ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 361 - 366 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

C-Privacy: Collaborative Data Publishing to Preserve Integrity in Data Dissemination

N.LALITHA[1], S.AMARNATH BABU [2]
[1]II year M.Tech, India,lalitha.nagaram@gmail.com
[2]Associate professor, India, amardots@gmail.com

Abstract: we consider the collaborative data publishing problem for anonym zing horizontally partitioned data at multiple data providers. We consider a new type of "insider attack" by colluding data providers who may use their own data records (a subset of the overall data) in addition to the external background knowledge to infer the data records contributed by other data providers. The paper addresses this new threat and makes several contributions. First, we introduce the notion of m-privacy, which guarantees that the anonymized data satisfies a given privacy constraint against any group of up to m colluding data providers. Second, we present heuristic algorithms exploiting the equivalence group monotonicity of privacy constraints and adaptive ordering techniques for efficiently checking m-privacy given a set of records. Finally, we present a data provider-aware anonymization algorithm with adaptive m-privacy checking strategies to ensure high utility and m-privacy of anonymized data with efficiency. Experiments on real-life datasets suggest that our approach achieves better or comparable utility and efficiency than existing and baseline algorithms while providing m-privacy guarantee Slicing partitions the data set both vertically and horizontally. Vertical partitioning is done by grouping attributes into columns based on the correlations among the attributes. Each column contains a subset of attributes that are highly correlated. Horizontal partitioning is done by grouping tupelos into buckets efficiently checking m-privacy given a set of records. We present a data provider-aware anonymization algorithm with adaptive m-privacy checking strategies to ensure high utility and m-privacy of anonym zed data with efficiency. We introduce a novel data anonymization technique called slicing to improve the current state of the art. The basic idea of slicing is to break the association cross columns, but to preserve the association within each column. This reduces the dimensionality of the data and preserves better utility than generalization and bucketization

INTRODUCTION

Data mining is widely used by researchers for science and business purposes. Data collected from individuals are important for decision making or pattern recognition. Privacy-preserving processes have been developed to sanitize private information from the samples while keeping their utility. Slicing protects privacy because it breaks the associations between uncorrelated attributes, which are infrequent and thus identifying Slicing preserves utility because it groups highly correlated attributes together, and preserves the correlations between such attributes.

To avoid the identification of records in micro data, uniquely identifying information like names and social security numbers are removed from the table. However, this first sanitization still does not ensure the privacy of individuals in the data field using the seemingly innocuous attributes gender, date of birth, and 5-digit zip code. In fact, those three attributes were used to link Massachusetts voter registration records (which included the name, gender, zip code, and date of birth) to supposedly anonym zed medical data from GIC (which included gender, zip code, date of birth and diagnosis). This "linking attack" managed to uniquely identify the medical records of the governor of Massachusetts in the medical data

Sets of attributes (like gender, date of birth, and zip code in the example above) that can be linked with external data to uniquely identify individuals in the population are called quasi-identifiers. To counter linking attacks using quasi identifiers, Samarati and Sweeney proposed a definition of privacy called k-anonymity

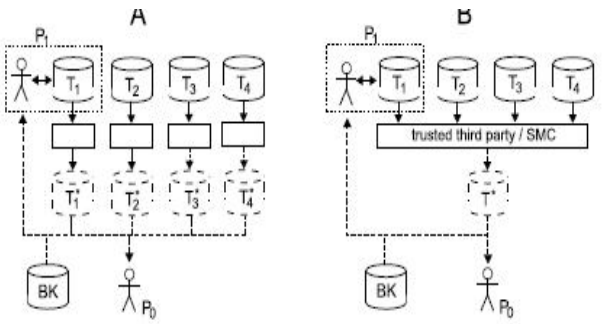


Fig. 1. Distributed data publishing settings.

A table satisfies anonymity if every record in the table is indistinguishable from at least k − 1 other records with respect to every set of quasi-identifier attributes; such a table is called a k-anonymous table. Hence, for every combination of values of the quasi-identifiers in the k-anonymous table, there are at least k records that share those values. This ensures that individuals cannot be uniquely identified by linking attacks

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 361 - 366 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

Existing System

We assume the data providers are semi-honest , commonly used in distributed computation setting. They can attempt to infer additional information about data coming from other providers by analyzing the data received during the anonymization. A data recipient, e.g. P0, could be an attacker and attempts to infer additional information about the records using the published data (T∗)

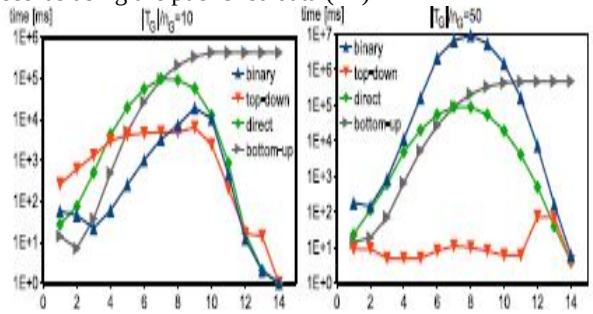


Fig2. Runtime vs. m

and some background knowledge (BK) such as publicly available external data.

The C-privacy verification algorithm, we can now use it in anonymization of a horizontally distributed dataset to achieve m-privacy. In this section, we will present a baseline algorithm, and then our approach that utilizes a data provider-aware algorithm with adaptive m-privacy checking strategies to ensure high utility and m-privacy for anonymized data. The algorithm first generates all possible splitting points, π, for QI attributes and data providers. In addition to the multidimensional QI domain space, we consider the data provider or data source of each record as an additional attribute of each record, denoted as A0 .Introducing this additional attribute in our multi-dimensional space adds a new dimension for partitioning. This leads to more splits resulting 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 can impose a total order on the providers.

We monitor that this multi set-based generalization is the same to a trivial slicing scheme where each column contains exactly one attribute, because both approaches preserve the exact values in each attribute but split the association between them within one potential split point. We observe that while one-attribute-per-column slicing preserves attribute distributional information, it does destroy attribute correlation, for the reason that each attribute is in its own column. In slicing, one groups associated attributes together in one column and save their correlation. For instance, in the sliced table shown in Table correlations between Age and Sex and correlations between Zip code and Disease are conserved. In fact, the sliced table encodes the same amount of information as the original data with observe to correlations between attributes in the same column.

C-Privacy and Syntactic Privacy Constraints.

Let *C* be aSyntactic privacy constraint, i.e., a constraint that preserves Data truthfulness at the record level, e.g., *k*-anonymity, *l*diversity, and *t*-closeness. *T* satisfying *C* will only Guarantee 0-privacy w.r.t. *C*, i.e., *C* is not guaranteed to Hold for every QI group after excluding records belonging to any single data provider. *M-Privacy* is defined w.r.t. a Privacy constraint *C*, and hence will inherit all strengths And weaknesses of *C*. *m*-Privacy w.r.t. *C* protects against Privacy attacks issued by any *m*-adversary if and only if, *C* protects against the same attacks by an external data recipient. *m*-Privacy notion is orthogonal to the privacy constraint *C* being used, and enhances privacy it defines to settings, where up to *m* data providers collude.

One-Attribute-Per-Column Slicing Data:

We observe that while one-attribute-per-column slicing preserves attribute distributional information, it does destroy attribute correlation, for the reason that each attribute is in its own column. In slicing, one groups associated attributes together in one column and save their correlation. For instance, in the sliced table shown in Table correlations between Age and Sex and correlations between Zip code and Disease are conserved. In fact, the sliced table encodes the same amount of information as the original data with observe to correlations between attributes in the same column

Table .1.Data for Non-Sensitive & Sensitive

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	13053	28	Russian	Heart Disease
2	13068	29	American	Heart Disease
3	13068	21	Japanese	Viral Infection
4	13053	23	American	Viral Infection
5	14853	50	Indian	Cancer
6	14853	55	Russian	Heart Disease
7	14850	47	American	Viral Infection
8	14850	49	American	Viral Infection
9	13053	31	American	Cancer
10	13053	37	Indian	Cancer
11	13068	36	Japanese	Cancer
12	13068	35	American	Cancer

There will likely be multiple adversaries with different levels of knowledge, each of which is consistent with the full joint distribution. Suppose Bob has a disease that is (a) very likely among people in the age group [30-50], but (b) is very rare for people of that age group who are doctors.

Comparison with Bucketization.

To compare slicing with bucketization, we first note that bucketization can be viewed as a special case of slicing, where there are exactly two columns: one column contains only the SA, and the other contains all the QIs. The ad-vantages of slicing over bucketization can be understood as follows. First, by partitioning attributes into more than two columns, slicing can be used to prevent membership dis-closure. Our empirical evaluation on a real dataset shows that bucketization does not prevent membership disclosure in Section Second, unlike bucketization, which requires a clear sep-aration of QI attributes and the sensitive attribute, slicingcan be used without such a separation. For dataset such as the census data, one often cannot clearly separate QIs from

SAs because there is no single external public database that one can use to determine which attributes the adversary al-ready knows. Slicing can be useful for such data.Finally, by allowing a column to contain both some QI attributes and the

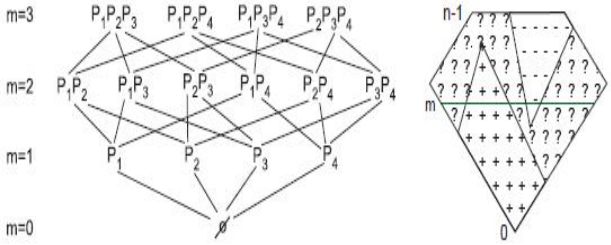
ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 361 - 366 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

sensitive attribute, attribute correlations between the sensitive attribute and the QI attributes are preserved. For example, in Table 1(f), Zipcode and Disease form one column, enabling inferences about their correla-tions. Attribute correlations are important utility in datapublishing. For workloads that consider attributes in isola-tion, one can simply publish two tables, one containing all QI attributes and one containing the sensitive attribute.

Adversary Space Enumeration

Given a set of *nG* data providers,the entire space of *m*adversaries (*c* varying from 0 to *nG*−1) can be represented using a lattice shown in. Each node at layer *m* Represents an *c*-adversary of a particular combination of *c* 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.



C –Adversary space and pruning strategies upward and downward

Proposed System

An adversary who only knows the interaction of age and illness will think that it is very likely for Bob to have that disease. However, an adversary who also knows that Bob is a doctor is more likely to think that Bob does not have that disease. Thus, although additional knowledge can yield better inferences on average, there are specific instances where it does not. Thus the data publisher must take into account all possible levels of background knowledge.

In this section we present two attacks, the homogeneity attack and the background knowledge attack, and we show how they can be used to compromise a k-anonymous dataset.

Table3. PARAMETERS AND VALUES

Name	Description	Verification	Anonymization
α	Weight paramter	0.3	0.8
m	Power of m-privacy	5	3
n	Total number of data providers	–	10
n_G	Number of data providers contributing to a group	15	–
$\|T\|$	Total number of records	–	45,222
$\|T_G\|$	Number of records in a group	{150, 750}	–
k	Parameter of k-anonymity	50	30
l	Parameter of l-diversity	4	4

Homogeneity Attack: Alice and Bob are antagonistic neighbors. One day Bob falls ill and is taken by ambulance to the hospital. Having seen the ambulance, Alice sets out to discover what disease Bob is suffering from. Alice discovers the 4-anonymous table of current inpatient records published by the hospital (Figure 2), and so she knows that one of the records in this table contains Bob's data.

Since Alice is Bob's neighbor, she knows that Bob is a 31-year-old American male who lives in the zip code 13053.Therefore, Alice knows that Bob's record number is 9,10,11, or 12. Now, all of those patients have the same medical condition (cancer), and so Alice concludes that Bob has cancer

Table.2 C-Adversy & C-diversity

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	130**	< 30	*	Heart Disease
2	130**	< 30	*	Heart Disease
3	130**	< 30	*	Viral Infection
4	130**	< 30	*	Viral Infection
5	1485*	≥ 40	*	Cancer
6	1485*	≥ 40	*	Heart Disease
7	1485*	≥ 40	*	Viral Infection
8	1485*	≥ 40	*	Viral Infection
9	130**	3*	*	Cancer
10	130**	3*	*	Cancer
11	130**	3*	*	Cancer
12	130**	3*	*	Cancer

Background Knowledge Attack: Alice has a pen friend named Unmake who is admitted to the same hospital as Bob, and whose patient records also appear in the table shown in Figure 2. Alice knows that Umeko is a 21 year old Japanese female who currently lives in zip code 13068. Based on this information,

Alice learns that Umeko's information is contained in record number 1, 2, 3, or 4. Without additional information, Alice is not sure whether Umeko caught a virus or has heart disease.

This set of experiments compares estimates of our provider aware and the baseline approaches, and evaluates the overhead of our solution. Due to high runtime of protocols, we estimated their computation times using runs of TTP algorithms and computation times of sub protocols. As a comparison, we implemented an independent approach in which each provider anonymized its data on its own. We observe that its runtime is independent of m and n, and equals to 1.2 seconds (not shown). However, the query error or the utility of the anonymized data is significantly worse than the collaborative setting (Appendix C.3).Attack Power.We first evaluate both anonymization heuristics with varying attack power m. shows theestimated computation time with varying m for both approaches. As expected for EG monotonic constraints, increasing m results in stopping anonymization process

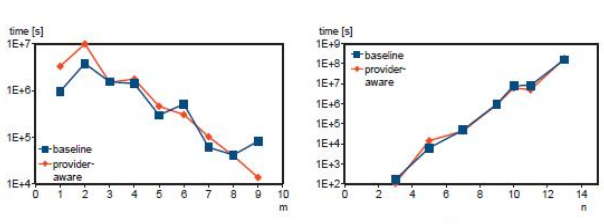


Fig.Computation time Vs M and number of providers

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 361 - 366 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

MEMBERSHIP DISCLOSURE PROTECTION

Let us first examine how an adversary can infer member-ship information from bucketization. Because bucketization releases the QI values in their original form and most individuals can be uniquely identified using the QI values, the adversary can simply determine the membership of an individual in the original data by examining the frequency of the QI values in the bucketized data. Specifically, if the frequency is 0, the adversary knows for sure that the individual is not in the data. If the frequency is greater than 0, the adversary knows with high confidence that the individual is in the data, because this matching tuple must belong to that individual as almost no other individual has the same QI values.The above reasoning suggests that in order to pro-tect membership information, it is required that, in the anonymized data, a tuple in the original data should have a similar frequency as a tuple that is not in the original data. Otherwise, by examining their frequencies in the anonymized data, the adversary can differentiate tuples in the original data from tuples not in the original data.We now show how slicing protects against membership disclosure. Let D be the set of tuples in the original data and let D be the set of tuples that are not in the original data. Let Ds be the sliced data. Given Ds and a tuple t, the goal of membership disclosure is to determine whether $t \in D$ or $t \in D$. In order to distinguish tuples in D from tuples in D, we examine their differences. If $t \in D$, t must have atleast one matching buckets in Ds. To protect membership information, we must ensure that at least some tuples in D should also have matching buckets. Otherwise, the adver-sary can differentiate between $t \in D$ and $t \in D$ by examining the number of matching buckets.We call a tuple an original tuple if it is in D. We call a tuple a fake tuple if it is in D and it matches at least one bucket in the sliced data. Therefore, we have considered two measures for membership disclosure protection. The first measure is the number of fake tuples. When the num-ber of fake tuples is 0 (as in bucketization), the membership information of every tuple can be determined. The second measure is to consider the number of matching buckets for original tuples and that for fake tuples. If they are sim-ilar enough, membership information is protected because the adversary cannot distinguish original tuples from fake tuples.Slicing is an effective technique for membership disclosure protection. A sliced bucket of size k can potentially match kc tuples. Besides the original k tuples, this bucket can in-troduce as many as $kc - k$ tuples in D, which is $kc-1 - 1$times more than the number of original tuples. The exis-tence of such tuples in D hides the membership information of tuples in D, because when the adversary finds a matching bucket, she or he is not certain whether this tuple is in D ornot since a large number of tuples in D have matching buck-ets as well. In our experiments (Section 6), we empirically evaluate slicing in membership disclosure protection. However, it is well known that Japanese have an extremely low incidence of heart disease. Therefore Alice concludes with near certainty that Umeko has a viral infection.Two popular anonymization techniques are generalization and bucketization. Generalization replaces a value with a "less-specific but semantically consistent" value. Three types of encoding schemes have been proposed for generalization: global recoding, regional recoding, and local recoding. Global recoding has the property that multiple occurrences of the same value are always replaced by the same generalized value.

Table 4.Generalised table

Age	Sex	Zipcode	Disease
[20-52]	*	4790*	dyspepsia
[20-52]	*	4790*	flu
[20-52]	*	4790*	flu
[20-52]	*	4790*	bronchitis
[54-64]	*	4730*	flu
[54-64]	*	4730*	dyspepsia
[54-64]	*	4730*	dyspepsia
[54-64]	*	4730*	gastritis

The Anonymized Table T

⋆. Since the quasi-identifiers might uniquely identify tuples in T , the table T is not published; it is subjected to an anonymization procedure and the resulting table T⋆ is published instead.

There has been a lot of research on techniques for anonymization (see Section 7 for a discussion of related work). These techniques can be broadly classified into generalization techniques, generalization with tuple suppression techniques, and data swapping and randomization techniques In this paper we limit our discussion only to generalization techniques.

Table 5.Bucketized table

Age	Sex	Zipcode	Disease
22	M	47906	flu
22	F	47906	dyspepsia
33	F	47905	bronchitis
52	F	47905	flu
54	M	47302	gastritis
60	M	47302	flu
60	M	47304	dyspepsia
64	F	47304	dyspepsia

Privacy preserving data analysis and publishing has received considerable attention in recent years most work has focused on a single data provider setting and considered the data recipient as an attacker.

A large body of literature assumes limited background knowledge of the attacker and defines privacy using relaxed adversarial notion by considering specific types of attacks. Representative principles include k-anonymity l-diversity and t-closeness. Few recent works have modeled the instance level background knowledge as corruption and studied perturbation techniques

Table.6 Multibased-generalization

Age	Sex	Zipcode	Disease
22:2,33:1,52:1	M:1,F:3	47905:2,47906:2	dysp.
22:2,33:1,52:1	M:1,F:3	47905:2,47906:2	flu
22:2,33:1,52:1	M:1,F:3	47905:2,47906:2	flu
22:2,33:1,52:1	M:1,F:3	47905:2,47906:2	bron.
54:1,60:2,64:1	M:3,F:1	47302:2,47304:2	flu
54:1,60:2,64:1	M:3,F:1	47302:2,47304:2	dysp.
54:1,60:2,64:1	M:3,F:1	47302:2,47304:2	dysp.
54:1,60:2,64:1	M:3,F:1	47302:2,47304:2	gast.

Privacy preserving data analysis and publishing has received Considerable attention in recent year. Mostwork has focused on a single data provider setting and considered the recipient as an attacker. A large body of literature assumes limited background knowledge of the attacker, and defines privacy using relaxed adversarial notion by considering specific types of attacks. Representative principles include k-anonymity, ldiversity, and t-closeness . A few recent works

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 361 - 366 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

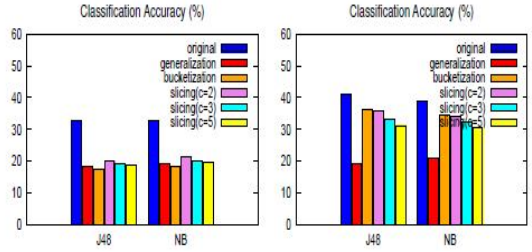
have modeled the instance level background knowledge as corruption, and studied perturbation techniques under these syntactic privacy notions. In the distributed setting that we study, since each data holder knows its own records, the corruption of records is an inherent element in our attack model, and is further complicated by the collusive power of the data providers. On the other hand, differential privacy is an unconditional privacy guarantee but only for statistical data release or data computations. There are some works focused anonymizations of distributed data. Studied distributed anonymizationfor vertically partitioned data using k-anonymity. Hong et al. studied classification on data collected from individual data owners, while maintaining k-anonymity. Jurczyket .proposed a notion called l└ -site-diversity to ensure anonymity for data providers in addition to privacy of the data subjects. Mironov et al. Studied SMC techniques to achieve differential privacy. Mohammed et al. proposed SMC techniques for anonym zing distributed data using the notion of LKC-privacy to address high dimensional data. Gal et al. proposed a new way of anonymization multiple sensitive attributes, which could be used to implement m-privacy w.r.t. l-diversity with providers as one of sensitive attributes. However, this approach uses the
same privacy requirements for all sensitive attributes, while m-privacy has no such limitation Nergiz et al. proposed a look ahead approach in horizontally distributed anonymization. In their approach providers disclose some information about data in order to decide if collaborative anonymization will gain more information than individual one. We leave for the future research applying the look ahead approach to colluding
scenarios considered with m-privacy. Our work is the first that considers data providers as potentialattackersthecollaborative data publishing setting and explicitly models their inherent instance knowledge as well as potential collusion between them. The m-privacy verification problem in the combinatorial-adversary search space is reminiscent of the frequentItemset mining problem in which the search space is the Combination of all items. An example of EG monotonic constraints is support, which is used in mining itemsets. Each
Item corresponds to a single data provider, and a frequent Itemset represent a group of private records. Due to the apriority property of frequent itemsets or EG monotonicity of the frequency count, both upward and downward pruning
are possible. Taking advantage of the dual-pruning is an essential point of the algorithm presented in . The main difference with our approach is the goal of constraint verifications. To find frequent itemsets, all itemsets need to be decided either by checking or pruning. Checking m privacy of a group of records for EG monotonic privacy requires finding out if all m-coalitions are not able to Compromise privacy of remaining records (Corollary) After simple modifications (e.g., not using early stop) our algorithm can be used to find frequent itemsets and the dual-pruning algorithm can be used to verify m-privacy,but in both cases they will not be efficient.

PRIVACY THEARTS

When publishing micro data, there are three types of pri-vacy disclosure threats. The first type is membership disclo-sure. When the dataset to be published is selected from a large population and the selection criteria are sensitive (e.g.,only diabetes patients are selected), one needs to prevent ad-versaries from learning whether one's record is included in
the published dataset. The second type is identity disclosure, which occurs when an individual is linked to a particular record in the released table. In some situations, one wants to protect against identity disclosure when the adversary is uncertain of member-ship. In this case, protection against membership disclo-sure helps protect against identity disclosure. In other sit-uations, some adversary may already know that an individual's record is in the published dataset, in which case, membership disclosure protection either does not apply or is insufficient.

Impact of Privacy Constraints

We also performed a set of experiments evaluating the impact of the privacy constraints on the utility of data using anonymization algorithms for *m* privacy.In our experiments, the constraint is defined as a conjunction of *k*-anonymity and *l*-diversity.showsruntime and query errors with varying privacy constraint restrictiveness (varying *k* and *l*). Query error values are relativeanddependent from selectiveness of queries Query error values are different for different queries, but our algorithmwill always have the same or lower error comparing to the baseline.



Fig(a) SENSITIVE(ooc-15) Fig(b) QI(occ-15)

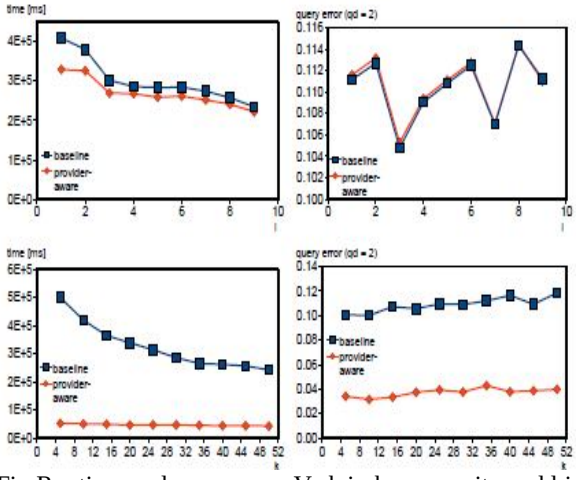


Fig.Runtime and query error Vs k in k anonymity and l in l diversity used in m-privacy

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 361 - 366 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

Adaptive *m*-Privacy Verification. In this experiment, we evaluated the benefit of the adaptive selection of *m*-privacy Verification algorithms. Compares the runtime of adaptive anonymization algorithm with two other *m*-privacy checking strategies with varying *jTj* and constant *nG*. Forsmall values of *jTj*, the algorithm using adaptive verification strategy follows the *binary* and then the *top-down* algorithms, as we expected. However, for values of *jTj* > 300, our algorithm outperforms the non-adaptive strategies. The reason is that anonymization of a large number of records requires verification of *m*-privacy for many subgroups of different sizes. Adapting to such variety of groups is crucial for achieving high efficiency.

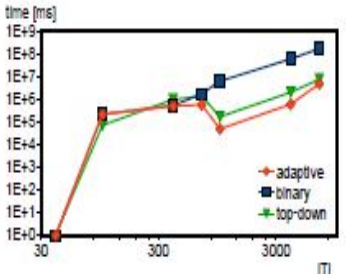


Fig.Runtime of adaptive and non adaptive m-privacy verifications Vs (|T| log-log scale).

CONCLUSION

In this paper, we considered a new type of potential attackers in collaborative data publishing – a coalition of data providers, called m-adversary. To prevent privacy disclosure by any m-adversary we showed that guaranteeing m-privacy is enough. We presented heuristic algorithms exploiting equivalence group monotonicity of privacy constraints and adaptive ordering techniques for efficiently checking m-privacy.

We introduced also a provider-aware anonymization algorithm with adaptive m-privacy checking strategies to ensure high utility and m-privacy of anonymized data. Our experiments confirmed that our approach achieves better or comparable utility than existing algorithms while ensuring m-privacy efficiently.

There are many remaining research questions. Defining a proper privacy fitness score for different privacy constraints is one of them. It also remains a question to address and model the data knowledge of data providers when data are distributed in a vertical or ad-hoc fashion. It would be also interesting to verify if our methods can be adapted to other kinds of data such as set-valued data.

REFERENCES

[1] C. Dwork, "Differential privacy: a survey of results," in Proc. of the 5th Intl. Conf. on Theory and Applications of Models of Computation, 2008, pp. 1–19.

[2] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," ACM Comput. Surv., vol. 42, pp. 14:1–14:53, June 2010

[3] C. Dwork, "A firm foundation for private data analysis," Commun. ACM, vol. 54, pp. 86–95, January 2011.

[4] N. Mohammed, B. C. M. Fung, P. C. K. Hung, and C. Lee, "Centralized and distributed anonymization for high-dimensional healthcare data,"ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 4,no. 4, pp. 18:1–18:33, October 2010.

[5] W. Jiang and C. Clifton, "Privacy-preserving distributed k-anonymity,"in Data and Applications Security XIX, ser. Lecture Notes in Computer Science, 2005, vol. 3654, pp. 924–924.

[6] W. Jiang and C. Clifton, "A secure distributed framework for achieving k-anonymity," VLDB J., vol. 15, no. 4, pp. 316–333, 2006.

[7] Y. Lindell and B. Pinkas, "Secure multiparty computation for privacypreservingdata mining," *The Journal of Privacy and Confidentiality*,vol. 1, no. 1, pp. 59–98, 2009.

[8] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam,"l-diversity: Privacy beyond k-anonymity," in *ICDE*, 2006, p. 24.

[9] P. Samarati, "Protecting respondents' identities in microdata release,"*IEEE T. Knowl. Data En.*, vol. 13, no. 6, pp. 1010–1027, 2001.

[10] L. Sweeney, "k-anonymity: a model for protecting privacy," *Int. J.Uncertain. Fuzz.*, vol. 10, no. 5, pp. 557–570, 2002.

[11] N. Li and T. Li, "t-closeness: Privacy beyond k-anonymity and ldiversity,"in In Proc. of IEEE 23rd Intl. Conf. on Data .

AUTHORS:

N**agaram Lalitha** received the B.Tech degree in Information Technology from JNTU Ananthapur in 2012 & pursuing her M.Tech in Software Engineering from JNTU Kakinada.

S.Amarnath Babu is presently working as Associate Professor Dept of Computer Science and Engineering in St.Ann's College of Engineering and Technology, Chirala. He received the B.Tech degree from Acharya Nagarjuna Bapatla, and M.Tech in JNTU Hyderabad. He has more than 11years of Excellence in Teaching Experience .He Published 6 Conferences and 5 journals.