Hide-n-Seek: An Intent-aware Privacy Protection Plugin for Personalized Web Search

Puxuan Yu
Wuhan University
Wuhan, China 430072
pxyuwhu@gmail.com

Wasi Uddin Ahmad
University of California, Los Angeles
Los Angeles, CA 90095
wasiahmad@ucla.edu

Hongning Wang
University of Virginia
Charlottesville, VA 22904
hw5x@virginia.edu

ABSTRACT

We develop *Hide-n-Seek*, an intent-aware privacy protection plugin for personalized web search. In addition to users' genuine search queries, Hide-n-Seek submits k cover queries and corresponding clicks to an external search engine to disguise a user's search intent grounded and reinforced in a search session by mimicking the true query sequence. The cover queries are synthesized and randomly sampled from a topic hierarchy, where each node represents a coherent search topic estimated by both n-gram and neural language models constructed over crawled web documents. Hide-n-Seek also personalizes the returned search results by re-ranking them based on the genuine user profile developed and maintained on the client side. With a variety of graphical user interfaces, we present the topic-based query obfuscation mechanism to the end users for them to digest how their search privacy is protected.

CCS CONCEPTS

•Security and privacy →Privacy protections; •Information systems →*Query intent; Personalization;*

KEYWORDS

Intent-aware query obfuscation, privacy protection, personalization

ACM Reference format:
Puxuan Yu, Wasi Uddin Ahmad, and Hongning Wang. 2018. Hide-n-Seek: An Intent-aware Privacy Protection Plugin for Personalized Web Search. In *Proceedings of SIGIR '18, July 8–12, 2018, Ann Arbor, MI, USA.*, , 4 pages.
DOI: http://dx.doi.org/10.1145/3209978.3210181

1 INTRODUCTION

Modern search engines, such as Google, Bing and Yahoo, track, analyze, and exploit users' personal information and search behaviors to customize search results in a per-user basis. The obvious purpose is to speculate users' intent to provide them with more relevant and useful content, so as to improve the search effectiveness. Although personalization in search engines provides convenient way for searchers to access information, it also causes increasingly public concern about personal information disclosure. According to [8], almost 75% of search engine users in the U.S. are uncomfortable with their personal information being tracked and used to personalize their future search.

SIGIR '18, July 8–12, 2018, Ann Arbor, MI, USA.
 ISBN 978-1-4503-5657-2/18/07.
DOI: http://dx.doi.org/10.1145/3209978.3210181

While many solutions for privacy protection in personalized web search mainly focus on the *identifiability* aspect of privacy through secured communication and encrypted data storage, another essential aspect of privacy, *linkability*, has been overlooked. Linkability means that an internet service provider could link multiple queries to the same user and learn detailed information about the user's background and behaviors. In other words, users' privacy could be breached even though their true identities remain unrevealed. Negligence on linkability leaves search engine users with little control to avoid their personal information being abused, such as for targeted advertising and digital discrimination.

In this work, we develop a web browser plugin called Hide-n-Seek to protect the linkability aspect of privacy for search engine users. The purpose of our plugin is to countermeasure potential privacy breach in the process of web search accomplished by search log based personalization techniques, which are currently adopted by many search engines. Hide-n-Seek hides a user's true search intents by submitting a set of *sequentially* generated cover queries in addition to the true user query sequence submitted in a search session. The cover queries generated by Hide-n-Seek mimic the gradually developed user intents during a search session to give the adversary minimal scope to identify the true user search intent. To compensate the degenerated search quality caused by the injected noisy search behaviors to the search engine end, the plugin re-ranks the received search results based on a noise-free user profile, developed and maintained on the client side. In a per-user basis, as the client-side profile gets enriched, Hide-n-Seek becomes more effective in personalizing the search results. Moreover, Hide-n-Seek offers multiple visualization interfaces to demonstrate the effectiveness of its privacy protection mechanism.

2 RELATED WORKS

Our developed system falls into the obfuscation-based private web search (OB-PWS) scheme [4]. In OB-PWS, cover queries are generated and sent to the search engine in addition to the users' true queries to prevent any accurate inference of users' search intent. Browser extensions, such as TrackMeNot [7], are developed following the OB-PWS notion to conceal users' general interests. GooPIR [6], similarly to TrackMeNot, constructs cover queries by selecting keywords from a public dictionary. For each real query of a user, GooPIR generates k cover queries, that are submitted together with the real queries. The simultaneous real and cover query submission prevents the adversary from exploiting query timing or meta-data to distinguish the cover queries. However, existing obfuscation-based privacy protection approaches [3] anonymize user queries in an isolated manner without considering the relatedness between consecutive user queries belonging to the same search task. This leaves adversary the potential to identify users' true intent from

their sequentially submitted queries by distinguishing them from isolated cover queries.

To solve this information leaking, Hide-n-Seek formulates a sequence of related cover queries and clicks with respect to a user's gradually developed search intents. Hide-n-Seek achieves both query-level and task-level plausibility in the generated obfuscating queries via controlled statistical sampling and is confirmed by the conducted user study.

3 HIDE-N-SEEK ARCHITECTURE

The operations of Hide-n-Seek plugin rely on two major components: the back-end server with configured database, and the front-end browser plugin. Since we want to take the pressure of intense computation off users' devices, cover queries and personalized rankings are computed by the back-end programs. But the computation is portable to the front-end for fully private computation. The back-end component of Hide-n-Seek utilizes two different approaches to construct the cover queries, and the front-end component provides various visualization interfaces for users to examine information regarding their submitted queries, cover queries, and the distribution of the true and cover query topics. Figure 1 highlights the overview of the system. In the following, we describe each component of Hide-n-Seek in detail.

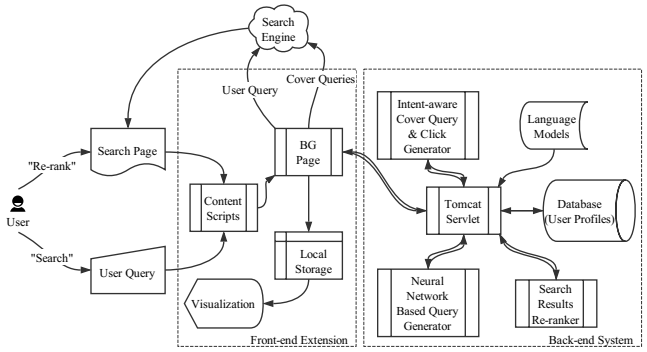


Figure 1: Overview of Hide-n-Seek System.

3.1 Back-end System

The back-end system can be divided into the following parts: an intent-aware cover query and click generator, a neural network based query generator, a search result re-ranker for improving personalized search experience, a secure database for storing historical data of anonymous users, and a Tomcat-based Java servlet integrating the four components together to respond to user requests. Out of the k cover queries generated for each true query, $k-1$ of them are generated by a language model based query generator, while the other one is generated by the neural network based module.

Intent-aware Query and Click Generator. The intent-aware query generator [1] primarily handles two major tasks, (1) identifying search intent of true user queries and (2) generating a sequence of cover queries mimicking a similar intent pattern to disguise users' true intents. To identify search intent, Hide-n-Seek utilizes the topic hierarchy defined in the Open Directory Project (ODP) [9] as the predefined intent tree. All the nodes in the intent tree are associated with a set of documents which are crawled from the web,

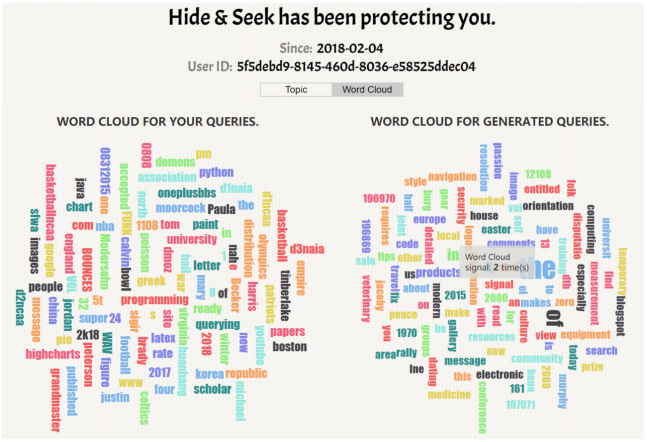


Figure 2: Word-cloud based on submitted true user queries and cover queries

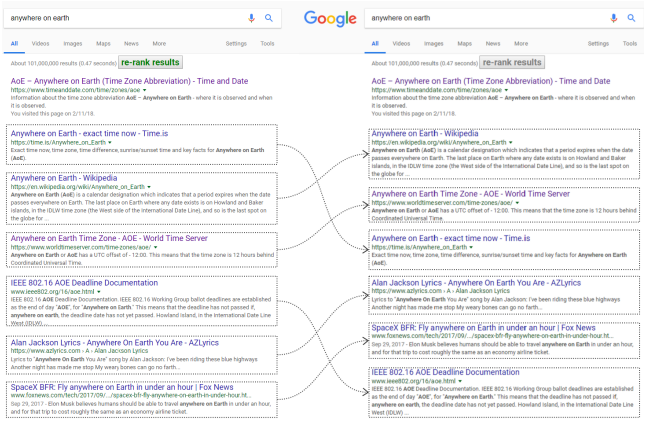


Figure 3: Search results are re-ranked based on client-side user profile.

and an inverted index is created on the documents to accelerate access to such documents. Hide-n-Seek uses an inverted index to perform intent inference on the matched tree nodes. Every node in the intent tree forms n-gram language models based on the web documents categorized under the ODP tree node. After identifying the query intent, Hide-n-Seek samples cover queries based on the n-gram language models that are associated with a selected cover query topic. Hide-n-Seek uses a positional click model trained on a large reference search log, and sample clicks from it accordingly. Figure 2 shows a word-cloud visualization of a user's submitted queries and generated cover queries from our system.

Neural Network Based Query Generator. In addition to language models based technique, Hide-n-Seek also utilizes a neural sequence to sequence approach [10] to generate context-aware cover queries by taking the previously submitted queries in the same search session into account [2]. For the first user query in a search session, the NN-based query generator samples query from a predefined list of queries collected from AOL search log [5], which are grouped under the predefined ODP intent tree.

Search Results Re-ranker. Hide-n-Seek re-ranks the search results based on the the user profile developed and maintained on the client-side to improve users' search experience. Hide-n-Seek utilizes a language model estimated based on the client-side user profile and assigns score to the returned documents as follows,

$$UPScore(d) = \sum_{w \in q \cap d} \log \frac{(1-\lambda)p_{ml}(w|d) + \lambda p(w|C)}{\lambda p(w|C)} \cdot tf(w)$$

To combine the search results ranking provided by the search engine and ranking computed using the client-side user profile, Hide-n-Seek uses Borda's rank aggregation method [11]. Specifically, Borda's method assigns a ranking score to a document as follows.

$$score(d) = \alpha/R_1(d) + (1-\alpha)/R_2(d) \quad (1)$$

where $R_i(d)$ denotes the ranking order of document d in ranker i's ranked list, and α controls the weight of each ranker. Figure 3 shows an example of original and re-ranked search results.

Secured Database. In order to ensure run-time efficiency and to save storage space for users, we adopt a secure relational database in our system's back-end. The storage also makes sure that users would not lose their profiles by software malfunction from the client side. The sole user identifier used in the database is a 32-bit string randomly and uniquely generated for each user at installation. Hide-n-Seek does not require nor store users' sensitive data, e.g., GPS coordinates, IP addresses, MAC addresses, or Google accounts, therefore we or anybody else cannot trace back to users' browsers, devices or themselves from our database.

Tomcat Servlet. The tomcat servlet is the glue for the entire back-end system and is also the bridge between the front-end and the back-end. For different types of requests from users, such as registration, request for cover queries, and request for personalized rankings, the servlet forwards them to different functions to process, and returns results back to the end users.

3.2 Front-end Browser Plugin

Hide-n-Seek is currently available for Google Chrome and tested with Google search engine running on all desktop operating systems. Like all other Chrome extensions, our extension is built on web technologies such as HTML, JavaScript and CSS, and strictly follows the standard development architecture which contains the background page, UI pages and content scripts.

The Background Page. The background page plays an important and versatile role in Hide-n-Seek, which will be explained next.

• **Communication.** Regular web pages can not exchange information with remote servers at will due to the "same origin policy"; and therefore, all communication between front-end and back-end must be made through the background page.

• **Data Manipulation.** Although most of the data is stored on the remote database, many important data, such as user ID and settings, is still stored locally via reliable key/value storage. Other data is stored for visualization purposes, such as the topical distribution of users' historical queries and the last simulation process.

• **Simulation.** The core function of Hide-n-Seek is to send sequential cover queries to search engines to disguise the true queries. After fetching a set of cover queries from the back-end, the background page stores their topics locally and places queries themselves into a keyword queue. In every n seconds, the background page takes a cover query from the queue and launches a simulation with it in an unfocused tab. Simulation includes submitting the query to the search engine and clicking on a document selected by the back-end program on the results page. In an effort to enhance user experience, we ensure our system brings minimal distraction to users by running simulation in an unfocused tab, pre-checking the documents to click on for harmful information or downloads, and closing those tabs as soon as the queue becomes empty.

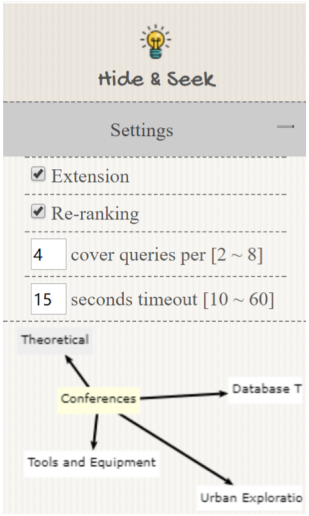


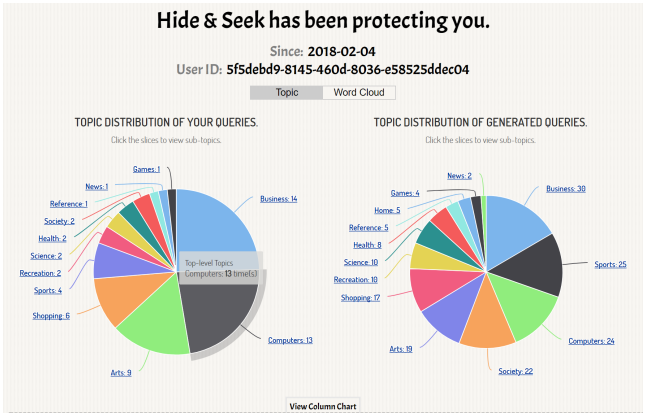
Figure 4: Pop-up Window of Hide-n-Seek.

Content Scripts. JS scripts that automatically insert into the specified pages are called content scripts. In this case, the most commonly used content script is triggered by matching to the Google search pages. Under the premise that the plugin is enabled, the script first extracts the user's search keywords and sends them to the background, and continues to monitor the user's clicks on the page; once a click occurs, the clicked content is sent to the background for user profile update.

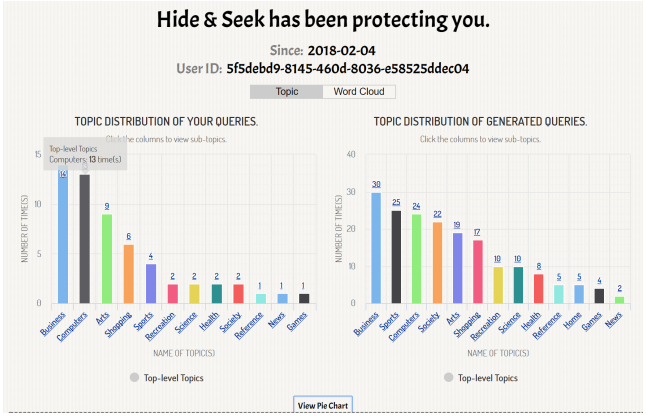
If the "re-ranking" feature is enabled, Hide-n-Seek will send the search result content back to the server for rearrangement via the background page at the moment when the search result page loads. The search result page displays a re-ranking button as soon as the new ranking arrives. The latency of this process is minimized that users can not experience the delay associated with extra messaging even in normal network environments.

UI Pages. Building on the native user interface (pop-up windows and options menu) of Chrome extension, we build another individual page to demonstrate the content and distribution of generated cover queries, together with the users' submitted genuine queries.

• **Pop-up window.** The end user can open the pop-up window by clicking on the small icon at the upper right corner of the browser interface. The settings menu is integrated into a fold-able drawer, allowing users to quickly adjust to their preferences. The provided options include: 1) extension's on-off switch; 2) re-ranking feature's on-off switch; 3) number of cover queries submitted for each simulation; 4) maximum waiting time for each simulation. Below the settings menu, we have created a space that uses the "force directed



(a) Top-level Topics in Pie Charts



(b) Top-level Topics in Histograms

Figure 5: An example of Hide-n-Seek Topic Distribution Visualization.

graph" to illustrate how Hide-n-Seek obfuscated the user's original query topic in the last search activity.

• **Visualization.** As described in [1, 2], our core query generators and document re-ranker show advantages over other baselines in terms of search effectiveness, privacy protection and statistical query plausibility. Before bringing in quantified measurements into the system, we employ user-friendly visualization methods to help users digest the effectiveness of our plugin.

We summarize user and system behaviors from the moment the user started using our plugin in the visualization page, which is accessible via the portal on the pop-up window. From the "word cloud" tab, we use word cloud to compare the difference between user's genuine search queries and system submitted queries. Mouse hovering can show the frequency of occurrence of the individual querying word. Figure 2 shows an example.

In the "topic" tab, we use pie charts and histograms to illustrate the thematic distribution of queries submitted by users and the system, as shown in Figure 5. We add the interactive "drill-down" feature to the interface; therefore, unlike the force directed graph in the pop-up window, the visualization page shows three layers of topics covered in the ODP tree. Figure 5a shows the top-level topics of user and generated queries. Mouse hovering displays frequencies of the submitted query, and clicks take users into the next level of topics. Figure 5b displays the same information but in the form of histograms. As illustrated in Figure 5, even after submitting less than 60 user searches in a short period of time, it is clear that Hide-n-Seek well spreads the topics of cover queries. Based on our user study, Hide-n-Seek is capable of producing sequences of queries coming from more average and significantly diverse topics.

4 CONCLUSION AND FUTURE WORKS

We have developed Hide-n-Seek, a browser plugin that implements an intent-aware privacy protection mechanism for personalized web search. It offers a set of useful visualization tools which enable users to digest their true search intents and system-generated cover intents. As for our future work, we plan to improve Hide-n-Seek in terms of plausible deniability and enable dynamic adjustment of the amount of cover queries to enhance privacy protection. Currently most of computation happens in our back-end for the consideration of scalability and runtime efficiency. This might cause unexpected data breach when the back-end database is compromised. We plan to explore the secure computation techniques to avoid any possible data breaches. We are also working on extending the plugin to other major search engines to provide user a more smooth search experience while using Hide-n-Seek.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their insightful comments. This work was supported in part by National Science Foundation Grant IIS-1553568.

REFERENCES

[1] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2018. Intent-aware Query Obfuscation for Privacy Protection in Personalized Web Search. In *Proceedings of the 41st ACM SIGIR*.

[2] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2018. Multi-Task Learning for Document Ranking and Query Suggestion. *International Conference on Learning Representations* (2018).

[3] Wasi Uddin Ahmad, Md Masudur Rahman, and Hongning Wang. 2016. Topic Model based Privacy Protection in Personalized Web Search. In *Proceedings of the 39th ACM SIGIR*. ACM, 1025–1028.

[4] Ero Balsa, Carmela Troncoso, and Claudia Diaz. 2012. OB-PWS: Obfuscation-based private web search. In *Security and Privacy (SP), 2012 IEEE Symposium on*. IEEE, 491–505.

[5] Michael Barbaro, Tom Zeller, and Saul Hansell. 2006. A face is exposed for AOL searcher no. 4417749. *New York Times* 9, 2008 (2006), 8For.

[6] Josep Domingo-Ferrer, Agusti Solanas, and Jordi Castellà-Roca. 2009. h(k)-Private information retrieval from privacy-uncooperative queryable databases. *Online Information Review* 33, 4 (2009), 720–744.

[7] Daniel C Howe and Helen Nissenbaum. 2009. TrackMeNot: Resisting surveillance in web search. *Lessons from the Identity Trail: Anonymity, Privacy, and Identity in a Networked Society* 23 (2009), 417–436.

[8] Kristin Purcell, Joanna Brenner, and Lee Rainie. 2012. Search engine use 2012. (2012).

[9] Chris Sherman. 2000. Humans Do It Better: Inside the Open Directory Project. *Online* 24, 4 (2000).

[10] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.

[11] H Peyton Young. 1974. An axiomatization of Borda's rule. *Journal of economic theory* 9, 1 (1974), 43–52.