Demonstration of X-Search performances

Sandrine Juillard

Abstract—**Privacy on internet became a major issue these past years. The rapid enhancement of data analytics has led to a new economic model base on the analyze of user behavior to personalize advertising.[1] X-Search is a proxy that aim to protect users from curious web search engine. The novelty of X-Search protocol [2] compared to other common secure web search protocol is the use of Soft Guard Extention enclaves (SGX). As part of the "Parcours Ingénieur-Docteur", this report aim to implement a prototype base on X-Seach protocol and proceed to further evaluation. In the present case, we will focus on the accuracy of the X-Search and the impact of SGX on the responsiveness of the solution in order to answer the two following question: "In what extend X-Search protocol is accurate " and "Is it reliable to use SGX in such context as Web Search" .**

Codes available here

Index Terms—**security, SGX, web search, privacy**

I. Introduction

APPLICATIONS we use daily, produce and accumulate numerical data. By collecting one's data, and comparing them to other users' data, analytic is able to create an accurate portrait of one's personality and thus, found the most accurate way to reach the user. These methods can be assimilated with manipulation and have been the cause of many scandals [1]. Hence, private Web search has been an active research area in the last decade in order to counterbalance the numerous threats open due to the oversharing of users' search queries to private companies. The purpose of X-search is to protect users from curious web search engine such as Bing, Google or Yahoo, from collecting data though queries. Basically, it provides a proxy that guarantee anonymity to the users and make queries unusable for analytics. The solution proposed aim to overcome the responsiveness / robustness compromise compare to others existing solutions like Tor browser[3]. Evaluations done in the original paper showed how X-search provides a good compromise; while being 3 times faster than Tor in average, it guarantee anonymity and is very robust to Re-identification attacks (section II). However, the protocol does not provide a perfectly accurate response to a query, the output given by the proxy is not the exact same answer as if the query was sent directly to the search engine. Hence the accuracy of the protocol have been measured in the original paper to proved it's reliability. However, several questions remains and it is possible to go further in the analyse of X-search performance. Two questions will be answered in this paper: "In what extend it is accurate" and "Is it reliable to use SGX in such context as Web Search". Hence, the first evaluation will focused on the accuracy of X-search, by evaluating it thought various dataset and the second will focus on the responsiveness. By knowing the time delay taken by each step of the protocol, it is possible to determine what step is the longest and should be optimised. Moreover, it is possible to judge, from the part of time taken by SGX, the reliability of using it in such context where responsiveness is essential.

II. Background

A. Private Web Search

Private web search field is aiming to protect users against curious web search engine. The main treat is not hackers or to find potential security issues in a system, but to make possible for users to enjoy web search services while preventing them from oversharing the information given through their queries. Private Web Search solutions can be classified into three main categories.

Unlikability: In this category , privacy is guaranteed by the anonymity. Thus, origin of queries can't be associated with a user. The main protocol in this category is the Onion Routing protocol. Onion Routing guarantee anonymity by sending the query through multiple nodes. The nodes' path is selecting randomly and hidden using secure encrypted keys. The set of keys form the 'onion'. When the query is received by a node, it removed "one layer" by deciphering the query revived and then forward it to the next node. Despite the protocol protect effectively the anonymity of its users, it suffer from re-identification attacks. Indeed, query themselves can be used to break the unlikability. Re-identification attacks is able to link a profile with it's queries using the high correlations between queries from the same user. In addition, the main inconvenient of this protocol is its slowness. Tor is one of the most popular implementation of this protocol. An other issue of this protocol is that it can be biased by untruthful/malicious nodes. However, protocol such as RAC[4], which is an adaptation of the onion routing protocol, prevent malicious nodes to broke the protocol by grouping them into a virtual ring. A node have to broadcast messages to all nodes that are inside the ring including it's predecessor, thus, if a predecessor does not received a message, it is informed that a node dropped a message.

Indistinguishability: This category of solution aim to spoil user profiles by sending dummy requests to drown real request or by directly altered real request. However, this technique is still vulnerable to re-identification attacks. In fact, the fake queries generated are usually very distinct compared to real one. Thus, the correlation between true queries and fake queries are low. (Example of indistinguihability for geo-localisation devices privacy [5])

Alternative Search Engine: This last categories propose secure web search. Generally, these search engines proposed a fully encrypted protocol which, despite being perfectly effective, is not feasible in terms of performance for an normal user [6].

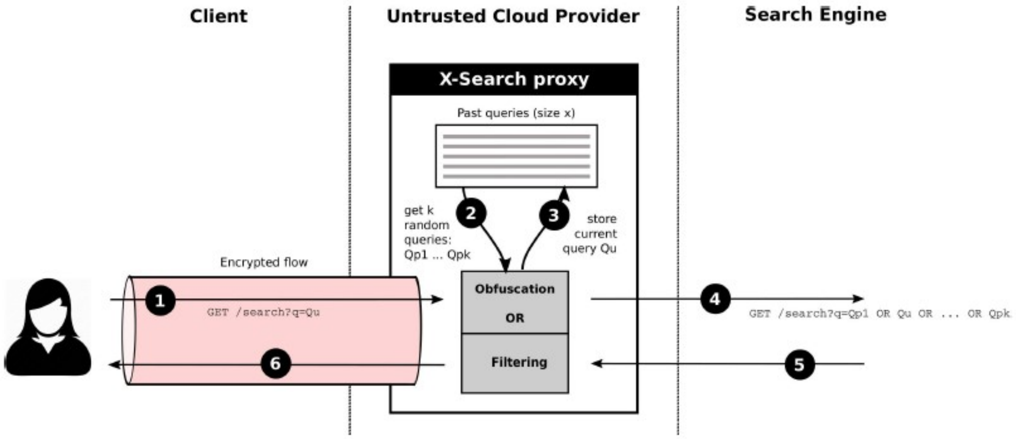


Fig. 1. The X-Search architecture and execution flow [2]

B. SGX Software

Software Guard Extensions (SGX) is a software dev kit developed with the 6generation of Intel processors. It enables the use of a secure execution environment named enclave. An enclave is isolated from the device on witch it's executed by encrypting the totality of the program data. Encryption keys are stored inside the processor itself, thus data can only be read inside the CPU [7]. In this configuration, whatever the level of privilege accessed, data treated inside the enclave can't be reveled or altered. Enclaves are an efficient protection for performing sensitive operation on an untrested platform. In practice, an application can create a trusted execution environment with SGX in order to perform sensitive operations. The enclave initialised contain a set of functions. When a sensitive operation have to be made, the application will call a function inside the enclave. The application and its enclave communicate trough messages named Ecall (Enclave Calls) and Ocall (Outside Calls). Ecalls is a call from the application to the enclave and Ocalls allow the enclave to call of function outside the enclave. Indeed, Ocalls are usually used to perform system call function because an enclaves can't do any operation of that type.[8]

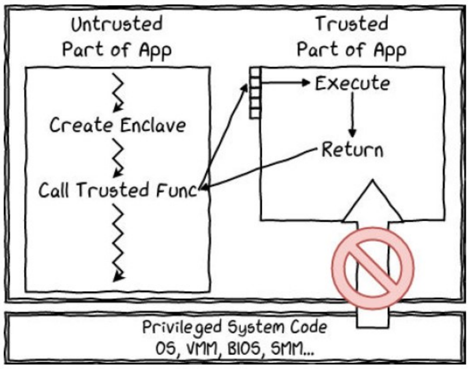
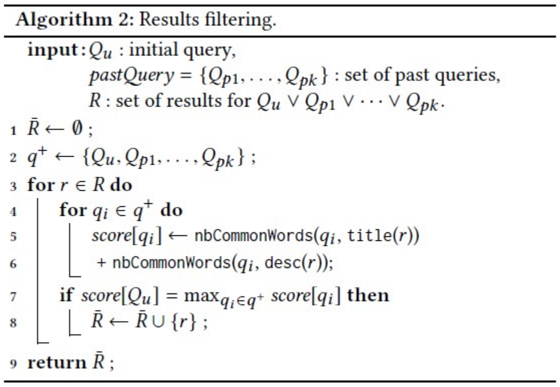


Fig. 2. Overview of an application using Soft Guard Extention [7]



```
Algorithm 2: Results filtering.
input: $Q_u$ : initial query,
       $pastQuery = \{Q_{p1}, \ldots, Q_{pk}\}$ : set of past queries,
       $R$ : set of results for $Q_u \vee Q_{p1} \vee \cdots \vee Q_{pk}$.
$\bar{R} \leftarrow \emptyset$ ;
$q^+ \leftarrow \{Q_u, Q_{p1}, \ldots, Q_{pk}\}$ ;
for $r \in R$ do
    for $q_i \in q^+$ do
        $score[q_i] \leftarrow$ nbCommonWords($q_i$, title($r$))
          + nbCommonWords($q_i$, desc($r$));
    if $score[Q_u] = \max_{q_i \in q^+} score[q_i]$ then
        $\bar{R} \leftarrow \bar{R} \cup \{r\}$ ;
return $\bar{R}$ ;
```

Fig. 3. Pseudo-Algorithm of the filtering Algorithm from the X-Search original paper

C. Challenges in Private Web Search Area

X-search propose a protocol combining unlikability and indistinguishably allowing responsiveness and robustness to re-identification attacks. Indeed, thanks to Soft Guard Extension Enclave, the proxy can be deploy on any cloud as a node that be will be guaranteed unbiased regardless the trustfulness of the platform. In addition it also protect user from re-identification attacks by drawing real request with real users request. In fact, past queries from users can be securely stored inside the enclave providing an ideal dataset for dummy queries. Since both real and dummy queries are from real users, the correlation between users is impossible to determine. Hence, X-Search bring a swift and reliable solution to guaranteed privacy while using web search engine.

III. X-Search overview

The execution flow of X-search is depict figure 1:

$\boxed{1}$ The user send its request to a node hosting X-search proxy using an encrypted flow. A specific broker is in charge of managing the encryption keys to enable a protected communication.

$\boxed{2}$ X-search proxy receive the query. From an history of past queries (witch update each time a request is made), the original queries of the user is combine with K randoms queries into a single one using the ”OR” operator[9]. This is the **obfuscation** treatment.

3 The current user query is added anonymously to the database.

4 The obfuscated query is sent to a web search engine.

5 The result from a web query is a set of links, titles and description.The answer to an obfuscated query is the result from each queries mixed together. Results as selected by the number of words in common between the original query and the titles and descriptions. The score is calculate for each queries that have been obfuscate. The set of answers sent to the user are results that have more common word with the original query than any dummy queries it has been obfuscated with 3.

6 The result obtain (sets of links, titles and descriptions) is forward to the user. In addition, theses results are tampered by the proxy to remove any URL redirection to protect anonymity.

All X-search steps performed by the proxy will take place inside the enclave. An ocall is required to receiver and send the request. The history of past queries have a maximum size, reliant on SGX protected memory capacity of 128 MB, which is equivalent to stored up to 1 million queries. Concerning the algorithms, it is important to mention that the filtering method does not provide constant number of results as outputs. The filter select the answer that fit the original query, but it is possible that the first true result only appeared in the 10th position. Hence, calling for an increased numbers of results to the obfuscated query maximise the probability of having a reliable number of outputs.

IV. Experimental Setup

A simplified implementation of X-search have been developed to evaluate specifically two aspects that will be detailed in section V. Some specificity of X-search have been deliberately ignore; the implementation is minimal to focus on evaluations requirement. The protocol implemented is depict in figure 4. The X-search protocol is deployed on an Azure virtual machine. The algorithms are identical to theses presented in the original paper. The main difference between the original protocol and this one is that the history of queries is static and it haven’t been deployed as a proxy but rather as an app. Details of the implementation is describe Table I. This implementation will be compared with performance of two others variants of the implementation: A strictly identical implementation performed fully outside the enclave (referred as Algorithms implementation). And an implementation that do raw queries to the API (request + parsing).

A. Technical Problems encounter

Unfortunately, Virtual machine of azure enabling Soft guard extension are not available for trial subscription. The VM needed to run SGX are labelled ”Confidential compute” [10]. To get around this inconvenient evaluation will be run on a regular VM using the simulation mode of SGX but the time taken by the enclave will not be measured correctly as wanted. Indeed, the simulation mode works in the same way as the debug mode except the fact that true hardware is not exercised, instead the Intel SGX instructions are simulated in software [11]. Results from the original paper will be used to answer addressed questions, even if the implementation is not strictly identical. In addition, the Azure Bing API is mildly expensive. Hence, with a limited credit, the number of request by experiment had to be limited. (The cheaper subscription being approximately 3$ per 1000 queries, 1000 queries per seconds).

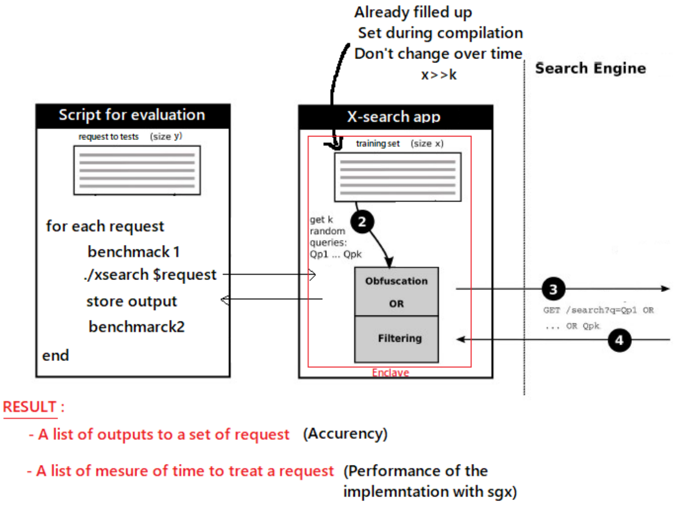


Fig. 4. New implementation’s overview

TABLE I
Summery table of implementation settings

	SGX Original	New implementation
Test dataset	Source: 100 most active users from AOL[12] Size: 2/3 of the database	Source: random queries from AOL Size: 5000 queries
History	Size: 1M queries Training: 2/3 of the database	Size:2500 Static / no training
Web seach engin	bing.com	Azure bing search API
Operating system	Ubuntu Desktop LTS 14.04	Ubuntu Server LTS 18.04
Processor	Intel® Core™ i7-6700	Intel® Xeon® Platinum 8272CL
RAM	8 GiB	3.5 GiB
Enclave	Regular mode	Simulation mode

V. Evaluation

In the original paper, the evaluation aim to determined if the objectives of responsiveness, robustness to re-identification attack are complete and if the accuracy is correct. Inspired by these previous evaluation, this section will present you the two new evaluations designed to analyse further X-Search.

A. Algorithm Accuracy

Measuring the accuracy of responses mean measuring if the responses given by the proxy X-search are equivalent to the responses given directly by the search engine. The accuracy of the proxy is measured with the same metrics used in the original paper which are define below.

$$precision = \frac{|R_{or} \cap R_{xs}|}{|R_{xs}|} \qquad recall = \frac{|R_{or} \cap R_{xs}|}{|R_{or}|}$$

We consider the precision and the recall . Precision will evaluate the proportion of results from X-search that are actually related to the original query. It will be measured by the intersection between R_{or}, result from original query and R_{xs},result from X-search, divided by the number of queries given by X-search (i.e., correctness). The recall will measured the proportion of results from original results that are effectively transmitted to the user (i.e., completeness). R_{os} size i set to 10. The number of results to the obfuscated query and treated by the filtering algorithm is set to 50. No information about these parameters have been found in the original paper.

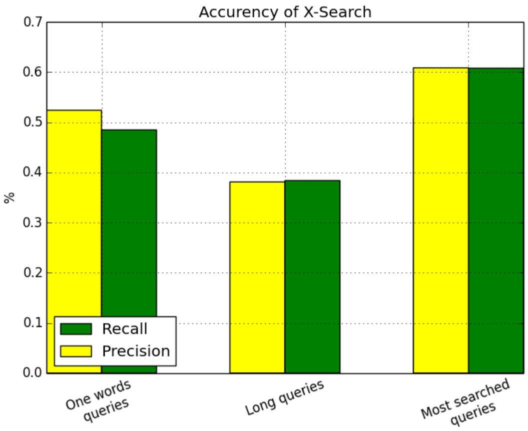


Fig. 5. Accuracy of X-Search by choices of dataset

In the original paper, queries have been selected from the 100 most active user and tested for different values of K. In order to see in what cases X-search protocol is weak and in which it is accurate, we tested different dataset classed by categories and test the precision and recall for each one. Categories are: One word queries, more than one word queries, 1000 most searched queries. Surprisingly, the percentage of recall and precision obtained is very different from the original paper. For K=2 (which mean that the original query is obfuscated with 1 query), the precision and recall are around 50% for One Word queries. These result can be raised by increasing the number of results taken as input of the filtering algorithm and lowering the size of R_{os}. We also observed that Most searched queries have better accuracy while long queries have worst one.

Furthermore, the experiment presented table II put in light weaknesses of the OR operator which can explain such low result. When a query contain a mistake Bing noramlly automatically give results of the correct query and mention ”Showing results for *”corrected query”*”. This mechanism does not works with obfuscated queries. Moreover, it is clear that if one query is very common compare to the others, as google over patti smith in our experiment, it will be way more represented. Hence, in order to works properly, X-search queries have to be equivalently common should not contain mistakes. In addition, short queries is preferred to long queries with several words.

TABLE II
STATISTICS OVER THE 10 FIRST RESULTS FOR DIFFERENT QUERIES

		Results for Patti smith	Results for google
No spelling mistake	*Patti Smith OR google*	30%	70%
One spelling mistake	*Petti Smith OR google*	0	100%
	Patti smith OR goole	90%	10%
Two spelling mistakes	*Petti smith OR goole*	10%	90%

B. System perofrmance with SGX

This evaluation 6 will focus on the responsiveness of the application. This mean measuring the time taken by X-search to answer a query. In the original paper, the time is measured for an end-user. In our implementation, no encrypted communication nor user friendly UX have been developed. We will focus on the delta of time added by SGX and the algorithms compare to a raw request to Bing using the Azure API [13]. Parsing of the responses from the web search engine have been done in both implementation. This evaluation will provide an approximation of the impact of the usage of soft guard extension on the responsiveness by measuring the CDF (Cumulative Distributed Function) of time latency from several variant of implementation. As showed on figure 6, the use of SGX in simulation mode have a negligible impact on responsiveness compare to the algorithms. Indeed, the delay between the two implementation in average Δ_{sgx} is inferior to 0.01s. Hence, approximately 95% of $\Delta_{X-search}$ is caused by the treatment time of the algorithms Δ_{algo} (Filtering/Obfuscate).

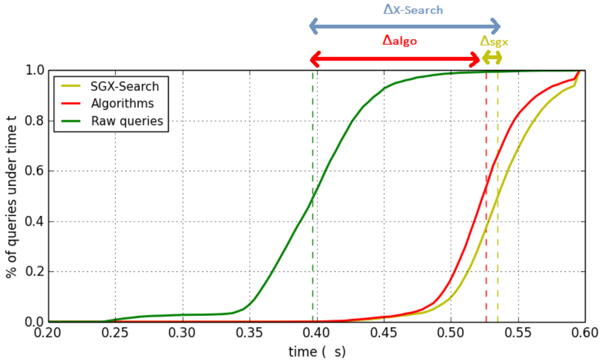


Fig. 6. CFD of the time taken by different of web search’s implementations

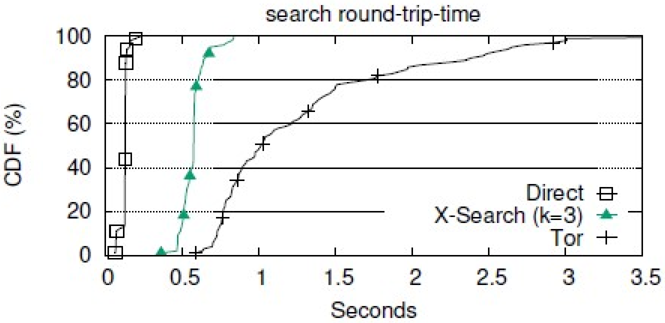


Fig. 7. Evaluation of X-Search responsiveness from the original paper

The result from the original paper showed close result from ours, around 0.5s. Hence, despite the two implementation are not strictly identical, it is possible to interpret that SGX in simulation mode and in regular mode to have close performances. Hence, SGX delay seems to be negligible compare to algorithms treatment delay.

VI. Conclusion

In the light of the experiments, SGX responsiveness performance reliability have been proved in such context as Web search, where speediness is a keen interest. Others SGX application support our conclusion. For instance, SGX have been used to encrypt entries of keyboard and mouse which require a treatment below 200ms [14]. Our experiment does not provide the part taken by the enclave since it was tested in simulation mode, but though the evaluation from the original paper and the knowledge from article related to SGX, we can certainly assure that the time added by SGX is not a problem. However, performance of an application can be reduced by numerous ecall and ocall. In fact, such operation perform context switches and solution to optimised this aspect have already been examined. Among others proposition, [15] suggest an alternative method of communication between the enclave and the untrusted part of the application, using shared memory. Concerning the accuracy of the solution, result are reliable provided that queries are short, common and does not contain spelling mistakes. Indeed, the use of the ”OR” operator provide less significant responses [16]. Despite X-search seems to be impractical, the ideas of using Soft Guard extension for private Web search is interesting since it provide the guarantee that nodes aren’t malicious. The idea have been developed to adapt Tor browser [17] and to enhance crytpocurrency protocols [18]. Hence as said in the original paper, if with time every personal PC became compatible with SGX, such protocol using enclaves to guaranteed trustfulness of nodes could be democratise.

References

[1] S.Juillard, “Why are we scared of big data,” *Enseirb Matemca - EC9LC301 LV1 Anglais*, 2020.

[2] A. Felber M.Pasin R.Pires V.Schiavoni S.Ben Mokhtar, “X-search : Revisitingprivateweb search using intel sgx,” *Association for Computing Machinery*, 2017.

[3] B. E. J. H. E. K. J. S. M. den Toom R.M. van der Laan, “ Thirteen Years of Tor Attacks,” *Computer Science, Delft University of Technology*, 2018.

[4] S. B. M. G. B. A. D. V. Q. A. Shoker, “RAC: a Freerider-resilient, Scalable, Anonymous Communication Protocol,” *Conference on Distributed Computing Systems*, 2013.

[5] M. A. N. B. K. C. C. Palamidessi, “Geo-Indistinguishability: Differential Privacy for Location-Based Systems,” 2013.

[6] C. Gentry, “A fully homomorphic encryption scheme ,” September 2009.

[7] A. Adamski, “Overview of Intel SGX.Quarkslab’s blog,” 2018.

[8] S.Juillard, “Parcours ingénieur-docteur,” *Enseirb Matemca*, 2020.

[9] B. documentation, “Advanced search options,” 2020.

[10] A. Documentation, “Quickstart: Deploy an Azure confidential computing VM in the Azure portal,” 2020.

[11] Intel, “How to run intel® software guard extensions’ simulation mode,” 2016.

[12] C. T. G. Pass, A. Chowdhury, “A Picture of Search,” *he First International Conference on Scalable Information Systems*, 2006.

[13] A. documentation, “ What is the Bing Web Search API,” 2020.

[14] C. H. C. for Information Security, “Fidelius: Protecting user secrets from compromised browsers,” 2019.

[15] T. A. Ofir Weisse, Valeria Bertacco, “Regaining lost cycles with hotcalls: A fast interface for sgx secure enclaves,” 2017.

[16] B. J. J. Caroline M. Eastman, “The Appropriate (and Inappropriate) Use of Query Operators and Their Effect on Web Search Results ,” *ASIS&T*, 2004.

[17] S. Kim, J. Han, J. Ha, T. Kim, and D. Han, “Sgx-tor: A secure and practical tor anonymity network with sgx enclaves,” *IEEE/ACM Transactions on Networking*, vol. 26, no. 5, pp. 2174–2187, 2018.

[18] I. M. K. W. M. Schneider and K. K. E. Z. G. Karame, “Bite: Bitcoin lightweight client privacy using trusted execution,” *28th USENIX Security Symposium*, 2019.