[1]Vipin Vijayachandran

Dr. R. Suchithra[2]

Adaptive Local Differential Privacy Approach for Privacy-Preserving Machine Learning

Abstract: - We live in a world where data is being generated and collected from IoT devices, browsers, applications, and mobile devices continuously, and the pressing demand for data processing techniques that preserve the privacy of individuals is growing day by day. Optimal approaches to preserving privacy vary based on applications and requirements, and different methods include anonymization, homomorphic encryption, and differential privacy. Differential privacy is considered one of the methods to ensure the privacy of the individuals in a data set. This research explores the application of local differential privacy in machine learning, how to adjust utility vs. trade-off based on the importance of features in the dataset, and the feedback mechanism from the aggregator for setting the privacy budget. The research aims to develop a sample model using the adaptive differential privacy method by utilizing feedback from the aggregator. We propose the AdaDPriv approach, which involves a meticulous feature selection process and the addition of noise to minimize privacy vs. utility trade-off and an assessment of its advantages in practical scenarios. The result suggests that the methods provide plausible deniability for the individual when there is a data leak, allowing the analysis to derive almost similar outputs as the original dataset.

Keywords: differential privacy, local differential privacy, adaptive budget, budget feedback mechanism.

1 Introduction

In general terms, privacy in data refers to protecting an individual's personal information by controlling access to the data and sharing information only with the previously agreed-upon parties and only to the correct amount as required for the services. Each individual has the right over their data and to decide to what extent it is stored, utilized, and shared with a third party. Each individual, culture, and generation has different views about privacy, which is more subjective and depends on the laws of each country. In the modern world, it is essential to share the data for getting the services from the service providers, and service providers collect this information to provide their services, targetted sales and advertisement, and data analysis and insight generation. Sometimes this information gets shared with a third party, leaked, or hacked, and the confidential information becomes publicly available. Encryption and anonymization are not often enough to secure the user's data. One important aspect of data analysis is that we often have to share data between individuals and companies to derive meaningful insights and provide services. Differential privacy algorithms use controlled random noise to perturb the data and protect the individual's identity while allowing meaningful data analysis. Differential privacy (DP) is a mathematical framework for preserving privacy during data analysis and sharing. Differential privacy is widely used in healthcare, finance, and social media. DP methods introduce noise to the dataset in a predetermined amount to preserve an individual's privacy by providing plausible deniability in the event of a data leak. While we need to preserve privacy, it is also essential to analyze data to find trends and insights, and we also need to train machine learning and deep learning models on the dataset for various reasons. One significant aspect of differential privacy is that the privacy guarantee holds even when the attacker has extra background information or additional data. Simultaneously, it preserves the data's utility to derive valuable insights.

2 Preliminaries

Differential privacy enables us to learn useful information from the data without identifying the specific individuals in the dataset. Let us say a person needs to have powered eyeglasses to identify an object at a distance. His objective is to identify cars. With glass, he can identify the vehicle, the license plate number, and the driver sitting inside. Without the glass, the car may appear blurry, and he may not read the vehicle's license plate. But he can still understand the color of the car, the type of the car, and probably the model and make of the vehicle. Now

[1 1*]Department of Computer Science, Jain University, Bangalore, India.

[2*]Department of Computer Science, Presidency College, Bangalore, India.

*Corresponding author(s). E-mail(s): vipinvx@gmail.com;

Contributing authors: suchithra.suriya@gmail.com;

imagine that the car is a bit farther away. This time, he can still understand that it is a car, but he cannot understand the make and model. When we increase the distance, the person cannot understand that it is a car. At this point, the data cannot be used for analysis. Here, we can consider the distance as noise. When more noise is added, the data becomes more blurry. One can add noise to the amount that is useful for the requirement. After a limit, data is not useful anymore. This is called the privacy vs. utility trade-off. Differential privacy mechanisms work like this. It uses carefully selected amounts of noise to ensure that the presence or absence of any individual record in the dataset does not significantly affect the algorithm's outcome, making it difficult for an attacker to infer information about a specific individual. By satisfying the differential privacy definition, an algorithm ensures that the privacy of individuals is protected while allowing for valuable insights to be drawn from the dataset.

Definition

A randomized algorithm M gives ϵ-differential privacy if, for any two datasets $D1$ and $D2$, and any subset S of the output space, the following inequality holds [1]

$$P(M(D1) \in S) \leq \exp(\epsilon) \times P(M(D2) \in S) \quad (1)$$

where M is a Randomized algorithm that takes a dataset as input, and ϵ is the privacy budget. The privacy budget indicated by ((epsilon)) in this case determines the level of privacy. A lower value provides better privacy, while a higher value provides lower privacy but improved accuracy in the analysis.

Advantageous

Differential privacy offers a theoretically proven mathematical guarantee of the privacy of the individuals in a large dataset, and the guarantee holds even when the attacker knows about the background information of the individuals in the dataset. Although the privacy vs. utility trade-off exists, users can carefully select the parameters to choose the level of privacy they need. A larger ϵ indicates less privacy, while a smaller ϵ indicates a strong privacy guarantee. This provides more flexibility for use and also helps in wide adaptability in different areas such as healthcare, retail, social media, and advertising, among others; it can also help the organization meet legal requirements like GDPR due to the mathematical guarantee it offers on the privacy of the individual and the agencies can validate the risks quantitatively. It can help share the data without worrying about privacy, resulting in better coordination and public trust. The algorithms are often not compute intensive like other methods like encryption, which makes them suitable for use in IoT devices, edge devices, and large datasets.

Disadvantageous

The differential is vulnerable to composition or reconstruction attacks. Suppose differential privacy is used while retrieving data from a dataset dynamically. In that case, several queries can be run against the differentially private dataset to identify the actual values of specific rows, and these results can be combined with data from different sources to identify the individuals. Another disadvantage is the utility vs. privacy tradeoff. One has to select ϵ by carefully considering the privacy and domainspecific requirements. There is a lack of standardization across standards to compare one method to another in an industrial use case. Algorithms are efficient in the case of smaller datasets. Still, when the data grows to millions and billions of rows, adding noise on the go becomes computationally tricky, resulting in low response time.

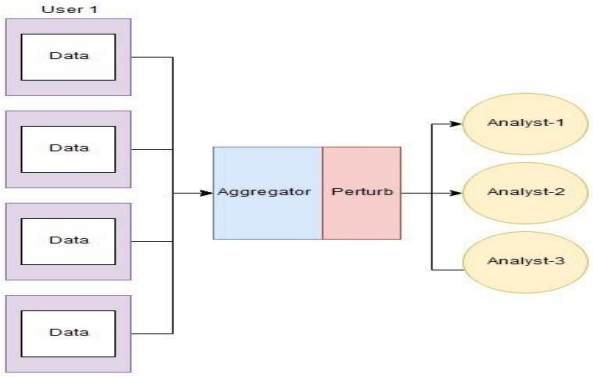

Fig. 1 *GlobalDifferentialPrivacy*

Global vs. Local Differential Privacy

Differential privacy can be broadly categorized into global differential privacy (GDP) and local differential privacy (LDP). In global differential privacy, the data sources of data generators trust a curator. The curator securely stores the data and applies differential privacy while retrieving the data for analysis. Here, the actual data is already with the curator. Still, the curator has decided to use the data responsibly and share it with third parties or analysts by applying the DP techniques. Data from different sources, users, or other entities is often combined for the analysis. Because the noise is calculated and used globally with more data at hand, the utility of the data can be preserved. Since the calculations are done at the aggregator level with better computational resources, more complex algorithms can be used to protect privacy. The downside is that the clients must trust the curator and can never be sure that the data is used and shared responsibly. Other than the guarantees from the curator. The opposite of GDP is local differential privacy (LDP)[2]. The clients perform the perturbation before transmitting the data to the central server. The curator is not

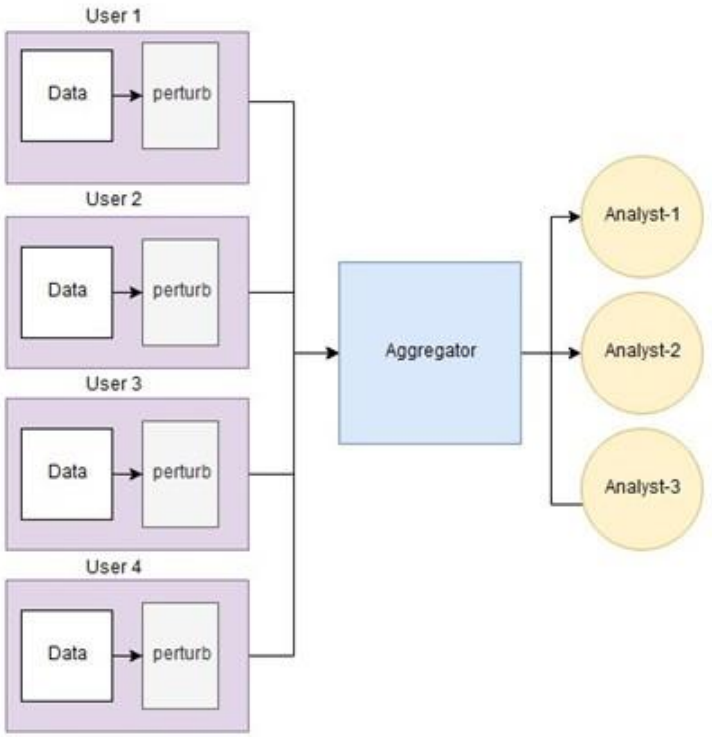

Fig. 2 *LocalDifferentialPrivacy*

aware of the original data, and hence, this results in better security. Clients can ensure privacy in case of a data leak from the curator, as the data was already perturbed before being transmitted to the curator. Hence, this offers better privacy guarantees. At the same time, because the perturbation is performed locally, there is a chance to lead to a higher noise, which will affect the utility of the data. Turing ϵ value becomes difficult because the local device cannot access data from different sources and is aware of only a limited amount of locally available data. Clients are often on the Internet of Things with less processing power and bandwidth, which cannot utilize complex processing algorithms. This technique is also helpful in federated machine learning, where the data is not transmitted to the server. Instead, models are trained and run in edge devices. Which provides privacy and preserves machine learning environments. This paper will explore the challenges in turning the privacy parameters in LDP into machine learning applications.

3 Related Work

With the introduction of big data analysis, cloud computing, IoT, and wearable technologies, the need for privacy-preserving data analysis and machine learning has become exceptionally important. Privacy-preserving machine learning is critical because a large amount of data contains personally identifiable information (PII). Yet, it is

crucial for data analysis and machine learning. Various techniques, such as anonymization, homophonic encryption, differential privacy, and a combination, can be used for successful research, sharing, and data storage

K-Anonymity model [3] is one of the earlier and well-known models used to anonymize data. This model protects privacy by combining quasi-identifier attributes in an equivalence class (EC) for K users. Identifying the suitable K-value is a challenge, and nowadays, with the dynamic nature of data and the large number of attributes captured, computing the K-value is a very resource-intensive and challenging process -Diversity Privacy Model [4] and t-closeness Privacy Model [5] models were subsequently tried to resolve the issues faced by the K-Anonymity models. But still, the privacy vs utility aspects continue to be a challenge. Secure Multi-Party Computation (MPC)[6] is another method to perform privacy-preserving machine learning. In this technique, participants agree to compute an output by utilizing private data without revealing their personal data to other participants. After processing, all participants combine their results to obtain the final results without obtaining confidential data. The hurdles in the real-world use of MPC are the lack of standardized MPC protocols, the possibility of malicious participants, and the complexity.

Homomorphic encryption[7] is another promising method that can ensure privacy while data is still secure and in encrypted form. In Homomorphic encryption,some calculations can be performed on encrypted cipher text without decrypting the cipher text. There are three primary forms [8] such as Partially Homomorphic Encryption (PHE)[**?**], Somewhat Homomorphic Encryption (SHE) [9], Fully Homomorphic Encryption (FHE) [10] [11]. Homomorphic encryption is resource-intensive, can cause challenges in key management, and has limitations in the operations performed on the encrypted data.

The other central area used for privacy-preserving machine learning is differential privacy methods.In this session, we will present some recent papers focusing on differential privacy in key-value pair data, which shows a growing interest in this field and recent advances and challenges in this area.

Researchers have proposed multiple mechanisms to release key-value pair data without compromising the privacy of individuals and other entities.RAPPOR [12] proposed by Erlingsson et al., was one of the first LDP techniques for frequency estimation in real-world datasets. It is an algorithm designed to collect aggregate statistics from Chrome browsers in 2014. RAPPOR uses randomized bit flips and randomized responses to preserve privacy.

Xiaolan Gu et al. wrote a paper in 2019 called ”PCKV: Locally Differentially Private Correlated Key-Value Data Collection with Optimized Utility,” which talked about how to use correlated perturbance to make differential privacy in key-value pair data more useful. Lin Sun et al., in their paper titled Conditional Analysis for Key-Value Data with Local Differential Privacy” [13], proposed conditional frequency estimation and conditional mean estimation for key-value pair data. This work was based on PrivKV [14] proposed by Qingqing Ye et al. in 2019, one of the first papers on mean and frequency estimation in key-value pair data. They introduced two iterative solutions, PrivKVM and PrivKVM+, to improve the estimation accuracy through a series of iterations without user involvement.

Adaptive local differential privacy (ADP) [15] methods try to balance the privacy and utility tradeoffs by adjusting the parameters based on feedback from external systems. Unlike the traditional methods, which treat all attributes the same. ADP methods add noise in such a manner to ensure the high utility of data and hence try to solve the utility vs tradeoff problem. L. Fan et al. suggested a novel approach called FAST ([16]) for use in real-time series data monitoring because the standard mechanisms had limited utility due to the high correlation between data values. FAST utilized feedback loops based on observed values to adjust the prediction model and sampling rate dynamically.

A recent paper by B. Niu et al. introduced AdaPDP [17] to maximize the utility by dynamically selecting the underlying noise generation algorithms based on different distributions and privacy settings. This model performs multiple samplings to satisfy the privacy requirements of the users. Each user may have different expectations of privacy and requirements based on culture, nationality, awareness, and income levels.”One size fits all” mechanisms often result in inadequate privacy for one set of users while overprotecting other users. Z. Jorgensen et al. introduced the concept of ”Personalised differential privacy.” (PDP)[18], which specified the privacy requirements at the user level and not based on global parameters.PMF (Probabilistic matrix factorization) is widely used in recommender systems. S Zhang et al. introduced a method called PDP-PMF (Probabilistic matrix

factorization for personalized data privacy") [19] to specify item-level privacy instead of global privacy for all users in 2019 and showed its effectiveness in recommendation quality and showed that is is helpful in real-world scenarios.

.Adaptive Laplace Mechanism [15] introduced by Phan et al. devised a method for satisfying differential privacy in deep learning that adapts noise based on the importance of features.

TDAP, created by Utaliyeva et al., adjusts how much noise is added to a dataset based on the needs of the analysis[20]. Adaptive differential privacy helps us solve the privacy vs. utility tradeoff to a greater extent. Apart from these, many researchers are working on other studies to explore the idea of adaptive differential privacy based on language models, machine learning, and other mathematical concepts.

4 Methodology

In this research, we explore the option of adaptive feedback from the aggregator to fine-tune the privacy and utility tradeoff of differential privacy. An adaptive feedback mechanism can be used to redistribute the noise based on the model's requirements and the importance of the features to maintain the model's accuracy while still preserving the model's privacy. The adaptive differential privacy model works based on the following concepts:

1. To understand the importance of features in the dataset, the user or aggregator decides the importance of each feature in the dataset by using various options like correlations, t-tests, or any other method based on the model's requirements. The most significant features must be identified in advance for the algorithm to preserve the accuracy of the model while maintaining the privacy of the individuals.

2. Set the privacy parameter expiration policy. This should be set carefully after analyzing the trends and events that could potentially affect the data for training the model.

3. Select the differential privacy algorithm based on the users' requirements and add noise based on the importance of the features. Noise will be distributed in such a way that the importance or predictive power of features remains inversely proportional to the significance of the feature.

4. Perform the model training using the required algorithm and evaluate the model's effectiveness. This process will continue until we get reasonable accuracy with the maximum possible privacy parameters.

5. Sent back the privacy parameters to the devices so that they can set the values as per the requirement

6. After the expiration of the privacy parameters based on the predefined time, rerun the process to adapt to the latest needs.

Algorithm 1 AdaDPriv

Input: Processed Data D, Initial Privacy Parameter ε, Minimum Viable accuracy M Essential features arranged from f[0] to f[n], with 0 being the lowest and n being the highest

Output: Calculated privacy parameter: β

1: $i = 0$

2: $m' = 0$

3: $e = 0$

4: $\beta[] = 0$

5: $firstIteration = True$

6: **while** $D[i] \leq length(D) - 1$ **do**

7: **while** $M' \leq M$ **do**

8: **if** $firstIteration = True$ **then** $e = \varepsilon$

9: **end if**

10: Perturb $D[i]$ using e ▷ Perturbation of data will be handled by the differential privacy algorithm

11: Calculate Accuracy M'

12: **if** $M \approx M'$ **then** ▷ Approximation criteria be defined separately $\beta[i] = e$ *firstIteration = True*

13:

14: $i = i + 1$ $M' = 0$

15: Exit Loop

16:

17: **end if** 18: **if** $M < M'$ **then**

19: Increase e ▷ Amount of increase defined separately

20: **else**If $M > M'$

21: Decrease e CommentAmount of decrease defined separately

22: **end if** 23: **end while**

24: **end while** 25: Send β to all devices

---

5 Experiment

We have used Python programs to simulate the procedure. The study has been conducted with synthetic and publicly available datasets downloaded from Kaggle. It has been found that the adaptive privacy model is better than the local differential privacy model when it comes to real-world problems when we know the essential attributes in advance. We can control the parameters on each client device.

Initial parameters		
Symbol	Description	Default value
ε	Initial privacy parameter	0.5
M	Minimum viable accuracy (MVP)	65%
β	Calculated Privacy parameter	0

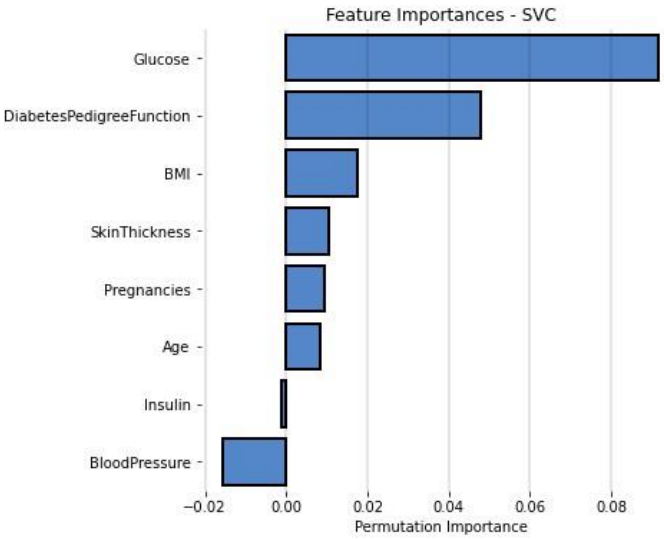

Fig. 3 *Important features*

The most important features of the dataset are ”Glucose” and ”DiabeticsPedigreeFunction,” followed by ”BMI. As per our model, we should add less noise to these variables and more noise to other attributes like age, blood pressure, etc., to get more privacy for the individuals contributing to the experiment. As per our algorithm, we will start with the attribute ”Blood Pressure” first, followed by Insulin, Age, and Pregnancies, to the most important feature, ”Glucose,” as shown in figure Fig(3)

We use the Laplace mechanism to add noise to each processed attribute in the dataset to get the noise based on the AdaDPriv model. We used logistic regression for this experiment, and without the added noise, it gave 78% accuracy. We added the noise to each attribute with an initial privacy budget of 1 and MVP of 77

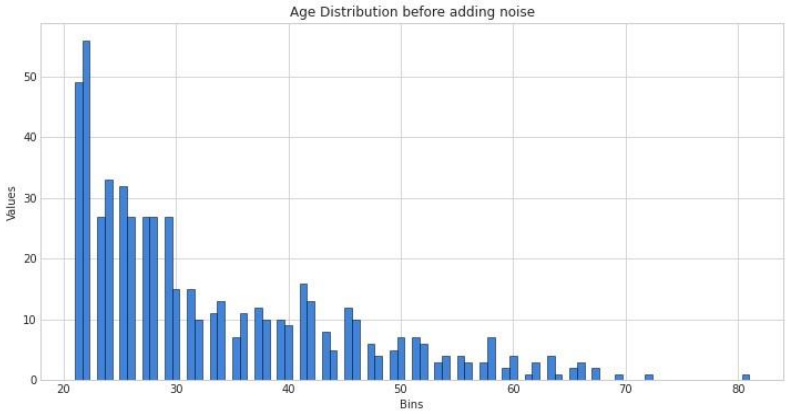

Fig. 4 *AgeDistributionBeforeNoise*

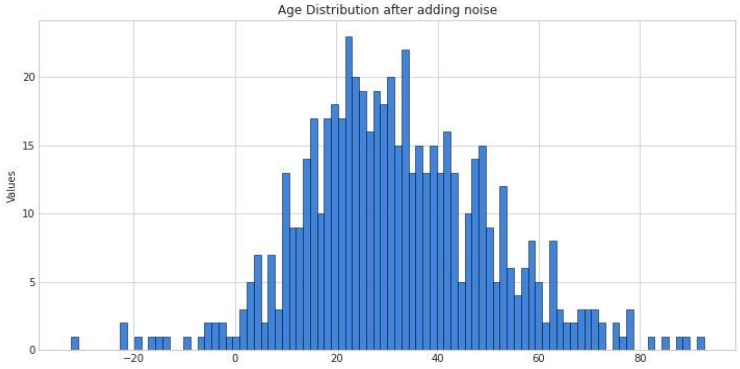

Fig. 5 *AgeDistributionAfteraddingNoise*

In this experiment, age is the most significant feature that could pose a security threat or violate the individual’s privacy. Fig. 4 depicts the age distribution before the perturbance of data, clearly showing the right-skewed distribution. We considered this feature to demonstrate how the algorithm can help protect the individual’s privacy. As shown in Fig. 5, the age distribution has completely changed due to the high amount of noise added to the data. However, the logistic regression gave 77 percent accuracy. Now, the computed β will be sent to all local devices to set the local differential privacy for each individual.

Accuracy assessment		
Metrics	Before Adding Noise	After Adding Noise
Accuracy	0.78	0.77
Precision	0.77	0.72
Recall	0.54	0.64
F1 Score	0.63	0.63
AUC Score	0.84	0.85

6 Conclusion

We introduced the AdaDPriv procedure to complete the privacy vs. utility tradeoff of traditional differential privacy methods. This helps us to consider each attribute in a dataset separately based on the importance of the predictive power of features and apply the differential privacy algorithms. This is useful when the dataset includes features that could help to identify a person who is present but less significant in calculating the outcome can be distorted more. In contrast, the essential features can remain less volatile to preserve the model's accuracy. Although this has shown promise, the process is compute-intensive. It needs the initial training and testing datasets to understand the importance of features and overall accuracy expectations. Challenges in these areas should be explored further, and more lightweight methods should be developed to fight the privacy vs. utility tradeoff.

References

[1] Dwork, C.: Differential Privacy. In: Bugliesi, M., Preneel, B., Sassone, V.,
Wegener, I. (eds.) Automata, Languages and Programming. Lecture Notes in Computer Science, pp. 1–12. Springer, Berlin, Heidelberg (2006). https://doi.org/ 10.1007/11787006 1

[2] Yang, M., Lyu, L., Zhao, J., Zhu, T., Lam, K.-Y.: Local Differential Privacy and Its Applications: A Comprehensive Survey. arXiv. arXiv:2008.03686 [cs] (2020). https://doi.org/10.48550/arXiv.2008.03686 . http://arxiv.org/abs/2008.03686 Accessed 2023-12-16

[3] Sweeney, L.: k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY.
International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **10**(05), 557–570 (2002) https://doi.org/10.1142/S0218488502001648 . Publisher: World Scientific Publishing Co. Accessed 2023-12-18

[4] L-diversity: Privacy beyond k-anonymity: ACM Transactions on Knowledge Discovery from Data: Vol 1, No 1. https://dl.acm.org/doi/abs/10.1145/1217299.
1217302 Accessed 2020-10-07

[5] Li, N., Li, T., Venkatasubramanian, S.: t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. In: 2007 IEEE 23rd International Conference on Data Engineering, pp. 106–115 (2007). https://doi.org/10.1109/ICDE.2007.367856 . ISSN: 2375-026X. https://ieeexplore.ieee.org/document/4221659 Accessed 2023-12-18

[6] Evans, D., Kolesnikov, V., Rosulek, M.: A Pragmatic Introduction to Secure Multi-Party Computation. Foundations and Trends® in Privacy and Security **2**(2-3), 70–246 (2018) https://doi.org/10.1561/3300000019 . Publisher: Now Publishers, Inc. Accessed 2023-12-18

[7] Yi, X., Paulet, R., Bertino, E.: Homomorphic Encryption. In: Yi, X., Paulet, R., Bertino, E. (eds.) Homomorphic Encryption and Applications. SpringerBriefs in Computer Science, pp. 27–46. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-12229-8 2 . https://doi.org/10.1007/978-3-319-12229-8 2 Accessed 2023-12-19

[8] Acar, A., Aksu, H., Uluagac, A.S., Conti, M.: A Survey on Homomorphic Encryption Schemes: Theory and Implementation. ACM Computing Surveys **51**(4), 79–17935 (2018) https://doi.org/10.1145/3214303 . Accessed 2023-12-19

[9] Secure pattern matching using somewhat homomorphic encryption | Proceedings of the 2013 ACM workshop on Cloud computing security workshop. https://dl. acm.org/doi/abs/10.1145/2517488.2517497 Accessed 2023-12-19

[10] Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) Fully Homomorphic Encryption without Bootstrapping. ACM Trans. Comput. Theory **6**(3) (2014) https://doi.org/10.1145/2633600 . Place: New York, NY, USA Publisher: Association for Computing Machinery

[11] Yi, X., Paulet, R., Bertino, E., Yi, X., Paulet, R., Bertino, E.: Fully Homomorphic Encryption, pp. 47–66. Springer, Cham (2014). https://doi.org/10.1007/ 978-3-319-12229-8 3 . Book Title: Homomorphic Encryption and Applications Series Title: SpringerBriefs in Computer Science. https://link.springer.com/10. 1007/978-3-319-12229-8 3 Accessed 2023-12-19

[12] Erlingsson, , Pihur, V., Korolova, A.: RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, pp. 1054–1067. ACM, Scottsdale Arizona USA (2014). https://doi.org/10.1145/2660267.2660348 . https://dl.acm.org/doi/10.1145/2660267.2660348 Accessed 2023-03-19

[13] Sun, L., Zhao, J., Ye, X., Feng, S., Wang, T., Bai, T.: Conditional Analysis for Key-Value Data with Local Differential Privacy. arXiv. arXiv:1907.05014 [cs] (2019). http://arxiv.org/abs/1907.05014 Accessed 2023-03-25

[14] Ye, Q., Hu, H., Meng, X., Zheng, H.: PrivKV: Key-Value Data Collection with Local Differential Privacy. In: 2019 IEEE Symposium on Security and Privacy (SP), pp. 317–331 (2019). https://doi.org/10.1109/SP.2019.00018 . ISSN: 23751207

[15] Winograd-Cort, D., Haeberlen, A., Roth, A., Pierce, B.C.: A framework for adaptive differential privacy. Proceedings of the ACM on Programming Languages **1**(ICFP), 10–11029 (2017) https://doi.org/10.1145/3110254 . Accessed 2023-12-19

[16] Fan, L., Xiong, L.: An Adaptive Approach to Real-Time Aggregate Monitoring With Differential Privacy. IEEE Transactions on Knowledge and Data Engineering **26**(9), 2094–2106 (2014) https://doi.org/10.1109/TKDE.2013.96 . Conference Name: IEEE Transactions on Knowledge and Data Engineering

[17] Niu, B., Chen, Y., Wang, B., Wang, Z., Li, F., Cao, J.: AdaPDP: Adaptive Personalized Differential Privacy. In: IEEE INFOCOM 2021 - IEEE Conference on Computer Communications, pp. 1–10 (2021). https://doi.org/10.1109/INFOCOM42981.2021.9488825 . ISSN: 2641-9874

[18] Jorgensen, Z., Yu, T., Cormode, G.: Conservative or liberal? Personalized differential privacy. In: 2015 IEEE 31st International Conference on Data Engineering, pp. 1023–1034 (2015). https://doi.org/10.1109/ICDE.2015.7113353 . ISSN: 2375-026X

[19] Zhang, S., Liu, L., Chen, Z., Zhong, H.: Probabilistic matrix factorization with personalized differential privacy. Knowledge-Based Systems **183**, 104864 (2019) https://doi.org/10.1016/j.knosys.2019.07.035 . Accessed 2023-03-25

[20] Utaliyeva, A., Shin, J., Choi, Y.-H.: Task-Specific Adaptive Differential Privacy Method for Structured Data. Sensors **23**(4), 1980 (2023) https://doi.org/10.3390/ s23041980 . Number: 4 Publisher: Multidisciplinary Digital Publishing Institute. Accessed 2023-03-30