IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 2, Issue 2, Apr-May, 2014
ISSN: 2320 – 8791 (Impact Factor: 1.479)
www.ijreat.org

Privacy Preserving High-Dimensional Data Mashup

Megala.K[1], Amudha.S[2]

[1]PG Student, Department of Computer Science and Engineering, Sriram Engineering College, Perumpattu-602 024, Tamil Nadu, India

[2]Assistant Professor, Department of Computer Science and Engineering, Sriram Engineering College, Perumpattu-602 024, Tamil Nadu, India

Abstract

The goal of this project is protecting privacy of online users in social networks. Mashup is integrating different service providers to expertise and to deliver highly customizable services to their customers. Data mashup is an application that aims at integrating data from multiple data providers based on the users request. However, integrating data from multiple sources brings about three challenges:

1. Simply joining multiple private data sets together would reveal the sensitive information to the other data providers.
2. The integrated (mash up) data could potentially sharpen the identification of persons and therefore, expose their person-specific sensitive information that was not available before the mash up.
3. The mash up data from multiple sources often contains many data attributes.

When enforcing a traditional privacy model such as K-anonymity, the high-dimensional data would assist from the problem known as the curse of high dimensionality, resulting in ineffective data for further data analysis. This paper resolves a privacy problem in a real-life mashup application for the online advertising industry in social networks, and proposes a service-oriented architecture along with a privacy-preserving data mashup algorithm to address the aforementioned challenges.

Index Terms—*Privacy protection, anonymity, data mashup, data integration, service-oriented architecture, high dimensionality.*

1. INTRODUCTION

MASHUP service is a web technology that combines various information from multiple sources into a single web application. An example of a successful mash up application is the combination of real estate information into Google Maps, which allows users to browse on the map for properties that satisfy their specified requirements. Developers create mashups by combining components of existing Web sites and applications. Mashup combine views, data, and logic from existing Web sites or applications to create novel applications that focus on situational and passing problems. This paper focuses on data mash up, a special type of mash up application that aims at integrating data from multiple data providers depending on the service request from a user (a data beneficiary).

An information service request can be a common count statistic task or a stylish data mining task such as classification analysis. Conceptually, mashups are simply new Web applications that repurpose alive Web data and APIs. Well-structured mashups therefore include all three aspects of an equivalently well designed Web application, data models, views, and interaction controllers. Also, mashups often intervene between mixed providers Web APIs. The advertisements are posted to the user account by the admin or the mash up coordinator depending upon the category of the user. This paper use one more special attribute in the registration form. Generally the social network websites registration form consist of some basic details during signup like Name, Age, Gender, Username etc., This paper proposes hiding option is also enable for protecting the sensitive information.

2. THE CHALLENGES

The problem can be generalized as follows: social network a company A and B observe different sets of attributes about the same set of individuals (members) identified by the common User ID. Every time a social network member visits another member's webpage, an advertisement is chosen to be displayed. Companies A and B want to implement a data mashup application that integrates their membership data, with the goal of improving their advertisement selection strategy. The analysis includes gathering general count statistics and building classification models. In addition to companies A and B, other partnered advertising companies need access to the final mashup data. The solution presented in this paper is not limited only to the social networks sector but is also applicable

to other similar data mashup scenarios. The challenges of developing the data mashup application are summarized as follows.

Challenge 1: Privacy concerns

The members are willing to submit their personal data to a social network company because they consider the company and its developed system to be trustworthy. Yet, trust to one party may not necessarily be transitive to a third party. Many agencies and companies believe that privacy protection means simply removing explicit identifying information from the released data, such as name, social security number, address, and telephone number. However, many previous works show that removing explicit identifying information is insufficient. An individual can be re-identified by matching other attributes called quasi-identifiers (QID). There are two types of privacy threats:

i. Record linkage

ii. Attribute linkage.

The data mashup problem further complicates the privacy issue because the data are owned by multiple parties. In addition to satisfying a given privacy requirement in the final mashup data, at any time during the process of generalization no data provider should learn more detailed information about any other data provider other than the data in the final mashup table. In other words, the generalization process must not leak more specific information other than the final mashup data.

Challenge 2: High dimensionality

The mashup data from multiple data providers usually contain many attributes. Enforcing traditional privacy models on high-dimensional data would result in significant information loss. As the number of attributes increases, more generalization is required in order to achieve K-anonymity even if K is small, thereby resulting in data useless for further analysis.

Challenge3: Information requirements

The data recipients want to obtain general count statistics from the mashup membership information. Also, they want to use the mashup data as training data for building a classification model on the Class attribute, with the goal of predicting the behaviour of future members. One frequently raised question is: to avoid privacy concerns, why doesn't the data provider release the statistical data or a classifier to the data recipients? In many real-life scenarios, releasing data is preferable to releasing statistics for several reasons. First, the data providers may not have in-house experts to perform data mining. They just want to share the data with their partners. Second, having access to the data, data recipients are flexible to perform the required data analysis. It is impractical to continuously request data providers to produce different types of statistical information or to fine-tune the data mining results for research purposes for the data recipients

3. CONTRIBUTIONS

This paper is the first work that addresses all the aforementioned challenges in the context of mashup service. The contributions are summarized as follows.

Contribution 1

The new privacy problem through a collaboration with the social networks industry and generalize the industry's requirements to formulate the privacy-preserving high-dimensional data mashup problem. The problem is to dynamically integrate data from different sources for joint data analysis in the presence of privacy concerns.

Contribution 2

The Service-oriented architecture introduced for privacy-preserving data mashup in order to securely integrate private data from multiple parties. The generalized data have to be as useful as possible to data analysis. Generally speaking, the privacy goal requires anonymizing identifying information that is specific enough to pinpoint individuals, whereas the data analysis goal requires extracting general trends and patterns. If generalization is carefully performed, it is possible to anonymize identifying information while preserving useful patterns.

Contribution 3

Data mashup often involves a large volume of data from multiple data sources. Thus, scalability plays a key role in a data mashup system. After receiving a request from a data recipient, the system dynamically identifies the data providers and performs the data mashup. Experimental results on real-life data suggest that our method can effectively achieve a privacy requirement without compromising the information utility, and the proposed architecture is scalable to large data sets.

4. EXISTING SCENARIO

A data mash up application can help ordinary users explore new knowledge; it could also be misused by adversaries to reveal sensitive information that was not available before the mash up. High dimensionality is a critical obstacle for achieving effective data mash up because the integrated data from multiple parties usually contain many attributes. Enforcing traditional K-anonymity on high-dimensional data will result in significant information loss.
First define the LKC-privacy model and the information service measure on a single data table, then extend it for privacy-preserving high-dimensional data mashup from multiple parties.

a. Isolation Measure

Consider a relational data table $T(UID,D_1,...,D_m,S_1,...,S_e,Class)$ (e.g., Table 1). UID is an explicit identifier, such as User ID or SSN. In practice, it should be replaced by a pseudo identifier, such as a record ID, before publication. Each D_i is either a categorical or numerical attribute. Each S_j is a categorical sensitive attribute. A record has the form $(v_1,..., vm, s1,...,s_e,cls)$, where v_i is a domain value in D_i, s_j is a sensitive value in S_j, and cls is a class value in Class. The data provider wants to protect against linking an individual to a record or some sensitive value in T through some subset of attributes called a quasi-identifier QID $\subseteq \{D_1,..., Dg\}$.

One data recipient, who is an adversary, seeks to identify the record or sensitive values of some target victim V in T. Assume that the adversary knows at most L values of QID attributes of the victim. The qid denotes such prior known values, where $|qid| \leq L$. Based on the prior knowledge qid, the adversary could identify a group of records, denoted by T[qid] that contains qid. $|T[qid]|$ denotes the number of records in T[qid]. The adversary could launch two types of privacy attacks based on T[qid].

i. Record linkage
ii. Attribute linkage

b. Service Measure

The measure of information utility varies depending on the user's specified information service request and the data analysis task to be performed on the mashup data. Based on the information requirements specified by the social network data providers, we define two utility measures. The first aim is preserving the maximal information for classification analysis. Second, minimizing the overall data distortion when the data analysis task is unknown.

i. Service Measure for Classification Analysis.
ii. Service Measure for data analysis.

c. Privacy-Preserving High-Dimensional Data Mashup

Consider n data providers(y) and they are having own data table (T_y).T_y is the type of attribute and it also having 4 parameters namely { QID (Quasi-Identifying attribute), UID(User identification),S(Sensitive Information),Class } for the each data providers. These parameters are considered as the same set of records for different data providers. UID and Class are the shared attributes among all data providers. For example consider two data providers as y and z and Their QID and S factor is QID_y, QID_z,, S_y, S_y respectively. The privacy preserving algorithm helps us to check the values of these factors. If Quasi factors of y and z are different means the sensitive information cannot be accessed between them. This algorithm only provides the information about the user when only if the UID information is match with another data provider's UID value. This algorithm also provides the minimal information with the help of LKC privacy requirement on mashup table. Using this minimal information the mashup coordinator chooses the type of details about the user.i.e whether the information is local or global. If its local means all the values of QID_j are known by the one provider, else it is declared as global.

5. PROPOSED SYSTEM

The mash up coordinator receives an information service request from the data recipient and establishes connections with the data providers who can contribute their data to fulfill the request.

The mash up coordinator executes the privacy-preserving algorithm to integrate the private data from multiple data providers and to deliver the final mash up data to the data receiver. Note that the proposed solution does not require the mash up coordinator to be a trusted party.

Though the mash up coordinator manages the entire mash up service, our solution guarantees that the mash up coordinator does not gain more information than the final mash up data, thereby protecting the data privacy of every participant by using hide details option in the starting phase of the process. The mash up coordinator can be any one of the data providers

or an independent party. This makes our architecture realistic for the reason that a trusted party is not always available in real-life mash up scenarios. Two social networks are created in the proposed work. Using these networks we provide the high dimensional security at the registration phase itself and also provide the advertisement to the data recipients depending upon the user category.

SOA for Privacy Preserving Data Mashup

Figure 1 describes the architecture design of the privacy preserving confidential data mash up model. The data recipient containing the browser view model and the data mining model. Data mining model is used to extract the details of the customer for integration. The mash up coordinator containing the web services and session. Each data holder must connect with the session of the mash up coordinator part. Web services provides the different kind of services related with the aim of the process and stored in the private database.

The data mash up process can be divided into two phases. In Phase I, the mash up coordinator receives the service request from the data recipient and establishes connections with the data provider who can contribute their data to fulfill the request. In Phase II, the mash up coordinator executes the privacy-preserving algorithm to integrate the private data from multiple data providers and to deliver the final mash up data to the data receiver. Note that the proposed solution does not require the mash up coordinator to be a trusted party. Though the mash up coordinator manages the entire mash up service, the solution guarantees that the mash up coordinator does not gain more details than the final mash up data, thereby shielding the data privacy of every participant.

The mash up service have the following merits.
They are,

1. Increase productivity
2. Increase innovation
3. Improve data security
4. Reduce burden to IT departments/increase freedom for business users from IT
5. Increase standardization across the enterprise
6. Bring an "App Store" model approach to development, vs. big-bang project.

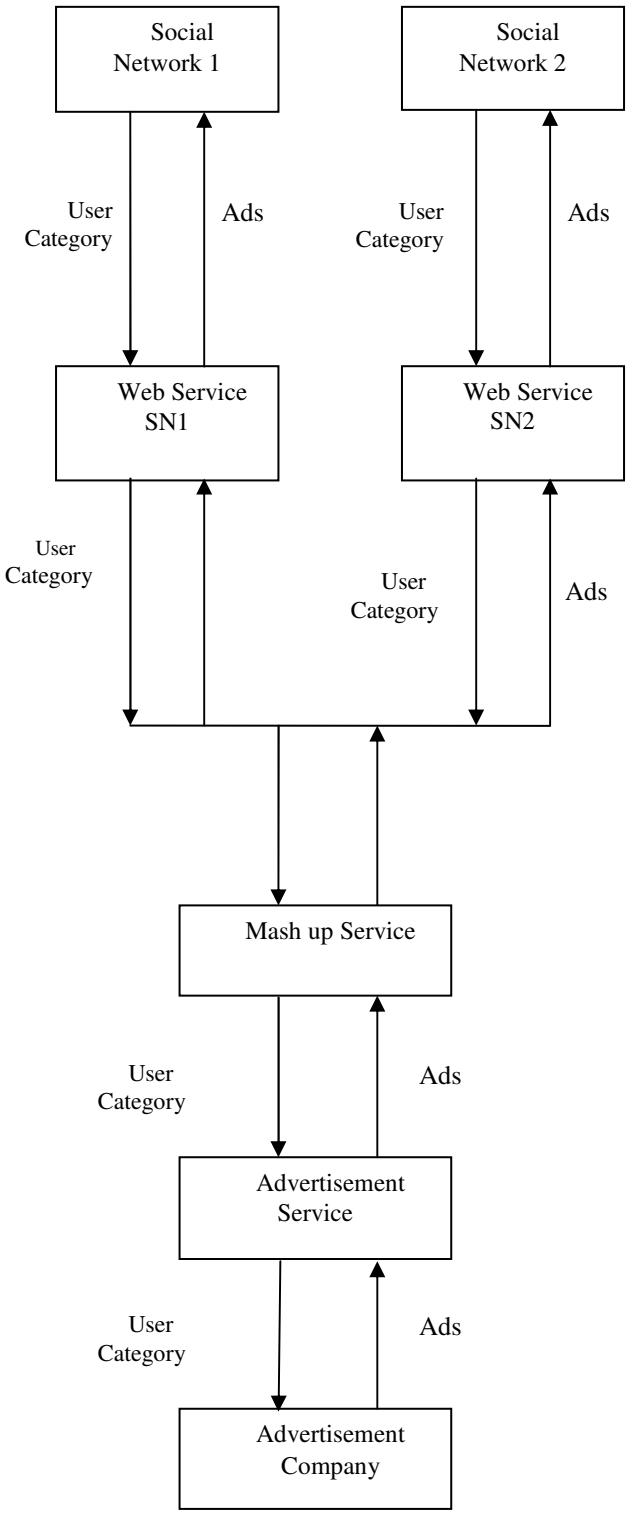

Figure 1: Service-oriented architecture for PHD Mashup

Phase I: Session Establishment

The objective of Phase I is to establish a common session context between the data recipient and the contributing data providers. An operational context is successfully established by proceeding through the steps of data recipient authentication, contributing data provider's identification, session context initialization, and common requirements negotiation.

Authenticate data recipient: The mashup coordinator first authenticates a data recipient to the requested service, generates a session token for the current recipient interaction, and then identifies the data providers accessible by the data recipient. Some data providers are public and are accessible by any data recipients.

Identify contributing data providers: Next, the mashup coordinator queries the data schema of the accessible data providers to identify the data providers that can contribute data for the requested service. To facilitate more efficient queries, the mashup coordinator could prefetch data schema from the data providers (i.e., the pull model), or the data providers could update their data schema periodically (i.e., the push model).

Initialize session context: Next, the mashup coordinator notifies all contributing data providers with the session identifier. All prospective data providers share a common session context that represents a stateful presentation of information related to a specific execution of the privacy preserving mashup algorithm called PHDMashup. An established session context contains several attributes to identify a PHDMashup process, including the data recipient's address; the data providers' addresses and certificates; an authentication token that contains the data recipient's certificate; and a unique session identifier that uses an end-point reference (EPR) composed of the service address, a PHDMashup process identifier and runtime status information about the executed PHDMashup algorithm.

Negotiate privacy and information requirements: The mashup coordinator is responsible to communicate the negotiation of privacy and information requirements among the data providers and the data recipient. Specifically, this step involves negotiating cost, LKC-privacy requirement, sensitive information, and expected information quality. For example, in the case of classification analysis, information quality can be estimated by classification error on some testing data.

Phase II: Privacy-Preserving High-Dimensional Data Mashup

PHDMashup algorithm use to evaluate the impact on classification quality. A service oriented architecture (SOA) that describes the communication paths of all participating party, followed by a privacy-preserving high-dimensional confidential data mash up algorithm that can efficiently identify a suboptimal resolution for the problem. SOA is an architectural model for developing and integrating heterogeneous information systems with strict message-driven communication model. Following the SOA design principles, the resulting system has several attractive properties including interoperability and loosely coupling. Interoperability means capability of allowing platform-independent design of the system components based on a common understanding of service component and interfaces. Loosely coupling refers to the capability of minimizing dependencies among the system components and therefore, improving the overall elasticity, scalability, and fault tolerance of a system. This paper, describes data sources can be dynamically composed to serve new mashup requests depending on the data analysis tasks and privacy requirements. SOA having the capabilities of interoperability and loosely coupling has become a natural choice to tackle the heterogeneity of different potential data providers.

1:initialize T_g to embrace one testimony containing topmost values,
2: initialize $UCut_i$ to embrace only topmost values and update IsValid (s) for every s ε $UCut_i$
3: while v ε $UCut_i$ s.t. IsValid(s) do
4: find the local winner β that has the highest Score(β),
5: communicate Score(β) with provider B to determine the global winner z,
6: if the winner z is local then
7: specialize z on T_g,
8: instruct provider B to specialize z,
9: else
10: wait for the instruction from providerB.
11: specialize z on T_g using the instruction,
12: end if
13: replace w with child(z) in the local copy of $UCut_i$,
14: update Score(s) and IsValid(s) for every candidate s ε $UCut_i$, 15: end while
16: return T_g and $UCut_i$

Figure 2: Algorithm PHDMashup for Provider A (Same as Provider B)

The nature of the top-down approach implies that Tg is always more general than the final mash up table and therefore, does not violate necessities. At each iteration, the data provider

cooperate to perform the same identified specialization by communicating some count statistics information that satisfies necessities. Below, describes about the key steps: find the winner contender (Lines 4-5), perform the winner specialization (Lines 7-11), and update the score and status of contenders (Line 14). For provider A, a local attribute refers to an attribute from T_A.

6. EMPIRICAL STUDY

The objectives of the empirical study are to evaluate the benefit of data mashup for joint data analysis, and the impacts of anonymization and dimensionality on the data quality with respect to the information requirements.

Benefits of Mashup

A trivial yet incorrect solution to avoid privacy concerns is to not integrate the data; each data provider simply performs the classification analysis on its own attributes and releases the data mining result, such as the classifier, to the data recipient. The first goal is to illustrate the benefit of data mashup over this trivial solution with respect to the classification requirement.

To evaluate the impact on classification quality, use all records for anonymization, build a C4.5 classifier on 2/3 of the anonymized records as the training set (30,162 records), and measure the classification error on 1/3 of the anonymized records as the testing set (15,060 records). Both the training and testing steps use all 14 attributes. Lower classification error means better data quality. To collect the two types of classification errors from the testing set: Mashup Classification Error (MCE) is the error on the mashup data produced by our PHDMashup algorithm. Source error (SE) is the error on individual raw data table without generalization. SE for T_A, denoted by SE(A), is 17.7 percent and SE for T_B, denoted by SE(B), is 17.9 percent. SE- MCE measures the benefit of data mashup over individual private table.

Fig. 3 depicts the MCE for the adversary's prior knowledge L ¼ 2, L ¼ 4, and L ¼ 6 with confidence threshold C ¼ 20% and anonymity threshold K ranging from 20 to 100. For example, MCE ¼ 16:3% for L ¼ 4 and K ¼ 60, suggesting that the benefit of mashup, SE-MCE, is approximately 1.5 percent. This experiment demonstrates the benefit of data mashup over a wide range of privacy requirements. The benefit for all test cases illustrated in Fig. 3 spans from 1.3 to 2.1 percent. The benefit decreases as L increases because more generalization is required in order to thwart the linkage attacks. In practice, the benefit is more than the accuracy consideration because our method allows the participating data providers to share data for joint data analysis, rather than sharing a classifier from each provider.

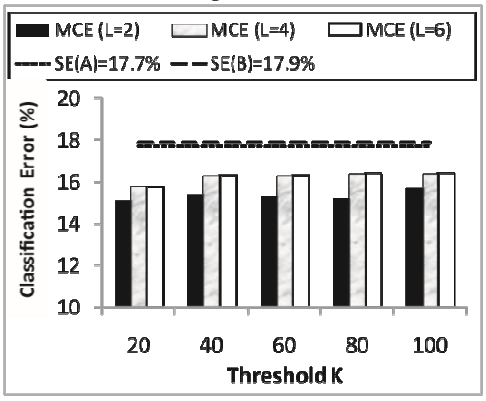

Fig 3 Benefits of mashup (C=20%)

Impacts of Anonymization

The Second goal is to illustrate the impacts for achieving LKC-privacy with respect to classification analysis and general data analysis.

To evaluate the impacts on classification quality to Collect several classification errors, in addition to MCE, from the testing set: Baseline Error (BE) is the error measured on all 14 raw data attributes without generalization. BE-MCE represents the cost in terms of classification quality for achieving a given LKC-privacy requirement. A naive method to avoid record and attributes linkages is to simply remove all QID attributes. Thus, by measuring Upper bound Error (UE), which is the error on the raw data with all QID attributes removed. UE-MCE represents the benefit of proposed method over the naive approach. The experimental results of this method are as follows

Fig. 4 depicts the MCE for the adversary's prior knowledge L ¼ 2, L ¼ 4, and L ¼ 6 with confidence threshold C ¼ 20% and anonymity threshold K ranging from 20 to 100. For example, at L ¼ 4, K ¼ 60, and C ¼ 20, MCE ¼ 16:3%. The cost is MCE-BE ¼ 1:6%, where BE ¼ 14:7%. The benefit is UE-MCE ¼ 8:3%, where UE ¼ 24:6%. For all test cases in Fig. 4, the cost MCE-BE spans from 0.4 percent to 1.7 percent and the benefit UE-MCE spans from 8.2 to 9.5 percent. This result illustrates that the cost of anonymization is low and the benefit of anonymization is high, suggesting that accurate classification and privacy protection can coexist even for a wide range of anonymity threshold K. Typically, there are redundant classification patterns in the data. Though generalization may eliminate some useful patterns, other patterns emerge to help the classification task.

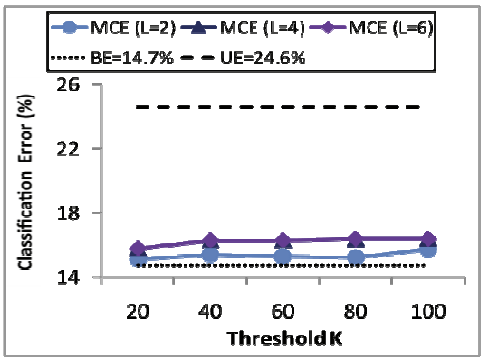

Fig. 4. Impacts on classification analysis (C ¼ 20%).

Impacts of Dimensionality

The third goal is to evaluate the impact of dimensionality, i.e., the number of QID attributes, on the data quality with respect to the distortion metric proposed in [42]. Each time a categorical value is generalized to the parent value in a record, there is one unit of distortion. For a numerical attribute, if a value v is generalized to an interval (a, b), there is (b-a)/ (f2, f1) unit of distortion for a record containing v, where (f1,f2) is the full range of the numerical attribute. The distortion is normalized by the number of records. The distortion per record (DPR) is separately computed for categorical attributes and numerical attributes, denoted by DPR_Categorical and DPR_Numerical, respectively.

Fig. 5 depicts the DPR Categorical and DPR Numerical for the adversary's prior knowledge L ¼ 4 with confidence threshold C ¼ 20% and anonymity threshold K ¼ 60 for 4, 7, 10, and 13 QID attributes. DPR Categorical spans from 3.98 to 11.24 and DPR Numerical spans from 0.62 to 4.05. This result illustrates that the distortion per record generally increases as the number of QID attributes increases because more generalizations are required in order to achieve the same LKC-privacy requirement.

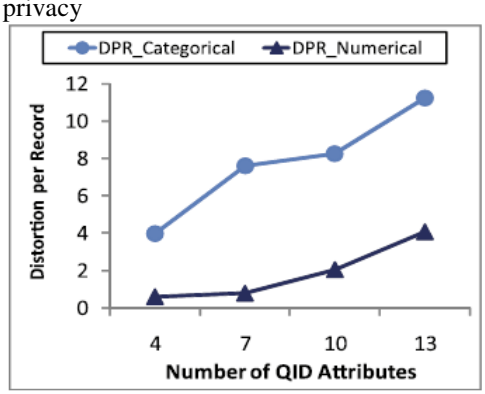

Fig 5, Impacts of dimensionality (L=4, K=60, and C=20%)

Efficiency and Scalability

The proposed method takes at most 20 seconds for every previous experiment. Out of the 20 seconds, approximately 8 seconds is spent on initializing network sockets, reading data records from disk, and writing the generalized data to disk. The actual costs for data anonymization and network communication are relatively low.

The other claim is the scalability of handling large data sets by maintaining count statistics instead of scanning raw records. In order to evaluate this claim on an enlarged version of the Adult data set and to combine the training and testing sets, giving 45,222 records, and for each original record r in the combined set, and create α-1 variations of r, where α> 1 is the blowup scale. Together with original records, the enlarged data set has α×45,222 records.

7. SUMMARY

The experiments verified several claims about the PHDMashup algorithm. First, data mashup leads to improved information utility compared to the information utility separately available on each private table. Second, PHDMashup achieves a broad range of LKC- privacy requirements without significantly sacrificing the information utility. The cost for anonymization is low, and the benefit is significant. Third, our proposed architecture and method are scalable for large data sets. Our work provides a practical solution to the problem of high-dimensional data mashup with the dual goals of information sharing and privacy protection.

8. RESULTS AND DISCUSSION

In this information structural design to create the two social networks. The web services are created by the help of the user category that are enrolled by the user during registration. In registration they give their entire information to the social network and then these details are submitted to the data providers' .In existing information the entire details about the user is viewed by the all data providers. So it may be threaten to the user to overcome this problem here we use the mash up algorithm with k-anonymity model for providing security.

To apply a data mashup function for the online advertising industry in social networks, and generalize their privacy and information requirements to the problem of privacy preserving data mashup for the purpose of joint data analysis on the high-dimensional data.

9. CONCLUSION

By implementing a data mashup application for the online advertising industry in social networks, and generalize their privacy and information requirements to the problem of privacy-preserving data mashup for the purpose of joint data analysis on the high-dimensional data. In order to formalize this problem as achieving the LKC-privacy on the mashup data without revealing more detailed information in the process. Just by presenting a solution and evaluate the benefits of data mashup and the impacts of generalization. Compared to classic secure multiparty computation, a unique feature of our method is to allow data sharing instead of only result sharing. This feature is especially important for data analysis that requires user interaction. Being able to share data records would permit such exploratory data analysis and explanation of results.

Finally, it is better to share the experience of collaboration with industrial practitioners. In general, industrial practitioners prefer a simple privacy model that is intuitive to understand and to explain to their clients, such as LKC-privacy. Often, their primary concern is whether or not the anonymous data are still effective for data analysis; solutions that solely satisfy some privacy requirement are insufficient. The industry demands anonymization methods that can preserve information for various data analysis tasks.

REFERENCES

[1] B.C.M. Fung, K. Wang, and P.S. Yu, "Anonymizing Classification Data for Privacy Preservation," IEEE Trans. Knowledge and Data Eng., vol. 19, no. 5, pp. 711-725, May 2007.

[2] L. Sweeney, "Achieving k-Anonymity Privacy Protection Using Generalization and Suppression," Int‟l J. Uncertainty, Fuzziness, and Knowledge-Based Systems, vol. 10, no. 5, pp. 571-588, 2002.

[3] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "ℓ -Diversity: Privacy Beyond k-Anonymity," ACM Trans. Knowledge Discovery from Data, vol. 1, no. 1, Mar. 2007.

[4] P. Samarati, "Protecting Respondents‟ Identities in Microdata Release," IEEE Trans. Knowledge and Data Eng., vol. 13, no. 6, pp. 1010-1027, Nov. 2001.

[5] K. LeFevre, D.J. DeWitt, and R. Ramakrishnan, "Workload-Aware Anonymization," Proc. 12th ACM Int‟l Conf. Knowledge Discovery and Data Mining (SIGKDD), Aug. 2006.

[6]C.C. Aggarwal, "On k-Anonymity and the Curse of Dimensionality," Proc. 31st Very Large Data Bases, pp. 901-909, 2005.

[7] B.C.M. Fung, K. Wang, R. Chen, and P.S. Yu, "Privacy-Preserving Data Publishing: A Survey of Recent Developments," ACM Computing Surveys, vol. 42, no. 4, pp. 14:1-14:53, June 2010.

[8] K. Wang, B.C.M. Fung, and P.S. Yu, "Handicapping Attacker's Confidence: An Alternative to k-Anonymization," Knowledge and Information Systems, vol. 11, no. 3, pp. 345-368, Apr. 2007.