© 2016 IJSRSET | Volume 2 | Issue 3 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Privacy Preservation of Big Data Using Hadoop

Vishal Phadtare, Samir Kashid, Monika Dherange, Prof. Pramod Murkute

College of Engineer and Management Studies, Pune University, Maharashtra, India

ABSTRACT

In Big data applications data collection has grown continuously, due to this it becomes expensive to manage, capture or extract and process data using existing software tools. Performing data analysis is becoming expensive with increasing large volume of data in data warehouse. Data privacy is one of the challenges in data mining with big data. To preserved the privacy of the user we need to use some method so that data privacy is preserve and at the same time increase the data utility. In existing centralized algorithms it assumes that the all data should be at centralized location for anonymization which is not possible for large scale dataset. And there was distributed algorithms which mainly focus on privacy preservation of large dataset rather than the scalability issue. In the proposed system we focus to maintain the privacy for distributed data, and also overcome the problems of M-privacy and secrecy approach with new anonymization and slicing technique. Our main goal is to publish an anonymized view of integrated data, which will be prevents the vulnerable attacks. We use MR-Cube approach which addresses the challenges of large scale cube computation with holistic measure. Slicing contains tuple partition, vertical and horizontal partition, generalization, slicing and anonymization. At the slicing is successful then anonymized data can easily access by user effectively.

Keywords : Privacy, security, integrity, and protection, distributed databases SMC, TTP

I. INTRODUCTION

There is an increasing need for sharing data that contain personal information from distributed databases. For example, in the healthcare domain, a national agenda is to develop the Nationwide Health Information Network (NHIN)1 to share information among hospitals and other providers, and support appropriate use of health information beyond direct patient care with privacy protection.

Privacy preserving data analysis, and data publishing have received considerable attention in recent years as promising approaches for sharing data while preserving individual privacy. In a non-interactive model, a data provider (e.g., hospital) publishes a "sanitized" version of the data, simultaneously providing utility for data users (e.g., researchers), and privacy protection for the individuals represented in the data (e.g., patients). When data are gathered from multiple data providers or data owners,two main settings are used for anonymization .One approach is for each provider to anonymize the data independently (anonymize-and-aggregate, Fig. 1(a)), which results in potential loss of integrated data utility. A more desirable approach is *collaborative data publishing* which anonymizes data from all providers as if they would come from one source (aggregate-and-anonymize,Fig. 1(b)), using either a trusted third-party (TTP) or Secure Multi-party Computation (SMC) protocols.

Problem Settings. We consider the collaborative data publishing setting (Fig. 1(b)) with horizontally distributed data across multiple data providers, each contributing a subset of records *Ti*. Each record has an owner, whose identity should be protected. Each record attribute is either *an identifier*, which directly identifies the owner, or *a quasiidentifier* (QID), which may identify the owner if joined with a publicly known dataset, or a sensitive attribute,which should be also protected. As a special case, a data provider could be the data owner itself who is contributing its own records. A data recipient may have access to some background knowledge (*BK* in Fig. 1), which represents any publicly available information about released data, e.g., Census datasets.

Our goal is to publish an anonymized view of the integrated data, $T*$, which will be immune to attacks. Attacks are run by *attackers*, i.e., a single or a group (*a coalition*) of external or internal entities that wants to breach privacy of data using background knowledge, as well as anonymized data. Privacy is breached if one learns anything about data.

Existing Solutions. Collaborative data publishing can be considered as a multi-party computation problem, in which multiple providers wish to compute an anonymized view of their data without disclosing any private and sensitive information. We assume the data providers are semi-honest commonly used in distributed computation setting. A trusted third party (TTP) or Secure Multi-Party Computation (SMC) protocols [6] can be used to guarantee there is no disclosure of *intermediate* information *during* the anonymization. However, neither TTP nor SMC protects against inferring information using the anonymized data.

The problem of inferring information from anonymized data has been widely studied in a single data provider settings [3]. A data recipient that is an attacker, e.g.,$P0$, attempts to infer additional information about data records using the published data, $T*$, and backgroundknowledge, *BK*. For example, *k*-anonymity [10], [11] protects against identity disclosure attacks by requiring each quasi-identifier equivalence group (QI group) to contain at least *k* records. *l*-Diversity requires each QI group to contain at least *l* "well-represented" sensitive values Differential privacy guarantees that the presence of a record cannot be inferred from a statistical data release with little assumptions on an attacker's background knowledge.

New Challenges. Collaborative data publishing introduces a new attack that has not been studied so far. Each data provider, such as $P1$ in Fig. 1, can use both, anonymized data $T*$, and its own data $T1$ to infer additional information about other records. Compared to the attack by the external recipient in the second scenario, each provider has additional data knowledge of its own records,which can help with the attack. This issue can be further worsened when multiple data providers collude with each other.

In the social network or recommendation setting, a user may attempt to infer private information about other users using the anonymized data or recommendations assisted by some background knowledge and her own account information.Malicious users may collude or even create artificial accounts as in a shilling attack

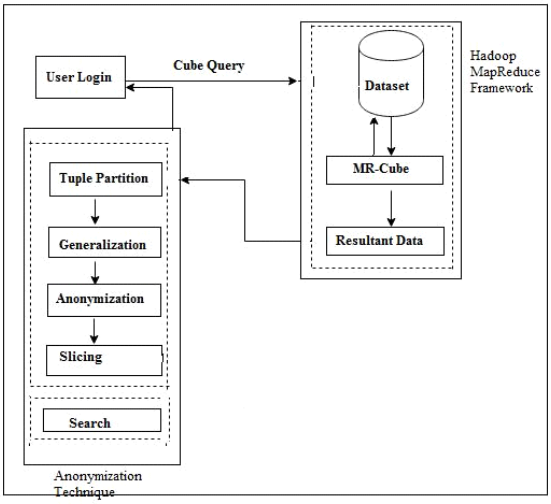


Figure 1. System Architecture

Contributions. We define and address this new type of "insider attack" by data providers in this paper. In general,we define an *m-adversary* as a coalition of *m* colluding data providers or data owners, and attempts to infer data records contributed by other data providers. Note that adversary models the external data recipient, who has only access to the external background knowledge. Since each provider holds a subset of the overall data, this inherent data knowledge has to be explicitly modeled, and considered when the data are anonymized.

We address the new threat introduced by *m*-adversaries, and make several important contributions. First, we introduce the notion of *m*-privacy that explicitly models the inherent data knowledge of an *m*-adversary, and protects anonymized data against such adversaries with respect to a given privacy constraint. For example, in Table 1 $T*b$ is an anonymized table that satisfies *m*-privacy ($m = 1$) with respect to *k*-anonymity and *l*-diversity ($k = 2$, $l = 2$).

Second, for scenarios with a TTP, to address the challenges of checking a combinatorial number of potential *m*-adversaries, we present heuristic algorithms for efficiently verifying *m*-privacy given a set of records. Our approach utilizes effective pruning strategies exploiting the equivalence group monotonicity property of privacy constraints and adaptive ordering techniques based on a novel notion of privacy fitness. We also

present a data *provideraware* anonymization algorithm with adaptive strategies of checking *m*-privacy, to ensure high utility and *m*-privacy of sanitized data with efficiency.Compared to our preliminary version [1], our new contributions extend above results. First, we adapt privacy verification and anonymization mechanisms to work for *m*-privacy w.r.t. to any privacy constraint, including nonmonotonic ones. We list all necessary privacy checks and prove that no fewer checks is enough to confirm *m*-privacy.Second, we propose SMC protocols for secure *m*-privacy verification and anonymization. For all protocols we prove their security, complexity and experimentally confirm their efficiency.

II. METHODS AND MATERIAL

2. m-Privacy Definition

We first formally describe our problem setting. Then, we present our m-privacy definition with respect to a privacy constraint to prevent inference attacks by m-adversary,followed by properties of this new privacy notion. Let T = {t1, t2, . . .} be a set of records with the same attributes gathered from n data providers P ={P1, P2, . . . , Pn}, such that Ti ⊆ T are records provided by Pi. Let AS be a sensitive attribute with a domain DS.

If the records contain multiple sensitive attributes then, we treat each of them as the sole sensitive attribute, while remaining ones we include to the quasi-identifier [12]. However, for our scenarios we use an approach, which preserves more utility without sacrificing privacy [15].Our goal is to publish an anonymized table T∗ while preventing any m-adversary from inferring AS for any single record. An m-adversary is a coalition of data users with m data providers cooperating to breach privacy of anonymized records.

2.1 m-Privacy

To protect data from external recipients with certain background knowledge BK, we assume a given privacy requirement C is defined as a conjunction of privacy constraints:C1 ∧ C2 ∧ . . . ∧ Cw. If a group of anonymized records T∗satisfies C, we say C(T∗) = true. By definition C(∅) is true and ∅ is private. Any of the existing privacy principles can be used as a component constraint Ci.We now formally define a notion of m-privacy with respect to a privacy constraint C, to protect the anonymized data against m-adversaries. The notion explicitly models the inherent data knowledge of an m-adversary, the data records they jointly contribute, and requires that each QI group, excluding any of those records owned by an m-adversary, still satisfies C. Note that this observation describes monotonicity of m-privacy with respect to the number of adversaries, and is independent from the privacy constraint C and records. In the next section we investigate monotonicity of m-privacy with respect to records for a given value of m.

2.2 Monotonicity of Privacy Constraints

Monotonicity of privacy constraints is defined for a single equivalence group of records, i.e., a group of records that QI attributes share the same generalized values. Let A1 be a mechanism that anonymizes a group of records T into a single equivalence group, T∗ = A1(T).Generalization based monotonicity of privacy constraints has been already defined in the literature Its fulfillment is crucial for designing efficient generalization algorithms In this paper we will refer to it as generalization monotonicity.

In the definition of generalization monotonicity there is an assumption that original records have been already anonymized into equivalence groups, which are used for further generalizations. In this paper, we introduce more general and record-based definition of monotonicity in order to facilitate the analysis, and design efficient algorithms for verifying m-privacy w.r.t. C.

3. Verification of m-Privacy

Checking whether a set of records satisfies m-privacy creates a potential computational challenge due to the combinatorial number of m-adversaries. In this section we first analyze the problem by modeling the adversary space. Then, we present heuristic algorithms with effective runing strategies and adaptive ordering techniques for efficiently checking m-privacy w.r.t. an EG monotonic constraint C. Implementation of introduced algorithms can be run by a trusted third party (TTP). For scenarios without such party, we introduce secure multi-party (SMC) protocols.Finally, in Appendix B.1, available online, we present modifications of TTP heuristics and SMC protocols to verify m-privacy w.r.t. non-EG monotonic privacy constraints.

3.1 Adversary Space Enumeration

Given a set of nG data providers, the entire space of madversaries (m varying from 0 to nG−1) can be represented using a lattice shown in Fig. 2. Each node at layer m represents an m-adversary of a particular combination of m providers. The number of all possible m-adversaries is given by _nG m_. Each node has parents (children) representing their direct super- (sub-) coalitions. For simplicity the space is depicted as a diamond, where a horizontal line at a level m corresponds to all m-adversaries, the bottom node to 0-adversary (external data recipient), and the top line to (nG − 1)-adversaries.In order to verify m-privacy w.r.t. a constraint C for a set of records, we need to check fulfillment of C for all records after excluding any possible subset of m-adversary records. When C is EG monotonic, we only need to check Cfor the records excluding all records from any m-adversary (Observation 2.3), i.e., adversaries on the horizontal line.Given an EG monotonic constraint, a direct algorithm can sequentially generate all possible and then check privacy of the corresponding remaining records. In the worst-case scenario, when m = nG/2, the number of checks is equal to the central binomial coefficient Thus, the direct algorithm is not efficient enough.

3.2 Heuristic Algorithms for EG Monotonic Constraints

In this section, we present heuristic algorithms for efficiently checking m-privacy w.r.t. an EG monotonic constraint.Then, we modify them to check m-privacy w.r.t. a non-EG monotonic constraint.

The key idea of our heuristics for EG monotonic privacy constraints is to efficiently search through the adversary space with effective pruning such that not all m-adversaries need to be checked. This is achieved by two different pruning strategies, an adversary ordering technique, and a set of search strategies that enable fast pruning.

Pruning Strategies. The pruning is possible thanks to the EG monotonicity of m-privacy If a coalition is not able to breach privacy, then all its subcoalitions will not be able to do so as well, and hence do not need to be checked (downward pruning). On the other hand, if a coalition is able to breach privacy, then all its super-coalitions will be able to do so as well, and hence do not need to be checked (upward pruning). In fact, if a sub-coalition of an m-adversary is able to breach privacy,then the upward pruning allows the algorithm to terminate immediately as the m-adversary will be able to breach privacy (early stop). Fig. 2 illustrates the two pruning strategies where + represents a case when a coalition does not breach privacy and otherwise.

The Top-Down Algorithm. The top-down algorithm checks the coalitions in a top-down fashion using downward pruning, starting from (nG−1)-adversaries, and moving down until a violation by an m-adversary is detected or all m-adversaries are pruned or checked.

The Bottom-Up Algorithm. The bottom-up algorithm is similar to the top-down algorithm. The main difference is in the sequence of coalition checks, which is in a bottom up fashion starting from 0-adversary, and moving up. The algorithm stops if a violation by any adversary is detected (early stop) or all m-adversaries are checked.

The Binary Algorithm. The binary algorithm (Algorithm1), inspired by the binary search algorithm, checks coalitions between (nG − 1)-adversaries and m-adversaries,and takes advantage of both pruning strategies Thanks to EG monotonicity of the privacy constraint,we do not consider coalitions of less than m adversaries The goal of each iteration in the algorithm is to search for a pair of coalitions Isub and Isuper, such that sub-coalition of Isuper, and Isuper breaches privacy, while Isub does not. Then, Isub and all its sub-coalitions are pruned (downward pruning), Isuper and all its super-coalitions are pruned (upward pruning) as well.

Adaptive Selection of Algorithms. Each of the above algorithms focuses on different search strategy, and hence utilizes different pruning. Which algorithm to use is largely dependent on the characteristics of a given group of providers. Intuitively, the privacy fitness score which quantifies also the level of privacy fulfillment of the group, may be used to select the most suitable algorithm.The higher the fitness score, the more likely m-privacy will be satisfied, and hence the top-down algorithm with downward pruning will significantly reduce the number of adversary checks. We utilize such strategy in the anonymization algorithm (discussed later), and experimentally evaluate it.

4. Secure m-Privacy Verification Protocols

All the above algorithms can be run by a trusted third-party (TTP). For settings without such a party, data providers need to run an SMC protocol. We assume that all providers are semi-honest, i.e., honest but curious. In this section we present secure protocols to verify m-privacy w.r.t. EG monotonic constraint C. A secure m-privacy verification protocol for a non-EG monotonic constraint is an extension of the bottom-up approach. Due to space limit details of such protocol were moved to Appendix D.2, available online.

Note that the TTP can recognize duplicated records, and treats them in the appropriate way. For SMC protocols all records are unique, and duplicates are not detected.

Preliminaries. Our SMC protocols are based on Shamir's secret sharing , encryption, and other secure schemas. In a secret sharing scheme, the owner of a secret message s prepares and distributes nG shares, such that each party gets a few shares (usually one). We use [s] to denote the vector of shares and [s]i to refer to an ith share sent to Pi.An algorithm reconstructing s requires any r shares as its input. To prevent any coalition of up to m providers to reveal intermediate results, we set r = m + 1. Note that receivers of shares do not have to be providers and trusted. They could be run as separate processes in a distributed environment, e.g., cloud, and still computations would stay information-theoretically secure In our implementation and complexity analyzes, we have used SEPIA framework

4.1 Secure EG Monotonic m-Privacy Verification

Assume that a group of data records is horizontally distributed among nG data providers. They would like to securely verify, if anonymization of their records into one QI group, is m-private w.r.t. C. Additionally, assume that verification of privacy defined by C is given (described below), and all providers have already elected a leader P_. Before verifying m-privacy the leader securely sorts data providers.

Secure Sorting and Adaptive Ordering. The main responsibility of the leader is to determine m-privacy fulfilment with as little privacy checks as possible. Our heuristic minimizes the number of privacy checks by utilizing EG monotonicity of C and adaptive ordering of m-adversary generation To define such order, P_ runs any sorting algorithm, which sorts providers by fitness scores of their local records, with all comparisons run securely. Applying the adaptive ordering heuristic uncovers the order of fitness scores of data providers. Without such ordering more privacy checks need to be performed. Our implementations of secure sorting protocol utilizes the Shamir's secret sharing scheme with r shares required to reconstruct a secret. To ensure m-privacy we set r = m+1.Thus, for nG data providers the protocol requires running a sorting algorithm, which takes O(nG log nG) secure comparisons. Each secure comparison has the same complexity, i.e.,requires a few secure multiplications, where each multiplication takes O(m2) time [21]. Thus, the secure sorting time complexity is equal to O(m2nG log nG). Each secure multiplication requires passing nG(nG − 1) messages in total,although only (m+1)2 of them are needed to get the result.Thus, the communication complexity is O(n3G log nG).

Secure m-Privacy Verification Protocol. After finding the order of data providers, the leader P_ starts verifying privacy for different coalitions of attackers, which are generated in specific order. A general scheme of secure mprivacy verification is the same for all heuristic algorithms Common steps are as follows. In the main loop P_ verifies privacy of records for m-adversaries until m-privacy can be decided (line 3). Note that in order to determine m-privacy w.r.t. EG monotonic C, it is enough to check privacy for all scenarios with exactly m attackers (Corollary 2.3). In the loop, P_ generates and broadcasts a coalition of potential adversaries I, so each party can recognize its status (attacker/non-attacker) for the current privacy check. Then, the leader runs the secure privacy verification protocol for I (line 6). If privacy could be breached,and I has no more than m data providers, then the protocol stops and returns negative answer (line 7). Otherwise, the information about privacy fulfillment is used to prune (upwards or downwards) a few potential m-adversaries (line 9). Finally, if m-privacy w.r.t. C can be decided, then P_ returns the results of m-privacy verification (line 10).For the binary algorithm, secure m-privacy verification protocol is also run by P_, which executes all steps of the Algorithm 1. The only difference is privacy verification, which is implemented as an SMC protocol. Due to lack of space details of this protocol are skipped.

4.2 Secure Privacy Constraint Verification

To allow using any privacy constraint in our m-privacy verification protocol, secure privacy verification is implemented as a separate protocol, and results of its runs are disclosed. Presenting verification protocols for any privacy constraint is out of the scope of this paper, but we present secure protocols to verify k-anonymity and l-diversity. All implementations use Shamir's secret sharing [19] as their main scheme. For a few subprotocols we use encryption (commutative, homomorphic, etc.), and other secure schemas for efficiency. Assume that there are nG data providers, and each data provider Pi provides Ti records.

Secure k-Anonymity Verification. To securely verify k-anonymity, the leader counts all records s = |T| using the secure sum protocol and securely compares s with k. Our implementation of the secure sum protocol uses only Shamir's secret sharing scheme First, all data providers run secure sum protocol in order to compute total number of records s. To avoid disclosing sits values is stored in distributed shares [s] (line 1). Finally,all providers securely compare [s] with k [21]. As the result,each provider gets a share of 1 if k-anonymity holds or a share of 0 otherwise (line 2).

5. Anonymization for m-Privacy

After defining the m-privacy verification algorithms and protocols, we can use it to anonymize a horizontally distributed dataset while preserving m-privacy w.r.t. C. In this section, we present a baseline algorithm, and then our approach that utilizes a data provider-aware algorithm with adaptive verification strategies to ensure high utility and m-privacy for anonymized data. We also present an SMC protocol that implements our approach in a distributed environment, while preserving security.For a privacy constraint C that is generalization monotonic, m-privacy w.r.t. C is also generalization monotonic (Theorem 2.1), and most existing generalization-based anonymization algorithms can be easily modified to guarantee m-privacy w.r.t. C. The adoption is straightforward, every time a set of records is tested for privacy fulfillment, we check m-privacy w.r.t. C instead. As a baseline algorithm to achieve m-privacy, we adapted the multidimensional Mondrian algorithm [18] designed for k-anonymity. The main limitation of such adaptation is that groups of records are formed oblivious of the data providers, which may result in over-generalization in order to satisfy m-privacy w.r.t. C.

5.1 Anonymization Algorithm

We introduce a simple and general algorithm based on the Binary Space Partitioning (BSP) (Algorithm 3). Similar to the Mondrian algorithm, it recursively chooses an attribute to split data points in the multidimensional domain space until the data cannot be split any further without breaching m-privacy w.r.t. C. However, the algorithm has three novel features: 1) it takes into account the data provider as an additional dimension for splitting; 2) it uses the privacy fitness score as a general scoring metric for selecting the split point; 3) it adapts its m-privacy checking strategy for efficient verification. The pseudo code for our provider-aware anonymization algorithm is presented in Algorithm 5.

Provider-Aware Partitioning. The algorithm first generates all possible splitting points, π, for QI attributes and data providers (lines 1 to 2). In addition to the multidimensional QI domain space, we consider the data provider of each record as its additional attribute A0. For instance,each record t contributed by data provider P1 will have t[A0] = P1. Introducing this additional attribute adds also a new dimension for partitioning. Using A0 to split data points decreases number of providers in each partition, and hence increases the chances that more sub-partitions will be m-private and feasible for further splits. This leads to a more precise view of the data, and have a direct impact on the anonymized data utility. To find the potential split point along this dimension, we impose a total order on the providers, e.g., sorting the providers alphabetically or based on the number of records they provide, and partition them into two group with approximately the same size.

5.2 Secure Anonymization Protocol

Algorithm 5 can be executed in a distributed environment by a TTP or by all providers running an SMC protocol. In this section we present a secure protocol for semi-honest providers. As an SMC schema we use Shamir's secret sharing, but, when needed, we employ also encryption.The key idea of the protocol is to use existing SMC protocols. The first step for all

providers is to elect the leader P_ by running a secure election protocol which then runs Algorithm 6.The most important step of the protocol is to choose an attribute used to split records based on fitness scores of record subsets. Splitting is repeated until no more valid splits can be found, i.e., any further split would return records that violate the privacy.

Secure anonymization protocol runs as follows. First, the median of each attribute Ai is found by running the secure median protocol (line 4, [27]). All records with the Ai values less than the median and some records with the Ai values equal to the median establish the distributed set Ts,i.Remaining records define the distributed set Tg,i. Then, mprivacy w.r.t. C is verified for Ts,i by running the secure verification protocol, i.e., either Algorithm 2 or 10 (line 8).w.r.t. C, then this split becomes a candidate split. For each candidate split, minimum fitness score of Ts,i and Tg,i is computed (secure fitness score protocol is described below).

Among candidate splits, the one with the maximal fitness score is chosen, and the protocol is run recursively for its subpartitions (lines 21 to 22). If no such attribute can be found for any group of records, the protocol stops.Secure m-privacy anonymization protocol calls three different SMC subprotocols: the secure median [27], [28], the secure m-privacy verification (Section 4), and the secure fitness score (Algorithm 7). The last protocol needs to be defined for each privacy constraint C (described below). For the sake of this analysis, we assume that all these protocols are perfectly secure, i.e., all intermediate results can be inferred from the protocol outputs.At each anonymization step following values are disclosedimedians si of all QID attributes, fulfillment of m-privacy w.r.t. C for records split according to every computed median, and, for m-private splits, the order of privacy fitness scores of all verified subsets of records.Medians of all QID attributes need to be revealed to allow each provider defining its local subgroups of records.

III. RESULTS AND DISCUSSION

Experiments

We run two sets of experiments for *m*-privacy w.r.t. *C* with the following goals: 1) to compare and evaluate the different *m*-privacy verification algorithms and 2) to evaluate and compare the proposed anonymization algorithm with the baseline algorithm in terms of both utility and efficiency. All experiments have been run for scenarios with a trusted third party (TTP), and without it (SMC protocols). Due to space restrictions all experiments for a TTP setting are in the previous version of the paper [1] and in Appendix C,available online.

6.1 Experiment Setup

We merged the training and testing sets of the Adult dataset2. Records with missing values have been removed. All remaining 45,222 records have been randomly distributed among *n* providers. As a sensitive attribute *AS* we chose *Occupation* with 14 distinct values.

Name	Description	Verification	Anonymization
m	Power of m-privacy	3	3
n	Number of data providers	–	10
n_G	Number of data providers contributing to a group	10	–
$\|T\|$	Total number of records	–	1000
$\|T_G\|$	Number of records in a group	150	–
k	Parameter of k-anonymity	30	30
l	Parameter of l-diversity	3	3

Table 1: Experiment Settings and Default Values of SMC Protocols

To implement SMC protocols, we have enhanced the SEPIA framework which utilizes Shamir's secret sharing scheme Security of communication is guaranteed by the SSL using 128-bit AES encryption scheme. For the secure *l*-diversity protocol we have used commutative Pohlig-Hellman encryption scheme with a 64-bit key

6.2 Secure *m*-Privacy Verification

The objective of the first set of experiments is to evaluate the efficiency of different heuristics in generating attacker coalitions for privacy verification. Note that computation times are presented in seconds, not milliseconds.

Attack Power. In this experiment, we compare *m*-privacy verification heuristics against different attack powers, and different number of data providers. Fig. 4(a) shows computation time with varying *m* and *nG* for all heuristics. Similar to the TTP implementation, the secure protocols for the *top-down* and *binary* algorithms demonstrate the best performance. The difference between these two approaches is negligible for most values of *m*. The *direct* approach is not that efficient as the above algorithms except small and large values of *m*. The *bottom-up* approach is useful only for very small values of *m*. Numbers of messages that are generated, while running protocols (not shown), are between 104 and 106 for different *m*, and lead to the same conclusions.

Binary Algorithm:

Data: Anonymize records DATA from providers P, an EG monotonic C, a fitness scoring function score F , and the n.

Result: if DATA is private secure C then True, else false

1. sites = sort_sites(P, increasing order, scoreF)
2. Apply slicing
3. while verify data-privacy(DATA, n, C) = 0 do
4. super = next_instance size(n− 1)&& (size_of_tupples (Σ)) // identification of column
5. if privacy breached_by(Psuper, C) = 0 then
6. prune_all_sub-instances_downwards(Psuper)
7. continue
8. Psub = next_sub-instance_of(Psuper,n)
9. if privacy_is_breached_by(Psub, C) = 1 then
10. return 0 // early stop
11. while instance_between(Psub, Psuper) do
12. I = next_instance between(Psub, Psuper)
13. if privacy breached_by(P,C) = 1 then
14. Psuper = P
15. else
16. Psub = P
17. prune_all_sub-instances_downwards(Psub)
18. prune_all_super-instances_upwards(Psuper)
19. return 1

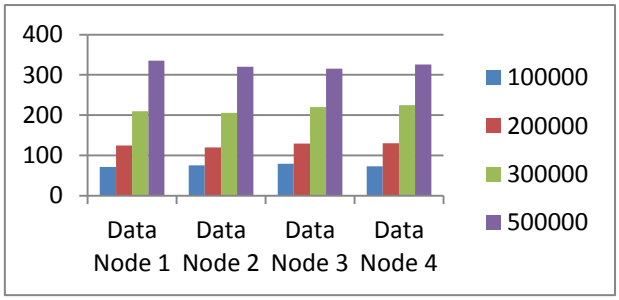
Table 2 shows the final output of system after execution of all algorithms. Basically slicing will provide the better security than existing approach

Name	Address	Zip code	Age	Disease
Jhon	Nairobi	452***	[0-25]	******
Ruby	Melbourne	145***	[26-50]	******
Alex	Sidney	365***	[26-50]	******
Bobby	Jakarta	356***	[51-75]	******

Table 2: 1-Slicing Result

In first experiment we have generate high dimension health care data with 300000 records and execute the system with local database, when we compare the complexity its very high. Its take around 8 minutes for all execution for all commands.

In second experiment we execute the system with distributed servers on windows platform with same dataset and used 4 data servers. Finally collect the result, below tables' shows Below figure shows the server execution time for different high dimension databases.



IV. CONCLUSION

In real world applications managing and mining Big Data is Challenging task, as the data concern large in a volume, distributed and decentralized control and complex. To preserving privacy of the distributed data we need technique which handle this data without

data/information loss and the resultant anonymized data will available for users.

Data provider first remove all explicit identifier from the data but simply removing explicit identifying information is not sufficient for protecting privacy. To handle and compute this large scale Data we used MR-Cube approach to compute large scale data sets. To overcome the problems of M-privacy and secrecy approach we use new anonymization and slicing technique.

V. REFERENCES

[1] Xindong Wu, Fellow, IEEE, Xingquan Zhu, Senior Member, IEEE, Gong-Qing Wu,and Wei Ding, Senior Member, IEEE Data Mining with Big Data in IEEE TRANS-ACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 26, NO. 1, JANUARY 2014.

[2] Arnab Nandi, Cong Yu, Philip Bohannon, and Raghu Ramakrishnan, Fellow, IEEE,Data Cube Materialization and Mining over MapReduce TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 6, NO. 1, JANUARY 2012 .

[3] Benjamin C.M. Fung, Ke Wang, and Philip S. Yu, Fellow, IEEE, Anonymizing Classification Data for Privacy Preservation in IEEE TRANSACTIONS ON KNOWL- EDGE AND DATA ENGINEERING, VOL. 19, NO. 5, MAY 2007.

[4] D. Mohanapriya, Dr.T.Meyyappan, High Dimensional Data Handling Technique Using Overlapping Slicing Method for Privacy Preservation in International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 6, June 2013.

[5] Tiancheng Li, Ninghui Li, Jian Zhang, Ian molloy Slicing: A New Approach for Privacy Preserving Data Publishing in IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 24, NO. 3, MARCH 2012.

[6] Madhuri Patil, Sandip Ingale Privacy Control Methods for Anonymous And Confidential Database Using Advance Encryption Standard in International Journal of Computer Science and Mobile Computing, Vol. 2, Issue. 8, August 2013.

[7] D. Mohanapriya, Dr.T.Meyyappan, High Dimensional Data Handling Technique Using Overlapping Slicing Method for Privacy Preservation in International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 6, June 2013.

[8] Senthil Raja M And Vidya Bharathi D Enhancement of Privacy Preservation in Slicing Approach Using Identity Disclosure Protection in ITSI Transactions on Electrical and Electronics Engineering (ITSITEEE) Volume -1, Issue -2, 2013.

[9] Xuyun Zhang, Laurence T. Yang, Senior Member, IEEE, Chang Liu, and Jinjun Chen, Member, IEEE A Scalable Two-Phase Top-Down Specialization Approach for Data Anonymization Using MapReduce on Cloud in IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 25, NO. 2, FEBRUARY 2014.

[10] Dhanshri S. Lad , Rasika P. Saste, Di_erent Cube Computation Approaches: Survey Paper(IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) ,4057-4061, 2