Decentralized Topic-Modeling and Proxy-Routing Based Obfuscation Methods for Web Search Privacy

Yuya Jeremy Ong

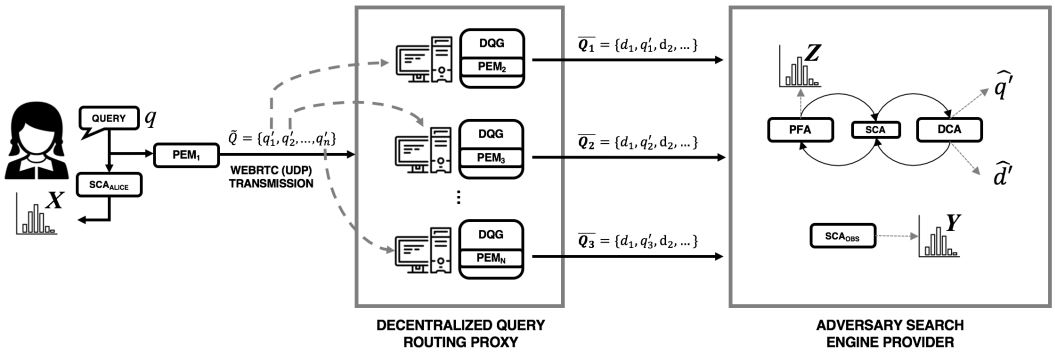
Introduction

Search providers, such as Google, Bing, and Yahoo have offered services which organizes, sorts, and indexes information so that it is readily available for users who utilizes such services. However, to ensure that these services can provide the most optimal experiences to the users, companies have made the effort to profile and tailor query results to ensure that they fit the needs and responses of their users – to ensure that both the users who are utilizing their services obtain the results they seek, while the search engine providers have the continuous traffic and attention of users through the click-through rates achieved by also offering advertisement to generate revenue through their search services. However, many privacy advocates and some individuals have begun to hold suspicions and concerns that such services are a violation of personal privacy. Although these companies are transparent in the sense that they have been using users' information to improve the search quality, their actions are often times seen as privacy invasive – thus, denoting the term ***innocent, but curious***, where the intention is benign, but the methods they utilize causes more harm than benefits to the intended users.

To mitigate such risks of profiling, recent work has been done on methods to avoid such services from learning and profiling users' through various protocols and methods. The primary approaches in this space has been either a cryptographic hash function to encrypt information between the query sender and the services, or through obfuscation methods by injection of dummy terms amongst the real query terms. However, many of the proposed methods often face issues of protection quality and feasibility of the adversary to implement such measures while still retaining a level of anonymity in the methods used to protect against profiling, as according to Balsa et. al [1], who proposed a guiding framework to design secure obfuscation-based private web-search systems.

In this paper, we propose, at a very high level, a novel architecture which makes use of a decentralized topic distribution modeling and search query proxy-routing obfuscation protocol. The system aims to obfuscate user's query from both the profile-based and the query-based analysis operated from the adversarial side of the search provider and aims to complicate the process of recovering, profiling, or learning further about a specific user based on the observed information. This architecture was inspired by a fusion of various different methodologies for protection, and aims to utilize each particular element within the system to fend against potential adversarial action of attempting to gain information on its users. We achieve this through utilizing a very similar Tor-like proxy based approach [2] for routing queries from different source origin devices, a method to obfuscate queries through a network of decentralized topic-modeling interchanges, and a cryptographically-inspired probabilistic method to verify and ensure the correctness and integrity of the responses aggregated through this multi-party security protocol. In the following sections, we describe each of the corresponding components respectively, provide concrete examples of utilization of the architecture, and finally provide an analysis for feasibility of implementation and potential caveats and vulnerabilities our system may encounter, if this were to be implemented practically.

The Architecture



One of the primary areas and assumptions which Balsa et. al makes are the system is self-contained within a single machine environment, where the number of parties involved in the entire process of generating obfuscated materials are concentrated at a fixed user. Although in practice, this certainly enhances scalability and implementation, this however compromises other issues such as cookie-based, browser/device fingerprinting, geolocation, and other meta-data related information which are often associated to help profile and narrow user's information easily. Even if the user were to try to obfuscate their queries through generating dummy information, the search engine is still able to verify the authenticity of the user's profile through mapping other semantics associated with the user and therefore can perform various other methods to easily eliminate or narrow down possible candidate dummy queries amongst the real queries the user sends to the adversary.

This method utilizes a distributed proxy-routing method to impose an additional layer of obfuscation through a multi-party distributed method for routing queries through a decentralized network of computers which are connected through a peer-to-peer UDP based protocol (i.e. WebRTC). Similar to how Tor works by passing encrypted packets between a network of Tor bridges, this architecture works in a very similar manner where we shuffle the location of the source query through another user's computer. This method therefore allows for two key properties which helps attempts to solve the issues of protecting user privacy: i) obfuscation of source makes it difficult for the adversary to trace where the originating query comes from as all context metadata associated with the query would render useless, and ii) the use of distributed queries amongst a group of networks who also utilize the service essentially takes on elements of another person's profile, therefore increasing the overall entropy of the information spread amongst a larger population – thus, making it even probabilistically harder to narrow down the candidate source of where the query originated from.

As the user initializes a query to the tool, the system first randomly establishes contact with other nodes in the network through some randomized rendezvous protocol and establish a cooperation web between the user who is provisioning the query and the nodes which will be used to initialize the query in proxy of the original user. The user who is issuing the query then generates a number of semantically similar, but not precise

terms, based on the requested search terms through the Profile Embedding Model, PEM, a word embedding model which makes use of a compact version of Mikolov et. al's [4] word embedding model that runs locally on the person's machine, thus generating set (i.e. through methods such as k-NN) such that each term in the set contains similar terms based on some given threshold defined by the user measured by the cosine similarity metric (i.e. a value which should be very close to 0, which indicates close semantic relationships). Each PEM is curated through a seed model, which is constantly updated (via online learning) by a combination of active web-crawling through using a conglomeration of previous search terms issued other users and extracting relevant contextual pages to build a much more diverse corpus model.

Every user who participates in this decentralized network has their own respective PEM, which varies slightly from one user to another. In our system, we make use of this property of word embedding vectors to obfuscate the query not only from the adversary, but also from the user which participates within the network and also make use of the geometric variance of the word embedding to our advantage to probabilistically verify the integrity of the dummy query came from the trusted source and not. Given the vector, denoted by generated from the query requester's PEM_1, we send the corresponding vector value to the proxy node which in return will give us back the corresponding word closest to the vector, denoted as PEM_l. We then use the corresponding word (if found within the corpus of PEM_1, and retry until we find a compatible term) as our reference vector for that corresponding proxy node. With the given terms, we can generate a set of both dummy and real queries to be issued to the proxy node through the given reference vector by curating even more semantically similar terms based on the given reference vector and mixing it with the real search query term. By performing this operation, we essentially attempt to dilute the given topics of interest proposed by query issuer (i.e. if the topics were to be fed through an SCA), by disseminating the vector to other profiles in the context of the proxy user to make them indistinguishable. We then perform this operation for each of the nodes who are participating in the network by providing a set for both similar queries and fake queries and have each node execute the query, of the set and aggregate them to the original query issuer. The resulting query will then be cross referenced through a ranking method which utilizes cosine similarity to filter and rank the most relevant results. Note that the resulting page will not provide the final result straight from the original search engine as that would compromise the profiling obfuscation scheme, and instead, provide an alternative page to avoid any further tracking of the user's click through behavior which the adversary would exploit.

Analysis and Evaluation

The proposed algorithm mentioned in the previous section has various key properties which we have outlined. In this section, we explicitly reiterate those key points as well as raise some of the potential caveats, both from a theoretical and practical standpoint on the potential vulnerabilities of this architecture.

Through the use of a proxy-based routing schema of querying, where the query issuer outsources the queries to other computers can theoretically deny entirely their case for submitting a given query from their machine, therefore denying almost with a high certainty that they never issued the query to begin with from the beginning. Furthermore, the additional layer of obfuscation promoted by the proxy layers make the query further indistinguishable as the topic correlations associated with the given context queries (i.e. queries which are both observed from a temporally and topic based view) make it difficult to trace the association of topics and users. While such architecture does enhance query deniability and indistinguishability by providing ensured protection against browser finger printing, query-based profiling, and user-based profiling through its use of a distributed and decentralized topic modeling architecture for balancing the equilibrium of the topic distributions and shuffling of profiles across a wide network of nodes, the system is never absolutely and completely secure against some potential attacks from the adversary, which raises new challenges and open problems for improving this architecture.

First, we made the assumption that the system we have used, including the computer, browser, network, and other key user-client infrastructure is not in any way compromised by the adversary. One potential counter-attack vector which can hamper and make the system vulnerable against our model is to have the adversary plant fake nodes, acting as a pseudo-proxy or man-in-the-middle between the user and the adversary's search engine, records the transactions and augments the queries to reverse the profiling. Furthermore, we would have to be cautious of similar attack patterns and vulnerabilities which has occurred in similar distributed architectures such as those found in the Tor architecture [3]. However, for such counter attacks to be implemented, the adversary would have to take over a significant portion of the decentralized network, which would be considered very expensive and time consuming. A potential method to avoid such attacks would be to devise some peer-based trust protocol to ensure the integrity of the system to be randomized across different regions and have methods to verify the source of each proxy nodes. Another interesting avenue to explore would be to make use of lattice-based cryptography which merges the use of basis vectors of the collection of similar query terms generated by the requester and utilize basis vectors for the generation of public and private keys – these systems have been noted to be very secure as they require significantly hard computationally for both classical and quantum based computing. The application of lattice based cryptography with word embeddings would not only double as a method for obfuscating the text for the adversary, but also for the proxy-nodes which are carrying out the queries.

Another key vulnerability to our system is based on the size of the network which makes use of the platform. For the system to be very effective against sampling attacks by the adversary, we would have to have a significant number of nodes which are willing to work together to further introduce a significant level of entropy in the query term, such that the terms which are utilized are not repeated again making the repeat rate distribution of topics sent to the adversary saturated – ideally we want this to be sparse so that it would be harder for the adversary to reconstruct the individual topic interest profiles hard for the user. Also, another method of concern is building a robust method to pair users through a peer list server which would have to be consistently updated and secured. Design of such distributed system would therefore be a non-trivial task and much thought would have to be put into ensuring the security of the proxy-routing process.

Finally, a practical vulnerability of this system would be the amount of computational resource which is necessary to store and compute these vector operations can be expensive, depending on the dimension of the vectors. The trade-off in this case would be between the size of the topic corpus and the dimensionality of the word embedding vectors which can influence the amount of sparsity and variance of topic that is offered within the system. Furthermore, one of the largest caveat for any distributed system is the amount of network resource redundancy which is the major bottleneck of our system. The user would essentially would compromise much more computational resources on their end to obtain a higher-level of security. However, modern systems today are becoming much more capable of solving much more complex and dense optimization problems with the improvement of libraries (i.e. WASM) and hardware (i.e. improved CPU with neural architectures and GPUs) which can potentially improve and offset the cost of the computational complexity of the system.

Use Cases: Generation of Sensitive-Content Related Queries

Given the nature of this tool, if such system generates or submits queries which are sensitive, this would potentially profile the user as being prone to having a certain disease or condition. This, in effect would cause the system to have some information gain which can potentially influence future queries and augment the entire query of the user. Given this, if the system we built provided the query results in its vanilla form, where we did not provide the alternative interface for the link results, and instead use the original adversary's page, a strategic adversary would exploit the click through rate and conversion rate of which page the user click to verify and further profile the user, while the naïve adversary would not make use of such behavioral actions to further profile the user.

To take it a step further, if the adversarial search service provided an architecture for detecting cookies, other daemon services like advertisement scripts and other third party monitoring services can detect this behavior and profile the user as having interest or no interest in the search. Thus, in light of this discussion, the example of where the user's system generates cancer related queries, if the user did not click or take any action, the strategic adversary would immediately flag the user for not having such relationship with the given topic and with high probability mark the given topic as not associated with the user. Likewise, if the user did click on all the topics pertaining to cancer, then the profiler can easily understand the relevance of the topic based on the click through rate of the links from the search engine, therefore easily profiling the user as having interest in the topic of cancer.

If the system completely ignores any sensitive topics, this in a very high level will reduce the possible corpus space for which the entire obfuscation system can use to further confuse the adversary. By eliminating more topics, we reduce the search space for the adversary to further narrow down the candidate topics, and therefore makes the entire system vulnerable to query based attacks. Hence, it is important to ensure that we have a high variability in the number of topics each PEM holds and ensure that we equally distribute the dummy topics submitted to the search engine.

Conclusions

In this report, we proposed a method to obfuscate query terms through a decentralized and distributed topic-modeling approach through a proxy-based routing methodology. We further evaluated both the strengths and weaknesses of our architecture from various aspects along with introducing new challenges and open areas which can potentially improve the security of our platform. In short, we introduced a novel framework which makes use of a multi-party based distributed computational platform to obfuscate both query and profile based systems.

References

1. Balsa, Ero, Troncosco, Carmela, and Diaz, Claudia. OB-PWS: Obfuscation-Based Private Web Search. In Proceedings of IEEE Symposium of Security and Privacy, 2012.
2. Dingledine, R., Mathewson, N., & Syverson, P. (2004). *Tor: The second-generation onion router*. Naval Research Lab Washington DC.
3. Abbott, T. G., Lai, K. J., Lieberman, M. R., & Price, E. C. (2007, June). Browser-based attacks on Tor. In *International Workshop on Privacy Enhancing Technologies* (pp. 184-199). Springer Berlin Heidelberg.
4. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111-3119).