Revealing PIR Protocols Protected Users

Rafiullah Khan
Capital University of Science and Technology
Pakistan
rafiyz@gmail.com

ABSTRACT: *Privacy preservation is one of the major issue in digital products. In current era, web search engine has become vital tool for internet users. However, most search engines maintains user profile and analyze them which could compromise privacy. In order to preserve the user privacy from a search engine, many privacy preserving solution have been proposed including query obfuscation, anonymizing networks and P2P PIR Protocols. However, first two solutions cannot preserve user privacy completely and they have their own drawbacks as well, while the P2P PIR are passing through evolution stage. This paper focus on P2P PIR protocols, Useless User Profile (UUP), User Private Information Retrieval (UPIR) and their variants and tried to evaluate and quantified their techniques in preserving user privacy. In this research association rule mining technique is used to expose the source of query of interest. Each protocol is simulated under varying parameters of Group Size, Number of Query of Interest (QoI) and association rule mining Window Size over the released data of AOL. The results shows that on the average the proposed adversarial model exposed 85.34% and 80.75% QoI source correctly of UUP and UPIR protocol respectively. Similarly based on results, it is also concluded that less number of QoI submitted by a user and large groups size offer more protection to the user.*

Keywords: PIR; Private Information Retrieval; Data mining; ARM; Query Privacy; Web Search; Search Privacy.

Received: 18 January 2016, Revised 20 February 2016, Accepted 27 February 2016

1. Introduction

Since from the beginning of the World Wide Web Services, diverse kind of computer manageable data is pouring in from various sources. The recent advancement in the communication networks facilitate user to access and share a large volume of the information in minimum amount of time[1]. The service started on the concept of global information system [2] is now became a huge pile of information of almost every sort [3]. However, in order to find most relevant and specific information efficiently search engine were introduced in 90's era [4]. The job of web search engine is to fetch information against user query. For better search results search engines maintain user's profile for analysis [5]. Maintaining user's profile could also be helpful for recommendation system and customized user settings however, the user's profile may contains personal and sensitive information which can breach user's privacy.The issue of maintaining user profile has received considerable public attention in 2005, when Google was asked by US Department of Justice to submit logs of two months in 2006 [5]. Then in 2006 AOL release their search log which contains 20 million search records of 658000 users [6]. Although they replace the users IP address with fictitious ID however, New York Times was able to deduce some of the users correctly based on their search patterns and personal information [7].

In response some techniques have been proposed to intact user privacy while using web search engine. These techniques includes query obfuscation tool [8], query scrambling [9], anonymizing networks [3] and Peer to Peer (P2P) private information retrieval (PIR) protocols[5, 10-13]. In this research our focus is to evaluate and quantify the privacy offer by P2P PIR protocols against adversarial web search engine. We select two major protocols with their variants proposed by two research groups. The protocols includes UUP (Useless User Profile) [10], UUP with Lindell and Waisbard Improvement [14], UUP with Untrusted Partner [15, 16], UPIR (User Private Information Retrieval) [11, 12], UPIR with Optimal Configuration [17] and UPIR with Self-Submission [18]. All of them are not yet deployed completely due to high computation and communication overheads [8] however their complete working model is available in literature which is more than enough for simulation.

The rest of the paper is organized as follows. After introduction section II cantinas problem formulation and related work, the proposed adversarial model is discussed in section III. Section IV contains experimentation setup which contains the dataset properties, artificial log creation method. Last two sections contains the results, conclusions and future work.

2. Problem Formulation And Related Work

This paper investigates if it is possible for a search engine to find the source of the QoI if the user submitted the query using any P2P PIR protocol. Using UPIR and its variants, user write his/her search query on a shared memory location. Some other group member access the memory location, submit the query to the search engine and then write the reply on the same memory location for collection [11, 12, 17, 18]. While in UUP and its variants, the search queries of all the group members are first shuffled within the group among the group members and then submitted to the search engine [5, 10, 14-16]. In this research it is assumed that search engine will ignore the rest of the information present in user request except user queries to uncover the source. Based on working of protocols specified in the literature, in this study a java based model has been developed for simulation of protocols and creation of artificial log. The main aim of this research is to find up to what extent these protocol provide privacy to the users against the adversarial search engine who wants to know the source of “QoI”. The proposed adversarial model assume a search engine who is interested to reveal the source of “QoI”. For conducting experiments, the protocols (mentioned previously) are simulated over real users search queries from the dataset released by AOL in 2006. Although AOL data is not well suited data for experiments as it represent different time period (year 2006), however for two reasons the dataset was selected: First only queries keywords are needed to uncover the user, while the rest of the information is not essential, secondly this is the only dataset available which contains real user queries.

In terms of related work, many adversarial models have been developed in recent past to evaluate the effectiveness provided by different tools for search privacy. Peddinti and Saxena evaluate and quantified the privacy provided by two major tools TrackMeNot (TMN) [3, 8] and The Onion Routing (Tor) [3] using machine learning algorithms. For evaluation first they simulate both TMN and Tor to create an artificial web search engine log and they launched their proposed attack over that artificial log. In case of TMN their aim was to identify the machine generated queries and user generated queries. In case of Tor, their aim was to identify the quires of user of interests (UoI). In both cases they used AOL dataset and they assumed that the search engine have a partial profile of the UoI. For that purpose they used 2/3 of the data for training and 1/3 of the data for testing. Their success rate in identifying original queries from TMN generated queries was 48.88% on the average. While in Tor, their average true positive rate was 25.95% in 99 users relay and 19.95% was for 999 relay users. They also claimed that few user of interest are identified with 80% to 98%. Similarly Gervais et al proposed a linkage function which calculate the similarity between pair of queries that predicts weather source of any two queries is same or not [19]. The linkage function uses time, structure, content, landing web page and other information to find similarity. Their attack model first use machine learning algorithms to find linkability between user queries then quantify privacy of the user according to linkage attack. They quantify user privacy in two levels, at query level and semantic level [19]. Petit et al evaluate the privacy provided by Tor, TMN and GooPIR tools using their “SimAttack” technique. SimAttack is used to find similarity between submitted query and user profile. Their success rate was 36.7%, 45.3% and 51.6% in Tor, TMN and GooPIR respectively [20].

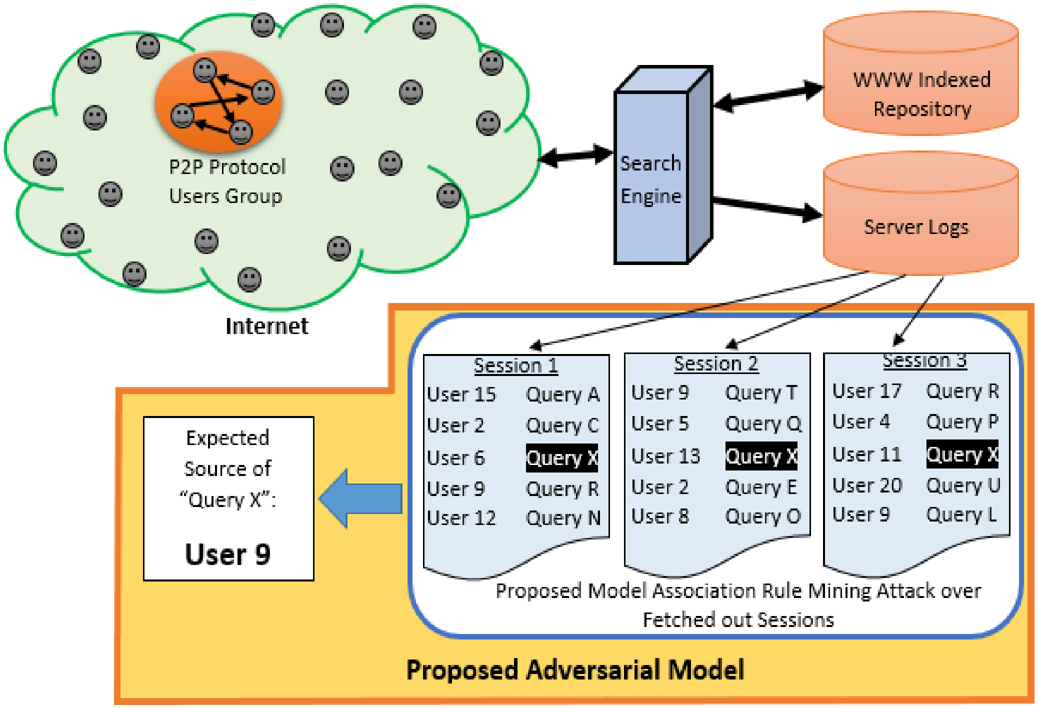
3. The Proposed Adversarial Model

In the proposed Adversarial Model search engine is an adversary entity who is trying to breach user privacy. This paper assume that the server is only interested to find the source of a QoI, which is submitted to the Web Search Engine through any P2P PIR protocol. The proposed adversarial model uses a modified form of Association Rule mining Technique to discover the user. Association Rule Mining (ARM) techniques is used to discover the relation between the variables in large databases and other information repositories [21, 22]. ARM rules are created by analyzing data for frequent patterns which are measured in terms of support Eq. (1) and confidence Eq. (2) to find their importance [22]. For example in market basket analysis if in 8 transection (out of 10 transections) customer also buys milk when she buys a dozen eggs then we an ARM rule that "If a customer buys a dozen eggs, she is 80% likely to purchase milk". Support and Confidence are two measures use to find strength of the ARM rule. Support Eq. (1) represents a probability a transection contains "X U Y" where X and Y both are variables. While Confidence Eq. (2) represents conditional probability that a transection having X variable also contains Y variable [23].

$$Support(X) = \frac{Frequency(X \bigcup Y)}{Total\,Tran\,sec\,tions} \tag{1}$$

$$Confidence(X \rightarrow Y) = \frac{Frequency(XUY)}{Frequency(X)} \tag{2}$$

In proposed adversarial model a modified form of ARM technique is used. According to the proposed adversarial model where in the log a Query of Interest (QoI) appears, take a fix number of record appears immediately before and immediately after and name it as a single session window. After finding all the sessions, the frequency of each user appear in the windows is calculated. The user with maximum frequency is considered as the possible source (as shown in Fig. 1) and the Frequency (X U Y)/ Frequency (X) represents both Support and Confidence in in this paper where "X" represents the "QoI" and "Y" represents users. The working of proposed adversarial model is shown in Figure 1.



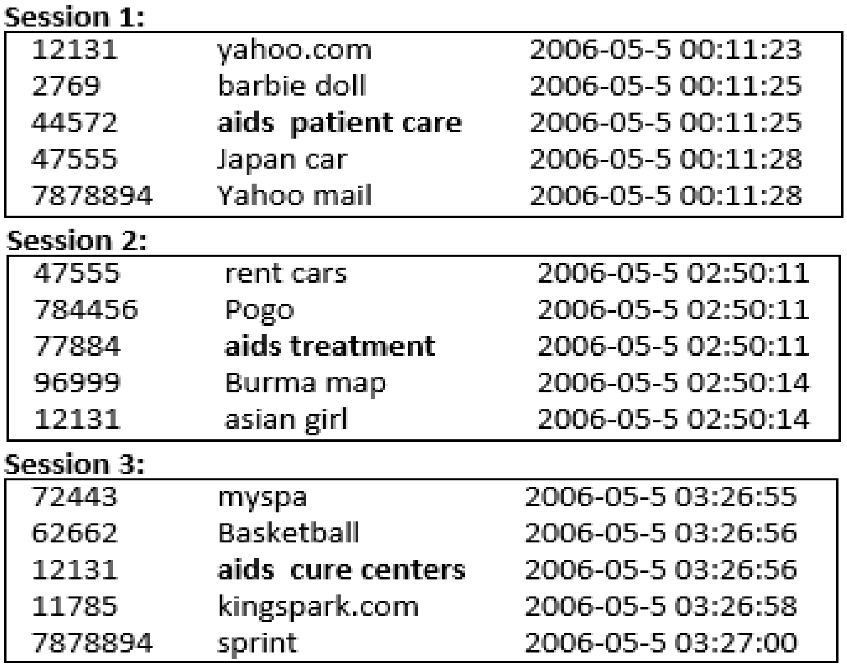
Session 1	
User 15	Query A
User 2	Query C
User 6	Query X
User 9	Query R
User 12	Query N

Session 2	
User 9	Query T
User 5	Query Q
User 13	Query X
User 2	Query E
User 8	Query O

Session 3	
User 17	Query R
User 4	Query P
User 11	Query X
User 20	Query U
User 9	Query L

Figure 1. Proposed Adversarial Model

For example consider a search engine query log which is created using one of the above mentioned PIR protocols. There are some queries appeared in a log with their genuine users but source of some queries are hidden using any PIR protocolmentioned above. In order to uncover the source the proposed adversarial model will initially need Queries of Interests (QoI) as a seed. Let in above mentioned case "QoI" are health related queries and "QoI" seed file is composed on "aids patient care", "aids treatment" and "aids cure centers".



Session 1:

12131	yahoo.com	2006-05-5 00:11:23
2769	barbie doll	2006-05-5 00:11:25
44572	**aids patient care**	2006-05-5 00:11:25
47555	Japan car	2006-05-5 00:11:28
7878894	Yahoo mail	2006-05-5 00:11:28

Session 2:

47555	rent cars	2006-05-5 02:50:11
784456	Pogo	2006-05-5 02:50:11
77884	**aids treatment**	2006-05-5 02:50:11
96999	Burma map	2006-05-5 02:50:14
12131	asian girl	2006-05-5 02:50:14

Session 3:

72443	myspa	2006-05-5 03:26:55
62662	Basketball	2006-05-5 03:26:56
12131	**aids cure centers**	2006-05-5 03:26:56
11785	kingspark.com	2006-05-5 03:26:58
7878894	sprint	2006-05-5 03:27:00

Figure 2. Sessions related to Query of Interest (QoI)

These queries are sensitive in nature and user don't want to disclose his/her identity. Our adversarial model will then search queries present in seed file. When a query match found, we will take fix number of record appears immediately before and immediately after the "QoI". In example we took 2 records before and 2 records after the "QoI" appears which make it a window of 5 records. After fetching out all three windows (as shown in Fig. 2.), we will find the frequency of each user appear in window. The frequency of each user is shown in the Table 1.

User ID	Frequency	Support	Confidence
12131	3	3/3	1.00
2769	1	1/3	0.33
44572	1	1/3	0.33
47555	2	2/3	0.66
7878894	2	2/3	0.66
784456	1	1/3	0.33
77884	1	1/3	0.33
96999	1	1/3	0.33
72443	1	1/3	0.33
62662	1	1/3	0.33
11785	1	1/3	0.33

Table 1. User Frequency, Support AND Confidence

Now based on the frequency, support and confidence shown in table I, the strongest rule created “QoI ’! user 12131” with support and confidence of 1.00. Similarly both “User 47555” and “User 7878894” are also potential candidates but we can reject them both based on two assumptions. First we have a strongest rule with the support and confidence of 1.00 and secondly we can reject them both based on query analysis however the second assumption of rejection could be misleading therefore in this paper we will stick with the first assumption of strong rule. In our experiments we took minimum support of 50% however our result graphs only represents the strongest rules. We have also calculated the accuracy percentage of each protocol under multiple variables.

4. Experimentation Preliminaries

4.1 AOL Dataset Properties

The query log contains the collection of records from 1st of March, 2006 to 31st of May, 2006. It contains record of more than 20 million queries submitted by more than 650 thousand users. The query log was composed on five attributes i.e. AnonID, Query, QueryTime, ItemRank and ClickURL. AnonID was anonymous identifier, which was replaced before the release of the data. Originally it was real user identifier (may be IP Address or email ID) which was then replaced. The Query attribute contains query issued by the user. Before the released they removed punctuations from the query. QueryTime contains the date and time at which the query was submitted. ItemRank field contains the rank of the item clicked by the user in search result and ClickURL records the URL clicked by the user in search page [6].

4.2. Artificial Search Log Creation

For artificial log creation, first we divide all selected protocols in three groups based on group status and self-query submission parameter. First group contains those protocols who make dynamic groups and their group members can’t submit their own queries. Second group represents those protocols whose groups are static and their group member also can’t submit their own queries. Third group contains those protocols which use static group and all group member can submit their own queries as well. In first group we have selected UUP and its two variants Lindell and Waisbard Improvement and UUP with Untrusted Partner. In second group we have selected UPIR and UPIR with Optimal Configuration and in third group we have selected UPIR with self-query submission.

In other parameters of simulation and adversarial model, we have considered Group user selection, type of query, number of queries submitted by a user, query log period, ARM session window size and number of attempts as sown in Table II. Important parameters are discussed as follow:

Group Size: First parameter is group size which is one of the basic requirement of all protocols. In UUP and its variants, the protocols were testing on group size of 3, 4, 5, 10 users [10, 15, 16]. While in UPIR and its variants, group of “n” users are considered. In our experiments we took group of 3, 6 10 and 20 users in to evaluate the effect of group size in user privacy.

Group User Selection: the group user selection parameter is vary from protocol to protocol. For UUP and its variants we select random users from the log with in the time period of 30 minutes and use them as group members. For next query we repeated the same procedure as per mentioned in literature [5, 10, 14, 15]. For UPIR and UPIR Optimal Configuration group is static so in simulation we initially select random users and make it a static group for rest of the simulation.

Query Type: The crux of our hypothesis is that we can expose a user based on a query which is sensitive in nature. Where sensitive queries are those queries which a user may not willing to reveal to the outside world [3]. These queries might be benign like persons medical condition, searching job, political and religious affiliation or may be harmful and alarming like terrorism or child abuse. Our Query of Interest (QoI) is sensitive query.

Number of Queries: This parameters shows that how many times a person repeat a same or closely related queries using any P2P PIR protocol. We have several examples of users in AOL data who repeated the same query many time in a period of 60 minutes. Like user “966721”, “1750999”, “525025” and many other users repeated same queries many time in 60 minutes time period and all of them are sensitive in nature. In this research we assumed that a targeted user will submit minimum 3 queries, 5 queries and maximum 10 queries per day to see the effect of number of queries on exposure of user. We can also provide a file which contains “QoI’s”.

ARM Session Window Size: Session window size is one of the important feature for adversarial model. Session window will be composed on the user specified number of record appeared immediately before and after “QoI” record. These sessions are then used to find the Frequency, Support and Confidence of the rules. In this research we took 50, 100, 200 and 300 session window size which means 25, 50, 100 and 150 records respectively will be considered in session windows be before and after “QoI”.

Protocols	UUP, Lindell and Waisbard Improved UUP Untrusted Partner UUP, UPIR, UPIR Optimal Configuration
Group Size	3, 6, 10, 20 (Users)
Group Users Selection	Random from query log for UUP, Lindell and Waisbard Improved UUP, Untrusted Partner UUP
	Static Group of Specified users for UPIR, UPIR Optimal Configuration and UPIR Self Query Submission.
Group Status	Dynamic (UUP, Lindell and Waisbard Improved UUP Untrusted Partner UUP)
	Static (UPIR, UPIR Optimal Configuration, UPIR Self Query Submission)
Query type	Sensitive Query of Interest (QoI)
Number of Queries	3, 5, 10 Queries
Query Log Period	1 Month (from 2006-04-30 23:59:59 to 2006-5-31 23:59:58)
Total Entries in Log	1183990
ARM Session Window Size	50, 100, 200, 300
Number of Attempts	5

Table 2. Properties of Artificial Log Creation And Adversarial Model

5. Results And Discussion

In our experiments, we tried to expose the source of the QoI under variation of Group Size, Number of QoI and ARM Session Window size using variants of UUP and UPIR protocols. We conducted over 700 experiments with all possible combinations of selected parameters. In each experiment we computed the source visibility and accuracy. However in UPIR and its variants we have also calculated the visibility of the group users in percentage using Eq. (3). Each graph is composed on three sub-graphs where each sub-graph represents number of QoI submitted by the source.

$$Group\,Visibility = \frac{Total\ No\ of\ Visible users}{Group\,Size} * 100 \qquad (3)$$

Figure 3 represents the first group of protocols which contains UUP and its all selected variants. According to the three queries sub-graphs of Fig 3, the source visibility probability is 0.6 in first scenario where group size is 3 and

ARM window size is size is 50.The user visibility is getting better with bigger ARM window size but it again drop to 0.6 at 300 window size. The same kind of behavior is observed when group size taken as 6 users. For group size of 10 and 20 users, visibility dropped to 0.4 and 0.0 respectively. However the visibility getting better when we increase the window size. Rest of the sub-graphs shows similar behavior i.e. 100% visibility except 50 window size where visibility is less than 45%.

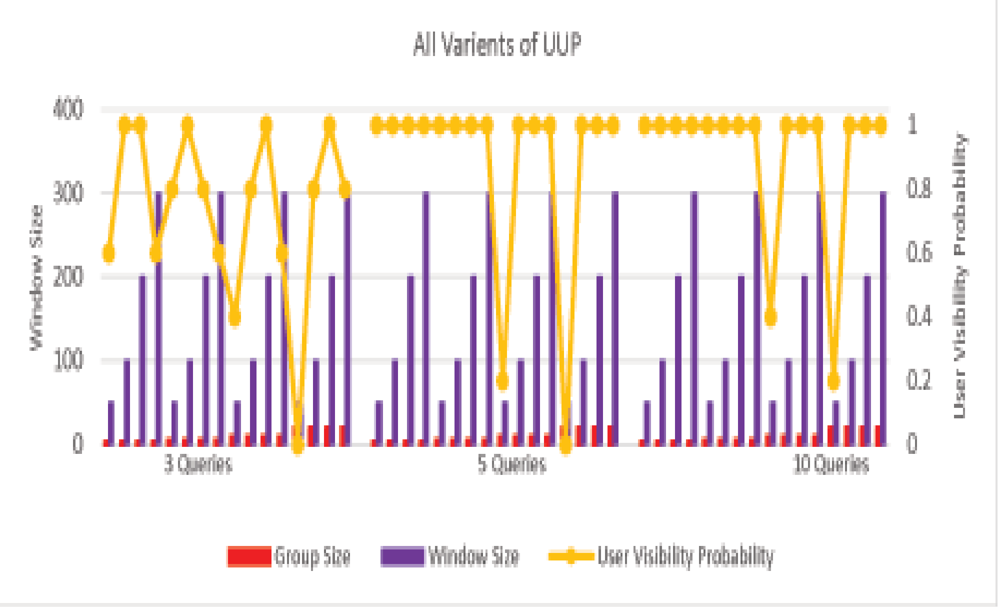


Figure 3. Source user exposing probability in UUP and its Selected Variants

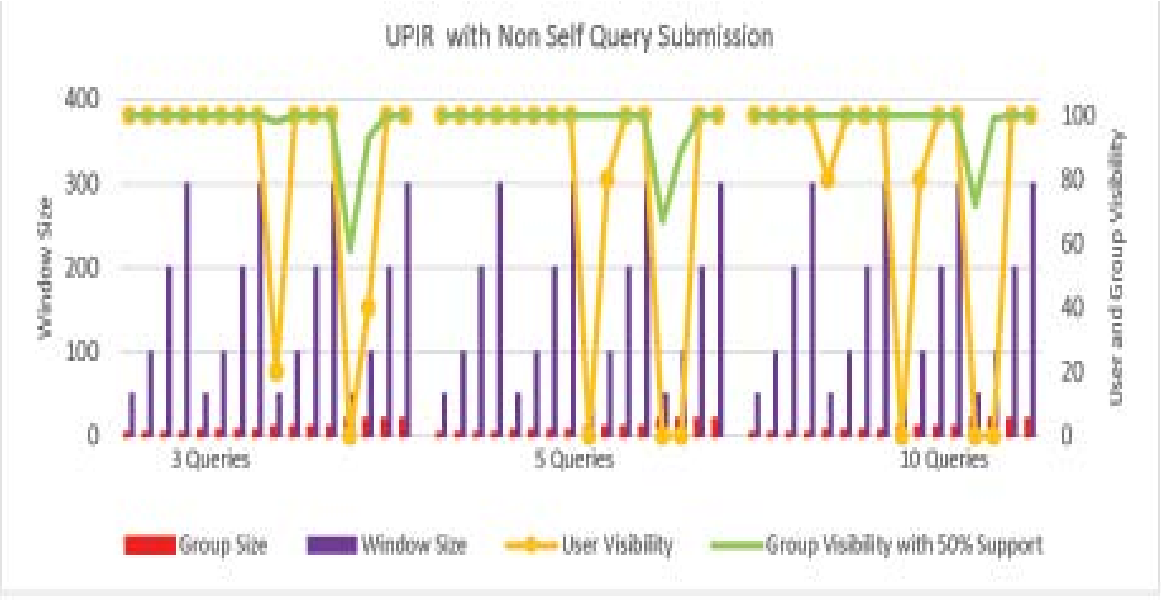


Figure 4. Source user visibility probability and Group visibility percentage in UPIR with Non Self Submission

Figure 4 represents the second group of protocols which contains non self-query submission UPIR and its optimal configuration variant. In Fig. 4, all sub-graphs shows almost same kind of behavior for both source user visibility and group visibility. In all scenarios user visibility and group visibility is more than 80% except the scenarios in which windows size is taken as 50 records where user visibility drop to 0% in some cases while group visibility drop to 80 to 60%.

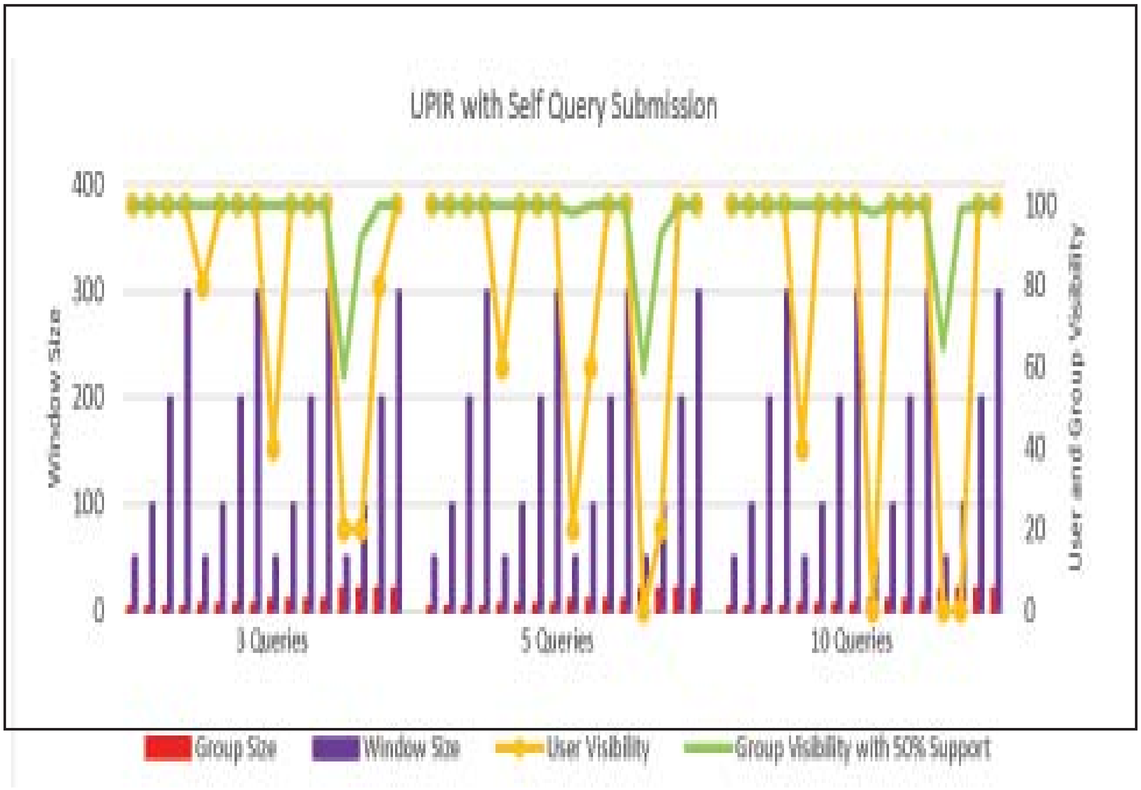


Figure 5. Source user visibility probability and Group exposing percentage in UPIR with Self-Submission

Protocols	No of Queries			Group Size			Window Size					Average
	3	5	10	3	6	10	20	50	100	200	300	
UUP (All selected Variants)	73.	88.8	91.3	93.3	93.3	78.3	81.7	55.0	96.7	98.3	88.3	85.34
UPIR (Non Self Submission)	85.0	81.3	78.8	100	98.3	81.7	55.0	41.7	75.0	100	100	81.5
UPIR (Self Submission)	83.8	78.8	77.5	100	90.0	76.7	53.3	46.7	75.0	98.3	100	80.0
Average	80.9	82.9	82.5	97.8	93.9	78.9	63.3	47.8	82.2	98.9	96.1	-

Table 3. Adversarial Model Accuracy In No of Dversarial Model Accuracy In No of Queries, Group Size And Window Size

Figure 5 represents the third group of protocol which contains self-query submission version of UPIR. The pattern in all sub-graphs is somewhat same. At window size of 50 records, the source visibility drop significantly with the of group size. Similarly at window size of 100 records, user visibility drop to 20%, 33% and 21% for scenarios of 3, 5 and 10 QoI respectively. As far as group visibility is concerned, in most cases more than 90% group members are exposed except in case window size of 50 records where group member visibility is 60 to 65%.

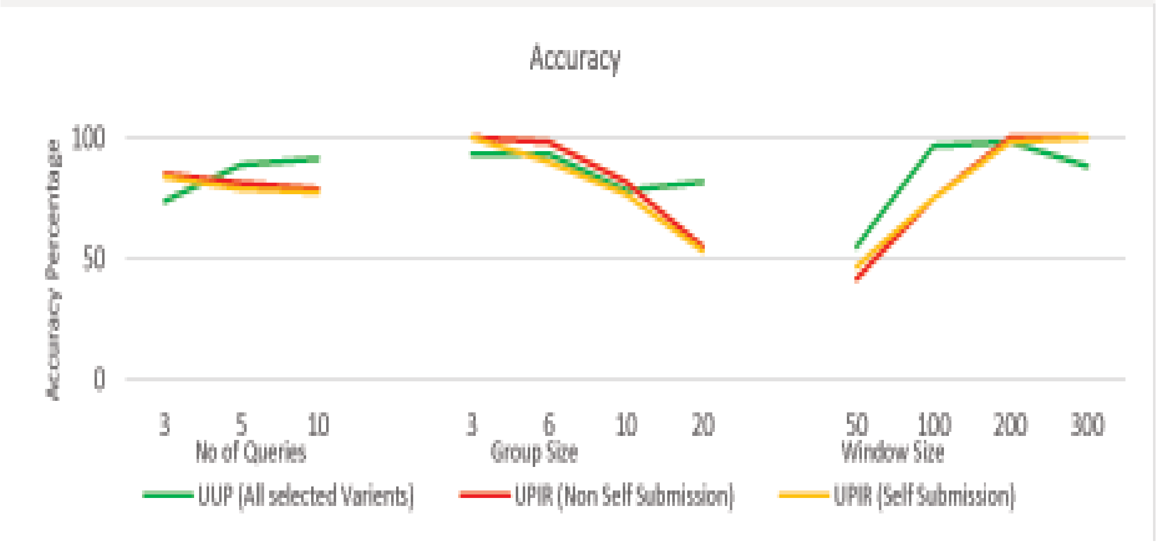


Figure 6. Adversarial Model Accuracy in No of queries, Group Size and Window size

Table III provides the accuracy of the proposed adversarial model in three different aspects i.e. Number of Queries, Group Size and Window size. According to the results, average accuracy for UUP, UPIR non self-query submission and UPIR self-query submission is 85.34%, 81.5% and 80.0% respectively. Similarly with increase in number of queries and window size the accuracy increases while the accuracy decreases as the group size increases.

6. Conclusion And Future Work

This paper studied the problem of identifying the source of QoI which is received through one of the six selected P2P PIR protocols UUP, UPIR and their variants. From the result it is demonstrated that an adversarial search engine can expose the source of the QoI with the accuracy of above 75% using association rule mining technique. The results shows that in every protocol large group size and less number of quires submission offer better resistance against ARM technique. However this can be counter by taking bigger window size. UUP works with dynamic group in which user visibility is easier while in UPIR, user could hide in the group and bit difficult to find out the source user but the group user visibility can narrow down our search. Although this study verified the success of the proposed adversarial model over six protocols, however in nutshell we can say that the P2P PIR protocols can be compromised using Association rule mining techniques and this technique might work on other P2P PIR protocols as well. Therefore it is recommended for the researchers while designing the new protocols, must also consider the association rule mining factor as well in order to preserve user privacy. Similarly time of query submission can also be important. If a QoI source user do not submit other group member query for some specified period of time in that case his/her query might miss the ARM window. The best way to retrieve the information privately from the web search engine using P2P protocols is to use bigger groups and submission of less queries if possible. Another solution could be the use of different search engines for more than two queries.

One of our next goal is to find all the queries submitted by User of Interest (UoI) using any P2P PIR protocols. Privacy preservation is vast area and the researchers of this area must consider all the known aspects which can breach the privacy of the user.

References

[1] Khan, R. , Ali, S.(2013). Conceptual Framework of Redundant Link Aggregation, *Computer Science & Engineering: An International Journal (CSEIJ)*, vol. Vol: 3.

[2] Jenkins, W. F. (1946). A Logic Named Joe, *Astounding*, 37 (1) 139-155.

[3] Peddinti, S. T., Saxena, N (2014). Web search query privacy: Evaluating query obfuscation and anonymizing networks, *Journal of Computer Security*, 22 (1) 155-199.

[4] Seymour, T., Frantsvog, D., Kumar, S (2011). History of search engines, *International Journal of Management and Information Systems*, 15 (4) 47.

[5] Romero-Tris, C., Viejo, A.,Castella -Roca, J (2015). Multi-party Methods for Privacy-Preserving Web Search: Survey and Contributions, Advanced Research in Data Privacy, p. 367-387: Springer.

[6] Pass, G., Chowdhury, A. C. Torgeson, C. A picture of search. p. 1.

[7] Jones, R., Kumar, R., Pang, B. I know what you did last summer: query logs and user privacy. p. 909-914.

[8] Peddinti, S. T., Saxena, N. On the privacy of web search based on query obfuscation: a case study of TrackMeNot. p. 19-37.

[9] Arampatzis, A. ., Drosatos, G., Efraimidis, P.S. (2015). Versatile Query Scrambling for Private Web Search, *Information Retrieval Journal*, 18 (4) 331-358.

[10] Castella -Roca, J., Viejo, A., Herrera-Joancomarta, J. (2009). Preserving user's privacy in web search engines," Computer Communications, 32 (13) 1541-1551.

[11] Domingo-Ferrer, J., Bras-Amoros, M. Peer-to-peer private information retrieval. p. 315-323.

[12] Domingo-Ferrer, J., Bras-AmorÃ³. M, Q. Wu, Q (2009). User-private information retrieval based on a peer-to-peer community, *Data & Knowledge Engineering*, 68 (11) 1237-1252.

[13] Stokes, K., Bras-Amoros, M. On query self-submission in peer-to-peer user-private information retrieval. p. 7.

[14] Lindell, Y., Waisbard, E. Private web search with malicious adversaries. p. 220-235.

[15] Romero-Tris, C., Castella-Roca, J., Viejo, A (2012). Multi-party private web search with untrusted partners, Security and Privacy in Communication Networks, p. 261-280: Springer.

[16] Romero-Tris, J. Castella -Roca, and A. Viejo (2014). Distributed system for private web search with untrusted partners, Computer Networks, vol. 67, p. 26-42.

[17] Stokes, K., bras-Amoros, M(2010). Optimal configurations for peer-to-peer user-private information retrieval, *Computers & Mathematics with Applications*, 59 (4) 1568-1577.

[18] Stokes, K. , Bras-Amoros, M. On query self-submission in peer-to-peer user-private information retrieval. p. 7.

[19] Gervais, A., Shokri, R. Singla, A. Quantifying web-search privacy. p. 966-977.

[20] Petit, A., Cerqueus, Boutet, T. (2016). SimAttack: private web search under fire, *Journal of Internet Services and Applications*, 7 (1) 1.

[21] Piatetsky-Shapiro, G. (1991). Discovery, analysis, and presentation of strong rules, Knowledge discovery in databases, p. 229-238.

[22] Agrawal, R. , Imielianki, T., Swami, A (1993). Mining association rules between sets of items in large databases, ACM SIGMOD Record, 22 (2) 207-216.

[23] Han, J., Kamber, M., Pei, J. (2011) Data mining: concepts and techniques, 3rd ed. Elsevier.