Privacy for User Behavior Data

PERAM VARDHANI[1], PENUMUDI YAMUNASAI[2], POPURI BINDU SRI[3]
[1,2,3] *Computer Science and Engineering, Vasireddy Venkatadri Institute of Technology*

Abstract- ***Tons of online user behavior data are being generated every day on the booming and ubiquitous Internet. Growing efforts have been devoted to mining the abundant behavior data to extract valuable information for research purposes or business interests. However, online users' privacy is thus under the risk of being exposed to third-parties. The last decade has witnessed a body of research works trying to perform data aggregation in a privacy-preserving way. Most of existing methods guarantee strong privacy protection yet at the cost of very limited aggregation operations, such as allowing only summation, which hardly satisfies the need of behavior analysis. In this paper, we propose a scheme PPSA, which encrypts users' sensitive data to prevent privacy disclosure from both outside analysts and the aggregation service provider, and fully supports selective aggregate functions for online user behavior analysis while guaranteeing differential privacy. We have implemented our method and evaluated its performance using a trace-driven evaluation based on a real online behavior dataset. Experiment results show that our scheme effectively supports both overall aggregate queries and various selective aggregate queries with acceptable computation and communication overheads.***

I. EXISTING SYSTEM

Privacy-preserving aggregation on sensitive user data has raised much attention recently, including health care, time-series data, wireless sensor network data, and online behavior data for analysis an advertising. In general, there are two types of systems in previous work.

Centralized Systems: - In a centralized system, all the user data are stored on the server. It is important that users encrypt or encode their data before sending them to the server. The server holds the encrypted data, but it can only compute answers to queries obliviously, e.g., [37]–[39]. However, these proposals have different goals than our system and do not support selective aggregation. .Moreover, they do not guarantee differential privacy. Homomorphic encryption is a common method to achieve aggregation of encrypted data without decryption, such as. Chen et al. [42] used an order preserving hash-based function to encode both data and queries instead. But they do not have the same goal as us and cannot evaluate selective aggregation. Li et al. [43] proposed a system that processes range queries, which yet does not compute aggregation and assumes analysts to be trusted. On the contrary, PPSA combines differential privacy and Homomorphic encryption, and is able to selectively aggregate encrypted user data.

Distributed Systems:- In a distributed system, clients need to proactively, or passively send required data to the aggregator in a private way. But both rely on the participation of clients. These systems all require online users, so analysis cannot go on when most of the users are offline. Homomorphic encryption is also applicable in distributed system. For instance, PASTE exploits differential privacy and homomorphic cryptography but it allows only summation of user data and the aggregator knows the private key.

Castelluccia et al. use symmetric homomorphic encryption so they need a trusted aggregator, and they also allow only additive aggregation. DJoin aims to support distributed and differentially private query answering service, but it applies to privacy-preserving data join between two parties, which is a different scenario with ours. Secure Multi-Party Computation (SMC) requires that all participants must be simultaneously online and interact with each other periodically, which is infeasible for practical scenarios. Some previous researches have noticed the problem of client churn.

For instance, Shi et al.[12] proposed a system that can tolerate some offline clients, but a trusted dealer and a trusted initial setup phase between all participants and the aggregator are still needed. Rottondi et al. [47] also discussed node failures but they addressed a different issue from the problem in this paper.

Disadvantages: - There is no analyst queries that the aggregation of an attribute selected by multiple different Boolean attributes. Less security on Data Attributes

PROPOSED SYSTEM

In the proposed system, the system has described the challenges of making online user data aggregation while preserving users' privacy. Based on BGN homomorphic cryptosystem, we have designed the first system that is able to securely and selectively aggregate user data, making it practical in realistic data analytics. It guarantees strong privacy preservation by utilizing differential privacy mechanism to protect individuals' privacy. The system has presented PPSA to evaluate aggregation selected by one boolean attribute, and extended it to aggregation selected by multiple boolean attributes and by one numeric attribute. Extensive experiments have shown that PPSA supports various selective aggregate queries with acceptable overhead and high accuracy.

Advantages:-

- The system implemented PPSA and does a trace-driven evaluation based on an online behavior dataset.
- Privacy-Preserving Overall Aggregation Algorithm.
-

SYSTEM REQUIREMENTS

➢ H/W System Configuration

➢ Processor - Pentium –IV

➢ RAM - 4 GB (min)

➢ Hard Disk - 20 GB

➢ Key Board - Standard Windows Keyboard

➢ Mouse - Two or Three Button Mouse

➢ Monitor - SVGA

Software Requirements:

Operating System - Windows XP

Coding Language - Java/J2EE (JSP,Servlet)

Front End - J2EE

Back End - MySQL

II. INTRODUCTION

Online user behavior analysis studies how and why users of e-commerce platforms and web applications behave. It has been widely applied in practice, especially in commercial environments, political campaigns, and web application development [1]–[3].

Data aggregation is one of the most critical operations in behavior analysis. Nowadays, the aggregation tasks for user data are outsourced to third-party data aggregators including Google Analytics, comScore, Quantcast, and StatCounter. While this tracking scheme brings great benefits to analysts and aggregators, it also raises serious concerns about disclosure of users' privacy [4]. Aggregators hold detailed data of users' online behaviors, from which demographics can be easily inferred [5]. To protect users' privacy, government and industry regulations were established, e.g., the EU Cookie Law [6] and W3C

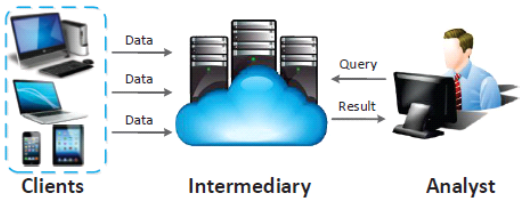

Fig. 1: **System overview:** Clients are installed on user side. The intermediary collects data from clients, computes aggregate statistics, and answers queries issued by the analyst. The intermediary should also ensure users' privacy is not leaked.

Do-Not-Track [7], which significantly restricts the analysis of users' online behaviors [4]. To address the conflict between the utility of analysis results and users' privacy, much effort has been devoted to designing protocols that allow operations on user data while still protecting users' privacy (e.g., [4], [8]–[14]). Unfortunately, existing schemes guarantee strong privacy at the expense of limitations on analysis. Most of them can only compute summation and mean of data over all users without filter or selection, i.e., overall aggregation. Some previous methods allow more complex computations [13]–[15]. For instance, Jung et al. [13] proposed a system that can perform multivariate polynomial evaluation. Unfortunately, they still do not support selection. However, selective aggregation is one of the most important operations for queries on databases. It can be used to tell the difference among different user groups in a certain aspect. For instance, "select avg(income) from database group by gender".

As shown in Fig. 1, a typical privacy-preserving data aggregation system is composed of three parts: clients, intermediary (i.e., aggregation service provider) and

analyst. The intermediary collects data from clients (users' devices), does some calculations and evaluates aggregate queries issued by the analyst. A common assumption made in many existing systems is that the intermediary is not trusted.

Adding noise to the aggregate result is a common method to achieve stronger privacy preservation, differential privacy [16], without which individuals' privacy can be easily inferred from aggregate results by adversaries who have computational power and auxiliary information. There are two ways to add noise: either each client adds noise to its own data (e.g., [8], [12], [17]), or the intermediary obliviously adds noise to the aggregate result (e.g., [4], [10], [11]). PPSA adopts the latter way to achieve differential privacy, in which noise needs to be added obliviously so as to prevent the intermediary from determining the noise-free result when the noisy result is publicly released.

Adding noise to the aggregate result is a common method to achieve stronger privacy preservation, differential privacy [16], without which individuals' privacy can be easily inferred from aggregate results by adversaries who have computational power and auxiliary information. There are two ways to add noise: either each client adds noise to its own data (e.g., [8], [12], [17]), or the intermediary obliviously adds noise to the aggregate result (e.g., [4], [10], [11]). PPSA adopts the latter way to achieve differential privacy, in which noise needs to be added obliviously so as to prevent the intermediary from determining the noise-free result when the noisy result is publicly released. We present the first scheme PPSA that allows privacy-preserving selective aggregation on user data, which plays a critical role in online user behavior analysis.

We combine homomorphic encryption and differential privacy mechanism to protect users' sensitive information from both analysts and aggregation service providers, and protect individuals' privacy from being inferred. We prove that differential privacy can be achieved by adding two Geometric variables, which is computed via homomorphic encryption. Furthermore, we present a privacy analysis of PPSA.

We extend PPSA to two more scenarios to fully support more complex selective aggregation of user data. We utilize a calculation to evaluate aggregation selected by multiple Boolean attributes. We design a way of oblivious comparison between two integers, and utilize it to evaluate aggregation selected by a numeric attribute.

We implement PPSA and do a trace-driven evaluation based on an online behavior dataset. Evaluation results show that our scheme effectively supports various selective aggregate queries with high accuracy and acceptable computation and communication overheads.

Client Server

Over view:

With the varied topic in existence in the fields of computers, Client Server is one, which has generated more heat than light, and also more hype than reality. This technology has acquired a certain critical mass attention with its dedication conferences and magazines. Major computer vendors such as IBM and DEC, have declared that Client Servers is their main future market. A survey of DBMS magazine revealed that 76% of its readers were actively looking at the client server solution. The growth in the client server development tools from $200 million in 1992 to more than $1.2 billion in 1996.

Client server implementations are complex but the underlying concept is simple and powerful. A client is an application running with local resources but able to request the database and relate the services from separate remote server. The software mediating this client server interaction is often referred to as MIDDLEWARE.

The typical client either a PC or a Work Station connected through a network to a more powerful PC, Workstation, Midrange or Main Frames server usually capable of handling request from more than one client. However, with some configuration server may also act as client. A server may need to access other server in order to process the original client request.

The key client server idea is that client as user is essentially insulated from the physical location and formats of the data needs for their application. With the proper middleware, a client input from or report can transparently access and manipulate both local

database on the client machine and remote databases on one or more servers. An added bonus is the client server opens the door to multi-vendor database access indulging heterogeneous table joins.

What is a Client Server?

Two prominent systems in existence are client server and file server systems. It is essential to distinguish between client servers and file server systems. Both provide shared network access to data but the comparison dens there! The file server simply provides a remote disk drive that can be accessed by LAN applications on a file by file basis. The client server offers full relational database services such as SQL-Access, Record modifying, Insert, Delete with full relational integrity backup/ restore performance for high volume of transactions, etc. the client server middleware provides a flexible interface between client and server, who does what, when and to whom.

Why Client Server?

Client server has evolved to solve a problem that has been around since the earliest days of computing: how best to distribute your computing, data generation and data storage resources in order to obtain efficient, cost effective departmental an enterprise wide data processing. During mainframe era choices were quite limited. A central machine housed both the CPU and DATA (cards, tapes, drums and later disks). Access to these resources was initially confined to batched runs that produced departmental reports at the appropriate intervals. A strong central information service department ruled the corporation. The role of the rest of the corporation limited to requesting new or more frequent reports and to provide hand written forms from which the central data banks were created and updated. The earliest client server solutions therefore could best be characterized as "SLAVE-MASTER".

Time-sharing changed the picture. Remote terminal could view and even change the central data, subject to access permissions. And, as the central data banks evolved in to sophisticated relational database with non-programmer query languages, online users could formulate adhoc queries and produce local reports without adding to the MIS applications software backlog. However remote access was through dumb terminals, and the client server remained subordinate to the Slave\Master.

Front end or User Interface Design

The entire user interface is planned to be developed in browser specific environment with a touch of Intranet-Based Architecture for achieving the Distributed Concept.The browser specific components are designed by using the HTML standards, and the dynamism of the designed by concentrating on the constructs of the Java Server Pages.

Communication or Database Connectivity Tier

The Communication architecture is designed by concentrating on the Standards of Servlets and Enterprise Java Beans. The database connectivity is established by using the Java Data Base Connectivity.

The standards of three-tire architecture are given major concentration to keep the standards of higher cohesion and limited coupling for effectiveness of the operations.

Features of The Language Used

In my project, I have chosen Java language for developing the code.

About Java

Initially the language was called as "oak" but it was renamed as "Java" in 1995. The primary motivation of this language was the need for a platform-independent (i.e., architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

- Java is a programmer's language.
- Java is cohesive and consistent.
- Except for those constraints imposed by the Internet environment, Java gives the programmer, full control.

Finally, Java is to Internet programming where C was to system programming.

Importance of Java to the Internet

Java has had a profound effect on the Internet. This is because; Java expands the Universe of objects that can move about freely in Cyberspace. In a network, two categories of objects are transmitted between the Server and the Personal computer. They are: Passive information and Dynamic active programs. The Dynamic, Self-executing programs cause serious problems in the areas of Security and probability. But, Java addresses those concerns and by doing so, has opened the door to an exciting new form of program called the Applet.

Java can be used to create two types of programs

Applications and Applets: An application is a program that runs on our Computer under the operating system of that computer. It is more or less like one creating using C or C++. Java's ability to create Applets makes it important. An Applet is an application designed to be transmitted over the Internet and executed by a Java –compatible web browser. An applet is actually a tiny Java program, dynamically downloaded across the network, just like an image. But the difference is, it is an intelligent program, not just a media file. It can react to the user input and dynamically change.

III. SYSTEM STUDY

3.1 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY:- This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY:-
This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY:-
The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

IV. CONCLUSION

In this paper, we have described the challenges of making online user data aggregation while preserving users' privacy. Based on BGN homomorphic cryptosystem, we have designed the first system that is able to securely and selectively aggregate user data, making it practical in realistic data analytics. It guarantees strong privacy preservation by utilizing differential privacy mechanism to protect individuals' privacy. We have presented PPSA to evaluate aggregation selected by one boolean attribute, and extended it to aggregation selected by multiple boolean attributes and by one numeric attribute. Extensive experiments have shown that PPSA supports various selective

aggregate queries with acceptable overhead and high accuracy.

REFERENCES

[1] R. E. Bucklin and C. Sismeiro, "Click here for internet insight: Advances in clickstream data analysis in marketing," Journal of Interactive Marketing, vol. 23, no. 1, pp. 35–48, 2009.

[2] R. Bose, "Advanced analytics: opportunities and challenges," Industrial Management & Data Systems, vol. 109, no. 2, pp. 155–172, 2009.

[3] H. Chen, R. H. Chiang, and V. C. Storey, "Business intelligence and analytics: From big data to big impact." MIS quarterly, vol. 36, no. 4, pp. 1165–1188, 2012.

[4] I. E. Akkus, R. Chen, M. Hardt, P. Francis, and J. Gehrke, "Nontracking web analytics," in Proceedings of the ACM Conference on Computer and communications security (CCS), 2012, pp. 687–698.

[5] F. Roesner, T. Kohno, and D. Wetherall, "Detecting and defending against third-party tracking on the web," in Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation, 2012.

[6] Directive 2009/136/ec of the european parliament and of the council. [Online]. Available: http://eur-lex.europa.eu/ LexUriServ/LexUriServ.do?uri=OJ:L:2009: 337:0011:0036:en:PDF

[7] Web tracking protection. [Online]. Available: http://www.w3. org/Submission/web-tracking-protection/

[8] V. Rastogi and S. Nath, "Differentially private aggregation of distributed time-series with transformation and encryption," in Proceedings of the ACM International Conference on Management of Data (SIGMOD), 2010, pp. 735–746.

[9] B. Applebaum, H. Ringberg, M. J. Freedman, M. Caesar, and J. Rexford, "Collaborative, privacy-preserving data aggregation at scale," in Proceedings of the 10th Privacy Enhancing Technologies Symposium (PETS), 2010, pp. 56–74.

[10] R. Chen, I. E. Akkus, and P. Francis, "SplitX: high-performance private analytics," in Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM), 2013, pp. 315–326.

[11] R. Chen, A. Reznichenko, P. Francis, and J. Gehrke, "Towards statistical queries over distributed private user data," in Proceedings of the 9th Symposium on Networked Systems Design and Implementation (NSDI), 2012.

[12] E. Shi, T.-H. H. Chan, E. G. Rieffel, R. Chow, and D. Song, "Privacypreserving aggregation of time-series data," in Proceedings of the Network and Distributed System Security Symposium (NDSS), 2011.

[13] T. Jung, X. Mao, X.-y. Li, S.-J. Tang, W. Gong, and L. Zhang, "Privacy-preserving data aggregation without secure channel: multivariate polynomial evaluation," in Proceedings of the 32nd IEEE International Conference on Computer Communications (INFOCOM), 2013, pp. 2634–2642.

[14] D. Fiore, R. Gennaro, and V. Pastro, "Efficiently verifiable computation on encrypted data," in Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS), 2014, pp. 844–855.

[15] T. Jung, X.-Y. Li, and M. Wan, "Collusion-tolerable privacy preserving sum and product calculation without secure channel," IEEE Transactions on Dependable and Secure Computing (TDSC), vol. 12, no. 1, pp. 45–57, 2015.

[16] C. Dwork, "Differential privacy: A survey of results," in Proceedings of 5th International Conference on Theory and Applications of Models of Computation (TAMC), 2008, pp. 1–19.

[17] M. Hardt and S. Nath, "Privacy-aware personalization for mobile advertising," in Proceedings of the ACM conference on Computer and Communications Security (CCS), 2012, pp. 662–673.

[18] C. Dwork, "Differential privacy," in Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP), 2006, pp. 1–12.

[19] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in

private data analysis," in Theory of Cryptography, 2006, pp. 265–284.

[20] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in Proceedings of 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT), 2006, pp.486–503.

[21] S. Inusah and T. J. Kozubowski, "A discrete analogue of the laplace distribution," Journal of statistical planning and inference, vol. 136, no. 3, pp. 1090–1102, 2006.