On the Robustness of LDP Protocols for Numerical Attributes under Data Poisoning Attacks

Xiaoguang Li*~, Zitao Li^, Ninghui Li~, and Wenhai Sun~

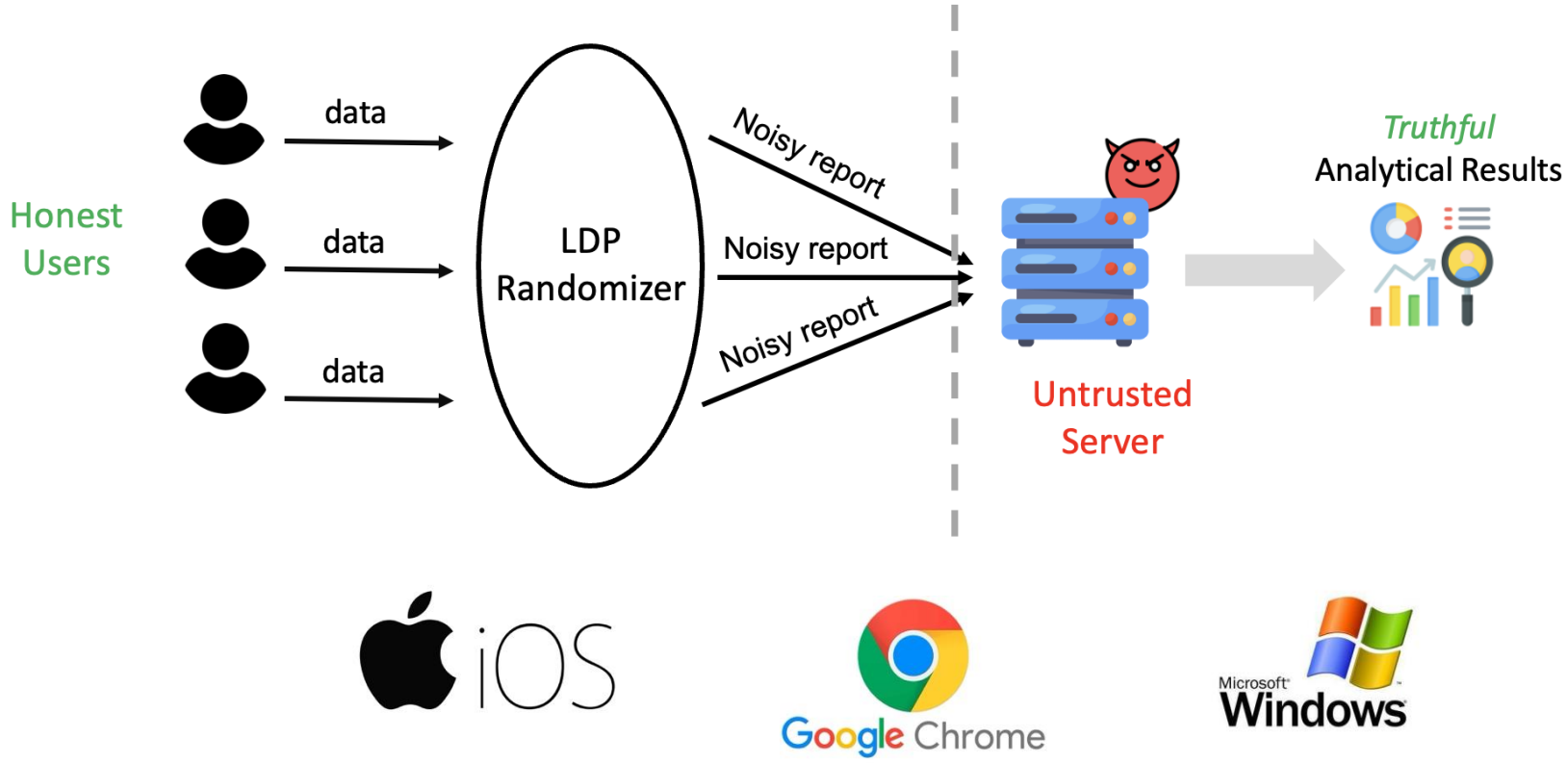
*Xidian University
^Alibaba Group (U.S.) Inc.
~Purdue University

Local Differential Privacy

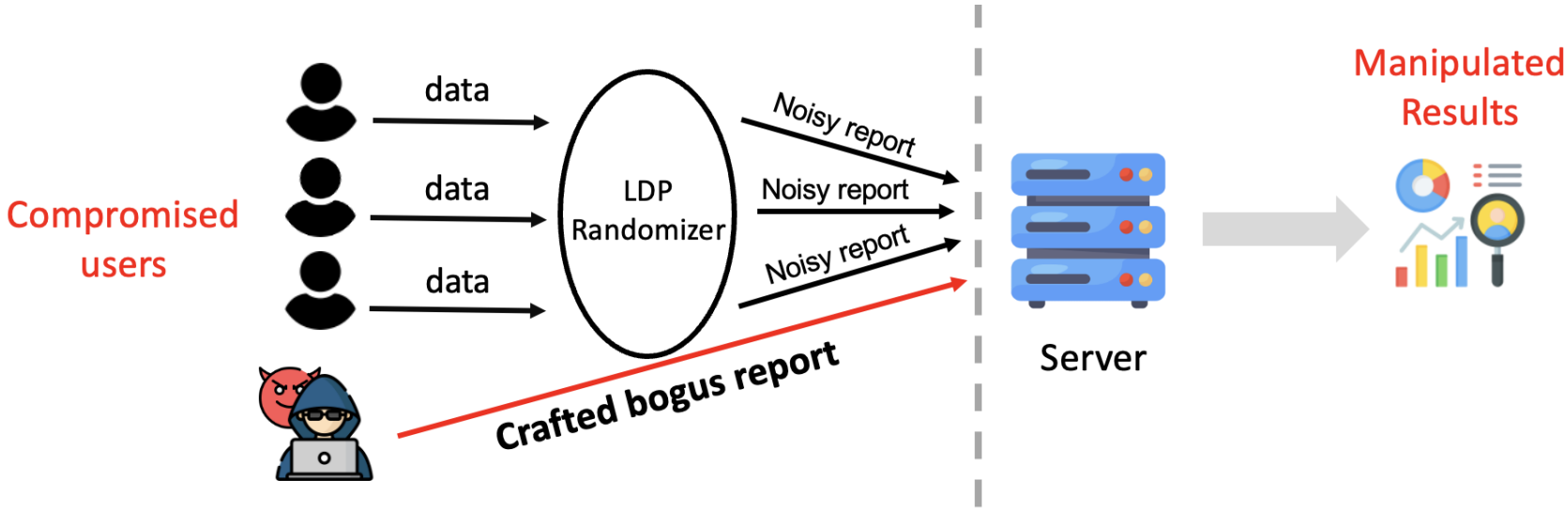
LDP [Duchi *et al*. FOCS'13]: A randomized algorithm M is ϵ-LDP if and only if

$$\Pr[M(x_1) = t] \leq e^{\epsilon} \Pr[M(x_2) = t]$$

where x_1 and x_2 are any pair of inputs in the domain.



Poisoning Attacks – A more realistic threat model



For example,

- Cao *et al.* USENIX Security'21
  - **Goal:** Promote target items
- Cheu *et al*. IEEE S&P'21
  - **Goal:** Degrade estimation accuracy
- Li *et al*. USENIX Security'23
  - **Goal:** Fine-grained result control

Profound adverse impacts on the Internet

- ❑ Undermine the mutual trust between users and service providers
- ❑ Detrimental to Internet freedom, e.g., stealthy censorship and suppression by abusing privacy-enhancing technology

Research Objectives and Contributions

Overarching goal: Understand the robustness of diverse state-of-the-art LDP protocols under data poisoning attacks

- New metrics
- Reveal new insights from protocol design to alleviate attack influence
- Enable fair comparison for protocol recommendation
- Explore effective mitigation

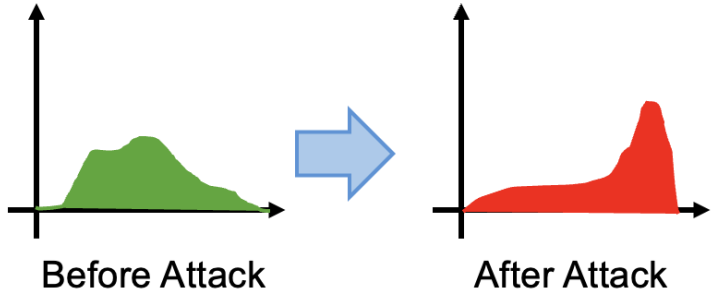
Attack-driven robustness evaluation framework

Attack Simulation

Target task: Numerical distribution

Threat model

- Attacker's capabilities
  - Compromise $\beta \in [0,1]$ of n users to inject fake data
  - Know the relevant parameters (e.g., ϵ) of LDP
  - Craft values in the output domain of LDP perturbation
- Attack goal
  - Shift the distribution to the right-most end of the domain



Baseline: Inject fake data directly in the input domain of LDP perturbation

- A universal attack
- Represent the minimum damage the attack may cause

Robustness Evaluation

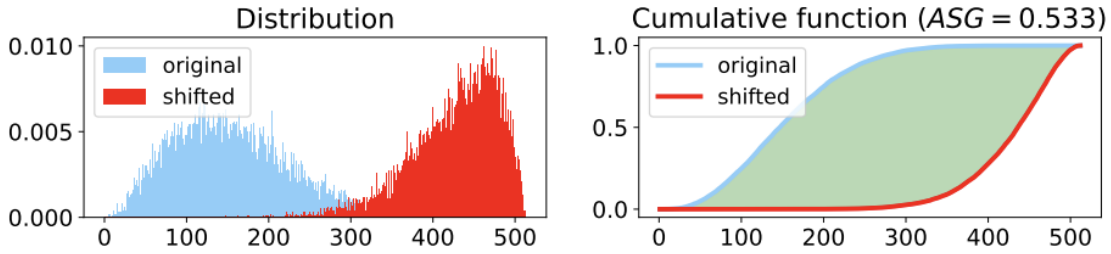
- ❖ $P(X, v)$: Cumulative distribution function over distribution X on value v
- ❖ $X, \hat{X}_a$: True distribution and estimated distribution after attack respectively

Metrics

- Absolute Shift Gain (ASG):

$$ASG(X, \hat{X}_a) = \sum_{v=1}^{m} P(X, v) - P(\hat{X}_a, v)$$

- Measure the value difference between cumulative distribution functions before and after attack



- Higher ASG indicates higher attack efficacy and lower robustness of target protocols

Limitations

- Sensitive to many factors, e.g., β and true data distribution
- Cannot indicate relative advantage compared to the baseline

Robustness Evaluation

Metrics

- Shift Gain Ratio (SGR):

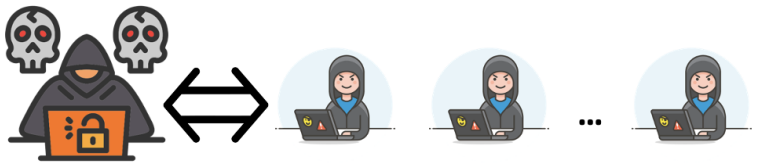
$$SGR(\hat{X}_a) = \frac{ASG(X, \hat{X}_a)}{ASG(X, X_a^{base})}$$

- Normalized by ASG of the baseline

❖ X_a^{base}: The skewed distribution estimate after baseline attack

Measure the attack efficacy at per-fake-user level

- How many fake users in the baseline is equivalent to one fake user in the proposed attack



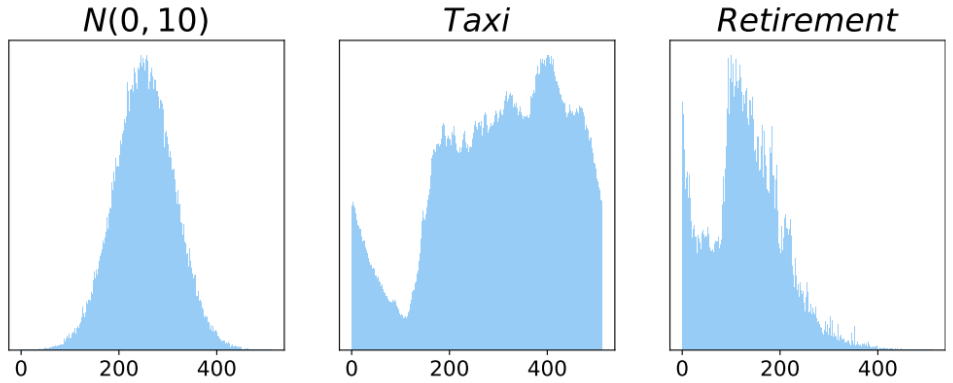
- Upper-bounded by $1/\beta$, i.e., β fake users in our attack equal to 100% fake users in the baseline
- The higher SGR, the higher attack efficacy and lower protocol robustness

Enable more meaningful robustness analysis across different LDP protocols

Experiment Setup

Datasets

- **Synthesized**: Norm
- **Real-world**: Taxi and Retirement

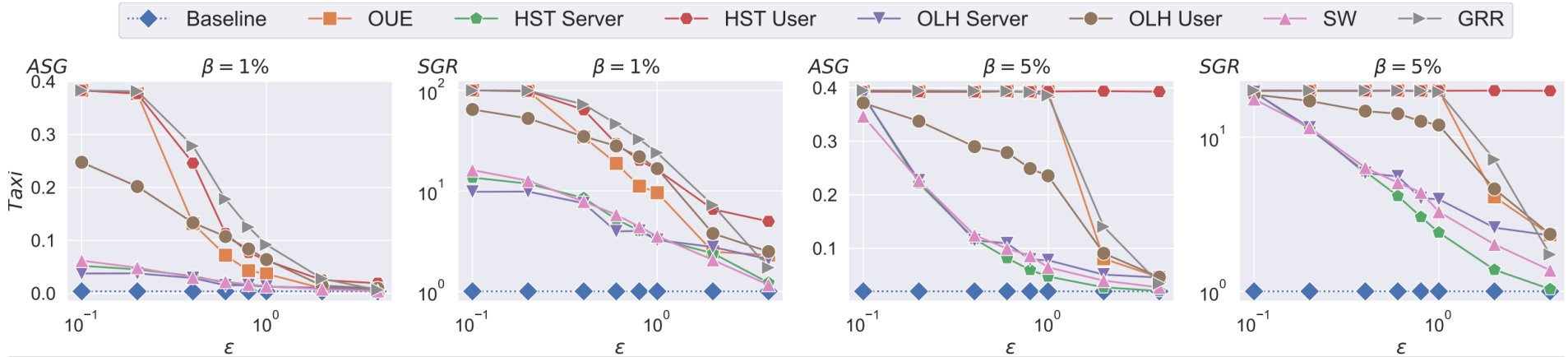


Target LDP protocols

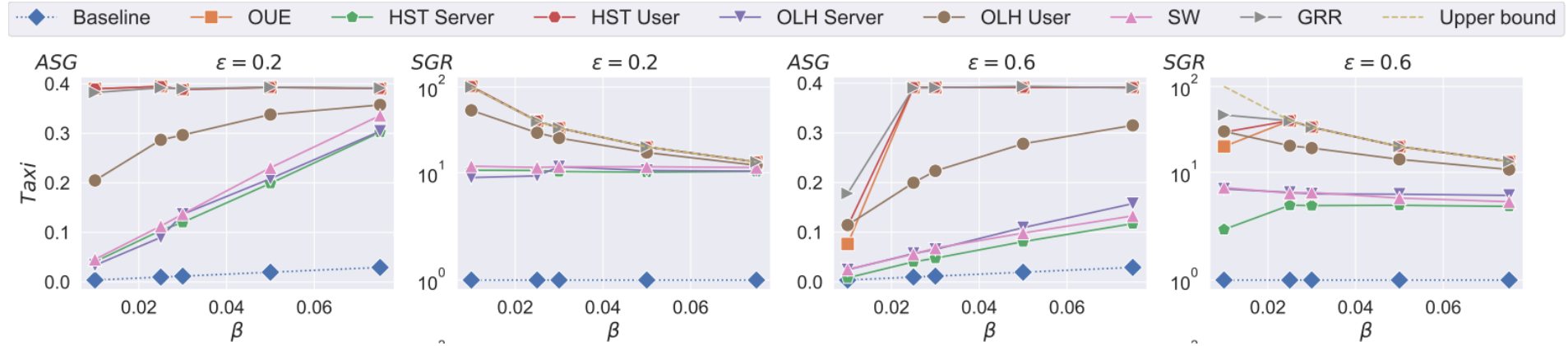
- Categorical frequency oracles (CFO) with binning
  - Direct encoding: **GRR** [Wang *et al*. USENIX Security'17]
  - Unary encoding: **OUE** [Wang *et al*. USENIX Security'17]
  - Local hashing based protocols: **HST** [Bassily *et al*. NeurIPS'17] and **OLH** [Wang *et al*. USENIX Security'17]
    - ❖ For OLH and HST, we differentiate the *Server* and *User* settings depending on who selects the hash function
- Distribution reconstruction
  - **SW** mechanism [Li *et al*. SIGMOD'20]

Experimental Results

Varying ϵ



Varying β



- SW and the *Server* setting of local hashing-based LDP protocols are the most robust
- OLH-*User* is slightly robust than HST-*User*
- GRR, OUE and HST-*User* can achieve the upper bound of SGR, indicating more vulnerable to the attack

**Refer to the paper for more results*

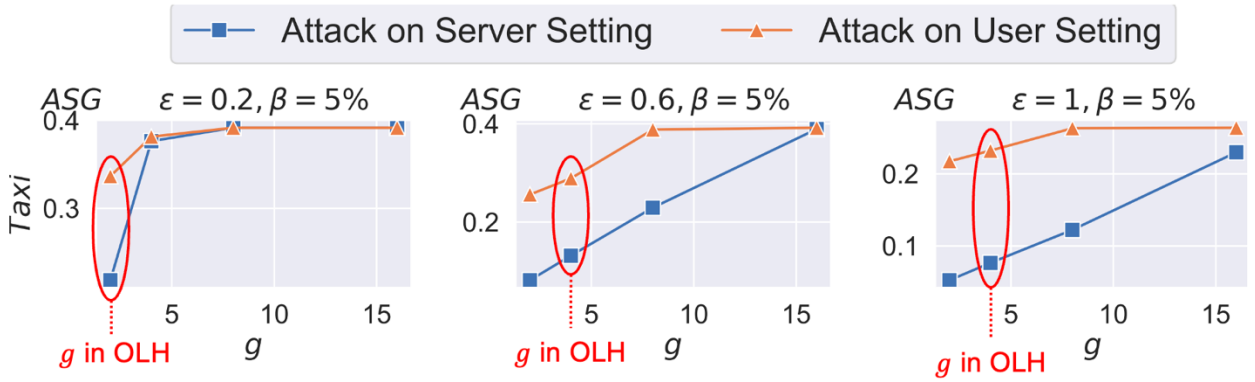
New Insights from Protocol Design into LDP Security

Prior results

- Privacy budget ϵ: Either trade-off or consistency depending on attack goals

Our findings

- The hash domain size g in OLH:
  - Small g leads to better security
  - An optimal g for utility is not always optimal for robustness



- The smoothing step in SW:
  - Diminish attack influence on target bins by averaging with adjacent bins

Zero-shot Detection

Challenges in practice

- The user data is *unknown*
- The attack strategy is *unknown*
- The method should be highly *sensitive* to data pollution

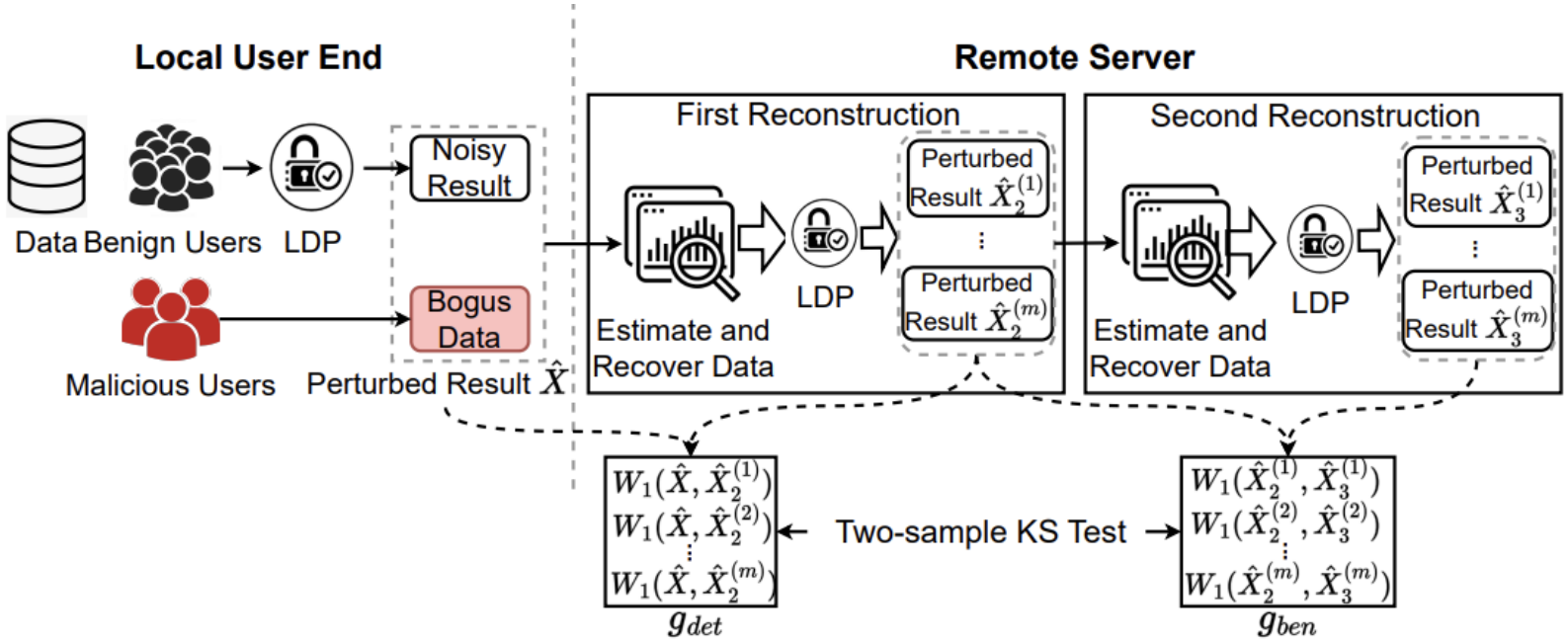
No prior work on numerical distribution

Adapt the malicious user detection (MUD) for categorical data

-- [Cao *et al.* USENIX Security'21]

- Frequent itemset mining to identify commonly supported items

Zero-shot Detection



- Randomness from bogus data is statistically different from LDP randomness
- In two-round reconstructions, measure distances between perturbed results
- g_{ben} as benchmark since no attack occurs and apply a two-sample KS test for detection

Detection Results

Metric -- Aera under the curve (AUC); A larger AUC means a better detection

Dataset	β	ϵ	HST-Server		HST-User		OLH-Server		OLH-User		OUE		GRR	
			Ours / MUD	ASG	Ours / MUD	ASG	Ours / MUD	ASG	Ours / MUD	ASG	Ours / MUD	ASG	Ours / MUD	ASG
Taxi	1%	0.2	0.4416 / –	0.0484	1.00 / –	0.391	0.476 / –	0.047	0.9952 / –	0.207	1.00 / –	0.391	1.00 / –	0.391
		0.6	0.4384 / –	0.0054	1.00 / –	0.101	0.4232 / –	0.0189	0.6224 / –	0.105	1.00 / –	0.071	1.00 / –	0.184
		1	0.3306 / –	0.008	0.9744 / –	0.058	0.392 / –	0.015	0.5784 / –	0.063	0.6808 / –	0.026	0.8992 / –	0.096
	2.5%	0.2	0.4972 / –	0.0992	1.00 / –	0.391	0.6167 / –	0.117	1.00 / –	0.285	1.00 / –	0.39	1.00 / –	0.391
		0.6	0.4872 / –	0.03432	1.00 / –	0.392	0.4933 / –	0.0407	1.00 / –	0.2012	1.00 / –	0.382	1.00 / –	0.387
		1	0.3696 / –	0.027	1.00 / –	0.344	0.4 / –	0.035	1.00 / –	0.146	0.9696 / –	0.11	1.00 / –	0.22
	5%	0.2	0.555 / –	0.18	1.00 / –	0.3912	0.6504 / –	0.204	1.00 / –	0.3353	1.00 / –	0.391	1.00 / –	0.391
		0.6	0.5352 / –	0.072	1.00 / –	0.391	0.5392 / –	0.106	1.00 / –	0.277	1.00 / –	0.391	1.00 / –	0.391
		1	0.4976 / –	0.048	1.00 / –	0.389	0.4844 / –	0.0771	1.00 / –	0.23	1.00 / –	0.389	1.00 / –	0.385
	7.5%	0.2	0.6211 / –	0.29	1.00 / –	0.391	0.7411 / –	0.31	1.00 / –	0.357	1.00 / –	0.392	1.00 / –	0.391
		0.6	0.6089 / –	0.108	1.00 / –	0.39	0.6494 / –	0.15	1.00 / –	0.31	1.00 / –	0.391	1.00 / –	0.391
		1	0.5667 / –	0.073	1.00 / –	0.391	0.5583 / –	0.11	1.00 / –	0.27	1.00 / –	0.391	1.00 / –	0.391
	10%	0.2	0.8432 / 0.575	0.382	1.00 / 0.575	0.392	0.7912 / 0.55	0.377	1.00 / 0.55	0.37	1.00 / 0.55	0.392	1.00 / –	0.391
		0.6	0.62 / –	0.15	1.00 / –	0.39	0.63 / –	0.207	1.00 / –	0.335	1.00 / –	0.39	1.00 / –	0.391
		1	0.5072 / –	0.097	1.00 / –	0.39	0.5338 / –	0.15	1.00 / –	0.303	1.00 / –	0.391	1.00 / –	0.391

- Our detection outperforms the existing method
- The detection shows better results with growing β and decreasing ϵ
- AUC is relatively low for Server setting

**Refer to the paper for more results*

What's Next?

Robust protocol design

- Security should be considered for LDP design in addition to privacy and utility
- Our metrics could help with relevant robustness analysis

Systematic exploration on defense

- Diversifying detection perspectives, e.g., fake users and overall anomaly behavior
- Effective recovery schemes for corrupted data collection
  - ❑ Attack-aware LDP post-processing

Conclusion

We designed a robustness evaluation framework and studied state-of-the-art LDP protocols for distribution estimation

SW and CFO with binning under Server setting are preferred against data poisoning attacks

We revealed new factors relating to LDP security, i.e., g in OLH and the smoothing in SW

We proposed a novel effective zero-shot detection method