Privacy Protection in Web Search

Rekha Joshi

Abstract— *This paper presents web search has demonstrated in improving the quality of various search services on the internet, user reluctance to disclose the private information during search has become major barrier for the wide proliferation of password. Protection in password authentication model user preferences as hierarchical user profiles, a password framework know as user profile search that can adaptively generalize profile by search query while respecting user specified privacy requirements. Our work provides utility of personalization and the privacy risk of exposing the generalized profile using Greedy algorithm is a method for deciding whether personalizing a query is efficient.*

Keywords— *Data Mining, Web Search, Query Personalization, Greedy Algorithm.*

I. INTRODUCTION

Personalization of information access indeed to face considerable growth of data heterogeneity of the roles and needs to the rapid development of mobile system becomes important to propose a personalized system able to provide user with relevant information need. System must into account the different characteristics of the user and all contextual situations that influence his behavior during his interaction with information system. A generic model of profile access according to which the personalization system is articulated based mainly on profiles context user's preferences. Profiles are knowledge containers context defines a set of parameters that characterize the environment of the system user preferences represent the expectations of the user. Ontology is best the candidate for representing knowledge about users to have a shared understanding between people or software agents of terms and their relations a controlled vocabulary. Ontologies have been proven and effective information means for modeling a user context can be very useful tool because they may present an overview of the domain related to a specific area of interest and used for browsing query refinement, provides rich semantics for humans to work with required formalism for computers to perform mechanical processing. Ontology is used to model the user profile has already been proposed in various applications like web search [3], [2] and personal information management [1]. However, up to this point, ontologies modeling user profiles are application-specific, with each one having been created specifically for a particular domain. Taking into account the continuing incorporation of ontologies in new applications, there is an emerging need for a standard ontology that will model user profiles; this standard ontology will facilitate the communication between applications and serve as reference point when profiling functionalities need to be developed.

Over the past decade growth of information available on the web gathering useful information from the web has become a challenging issue for users. Web users expect more intelligent systems to gather the useful information from the large size of web related data sources, user profiles represent the concept models possessed by users when gathering web information. A concept model is implicitly either local or global analysis method is effective for gathering the global knowledge. Multidimensional ontology mining method specificity for analyzing the concept specified machine-readable documents.

II. RELATED WORK

Many profile representations are available in the literature to facilitate different personalization strategies. Earlier techniques utilize term lists/vectors or bag of words to represent their profile. However, most recent works build profiles in hierarchical structures due to their stronger descriptive ability, better scalability, and higher access efficiency. The majority of the hierarchical representations are constructed with existing weighted topic hierarchy/graph, such as ODP, Wikipedia and so on. Another work builds the hierarchical profile automatically via term-frequency analysis on the user data. Generally there are two classes of privacy protection problems for PWS. One class includes those treat privacy as the identification of an individual. The other includes those consider the sensitivity of the data, particularly the user profiles, exposed to the PWS server. Typical works in the literature of protecting user identifications (class one) try to solve the privacy problem on different levels, including the pseudo identity, the group identity, no identity, and no personal information. Solution to the first level is proved to fragile. The third and fourth levels are impractical due to high cost in communication and cryptography. Therefore, the existing efforts focus on the second level. The useless

user profile (UUP) protocol is proposed to shuffle queries among a group of users who issue them. As a result any entity cannot profile a certain individual. These works assume the existence of a trustworthy third-party anonymizer, which is not readily available over the Internet at large. Viejo and Castell-a-Roca use legacy social networks instead of the third party to provide a distorted user profile to the web search engine. In the scheme, every user acts as a search agency of his or her neighbors. They can decide to submit the query on behalf of who issued it, or forward it to other neighbors. The shortcomings of current solutions in class one is the high cost introduced due to the collaboration and communication. The solutions in class two do not require third-party assistance or collaborations between social network entries. In these solutions, users only trust themselves and cannot tolerate the exposure of their complete profiles an anonymity server. Krause and Horvitz employ statistical techniques to learn a probabilistic model, and then use this model to generate the near-optimal partial profile. Limitation in this work is that it builds the user profile as a finite set of attributes, and the probabilistic model is trained through predefined frequent queries. These assumptions are impractical in the context of PWS. Xu et al. proposed a privacy protection solution for PWS based on hierarchical profiles. Using a user-specified threshold, a generalized profile is obtained in effect as a rooted subtree of the complete profile. Unfortunately, this work does not address the query utility, which is crucial for the service quality of PWS. Xiao and Tao proposed Privacy-Preserving Data Publishing (PPDP). A person can specify the degree of privacy protection for her/his sensitive values by specifying "guarding nodes" in the taxonomy of the sensitive attribute. Teevan et al. collect a set of features of the query to classify queries by their click entropy. While these works are motivate in questioning whether to personalize or not to, they assume the availability of massive user query logs and user feedback.

III. PROBLEM DEFINITION

To protect user privacy in profile-based passwords, researchers have to consider two contradicting effects during the search process. On the one hand, they attempt to improve the search quality with the personalization utility of the user profile. They need to hide the privacy contents existing in the user profile to place the privacy risk under control. Significant gain can be obtained by personalization at the expense of only a small and less-sensitive portion of the user profile, namely a generalized profile. Thus, user privacy can be protected without compromising the personalized search quality. In general, there is a tradeoff between the search quality and the level of privacy protection achieved from generalization. Unfortunately, the previous works of privacy preserving password are far from optimal.

A greedy algorithm is method to provide password security and is a mathematical process that recursion set of objects from the smallest possible methods, problem solving recursion is a solution to smaller instances of the same problem. Greedy algorithm looks for simple easy to implement solution to complex multiple problems by deciding which step will provide the most obvious advantage. Benefits to using a greedy algorithm is that solutions to smaller instances of the problem can be straightforward and easy to understand and entirely possible that the most optimal short-term solutions may lead to the worst long term outcome. Greedy algorithms are often used in ad-hoc mobile networking to efficiently route packets with the number of hops.

Greedy dynamic programming solves by combining the solutions to sub-problems that contain common sub-problems, using Divide and conquers to solve inefficient as the same common sub-problems have to be solved many times. Dynamic programming will solve each of them once and stored in a table for future reference.

Characterize optimal sub-structure

Recursively define the value of an optimal solution

Compute the value bottom up

Construct an optimal solution

Greedy dynamic programming is suitable for problems with optimal substructure consists of optimal solutions to sub-problems and few sub-problems in total many recurring instance of each.

3.1. Methods to search Personalized Data

When employing a server-side personalized search strategy, there are two main opportunities for the personal information submitted to the service to be compromised. The first vulnerable place is during the initial transaction when the user submits their set of personal information to the search provider. If this information is sent to the provider in simple plaintext, then the user's information can be easily intercepted via a packet sniffing mechanism and then used however the interceptor may see fit. The second opportunity for privacy to be lost occurs if a malicious security breach occurs on the servers that house the personal information for the users of the search provider. This breach could lead to the loss of any privacy that users believed they had with their personal information on the search provider's

servers.One basic way of ensuring that users' personal data remains private, in lieu of the outlined security problems, is to encrypt the personal information while in transit between the client/server and while stored in the search provider's database. This method will prevent any personal user information from existing in a plaintext format which is intrinsically vulnerable. Methods to encrypt the personal information and transport it will be discussed later.

Client-side personalized search strategy avoids the privacy risk of storing personal user information on search providers' severs by letting the client maintain and be responsible for their own 'set' of personal information. With this information, the client transports it to the search provider whenever they perform a search. The search provider will then take the received personal information along with the search query and then perform a personalized search for the client. By allowing the user to maintain their own personal data it increases the privacy for the user and thus, the search provider will not have to store a copy of the data on their severs. This allows the search provider to avoid responsibility for the integrity and privacy of this data. This technique it does have a few limitations however. The first issue is that this process is bandwidth intensive. A server-side search strategy needs only to transmit the search query to the provider during each user session. Client-side strategy on the other hand will typically require, depending on how the personalized search service is engineered, the client to submit their set of personal information alongside each search query. Most often the personal information will be vastly larger than the simple 2-5 word search query that the user is submitting. This forces the search provider and the client to deal with a much larger workload of bandwidth then they would have to deal with otherwise. As for the actual privacy concern with this set up, by making the user submit their personal information alongside their search query at every instance increases the chance that the information could be intercepted, like mentioned before, by a packet sniffing mechanism. Unless the transmission was applying a basic security mechanism such as encryption (opposed to allowing the transmission to exist in plaintext), the user's personal information for the search provider will be vulnerable more often then it would be if a server-side strategy was being applied.

3.2. Cryptography in Personalized Data

To handle privacy using encryption for storage of personal user information the following plan could be adopted.

i. Search provider encrypts all personal user information within their databases using their public key.
ii. When needed to perform a personalized search, the specific user's data is withdrawn from the database, decrypted with the search provider's private key and then fed into the program that performs the personalized search.
iii. The instance of that user's personal data that has been withdrawn and currently in plaintext will then be destroyed.

Having the personal information of users exist in plaintext for as little time as possible is the primary goal of this strategy to ensure user privacy. Providing that the search provider's private key can remain private, the provider should be able to maintain user privacy at all times. This system does not account for privacy breaches from within the actual search provider's organization however. An internal attacker may have access to the private key of the organization and thus, find a method of accessing the database and acquiring the personal information of their clients. Securing client-side personalized search is similar to securing the transport phase of server-side personalized search. Each time the user performs a personalized search, the user's information for the search provider will have to be transported in the same fashion as outlined. The only difference here is that the search provider will return the user's queried results and then destroy the user information that was sent to them. As secure as this method may be, extra iterations of encryption and decryption will be necessary as the user is sending their encrypted personal information alongside each of their search queries. This limitation will increase the processor load on both the client machine and the server as they will continually have to encrypt and decrypt the transmissions respectively.

IV. COMPARATIVE STUDY

A user profile is typically generalized for only once offline, and used to personalize all queries from a same user indiscriminatingly. Such "one profile fits all" strategy certainly has drawbacks given the variety of queries. Profile-based personalization may not even help to improve the search quality for some ad hoc queries, though exposing user profile to a server has put the user's privacy at risk. A better approach is to make an online decision on whether to personalize the query and what to expose in the user profile at runtime. This considers, all the sensitive topics are detected using an absolute metric called surprisal based on

the information theory, assuming that the interests with less user document support are more sensitive. They usually refine the search results with some metrics which require multiple user interactions, such as rank scoring, average rank, and so on. This paradigm is, however, infeasible for runtime profiling, as it will not only pose too much risk of privacy breach, but also demand prohibitive processing time for profiling. Thus, we need predictive metrics to measure the search quality and breach risk after personalization, without incurring iterative user interaction. Compare earlier framework our proposed work shows efficient results such as User customizable Privacy-preserving Search framework is a privacy-preserving personalized web search framework, can be generalize profiles for each query according to user-specified privacy requirements. Development in two simple but effective generalization algorithms, GreedyDP and GreedyIL, to support runtime profiling. GreedyDP tries to maximize the discriminating power (DP), GreedyIL attempts to minimize the information loss (IL). This framework assumes that the queries do not contain any sensitive information, and aims at protecting the privacy in individual user profiles while retaining their usefulness for PWS. User privacy preserving consists of a non-trusty search engine server and a number of clients. Each client or user accessing the search service trusts no one but himself/ herself. The key component for privacy protection is an online profiler implemented as a search proxy running on the client machine itself. The proxy maintains both the complete user profile, in a hierarchy of nodes with semantics, and the user-specified (customized) privacy requirements represented as a set of sensitive-nodes.During the offline phase, a hierarchical user profile is constructed and customized with the user-specified privacy requirements. The online phase handles queries as When a user issues a query qi on the client, the proxy generates a user profile in runtime in the light of query terms. The output of this step is a generalized user profile Gi satisfying the privacy requirements. The generalization process is guided by considering two conflicting metrics, namely the personalization utility and the privacy risk, both defined for user profiles.The query and the generalized user profile are sent together to the PWS server for personalized search.The search results are personalized with the profile and delivered back to the query proxy and finally, the proxy either presents the raw results to the user, or ranks them with the complete user profile.

V. CONCLUSION

Our work presents a framework for privacy protection knows as user profile search for personalized web search, potentially be adopted by any password that captures user profiles in a hierarchical taxonomy. This framework allows users to specifycustomized privacy requirements via the hierarchicalprofiles. In addition, user profile search also performed onlinegeneralization on user profiles to protect the personalprivacy without compromising the search quality. Greedy algorithms for the online generalization revealed that user profile search could achieve quality search preserving user's effectiveness of our solution.

REFERENCE

[1] V. Katifori,, A. Poggi,. M. Scannapieco, T. Catarci, & Y. Ioannidis (2005). OntoPIM: how to rely on a personal ontology for Personal Information Management. In *Proc. of the 1st Workshop on The Semantic Desktop*.

[2] S. Lawrence, (2000). Context in web search. IEEE Data Engineering Bulletin, 23(3):25-32

[3] J. Trajkova, S. Gauch, Improving Ontology-based User Profiles, Proc. of RIAO 2004, University of Avignon (Vaucluse), France, April 26-28, 2004, pp. 380-389

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

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

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

[7] X. Shen, B. Tan, and C. Zhai, "Context-Sensitive InformationRetrieval Using Implicit Feedback," Proc. 28th Ann. Int'l ACM
SIGIR Conf. Research and Development Information Retrieval, (SIGIR), 2005.

[8] F. Qiu and J. Cho, "Automatic Identification of User Interest for Personalized Search," Proc. 15th Int'l Conf. World Wide Web(WWW), pp. 727-736, 2006.