PCKV: Locally Differentially Private Correlated Key-Value Data Collection with Optimized Utility

Xiaolan Gu[*], Ming Li[*], Yueqiang Cheng[**], Li Xiong[#] and Yang Cao[†]

[*] University of Arizona [**] Baidu Security [#] Emory University [†] Kyoto University

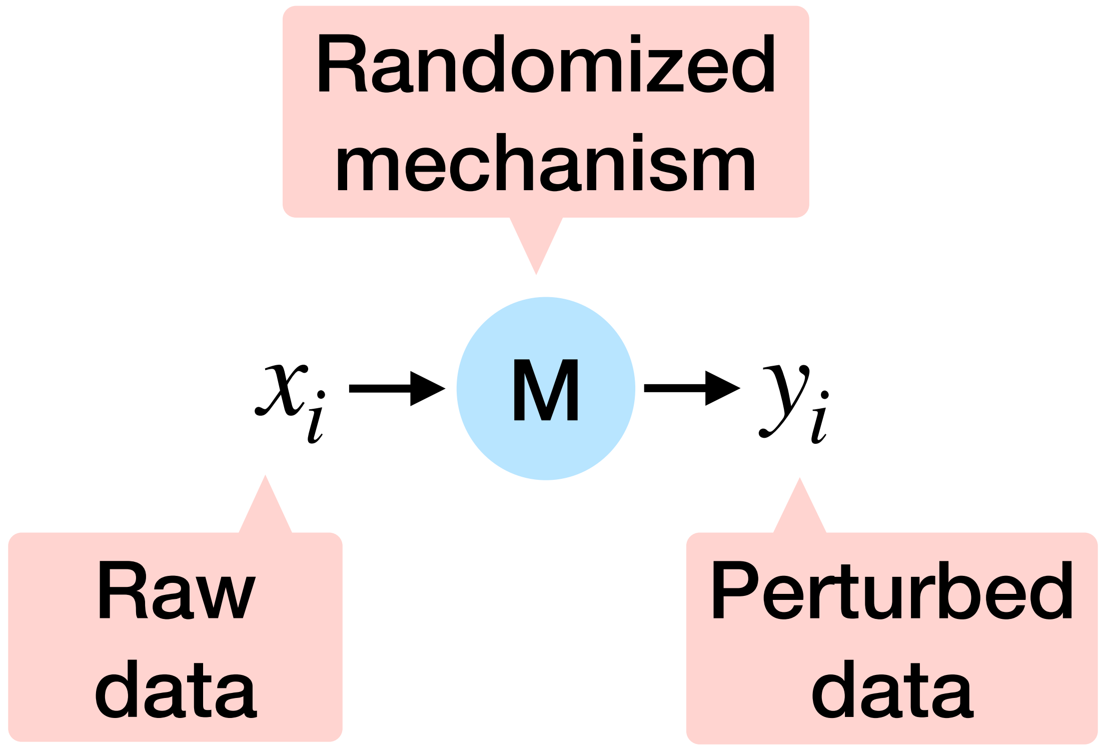
USENIX Security Symposium, August 2020

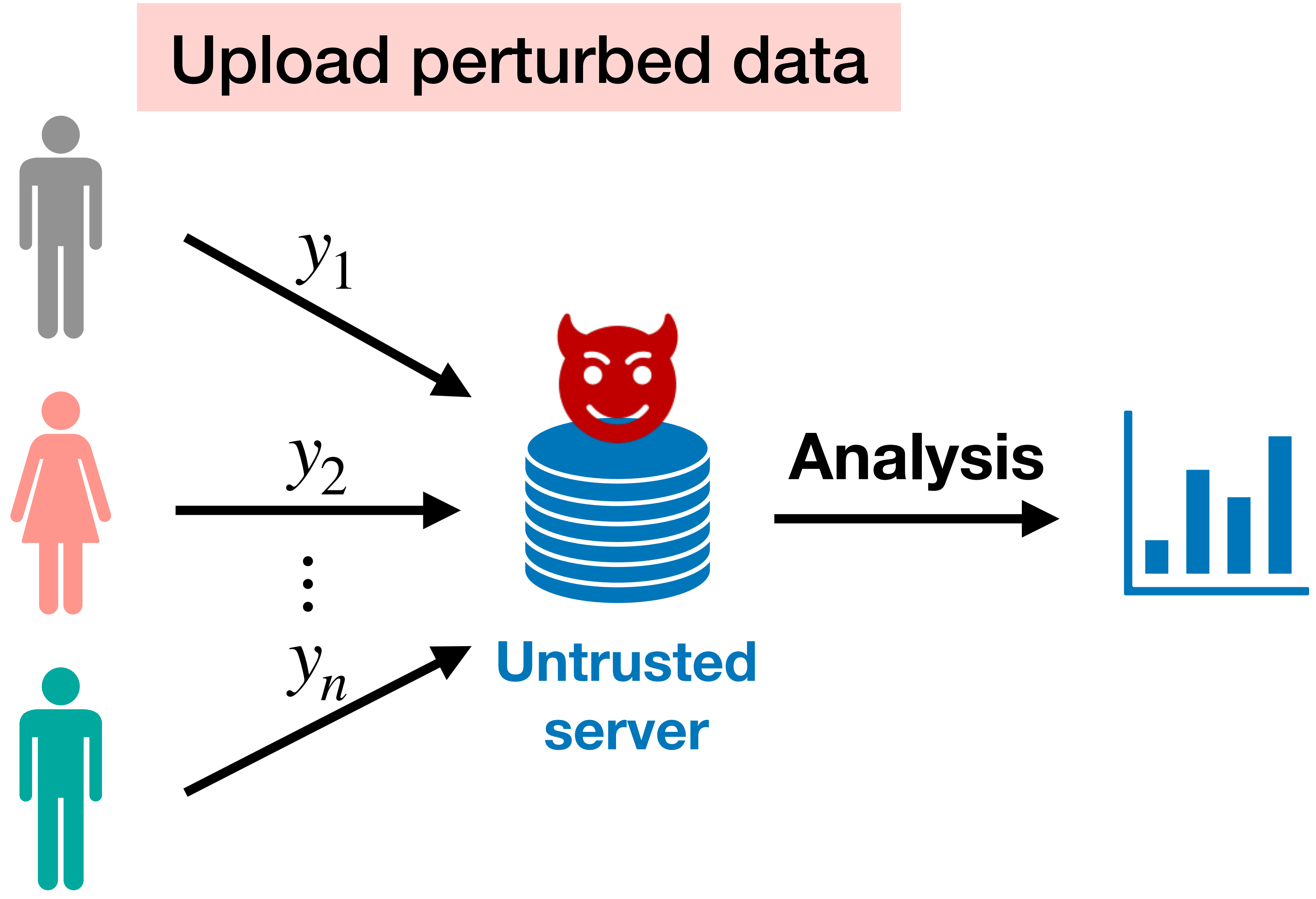
Overview

- Background of LDP
- Problem Statement and Existing Mechanism
- Our Framework: PCKV
- Experiments
- Conclusion

Background

- Companies are collecting our private data to provide better services (Google, Facebook, Apple, Yahoo, Uber, …)
- However, privacy concerns arise
  - Yahoo: massive data breaches impacted 3 billion user account, 2013
  - Facebook: 267 million users' data has reportedly been leaked, 2019
  - …
- Possible solution: locally private data collection model





Local Differential Privacy (LDP) [Duchi et al, FOCS' 13]

A mechanism M satisfies ϵ-LDP if and only if for any pair of inputs x, x' and any output y

$$\frac{\Pr(M(x) = y)}{\Pr(M(x') = y)} \leqslant e^{\epsilon}$$

- x, x' : the possible input (raw) data (generated by the user)
- y : the output (perturbed) data (public and known by adversary)
- ϵ : privacy budget (a smaller ϵ indicates stronger privacy)

An adversary cannot infer whether the input is x or x' with high confidence (controlled by ϵ)

Applications of LDP

Google Developers

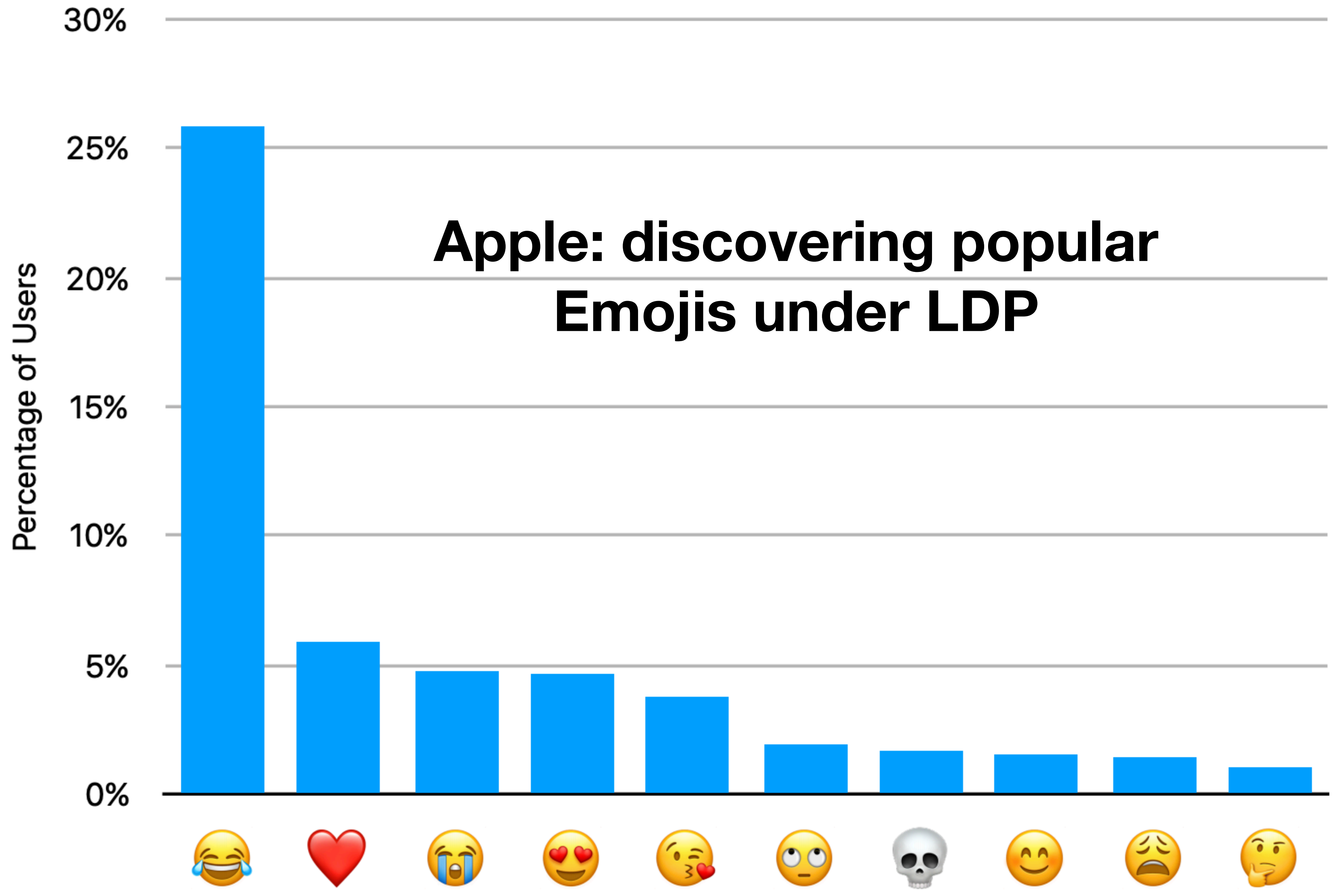
Blog of our latest news, updates, and stories for developers

Enabling developers and organizations to use differential privacy

Thursday, September 5, 2019

Posted by Miguel Guevara, Product Manager, Privacy and Data Protection Office

Source:
https://developers.googleblog.com/2019/09/enabling-developers-and-organizations.html



Source:
https://machinelearning.apple.com/2017/12/06/learning-with-privacy-at-scale.html

LDP Protocol: Randomized Response

- Randomized Response (RR) [Warner, 1965]: reports the truth with some probability (for binary answer: yes-or-no)

Advanced versions: Unary Encoding, Generalized RR, …

- Example: Is your annual income more than 100k?

Truth x	Response y	
1	1	w.p. p
	0	w.p. $1-p$
0	1	w.p. $1-p$
	0	w.p. p

Frequency of response y

Frequency estimation: $\hat{f} = \dfrac{f-(1-p)}{2p-1}$

Unbiasedness: $\mathbb{E}[\hat{f}] = f^*$

True frequency

To satisfy ϵ-LDP: $p = \dfrac{e^\epsilon}{e^\epsilon + 1}$ **(since** $\dfrac{p}{1-p} = e^\epsilon$ **)**

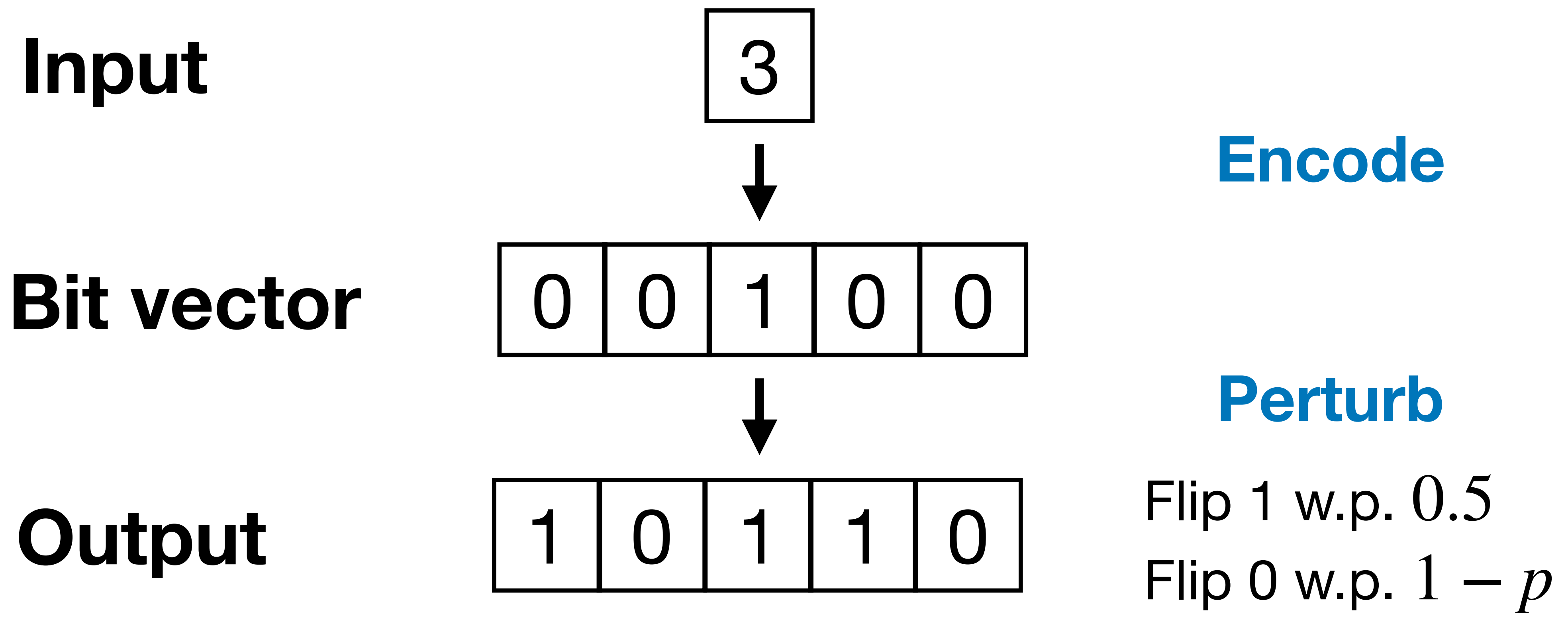
$\mathbb{E}[f] = f^*p + (1-f^*)(1-p) = (2p-1)f^* + (1-p)$

Extend RR for General Cases

- Assume the domain size is d (taking $d = 5$ for example)

Optimized Unary Encoding (OUE)

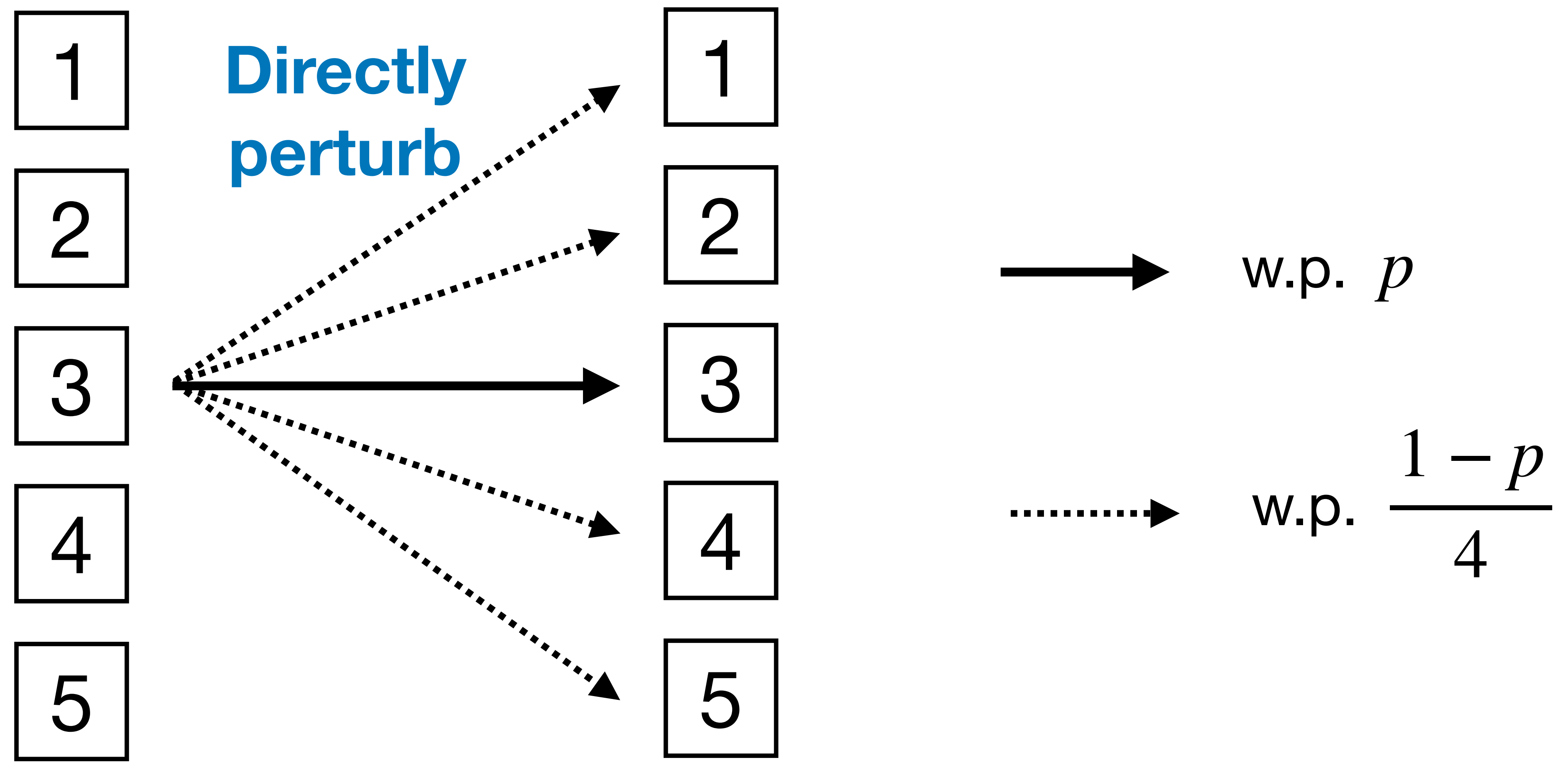
[Wang et al, USENIX Security' 17]



To satisfy ϵ-LDP: $p = \dfrac{e^{\epsilon}}{e^{\epsilon}+1}$

Staircase or Generalized RR (GRR)

[Kairouz et al, NeuIPS' 16]

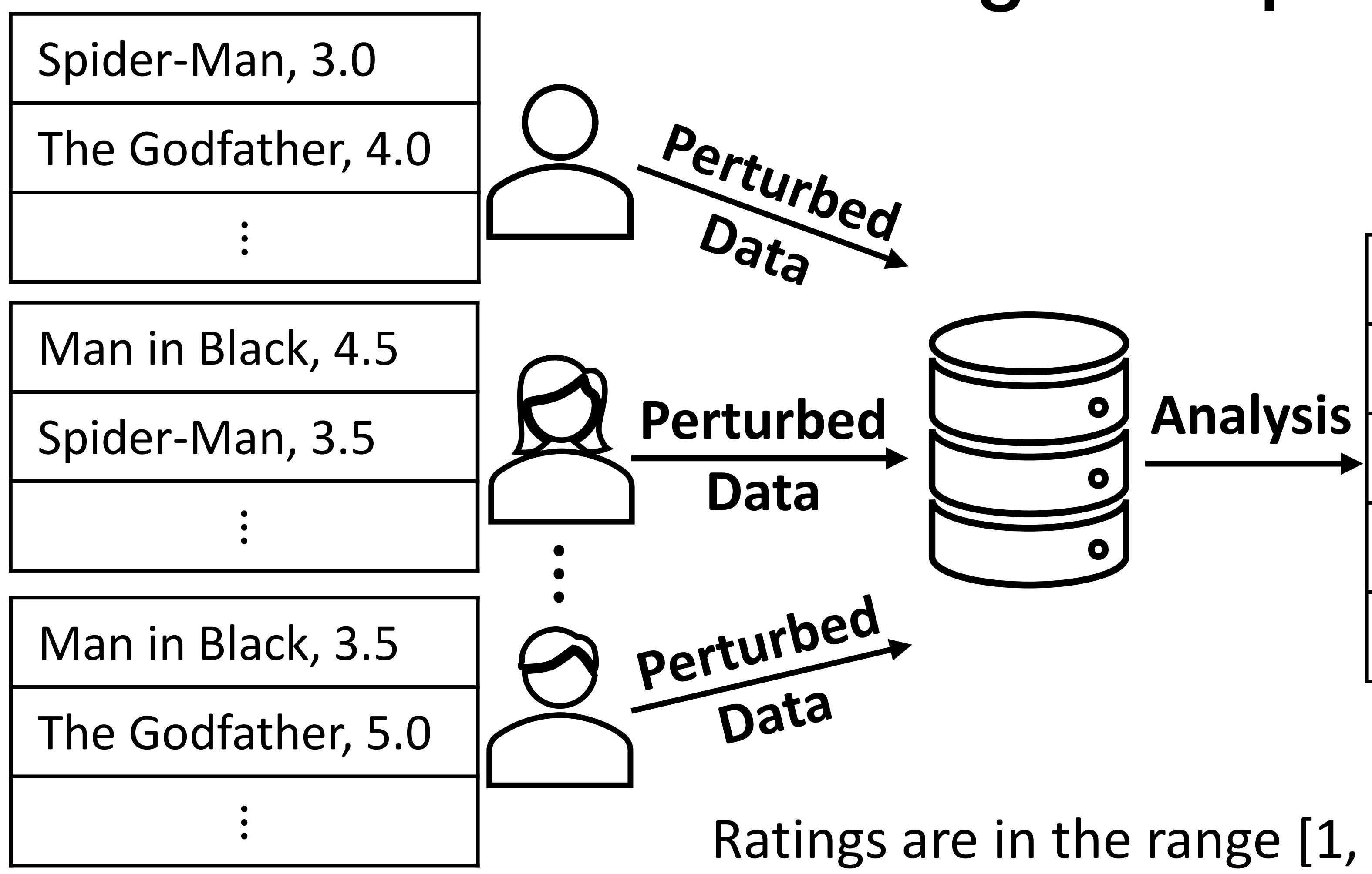


To satisfy ϵ-LDP: $p = \dfrac{e^{\epsilon}}{e^{\epsilon}+d-1}$

RR, OUE and GRR are building block mechanisms for frequency aggregation

Key-Value Data Collection

A motivating example (movie rating system)



Movies	# Ratings	Avg. Rating
Man in Black	1200	4.1
Spider-Man	1000	3.3
The Godfather	200	4.7
⋮	⋮	⋮

Ratings are in the range [1, 5]

- Data Type: each user has multiple key-value pairs
- Data Domain: key in $\{1,2,\cdots,d\}$, value in $[-1,1]$
- Task: frequency and mean estimation
- Threat Model: honest-but-curious server
- Objectives: good privacy-utility tradeoff

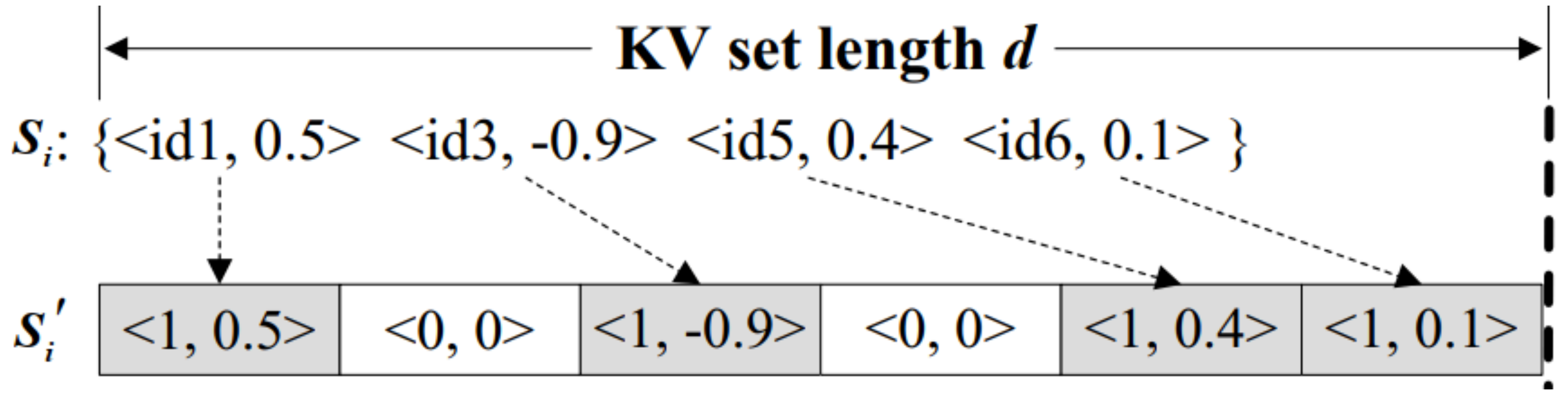
Challenges

1. Each user has different number of key-value pairs.

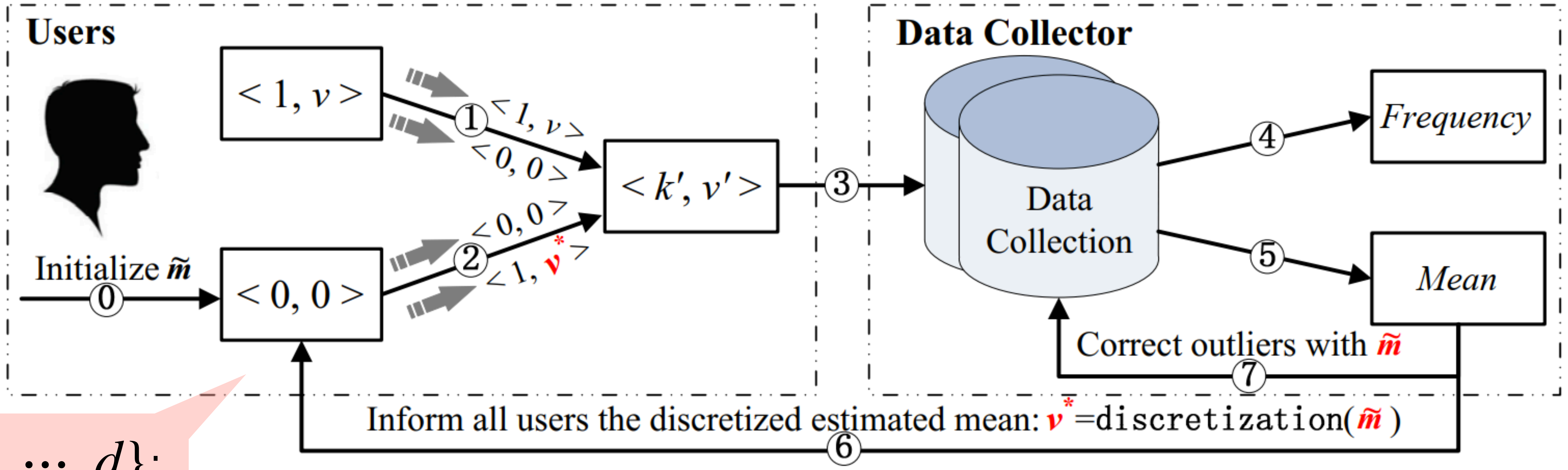
   Reporting all pairs will lead to a small budget and large error in each pair

   Sampling an index j from the whole domain (with size d) and reporting the j-th pair cannot make full use of the original pairs
2. If a fake key is reported, how to report the corresponding value?
3. How to design an optimal mechanism with the best privacy-utility tradeoff?

Existing Mechanism: PrivKVM [Ye et al, S&P' 19]



Step 1. Convert key-value pairs into a vector



Step 2. Iteratively update the mean of each key
(use sequential composition)

In each round, each user 1) randomly samples an index j from $\{1,\cdots,d\}$; 2) privately reports the j-th pair (if a fake key is reported, then the value will be perturbed from the estimated mean by the server)

Limitations of PrivKVM

- Multiple rounds requires all users to be always online and the privacy budget in each round is very small (thus large error).
- The naive sampling protocol may not work well for a large domain.
- No improved privacy budget composition (although key and value are perturbed with some correlation).

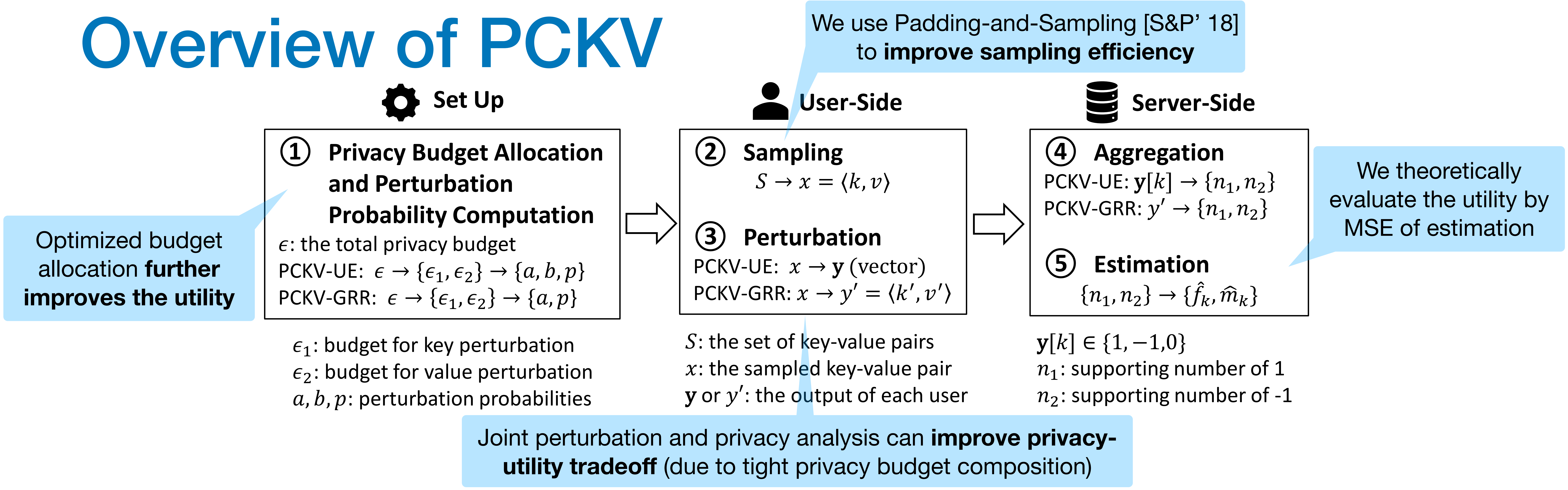
Our Mechanism

- Only one round
- Advanced sampling protocol
- Tight privacy budget composition (and optimized budget allocation)

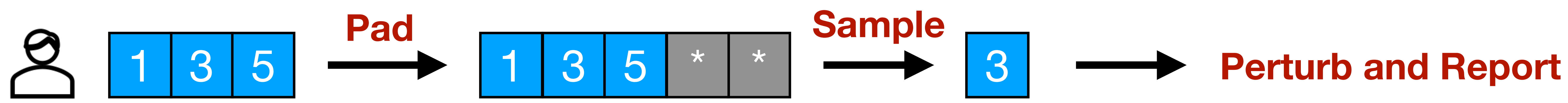
Outline

- Background of LDP
- Problem Statement and Existing Mechanism
- Our Framework: PCKV
- Experiments
- Conclusion

Overview of PCKV



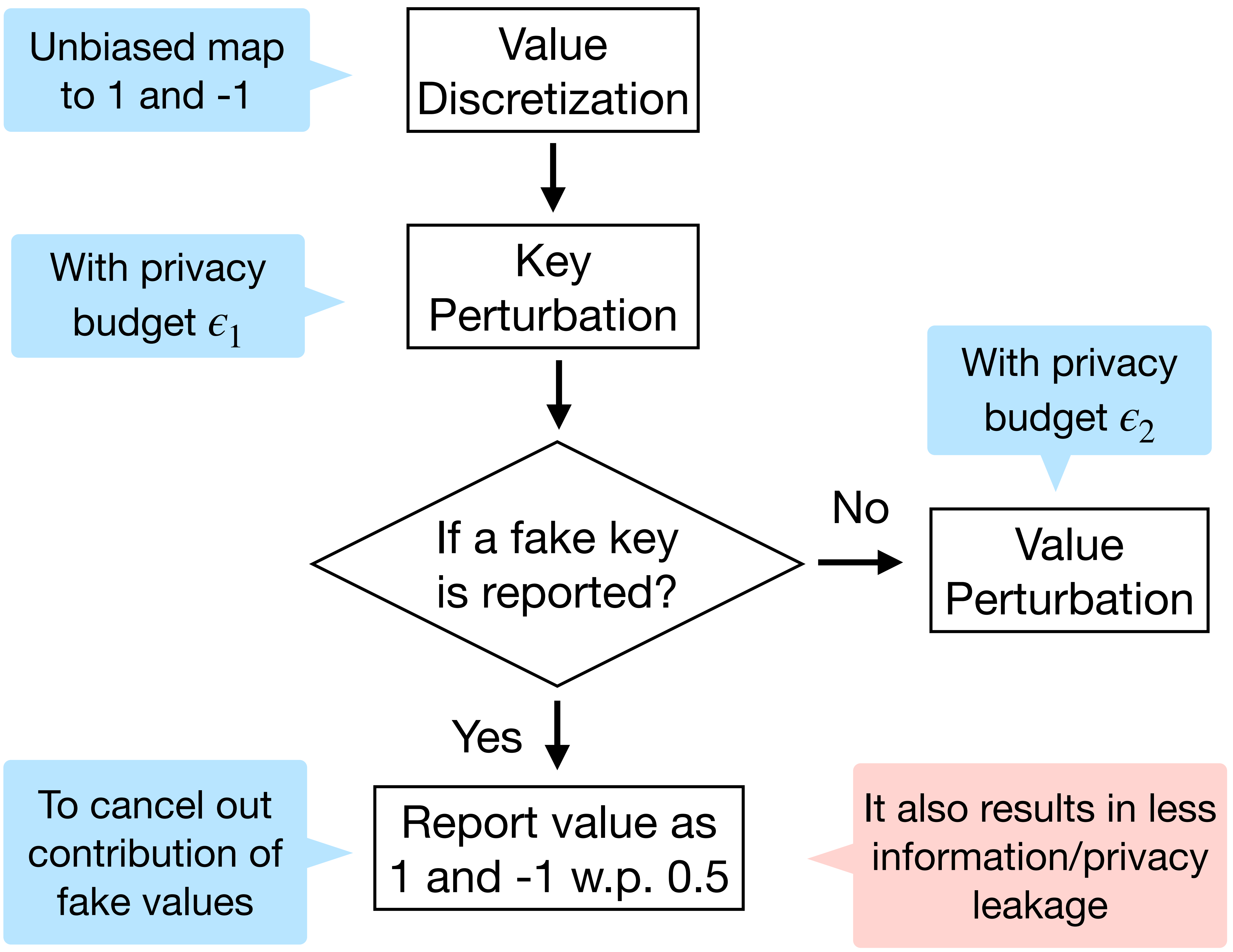
- Advanced sampling protocol: each user pads her keys into a uniform length ℓ by some dummy keys



- Joint privacy analysis: in an end-to-end way (instead of directly using sequential composition)
- Optimized allocation of ϵ_1 and ϵ_2: by minimizing MSE of estimation under tight budget composition

Perturbation and Privacy Analysis

Joint/Correlated Perturbation



Joint Privacy Analysis

The final privacy budget is less than $\epsilon_1 + \epsilon_2$

- PCKV-UE has tighter privacy budget composition than directly using sequential composition

$$\epsilon = \max\{\epsilon_2, \epsilon_1 + \ln[2/(1+e^{-\epsilon_2})]\} \leqslant \epsilon_1 + \epsilon_2$$

(because $\epsilon_1 \geqslant 0$ and $\frac{2}{1+e^{-\epsilon_2}} \leqslant e^{\epsilon_2}$)

- PCKV-GRR has similar tight budget composition and additional privacy benefit from sampling.

- PrivKVM does not have tight budget composition (because the fake value is reported with two different probabilities).

Aggregation and Estimation

- The server aggregates the supporting numbers of value 1 and -1 for the k-th key.
- Estimated frequency $\hat{f}_k$: multiplied by ℓ due to sampling, where $\mathbb{E}[\hat{f}_k] = f_k^*$

Unbiased

- Estimated mean $\hat{m}_k = \frac{\text{calibrated sum}}{\text{calibrated counts}}$, where $\mathbb{E}[\hat{m}_k] \rightarrow m_k^*$ when $n \rightarrow \infty$

Asymptotically Unbiased

- The MSEs of $\hat{f}_k$ and $\hat{m}_k$ depend on how to balance ϵ_1 and ϵ_2 under a fixed total privacy budget ϵ

Tractability of theoretical analysis

Optimized Privacy Budget Allocation

A function of ϵ_1, ϵ_2

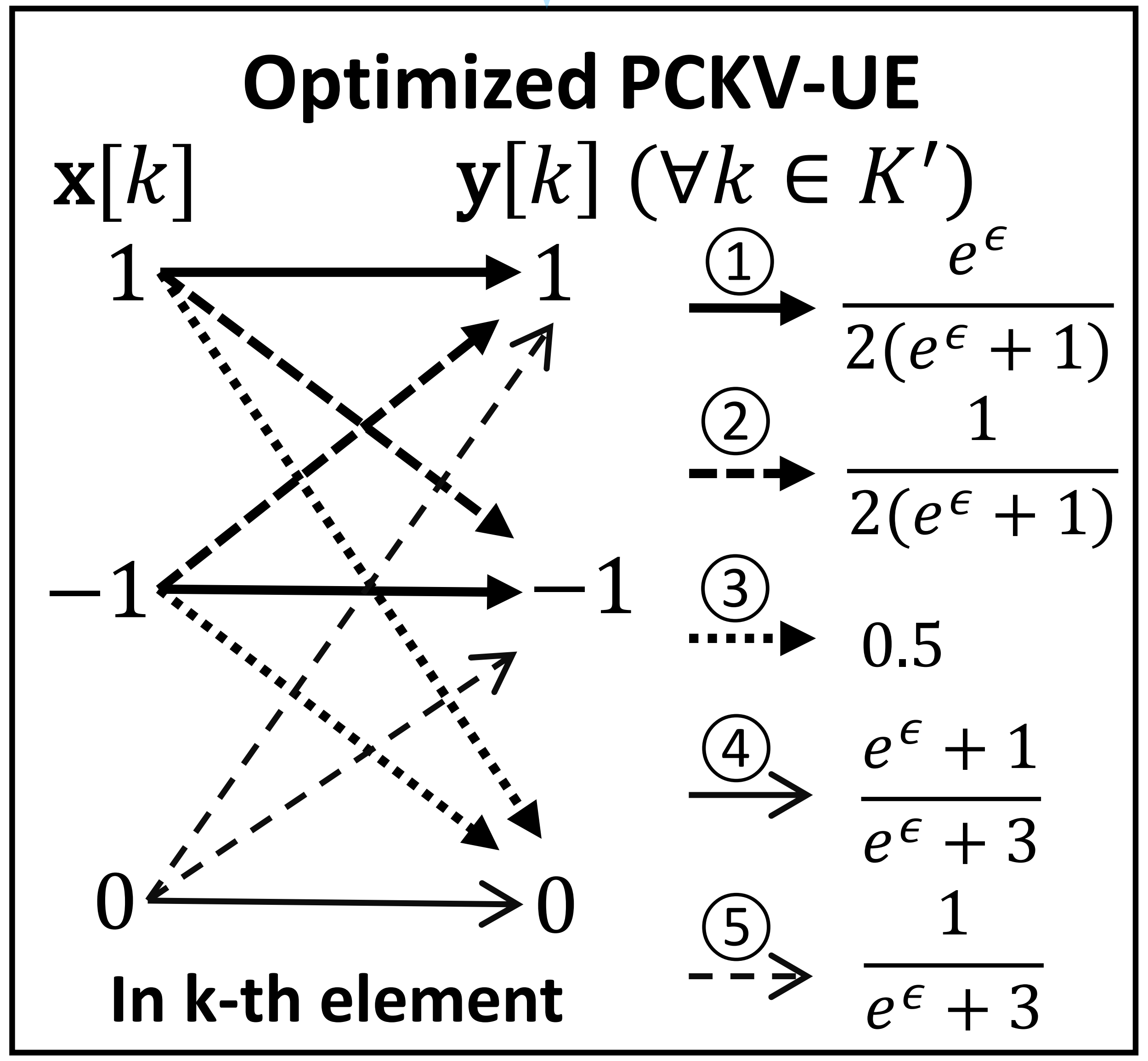
Relationship among ϵ_1, ϵ_2 and ϵ

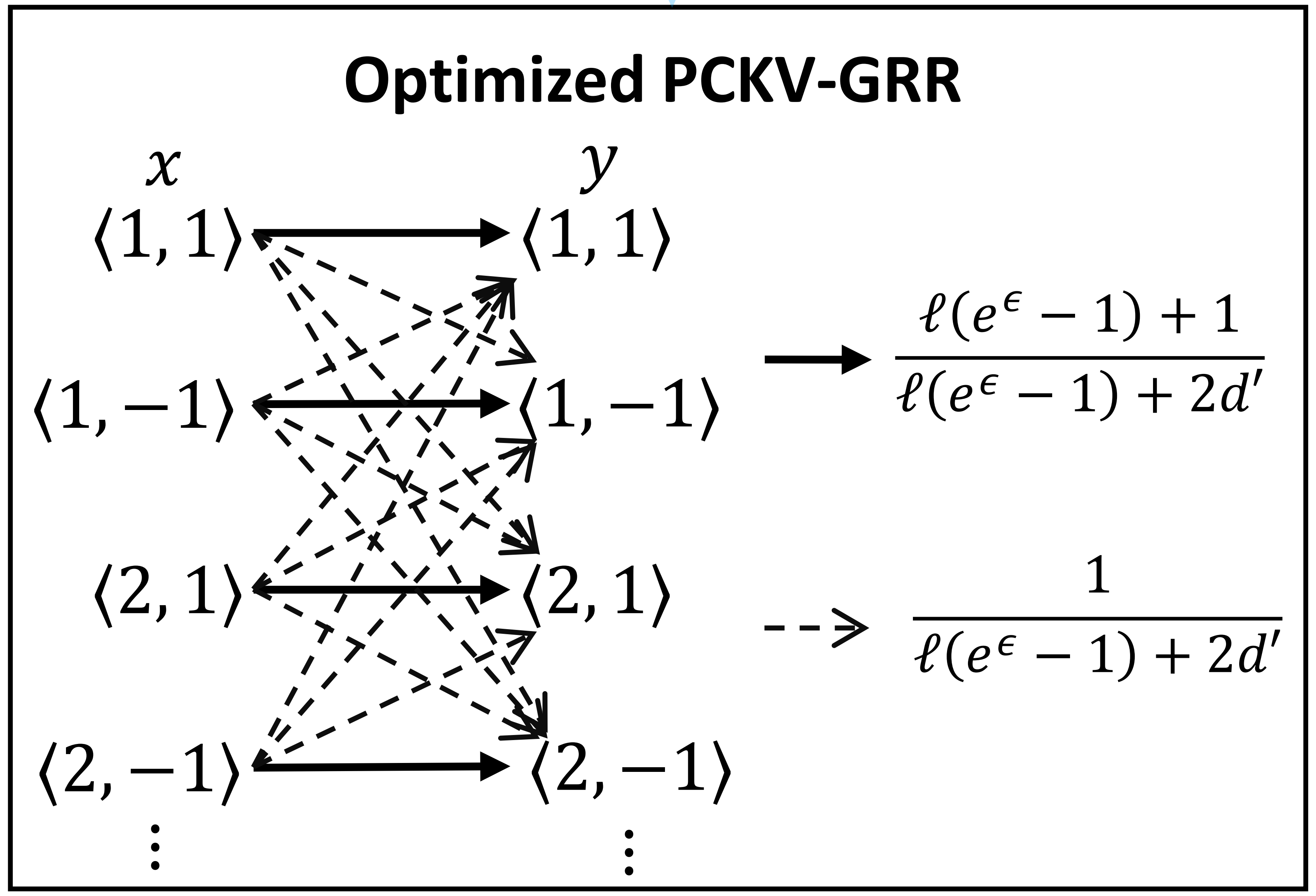
How to optimally determine ϵ_1, ϵ_2 when given ϵ

$\boxed{\min \text{ MSE}}$ + $\boxed{\text{Tight Composition}}$ $\Rightarrow$ $\boxed{\text{Optimized Allocation}}$

$\epsilon_1 = \ln[(e^\epsilon + 1)/2], \quad \epsilon_2 = \epsilon$

$\epsilon_1 = \ln[\ell \cdot (e^\epsilon - 1)/2 + 1], \quad \epsilon_2 = \ln[\ell \cdot (e^\epsilon - 1) + 1]$



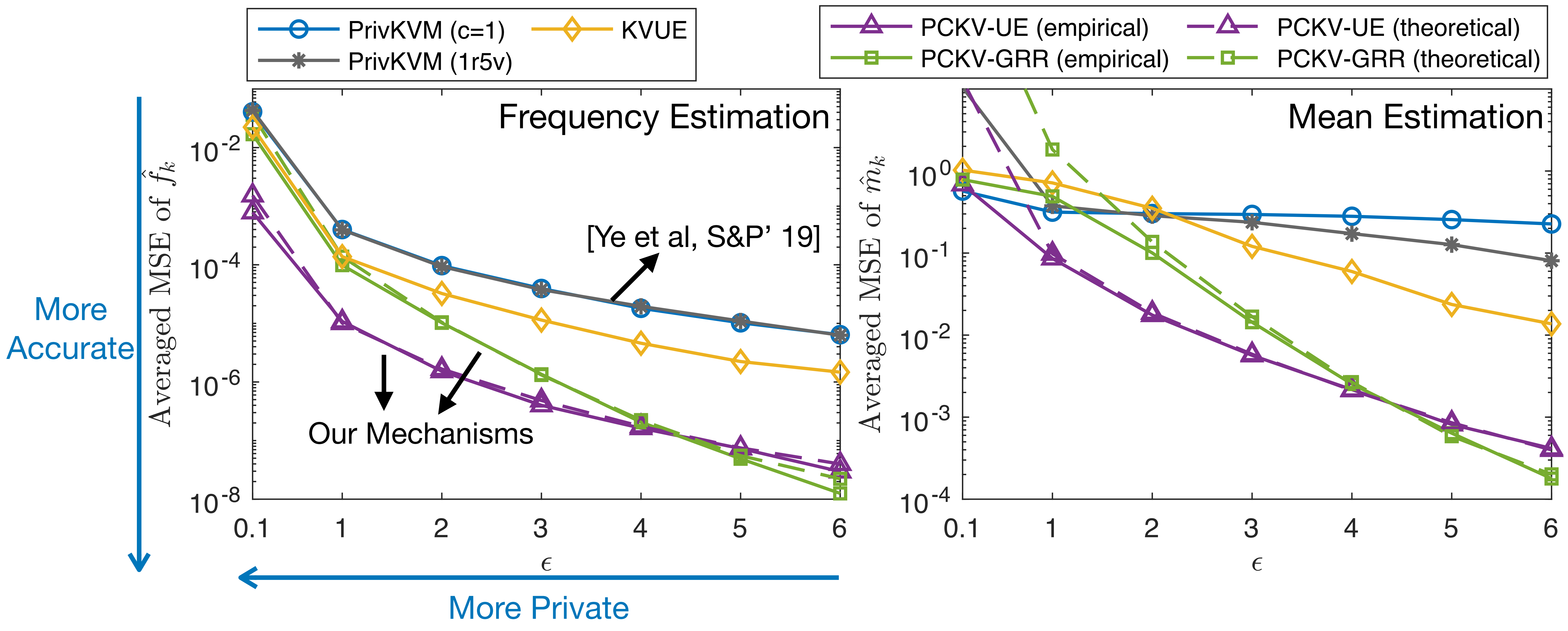


Final Perturbation (after sampling)

Summary of PCKV

- Step 1. Choose the advanced sampling protocol
- Step 2. Jointly perturb key-value and jointly analyze the privacy (which provides tight privacy budget composition)
- Step 3. Optimally put things together (i.e., optimized privacy budget allocation under a fixed total budget)

Experiments



Improvements of PCKV

- Advanced sampling protocol
- Tight budget composition
- Optimized budget allocation

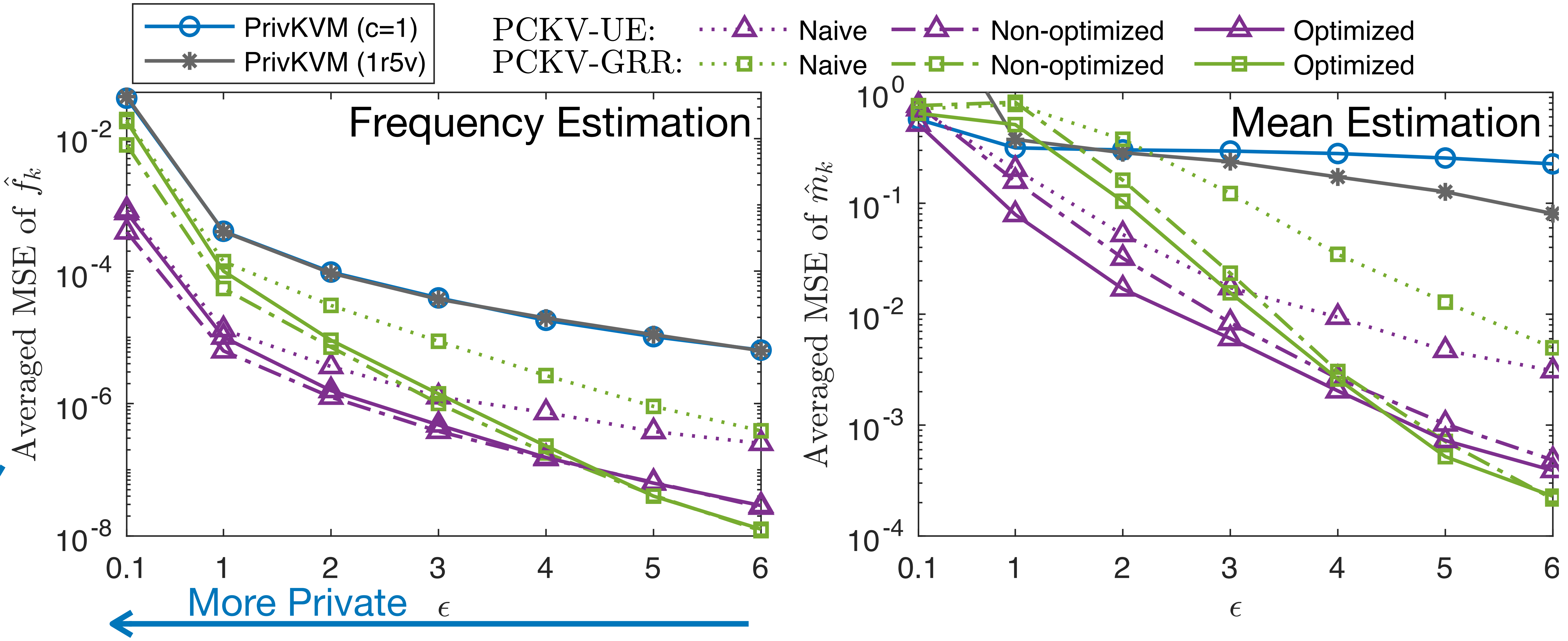
- The theoretical results close (dashed lines) to the empirical results (solid lines)
- Our mechanisms outperforms existing ones on both frequency and mean estimation

Experiments

Benefit from each improvement

- Tight Budget Composition v.s. Sequential Composition
- Optimized Budget Allocation v.s. Non-optimized

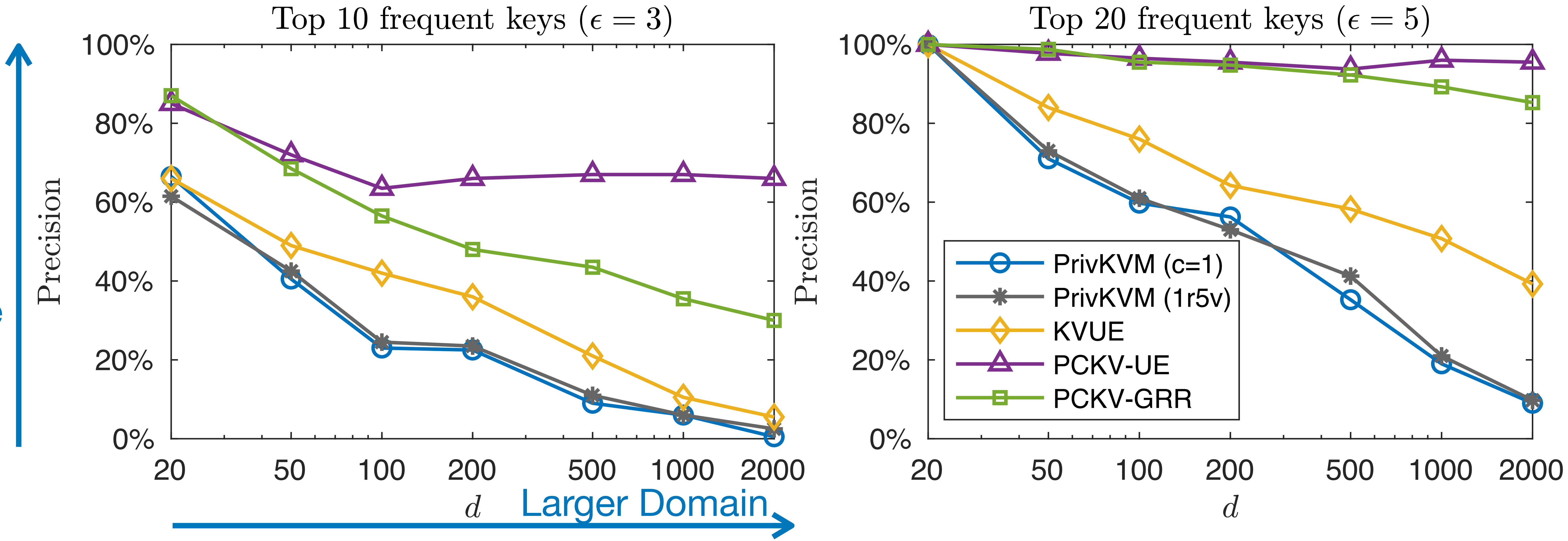
More Accurate



Success of top frequent keys identification (varying domain size)

- PCKV mechanisms outperforms other ones
- PCKV-UE has smaller impact from large domain size

More Accurate

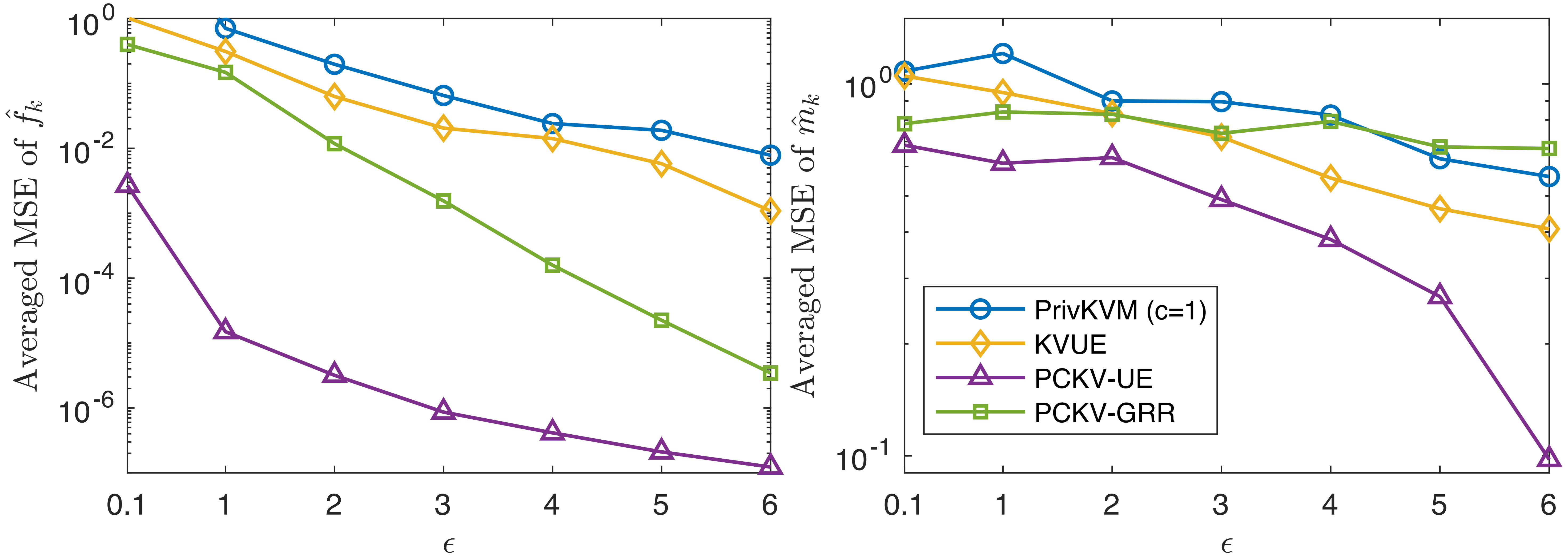


Real-world Data

Amazon Dataset

ratings: 2M
users: 1M
keys: 249K

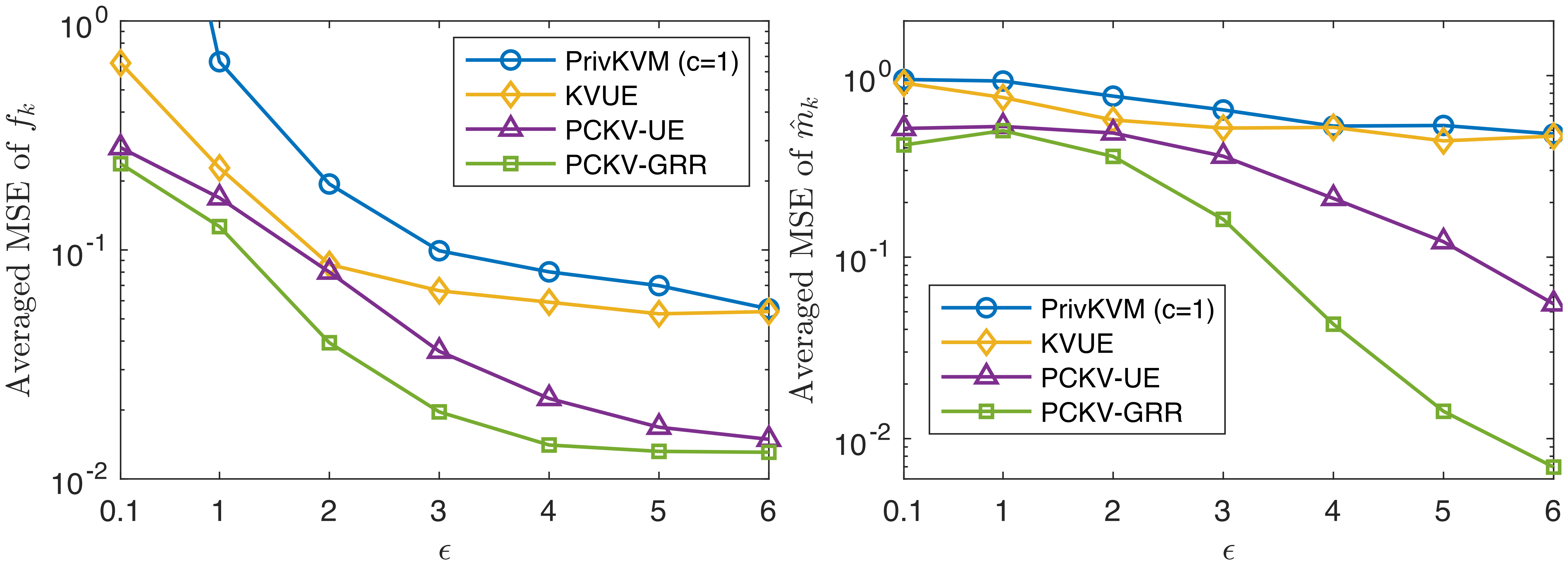
Data source: https://www.kaggle.com/skillsmuggler/amazon-ratings



Movie Dataset

ratings: 20M
users: 138K
keys: 26K

Data source: https://www.kaggle.com/ashukr/movie-rating-data



Conclusion

- The advanced sampling protocol can improve the sampling efficiency and the utility.
- Joint/correlated perturbations of key and value (rather than independent ones) can provide more options for mechanism design and the chance to choose the optimized one.
- Joint privacy analysis can lead to better privacy-utility tradeoff (because it results in tighter privacy budget composition than sequential composition)

Future work

- Study the optimized strategy of choosing ℓ in Padding-and-Sampling protocol.
- Extend the correlated perturbation and tight composition analysis to other general types of multi-dimensional data.

Thanks for your attention !

Q&A

Contact Information: xiaolang@email.arizona.edu