IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 13, Issue 1 (Jul. - Aug. 2013), PP 51-60
www.iosrjournals.org

Achieving Privacy in Publishing Search logs

D.Srivalli[1], P.Nikhila[2], Ch. Jayachandra[3]
[1]*M.Tech (S.E), VCE, Hyderabad, India,*
[2]*M.Tech (S.E), VCE, Hyderabad, India,*
[2]*M.Tech (C.S.E), VCE, Hyderabad, India,*

Abstract: - *The "database of intentions," collects by the search engine companies for the histories of their users search queries. These searchlogs are a gold mine for researchers. The Search engine companies, however, are wary of publishing search logs in order not to disclose sensitive information. In this paper, we are analysing algorithms for publishing frequent queries, keywords and clicks of a search log. Our evaluation includes the applications that use search logs for improving both search experience and search performance, and our results show that the ZEALOUS' output is sufficient for these applications while achieving strong formal privacy guarantees. We are using two real applications from the information retrieval community: Index caching, as the representative application for search performance, and for the query substitution, as a representative application for search quality. For both applications, the sufficient statistics are histograms of keywords, queries, or query pairs.*

Keywords: - *web search, information technology, database management, data storage and retrieval.*

I. Introduction

This work is part of a newly emerging rigorous study of data privacy, inspired by research in cryptography, which acquired the name of private data analysis. This line of work presents precise mathematical definitions of data privacy that give meaningful guarantees in the presence of a strong, realistic adversary. To provide protocols satisfying the definitions, these works employ a technique called output perturbation, according to which the results of data analysis. They are released after the addition to small amount of random noise. Data privacy is a fundamental problem of the modern information infrastructure. Collections of personal and sensitive data, previously the purview of governments and statistical agencies have become ubiquitous as database systems have grown larger and cheaper. Increasing volumes of information are collected and archived by health networks, financial organizations, search engines, intrusion detection systems; social networking systems, retailers and other enter prizes. The potential social benefits from analyzing these databases are enormous.

The main challenge is to release aggregate information about the databases while protecting the privacy of individual contributors. In contrast to statistical databases and randomized response methods, the records in question contain actual, unperturbed data associated with individuals.

Some of the attributes may be sensitive, e.g., health-related attributes in medical records. Therefore, identifying attributes such as names and Social Security numbers are typically removed from micro data records prior to release.

The published records may still contain quasi-identifiers e.g., demo- graphic attributes such as ZIP code, age, or sex. Even though the quasi-identifier attributes do not directly reveal a person's identity, they may appear together with the identity in another public database, or it may be easy to reconstruct their values for any given individual. Our paper concludes with an extensive experimental evaluation, where we compare the utility of various algorithms that guarantee anonymity or privacy in search log publishing. Our evaluation includes applications that use search logs for improving both search experience and search performance, and our results show that ZEALOUS' output is sufficient for these applications while achieving strong formal privacy guarantees.

We believe that the results of this research enable search engine companies to make their search log available to researchers without disclosing their users' sensitive information.
Search engine companies can apply our algorithm to generate statistics that are (ϵ, δ) probabilistic differentially private while retaining good utility for the two applications we have tested. Beyond publishing search logs we believe that our findings are of interest when publishing frequent item sets, as ZEALOUS protects privacy against much stronger attackers than those considered in existing work on privacy-preserving publishing of frequent items/ item sets.

1.1Related Work:

1.1.1The Cost of Privacy:

Re-identification is a major privacy threat to public datasets containing individual records. Many privacy protection algorithms rely on generalization and suppression of quasi- Identifier attributes such as ZIP code and birth date. Their objective is usually syntactic sanitization: for example, k- anonymity requires that each quasi-identifier tuple appear in at least k records, while l diversity requires that the distribution of sensitive attributes for each quasi-identifier have high entropy. The utility of sanitized data is also measured syntactically, by the number of generalization steps applied or the number of records with the same quasi-identifier. We use the same datasets from the UCI machine learning repository as were used in previous research on generalization and suppression. Our results demonstrate that even modest privacy gains require almost complete destruction of the data-mining utility. In most cases, trivial sanitization provides equivalent utility and better privacy than k-anonymity, l diversity, and similar methods based on generalization and suppression.

1.1.2 Private Data Analysis:

We introduce a new, generic framework for private data analysis. The goal of private data analysis is to release aggregate information about a data set while protecting the privacy of the individuals whose information the data set contains. Our framework allows one to release functions f of the data with instance-based additive noise. That is, the noise magnitude is determined not only by the function we want to release, but also by the database itself. One of the challenges is to ensure that the noise magnitude does not leak information about the database. To our knowledge, this is the first formal analysis of the effect of instance-based noise in the context of data privacy. Our framework raises many interesting algorithmic questions. Namely, to apply the framework one must compute or approximate the smooth sensitivity of f on x. We show how to do this efficiently for several different functions, including the median and the cost of the minimum spanning tree. We also give a generic procedure based on sampling that allows one to release f(x) accurately on many databases x.

1.1.3 Set-valued data:

Set-valued data, in which a set of values are associated with an individual, is common in databases ranging from market basket data, to medical databases of patients' symptoms and behaviors, to query engine search logs. Anonymizing this data is important if we are to reconcile the conflicting demands arising from the desire to release the data for study and the desire to protect the privacy of individuals represented in the data. Unfortunately, the bulk of existing anonymization techniques, which were developed for scenarios in which each individual is associated with only one sensitive value, are not well-suited for set-valued data. In this paper we propose a top-down, partition-based approach to Anonymizing set-valued data that scales linearly with the input size and scores well on an information-loss data quality metric. We further note that our technique can be applied to anonymizing the infamous AOL query logs, and discuss the merits and challenges in Anonymizing query logs using our approach.

1.1.4 Non-Interactive Database Privacy:

We demonstrate that, ignoring computational constraints, it is possible to privately release synthetic databases that are useful for large classes of queries – much larger in size than the database itself. Specifically, we give a mechanism that privately releases synthetic data for a class of queries over a discrete domain with error that grows as a function of the size of the smallest net approximately representing the answers to that class of queries.

We show that this in particular implies a mechanism for counting queries that gives error guarantees that grow only with the VC-dimension of the class of queries, which itself grows only logarithmically with the size of the query class.

We also show that it is not possible to privately release even simple classes of queries (such as intervals and their generalizations) over continuous domains. Despite this, we give a privacy-preserving polynomial time algorithm that releases information useful for all half space queries, given a slight relaxation of the utility guarantee. This algorithm does not release synthetic data, but instead another data structure capable of representing an answer for each query. We also give an efficient algorithm for releasing synthetic data for the class of interval queries and axis-aligned rectangles of constant dimension.

Finally, inspired by learning theory, we introduce a new notion of data privacy, which we call distribution al privacy, and show that it is strictly stronger than the prevailing privacy notion, differential privacy.

II. System Design and Architecture

2.1 Search Logs:

A search log contains valuable information about the searches and corresponding actions performed by users as they interact with a search engine. For example, a web search log collects queries and clicks of users issued to an Internet search engine. Alternatively, a search log may contain queries issued by users and actions performed on the displayed results (e.g., logs for enterprise search, mobile search, database search, product catalog search/transactions, and so forth).A search log can be very useful for providing access to customer services. For example, accessing a search log can help a company improve existing products and services (e.g., keyword advertising) and build new products and services.

Moreover search logs are very valuable data sources that are currently not available to the research community. For example, in many instances an Internet search log is more useful than a web crawl or document repositories as the search log may be used to understand the behavior of users posing queries, and obtain algorithms for problems such as computing related searches, making spelling corrections, expanding acronyms, determining query distributions, query classification, and/or tracking the change of query popularity over time. Advertisers can use such a log to better understand how users navigate to their web pages, gain a better understanding of their competitors, and improve keyword advertising campaigns.

However a search log contains a considerable amount of private information about individuals, and thus a search company cannot simply release such data. Indeed, user searches provide an electronic trail of confidential thoughts and identifiable information. For example, users may enter their own name or the names of their friends, home address, and their medical history as well as of their family and friends. In the case of web search logs, users may enter their credit card number and/or social security number as a search query, just to find out what information is present on the web.

There is shown general block diagram representing a search log **a.1** being processed by a transformation mechanism/algorithm **a.2** into output data. As used herein, a “search log” includes any log that records queries issued by users and actions (e.g. click or purchase) performed on the displayed results. For example, one suitable search log may be a web search log in which each result is a URL and each action is a click. Another suitable search log may comprise a product or service catalog search and transaction log, in which each result is a product or service result, and each action is a purchase. The output data includes query counts **a.3** (e.g., comprising query, query count pairs), and two synapses A and B. The query counts represent an approximate number of times that each query that is safe to publish (as described below) occurs in the log **a.1**.

In this example, the output data also includes one synopsis A, in the form of a privacy-preserving query-action graph **a.4**, comprising a graph over the set of queries and the set of results, where there is an edge connecting a query to a result with weight equal to the number of actions on the result for the query. The query-action graph may be based on the top results for each query, e.g., the graph may represent the approximate query-result action counts for the top results for each query.

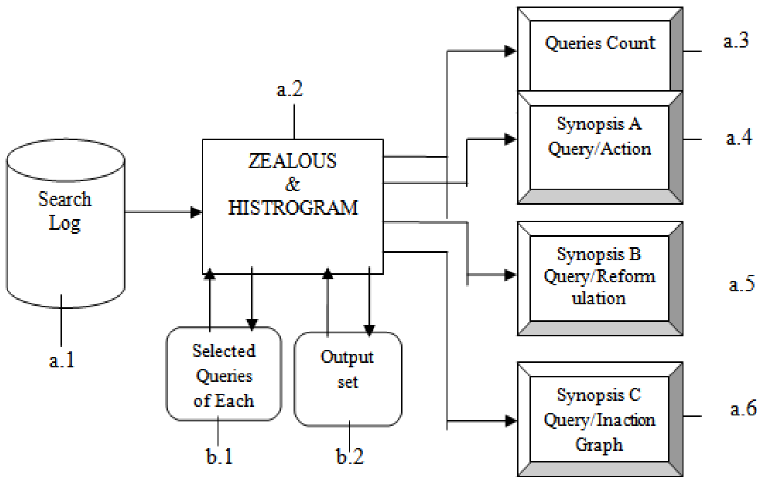


Fig1. Search Log Analysis

Another synopsis B is in the form of a privacy-preserving query-reformulation graph **a.5**, comprising a directed graph over the set of queries, where there is an edge from one query to another query with weight equal to the number of times the first query is reformulated to the second query. The query-reformulation graph may be based on the top reformulations (related/suggested queries that are returned) for each query that are clicked, that is, the graph may represent the query-result reformulation counts for the top related queries returned with the response for each query.

Another synopsis C is in the form of a privacy-preserving query-inaction graph **a.6**. As described below, this reflects for queries and URLs the number of times that a URL was provided in response to a query but no action was taken, e.g., the URL was not clicked, that is, was skipped. Note that in general, any set of d queries of the user can be retained, such as a random selection or pattern that limits the number to d; further, a limit may not be mandatory. Thus, further processing may take place on the search log itself, or on a subset of the queries for processing, e.g., on the block **b.1**.

Then, for every query q that remains in D, represented in FIG. 1 by the block **b.1**, starting with a first query at the occurrence count is obtained, that is, let M (q, D) denote the number of occurrences of q in D. The query q is evaluated against the threshold frequency count K, and released (to an output set **b.2** if and only if the count plus the noise exceeds the threshold frequency count K. In other words, information about any query is published if and only if $M(q, D)+Lap(b_1)\geqq K$. Note that for any or all noise values, the noise value may be positive, or may be negative such that when added, the noisy count decreases relative to the count value before adding; the noise also may be zero.

III. Analyzing Modules

3.1 Search Engine Formation:

Creating a search engine which scales even to today's web presents many challenges. Fast crawling technology is needed to gather the web documents and keep them up to date. Storage space must be used efficiently to store indices and, optionally, the documents themselves. The indexing system must process hundreds of gigabytes of data efficiently. Queries must be handled quickly, at a rate of hundreds to thousands per second.

Search engines play a crucial role in the navigation through the vastness of the web. Today's search engines do not just collect and index WebPages, they also collect and mine information about their users. They store the queries, clicks, IP-addresses, and other information about the inter actions with users in what is called a search log.

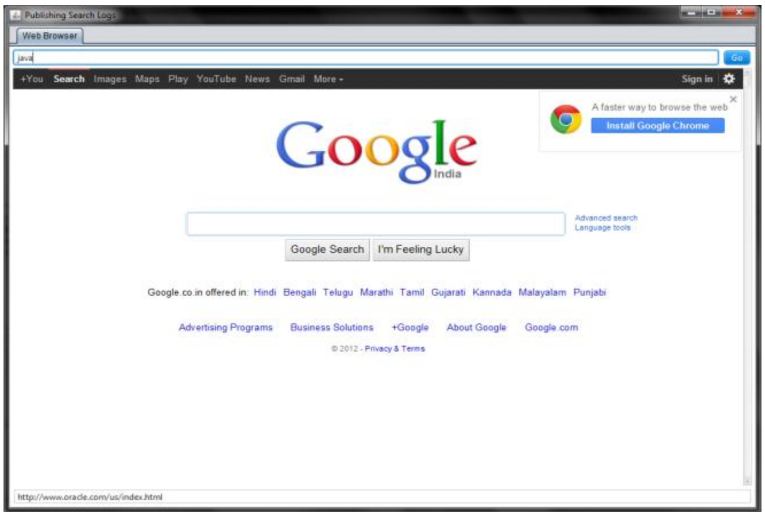
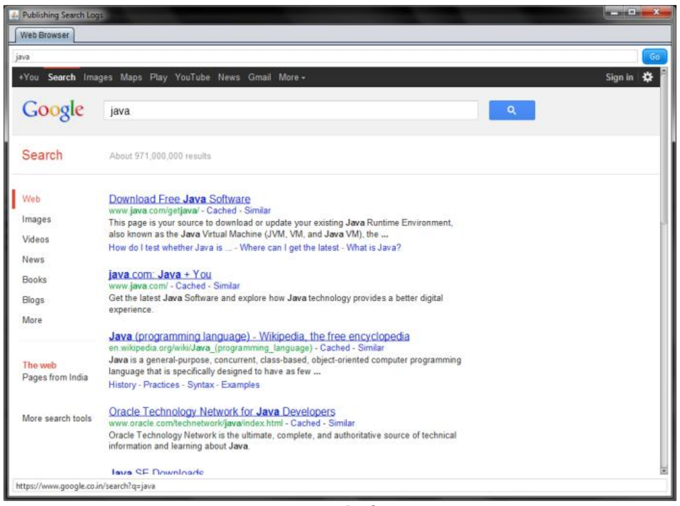

Publishing Search Logs
Web Browser
java
Go
+You Search Images Maps Play YouTube News Gmail More
Sign in
A faster way to browse the web
Install Google Chrome
Google India
Advanced search
Language tools
Google Search
I'm Feeling Lucky
Google.co.in offered in: Hindi Bengali Telugu Marathi Tamil Gujarati Kannada Malayalam Punjabi
Advertising Programs Business Solutions +Google About Google Google.com
© 2012 - Privacy & Terms
http://www.oracle.com/us/index.html

Fig2: google home page

3.2 Category of Formation:

To assure user acceptance, the category formation process must be efficient with respect to both category quality and response time. That AISEARCH is able to fulfil these performance requirements, which gives a snapshot of our comprehensive analysis based on document collections that were categorized by human editors. The performance requirements are also reflected in the implemented software technology: To compare different clustering's of search results, AISEARCH employs strategy patterns to make term weighting schemes, similarity measures, clustering algorithms, and cluster validity measures interchangeable at runtime. For efficient text handling, the symbol processing algorithms for text parsing, text compression and text comparison utilize specialized flyweight patterns.



Publishing Search Logs

Web Browser

java

Go

+You Search Images Maps Play YouTube News Gmail More Sign in

java

Search About 971,000,000 results

Web
Images
Videos
News
Books
Blogs
More

The web
Pages from India

More search tools

Download Free **Java** Software
www.**java**.com/get**java**/ - Cached - Similar
This page is your source to download or update your existing **Java** Runtime Environment, also known as the **Java** Virtual Machine (JVM, VM, and **Java** VM), the ...
How do I test whether Java is ... - Where can I get the latest - What is Java?

java.com: **Java** + You
www.**java**.com/ - Cached - Similar
Get the latest **Java** Software and explore how **Java** technology provides a better digital experience.

Java (programming language) - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**Java**_(programming_language) - Cached - Similar
Java is a general-purpose, concurrent, class-based, object-oriented computer programming language that is specifically designed to have as few ...
History - Practices - Syntax - Examples

Oracle Technology Network for **Java** Developers
www.oracle.com/technetwork/**java**/index.html - Cached - Similar
Oracle Technology Network is the ultimate, complete, and authoritative source of technical information and learning about **Java**.

https://www.google.co.in/search?q=java

Fig3: search for java

3.3 Log generation:

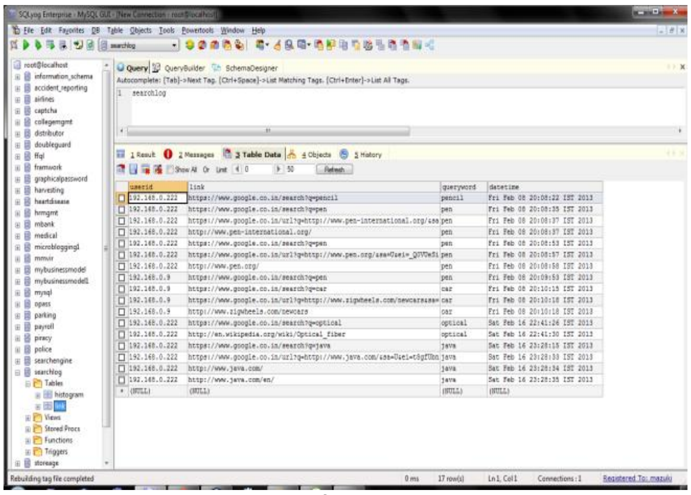
When a user submits a query and clicks on one or more results, a new entry is added to the search log. It is also called search history. This will helpful to find the users behaviour without loss of generality, traditional search log information are user-id, query, time and user clicks. Where a USER-ID identifies a user, a QUERY is a set of keywords, and CLICKS is a list of urls that the user clicked on. The user-id can be determined in various ways;

For example, through cookies, IP addresses, or user accounts. The search logs updated every time when user submits and browse a page.

Fig4: download java

3.4 Query clustering:

Query clustering is a process used to discover frequently asked questions or most popular topics on a search engine. This process is crucial for search engines based on question-answering. A query may be a well-formed natural language question, or one or more keywords or phrases. Once a user query is input, a list of documents is presented to the user, together with the document titles. Because the document titles in Encarta are carefully chosen, they give the user a good idea of their contents. Therefore, if a user clicks on a document, it is likely that the document is relevant to the query, or at least related to it. After applying classification in search logs the clusters will be formed. Clusters are similar search things.



```
1  searchlog
```

userid	link	queryword	datetime
192.168.0.222	https://www.google.co.in/search?q=pencil	pencil	Fri Feb 08 20:08:22 IST 2013
192.168.0.222	https://www.google.co.in/search?q=pen	pen	Fri Feb 08 20:08:35 IST 2013
192.168.0.222	https://www.google.co.in/url?q=http://www.pen-international.org/&sa	pen	Fri Feb 08 20:08:37 IST 2013
192.168.0.222	http://www.pen-international.org/	pen	Fri Feb 08 20:08:37 IST 2013
192.168.0.222	https://www.google.co.in/search?q=pen	pen	Fri Feb 08 20:08:53 IST 2013
192.168.0.222	https://www.google.co.in/url?q=http://www.pen.org/&sa=U&ei=_QVUeSi	pen	Fri Feb 08 20:08:57 IST 2013
192.168.0.222	http://www.pen.org/	pen	Fri Feb 08 20:08:58 IST 2013
192.168.0.9	https://www.google.co.in/search?q=pen	pen	Fri Feb 08 20:09:53 IST 2013
192.168.0.9	https://www.google.co.in/search?q=car	car	Fri Feb 08 20:10:15 IST 2013
192.168.0.9	https://www.google.co.in/url?q=http://www.zigwheels.com/newcars&sa=	car	Fri Feb 08 20:10:18 IST 2013
192.168.0.9	http://www.zigwheels.com/newcars	car	Fri Feb 08 20:10:18 IST 2013
192.168.0.222	https://www.google.co.in/search?q=optical	optical	Sat Feb 16 22:41:26 IST 2013
192.168.0.222	http://en.wikipedia.org/wiki/Optical_fiber	optical	Sat Feb 16 22:41:30 IST 2013
192.168.0.222	https://www.google.co.in/search?q=java	java	Sat Feb 16 23:28:15 IST 2013
192.168.0.222	https://www.google.co.in/url?q=http://www.java.com/&sa=U&ei=t8gfUbn	java	Sat Feb 16 23:28:33 IST 2013
192.168.0.222	http://www.java.com/	java	Sat Feb 16 23:28:34 IST 2013
192.168.0.222	http://www.java.com/en/	java	Sat Feb 16 23:28:35 IST 2013
(NULL)	(NULL)	(NULL)	(NULL)

Fig5: search for java local host

3.5 Keyword Checking:

The keywords formed by different users are different and holds user's uniqueness. The infrequent keyword is compared with the frequent keyword to find there is any sub keyword. If any such sub keyword is found in the infrequent keyword, then the keyword is qualified. Consider the keyword "lecture notes about search logs" is the frequent keyword as discovered by the Zealous algorithm.

The keyword "about search logs" is an infrequent keyword. But it is a sub keyword of the frequent item. If such infrequent item exists then those keywords are qualified to be published. This may improve the addition of useful entries in the log.

IV. Url Clicks

4.1 URL shortening:

The URL (Uniform Resource Locator) reveals the location of a resource in the web environment. Normally an URL contains the fields like protocol, authority, filename, host, path, port. The complete URL of an user click is likely to reveal the user's identity and hence the attributes like filename, path are removed. This procedure would conceal the exact visit of the user.

4.2 Multiple visits to same URL:

A user obtains several search results for the keyword provided for searching. The user chooses the link appropriate to his search intension. The several links chosen by the user may point to the same URL. This reveals that the user finds the information in that page which satisfies their need. Consider the keyword, exam results in the log. The URL clicked by the user from the search results are,

http://www.results.in/colleges/BEresults.html
http://www.results.in/colleges/MCAres.html
http://www.results.in/colleges/MEresutlts.html
http://www.results.in/colloges/MBAres.html

The above clicks of the user reveal that he finds the intended content on the web page http://www.chennairesults.in.The mentioned URL of the page is then qualified and is included in the published log.

4.3 The URL with the keyword:

The user searches by the keyword and obtains search results. Probably the URL chosen by the user may contain the keyword as its sub term. This denotes that it was a relevant click by the user. Such URLs can be included in the published log.

Consider the keyword, exam results is in the search log. The URL clicked by the user is http://www.examinfo.in then this URL is added in the published log. The URL containing the keywords which is chosen by the user, i.e. the entry in the log, showcase that the web page is of common interest. This highly depends on the user's way of providing the keyword and following the links in the result.

V. Providing Privacy

For providing privacy our proposed approach introduces a search log publishing algorithm called ZEALOUS. ZEALOUS ensures probabilistic differential privacy and it follows a simple two-phase framework.

In the first phase, ZEALOUS generates a histogram of items in the input search log, and then removes from the histogram the items with frequencies below a threshold. In the second phase, ZEALOUS adds noise to the histogram counts, and eliminates the items whose noisy frequencies are smaller than another threshold. Our proposed approach provides privacy to infrequent keywords and queries.

5.1 Algorithm:

The Zealous algorithm uses a two phase framework to discover the frequent items in the log and finally publishes it. To discover the frequent items, the Zealous algorithm uses two threshold values. The first threshold value is set based on the number of user contributions in the log. By this method of finding the frequent items, the result log achieves probabilistic differential privacy.

5.2 Algorithm ZEALOUS for Publishing Frequent Items of a Search Log:

Input: Search log S, positive numbers m,λ,T

1. For each user u select a set s_u of up to m distinct items from u's search history in S.[3]
2. Based on the selected items, create a histogram consisting of pairs (k,c_k) where k denotes an item and ck denotes the number of users u that have k in their search history s_u. We call this histogram the original histogram.
3. Delete from the histogram the pairs (k,c_k) with count ck smaller than T
4. For each pair (k,c_k) in the histogram, sample a random number $ή_k$ from the Laplace distribution Lap (λ) [4] and add $ή_k$ to the count ck, resulting in a noisy count: $c_k \leftarrow c_{k+} ή_{k.}$
5. Delete from the histogram the pairs $k,c_k)$ with noisy counts $c_{k \leq} T'$
6. Publish the remaining items and their noisy counts.

To understand the purpose of the various steps one has to keep in mind the privacy guarantee we would like to achieve. Steps 1, 2, and 4 of the algorithm are fairly standard. It is known that adding Laplacian noise to histogram counts achieves Є differential privacy. However, the previous section explained that these steps alone result in poor utility because for large domains many infrequent items will have high noisy counts. To deal better with large domains, we restrict the histogram to items with counts at least T in Step 2. This restriction leaks information and thus the output after Step 4 is not Є differentially private. One can show that it is not even (Є,δ) probabilistic differentially private (for δ < 1=2). Step 5 disguises the information leaked in Step 3 in order to achieve probabilistic differential privacy. In what follows, we will investigate the theoretical performance of ZEALOUS in terms of both privacy and utility. Sections 4.1 and 4.2 discuss the privacy guarantees of ZEALOUS with respect to (Є,δ) indistinguishability and (Є,δ) probabilistic differential privacy, respectively.

Section 4.3 presents a quantitative analysis of the privacy protection offered by ZEALOUS. Sections 4.4 and 4.5 analyze the utility guarantees of ZEALOUS.The main objective of Zealous algorithm is to figure out the frequent items in the log. The Zealous algorithm is applied to a sample search log collected from a local search engine to the items in the log like keywords and URL values. The log contained more than 200 entries with 58 users. The Zealous algorithm was applied to the log with the threshold values in the table.

Table 1: Keyword log of Zealous

Keyword	count
Sport stores 2009	31
Opera browser in mobiles	42
Laptop Models	53
Antivirus software for windows	45
New theme music	63
Exam results	28
Projects	32

The above are the keywords which have passed the filtration of the two phase framework. These keywords are identified as frequent keywords. Similarly it identifies the frequent URL clicks in the log by the two threshold values.

Table 2: URL log of Zealous

URL clicks	**count**
http://esupport.trendmicro.com	17
http://blogs.oracle.com	21
http://en.wikipedia.org	24
http://www.entrance-exam.net	19
https://www.mcafeeasap.com	22
http://technet.microsoft.com	25
http://www.whatis.com	39

However, Zealous algorithm leaves out the infrequent keywords in the log. However setting upon the threshold value is a challenging task. But in a search log, there will be several infrequent items. The infrequent item which has no possibility of revealing an user's identity has to be identified and it has to be published. Hence confess is proposed to qualify such infrequent items in the log.

Table 3: Keyword log of Confess

Keyword	**count**
Sport stores 2009	31
Opera browser in mobiles	42
Laptop mobiles	53
Antivirus software for windows	45
New theme music	63
Exam Results	28
Projects	32
Milk Choclates	33
Laptop Models	33
Choclates	12
Antivirus Software	4
Antivirus Software	20
Results	1
Theme Music	7
Sports	2

The above is the keyword log produced as the result of applying confess algorithm of finding the infrequent items. It can be noted that the keywords which are qualified is the part of the frequent keyword. Releasing such keyword, would improve the utility as the log will contain more entries when published.

Table 4: URL log of Confess

URL Clicks	Count
http://esupport.trendmicro.com	17
https://blogs.oracle.com	21
http://en.sportsstore.org	24
http://www.entrance-exam.net	19
https://www.mcafeeasap.com	22
http://technet.microsoft.com	25
http://whatis.com	39
https://docstoc.com	11
http://blogs.project.com	7
http://en.mcs-college.org	3
http://www.exam.net	4
https://www.webmasterworld.com	1
http://technet.puzzles.com	2
http://www.musics.net	6

The performance of the confess algorithm is analyzed through various parameters like response time, average number of items published in the log. Then the proposed confess algorithm is compared with the zealous algorithm to swot up the performance in terms of utility produced by the log. The below statistics show the average number of keywords published in the zealous log and the confess log. The average number of keyword Nk) is the ratio of the number of items released in the log to the total number of items in the original log. To perform this study various experimental search logs are considered.

Table 5: Average number of keywords

Trails	Zealous log-Nk	Confess log-Nk
1	0.03	0.192
2	0.32	0.130
3	0.02	0.07
4	0.189	0.193
5	0.321	0.381

We first give an overview of how our top-down Partition algorithm performs compared with the bottom-up a algorithm from with three datasets. We then drill down to the dataset BMS-WebView-2, varying different parameters to see how performance changes for these two approaches.

5.3 Clustering Based On Score Frequency Histogram:

Score frequency histogram method is a process that makes statistical analysis of the total scores of land evaluation units, divides the score range into several tiny ones, draws frequency histogram through frequency statistics towards the total score of every land evaluation unit in each percentile range, and finally, delimits the land level boundary according to total score frequency distribution. The method to draw cloud histogram according to cloud fuzzy frequency is as follows. Make the x-coordinate represent the total scores of appraisal units and Y-coordinate cloud fuzzy frequency. Sign the endpoint of each clear-defined interval on the x-coordinate.

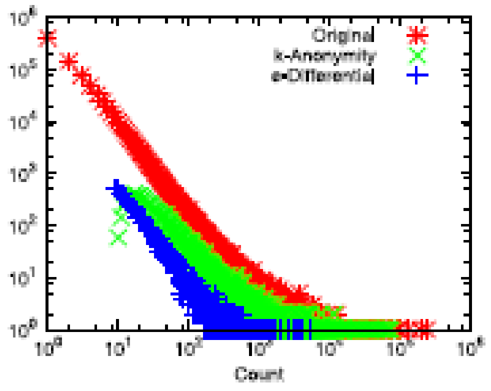


Fig 6. Keyword Counts

Then draw rectangles with the distance d of each interval as the bottom and the cloud fuzzy frequency as the height, and we finally get the cloud histogram. We see that the power-law shape of the distribution is well preserved. However, the total frequencies are lower for the sanitized search logs than the frequencies in the original histogram because the algorithms filter out a large number of items. We also see the cutoffs created by k. We observe that as the domain increases from keywords to clicks and query pairs, the number of items that are not frequent in the original search log increases. For example, the number of clicks with count equal to one is an order of magnitude larger than the number of keywords (Fig 2, 3) with count equal to one.

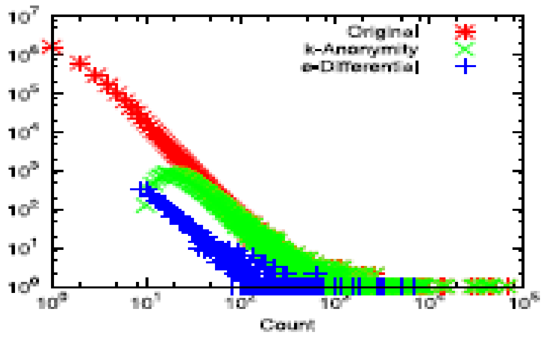


Fig 7. Query Counts

We mentioned the applicability of set-valued anonymization techniques to privacy preserving query log publishing. In this section we discuss query log anonymization as an extended application of our top-down partition-based anonymization, and present preliminary experimental results obtained from anonymizing the infamous AOL query logs using our technique.

5.4 Test Case:

Test Case	Check field	Objective	Expected Result
TC-001	Client	Not open the application	Should check software installation
TC-002	Client	Search any data	High priority doc will come first
TC-003	Server	Read all doc in trends and display	So,visible that doc,with username
TC -004	Server	Not calculating histogram	Should calculate histogram and combine the features
TC-005	Server	Check any data	Now should display all docs,so give privacy to user

VI. Conclusion

In this paper proposed work is to collect in frequent and in frequent search histories .we are using zealous algorithm and histogram counting operations finally to generate log files and counting operations. A topic of future work is the development of algorithms to release useful information about infrequent queries, keywords and clicks in a search log while preserving user privacy

References

[1] N. R. Adam and J. C. Wortmann, “Security-control methods for statistical databases: a comparative study”, ACM Computing Surveys, 25(4), 1989.

[2] N. Ailon, M. Charikar, and A. Newman, “Aggregating inconsistent information: ranking and clustering”, In STOC 2005, 684–693.

[3] A. Blum, C. Dwork, F. McSherry, and K. Nissim, “ Practical privacy: The SuLQ framework”, In PODS, 2005.

[4] S. Chawla, C. Dwork, F. McSherry, A. Smith, and H. Wee, “Toward privacy in public databases. In Theory of Cryptography Conference (TCC)”,2005, 363–385.

[5] S. Chawla, C. Dwork, F. McSherry, and K. Talwar. “On the utility of privacy-preserving histograms”, In 21st Conference on Uncertainty in Artificial Intelligence (UAI), 2005.

[6] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu. “Tools for privacy preserving data mining. SIGKDD Exploration”,4(2),2002,28–34.

[7] I. Dinur and K. Nissim, “Revealing information while preserving privacy”, In PODS, 2003,202–210.

[8] C. Dwork. “Differential privacy”, In ICALP, 2006,1–12.

[9] B.-C. Chen, K. LeFevre, and R. Ramakrishnan,“Privacy skyline: privacy with multidimensional adversarial knowledge”, In VLDB, 2007.

[10] V.Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati. k-anonymity, “Secure Data Management in Decentralized Systems”, 2007.

[11] A. Ev_mievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy-preserving data mining. In PODS, 2003.

[12] B. Fung, K. Wang, and P. Yu. “Top-down specialization for information and privacy preservation”, In ICDE, 2005.

[13] R. Jones, B. Rey, O. Madani, and W. Greiner, “Generating Query Substitutions,” Proc. 15th Int’l Conf. World Wide Web (WWW), 2006.

[14] S. Prasad Kasiviswanathan, H.K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, “What Can We Learn Privately?”, Proc. 49th Ann. IEEE Symp. Foundation of Computer Science (FOCS), 2008,531- 540.

[15] A. Korolova, K. Kenthapadi, N. Mishra, and A. Ntoulas, “Releasing Search Queries and Clicks Privately”, Proc. 18th Int’l Conf. World Wide Web (WWW), 2009.