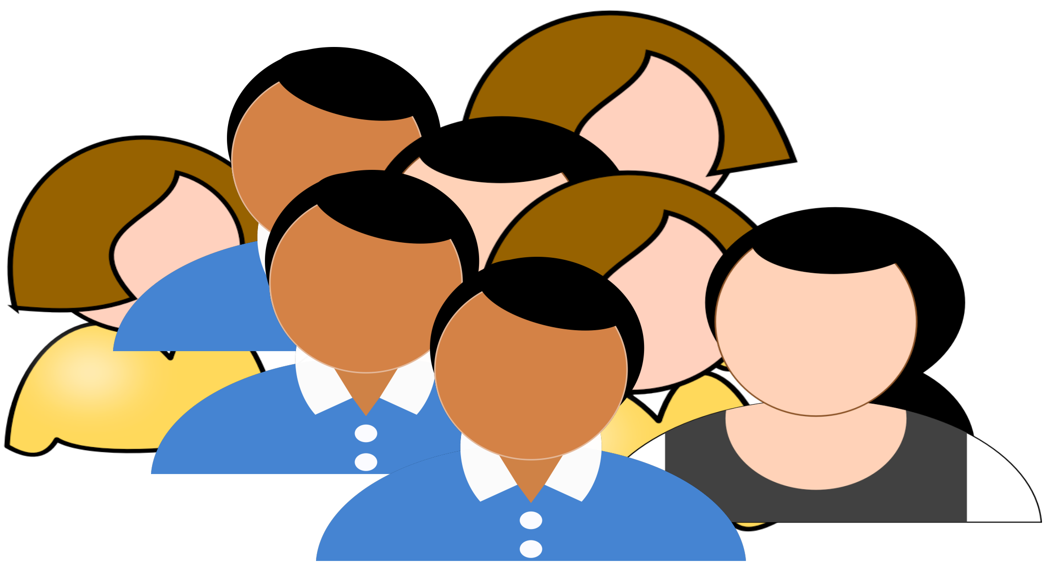
Building block for systems related to "hiding the metadata"

→ Anonymous Twitter

→ Anonymous surveys

→ Private messaging, etc.





Low-latency anonymity systems (e.g., Tor)

… **do not protect** against a global adversary

Mix-nets

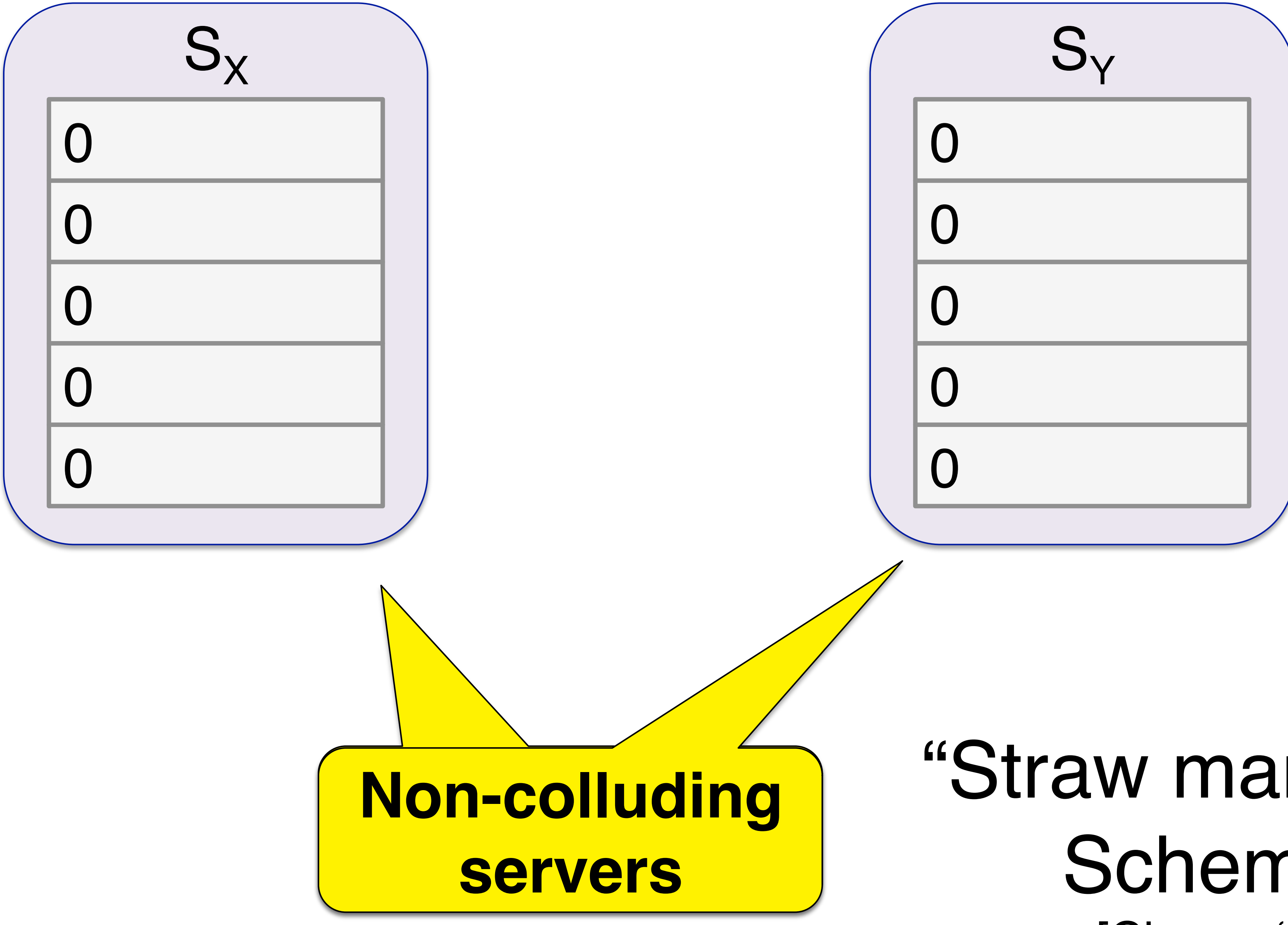
… require expensive ZKPs to protect against active attacks

Riposte is an anonymous messaging system that:

- protects against a near-global active adversary
- handles <u>millions</u> of users in an "anonymous Twitter" system

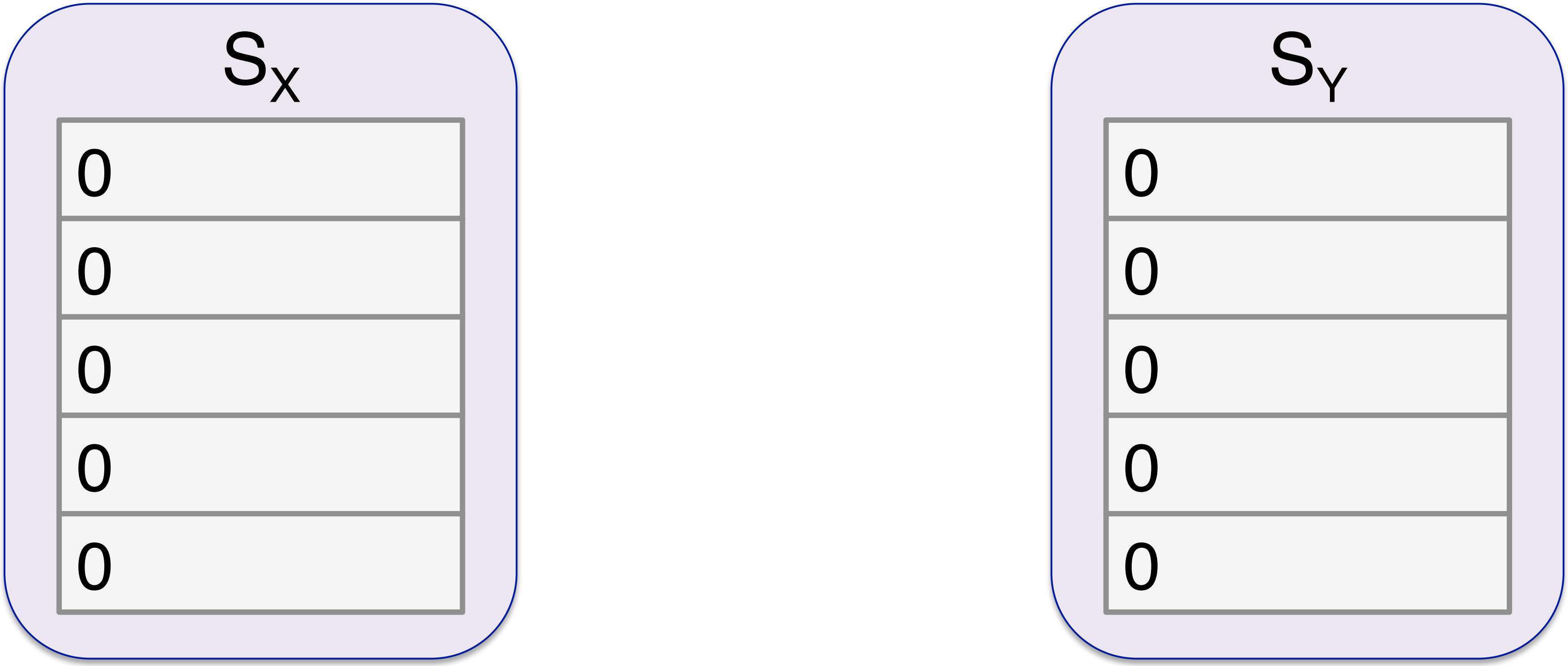
Outline

- Motivation
- **A “Straw man” scheme**
- Technical challenges
- Evaluation

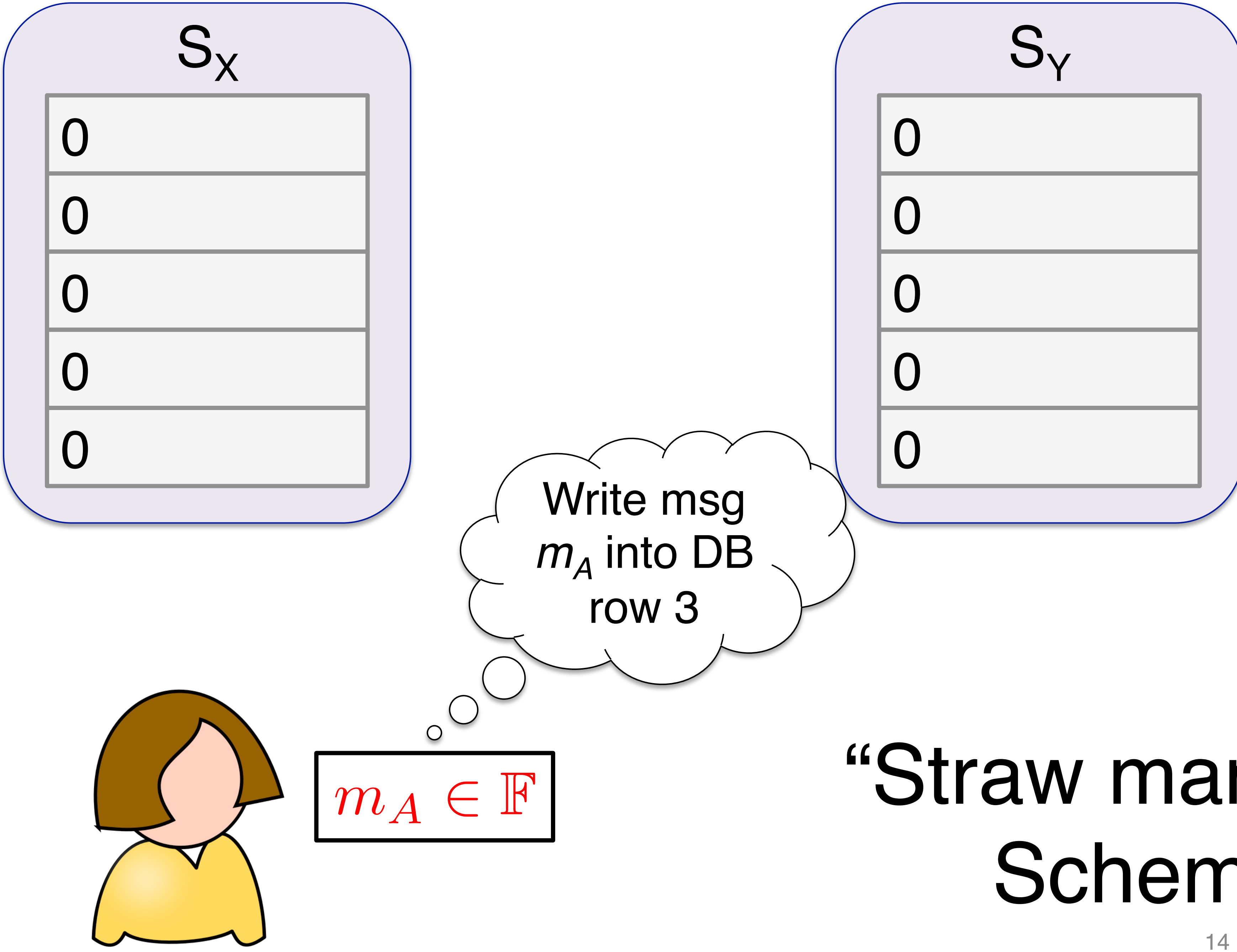


“Straw man” Scheme

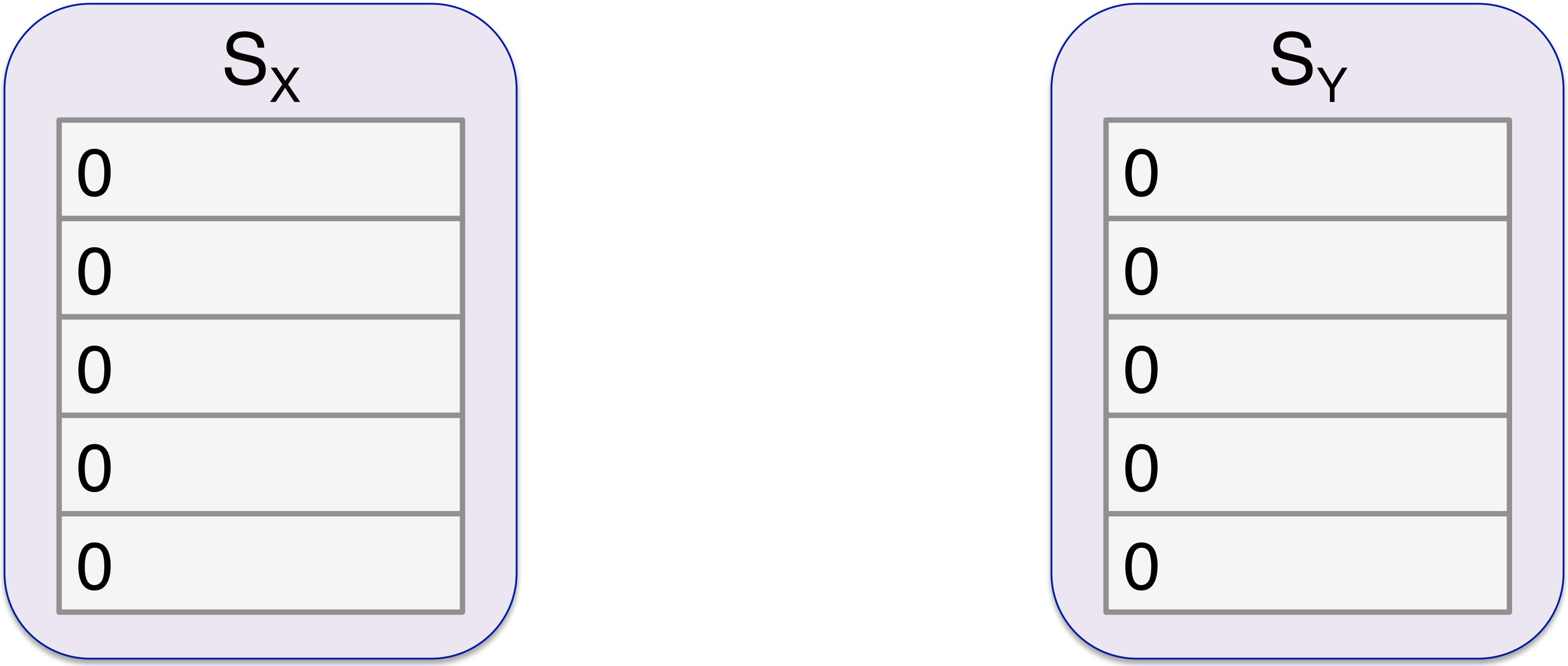
[Chaum ‘88]

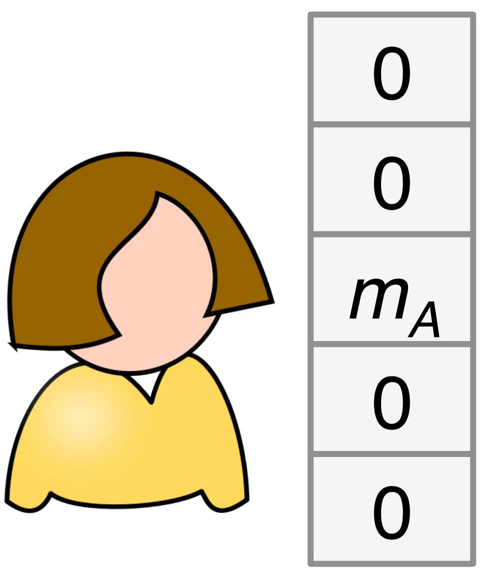


“Straw man” Scheme

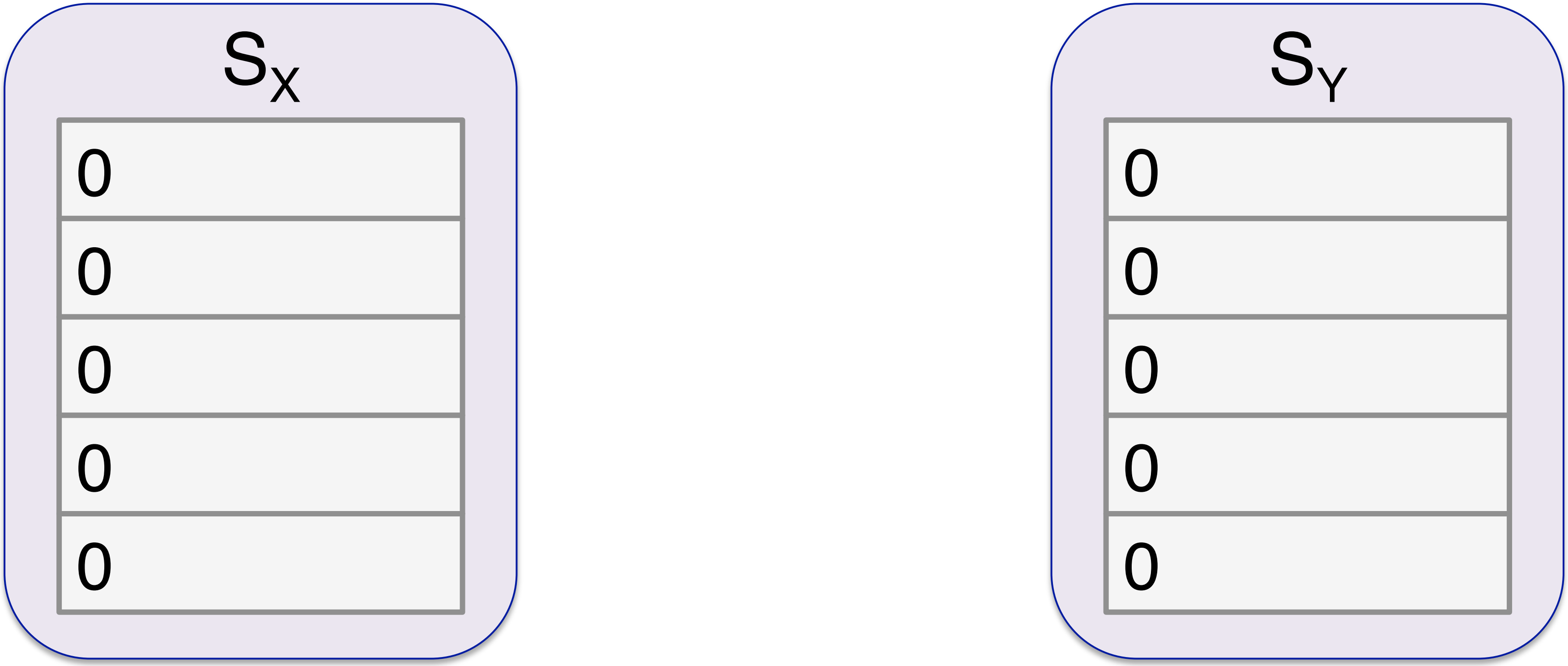


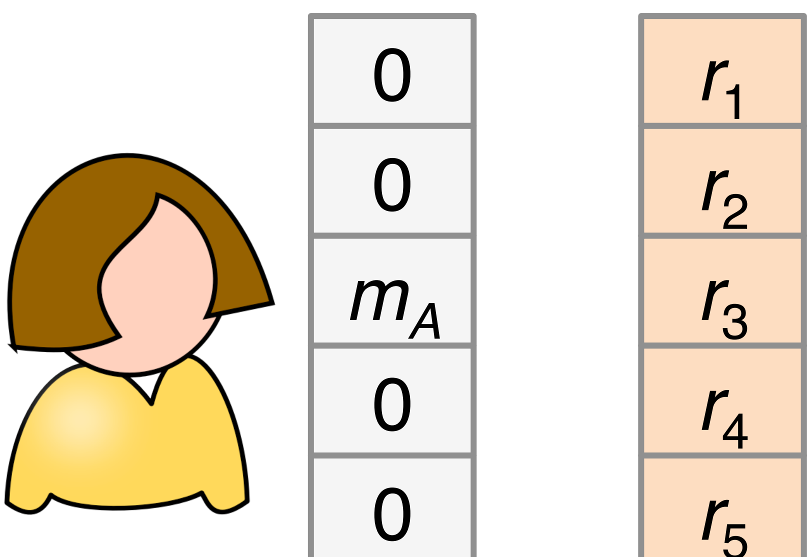
"Straw man" Scheme



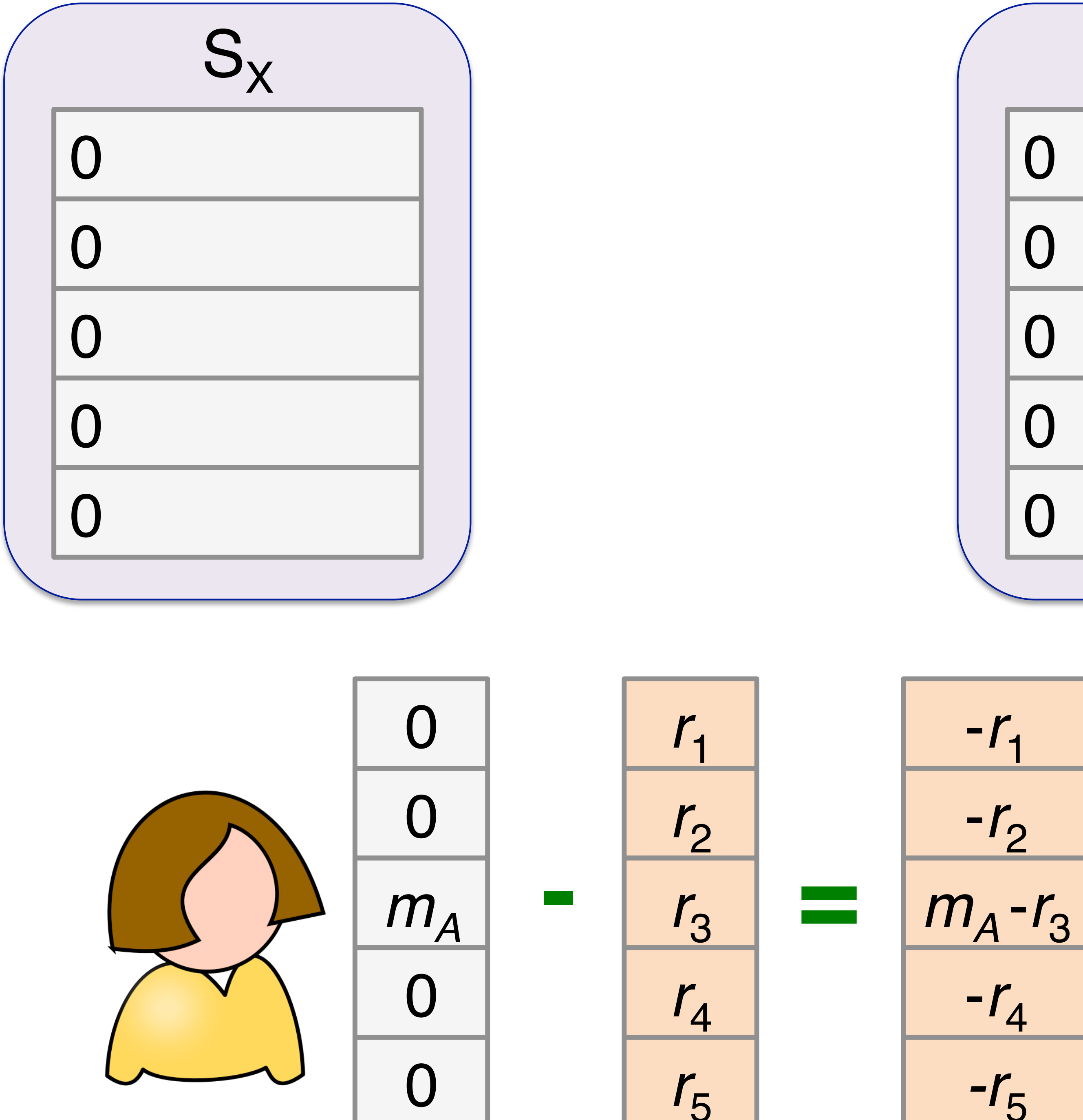


"Straw man" Scheme



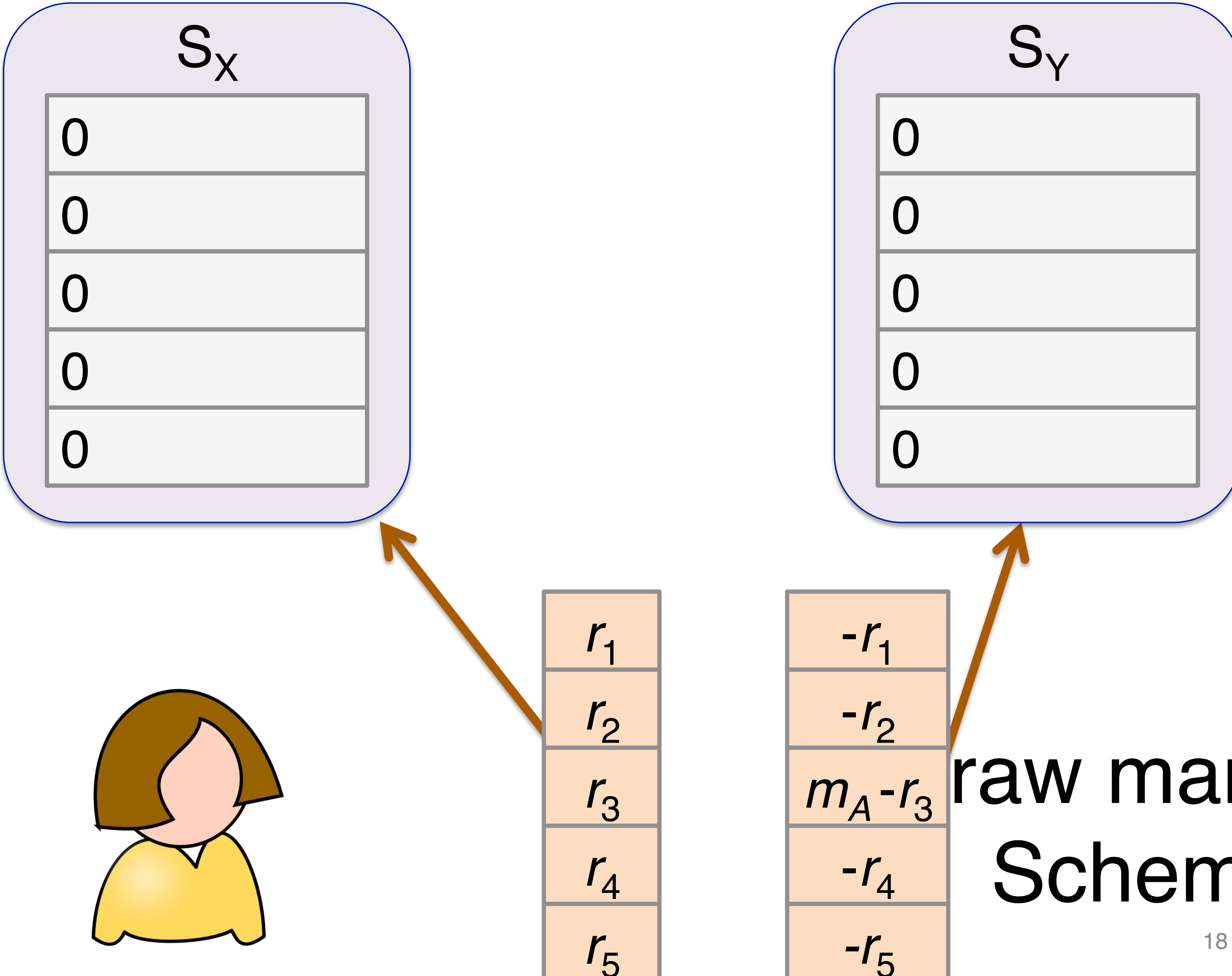


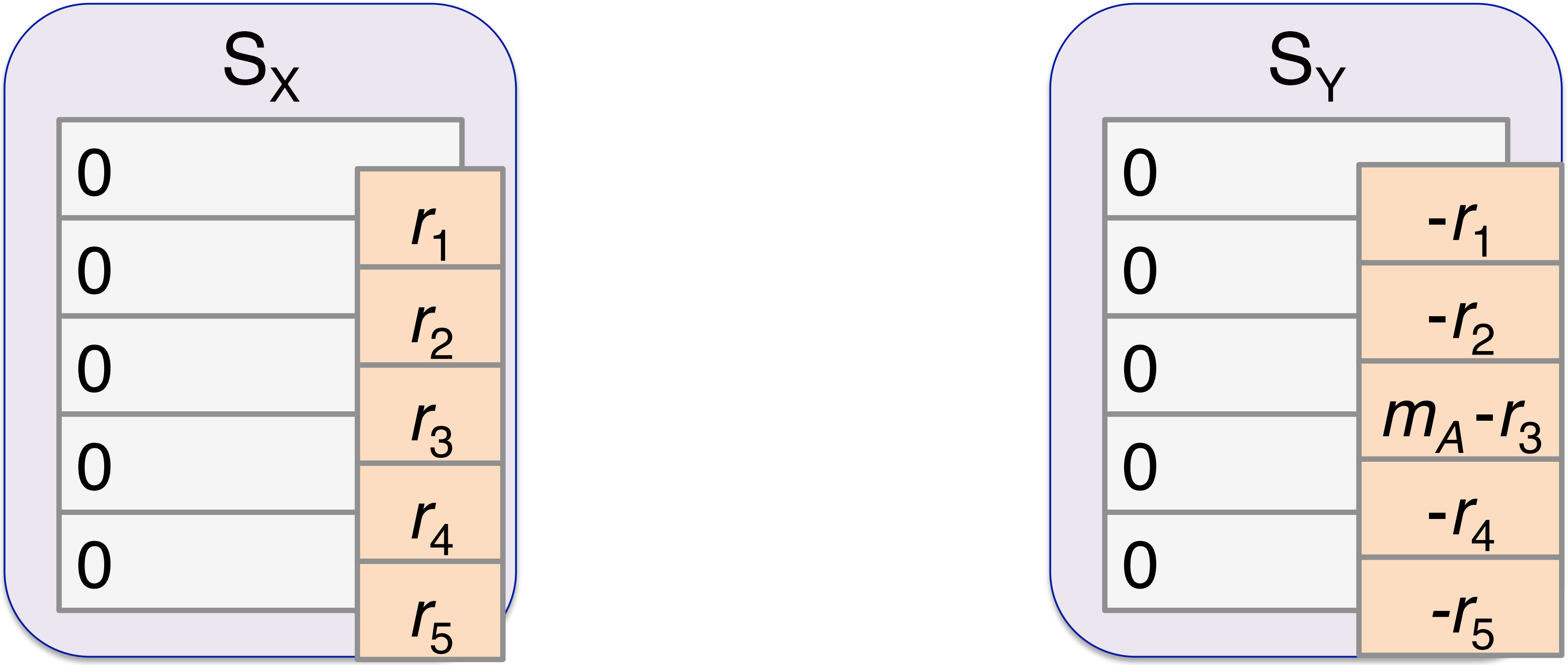
“Straw man” Scheme

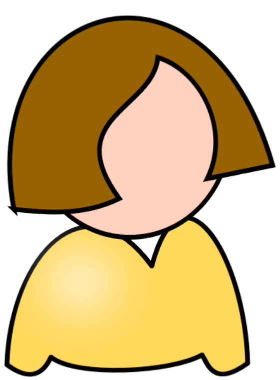


raw man”
Scheme

“Straw man” Scheme







"Straw man" Scheme

S_X	S_Y
r_1	$-r_1$
r_2	$-r_2$
r_3	$-r_3+m_A$
r_4	$-r_4$
r_5	$-r_5$

“Straw man” Scheme

S_X
r_1
r_2
r_3
r_4
r_5

S_Y
$-r_1$
$-r_2$
$-r_3+m_A$
$-r_4$
$-r_5$

0
0
0
0
m_B

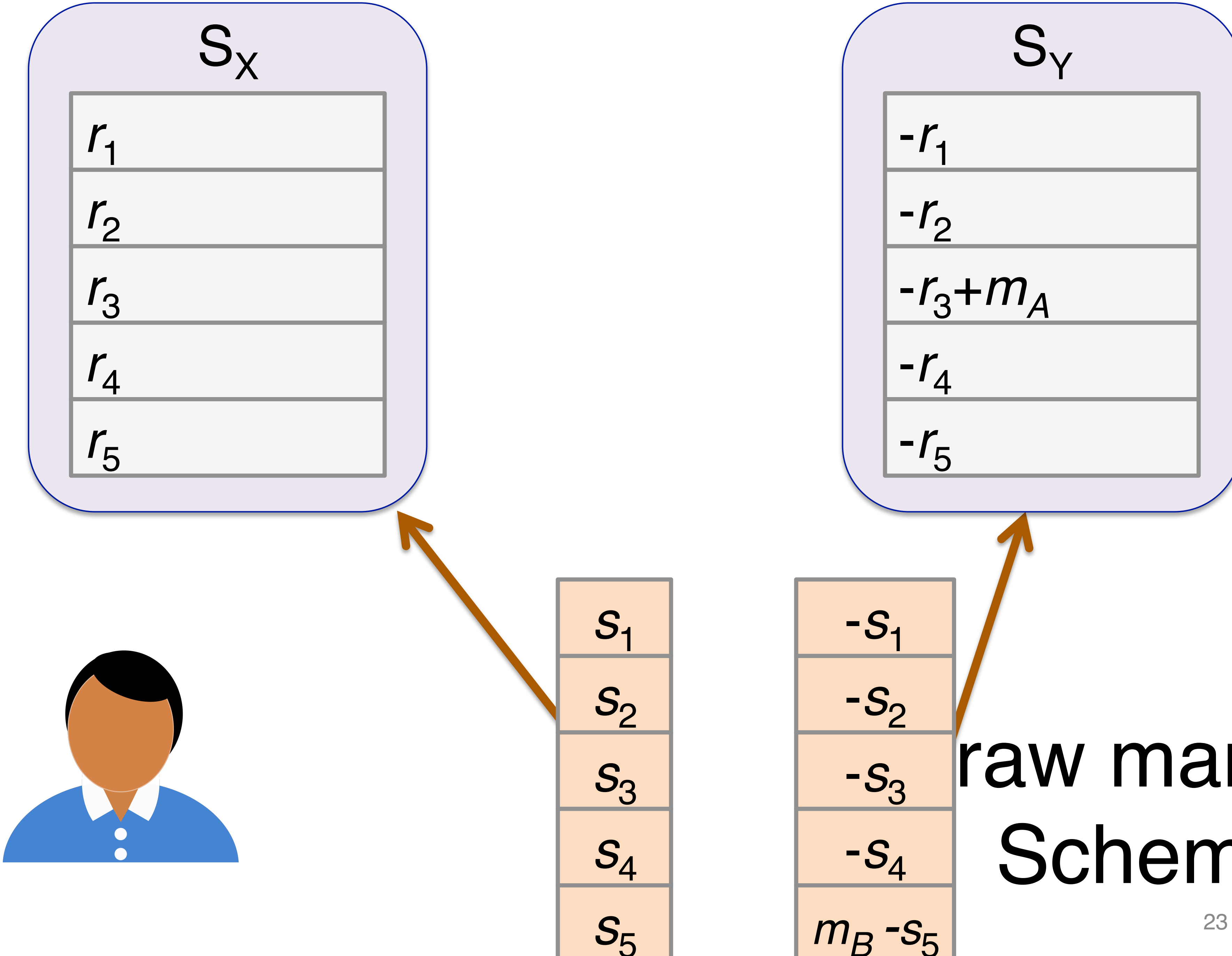
"Straw man" Scheme

S_X
r_1
r_2
r_3
r_4
r_5

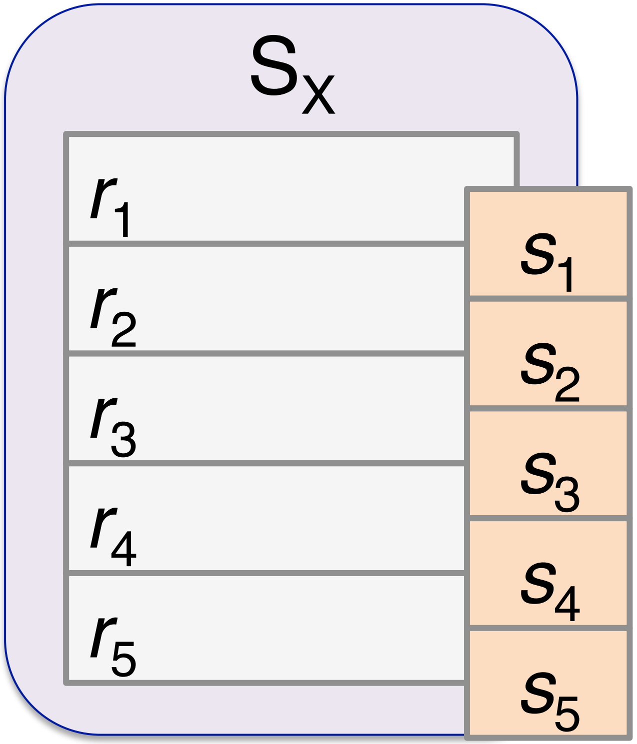
S_Y
$-r_1$
$-r_2$
$-r_3+m_A$
$-r_4$
$-r_5$

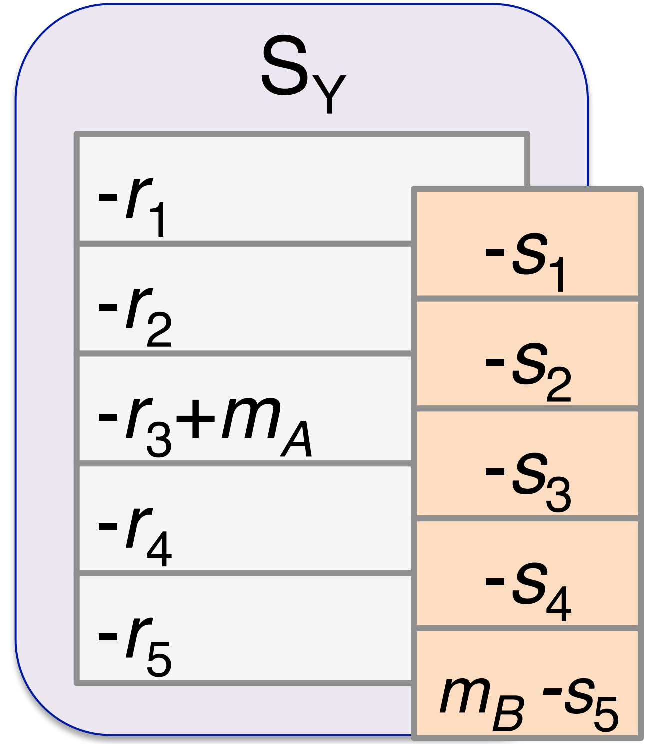
$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ m_B \end{bmatrix} - \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{bmatrix} = \begin{bmatrix} -s_1 \\ -s_2 \\ -s_3 \\ -s_4 \\ m_B - s_5 \end{bmatrix}$$

raw man”
Scheme



raw man” Scheme





“Straw man” Scheme

S_X
$r_1 + s_1$
$r_2 + s_2$
$r_3 + s_3$
$r_4 + s_4$
$r_5 + s_5$

S_Y
$-r_1 - s_1$
$-r_2 - s_2$
$-r_3 - s_3 + m_A$
$-r_4 - s_4$
$-r_5 - s_5 - m_B$

"Straw man" Scheme

S_X
$r_1 + s_1$
$r_2 + s_2$
$r_3 + s_3$
$r_4 + s_4$
$r_5 + s_5$

S_Y
$-r_1 - s_1$
$-r_2 - s_2$
$-r_3 - s_3 + m_A$
$-r_4 - s_4$
$-r_5 - s_5 - m_B$

“Straw man” Scheme

S_X
$r_1 + s_1$
$r_2 + s_2$
$r_3 + s_3$
$r_4 + s_4$
$r_5 + s_5$

S_Y
$-r_1 - s_1$
$-r_2 - s_2$
$-r_3 - s_3 + m_A$
$-r_4 - s_4$
$-r_5 - s_5 - m_B$

"Straw man" Scheme

S_X
$r_1 + s_1$
$r_2 + s_2$
$r_3 + s_3$
$r_4 + s_4$
$r_5 + s_5$

S_Y
$-r_1 - s_1$
$-r_2 - s_2$
$-r_3 - s_3 + m_A$
$-r_4 - s_4$
$-r_5 - s_5 - m_B$

“Straw man” Scheme

S_X
$r_1 + s_1$
$r_2 + s_2$
$r_3 + s_3$
$r_4 + s_4$
$r_5 + s_5$

$+$

S_Y
$-r_1 - s_1$
$-r_2 - s_2$
$-r_3 - s_3 + m_A$
$-r_4 - s_4$
$-r_5 - s_5 - m_B$

$=$

0
0
m_A
0
m_B

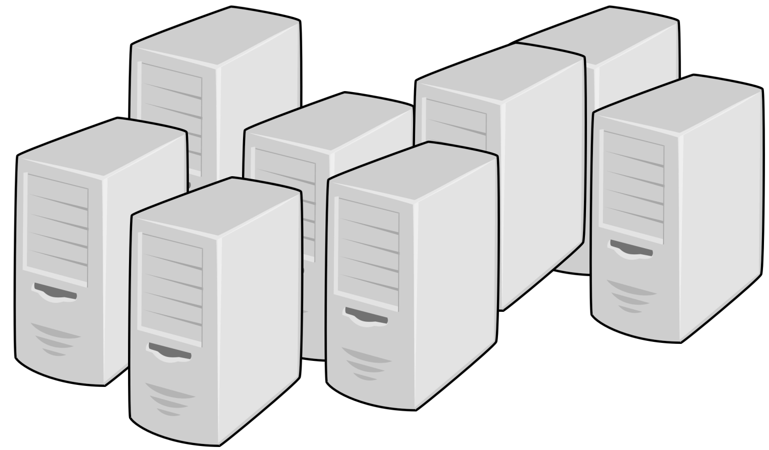
At the end of the day, servers combine DBs to reveal plaintext

“Straw man” Scheme

First-Attempt Scheme: Properties

"Perfect" anonymity as long as servers don't collude

- Can use *k* servers to protect against *k*-1 collusions



Practical efficiency: almost no "heavy" computation involved

Unlike a mix-net, storage cost is **constant** in the anonymity set size

Outline

- Motivation
- A “Straw man” scheme
- **Technical challenges**
- Evaluation

Outline

- Motivation
- A “Straw man” scheme
- **Technical challenges**
  - Collisions
  - Malicious clients
  - O(*L*) communication cost
- Evaluation

Outline

- Motivation
- A “Straw man” scheme
- **Technical challenges**
  - ~~Collisions~~
  - ~~Malicious clients~~
  - O(*L*) communication cost

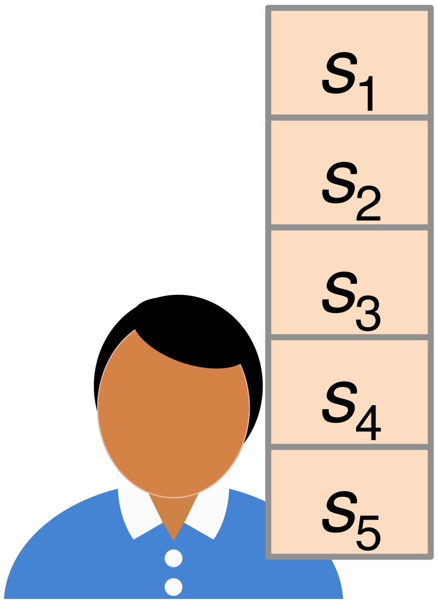
in the paper

- Evaluation

Challenge: Bandwidth Efficiency

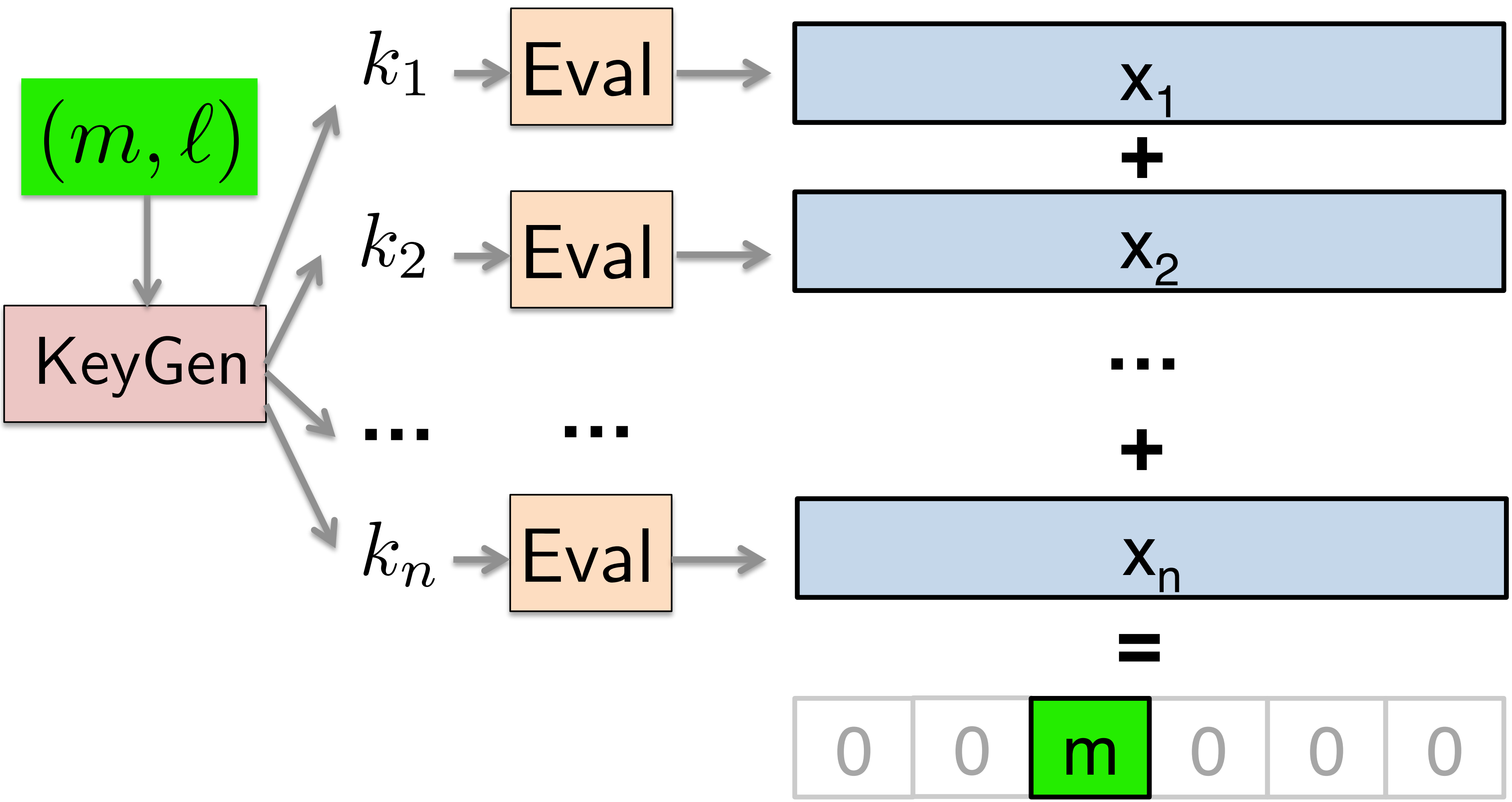
In "straw man" design, client sends DB-sized vector to each server

Idea: use a cryptographic trick to compress the vectors
→ Based on PIR protocols



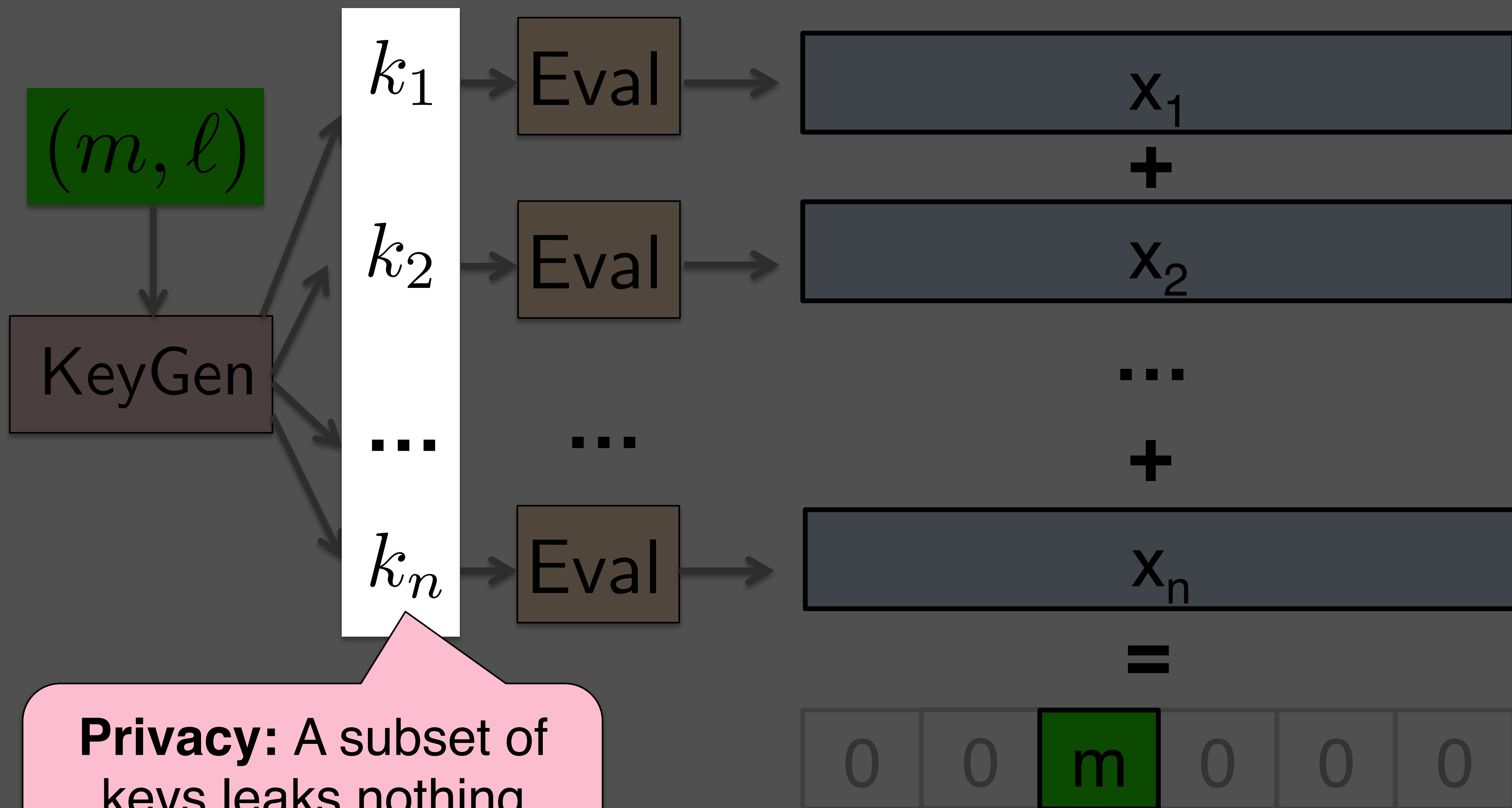
[Ostrovsky and Shoup 1997]

Distributed Point Function



[Gilboa and Ishai 2014]

Distributed Point Function

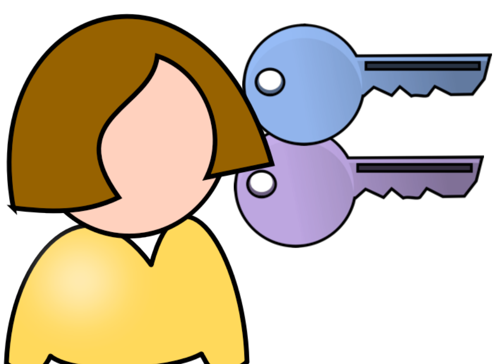


Privacy: A subset of keys leaks nothing about message or ℓ

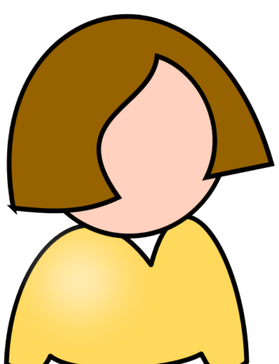
S_X
0
0
0
0
0

Eval(   )   Eval(   )

S_Y
0
0
0
0
0

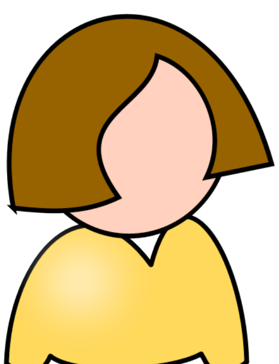


DPFs Reduce Bandwidth Cost



DPFs Reduce Bandwidth Cost

Alice sends
$L^{1/2}$ bits (instead of L)



- Two-server version just uses AES (no public-key crypto)
- With fancier crypto, privacy holds even if all but one server is malicious

[Chor and Gilboa 1997]
[Gilboa and Ishai 2014]

Outline

- Motivation
- Definitions and a “Straw man” scheme
- Technical challenges
- **Evaluation**

Bottom-Line Result

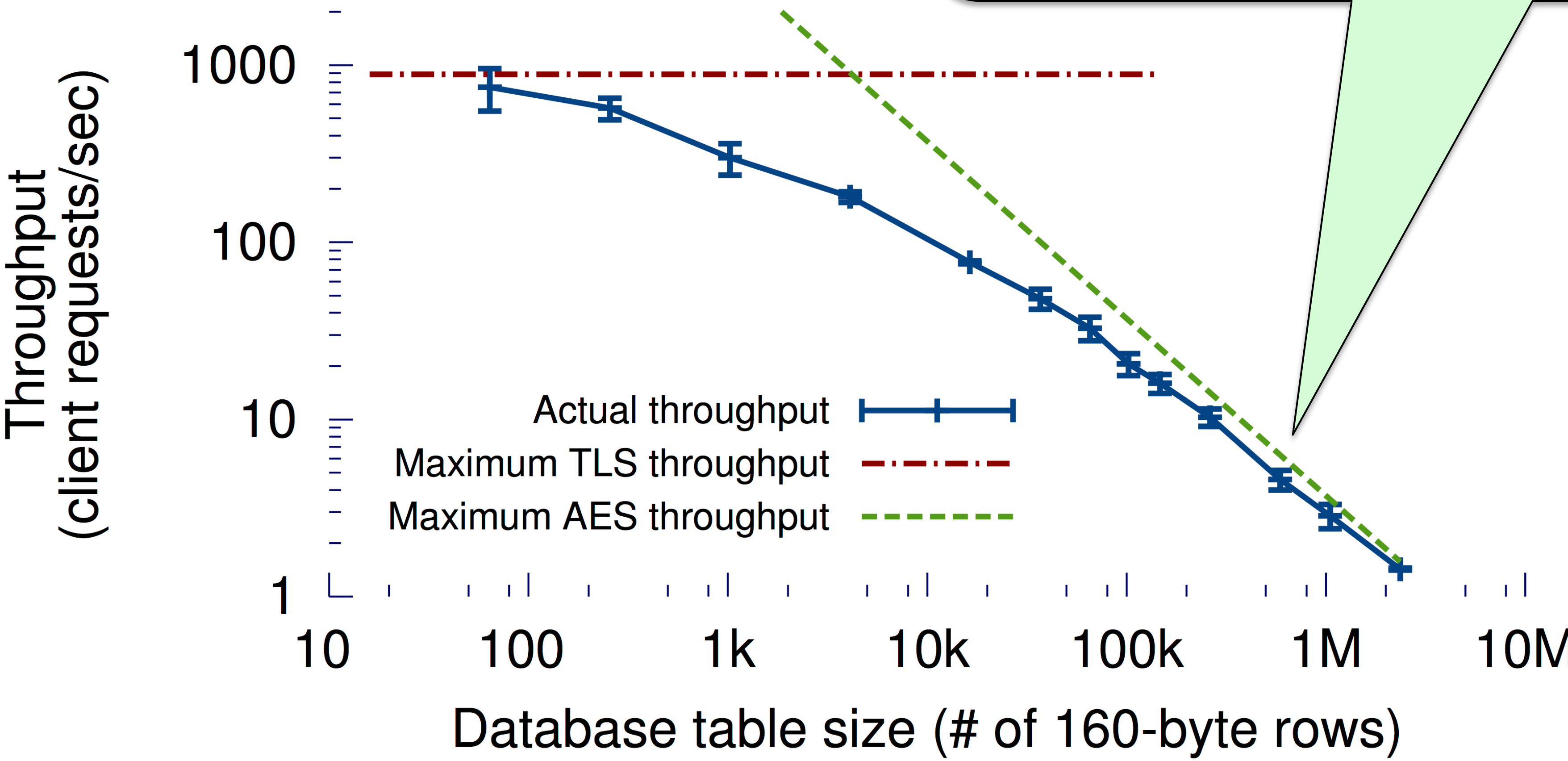
- Implemented the protocol in Go
- For a DB with 65,000 Tweet-length rows, can process **30 writes/second**
- Can process **1,000,000 writes** in 8 hours on a single server

→ **Completely parallelizable workload**

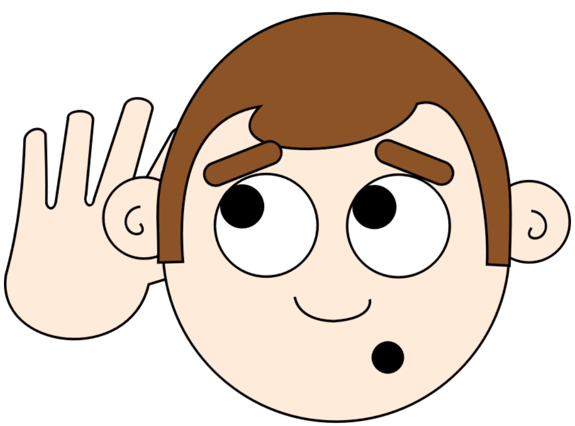
Throughp

(anonymous Tw

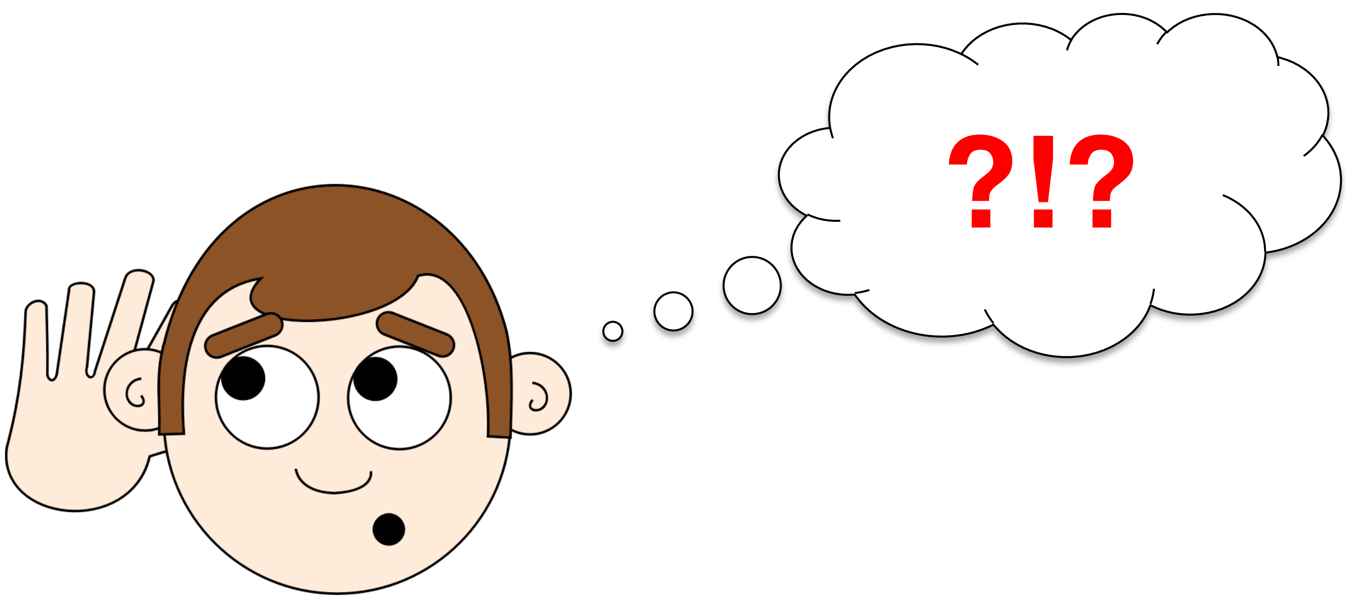
At large table sizes, AES cost dominates



Time	From	To	Size
10:12	Alice	**taxfraud@stanford.edu**	2543 B
10:15	Bob	Alice	567 B
10:17	Carol	Bob	450 B
10:22	Dave	Alice	9382 B



Time	From	To	Size
10:12	Alice	Riposte Server	207 KB
10:15	Bob	Riposte Server	207 KB
10:17	Carol	Riposte Server	207 KB
10:22	Dave	Riposte Server	207 KB



Conclusion

In many contexts, “hiding the metadata” is as important as hiding the data

Combination of crypto tools with systems design ➔ 1,000,000-user anonymity sets

Next step: Better performance at scale