(IJRST) 2015, Vol. No. 5, Issue No. IV, Oct-Dec **e-ISSN: 2249-0604; p-ISSN: 2454-180X**

AN EFFICIENT METHOD FOR PERSONALIZED WEB SEARCH BASED ON HOMOMORPHIC ENCRYPTION

***Akhila G.S, ** Mr.Prasanth R.S**

**M.Tech student, Department of Computer Science and Engineering*
Mohandas College of Engineering, Anad, Trivandrum
***Asst.Professor, Department of Computer Science and Engineering*
Mohandas College of Engineering, Anad, Trivandrum

ABSTRACT

Using Personalized Web Search (PWS) we can improve the quality of search results in the Internet. The existing UPS based Personalized Web Searching has many drawbacks. First, there may be a chance of eavesdropping when generalized profile forwarded to the server. Second, web server is vulnerable to web attacks like URL manipulation attacks. The impact of these attacks will affect user's personal information. So we introduce a new framework called UPES. Here, the data stored in the server-side and request from user will be in encrypted form. Homomorphic encryption is used for encrypting data. The experimental results show that this framework functioned in the best possible manner with the least waste of time and effort.

Keywords—PersonalizedWebSearch,UPS,Userprofile, Generalized user profile

1. INTRODUCTION

Personalized Web Search (PWS) is related to Web mining. Web mining is mining of data related to World Wide Web. It is divided into different categories like Web content mining, Web structure mining and Web usage mining. This PWS comes under the Web content mining. Web content mining can be thought of as extending the work performed by basic search engines. When same query submitted by different users, typical search engines return the same results regardless of who submitted the query. Here, there is no role for the user. Typically, each user has different information needed for his/her query. Therefore, the search results should be adapted to user with different information needs. Hence, introduced a new concept known as Personalized Web Search. PWS is a general category of search technique to improve the search quality based on individual user needs.

Now we have different types of search engines like Google, Yahoo!, Bing etc. But, the best search engine which supports PWS is Google. If a user creates a Google account, then a user profile is

(IJRST) 2015, Vol. No. 5, Issue No. IV, Oct-Dec e-ISSN: 2249-0604; p-ISSN: 2454-180X

automatically created at the server side. When user search through his/her account, the search engine returns the personalized search results after analysing the user profile of this particular user.

A user profile contains the personal information or interests of a particular person. Different profiling techniques are available to construct the user profile [1]. Before the user profile construction a system needs to identify the interests of users. The sources we have used in constructing a user's profile are: bookmarks from a social bookmarking site, web communities, blogs of interests etc. The first step in the construction of user profile is pre processing. The pre processing step involves stop word removal and stemming. These are then converted to feature vectors where the features are the terms in the documents after the pre processing step. After performing any clustering algorithm, we get several clusters and clusters would represent interests. So if we assign weightages to interest vectors on the basis of documents downloaded and browsed we get a fairer representation of a user's current interest. The weightages are calculated based on the number of documents assigned to each cluster. So user profile has very important role in effectiveness of search quality.

The rest of this paper is organized as follows: Section 2 reviews the existing system and its disadvantages. Section 3 introduces new system architecture and some preliminary knowledge. Section 4 further discusses the implementation of UPES. The experimental results and findings are reported in Section 5. Finally, Section 6 concludes the paper.

2. PROBLEM STATEMENT

In this section, we describe about existing system and its major drawbacks.

2.1 EXISTING SYSTEM

L.Shou et.al[2] introduced a framework called UPS (User customizable Privacy-preserving Search). Fig.1 shows the system architecture of UPS. This consists of a nontrusty search engine server and a number of clients. Here the users can customize their privacy requirements. The main component of this framework is an online profiler implemented as a search proxy running on the client machine itself. This proxy maintains both the complete user profile and the user specified privacy requirements represented as a set of sensitive nodes.

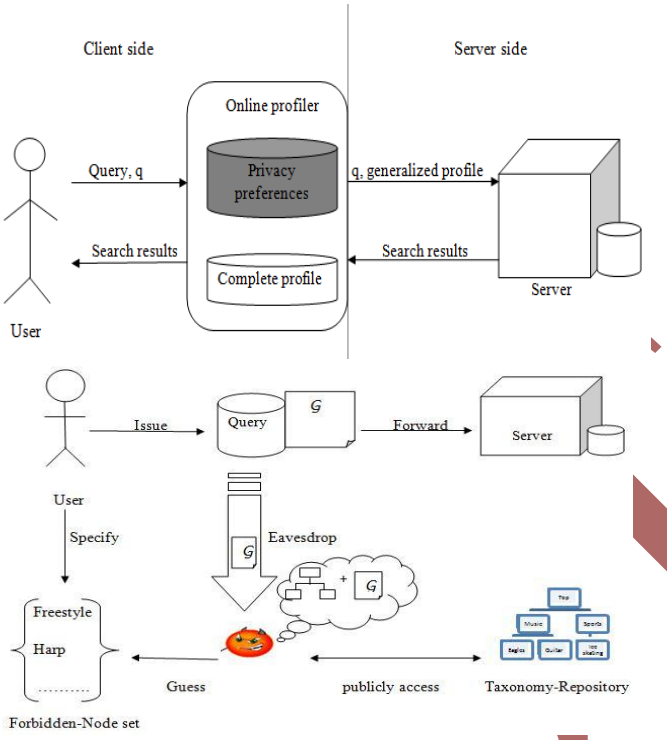

Fig.1:Architecture of UPS

Fig.2: Attack model of UPS

For each user, the framework works in two phases, namely the offline and online. During the offline phase, a hierarchical user profile is constructed and customized with the user specified privacy requirements. Here the user profile is created based on the user's browsing history and a data set, called WordNet. The WordNet is a huge topic hierarchy covering entire topic domain of human knowledge. It is a public accessible data set. By using this data set, the UPS could solve the problem "one profile fits all strategy". During the online phase, when user submits a query, the profiler generates a user profile in runtime in the light of submitted query. The output of this step is a generalized profile which satisfies all the privacy requirements of user. Then, the query and the generalized profile are sent together to the server for personalized search. The search results are personalized with the user profile and forwarded to the query proxy. Finally, the proxy presents the raw results to the user. Section 2.2 explains the drawbacks of the existing system and section 2.3 defines the current issues of web server.

2.2 ATTACK MODEL

There may be a chance of eavesdropping when generalized profile forwarded to the web server. Based on generalized profile, the attacker will attempt to hack the sensitive nodes of the user by recovering the hidden segments in the original user profile, and computing a confidence for each

recovered topic, relying on the background knowledge in the publicly available taxonomy, that is, WordNet. Fig.2 shows the attack model of UPS.

2.3 WEB ATTACKS

Attacks on web application are always harmful since they give the company a bad image.
A successful attack can have any of the following consequences:

- Website defacement
- Stolen information
- Modification of data, and particularly modification of users' personal data
- Web server intrusion

Some examples for this type of web attacks are describe below.

2.3.1 URL MANIPULATION ATTACKS

By manipulating certain parts of a URL, a hacker can get a web server to deliver web pages he is not supposed to have access to. The impact of these types of attacks is website defacement. Website defacement is an attack on a website that changes the visual appearance of the site or a webpage. These are typically the work of system crackers, who break into a web server and replace the hosted website with one of their own.

2.3.2 TRIAL AND ERROR ATTACKS

A hacker may possibly test directories and file extensions randomly in order to find important information. Here a few classic examples:
Search for directories making it possible to administer the site:

- http://target/admin/
- http://target/admin.cgi

Search for a script to reveal information about the remote system:

- http://target/phpinfo.php3

Search for backup copies. The .bak extension is generally used and is not interpreted by servers by default, which can cause a script to be displayed.

- http://target/.bak

Search for hidden files in the remote system. On UNIX system, when the site's root directory corresponds to a user's directory, the files created by the system may be accessible via the web.

- http://target/.bash_history
- http://target/.htaccess

2.3.3 DIRECTORY TRAVERSAL ATTACKS

These attacks involve modifying the tree structure path in the URL in order to force the server to access unauthorized parts of the site.

The reasons for these attacks are mainly due to all data are stored on server in plain form. No authorization is provided for web server which contains very important information.

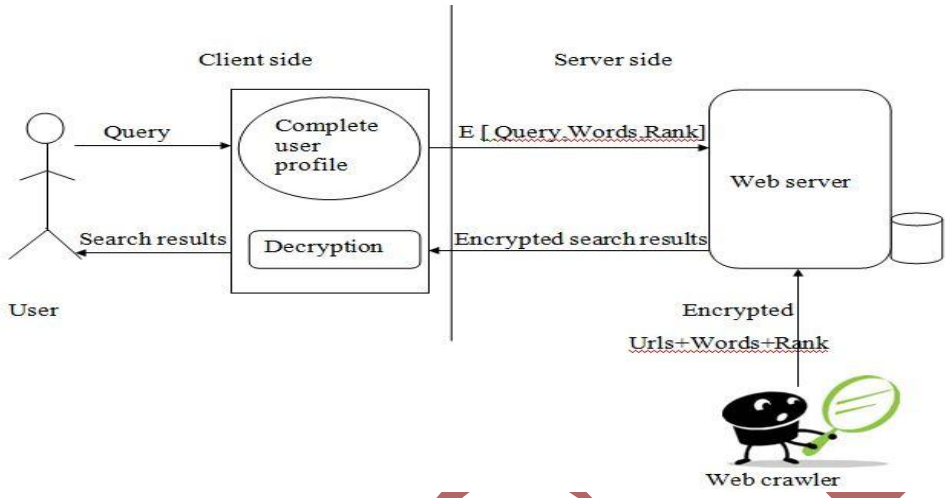

Fig.3: Architecture of UPES

3. PROPOSED SYSTEM

The above problems are addressed in our UPES (literally for User Privacy-preserving Encrypted Search) framework. As illustrated in Fig.3, we have a web crawler for crawling the web pages. In normal case, the crawled details like urls, words in urls, and corresponding rank are stored on server in plain form. But here, we store these details as encrypted form. So we can protect the web server from all types of web attacks. When the user submit a query, the server will get the encrypted combination of query, related words of this query from user profile, and corresponding ranks of each term. In Section 3.1, we present the creation of user profile. Request from user to the server is in the encrypted form, that's why we can protect the user's personal information and avoid eavesdropping problem. For encryption, here we use homomorphic encryption, described in Section 3.2. The user will get the results after decrypting the personalized search results from server. Thus we can protect user's privacy and web server from all types of attacks.

3.1 USER PROFILE

In UPES, each user profile adopts a hierarchical structure. Moreover, our profile is constructed based on the availability of a public accessible taxonomy that is WordNet. For constructing the user profile we need to track the user's search history. For each click to the links, the corresponding URL information stored. Using these information and WordNet information, we can construct user profile. In section 4, we present details about this step.

(IJRST) 2015, Vol. No. 5, Issue No. IV, Oct-Dec e-ISSN: 2249-0604; p-ISSN: 2454-180X

3.2 HOMOMORPHIC ENCRYPTION

Homomorphic encryption is a form of encryption which allows specific types of computations to be carried out on cipher text and generate an encrypted result which, when decrypted, matches the results of operations performed on the plaintext. It permits computing on encrypted data. The client can encrypt his data x and send the encryption Enc(x) to the server. The server can then take the cipher text Enc(x) and evaluate a function f on the underlying x obtaining the encrypted result Enc(f(x)). The client can decrypt this result achieving the wanted functionality, but the server learns nothing about the data that he computed on.

4. IMPLEMENTATION

The UPS framework is implemented on a PC with an Intel Core i5 2.67-GHz CPU and 4-GB main memory, running Microsoft Windows 7. All the algorithms are implemented in Java. The topic repository uses the WordNet. First step in our thesis work was download dataset from web. Here we provide an authentication for server. For each user, need to authenticate their identity to the server before accessing the server.

We can divide the overall procedure of the thesis into three modules.

Profile construction: This is an offline process for identifying user's interests, for constructing user profile that we need to track the user's search history. So, the first form gives an interface for the user to search their queries. For each click to the links, the corresponding url information stored into the table "*urlinfo*". Internally, we calculate the most frequent words in each url. After performing stemming, count each word, if the count is more than predefined value it will be stored into the corresponding entry in the table. These words are also stored into the table "*allwords*". Now we have frequent words, which might be the interested topics of the user, also we need to find corresponding related words in the WordNet. For that we use "*allwords*" table's information and dataset. Trace all related words and store those words into

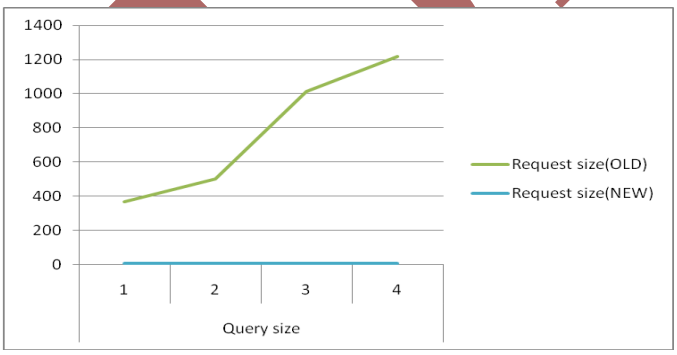

Fig.4: Request size against Query size

"*related_words*" table. Based on all these tables we constructed the user profile in a hierarchical structure. We used simple tree construction java code for the hierarchical structure.

Web crawling: A web crawler starts with a list of URLs to visit, called the seeds. As the crawler visits these URLs, it identifies all the hyperlinks in the page and adds them to the list of URLs to visit, called the crawl frontier. URLs from the frontier are recursively visited according to a set of policies.

Encryption and decryption: The crawled data are encrypted and stored on server. Here user's request also in encrypted form. We use substitution method for encryption. For encryption, take each input string, and then group them into four and find corresponding ASCII value. If it starts with minus cannot encrypt, so insert one in front of result. Otherwise, insert two. During decryption, reverse process will happen.

5. EXPERIMENTAL RESULTS

In this section, we present the experimental results of UPES. We conduct two experiments on UPES. In the first experiment, we check request size against query size. In existing system, user passes a generalized profile when submitting the query. But in UPES, it passes only top five terms related to submitted query. Fig.4 shows the request size against query size in both UPS and UPES. Second, we look at the query execution time of proposed system. Due to the bulk amount of request size, the query execution time of existing system is more than the proposed system. Fig.5 shows the query execution time against query time. And also, we can provide more security because of homomorphic encryption.

6. CONCLUSION

This paper provides a client-side and server-side privacy protection framework called UPES for personalized web search. By using UPES, we can perform online generalization on user profiles to protect the personal privacy without compromising the search quality. This framework could solve the problems of user's privacy. All types of web server attacks could be solve by using this framework. Here we provided an authentication for server. Thus using this framework, we can completely protect client-side and server-side privacy.

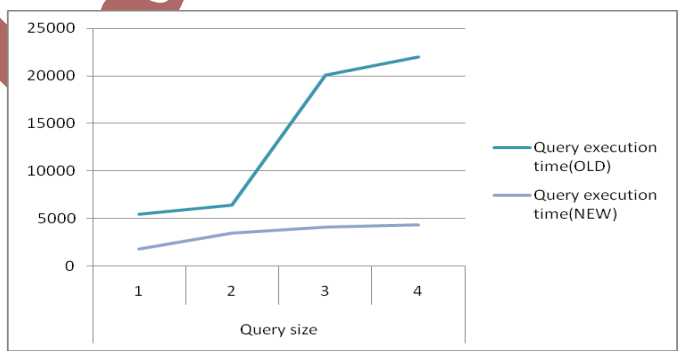

Fig.5: Execution time against Query size

REFERENCES

[1] R.S.Bhadoria, D.Sain, and R.Moriwal, "Data Mining Algorithms for personalizing user's profiles on Web", Vol.1,issue 2 IJCTEE.

[2] L.Shou,H. Bai,K.Chen, and G. Chen,"Supporting Privacy Protection in Personalized Web Search",vol.26,February 2014.

[3] A.Pretschner and S.Gauch,"Ontology-Based Personalizaed search and Browsing," Proc. IEEE 11th Int'l Conf.Tools with Artificial Intelligence,1999.

[4] L. Fitzpatrick and M.Dent,"Automatic Feedback Using Past Queries:Social Searching?," Proc.20th Conf.,1997.

[5] K.Sugiyama, K. Hatano, and M. Yoshikawa,"Adaptive Web Search Based on User Profile Constructed without any Effort from Users," Proc.13th Int'l Conf. World Wide Web (WWW), 2004

[6] M.Spertta and S.Gach,"Personalizing Search Based on User Search Histories,"Proc.IEEE/WIC/ACM Int'l Conf.Web Intelligence,2005.

[7] F.Liu,C.Yu and W.Meng,"Personalized Web search for improving retrieval effectiveness,"2004.

[8] Y.Xu,K.Wang,B.Zhang and Z.Chen,"Privacy-Enhancing Personalized Web search,"Proc.16th Int'l Conf.,2007.

[9] S. Latha, M. Rajaram, and S. N. Sivanandam, "A Survey on Semantic Web Mining based Web Search Engines", VOL. 6, NO. 10, OCTOBER 2011 ARPN Journal of Engineering and Applied Sciences.

[10] M. Rami Ghorab, Dong Zhou, Alexander O'Connor, and Vincent Wade, "Personalised information retrieval: survey and classification".

[11] R.S.achake and Prof.G.P.Potdar "A Survey on Personalized Search: An Web Information Retrieval System".

[12] V.Kakulapati , Dr. D. Vasumathi and S.Jena "Survey on Web Search Results Personalization Techniques".

[13] B. Tan, X. Shen, and C. Zhai, "Mining Long-Term Search History to Improve Search Accuracy," Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), 2006.

[14] X. Shen, B. Tan, and C. Zhai, "Implicit User Modeling for Personalized Search," Proc. 14th ACM Int'l Conf. Information and Knowledge Management (CIKM), 2005.

[15] Y. Zhu, L. Xiong, and C. Verdery, "Anonymizing User Profiles for Personalized Web Search," Proc. 19th Int'l Conf. World Wide Web (WWW), pp. 1225-1226, 2010.

[16] A. Krause and E. Horvitz, "A Utility-Theoretic Approach to Privacy in Online Services," J. Artificial Intelligence Research, vol. 39, pp. 633-662, 2010.

[17] G. Chen, H. Bai, L. Shou, K. Chen, and Y. Gao, "Ups: Efficient Privacy Protection in Personalized Web Search," Proc. 34th Int'l ACM SIGIR Conf. Research and Development in Information, pp. 615-624, 2011.