POSTER: MURS: Practical and Robust Privacy Amplification with Multi-Party Differential Privacy

Tianhao Wang*, Min Xu†, Bolin Ding‡, Jingren Zhou‡, Cheng Hong‡, Zhicong Huang‡, Ninghui Li*, Somesh Jha§
*Purdue University, †University of Chicago, ‡Alibaba, §University of Wisconsin-Madison

***Abstract*—When collecting information, local differential privacy (LDP) alleviates privacy concerns of users because their private information is randomized before being sent to the central aggregator. However, LDP results in loss of utility due to the amount of noise that is added to each individual data item. To address this issue, recent work introduced an intermediate server with the assumption that this intermediate server did not collude with the aggregator. Using this trust model, one can add less noise to achieve the same privacy guarantee; thus improving the utility.**

In this paper, we investigate this multiple-party setting of LDP. We first analyze the threat model and identify potential adversaries. We then make observations about existing approaches and propose new techniques that achieve a better privacy-utility tradeoff than existing ones. Finally, we perform experiments to compare different methods and demonstrate the benefits of using our proposed method.

I. Introduction

To protect data privacy in the context of data publishing, the concept of differential privacy (DP) has been proposed, and has been widely adopted [8]. DP mechanisms add noise to the aggregated result such that the difference between whether or not an individual is included in the data is bounded. Recently, local differential privacy (LDP) has been deployed by industry. LDP differs from DP in that random noise is added by each user before sending the data to the central server. Thus, users do not need to rely on the trustworthiness of the organization hosting the server. This desirable feature of LDP has led to wider deployment by industry (e.g., by Google [10], Apple [1], Microsoft [7], and Alibaba [11]). Meanwhile, DP is still deployed in settings where the centralized server can be trusted (e.g., the US Census Bureau deployed DP for the 2020 census [2]).

However, removing the trusted central party comes at the cost of utility. Since every user adds some independently generated noise, the effect of noise adds up when aggregating the result. While noise of scale (standard deviation) $\Theta(1)$ suffices for DP, LDP has noise of scale $\Theta(\sqrt{n})$ on the aggregated result (n is the number of users). This gap is essential for eliminating the trust in the centralized server, and cannot be removed by algorithmic improvements [4].

Recently, researchers introduced settings where one can achieve a middle ground between DP and LDP, in terms of both privacy and utility. This is achieved by introducing an additional party [5], [9], [3], [6]. One such setting is called the *shuffler model*, which introduces another party called the shuffler. Users perturb their information, and then send encrypted version of the perturbed information to the shuffler, who shuffles the users' information, and then sends them to the server. The server then decrypts the reports and aggregates the information. The shuffler learns nothing about the reported data (because of semantic-security of the encryption scheme), and the server learns less about each individual's report because it cannot link a user to a report because the user inputs it received are shuffled. However, if the shuffler and the server collude, the user obtains privacy protection only from perturbation, and there is no benefit from shuffling. In short, the role of the shuffler is to break the linkage between the users and the reports, thus providing some privacy boost. Due to the privacy boost, users can add less noise, while achieving the same level of privacy against the server. This boost, however, requires trusting that the shuffler will not collude with the server. This new model of LDP, which we call Multi-Party DP (MPDP), offers a different trade-off between trust and utility than DP and LDP.

Besides the shuffler-based approach, in the MPDP model, there is also another interesting direction that uses homomorphic encryption [6]. In particular, each user homomorphically encrypts his/her value using one-hot encoding. The additional server then multiplies the ciphertexts in each location to get the aggregated result (i.e., a histogram), and adds noise to the histogram to provide DP guarantee. Finally, the results are sent to the server. As homomorphic encryption requires one-hot encoding, the communication cost for this approach is can be large for big domains.

Since the MPDP model involves more parties, there could be different patterns of interaction and collusion among the parties. The possibilities of these colluding parties and the consequences have not been systematically analyzed. For example, existing work proves the privacy boost obtained by shuffling under the assumption that the adversary observes the shuffled reports and knows the input values of the users (except the victim). However, if the other users collude with the adversary, they could also provide their locally perturbed reports, invalidating any privacy boost due to shuffling. For another example, while the homomorphic encryption-based approach provides privacy guaranteed when the adversary consists of the server colluding with the users (except the victim), there is no privacy when the server colluding with the additional server. In this paper, we analyze the interaction and potential threats of the MPDP models in more detail. We present a unified view of privacy that generalizes DP and LDP. Different parties and possible colluding scenarios are then presented and analyzed.

Based on our observations, we propose MURS (stands for Multi Uniform Random Shufflers). MURS adopts the shuffler-based approach [5], [9], [3], as its communication cost is small for large domains. But different from existing work,

MURS introduces multiple shufflers and have them add noise. Moreover, we proposes a new mechanism that performs orders of magnitude better than existing work.

More specifically, we show that the essence of the privacy boosting [3] is a distribution from the LDP mechanism that is independent of the input value. By revisiting the local hashing idea, which was also considered in the LDP setting, we then set the independent distribution using the hashing idea and propose symmetric local hash (SLH). In SLH, each user reports a randomly selected hash function, together with a perturbation of the hashed result of their sensitive value.

Furthermore, from the observation of our systematic analysis of the MPDP model, we propose to have the additional server also introduce noise. In particular, the additional server, besides shuffling the received reports, adds some uniformly random reports so that when all other users collude with the server, there is still some privacy guarantee. We also suggest having more additional servers, which mitigates the threat when a single additional server colludes with the central server. As long as not all of the additional servers collude with the server, the boosted privacy guarantee still holds.

To summarize, the main contributions of this paper are:

- We provide a systematic analysis of the MPDP model and a principled way for analyzing privacy guarantees under various colluding scenarios. Several observations are made, which leads to the proposal of MURS.
- We instantiate MURS with two protocols: MURSS via onion encryption and MURSO via oblivious shuffling. MURS comes from three components: (1) theoretical improvement; (2) thorough analysis of MPDP; and (3) novel design of existing ideas. Compared with existing work, both protocols provide better trust guarantee and achieve better utility-privacy trade-off.
- We provide implementation details and measure utility and performance of MURSS and MURSO on real datasets. Moreover, we will open source our implementation so that other researchers can build on our results.

REFERENCES

[1] "Apple differential privacy team, learning with privacy at scale," available at https://machinelearning.apple.com/docs/learning-with-privacy-at-scale/appledifferentialprivacysystem.pdf.

[2] J. M. Abowd, "Protecting the confidentiality of america's statistics: Adopting modern disclosure avoidance methods at the census bureau," https://www.census.gov/newsroom/blogs/research-matters/2018/08/protecting_the_confi.html, 2018.

[3] B. Balle, J. Bell, A. Gascon, and K. Nissim, "The privacy blanket of the shuffle model," in CRYPTO, 2019.

[4] T. H. Chan, K.-M. Chung, B. M. Maggs, and E. Shi, "Foundations of differentially oblivious algorithms," in SODA. SIAM, 2019, pp. 2448–2467.

[5] A. Cheu, A. D. Smith, J. Ullman, D. Zeber, and M. Zhilyaev, "Distributed differential privacy via shuffling," in EUROCRYPT, 2019, pp. 375–403.

[6] A. R. Chowdhury, C. Wang, X. He, A. Machanavajjhala, and S. Jha, "Cryptε: Crypto-assisted differential privacy on untrusted servers," SIGMOD, 2019.

[7] B. Ding, J. Kulkarni, and S. Yekhanin, "Collecting telemetry data privately," in Advances in Neural Information Processing Systems, 2017, pp. 3574–3583.

[8] C. Dwork, "Differential privacy," in ICALP, 2006, pp. 1–12.

[9] Ú. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, K. Talwar, and A. Thakurta, "Amplification by shuffling: From local to central differential privacy via anonymity," in SODA, 2019, pp. 2468–2479.

[10] Ú. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in CCS. ACM, 2014, pp. 1054–1067.

[11] T. Wang, B. Ding, J. Zhou, C. Hong, Z. Huang, N. Li, and S. Jha, "Answering multi-dimensional analytical queries under local differential privacy," in SIGMOD, 2019.

MURS: Practical and Robust Privacy Amplification with Multi-Party Differential Privacy

Tianhao Wang, Min Xu, Bolin Ding, Jingren Zhou, Cheng Hong, Zhicong Huang, Ninghui Li, Somesh Jha

Purdue University, University of Chicago, Alibaba Group, University of Wisconsin-Madison

Background

♠ **Multi-Party Differential Privacy: Better trust than DP and better utility than LDP**

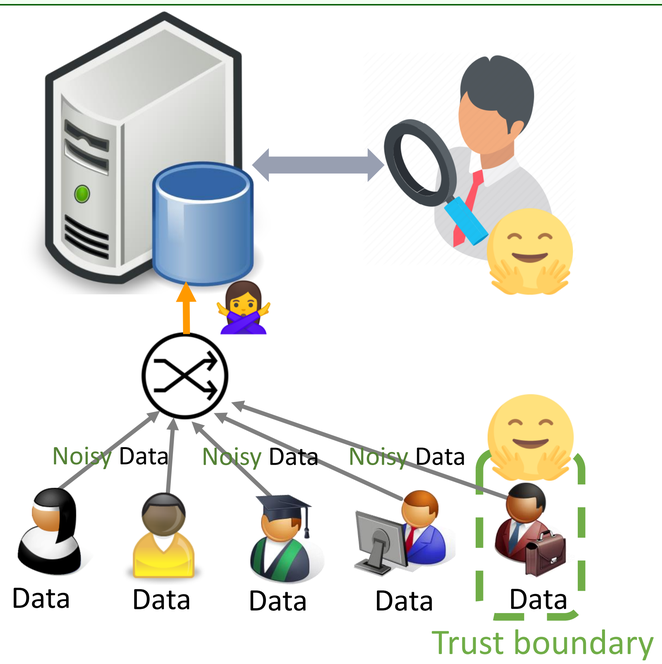

Figure 1: Multi-Party Differential Privacy System Model

♠ **Existing work either provides poor utility [5, 3, 2, 6] or high communication overhead [1, 4].**

♠ **The system model in Figure 1 is weak.**

System Analysis of MPDP

♠ **The server colluding with all other users.**

♠ **The server with $t \geq 1$ auxiliary servers.**

♠ **The auxiliary servers may poison the result.**

SLH

♠ **Utilizing the Local Hashing idea, we improve the utility of existing work.**

♠ **Each user execute:**

$$\mathsf{SLH}_{\epsilon,d}(x) = \langle H,\ \mathsf{GRR}_{\epsilon,d'}(H(x)) \rangle$$

where $d' = e^{\epsilon/2} + 1$**, and**

$$\forall_{x,y \in D}\ \Pr\left[\mathsf{GRR}_{\epsilon,d}(x) = y\right] = \begin{cases} p = \frac{e^{\epsilon}}{e^{\epsilon}+d-1}, & \textbf{if } y = x \\ q = \frac{1}{e^{\epsilon}+d-1}, & \textbf{if } y \neq x \end{cases}$$

MURS

♠ **Utilizing Onion Encryption and Oblivious Shuffle, we improve the threat model of existing work.**

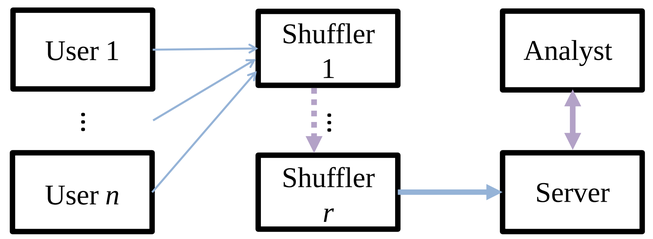

Figure 2: System Model of MURSS: Sequential Shuffle.

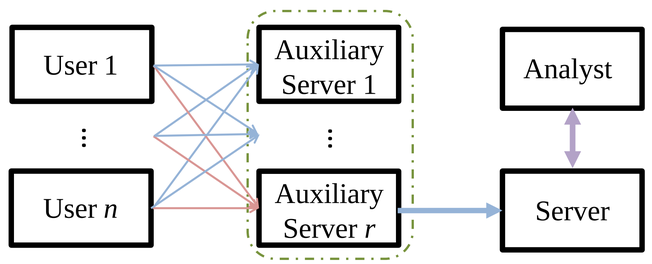

Figure 3: System Model of MURSO: Oblivious Shuffle.

Evaluation Results

♠ **Reasonable communication and computation overhead.**

♠ **Utility (mean absolute error) is orders of magnitudes better.**

Metric \ Method	MURSS $r=3$	MURSS $r=7$	MURSO $r=3$	MURSO $r=7$
User comp. (ms)	21	50	1.6	1.6
User comm. (Byte)	416	800	400	432
Aux. comp. (s)	213	214	0.2	0.7
Aux. comm. (MB)	224	416	429.8	3293.3
Server comp. (s)	213	213	65	65
Server comm. (MB)	128	128	392	408

Table 1: Computation and communication overhead of MURSS and MURSO for each user, each auxiliary server, and the server. We assume $n = 10^6$ and $r = 3$ or 7.

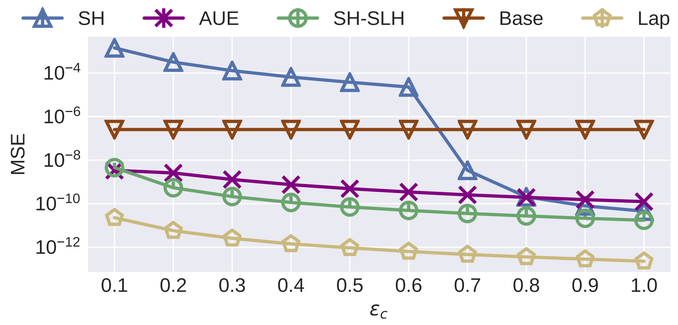

Figure 4: Utility Comparison. SH is from [2], AUE is from [1] (but communication cost is much larger), Base is uniform guess, and Lap is centralized DP.

References

[1] Victor Balcer and Albert Cheu. Separating local & shuffled differential privacy via histograms. *arXiv*, 2019.

[2] Borja Balle, James Bell, Adria Gascon, and Kobbi Nissim. The privacy blanket of the shuffle model. In *CRYPTO*, 2019.

[3] Albert Cheu, Adam D. Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In *EUROCRYPT*, pages 375–403, 2019.

[4] Amrita Roy Chowdhury, Chenghong Wang, Xi He, Ashwin Machanavajjhala, and Somesh Jha. Cryptε: Crypto-assisted differential privacy on untrusted servers. *SIGMOD*, 2019.

[5] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In *SODA*, pages 2468–2479, 2019.

[6] Badih Ghazi, Noah Golowich, Ravi Kumar, Rasmus Pagh, and Ameya Velingker. Private heavy hitters and range queries in the shuffled model. *arXiv*, 2019.