IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 2, Ver. V (Mar-Apr. 2014), PP 27-33
www.iosrjournals.org

Data attribute security and privacy in distributed database system

Ms. Pragati J. Mokadam[1]
(PKTC, Department of computer science and engineering, Pune university)

Abstract: *Now a days there are a need of data attribute security in distributed database while preserving privacy. In the proposed work, we consider problem related in publishing collaborative data for anonymizing vertically and horizontally partitioned data. We consider the attack which may use a subset of the overall data. After in view of entire research work we make the distributed database system in which first, we introduce the notion of data privacy which guarantees the privacy of anonymized data for different data provider. Second, we present algorithms for exploiting the monotonicity of privacy constraints for checking data privacy efficiently with the encryption schema using encryption algorithm. Third, we distribute the data to end user with the anonymization as well as security algorithm, and checking the authentication schema with TTP, which will give the assurance to provide high level security to database. Experiments we use the hospital patient datasets suggest that our approach achieves better or comparable utility and efficiency than existing and baseline algorithms while satisfying of proposed security work***.**

Keywords: *Distributed database, privacy, protection, security, SMC, TTP*

I. INTRODUCTION

Privacy preservation techniques are mainly used to reduce the leakage of formation about the particular individual while the data are shared and released to public. For this, the sensitive information should not disclose. Data is getting modified first and then published for further process. For this various anonymization techniques are followed and they are generalization, suppression, permutation and perturbation. By various anonymization techniques data is modified which retains sufficient utility and that can be released to other parties safely. Single organization does not hold the complete data. Organizations need to share data for mutual benefits or for publishing to a third party. For banking sector want to integrate their customer data for developing a system to provide better services for its customers. However, the banks do not want to indiscriminately disclose their data to each other for reasons such as privacy protection and business competitiveness.

Main goal is to publish an anonymized view of integrated data, T, which will be immune to attacks(fig1). Attacker runs the attack, i.e. a single or a group of external or internal entities that wants to breach privacy of data using background knowledge. Collaborative data publishing is carried out successfully with the help of trusted third party (TTP) or Secure Multi Party Computation (SMC) protocols, which guarantees that information or data about particular individual is not disclosed anywhere, that means it maintains privacy. Here it is assumed that the data providers are semi honest. A more desirable approach for collaborative data publishing is, first aggregate then anonymize (fig 1)[1].

In above diagram, T1,T2,T3 and T4 are databases for which data is provided by provider like provider P1 provides data for database T1. These distributed data coming from different providers get aggregate by TTP(trusted third party) or using SMC protocol. Then these aggregated data anonymized further by any anonimization technique. P0 is the authenticate user and P1 trying to breach privacy of data which is provided by other users with the help of BK(Background knowledge). This type of attack we can call as a “insider attack”. We have to protect our system from such a type of attacks

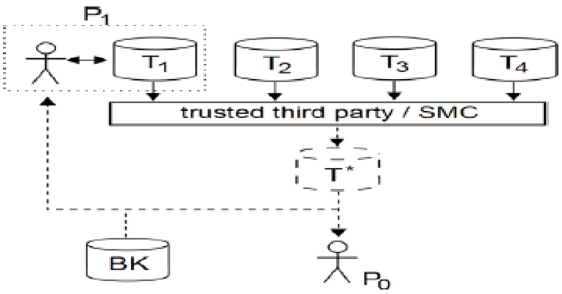


Fig1: Aggregate and Anonymize

We are studying different methods which are previously used for anonymization. We study privacy preserving data publishing (PPDP) [2]and LKC[3] model gives better result than traditional k anonymization model. And also Two party protocol DPP2GA[4], It is only privacy preserving protocol not SMC because it introduce certain inference problem. Many systems use k anonymization for providing privacy. Attacker can attack on anonymized system with the help of BK(background knowledge). L diversity helps to overcome this problem. In current research paper [1], authors introduce a m privacy algorithm which verify anonymization and L diversity. For this they consider generalization and bucketization techniques for maintaining anonimized view of data and also provide L diversity which help to increase privacy of data.

This paper proposed a system in which we used a new technology i.e slicing algorithm with which we also used encrypted data which improves security. Slicing is the process which gives better result than typical generalization and bucketization technique. It gives better results for high dimentional data. It can perform permutation within bucket. In slicing we can collaborate sensitive attribute with any quasi identifier. On this sliced data we use verification algorithms[1] which verifies that whether data is secured or not.

II. ATTACKS ON PRIVACY

1. Background knowledge

In real-life application, privacy protection is impossible due to the presence of the adversary's background knowledge [6]. Suppose the age is sensitive information. Assume an adversary knows that Alina age is 10 years younger than the average age of Indian women, but does not know the average age of Indian women. If the adversary has access to a statistical database that discloses the average age of Indian women, then Alina's privacy is considered to be compromised according to Dalenius definition, regardless of the presence of Alina's record in the database.

2. Linkage models

When an adversary is able to link a record owner to a record in a published data table called record linkage, to a sensitive attribute in a published data table called attribute linkage, or to the published data table itself called table linkage. We assume that the adversary knows the QID(quasi identifier) of the victim. In record and attribute linkages, we further assume that the adversary knows the victim's record is in the released table, and seeks to identify the victim's record and/or sensitive information from the table. In table linkage, the attacker seeks to determine the presence or absence of the victim's record in the released table. A data table is considered to be privacy-preserving if it can effectively prevent the adversary from successfully performing these linkages.

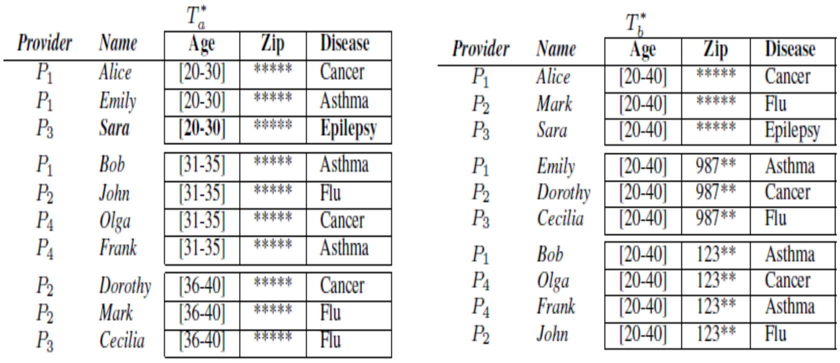
Example: A Record Linkage Model

In the attack of record linkage, some value qid on QID identifies a small number of records in the released table T , called a group. If the victim's QID matches the value qid, the victim is vulnerable to being linked to the small number of records in the group. In this case, the adversary faces only a small number of possibilities for the victim's record, and with the help of additional knowledge, there is a chance that the adversary could uniquely identify the victim's record from the group.

3. Attacks by External Data Recipient Using Anonymized Data:

Each data provider, such as P1 in Table 1[1], can also use anonymized data T∗ and his own data (T1) to infer additional information about other records. Compared to the attack by the external recipient in the first attack scenario, each provider has additional data knowledge of their own records, which can help with the attack.

TABLE I



T_a^*

Provider	Name	Age	Zip	Disease
P_1	Alice	[20-30]	*****	Cancer
P_1	Emily	[20-30]	*****	Asthma
P_3	Sara	[20-30]	*****	Epilepsy
P_1	Bob	[31-35]	*****	Asthma
P_2	John	[31-35]	*****	Flu
P_4	Olga	[31-35]	*****	Cancer
P_4	Frank	[31-35]	*****	Asthma
P_2	Dorothy	[36-40]	*****	Cancer
P_2	Mark	[36-40]	*****	Flu
P_3	Cecilia	[36-40]	*****	Flu

T_b^*

Provider	Name	Age	Zip	Disease
P_1	Alice	[20-40]	*****	Cancer
P_2	Mark	[20-40]	*****	Flu
P_3	Sara	[20-40]	*****	Epilepsy
P_1	Emily	[20-40]	987**	Asthma
P_2	Dorothy	[20-40]	987**	Cancer
P_3	Cecilia	[20-40]	987**	Flu
P_1	Bob	[20-40]	123**	Asthma
P_4	Olga	[20-40]	123**	Cancer
P_4	Frank	[20-40]	123**	Asthma
P_2	John	[20-40]	123**	Flu

Table II gives result of anonymization with QID {Age, Zip} and 3- diversity on sensitive attribute disease by using generalization and bucketization technique. To overcome above type of attacks we proposed a system which discussed further.

III. Privacy Preservation Of Data

We first formally describe our problem setting. Then, we present our data-privacy definition with respect to a privacy constraint to prevent inference attacks by data-adversary, followed by properties of this new privacy notion. Let *T = {t*1, *t*2, . . .*}* be a set of records with the same attributes gathered from *n* data providers *P ={P*1, *P*2, . . . , *Pn},* such that *Ti* 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 [10]. However, for our scenarios we use an approach, which preserves more utility without sacrificing privacy [11]. Our goal is to publish an anonymized *T** while preventing any data-adversary from inferring *AS* for any single record. An data-adversary is a coalition of data users with *n* data providers cooperating to breach privacy of anonymized records. When data are gathered and combined from different data providers, mainly two things are done, for anonymization process.

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: C1ɅC2Ʌ...ɅCw. If a group of anonymized records T* satisfies C, we say C(T*)=true. By definition C(Ø) is true and Ø is private. Any of the existing privacy principles can be used as a component constraint Ci. We now formally define a notion of data-privacy with respect to a privacy constraint C, to protect the anonymized data against data-adversaries. The notion explicitly models the inherent data knowledge of an data-adversary, the data records they jointly contribute, and requires that each QI group, excluding any of those records owned by an data-adversary, still satisfies C.

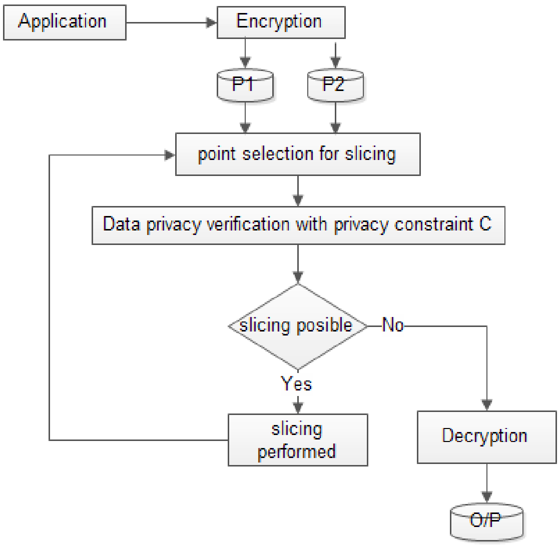

Application
Encryption
P1
P2
point selection for slicing
Data privacy verification with privacy constraint C
slicing posible
No
Yes
slicing performed
Decryption
O/P

Fig 2: Proposed system

Fig.2 shows our proposed system in which input data is given in encrypted format(attribute name will be in encrypted format). Select point for slicing[12]. Check that input data against privacy constraint C for data privacy. Check further is slicing is possible or not. If slicing possible then do it and if not then decrypt data. Our final output T* are anonymized data which will seen only by authenticate user. Any adversary can not breach privacy of data. In this system we are using horizontal as well as vertical partitioning over database. Slicing algorithm provide better column partitioning.

To understand this properly lets consider hospital management system for experiment. Let different departments are the providers who provides data from different sources. We consider disease as a AS (sensitive attribute) and age and zipcode are QI(quasi identifier). We can increase QI and AS also. This will our input data. We can take it in table form as in Table II.

TABLE II

Name	Age	Zip code	Disease	Gender	Password
Alice	22	12345	Cancer	Female	*******
Mark	24	23456	Epilepsy	Male	*******
Sara	25	12365	Flu	Female	*******
Bob	43	23478	High BP	Male	*******
Marry	31	12399	High Bp	Female	*******
Frank	33	23487	Cancer	Male	********
John	15	12356	Flu	Male	********
Siyara	19	23466	Epilepsy	Female	********

On this input data slicing algorithm will apply. Anonymization done on QI and also mantaines diversity for AS. We are using RSA algorithm for encryption. Our anather important attribute is name. we are using encryption on name. When it will save in database, it will in encrypted form. Without key or authentication no one can decrypt it. When any data adversary try to breach data, it will get anonymized sliced data where sensitive attribute maintains diversity and name is in encrypted form.

IV. FLOW OF SYSTEM

1. Mathematical flow of proposed system:

Let, S={s,e,P,T*,F}
Where S is a system of collaborative data publishing consist of database with certain attributes related to patient data for hospital management system. S consist of
s = distinct start of system
e = distinct end of system
P = Input of system from providers
T* = output of system
F = algorithms or functions having certain computation time

Let,
s = {Ru}
// request from user e.g doctor (There can be more than one authenticated user)
P= {DBp1,DBp2…….DBpn}
// database i.e data provided by providers
// Apply F on it.
F = {encryption algorithm(EA), slicing algorithm(SA), binary algorithm(BA), privacy verification algorithm(PA)}
T* = {RuˆDBpn}
// collaborative data according to user request and database which we have. F provides privacy and security to input data.
e = output in table format according to user authentication.
Success condition,
Ru[i] ≠ NULL, DBpn ≠ NULL
Failure condition,
Ru[i] = = NULL, DBpn = = NULL

2. Algorithms:

2.1 Slicing Algorithm:
Definition 1: (Attribute separation and Columns).

In attribute separation, D(database) consists of several subsets, such that each attribute belongs to exactly one subset. Each subset of attributes is called a column. Specifically, let there be C columns C1; C2; . . . Cc, then U(c)i=1,C=D; and for any $1 \leq i_1 \neq i2 \leq c$, $Ci_1 \cap Ci2 = \Theta$.

For simplicity of discussion, we consider only one sensitive attribute S. If the data contain multiple sensitive attributes, one can either consider them separately or consider their joint distribution [25]. Exactly one of the c columns contains S. Without loss of generality, let the column that contains S be the last column C. This column is also called the sensitive column. All other columns{C1,C2…… Cc-1} contain only QI attributes.

Definition 2 : (Tuple Partition and Buckets).

In tuple partition, T consist of several subsets, such that each tuple belongs to exactly one subset. This tuples subset is called a bucket. Specifically, let there be b buckets. B1,B2……B_b then $U^{b}_{i\text{-}1}B_i=T$ and for any $1 \leq i_1 \neq i_2 \leq b$, $Bi_1 \cap Bi_{2} = \theta$

Definition 3 (Slicing):

Given a microdata table T, a slicing of T is given by an attribute partition and a tuple partition.
For example, suppose tables a and b are two sliced tables. In Table a, the attribute partition is {{Age}, {Gender}, {Zipcode}, {Disease}} and the tuple partition is {{t1; t2; t3; t4}, {t5; t6; t7; t8}}. In Table b, the attribute partition is {{Age, Gender}, {Zipcode, Disease}} and the tuple partition is {{t1; t2; t3; t4}, {t5; t6; t7; t8}}.

Definition 4 (Column Generalization)

Given a microdata table T and a column Ci = (Xi1,Xi2,Xi3…….Xij) where Xi1,Xi2…..Xij are attributes, a column generalization for Ci is defined as a set of non overlapping j-dimensional regions that completely cover D[Xi1]* [Xi2] * ….. D[Xij] . A column generalization maps each value of Ci to the region in which the value is contained.

Column generalization ensures that one column satisfies the k-anonymity requirement. It is a multidimensional encoding and can be used as an additional step in slicing. Specifically, a general slicing algorithm consists of the following three phases: attribute partition, column generalization, and tuple partition. Because each column contains much fewer attributes than the whole table, attribute partition enables slicing to handle high-dimensional data. A key notion of slicing is that of matching buckets.

Definition 5 (Matching Buckets)

Consider sliced data and let (C1; C2; . . . ; Cc) be the columns. Let t be a tuple, and t[Ci] be the value of Ci of t. Let B be a bucket in the sliced table, and B[Ci] be the multiset of Ci values in B. We say that B is a matching bucket of t if for all T[C(i)]==B[C(i)] and 1 set if I set of C, 1≤i≤c,t[Ci]ЄB[Ci]

By using above slicing algorithm we can obtain anonymization and l diversity both. This two technique maintains the privacy of data.

2.2 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 ($\sum_{k=0}^{k=n-1}$ verify Kth touple) // 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
```

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 A be a mechanism that anonymizes a group of records T into a single equivalence group, T*=A1(T). A privacy constraint C is EG monotonic if and only if, for a group of records T such that its equivalence group A satisfies C, and any group of records T' , their anonymized union satisfies C, C(A1(T)) = true ⇒∀T',C(A1(T ∪T')) = true.

Before using above binary algorithm we have to check for EG monotonic constraint. First, it starts with $(n-1)$ adversaries, finds the first coalition of attackers that violates privacy, and assigns it to *Psuper* (lines from 4 to 7). Then, it finds an *Psub*, i.e., a sub-coalition of *Psuper*, which does not breach privacy (line 8). At each step, a new coalition *P* : *Psub* *P* *Psuper* (such that |*P*| =|*Psuper*|+|*Psub*|/2 line 12) is checked (line 13). If *P* can breach privacy, then *Psuper* is updated to *P* (line 14). Otherwise, *Psub* is updated to *P* (line 16). The algorithm continues until the direct parent child pairs *Psuper* and *Psub* are found (line 11). Then pruning[1] is performed (lines 17 and 18), and the algorithm starts the next iteration. The algorithm stops when data-privacy can be determined (line 3).

2.3 EG monotonic algorithm:

```
Data: List of providers P, an EG monotonic C, and the n.
Result: true if A1 (T) is data-private w.r.t. C, false otherwise.
1 sites = apply slicing algorithm(P, increasing order , score F )
2 use_adaptive_order_generator(sites, n)
3 while data-privacy decided() = false do
4 Ar = generate_next_coalition(P)
5 Broadcast coalition Ar.
// Runs protected privacy confirmation protocol.
6 privacy breached = is_privacy_breached_by(Ar)
7 if privacy breached and |Ar| _ n then
8 return false // early stop
9 prune_coalitions(Ar, privacy breached)
10 return data-privacy()
```

Privacy for different coalitions of attackers, which are generated in specific order. A general scheme of secure data privacy verification is the same for all heuristic algorithms. Common steps are as follows. In the main loop P' verifies privacy of records for data-adversaries until data-privacy can be decided (line 3). Note that in order to determine data-privacy w.r.t. EG monotonic C, it is enough to check privacy for all scenarios with exactly n attackers. In the loop, P' generates and broadcasts a coalition of potential adversaries Ar, 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 Ar (line 6). If privacy could be breached, and Ar has no more than n 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 data-adversaries (line 9). Finally, if data-privacy w.r.t. C can be decided, then P' returns the results of data-privacy verification (line 10).

3. Result:

By using slicing algorithm and encryption we can get output, Table III. This maintain 4-anonymization with QI {Age, Zipcode} and 3-diversity with {disease(AS),Gender}. Slicing performs permutations within buckets. This system protect our database from attacks like BK, linkage attack as this result provide no linking between tuples and attributes. Name is another important data which is encrypted. Attacker can breach this security and privacy of sensitive data.

TABLE III

Name	{Age, Zip code}	{Disease, Gender}	Password
*******	{22 ,12345}	{High Bp, Female}	*******
*******	{31,12399}	{Cancer, Female}	*******
********	{25 ,12365}	{Flu, Male}	*******
*******	{15,12356}	{Flu, Female}	*******
********	{24,23456}	{Epilepsy, Female}	*******
********	{33,23487}	{Cancer, Male}	********
********	{19,23466}	{High BP, Male}	********
*********	{43, 23478}	{Epilepsy, Male}	********

V. CONCLUSION

We consider a potential attack on collaborative data publishing. We used slicing algorithm for anonymization and L diversity and verify it for security and privacy by using binary algorithm of data privacy. Slicing algorithm is very useful when we are using high dimensional data. It divides data in both vertical and horizontal fashion. Due to encryption we can increase security. But the limitation is there could be loss of data utility.

Above system can used in many applications like hospital management system, many industrial areas where we like to protect a sensitive data like salary of employee. Pharmaceutical company where sensitive data may be a combination of ingredients of medicines, in banking sector where sensitive data is account number of customer, our system can use. It can be used in military area where data is gathered from different sources and need to secured that data from each other to maintain privacy.

This proposed system help to improve the data privacy and security when data is gathered from different sources and output should be in collaborative fashion. In future this system can consider for data which are distributed in ad hoc grid computing. Also the system can be consider for set valued data.

ACKNOWLEDGEMENT

I would like to thank my guide, Professor Dr. S. Singh for their valuable support and share their knowledge.

REFERENCES

[1] S. Goryczka, L. Xiong, and B. C. M. Fung, "m-Privacy for collaborative data publishing," in Proc. of the 7th Intl. Conf. on Collaborative Computing: Networking, Applications and Worksharing, 2011.

[2] 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.

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

[4] W. Jiang and C. Clifton, "Privacy-preserving distributed k-anonymity,"in DBSec, vol. 3654, 2005, pp. 924–924.

[5] W. Jiang and C. Clifton, "A secure distributed framework for achieving k-anonymity," The VLDB Journal Special Issue on Privacy Preserving Data Management, vol. 15, no. 4, pp. 316–333, 2006

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

[7] R. Sheikh, B. Kumar, and D. K. Mishra, "A distributed k-secure sum protocol for secure multi-party computations," J. of Computing, vol. 2, pp. 68–72, March 2010 (2002)

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

[9] P. Jurczyk and L. Xiong, " Distributed anonymization: Achieving privacy for both data subjects and data providers," in DBSec, 2009, pp. 191–207

[10] 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.

[11] Tiancheng Li, Ninghui Li, Jian Zhang,Ian Molloy,"Slicing: A New Approach for Privacy Preserving Data Publishing" IEEE transactions on knowledge and data engineering, vol. 24, no. 3, March 2012

[12] O. Goldreich, Foundations of Cryptography: Volume 2, 2004