

Online Anonymity for Personalized Web Services

Yabo Xu¹, Ke Wang², Guoliang Yang³, Ada W. C. Fu⁴

Sun Yat-sen University¹
Guangzhou, China

Simon Fraser University²
Burnaby, BC, Canada

Guangzhou Research
Institute of China
Telecom³

Chinese University of
Hong Kong⁴
Hong Kong, China

xuyabo@sysu.edu.cn

wangk@cs.sfu.ca

yanggl@gsta.com

adafu@cse.cuhk.edu.hk

ABSTRACT

To receive personalized web services, the user has to provide personal information and preferences, in addition to the query itself, to the web service. However, detailed personal information could identify the sender of sensitive queries, thus compromise user privacy. We propose the notion of *online anonymity* to enable users to issue personalized queries to an *untrusted* web service while with their anonymity preserved. The challenge for providing online anonymity is dealing with unknown and dynamic web users who can get online and offline at any time. We define this problem, discuss its implications and differences from the problems in the literature, and propose a solution.

Categories and Subject Descriptors

H.3.5 [Online Information Services]: Web-based services; K.4.1 [Public Policy Issues]: Privacy

General Terms

Algorithms, Performance, Design, Experimentation, Theory

Keywords

Online anonymity, web services, privacy, personalization

1. INTRODUCTION

The annual personalization survey conducted by Choicestream (<http://www.choicestream.com/news/>) from 2004 to 2006 consistently shows that 80% of consumers were interested in personalized web service and contents. On one hand, personalized web service offers users tailored services according to their personal preference, and thus, is far effective to meet users' need; on the other hand, personalized web service entails gathering considerable amounts of personal information from its users, thus, raise much of privacy concern.

Often, data collected by web services, such as query logs, are excellent candidates for various data mining applications. However, the use of such data raises privacy concerns if the data is not made anonymous enough. An example is the release of AOL query logs (New York Times, Aug 9 2006), where the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CIKM'09, November 2–6, 2009, Hong Kong, China.

Copyright 2009 ACM 978-1-60558-512-3/09/11...\$10.00.

searcher No. 4417749 was traced back to Ms Thelma Arnold. Recently, Google were ordered by a federal judge to turn over YouTube user log to Viacom in a lawsuit (New York Times, July 4 2008). If the user log is not made anonymous enough, video viewing habits of tens of millions of YouTube users may be under the risk of exposure.

To better understand how a user may be identified in a personalized web service, let us consider a toy but concrete example. Suppose that the user Albert submits a *query* $q=\{\text{diabetes, symptoms}\}$ to a vertical search engine like Healthline¹, which provides specialized search on health and medical information. Wishing to get personalized results, Albert registered his *personal information* d on date of birth, gender, zip code as required in the online registration form. Each query leaves a trace $\langle d, q, t \rangle$ on the site's query log, where t refers to the *query time*. We assume that the query log does not contain explicit identifying information of searchers. As a secondary use, the query log is published to a health care company for data mining research, or to the public as in the AOL case. In the following discussion, the *attacker* refers to a party that has access to the query log and seeks to re-identify the (sensitive) queries of Albert, called the *target*. Usually, the attacker has some sort of relationship with Albert, e.g., colleagues, neighbors, friends, enemies, etc. Consider the following two ways of re-identification.

- **Re-identification through personal information** Suppose that the attacker knows that Albert has searched the above site. Also suppose that the attacker knows Albert's date of birth, gender, zip code, which can be acquired from a public source such as a voter list [1]. The attacker then could narrow down the queries issued by Albert by matching this knowledge against the personal information d in the query log. As reported in [1], with as little knowledge as {date of birth, gender, zip code}, 87% of population in US is uniquely identifiable.
- **Re-identification through approximate query time** If the attacker also acquires an approximate query time such as "Albert used medsite two days ago", say from an office conversation, the attacker can further narrow down the candidate queries by excluding the entries in the query log that are not within this time interval.

In the above search scenario, a *personalized query* has two parts $\langle d, q \rangle$. The *query* q contains query terms on which the user wants to get results. This part is unstructured (i.e., free text) and contains sensitive information, meaning that the user does not want to be identified as the sender of the query. The *personal information* d contains demographic data of the user (such as age, gender, zip

¹ <http://www.healthline.com/>

code) and other preference information, and is used to guide the search of results tailored to the user's taste. This part is structured with pre-defined semantics and is usually obtained via a user sign-up interface. Note that this scenario is distinguished from the personalization based on information on the web service side such as query histories, which is beyond users' control.

The flow of a re-identification attack described above is illustrated

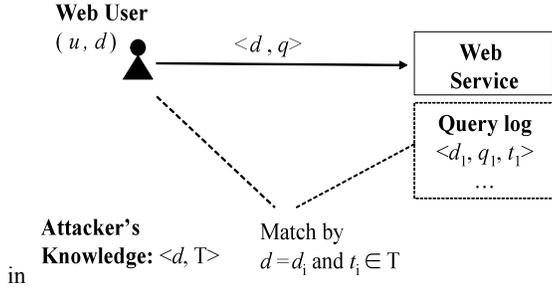


Figure 1. A web user u submits a personalized query $\langle d, q \rangle$ at time t to a web service. Over time, the web service collects a query log containing all entries $\langle d_i, q_i, t_i \rangle$. At a later time, the query log is used or published for data mining research. One recipient of the query log (possibly the web service itself), the attacker, has the *prior knowledge* $\langle d, T \rangle$ about some target user, that is, the attacker knows that the target user with the personal information d has issued some queries at the approximate query time T . The attacker's goal is to identify the target user's queries from the query log. The anonymity of the target user is compromised if a small number of entries $\langle d_i, q_i, t_i \rangle$ in the log match the prior knowledge $\langle d, T \rangle$.

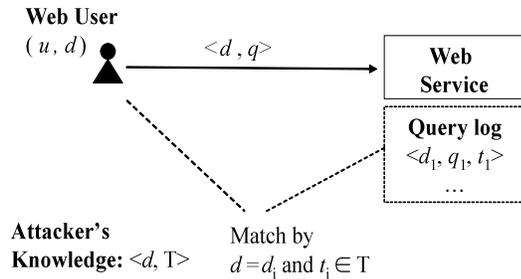


Figure 1. Attacks in personalized web services

Starting from end users' point of view, a key assumption in this work is that the web service is *untrusted* due to the collection of potentially identifying personal information d . Specifically, we adapt the *semi-honest* model [6] for the web service: the web service will follow the specified computation, but may seek to use the collected query log to identify the query of a target user. This assumption ensures a stronger privacy guarantee on the query log so that it can be used by the web service or published to third parties without causing end users' privacy concerns.

Our approach Under the assumption of the untrusted web service, there is a privacy threat because the web service owns both the personal information d and the query q . A detailed d may link a unique user or a small number of users to q . To break the link between the two, we introduce an untrusted third party called *user pool*, which also follows the semi-honest model. Instead of sending $\langle d, q \rangle$ to the web service directly, the user u first anonymizes d through the user pool and then sends $\langle d', q \rangle$ to the

web service, where d' is some generalization of d . The goal is to ensure that the generalized personal information d' cannot be linked to the user. We assume that all communications between a user and the web service/user pool are anonymous [2]. As a result, the user pool possesses the raw personal information d , but has no knowledge about the query q that u may send. The web service possesses the query q and the generalized personal information d' , but cannot identify u from d' because d' has been generalized.

Our contribution is summarized below.

Contribution I We introduce the notion of *online anonymity* to ensure that each query entry $\langle d', q, t \rangle$ in the query log cannot be linked to its sender. Specifically, $\langle d', q, t \rangle$ has (k, w) -online anonymity if at least k distinct users have issued a query using the generalized personal information d' and within the w proximity of the query time t . Therefore, if the attacker's knowledge T about query time is not more accurate than w , *all* of these users are possible candidates for the sender of $\langle d', q, t \rangle$. We will show that online anonymity provides defense against the web service.

Contribution II We propose an algorithm that achieves online anonymity through the user pool. A significant challenge comes from the assumption of untrusted web service and user pool, and dealing with the dynamic sets of online users. Specifically, to provide online anonymity, the user pool must track the online users who issued queries during a certain time interval and anonymize their personal information d in an online fashion. This tracking also entails some interaction between the user pool and web users. We propose a protocol for this interaction to guarantee that the additional information collected by the user pool cannot be used to compromise user anonymity.

Contribution III Although we focus on anonymizing the personal information d that is separately provided for the personalization purpose, in the same spirit, our approach can be extended to deal with personally identifying information that may be contained in the query d . In this sense, our work is also applicable to general web services where there is a need to anonymize the query, with or without personalization. We will discuss this extension in Section 3.3.

2. RELATED WORK

In *privacy preserving data publishing* [1], a trusted party, called publisher, collects all data (i.e., query logs) first and anonymizes the data for publishing. This scenario is not applicable to our web setting where personal information is held by individual web users and there is no trusted data publisher. To put our scenario into the context of data publishing, the query entries in the query log are data records, but they must be generalized *before* they are submitted to the web service. This distributed setting of data is similar to [3]. However, [3] considers a pre-defined set of users and cannot deal with the open web scenario where there is no pre-defined set users because users get online and offline arbitrarily.

In *anonymous communication* [2], systems aim to provide a communication channel for users to interact with the web service anonymously. Similarly, *privacy-preserving data collection* [7] addresses respondents' anonymity in a data collection process. All of these works do not address the re-identification of data subjects from the *content of data* transmitted. In contrast, our work assumes communication anonymity as the infrastructure and focuses on re-identification attacks arising from examining the content of data.

Another body of work makes use of an alias, including *anonymous user accounts* [4], *digital pseudonyms* [5], and *anonymous web browsing* (<http://www.anonymizer.com>). In the scenario of personalized web service, personal information is *required* for personalization and it is such information that links the queries to their senders. This threat exists independently of communication channels, pseudonyms, and user accounts used.

3. THE FRAMEWORK

This section describes our personalization framework, the assumptions and privacy notion used in this paper.

3.1 Infrastructures

We consider a timeline labeled by a sequence of *time units* denoted by 1, 2, 3, A time unit could be a second, a minute, an hour, or a fraction of such units. A time interval or *window* is a sequence of consecutive time units. The window size refers to the number of time units in the window. Figure 2 depicts the basic components and flow of information in our framework.

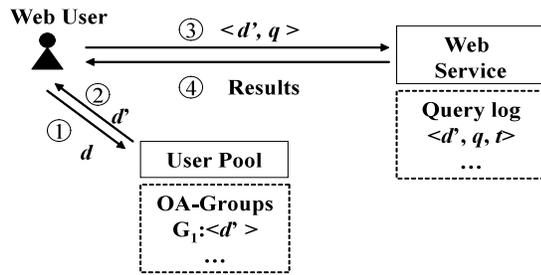


Figure 2. The framework

Web service and web users We consider a *web service* and a collection of *users*. A user initiates a query q to the web service, attached with his personal information d . Unlike database queries, a web query q consists of several query terms and is unstructured, unedited, and lack of pre-defined semantics. On receiving $\langle d, q \rangle$, the web service returns personalized services (or results) to the user. Over time, the web service collects and maintains a *query log* that contains all entries $\langle d, q, t \rangle$ ordered by the query time t .

The web service follows the *semi-honest model* [6]: it will follow the specified computation but may attempt to identify the queries sent by a user. We assume that there is an anonymous communication channel [2] between each user and the web service. This means that queries and results will be transmitted between users and the web service as expected, but the web service has no way to know the user behind a query and has no trace of different queries from the same user by observing where a query comes from.

User pool The function of *user pool* is to pull all web users together and determine for each web user the disclosure of personal information d in order to access personalized web services anonymously. We assume the *semi-honest model* for the user pool and an anonymous communication channel between each web user and the user pool. By choosing the semi-honest model for the user pool, we cover the case that the user pool may also seek to identify the sender of a query or even collude with the web service to do so. See more discussions on attackers below.

In implementation, the user pool can be hosted by either some third party as a public service, or by web services to offer their users anonymity so as to gain a competitive advantage over competitors. We envisage that the adoption process of the user pool might be similar to that of OpenID (<http://openid.net>), a shared identity service that allows Internet users to log on different web services using a single digital identity. It initially arose from open source community as an open and free service, but later gained its popularity among large sites with large organizations such as Google and Microsoft as providers.

The information flow is described in Figure 2. (1) Prior to sending a query q to the web service, a web user must first register with the user pool his personal information d . (2) The user pool returns to the web user the *generalized* personal information d' . d' contains less, but semantically consistent information, than d . For example, if d contains “Age=25”, d' may contain “Age in [20,30]”. (3) Subsequently the web user submits the generalized personalized query $\langle d', q \rangle$ to the web service. (4) Upon receiving $\langle d', q \rangle$, the web service returns the result to the user. As a result, the query log contains generalized query entries $\langle d', q, t \rangle$, instead of the raw query entries $\langle d, q, t \rangle$.

Attacker An *attacker* is a party that seeks to identify a query sent by a *target user* u from the query log. To do so, the attacker has the following information:

- **Query log.** We assume that the web service has published the query log (for research purpose) and the attacker is one of the recipients.
- **Personal information d of u .** The attacker has obtained the personal information d of u as public knowledge.
- **Approximate time interval T_q during which u has sent a query q .** The attacker knows that u sent a query q within a time interval T_q (but does not know the content of q). Often, T_q is an interval containing the actual query time because the attacker knows an approximate query time, but not the exact query time. The size of the interval T_q indicates the “power” of the attacker.

$\langle d, T_q \rangle$ is called *prior knowledge* of the attacker about u . Note that our semi-honest model for the web service and user pool covers the case that these parties may be the attacker if they obtain the above information. For example, the user pool could be one of the recipients of the published query log. In fact, the web service and the user pool may even collude to identify a user’s query.

The temporal accuracy of the attacker’s knowledge T_q about query time can be modeled as follows.

Definition 1 Consider the attacker with prior knowledge $\langle d, T_q \rangle$ on a query q , we say that the attacker is *w-oblivious* if $[t-w, t+w] \subseteq T_q$, where t is the actual query time of a query q . ■

In other words, a *w-oblivious* attacker has at least $\pm w$ error around the actual query time. The smaller the w is, the more precise the attacker’s knowledge T_q is about the query time, therefore, the more powerful the attacker is. Our goal is to provide users anonymity against the *w-oblivious* attacker for a given w .

3.2 Online Anonymity

Given the query log and the prior knowledge $\langle d, T_q \rangle$ on a target user u , the attacker tries to narrow down the candidate entries

$\langle d', q, t \rangle$ (in the query log) originating from u . Such entries must match $\langle d, T_q \rangle$, denoted $d \in d'$ and $t \in T_q$, that is, d' is a generalization of d and T_q contains the query time t . In general, not all matched entries originated from u because other users have similar generalized personal information d' and have issued a query within T_q . The more matched queries were sent by *different* users, the less certain the attacker is about whether a matched query was actually sent by u .

Definition 2 (k -identification) We say that the target user is k -identified if the queries matching $\langle d, T_q \rangle$ in the query log were sent by less than k distinct users. ■

To prevent k -identification, we require that, for each query $\langle d', q, t \rangle$ in the query log, there are at least k “similar” queries sent by distinct users: these queries share the same generalized personal information d' and were sent within time proximity from t that is not distinguishable by the attacker. This motivates the following privacy notion.

Definition 3 ((k, w) -online-anonymity) A query $\langle d', q, t \rangle$ in the query log is said to have (k, w) -online-anonymity if there are at least k queries $\langle d'_i, q_i, t_i \rangle$ in the query log such that each $\langle d'_i, q_i, t_i \rangle$ was sent by a distinct user, $d'_i = d'$, and $|t - t_i| \leq w$. The query log is said to have (k, w) -online-anonymity if all queries in the log have (k, w) -online-anonymity. ■

Theorem 1 If a query log has (k, w) -online-anonymity, for any query in the log, the sender of the query is not k -identified by any w -oblivious attacker.

Proof: Consider a query $\langle d', q, t \rangle$ in the query log with (k, w) -online-anonymity. Suppose that the attacker has the prior knowledge $\langle d, T_q \rangle$ about u and q . Since the attacker is w -oblivious (Definition 1), we have $[t-w, t+w] \subseteq T_q$. (k, w) -online-anonymity of $\langle d', q, t \rangle$ implies that at least k queries were sent within the interval $[t-w, t+w]$ by distinct users and those users share d' (Definition 3). All these queries match $\langle d, T_q \rangle$ because $[t-w, t+w] \subseteq T_q$. Since these queries were sent by k distinct users, from Definition 2, the sender of $\langle d', q, t \rangle$ is not k -identified. ■

3.3 Extension on General Queries

So far, we consider only attacks based on the personal information d provided for personalization. Sometime, the query q itself may contain personally identifying information. We can extend the notion of online anonymity to cover personally identifying information contained in the query q .

Suppose that the query q can be divided into two parts. The *private sub-query* of q^s refers to the set of private terms in q that the user wants to submit as it is and does not want to be identified as the sender. Such terms typically refer to financial information, health information, religion and political beliefs. The *public sub-query* of q^p refers to the set of public terms in q that may potentially identify the user and the user is willing to modify. For example, for a query $q = \{\text{“stripper club, Redmond WA”}\}$, q^s could be “stripper club” and q^p could be “Redmond WA”. We assume that public/private terms can be specified by the user.

With the above partition $\langle q^p, q^s \rangle$ of q , we can treat the public sub-query q^p as an extension of the personal information d and

generalize both d and q^p using our method. Unlike d , q^p is unstructured (i.e., free text). However, the issue of how to anonymize d and q^p is orthogonal to our approach in that it is entirely local to the user pool and any generalization algorithm can be plugged into our approach. Our focus is on the challenge of providing online anonymity in the open and dynamic web setting without assuming a trusted web service. The implication of this extension is that our approach is applicable to general web services, with or without personalization. In the absence of personalization, d and d' are empty, and anonymization focuses on the personally identifying information in the query q , i.e., q^p .

Due to the space limit, we are unable to describe how to achieve online anonymity in detail. The core idea is to group web users with similar characteristics, both on personal information and query time, so that their query entries in the query log provide (k, w) -online-anonymity for each other. The algorithms and evaluation will be presented in the full version of this paper.

4. CONCLUSION

This paper was motivated by two emerging trends: web users want personalized services and web users want privacy. One challenge is that personal information must be made anonymous under the assumption that the participating parties, including the web service, are not completely trusted, due to systematic collection of personal information in addition to queries. Another challenge is the online and dynamic nature of web users. We proposed the notion of online anonymity to protect web users and we proposed an approach to maintain online anonymity through time. Our approach makes use of a third party called the user pool and we do not require the user pool to be trusted. The simulation study on real US demographics showed promising results: it is feasible to achieve personalization for reasonable privacy settings.

5. REFERENCES

- [1] L. Sweeney. K-Anonymity: a model for protecting privacy. Intl. Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10(5), 2002.
- [2] L. Ahn, A. Bortz, and N. J. Hopper. K-anonymous message transmission. ACM Conference on Computer and Communications Security, 2003.
- [3] S. Zhong, Z. Yang and R. N. Wright. Privacy-enhancing k-anonymization of customer data. PODS 2005.
- [4] E. Gabber, P. B. Gibbons, A. Mayer, Y. Matias. How to make personalized Web browsing simple, secure, and anonymous. Proceedings of Financial Cryptography 1997.
- [5] A. Kobsa, and J. Schreck. Privacy through pseudonymity in user-adaptive systems. ACM Transactions on Internet Technology, 2003.
- [6] O. Goldreich. *Foundations of Cryptography*, Volume 1. Cambridge University Press, 2001.
- [7] Z. Yang, S. Zhong, R. N. Wright, Anonymity-preserving data collection. SIGKDD 2005.