HIERARCHICAL PRIVACY PRESERVING DISTRIBUTED FREQUENT ITEMSET MINING OVER VERICALLY DISTRIBUTED DATASET (HPPDFIM)

Fuad Ali moh. Alyarimi, Sonajharia Minz

Abstract— The process of defining association rules is dependent of frequent itemsets. Distributed data mining is a significant scenario due to the fact of large quantity of data and adaptive resource sharing between networks that provides computational strength. In this context data that is high in volume need to be distributed to various computational clients to perform data mining in distributed manner, which is scalable in regard to computation and process time. Henceforth it become a serious issue to protect the privacy of the data that distributed for mining. Here in this paper we explored a novel perturbation technique called connected guassian approach (CGA) to protect the data privacy that distributed and a distributed frequent itemset mining modal. The model here we proposed is referred as hierarchical privacy preserving distributed frequent itemset mining (HPPDFIM) that applied on vertically partitioned data tuples. The other underlying goal of the proposed model is that achieving privacy by using one way digestive standard during communication between distributed computational resources involved in DDM. The experiments revealed that the model is scalable and accurate unlike other perturbation standards. In a glance the goals of the proposed HPPDFIM is (i) protecting privacy during the distribution of the vertically partitioned data tuples even distributed computational resources compromised, (ii) achieving privacy during the conversations between participant distributed computational resources, (iii) and the frequent itemset mining should be done such that each node of the network that participating in mining process should not aware of the transaction elements of the other nodes that are involving in data mining process.

Index Terms— Anonymity data, Data Mining, Distributed Frequent Itemset Mining, guassian Perturbation, Perturbation Approach, privacy preserving data mining.

---

1. INTRODUCTION

Extracting the repeated patterns that are frequent than the desired threshold is an essential process of Data mining and knowledge discovery. Rapid progress in data collection and promulgation stratagies demands the reformation of the traditional mining modals. Along side a vivid progress experiencing in technologies related to data mining and knowledge discovery. In relation to these two circumstances, it is apparent that there is considerable scope for further research. In regard to desired computational resources to achieve scalability in mining and knowledge discovery (DM&KD) from high volumes of data, the distributed systems are considerably significant. Using distributed systems in data mining, which refers as distributed data mining exploring the need of research in different dimensions. One of that highly prioritized research issue is handling data identity leakage and privacy leakages. Preserving data privacy in DM&KD is considered as part of the process, henceforth evaluating the traditional algorithms scalability under association of privacy preserving techniques is seriously considered in earlier research [1][2][3].

The identity of an individual or of a transaction is clearly observable in a given transaction dataset. We are also obvious to observe the transactional scenarios, such as type of transaction, key attributes involved in that transactions, position and of those attributes in a transaction and conclusions from that transactionsset for decision making, future transaction forecasting from a given transaction dataset. Henceforth producing a transactional datset to third party data mining resources causes identity and privacy leakages.

In the context of literature, (i) if data allows the unautherized third party to discover confidential information, then it is threat called data inference, (ii) any of the privacy preserving model reorganize and modify the actual data to handle the datainference, such that actual state of the data and knowledge from that data remain concealed.

These privacy preserving techniques are conceptually different for distributed data mining (DDM). This is due to more than one third party participating in data mining and data distributed to these third party may recurrent or unique. Henceforth the privacy preserving techniques used in the case of single third party would not optimal in DDM.

- *Fuad Ali Mohammed Al-Yarimi is currently pursuing Ph.D degree program in computer science in Jawaharlal Nehru University,New Delhi, India, E-mail: frindh@gmail.com*
- *Sonajharia Minz, Profesor at School of Computer & Systems Science Jawaharlal Nehru University, New Delhi 110067 – India, (email: sonaminz@mail.jnu.ac.in)*

In our previous work [45] we considers the case when the data owner is unable to do knowledge extraction from a vast repository of data containing some sensitive information. In such a case the data miner may be a trusted party who may be allowed access to the data in its original form for knowledge extraction, so we didn't do any anonymity or perturbation to the data before encrypt. The paper only aims to explore the use of cryptographic techniques for secured channel for privacy preservation under the consideration of the data miner is a trusted party.

In this regard here we propose a Hierarchical Privacy Preserving Distributed Frequent Itemset Mining over Vertically Distributed data (HPPDFIM-VD) under the consideration of the data miner is not a trusted party so we need to perturbate the data before send it to the data miner.

The rest of the paper set as, first outline the recent proposals relevant to privacy preserving data mining, then we explore the proposed HPPDFIM-VD. Further we explore the performance analysis of the proposal, which followed by the conclusion and references.

2. RELATED WORK:

The exploration of the frequently cited privacy preserving with perturbation and cryptography models is follow.

Handling data inference using secure computation by multi-party, which is a cryptographic concept that also referred as Secure Malty Party Computation (SMPC) approach is optimal for high-end privacy [14], [15]. Henceforth any traditional mining algorithm under SMPC approach can preserve privacy for multiple users of the distributed environment [16]. Due to the obstacles such as computationally very expensive and failure to correlate hypothesis and expediency, these approaches are failed to be optimal.

In this regard considerable alternatives to SMPC have been proposed in recent literature, which are specific to the data mining task opted. The solutions devised in [14] are specific to decision tree mining over horizontally partitioned data. In the context of vertically partitioned data, a set of algorithms projected in [15] [18], the algorithms devised in [17] are specific to clustering approach. A data compression technique was devised in [19] to enable privacy preserving under collaborative distributed data mining and analysis. The modals [14] [15] [17] [18] [19] are providing the entire data such that it can't compromise for data inference. But untrusted parties may succed partially to infer the properties, which may be sensitive to an individual or a transaction. In this context, researchers coined a process called information hiding via property perturbation that progress the process of privacy preserving. In this regard the information hiding is achieved by transforming attributes of the given dataset such that it can be useful to apply data mining modals without compromising at scalability of the mining results. This information hiding modals follows either one the (i) Anonymizing [20], [21], [22], [23], [24], [25] (ii) attribute probability change [26], [27], [28] and (iii) Data perturbation [29], [30], [31], [32], [33], [34], [35], [36].

Privacy preserved data set related oprations is another research dimension of knowledge discovery strategy. In this regard set of modals devised in [37][39][40][41][42], which is two party based privacy preserved intersecting , combining two or more sets on state of one field. A study [38] explored that these proposed protocols vulunerable to leak information. While most of these protocols are equality based, algorithms in [38] compute arbitrary join predicates leveraging the power of a secure coprocessor. The models devisd in [43] are using Tiny trusted devices in regard to secure function evaluation.

The perturbation techniques build by using either additive or matrix ultiplication approaches. Few of the interesting additive based perturbation techniques can be found in [29], [30], [32], [33], [35] and matrix multiplication models are [31][34]. These models are optimal for continuous data. In recent time a multilevel perturbation model devised in [44]. Along the limits explored these perturbation techniques are optimal only for centralized data mining. The other factors in regard to this perturbation methods are (i) protection is centric to attributes identity and privacy, (ii) the genune relations between attributes, which extracted as mining results still disclosed to third party mining resource. Henceforth here in this paper we propose a Connected guassian approach to preturbate the attributes of the dataset in regard to achieve the privacy preserving in distributed data mining on vertically partitioned data. In the aim of preserving the privacy of emerged mining results at mining resource, the proposed technique perturbating even the positions of the attributes. Ideally we would get complete zero knowledge, but for a practical solution restricted information disclosure may be suitable. Finally, we quantify the accuracy and the efficiency of the algorithm, in view of the security restrictions.

Hierarchical Privacy Preserving Distributed Frequent Itemset Mining on Vertically Partitioned Dataset

The proposed HPPDFIM is a twofold model. In first fold the total dataset perturbated using proposed Connected Gaussian Approach (CGA), then the data will be partitioned vertically in non linier order, and the same will be distributed across the nodes. Here the proposed connected Gaussian approach protects the data anonymity and integrity even nodes compromised together and analyze. In second fold each node performs tuple level itemset mining on given vertical tuple of the dataset, which is in perturbated format. Then the nodes perform oneway communication in the order of tuples allotted. In the second fold of the process the node with first vertically partitioned tuple sends tuple level frequent itemsets tlfi to the node with second tuple of the vertically partitioned data. Then the node that received tlfi from its predecessor extract the multi tuple level frequent itemsets, here in this case it uses the tlfi received from its predecessor and tlfi found at that node,these multituple level frequent itemsets can be referred as mtlfi. Then this mtlfi will be send to next node in the order. Then mltfi that received and tlfi of the current node will be used to find mtlfi of the current node. The last node in the sequence then sends mtlfi that projected to data provider. Then the data provider removes the perturbstion from the received mtlfi, hence data provider can find final set of frequent itemsets.

A. Data Partitioning model:

Let D be the data base with E number of attributes and N number of transactions. The dataset D partitioned such that the N transactions data vertically tupled in nonleniar model and then send these tuples to nodes in a sequential order. Let D be the dataset having N transactions that are generated by set of attributes $\{a_1, a_2, a_3, ..., a_e\}$ Let consider a distributed network environment with m number data collection sensors, which are collecting data for attributes of the geven dataset D. The given dataset D vertically partitioned between data collection sensors (here after can be referred as node), such that tuple of attributes $\{a_1, a_2, a_3\}$ in node n_1 , attributes $\{a_4, a_5, ...a_{e-9}\}$ belongs to node n_2 , attributes $\{a_{e-9}, a_{e-8}\}$ belongs node n_3 , attributes $\{a_{e-i}, a_{e-j}, e_{e-k}\}$ belongs to node n_{m-1} and $\{a_{e-j}, ...a_e\}$ belongs to node n_m . The data collected for these attributes perturbated using proposed CGA and then partitions this data vertically according to the attribute partitions. And the same will be sent to appropriate nodes in the form of matrix.

B. Connected Perturbation Approach (CPA):

In general scenario of perturbation approaches for multiparty, the nexus of two or more parties leads to the exploration of the original state of the data and as frequent itemsets their influences in the business scenario. In this regard here we refine the general perturbation approaches as Connected Perturbation Approach, which prevents the leakage of the actual data state and their influence as frequent itemsets in business scenario even two or more parties compramized.

The base idea of the connected perturbation approach is, applying strategic hierarchical swaping between elements in transactions by their frequency. To protect the privacy in regard to this swaping we perturbate their frequency by hierarchical guassian noises. The steps follows.

Let dataset D with n number of transactions $T = \{t_1, t_2, t_3, ...t_{n-2}, t_{n-1}, t_n\}$. Each transaction t_j is a set of items $\{i_1, i_2, i_3, ..., i_{|t_j|}\}$ that are subset of total set of items I, which are using to generate dataset D. The data source authority decides support s of the frequency. The hierarchical guasian noise vector V of size s_{max} (here s_{max} indicates the max support of any item in given transaction dataset) will be generated such that the vector values must be in the range of 0 to 1 with fixed interval in ascending order.

Table 1: Hierarchical guassian noise vector preparation algorithm

Let s_{max} be the max support of any item in given transaction dataset
Let $g_{min} = rand\_double(min, max)$
Here g_{min} is minimum guassian perturbation threshold
Here $min := 0.0$ and $max := 0.1$
Let $g_{max} = 1 - g_{min}$
Here g_{max} is maximum guassian perturbation threshold
Let $g_{inc} = \frac{(g_{max} - g_{min})}{s_{max}}$
Here g_{inc} is guassian perturbation fixed increment threshold
Reset guassian noise vector V of size s_{max} to empty
foreach **i {i=0, 1, 2, 3,...., s_{max}}**
$\mathbf{V}_i = g_{min} + (i * g_{inc})$

Then these hierarchical guassian noise vector is used to perturbate the frequency of each item. Here we perturbate item names by adding fixed random string followed by the implication of a digestive algorithm such as MD5.

Table 2: Item position and frequency perturbation approach

For each item $i \in I$ begin
$s(i_j) = \mathbf{Find\_frequency}(i_j)$ //traverse the entire database and count the total occurances of the item i_j.
End;
Intialize vector PS as empty
For each item $i \in I$ begin
$ps(i_j) = \frac{s(i_j)}{V_j}$ // Here $ps(i_j)$ is perturbated support of the item i_j
Foreach j **such that** $\{j = 1....\|I\|\}$
$PS \leftarrow ps(i_j)$
end
For each transaction $t_i \in T$
begin

foreach **p {p=1.....|PS|/2}** begin
Swap items with support PS_p and items with support $PS_{s_{max}-p}$
end
end

Table 3: Field Name Perturbation approach

Let create a random string rs of given threshold size rl;
For each item $i \in I$ begin
$pi = MD5(i + rs)$
End

Forther this data vertically partitioned and distributed to the multyparties that are performing mining. The mining of frequent itemsets multiparies. The process of mining frequent itemsets on perturbated data at multiparies is explained in following sections

C. Tuple level Frequent Itemset (tlfi) Mining

Since the dataset is partitioned vertically, the total number of transactions at each node are N. By applying any of the popular itemset mining models such as apriori, FP-tree or BIDE the frequent itemsets of the locally available transactions should be found first by each node. Once the frequent itemsets with given support threshold found, then a boolean matrix that represents $tlfi$ should prepare such that rows represents frequent itemsets, columns represents transactions. Each column of the $tlfi$ should represent the presence of the frequent itemset in a given transaction, if frequent itemset r is existing in transaction c, then column $r\ X\ c$ represents 1 otherwise 0. If the node is first node then this node farwards $tlfi$ to its successor. If the node is not the first node in sequence, then continues to find the $mtlfi$, which described in following section.

D. Multi Tuple Level Frequent Itemset (mtlfi) Mining

In the sequence if a node n_i received $mtlfi_{i-1}$ from its predecessor, then the node n_i prepares $mtlfi_i$ as follows

$$mtlfi_i = tlfi_i \bigcup mtlfi_{i-1} \bigcup (tlfi_i * mtlfi_{i-1}) \quad \text{.......Eq(1)}$$

If the node n_i is not the last node in the sequence, then it farwards $mtlfi_i$ to next node n_{i+1}. This process continues til the last node in sequence found the $mtlfi_m$. Then this $mtlfi_m$ will be sent to the server (the process initiator).

E. Finding Frequent Itemsets.

Since $mtlfi_m$ received from last node in sequence is representing all the itemsets in their digestive format, hence server communicates with all nodes participated in mining process and collects the frequent itemsets in narmal format.

3. Analysis of the HPPDFIM by example

Let consider a dataset D with 6 transactions and 9 attributes. Let consider the coverage value 40% that represents support 3.6. As of the given dataset s_{min} is 1 and s_{max} is 4;

Let consider g_{min} as 0.0394;

Then the g_{max} is 0.9606;

The g_{inc} is 0.10673

The support of the items {A1, A2, A3, A4, A5, A6, A7, A8,A9} are {4, 3, 4, 4, 3, 3, 4, 3, 3}

The perturbated support of items {A1, A2, A3, A4, A5, A6, A7, A8,A9} are {27.372, 11.864, 11.123, 8.5778, 5.235 , 4.413 , 5.085 , 3.358 , 3.000} .

Order of items by perturubated support: {A9, A8, A6, A7, A5, A4, A3, A2, A1}. TO understand the process see table 4 & 5

Table 4: Actual dataset D

Transaction ID\attributes	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
T1	1	1	1	1	0	1	0	0	1
T2	1	0	1	0	1	0	1	0	0
T3	1	0	1	1	1	0	1	1	1
T4	0	1	0	1	0	1	1	0	0
T5	1	0	1	1	0	0	0	1	0
T6	0	1	0	0	1	1	1	1	1

Table 5: Position perturbated Dataset

Transaction ID\attributes	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
T1	1	1	1	0	0	1	1	1	1
T2	0	0	0	1	1	1	0	0	1
T3	1	0	0	1	1	1	1	0	1
T4	0	1	1	1	0	0	1	1	0

T5	0	0	0	0	0	1	1	0	1
T6	1	1	1	1	1	0	0	1	0

Field ID perturbation by salting random string and applying message digest technique:

Let consider rs is "v534$^gfjgfgSDFGH"

The resultant value after salting rs and applying MD5for field id

A1 is 346162c577bf5cba11dd7ce1098e3c0a

A2 is eb4210359fd0cd219ce8dd775cb2459c

A3 is c1e4e966ac715390a9986cd9c0ece2ee

A4 is 0454a42022ef166a73092cd0d75f57d7

A5 is 26d8473d4235dc8cb6ad79868d3b00e2

A6 is 99c4b622f8a5f1ec594356f0091c9cbc

A7 is 7185b700b8eb8ea2380c65f7c2372bb2

A8 is 442c57912f12e956ba36758da2358ee1

A9 is 17cfb4826cbaabd131413f0f2d69f315

Hence the final structure of the data at source can be found in table 6

Table 6: Data representation at source after perturbating frequency, position and field ids

Perturbated Attribute/ frequency and position perturbated transaction	T1	T2	T3	T4	T5	T6
346162c577bf5cba11dd7ce1098e3c0a	1	0	1	1	0	1
eb4210359fd0cd219ce8dd775cb2459c	1	0	0	0	0	1
c1e4e966ac715390a9986cd9c0ece2ee	1	0	0	0	0	1
0454a42022ef166a73092cd0d75f57d7	0	1	1	1	0	1
26d8473d4235dc8cb6ad79868d3b00e2	0	1	1	1	0	1
99c4b622f8a5f1ec594356f0091c9cbc	1	1	1	1	1	0
7185b700b8eb8ea2380c65f7c2372bb2	1	0	1	1	1	0
442c57912f12e956ba36758da2358ee1	1	0	0	0	0	1
17cfb4826cbaabd131413f0f2d69f315	1	1	1	0	1	0

Tuple of Attributes in node n_1:
`{346162c577bf5cba11dd7ce1098e3c0a,`

`eb4210359fd0cd219ce8dd775cb2459c,`

`c1e4e966ac715390a9986cd9c0ece2ee}`

Tuple of Attributes in node n_2:
`{0454a42022ef166a73092cd0d75f57d7`

`26d8473d4235dc8cb6ad79868d3b00e2}`

Tuple of Attributes in node n_3:
`{99c4b622f8a5f1ec594356f0091c9cbc`

`7185b700b8eb8ea2380c65f7c2372bb2`

`442c57912f12e956ba36758da2358ee1`

`17cfb4826cbaabd131413f0f2d69f315}`

Transaction matrix prepared from the data collected for attributes tuple $\{a_1, a_2, a_3\}$ for 6 transactions at node $n1$ is

Attributes → / Transactions ↓	346162c577bf5cba11dd7ce1098e3c0a	eb4210359fd0cd219ce8dd775cb2459c	c1e4e966ac715390a9986cd9c0ece2ee
T1	1	1	1
T2	0	0	0
T3	1	0	0
T4	0	1	1
T5	0	0	0
T6	1	1	1

Transaction matrix prepared from the data collected for attributes tuple $\{a_4, a_5\}$ for 6 transactions at node n_2 is

Attributes → / Transactions ↓	0454a42022ef166a73092cd0d75f57d7	26d8473d4235dc8cb6ad79868d3b00e2
T1	1	0
T2	0	1
T3	1	1
T4	1	0
T5	1	0
T6	0	1

Transaction matrix prepared from the data collected for attributes tuple $\{a_6, a_7, a_8, a_9\}$ for 6 transactions at node n_3 is

Attributes → Transactions ↓	99c4b622f8a5f1ec594356f0091c9cbc	7185b700b8eb8ea2380c65f7c2372bb2	442c57912f12e956ba36758da2358ee1	17cfb4826cbaabd131413f0f2d69f315
T1	1	1	1	1
T2	1	0	0	1
T3	1	1	0	1
T4	0	1	1	0
T5	1	1	0	1
T6	0	0	1	0

The frequent itemsets found at node n_1 are

({346162c577bf5cba11dd7ce1098e3c0a},

{eb4210359fd0cd219ce8dd775cb2459c},

{c1e4e966ac715390a9986cd9c0ece2ee},

{eb4210359fd0cd219ce8dd775cb2459c,c1e4e966ac715390a9986cd9c0ece2ee})

Then the $tlfi_{n_1}$ is as follow:

	T1	T2	T3	T4	T5	T6
{346162c577bf5cba11dd7ce1098	1	0	1	0	0	1
{eb4210359fd0cd219ce8dd775cb	1	0	0	1	0	1
{c1e4e966ac715390a9986cd9c0e	1	0	0	1	0	1
{eb4210359fd0cd219ce8dd775cb c1e4e966ac715390a9986cd9c0ec	1	0	0	1	0	1

Here above matrix represents

({346162c577bf5cba11dd7ce1098e3c0a},
{eb4210359fd0cd219ce8dd775cb2459c},
{c1e4e966ac715390a9986cd9c0ece2ee},
{eb4210359fd0cd219ce8dd775cb2459c,c1e4e966ac715390a9986cd9c0ece2ee}) [∈|∉]{T1, T2, T3, T4, T5, T6}

,

which indicates the existence of an itemset in a particular transaction.

Since node n_1 is first node in the sequence it will not perform the process of determining $mtlfi$, and farwards this $tlfi_{n_i}$ to next node n_2

The $tlfi_{n_2}$ found at node n_2 for frequent itemsets ({0454a42022ef166a73092cd0d75f57d7} {26d8473d4235dc8cb6ad79868d3b00e2}) is

	T1	T2	T3	T4	T5	T6
{0454a42022ef166a73092cd0d75	1	0	1	1	1	0
{26d8473d4235dc8cb6ad79868d3	0	1	1	0	0	1

Then at node n_2 the product of $tlfi_{n_1}$ and $tlfi_{n_2}$ will be found that referred as $tlfi_{n_1 X n_2}$

$tlfi_{n_1 X n_2} = tlfi_{n_1} X\ tlfi_{n_2}$:

	T1	T2	T3	T4	T5	T6
346162c577bf5cba11dd7ce1098e3c0a X 0454a42022ef166a73092cd0d75f57d7	1	0	1	0	0	0
eb4210359fd0cd219ce8dd775cb2459c X 0454a42022ef166a73092cd0d75f57d7	1	0	0	1	0	0
c1e4e966ac715390a9986cd9c0ece2ee X 0454a42022ef166a73092cd0d75f57d7	1	0	0	1	0	1
{eb4210359fd0cd219ce8dd775cb2459c,c1e4e966ac715390a9986cd9c0ece2ee} X0454a42022ef166a73092cd0d75f57d7	1	0	0	1	0	0
346162c577bf5cba11dd7ce1098e3c0a X 26d8473d4235dc8cb6ad79868d3b00e2	0	0	1	0	0	1
eb4210359fd0cd219ce8dd775cb2459c X 26d8473d4235dc8cb6ad79868d3b0	0	0	0	0	0	1

0e2						
c1e4e966ac715390a9986cd9c0ece2ee **X** 26d8473d4235dc8cb6ad79868d3b00e2	0	0	0	0	0	1
{eb4210359fd0cd219ce8dd775cb2459c,c1e4e966ac715390a9986cd9c0ece2ee} **X** 26d8473d4235dc8cb6ad79868d3b00e2	0	0	0	0	0	1

The final $\mathrm{tlfi}_{n_1 X n_2}$ after support analysis is

	T1	T2	T3	T4	T5	T6
c1e4e966ac715390a9986cd9c0ece2ee X 0454a42022ef166a73092cd0d75f57d7	1	0	0	1	0	1

Since the node n_2 is not the first node the sequence hence it constructs matrix mtlfi_{n_2} as follows

$$mtlfi_{n_2} = tlfi_{n_2} \cup mtlfi_{n_1} \cup tlfi_{n_1} X n_2$$

The matrix mltfi_{n_2} is

	T1	T2	T3	T4	T5	T6
{346162c577bf5cba11dd7ce1098	1	0	1	0	0	1
{eb4210359fd0cd219ce8dd775cb	1	0	0	1	0	1
{c1e4e966ac715390a9986cd9c0e	1	0	0	1	0	1
{eb4210359fd0cd219ce8dd775cb c1e4e966ac715390a9986cd9c0ec	1	0	0	1	0	1
{0454a42022ef166a73092cd0d75	1	0	1	1	1	0
{26d8473d4235dc8cb6ad79868d3	0	1	1	0	0	1
{c1e4e966ac715390a9986cd9c0e 0454a42022ef166a73092cd0d75f	1	0	0	1	0	1

Since the node n_2 is not last node of the sequence, hence farwards this mtlfi_{n_2} to next node n_3

The same process continues at node n_3 and determined tlfi_{n_3} and mtlfi_{n_3} matrices at node n_3 are

The matrix tlfi_{n_3} is:

	T1	T2	T3	T4	T5	T6
99c4b622f8a5f1ec594356f0091c9cbc	1	1	1	0	1	0
7185b700b8eb8ea2380c65f7c2372bb2	1	0	1	1	1	0
442c57912f12e956ba36758da2358ee1	1	0	0	1	0	1
17cfb4826cbaabd131413f0f2d69f315	1	1	1	0	1	0
{99c4b622f8a5f1ec594356f0091c9cbc, 7185b700b8eb8ea2380c65f7c2372bb2}	1	0	1	0	1	0
{99c4b622f8a5f1ec594356f0091c9cbc, 17cfb4826cbaabd131413f0f2d69f315}	1	1	1	0	1	0
7185b700b8eb8ea2380c65f7c2372bb2, 17cfb4826cbaabd131413f0f2d69f315	1	0	1	0	1	0

The matrix $(\mathrm{tlfi}_{n_3} X \mathrm{mtlfi}_{n_2})$ is:

	T1	T2	T3	T4	T5	T6
99c4b622f8a5f1ec594356f0091c9cbc X {26d8473d4235dc8cb6ad79868d3	1	0	1	0	1	0
7185b700b8eb8ea2380c65f7c2372bb2 X {26d8473d4235dc8cb6ad79868d3	1	0	1	1	1	0
442c57912f12e956ba36758da2358ee1 X {c1e4e966ac715390a9986cd9c0e	1	0	0	1	0	1
442c57912f12e956ba36758da2358ee1	1	0	0	1	0	1

X {eb4210359fd0cd219ce8dd775cb c1e4e966ac715390a9986cd9c0ec						
442c57912f12e956ba36758da2358 ee1 X {0454a42022ef166a73092cd0d75	1	0	0	1	0	1
17cfb4826cbaabd131413f0f2d69f3 15 X {346162c577bf5cba11dd7ce1098	1	0	0	1	0	1
17cfb4826cbaabd131413f0f2d69f3 15 X {26d8473d4235dc8cb6ad79868d3	1	0	1	0	1	0
{99c4b622f8a5f1ec594356f0091c9 cbc, 7185b700b8eb8ea2380c65f7c2372 bb2} X {26d8473d4235dc8cb6ad79868d3	1	0	1	0	1	0
{99c4b622f8a5f1ec594356f0091c9 cbc, 17cfb4826cbaabd131413f0f2d69f3 15} X {26d8473d4235dc8cb6ad79868d3	1	0	1	0	1	0
7185b700b8eb8ea2380c65f7c2372 bb2, 17cfb4826cbaabd131413f0f2d69f3 15 X {26d8473d4235dc8cb6ad79868d3	1	0	1	0	1	0

The matrix $mtlfi_{n_3}$ is: [$mtlfi_{n_3} = tlfi_{n_3} \cup mtlfi_{n_2} \cup (tlfi_{n_3} X mtlfi_{n_2})$]

[FDEB3FC83146079DAB257C7EF240D4FA]	3
[C6F2F93133905F75DA4B02CCC19AB66A]	3
[CA034CA406E03D1010A118F5B3677518]	4
[CA034CA406E03D1010A118F5B3677518, 6593D7B12FD418CDB35BBF438DE72F66]	4
[796BDC6F731D811A5F57EB0C06EFEF50]	3
[EBF1CA419D2EA2BCF2A208C3701A30E9]	4
[[796BDC6F731D811A5F57EB0C06EFEF50], [3048BEB67F69393F2F987E646F24194F]]	3
[[36EC455DE71F46C98F3131176973972A], [6593D7B12FD418CDB35BBF438DE72F66]]	3
[36EC455DE71F46C98F3131176973972A]	4
[[CA034CA406E03D1010A118F5B3677518], [36EC455DE71F46C98F3131176973972A]]	3
[8650E375EE80B2277A84FC9B85375E36]	3
[[C6F2F93133905F75DA4B02CCC19AB66A], [EBF1CA419D2EA2BCF2A208C3701A30E9]]	3
[[36EC455DE71F46C98F3131176973972A], [CA034CA406E03D1010A118F5B3677518, 6593D7B12FD418CDB35BBF438DE72F66]]	3
[3048BEB67F69393F2F987E646F24194F]	3
[6593D7B12FD418CDB35BBF438DE72F66]	4

Since the node n_3 is last node in the sequence, hence it sends $mtlfi_{n_3}$ as database level frequent itemsets to the server.

Then the server identifies the actual itemsets by removing the position and frequency perturbation that added at source level. The process explored below

Database level frequent itemsets with attribute representation is

A1	4
A1, A3	4
A1, A4	3
A2	3
A3	4
A4	4
A4, A1, A3	3
A4, A3	3
A5	3
A5, A7	3
A6	3
A6, A2	3
A7	4
A8	3
A9	3

Frequent Itemsets after removing the perturbation is:

A4, A1	3
A5, A7	3

A3, A1	4
A8	3
A7	4
A4, A3	3
A3	4
A3, A1, A4	3
A5	3
A2	3
A2, A6	3
A1	4
A9	3
A6	3
A4	4

The analysis of the frequent items after removing perturbation in frequency and position from the resultant frequent itemsets indicates the accuracy in identifying the frequent itemsets even from perturbated data. Here in this analysis we consider the basic itemset mining algorithm but it is quite evident to use any frequent itemset mining algorithm such as fptree, bide to find tlfi at each mining node.

CONCLUSION:

Here in this paper we explored a novel connected perturbation approach to preserve privacy in vertically partitioned distributed data mining. Most of the solutions currently available in recent literature either not compatible to distributed data mining or fail to avoid data leaks under nexus of one more data mining node authorities. In this regard the model that proposed here is perturbating actual dataset in connected manner. The proposed model that referred as "Hierarchical Privacy Preserving Distributed Frequent Itemset Mining Over Verically Distributed Dataset (HPPDFIM)" is perturbating the actual item frequency, postion and field id in connected manner. In future this model can be expanded to horizontally partitioned distributed data mining and multi level data mining.

REFERENCES:

[1]. W. Du and M. J. Atallah. Secure multi-party computation problems and their applications: A review and open problems. In Proceedings of the 2001 New Security Paradigms Workshop, Cloudcroft, New Mexico, Sept. 11-13 2001.
[2]. S. J. Rizvi and J. R. Haritsa. Privacy-preserving association rule mining. In Proceedings of 28th International Conference on Very Large Data Bases. VLDB, Aug. 20-23 2002.
[3]. Agrawal and C. C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In Proceedings of the Twentieth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Santa Barbara, California, USA, May 21-23 2001. ACM.
[4]. D. W.-L. Cheung, V. Ng, A. W.-C. Fu, and Y. Fu. Efficient mining of association rules in distributed databases. Transactions on Knowledge and Data Engineering, 8(6):911{922, Dec. 1996.
[5]. M. Kantarcioglu and C. Clifton. Privacy-preserving distributed mining of association rules on horizontally partitioned data. In The ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'02), June 2 2002.
[6]. P. Chan. On the accuracy of meta-learning for scalable data mining. Journal of Intelligent Information Systems, 8:5{28, 1997.
[7]. Prodromidis, P. Chan, and S. Stolfo. Meta-learning in distributed data mining systems: Issues and approaches, chapter 3. AAAI/MIT Press, 2000.
[8]. R. Chen, K. Sivakumar, and H. Kargupta. Distributed web mining using bayesian networks from multiple data streams. In The 2001 IEEE International Conference on Data Mining. IEEE, Nov. 29 - Dec. 2 2001.
[9]. R. Wirth, M. Borth, and J. Hipp. When distribution is part of the semantics: A new problem class for distributed knowledge discovery. In Ubiquitous Data Mining for Mobile and Distributed Environments workshop associated with the Joint 12th European Conference on Machine Learning (ECML'01) and 5th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'01), Freiburg, Germany, Sept.3-7 2001.
[10]. Y. Lindell and B. Pinkas. Privacy preserving data mining. In Advances in Cryptology { CRYPTO 2000, pages 36{54. Springer-Verlag, Aug. 20-24 2000.
[11]. R. Agrawal and R. Srikant. Privacy-preserving data mining. In Proceedings of the 2000 ACM SIGMOD Conference on Management of Data, Dallas, TX, May 14-19 2000. ACM.
[12]. C. Yao. How to generate and exchange secrets. In Proceedings of the 27th IEEE Symposium on Foundations of Computer Science, pages 162{167. IEEE, 1986.
[13]. O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game - a completeness theorem for protocols with honest majority. In 19th ACM Symposium on the Theory of Computing, pages 218{229, 1987.
[14]. Y. Lindell and B. Pinkas, "Privacy Preserving Data Mining," Proc. Int'l Cryptology Conf. (CRYPTO), 2000.
[15]. J. Vaidya and C.W. Clifton, "Privacy Preserving Association Rule Mining in Vertically Partitioned Data," Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, 2002.
[16]. O. Goldreich, "Secure Multi-Party Computation," Final (incomplete) draft, version 1.4, 2002.
[17]. J. Vaidya and C. Clifton, "Privacy-Preserving K-Means Clustering over Vertically Partitioned Data," Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, 2003.
[18]. A.W.-C. Fu, R.C.-W. Wong, and K. Wang, "Privacy-Preserving Frequent Pattern Mining across Private Databases," Proc. IEEE Fifth Int'l Conf. Data Mining, 2005.
[19]. B. Bhattacharjee, N. Abe, K. Goldman, B. Zadrozny, V.R. Chillakuru, M.del Carpio, and C. Apte, "Using Secure Coprocessors for Privacy Preserving Collaborative Data Mining and Analysis," Proc. Second Int'l Workshop Data Management on New Hardware (DaMoN '06), 2006.
[20]. C.C. Aggarwal and P.S. Yu, "A Condensation Approach to Privacy Preserving Data Mining," Proc. Int'l Conf. Extending Database Technology (EDBT), 2004.
[21]. Bertino, B.C. Ooi, Y. Yang, and R.H. Deng, "Privacy and Ownership Preserving of Outsourced Medical Data," Proc. 21st Int'l Conf. Data Eng. (ICDE), 2005.
[22]. D. Kifer and J.E. Gehrke, "Injecting Utility Into Anonymized Datasets," Proc. ACM SIGMOD Int'l Conf. Management of Data, 2006.
[23]. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "L-Diversity: Privacy Beyond K-Anonymity," Proc. Int'l Conf. Data Eng., 2006.
[24]. L. Sweeney, "K-Anonymity: A Model for Protecting Privacy," Int'l J. Uncertainty, Fuzziness and Knowledge-Based Systems (IJUFKS), vol. 10, pp. 557-570, 2002.
[25]. X. Xiao and Y. Tao, "Personalized Privacy Preservation," Proc. ACM SIGMOD Int'l Conf. Management of Data, 2006.
[26]. R. Agrawal, R. Srikant, and D. Thomas, "Privacy Preserving OLAP," Proc. ACM SIGMOD Int'l Conf. Management of Data, 2005.
[27]. W. Du and Z. Zhan, "Using Randomized Response Techniques for Privacy-Preserving Data Mining," Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, 2003.

[28]. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, “Privacy Preserving Mining of Association Rules,” Proc. ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining, 2002.

[29]. D. Agrawal and C.C. Aggarwal, “On the Design and Quantification of Privacy Preserving Data Mining Algorithms,” Proc. 20th ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems (PODS ’01), pp. 247-255, May 2001.

[30]. R. Agrawal and R. Srikant, “Privacy Preserving Data Mining,” Proc. ACM SIGMOD Int’l Conf. Management of Data (SIGMOD ’00), 2000.

[31]. K. Chen and L. Liu, “Privacy Preserving Data Classification with Rotation Perturbation,” Proc. IEEE Fifth Int’l Conf. Data Mining, 2005.

[32]. Z. Huang, W. Du, and B. Chen, “Deriving Private Information From Randomized Data,” Proc. ACM SIGMOD Int’l Conf. Management of Data (SIGMOD), 2005.

[33]. Li, J. Sun, S. Papadimitriou, G. Mihaila, and I. Stanoi, “Hiding in the Crowd: Privacy Preservation on Evolving Streams Through Correlation Tracking,” Proc. IEEE 23rd Int’l Conf. Data Eng. (ICDE), 2007.

[34]. K. Liu, H. Kargupta, and J. Ryan, “Random Projection-Based Multiplicative Data Perturbation for Privacy Preserving Distributed Data Mining,” IEEE Trans. Knowledge and Data Eng., vol. 18, no. 1, pp. 92-106, Jan. 2006.

[35]. S. Papadimitriou, F. Li, G. Kollios, and P.S. Yu, “Time Series Compressibility and Privacy,” Proc. 33rd Int’l Conf. Very Large Data Bases (VLDB ’07), 2007.

[36]. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, “On the Privacy Preserving Properties of Random Data Perturbation Techniques,” Proc. IEEE Third Int’l Conf. Data Mining, 2003.

[37]. R. Agrawal, A. Evfimievski, and R. Srikant, “Information Sharing across Private Databases,” Proc. ACM SIGMOD Int’l Conf. Management of Data, 2003.

[38]. R. Agrawal, D. Asonov, M. Kantarcioglu, and Y. Li, “Sovereign Joins,” Proc. 22nd Int’l Conf. Data Eng. (ICDE ’06), 2006.

[39]. C. Clifton, M. Kantarcioglu, X. Lin, J. Vaidya, and M. Zhu, “Tools for Privacy Preserving Distributed Data Mining,” ACM SIGKDD Explorations, vol. 4, no. 2, pp. 28-34, 2003.

[40]. B.A. Huberman, M. Franklin, and T. Hogg, “Enhancing Privacy and Trust in Electronic Communities,” Proc. First ACM Conf. Electronic Commerce, pp. 78-86, Nov. 1999.

[41]. M. Freedman, K. Nissim, and B. Pinkas, “Efficient Private Matching and Set Intersection,” Advances in Cryptology—EUROCRYPT, vol. 3027, pp. 1-19, 2004.

[42]. L. Kissner and D. Song, “Privacy-Preserving Set Operations,” Proc. Int’l Cryptology Conf. (CRYPTO), 2005.

[43]. Iliev and S. Smith, “More Efficient Secure Function Evaluation Using Tiny Trusted Third Parties,” Technical Report TR2005-551, Dept. of Computer Science, Dartmouth Univ., 2005.

[44]. Yaping Li, Minghua Chen, Qiwei Li, and Wei Zhang; "Enabling Multilevel Trust in Privacy Preserving Data Mining"; ieee transactions on knowledge and data engineering, vol. 24, no. 9, september 2012

[45]. Fuad Ali Mohammed Al-Yarimi, Sonajharia Minz; “Multilevel Privacy Preserving In Distributed Environment using Cryptographic Technique”; Proceedings of the World Congress on Engineering 2012 Vol I WCE 2012, July 4 - 6, 2012, London, U.K.