# Poster: *Differentially Private Decision Tree Learning from Distributed Data*

Parisa Kaghazgaran
Computer Science and Engineering Dept.
University of North Texas
Denton, TX, USA
parisakaghazgaran@my.unt.edu

Hassan Takabi
Computer Science and Engineering Dept.
University of North Texas
Denton, TX, USA
takabi@unt.edu

## I. INTRODUCTION

The goal of privacy preserving data sharing is to share data for further analysis without revealing sensitive information. In this work, we propose a new Secure Multi-Party Computation (SMPC) protocol using Differential Privacy (DP) to protect data privacy while applying decision tree algorithm to horizontally distributed data. Pure secure multiparty computation approaches (using cryptographic techniques) are not scalable when they are used to analyze big data. Therefore, more efficient solutions are needed.

DP can achieve any desired level of privacy under appropriate $\varepsilon$ measure. It does not need complex mathematical operations like cryptographic techniques so it is very efficient and it has been used in big data analysis [1, 2]. In this work, we investigate how to design a more efficient SMPC protocol in Privacy-Preserving Data Analysis (PPDA) using DP.

PPDA approaches address different problems. In summary, they are classified into either non-interactive or interactive approaches. Non-interactive approaches aim to perturb data and then release data for further analysis, e.g., adding noise to data to guarantee ε-DP. While interactive protocols are executed between two or more parties and they address the problem in one the following scenarios:

1. A data owner owns the entire private data and a data miner will do computation on data in a private manner.
2. Data is distributed among mutually untrusted parties and they want to collaborate without sharing their actual data.

Most of the current differential privacy techniques assume a central database with a single owner [3]. When the database is distributed or owned by different parties, the problem of statistical data sharing becomes the key bottleneck for collaborative analysis tasks. We investigate the possibility of using differential privacy in a distributed setting. Specifically, we focus on a scenario where multiple parties owning private data want to apply decision tree algorithm (as a case study) over the union of their data.

Data could be distributed either vertically or horizontally. In the vertical mode, different attributes of the data are collected by different parties. In horizontal mode, diverse records of the data with the same attributes are collected by different parties. Our collaborative model is built over horizontally distributed data.

## II. OVERVIEW OF PROPOSED APPROACH

### A. Decision tree

Decision tree is usually built in a top-down approach, using a greedy strategy to select the best attribute for split. In ID3 algorithm, entropy function determines which attribute has the minimum entropy that classifies the data efficiently. In our scenario, we need to compute entropy of all attributes from distributed data with privacy considerations. So, the problem is reduced to the collaborative computation of entropy function using differential privacy. Assume $T$ (data records) is distributed between $P_1$, $P_2$, and $P_3$ ($T_1$, $T_2$, and $T_3$). In general, we assume data is distributed among $n$ parties ($n>2$). The following equation shows how the entropy value for attribute $A$ is calculated. (the original entropy function has been converted to this format.)

$$H_c(T|A) = -\sum_{j=1}^{m}\sum_{i=1}^{l}\left|T(a_j,c_i)\right| \times \log\left|T(a_j,c_i)\right| + \sum_{j=1}^{m}\left|T(a_j)\right| \times \log\left|T(a_j)\right|$$

We need to calculate $|T(a_j)|$ (number of data records in which attribute $A$ has value $a_j$) and $|T(a_j, c_i)|$ (number of data records in which attribute $A$ has value $a_j$ and class attribute has value $c_i$) for all the attributes. Our protocol shows how to calculate $|T(a_j)|$ in distributed mode that is easily extensible to computation of $|T(a_j, c_i)|$.

$$\left|T(a_j)\right| = \left|T_1(a_j)\right| + \left|T_2(a_j)\right| + \left|T_3(a_j)\right|$$

For convenience we use these notations: $|T_1(a_j)|=d_1$, $|T_2(a_j)|=d_2$, $|T_3(a_j)|=d_3$, and $|T(a_j)|= d_1 + d_2 + d_3$

### B. Differential Privacy

Differential privacy is an architecture to define and enforce privacy for statistics on sensitive data. "The fundamental idea is that a query on a database is differentially private if the contribution of an individual in the database can only marginally influence the query result. A deterministic query can be made deferentially private by perturbing the result with a certain amount of noise" [1]. In our protocol the query ($f$) is the count and the aggregate result is sum of the query results that are perturbed by Laplace noise.

$$f(T_1,T_2,T_3) = f(T_1) + f(T_2) + f(T_3) = d_1 + d_2 + d_3$$

The amount of noise depends on the query itself and a variety of perturbation algorithms [4, 5] have been proposed for different queries and datatypes. In this project we use Laplace Mechanism (LM) to perturb the results. Since the query is the count, sensitivity or $\Delta f$ is considered to be one.

Algorithm 1 shows how LM is used to generate noise. Parties involved in the protocol execute this algorithm collaboratively in fully distributed setting.

In: $d_1, d_2, d_3$; $\lambda = \frac{\Delta f}{\varepsilon}$ Out: $(d_1 + d_2 + d_3) + Lap(\lambda)$

1: $r_x \leftarrow u_{(0,1]}; r_y \leftarrow u_{(0,1]}$

2: $r_z = \lambda(\ln r_x - \ln r_y)$

3: $w = d_1 + d_2 + d_3 + r_z$

4: return $w$

Algortihm1 - LM

### C. Secret Sharing

Another building block we use in our protocol is secret sharing. A secret sharing scheme allows splitting a secret into multiple shares. The secret can be recovered if enough shares are collected and combined; the shares reveal no information about the secret if enough shares are collected. In our approach $d_1$, $d_2$, and $d_3$ are the secrets. We adopt a very efficient and simple scheme named additive secret sharing in which all shares are required to recover a secret [6].

## III. The Proposed Protocol

The issue we address here is how to aggregate data and compute statistics without parties learning each other's data and how to perturb the result to achieve differential privacy. Parties involved in the protocol are data owners (users) and a computation party (CP). The protocol consists of three steps.

1. Each user provides CP with one share of its input that is perturbed by portion of the noise (black lines in figure 1).
2. Each user gives the next party another share of its input that is also perturbed by a portion of the noise. The receiver aggregates the received message with its own input and passes to the next party (red lines in figure 1).
3. CP is responsible for aggregating the shares and producing the result perturbed by DP noise ($r_z$ in LM algorithm).

To guarantee that the output of secure sum provides ε-differential privacy, the result should have only one noise drawn from Laplace Mechanism instead of the sum of $n$ noises from the same distribution. Our protocol satisfies this condition by Summation of partial noises to obtain $r_z$ . $[N_i]$ represents portion of the DP noise.

### A. Related Work

The most significant improvement of our approach in comparison with two other similar works [1, 2] is that we do not need any trusted party. *Zhang et al.* proposed an approach for distributed decision tree learning with differential privacy in which one of the data owners must be trusted [2]. *Eigner et al.* proposed a model for differentially private data aggregation in which at least one of the computation parties must be trusted otherwise computation parties can collude and retrieve the actual value of data owners [1]. *Jagnnathan et al.* proposed an approach in which a private random decision tree is generated from centralized dataset [7]. They do not consider the privacy of middle queries, and add DP noise to the leaves.

In the additive sharing scheme that we use, any linear combination of secret-shared values can be performed by each party locally, without any interaction while multiplication of two secret-shared values requires communication between all of them. In other words, if $[d_1]$ denotes that value $d_1$ is secret-shared among $P_1; \ldots; P_n$ operations $[d_1] + [d_2]$, $[d_1] + c$, and $c \times [d_1]$ are performed by each $P_i$ locally on its shares of $d_1$ and $d_2$, while computation of $[d_1] \times [d_2]$ is interactive. The protocol proposed in [1] for data aggregation uses several SMPC schemes for collaborative noise generation that leads to high computation and communication overhead. Our protocol is executed in $2n$ rounds, and each party is responsible to generate portion of the DP noise($n$ = number of data owners).

Decision tree learning only seeks for the attribute with minimum entropy and the actual value of entropy is not important. We conclude if we select ε small for more privacy we still could have correct splitting attributes. Therefore, privacy will not affect data utility.

## IV. Conclusion and Future Work

Differential privacy is an efficient technique for data mining algorithms that are complex and, also, their input often consists of a large amount of data records. In this work, we showed how to apply DP along with secret sharing to decision tree algorithm. In future, we aim to apply DP to other data mining algorithms.

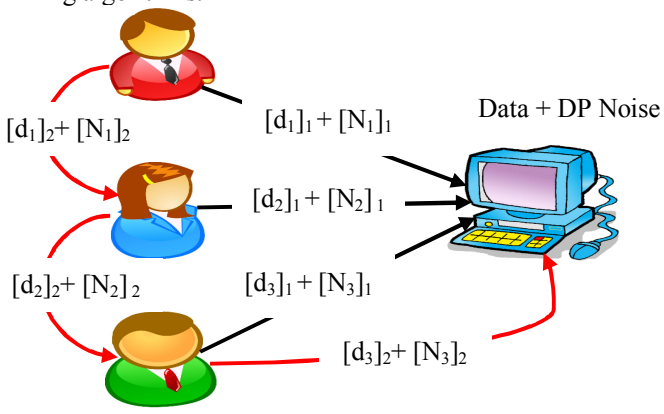


Fig. 1. Protocol Flow

## V. References


[1] F. Eigner, A. Kate, M. Maffei, F. Pampaloni, I. Pryvalov, "Differentially private data aggregation with optimal data utility", ACSAC '14 Proceedings of the 30th Annual Computer Security Applications Conference, pages 326-325, 2014.

[2] N. Zhang, M. Li, and W. Lou, "Distributed data mining with differential privacy," in 2011 IEEE International Conference on Communications (ICC), Jun. 2011, pp. 1–5, 2011.

[3] Y. Yang, Z. Zhang, G. Miklau, M. Winslett, and X. Xiao. "Differential privacy in data publication and analysis", In Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, pages 601–606, 2012.

[4] C. Dwork, F. McSherry, K. Nissim, and A. Smith. "Calibrating Noise to Sensitivity in Private Data Analysis", In TCC'06, pages 265-284, 2006.

[5] F. McSherry and K. Talwar. "Mechanism Design via Differential Privacy", In FOCS'07, pages 94-103, 2007.

[6] S. L. From and T. Jakobsen. "Secure Multi-Party Computation on Integers", Master's thesis, University of Aarhus, Denmark, 2006.

[7] G. Jagannathan, K. Pillaipakkamnatt, and R. N. Wright. 2012. A Practical Differentially Private Random Decision Tree Classifier. ACM Trans. Data Privacy 5, 1 (April 2012), 273-295.