Using a secure element to protect the users' profiles generated by web search engines

Mikel Iturbe[1], Jordi Castellà-Roca[2], and Alexandre Viejo[2]

[1] Universitat Oberta de Catalunya
Av. Tibidabo 39-43 E-08035 Barcelona, Spain
[2] Universitat Rovira i Virgili, UNESCO Chair in Data Privacy,
Departament d'Enginyeria Informàtica i Matemàtiques,
Av. Països Catalans 26, E-43007 Tarragona, Spain

Abstract. Web search engines (WSEs) are valuable tools that are widely used to find specific information in the World Wide Web. Recently, they have increased search result relevance by personalizing them according to the users' interests. Nevertheless, WSEs also pose an important privacy threat, as they profile users by storing and analyzing their previous search data. To address this privacy problem, current solutions propose new mechanisms that add a significant computation and communication overhead, and/or lack personalized search results. In this paper we present a server-side web search model that serves personalized search results while preserving the privacy of the users. In this model, a mechanism called the secure element (SE) acts as an intermediary between the web search engine and end users. The secure element forwards queries from the users to the WSE and later re-ranks search results according to the user's previous search behavior. All communication between the users and the secure element is encrypted to prevent eavesdropping. A privacy analysis shows that the scheme effectively protects users from being profiled by WSEs or external attackers.

Keywords: web search, privacy, secure element

1 Introduction

With an exponential growth of The World Wide Web, Web Search Engines (WSEs) have become an important tool for users that want to find content effectively on the Internet. In the searching process, the WSE provides several results related to the query introduced by the user. However, query text alone is usually not selective enough [16] to craft an ordered result list, as ambiguities can occur between different search terms (e.g. homographs) or as broad search queries can return an unmanageable number of results. Accordingly, it is necessary to deal with those issues by choosing which search results are relevant in order to sort them when the user retrieves them. This can be achieved by ordering the results according to their importance in the web, as Google's PageRank [19] does, and/or ranking the search results corresponding to each user interests, i.e. personalizing them.

In order to improve search results personalization, it is necessary to effectively identify users' interests that will allow ranking search results according to them. These personal interests are deduced from each user's previous behavior in searches, which is stored by the WSE. That is, the WSE creates a user profile based on their previous behavior [26,27], known as behavioral profile. This gives WSEs the possibility of refining search results for individual users that differ in interests or behavior. As a consequence, users receive tailored search results that suit them best.

As useful it might be, user data collected by WSE compromises users' privacy, as it can be used to directly identify individuals or extract important insight about their interests. Search queries can contain sensitive information such as confidential data (sexual tendency, religion, health issues. . .), identifying data (full name –collected when users' egosurf[3]–, ID or passport number. . .), quasi-identifying attributes (age, address, place of birth. . .) or even details about the users' everyday life they might not be comfortable sharing (hobbies, interests. . .). Based on this data, it is possible to link confidential data to a uniquely identified individual and extract valuable information about them, posing an important privacy concern. According to research presented by Sweeney [25] and Golle [11], it is possible to identify the majority of the United States population based on three pieces of personal information: gender, ZIP code[4] and date of birth. The 2006 AOL scandal [2] is a clear example of the real, existing risk of identifying an individual based on search data.

In addition, some WSEs such as Google, use gathered user data for commercial purposes and share it with third parties [12]. Plus, other popular search engines (Yahoo, Bing) use their search data to serve personalized advertisements to users [30,4]. Other privacy risks associated with behavioral profiles crafted by the WSEs are listed by Toch [27] and include unsolicited personalized advertisements or the disturbance "the mere feeling of being tracked" among many others.

Therefore, WSEs have little interest in preserving users' privacy, as gathered personal data is an important source of financial income. Likewise, in case of a WSE data leakage, personal information regarding millions of users would be made public. Those threats constitute an important privacy issue that has to be managed in order to protect users from leaking their personal data.

In consequence, it is necessary to implement a solution that will give users the advantage of receiving relevant search results according to their interests and, to do so while preserving their privacy.

[3] *Egosurfing* refers to the practice of searching for one's name in a WSE in order to check its presence on the Web.

[4] In the United States, ZIP codes refer to postal codes used and assigned by the US Postal Service.

2 Related work

The aforementioned problem has received wide attention from the scientific community and different privacy preserving mechanisms have been proposed. Those solutions can be classified in two big groups: solutions where an honest, collaborating WSE is assumed and solutions where no collaboration is expected and a dishonest WSE is assumed.

In the first group, several server-side [29] and client-side [23] solutions have been proposed. The mechanism proposed in [29] builds a hierarchical user profile of the users and each user decides what part of this profile is exposed to the WSE. However, this exposed portion of the profile can be enough for a dishonest WSE to gather important insight about the users. In [23], authors present a tool, UCAIR, that re-ranks search results on the client side. However, in this approach, search data is sent to the WSE directly, and therefore the system does not protect users' profiles from a dishonest WSE. Another approach, consists on using single-database Private Information Retrieval (PIR) [6], protocols, an idea first presented by [17]. Nevertheless, as it has been stated above, it is not realistic to trust whether a WSE will collaborate in providing user privacy. Hence, it is a safer option to pick out the opposite choice, that is, to assume WSEs will actively profile users.

In the second group, most research has been conducted on client-side solutions, where the issue has been addressed using two main approaches:

1. Search query obfuscation
2. Anonymous web browsing mechanisms.

Search query obfuscation. In this approach, two different types of mechanisms can be listed: systems where machine-generated queries are sent to the WSE, and mechanisms where users do not send search queries directly, as they are sent by other users on their behalf. The first case generally consists in adding clutter to user search queries. TrackMeNot [13], is an example of this approach. However, this approach has been proved to be ineffective, as it is possible to distinguish between user-generated and machine-generated data with high confidence [20]. In the second case, Castellà-Roca et al. [5] presented the Useless User Profile (UUP) protocol, later improved in [21], where users in a group would send queries that belonged to others. Viejo et al. [28] proposed a mechanism to use social network peers to send query data, which later [10] showed that was feasible to implement. However, whereas they provide effective privacy, these solutions do not provide personalized search results.

Anonymous web browsing mechanisms. These mechanisms anonymize the users' web browsing activity by using reliable cryptographic protocols. Examples would include onion routing systems such as Tor [8] or Private Web Search (PWS) [22], a Mozilla Firefox add-on that sends user queries through the Tor relay network without redirecting all general browsing through it. This approach does not offer personalized results nor fast responses.

As previously stated, most research has been conducted on the client-side, whereas server-side solutions remain relatively small in number. An example of implementation is the now defunct `Scroogle.com`, who worked as an intermediary between Google and the users, by giving their metadata, instead of the users', scraping the search results to later serve them without advertisements. Scroogle acted like a proxy, sending different users' search queries to Google with its own metadata. This approach proved to need significant computing resources in order to manage the users' connections. Moreover, it was not able to serve personalized search results to the users. In addition, in this model, the privacy issue is not fixed, as it only moves from the WSE to the proxy server; the proxy itself can still profile users.

Baeza-Yates et al. [1] presented a solution based on ad-hoc crowds as a mean to ensure web search privacy and personalization. In this scheme, user groups are formed around a common interest, represented by the search query (hence the ad-hoc term), that is later used to generate insights. Privacy is ensured as those crowds are big enough to ensure k-anonymity, a widely used anonymization model presented by Sweeney [25]. In this model, a particular query of a user cannot be distinguished from at least other $k - 1$ queries generated by other users. Nevertheless, this approach can lack effectiveness when some very unusual interests are not able to gather a user crowd that is big enough to ensure anonymity. Plus, it can be computationally expensive to classify user queries into different interests.

2.1 Contribution and plan of this paper

As it has been described, little research has been conducted in the field of server-side privacy enhancing personalized search solutions, let alone solutions that assume a dishonest WSE. No proposed solutions serve personalized search results while effectively preserving users' privacy from a dishonest WSE.

Hence, our aim is to build a server-side privacy preserving mechanism, a Secure Element (SE), that meets user privacy and personalization needs. This mechanism should be effective even when an untrusted WSE tries to gather user data. As a server-side solution, this approach could be directly used by users, as it does not require to run any specific software or tool in the local machine.

In order to increase the deployability and scalability of the mechanism, the SE should be as lightweight as possible. This approach would keep computation and communication overhead at a minimum.

In our proposal, users submit their queries through the Secure Element to the WSE. The WSE answers with the query's search results and the SE is the responsible to personalize those results, that is, re-ranking them according to the interests of the user. As stated above, the WSE should not be able to profile users based on gathered user search data, that is, it should not know which queries belong to whom. A cryptographic protocol, described in section 4 is used for this end.

The rest of the paper is organized as follows. Section 3 gives some background on different re-ranking schemes. Section 4 introduces our model by presenting the entities, privacy needs and a general overview of the proposal. Section 5 covers the detailed description of the protocol used for preserving privacy. Section 6 analyzes the security and privacy offered by the system. Finally, Section 7 draws some conclusions and final remarks.

3 Background

As stated previously, the SE is going to be the responsible of re-ranking search results to provide personalization to end users. Thus, it is necessary to analyze different re-ranking methods in order to choose the one that fits best in terms of re-ranking performance and overhead.

Several re-ranking approaches have been proposed in order to provide WSE users personalized search results. They differ both in terms of effectiveness and the user information they use to re-rank search results.

Some of them, have been proposed based on the familiarity of everyday web usage of the users, not directly related to web search. One of this re-ranking approaches was presented by Eickhoff et al. [9]. This model is based on defining atypical web searches, that is, queries related to topics which the user is not familiar with or does not normally interact with. The authors showed that these atypical web searches give more insight about the users than traditional profiling techniques. Automation of the process can be achieved through classification.

Another profiling approach based on the general web usage of the users was proposed by Nagpal et al. [18]. In this scheme, in order to create personalized web search results, the Twitter feed and email content of the user is mined. The resulting tool, called SLAN, works on the client side (thus eliminating users' privacy concerns) and also eliminates spam by filtering Twitter and email content. This approach yields similar or better results than commercial search engines.

Bennet et al. [3] present an hybrid technique to tune search results based on the user search history. This technique uses the user long (historic) and short (session) search history to curate search results. By using long-term data to rerank search results at the session beginning and short-time data as the session evolves, the presented model is shown to work better than historic or session data alone. This point is also backed by data provided by Eickhoff et al. [9].

This last technique is particularly interesting, as individual users can benefit of personalized web search data, even when they perform unique web searches. Moreover, compared to [1,9] it represents a lower cost, in terms of computing and communication. And, having in mind the re-ranking is computed at a secure(trusted)-element, there is no need of using extra privacy enhancing techniques.

At the same time, [9] is partly based in [3], as it uses the session and historic data for re-ranking once the atypical queries have been defined. However, automatically defining atypical queries can be a task that produces an important computational overhead that could not be feasible in our secure-element model.

Bennett's approach [3], is also valid for users with little variance in search terms, that is, for user or users without Twitter or email accounts as opposed to

[9,18]. Thus, it is only the users search history what deals with the re-ranking process.

For these reasons, the work presented in [3] is the most suitable according to our needs and is the one used in this work.

In order to guarantee users' privacy, a cryptographic protocol is defined that will prevent WSE and external attackers access to user data.

4 System model

4.1 Entities

In our scenario, three different entities are present:

- *Users.* They are the individuals who submit queries to the web search engine. Their motivation is to protect their own privacy.
- *Web search engine.* This is the server that performs the search the users have requested serving some results. As it has been stated, this entity has no motivation to preserve users' privacy.
- *Secure element.* It is the entity that personalizes the search results provided by the search engine while preserving users' privacy. It is an external and trusted element that works with the WSE in a distributed form in order to serve search results. It is managed by a third party with no affiliation to the WSE and as such, it is assumed that the SE cannot be modified or altered by the WSE. Its motivation is to protect users' privacy.

In our approach, users would connect to the SE who would act as a central node that provides privacy to users. This central node is the one that will communicate with the WSE in order to protect users' profiles.

4.2 Security and privacy requirements

We identify two different adversaries that threaten users' privacy in this scheme:

- *Web search engine.* Presented as an entity that performs web search, it is also the main adversary in this model. It gathers users' data in order to profile them for its own purposes.
- *External attacker.* Refers to the malicious user that, although does not take part in the querying process, does monitor the activity of legitimate users in order to extract meaningful insight from it. It can later send the collected data to a WSE or use it for other purposes.

Therefore, it is necessary to protect users' privacy from both adversaries to prevent their profiling. However, this should not pose a handicap to serve personalized search results to each user according to their interests.

4.3 Protocol overview

The proposed protocol preserves the privacy of users that perform web search queries by using a secure element that acts as an intermediary between the user and the WSE. This way, the SE sends the query on behalf of the user. Once the WSE performs the search and serves the search results, those results are re-ranked by the secure element, based on the previous query data provided by the user. Communication between the secure element and the user remains encrypted using a public key system in order to prevent eavesdropping. Therefore, the SE needs to have a public-private key pair. The user also verifies the integrity of the received search results to prevent data tampering.

In order to guarantee its viability, it is necessary to keep two qualities of the protocol at a minimum:

- Response time. The system must provide personalized search results within an acceptable time, compared to a WSE-only response time.
- Resource consumption. The mechanism must be as simple and lightweight as possible by reducing computing and communication overhead to a minimum. This would assure the viability of the mechanism with limited resources, even when a high number of users use the service.

5 Our proposal in detail

This section describes the process of submitting a search query thoroughly. Let us imagine a user U_i that wants to submit a query Q using our mechanism. A protocol execution consists of the following steps:

5.1 Initialization

In the first phase, the parameters for a secure communication are established between U_i and the secure element. Previous research shows that, WSE users average more than two search queries per session (2.84 according to [15] and 2.3 according to [24]). Moreover, the number of single-query sessions are declining, as users submit more queries per session than than they used to [14]. Therefore, it is useful to establish a session key that will be used to symmetrically encrypt the content between U_i and the SE, similar to SSL/TLS [7]. This way, we use less resources than encrypting query data asymmetrically. In this initialization phase, a session key is established:

1. U_i connects in a t_c time to the secure element
2. U_i sends a list of cryptographic cryptosystems it supports
3. The SE accepts the connection and chooses a suitable cryptosystem
4. The SE sends its public key to U_i
5. U_i creates a session key K_{SE,U_i} and sends it to the SE encrypting it with the public key of the SE
6. A secure communication channel is established, from now on all communication between the SE and U_i is encrypted symmetrically using K_{SE,U_i}, known by both recipients.

5.2 Sending the query

Once the parameters of the secure connection have been established, it is necessary to send U_i's query, Q, to the WSE. We introduce a small random latency time t_l so the WSE does not know when has U_i submitted the query. This phase works as follows:

1. U_i pads Q and later encrypts it using K_{SE,U_i} in order to build a standard-size ciphertext: $C_Q = E_{K_{SE,U_i}}(Q)$
2. Similarly, U_i pads and encrypts previous search queries $Q^i = Q'_{0^i}, \cdots, Q'_{m^i}$: $C_{Q^i} = E_{K_{SE,U_i}}(Q^i)$ These queries will later be used by the SE to re-rank search results.
3. U_i sends the C_Q and C_{Q^i} cryptograms to the SE
4. The SE decrypts C_Q and C_{Q^i} thus obtaining Q and Q^i
5. The SE waits a random t_l time before sending the query
6. The SE submits the unencrypted original Q to the WSE
7. The WSE gives the search results r of the query Q to the SE
8. The SE re-ranks r using the model presented in [3], using previous query data $Q^i = Q'_{0^i}, \cdots, Q'_{m^i}$ provided by U_i and thus creating the personalized search results $r' = \{l'_1, l'_2 \ \cdots l'_m\}$

5.3 Obtaining a result

After the SE has received and re-ranked U_i's search results, the third and last phase of the protocol consists of serving this data to U_i. As in the previous step, all communication data is encrypted and a random latency time ($t_{l'}$) is used to mask when the results have been received. Plus, a message digest is used to verify message integrity:

1. The SE calculates a digest of the personalized search results by using a hash function: $d = h(r')$
2. The SE encrypts r' using the previous session key K_{SE,U_i}: $C_{r'} = E_{K_{SE,U_i}}(r')$
3. The SE encrypts d using the session key K_{SE,U_i}: $C_d = E_{K_{SE,U_i}}(d)$
4. The SE waits a random $t_{l'}$ time before sending the response to U_i
5. The SE sends $C_{r'}$ and C_d to U_i
6. U_i decrypts $C_{r'}$ and C_d using K_{SE,U_i}.
7. U_i calculates a digest of the r' results that has received: $d' = h(r')$
8. U_i Verifies the integrity of the received search results by confirming that $d = d'$
9. If $d \neq d'$, U_i asks to the SE to resend $C_{r'}$ and C_d. Thus, the protocol goes back to step no. 5
10. After a t_r time, U_i can now browse through the personalized search results.
11. If U_i wants to submit another query, it goes back to *Sending the query* phase
12. If not, U_i disconnects from the SE in a t_d time
13. The SE erases relevant user data belonging to U_i: Q, Q^i, r and r' along with session key K_{SE,U_i}.

6 Security and privacy properties

This section covers the privacy evaluation of the presented protocol against the adversaries that have been defined on Section 4.2.

The main goal of our proposal is to preserve users' privacy when submitting queries to a WSE. Therefore, a successful attacker would be able to link a certain query to the user who has submitted it.

We assume that the attackers do not have enough computer power that would allow them to break current computationally secure cryptosystems. We will also assume the attackers have access to the t instants where:

- A user connects to the SE
- A user sends a message to the SE
- A user receives a message from the SE
- A user disconnects from the SE

6.1 Against the web search engine

As stated is Section 4.2, the WSE is the main adversary in this model. The WSE will try to profile users with all the data it has access to. The SE is managed by an independent third party, and consequently, the WSE has no access to the inner data of the SE (keys, queries that are being processed etc.). Still, the WSE receives all the queries through the secure element. Therefore, based on the submitted queries, the WSE could profile the SE instead of the end users. However, provided that the SE has a heterogeneous and large enough user-base, the created profile would be rendered useless, as the generic profile the WSE has crafted would not represent any of the individual users.

The WSE can also track user activity by monitoring the instant a user sends a message to the SE and the moment the WSE receives it from the SE. By establishing a mean delay between both instants (emission and reception), the WSE could identify the user that has sent a specific query based on those two timestamps. The same could be applied to response times. Nevertheless, the t_l and $t_{l'}$ random latency times change response time for each transmission, and thus, the WSE would not be able to confidently determine who has sent each query, as response times will vary from the mean delay time it has calculated.

Another possible way for the WSE to profile users could be based on the query length. If the WSE monitored the activity between users and the SE, it would be able to link short search queries to users who sent short cryptograms to the SE. The same would happen for long search queries. But, as it has been explained in Section 5.2, before submitting query data to the SE, the user pads the query until it has an standard length. This way, as all the data sent by the users has the same length, it could not be possible for the WSE to identify users based on the length of the query it has received.

6.2 Against external attackers

The second attacker that has been defined in Section 4.2 is the external attacker. This attacker can be a external agent that monitors all the traffic between the SE, the users and the WSE, or can also be a dishonest user connected to the SE that tries to gather insight about the other connected users.

Let us assume, the external attacker monitors all the traffic that occurs between users, the SE and the WSE in order to profile users. This scenario has already been covered in Section 6.1 supposing the WSE is the attacker who monitors the traffic and has been proven ineffective. If we go a step further and assume that the external attacker wants to tamper the search results to, for instance, guide the user to a malicious website, the message digest check will fail and thus, the SE will resend the search results until the user confirms the legitimacy of the received data. Therefore, it is not possible for the external attacker to tamper the search results either.

In case the external attacker connects to the SE posing himself as a legitimate user to gather data belonging to other users, it will not be able to decrypt other users data. As explained in Sections 5.1 and 5.2 the communications between the users and the SE is secured by a shared session key, which is unique for each session and thus, for each user. Thus, if a computationally secure cryptosystem is used to encrypt symmetrically the traffic between the user and the SE, the external attacker will not be able to access the encrypted traffic, as the attacker has no access to the secret key used by the other users.

7 Conclusions

We have presented a model that preserves users' privacy when performing web search, while serving them personalized search results. In this proposal a mechanism is built, called the secure element, that acts as a server-side intermediary between users and web search engines. It preserves user privacy by using a cryptographic protocol that encrypts communications between users and the secure element. In order to serve personalized search results, the secure element reranks search results based on historic and session query data. The performed privacy analysis shows that it is a viable mechanism as it effectively preserves user privacy.

7.1 Future work

Further improvements to this work can be classified in two main lines:

1. Implementation of the mechanism to measure protocol performance
2. Changes to the protocol in order to enhance users' privacy

In the first place, the implementation of the mechanism would allow to measure the performance of the protocol, by looking to the response time. A set of experimental tests would ensure the viability of the proposed system as a mechanism that has similar response times compared to a WSE-only implementation and effectively protects privacy.

In the second place, several changes can be proposed to the protocol in order to increase users' privacy. Even if the SE stores no data, a rewrite of the protocol that makes the mechanism private by design, that is, a system where not even the SE has direct access to user data, would eliminate further privacy concerns users might have. Another improvement possibility consists in using query classification to launch queries that share the same topic when an adequate number of k users is connected. This approach would prevent the WSE noticing the existence of queries related to a specific topic only when a particular user is connected to the SE.

References

1. Baeza-Yates, R., Maarek, Y.: Usage data in web search: benefits and limitations. In: Scientific and Statistical Database Management. pp. 495–506. Springer (2012)
2. Barbaro, M., Zeller, T., Hansell, S.: A face is exposed for aol searcher no. 4417749. New York Times 9(2008), 8For (2006)
3. Bennett, P.N., White, R.W., Chu, W., Dumais, S.T., Bailey, P., Borisyuk, F., Cui, X.: Modeling the impact of short-and long-term behavior on search personalization. In: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. pp. 185–194. ACM (2012)
4. Bing: Bing privacy statement (2013), `http://www.microsoft.com/privacystatement/en-us/bing/default.aspx`
5. Castellà-Roca, J., Viejo, A., Herrera-Joancomartí, J.: Preserving users privacy in web search engines. Computer Communications 32(13), 1541–1551 (2009)
6. Chor, B., Kushilevitz, E., Goldreich, O., Sudan, M.: Private information retrieval. Journal of the ACM (JACM) 45(6), 965–981 (1998)
7. Dierks, T.: The transport layer security (TLS) protocol version 1.2 (2008)
8. Dingledine, R., Mathewson, N., Syverson, P.: Tor: The second-generation onion router. Tech. rep., DTIC Document (2004)
9. Eickhoff, C., Collins-Thompson, K., Bennett, P.N., Dumais, S.: Personalizing atypical web search sessions. In: Proceedings of the sixth ACM international conference on Web search and data mining. pp. 285–294. ACM (2013)
10. Erola, A., Castellà-Roca, J., Viejo, A., Mateo-Sanz, J.M.: Exploiting social networks to provide privacy in personalized web search. Journal of Systems and Software 84(10), 1734–1745 (2011)
11. Golle, P.: Revisiting the uniqueness of simple demographics in the us population. In: Proceedings of the 5th ACM workshop on Privacy in electronic society. pp. 77–80. ACM (2006)
12. Google: Privacy policy (2012), `http://www.google.com/intl/en/policies/privacy/`
13. Howe, D.C., Nissenbaum, H.: TrackMeNot: Resisting surveillance in web search. Lessons from the Identity Trail: Anonymity, Privacy, and Identity in a Networked Society pp. 417–436 (2009)
14. Jansen, B.J., Spink, A.: How are we searching the World Wide Web? A comparison of nine search engine transaction logs. Information Processing & Management 42(1), 248–263 (2006)

15. Jansen, B.J., Spink, A., Saracevic, T.: Real life, real users, and real needs: a study and analysis of user queries on the web. Information processing & management 36(2), 207–227 (2000)
16. Jeh, G., Widom, J.: Scaling personalized web search. In: Proceedings of the 12th international conference on World Wide Web. pp. 271–279. ACM (2003)
17. Kushilevitz, E., Ostrovsky, R.: Replication is not needed: Single database, computationally-private information retrieval. In: Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium on. pp. 364–373. IEEE (1997)
18. Nagpal, A., Hangal, S., Joyee, R.R., Lam, M.S.: Friends, romans, countrymen: lend me your urls. using social chatter to personalize web search. In: Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work. pp. 461–470. ACM (2012)
19. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: bringing order to the web (1999)
20. Peddinti, S.T., Saxena, N.: On the privacy of web search based on query obfuscation: a case study of TrackMeNot. In: Privacy Enhancing Technologies. pp. 19–37. Springer (2010)
21. Romero-Tris, C., Castellà-Roca, J., Viejo, A.: Multi-party private web search with untrusted partners. In: Security and Privacy in Communication Networks, pp. 261–280. Springer (2012)
22. Saint-Jean, F., Johnson, A., Boneh, D., Feigenbaum, J.: Private web search. In: Proceedings of the 2007 ACM workshop on Privacy in electronic society. pp. 84–90. ACM (2007)
23. Shen, X., Tan, B., Zhai, C.: Ucair: Capturing and exploiting context for personalized search. In: Proceedings of the ACM SIGIR 2005 Workshop on Information Retrieval in Context (IRiX). p. 45 (2005)
24. Spink, A., Jansen, B.J., Wolfram, D., Saracevic, T.: From e-sex to e-commerce: Web search changes. Computer 35(3), 107–109 (2002)
25. Sweeney, L.: k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(05), 557–570 (2002)
26. Teevan, J., Dumais, S.T., Horvitz, E.: Potential for personalization. ACM Transactions on Computer-Human Interaction (TOCHI) 17(1), 4 (2010)
27. Toch, E., Wang, Y., Cranor, L.F.: Personalization and privacy: a survey of privacy risks and remedies in personalization-based systems. User Modeling and User-Adapted Interaction pp. 1–18 (2012)
28. Viejo, A., Castellà-Roca, J.: Using social networks to distort users profiles generated by web search engines. Computer Networks 54(9), 1343–1357 (2010)
29. Xu, Y., Wang, K., Zhang, B., Chen, Z.: Privacy-enhancing personalized web search. In: Proceedings of the 16th international conference on World Wide Web. pp. 591–600. ACM (2007)
30. Yahoo: AdChoices: Learn More About This Ad (2012), http://info.yahoo.com/privacy/us/yahoo/relevantads.html