Repeatable Oblivious Shuffling of Large Outsourced Data Blocks

Zhilin Zhang+, Ke Wang, Weipeng Lin,
Ada Wai-Chee Fu, Raymond Chi-Wing Wong
+Simon Fraser University, Amazon

Outsourcing in the Cloud

2019 Public cloud services market >$206.2 B

Source: Gartner's annual forecast of worldwide public cloud service revenue

SECURITY ALERT

By Keir Thomas, PCWorld | DECEMBER 23, 2010 10:08 AM PT

About | Practical security advice

NEWS

Microsoft Cloud Data Breach Heralds Things to Come

FOX NEWS U.S. World Opinion Politics Entertainment Business Lifestyle TV Fox Nation Listen More

Hot Topics AOC challenger | 'Squad' strikes Pelosi | Flesh-eating bacteria

ADVERTISEMENT

TECH • CHANGING FACE OF SECURITY

LinkedIn Lost 167 Million Account Credentials in

Sensitive data must be encrypted before putting on the cloud server

By Chris Baraniuk
Technology reporter

25 April 2016 Share

HULTON ARCHIVE

Voter records made publicly accessible online have since been removed

Secret

January 30, 2019 Mohit Kumar

CBS NEWS NEWS SHOWS LIVE

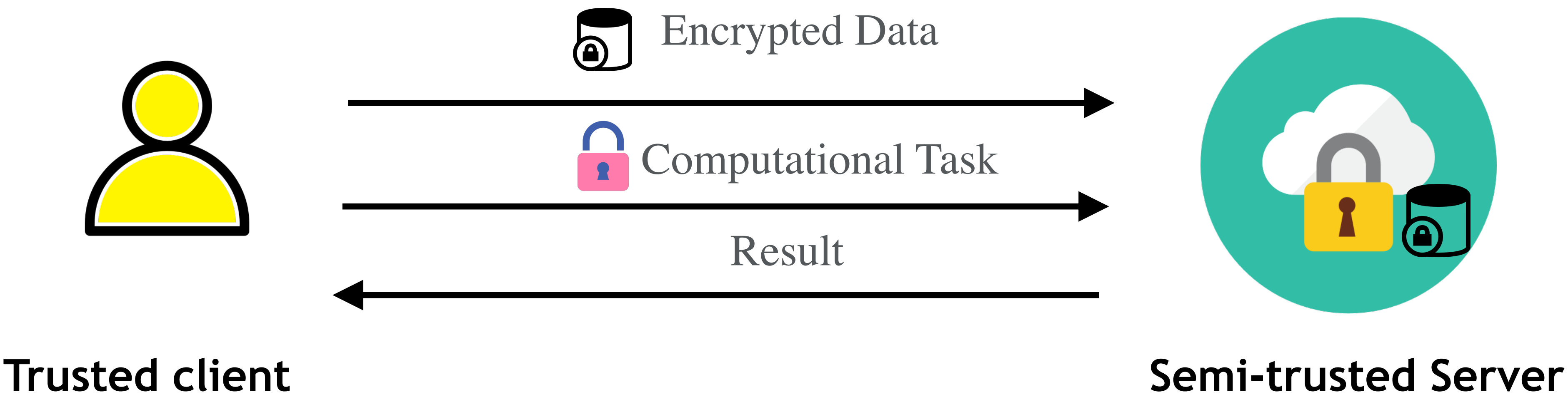
Hundreds of millions of Facebook user records were exposed on Amazon cloud server

BY JASON SILVERSTEIN
UPDATED ON: APRIL 4, 2019 / 11:35 AM / CBS NEWS

Cost of a Retail Data Breach: $179 Million for Home Depot

Mar 14, 2017 | Cybersecurity News

Secure Computation Outsourcing



Encryption is Insufficient

Input: [a], [b]

Task:

if $a>b$:

    branch 1 ← a=2, b=1

else:

    branch 2 ← a=1, b=2

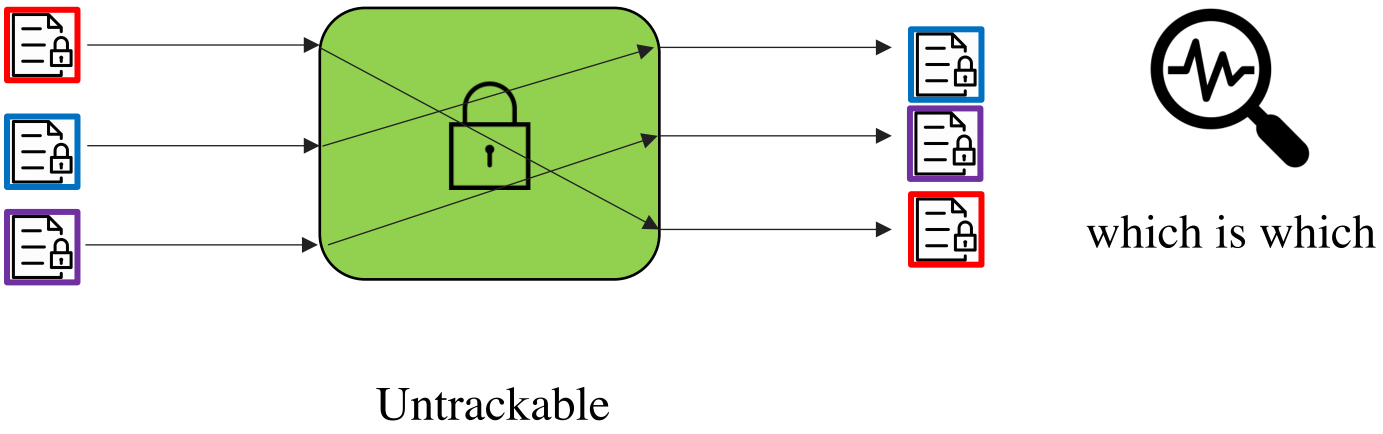
Oblivious algorithm: make the control flow be independent of the input data

- oblivious transfer/ sorting/ **shuffling**, etc.

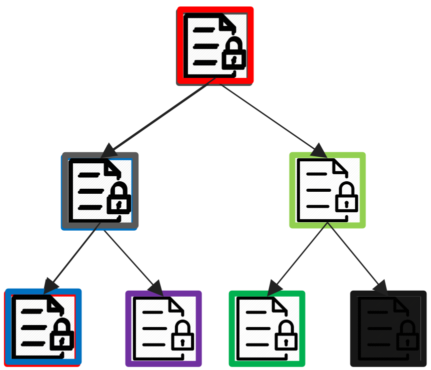
Problem

Oblivious Shuffling (OS)

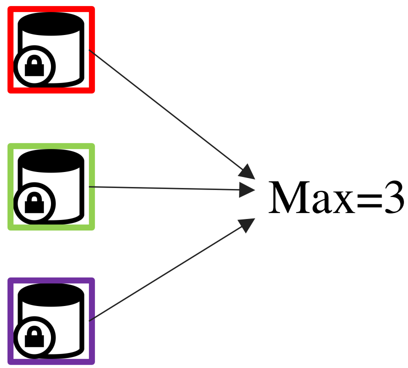
A shuffling of n encrypted data blocks $[B] = ([B_1], \cdots, [B_n])$ according to a permutation π is oblivious if the server is unable to infer π.



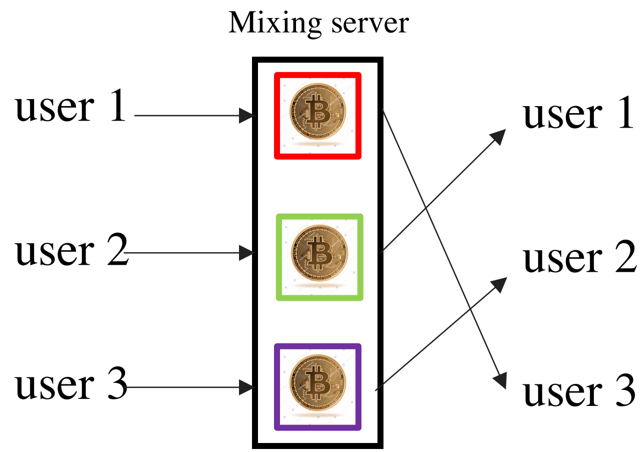
Application



private data access
(hide access pattern)



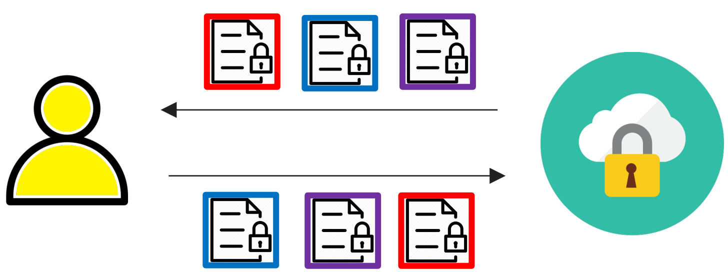
private data integration/sharing
(hide data source)



coin mixing in cryptocurrency
(hide owner anonymity)

State of the Art

All existing OS methods rely on the movement of outsourced data to the client.



download for shuffling

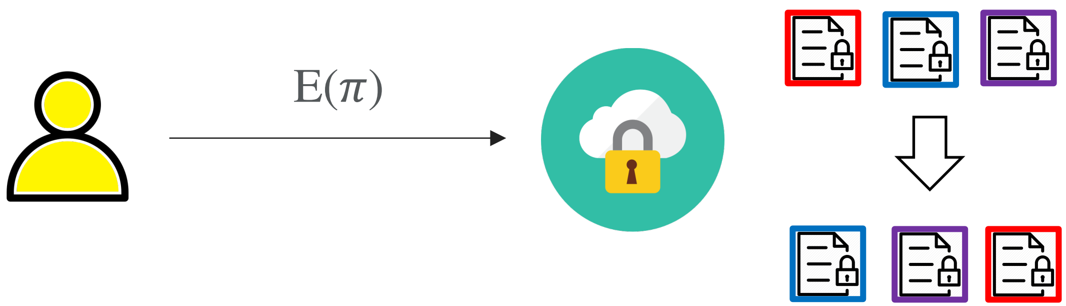
download for peel-off

heavy communication for shuffling
large-sized blocks

Repeatable Oblivious Shuffle

Definition

An oblivious shuffle of $[B] = ([B_1], \cdots, [B_n])$ is repeatable if it is performed by the server without increasing encryption layers.



Preliminaries

- Homomorphic matrix multiplication

$$[M_1] \odot M_2 = [M_1 \cdot M_2]$$

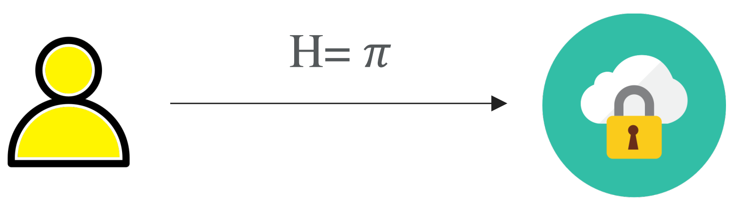
- Matrix based data shuffling

$$B \cdot \pi \ = (B_1, B_2) \cdot \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \Rightarrow (B_2, B_1)$$

Main Idea

Key Requirements

- repeatability: server side shuffling, no increase in encryption layers
- obliviousness: shuffling must be oblivious



$[B]\odot\pi \rightarrow [B \cdot \pi\,]$

split the information of π into
plaintext H and some ciphertext $[H_A]$

Formalization

$$[B^{(\eta)}] \leftarrow ROS(\pi^{(\eta)}, [B^{(\eta-1)}])$$

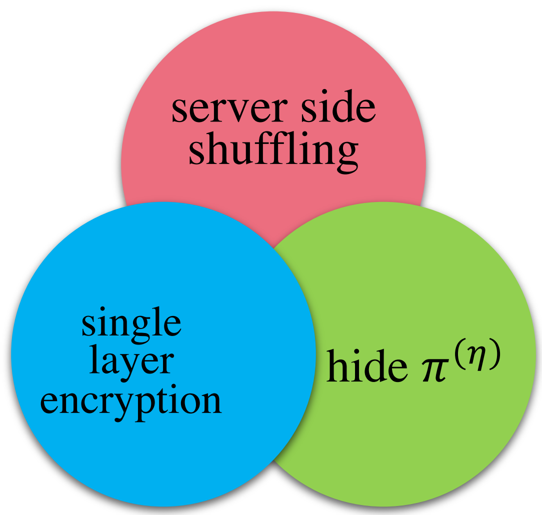
data after shuffling

$[B^{(\eta)}] = [B \cdot \pi^{\eta-1} \cdot \pi^{(\eta)}]$

permutation matrix

data before shuffling

$[B^{(\eta-1)}] = [B \cdot \pi^{\eta-1}]$



Construction

1. pick $\pi^{(\eta)}$
2. compute $H^{(\eta)}$ and $\left[H_A^{(\eta)}\right]$

$H^{(\eta)}$ and $\left[H_A^{(\eta)}\right]$

3. compute the shuffling result by

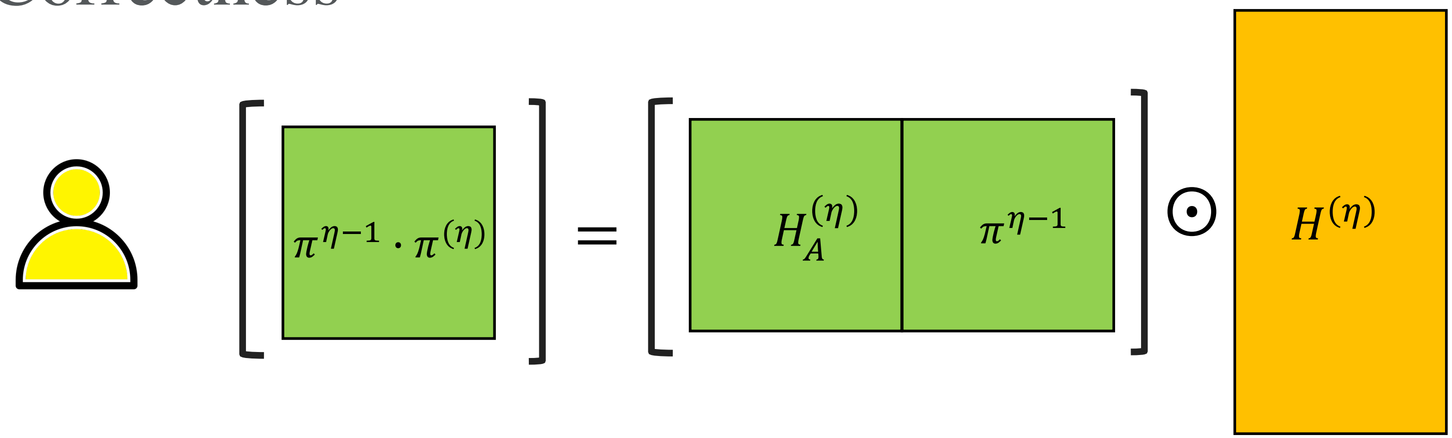
$$\left[B \times \pi^{\eta-1} \cdot \pi^{(\eta)} \right] = \left[B \times H_A^{(\eta)} \; \pi^{\eta-1} \right] \odot H^{(\eta)}$$

data blocks

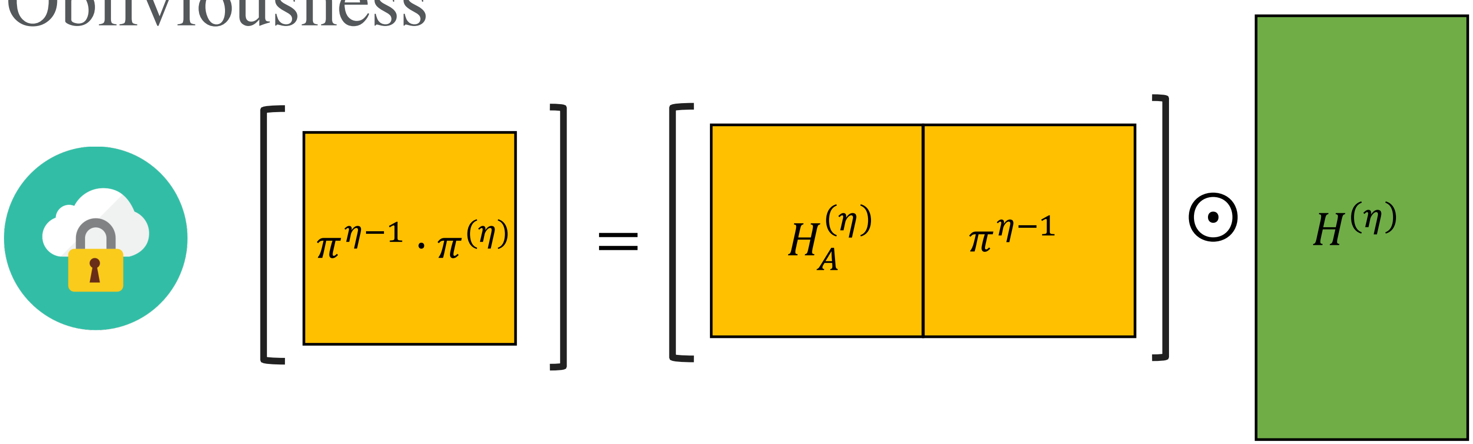
coefficient matrix

Analysis

Correctness



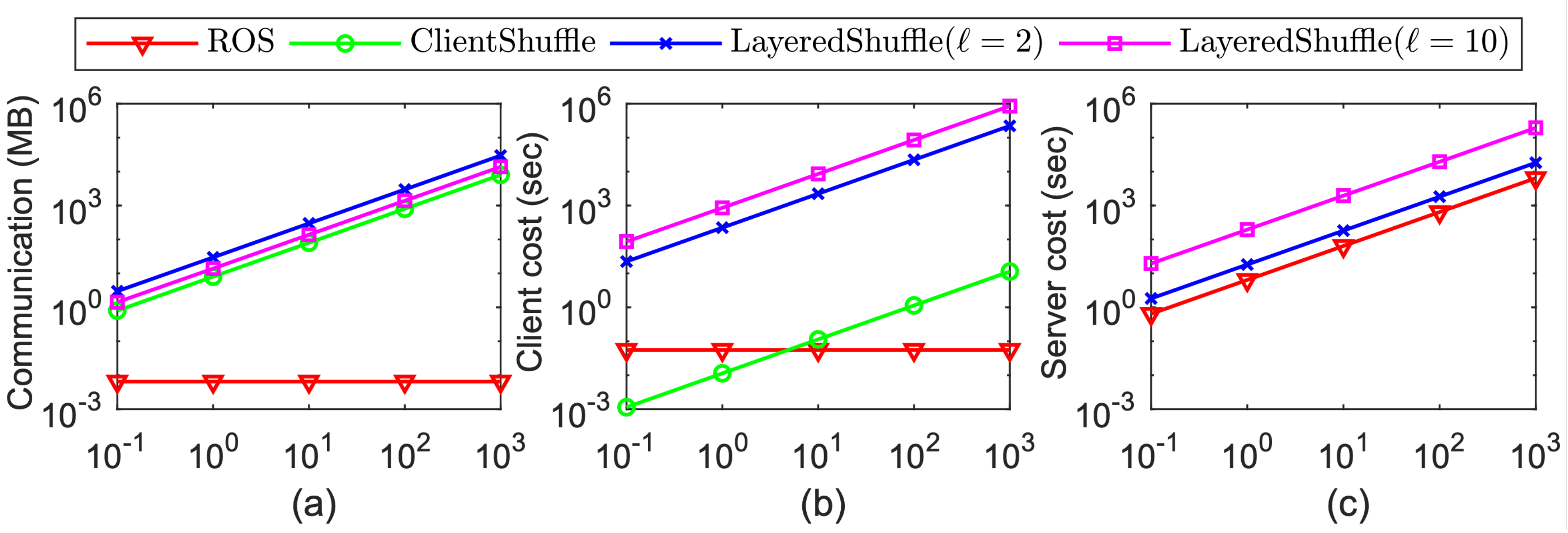
Obliviousness



Experimental Settings

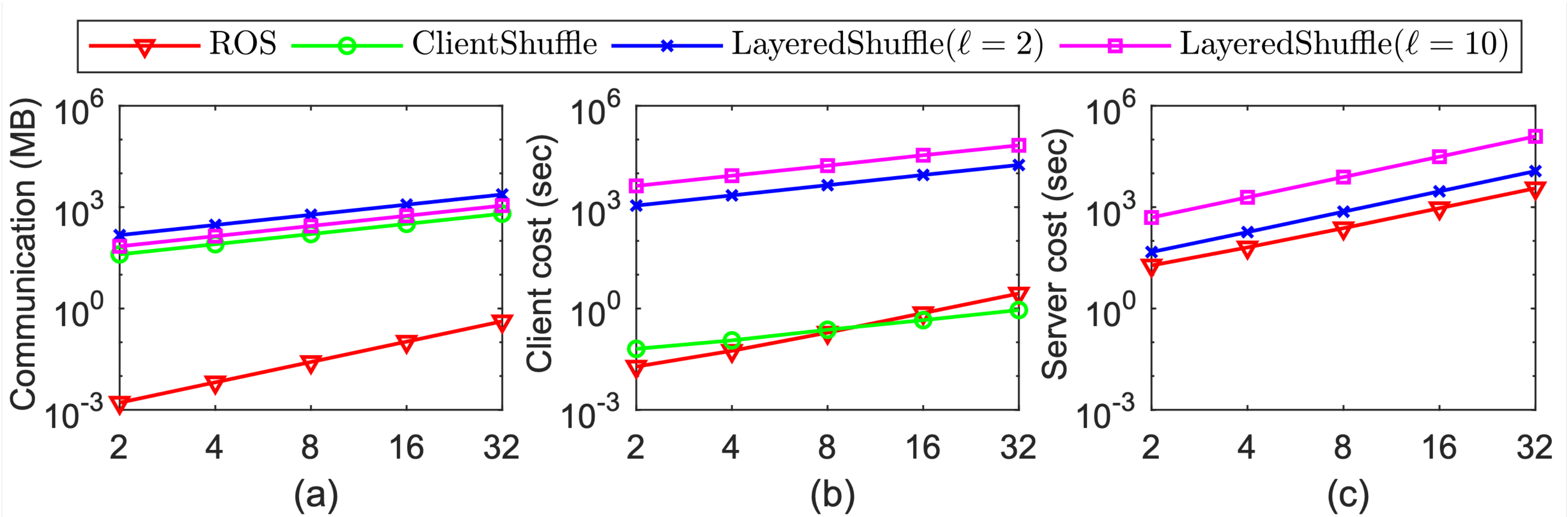
	Algorithm	Description
Our approach	ROS	Server-side shuffling without increasing encryption layer
Baseline	ClientShuffle	Client-side shuffling (download data for every shuffling)
	LayeredShuffle ($l = 2$)	Service-side shuffling with increasing encryption layers (download data for peeling off extra layers after every l shuffles)
	LayeredShuffle ($l = 10$)	

Effect of Block Size m



Shuffle cost w.r.t. block size m (MB) (n = 4, ClientShuffle has no server computation and thus not reported)

Effect of Block Number n



Shuffle cost w.r.t. block number n (m=10 MB, ClientShuffle has no server computation and not reported)


Q and A?