Queries in the Mist: Obfuscation as a Privacy Mechanism

Julian Henry
juhenry@seas.upenn.edu

Advised by Rajeev Alur
alur@seas.upenn.edu

April 29, 2020

1 Introduction

Bygone are the days when the single choice to list ones telephone number afforded practically total privacy. In the modern Internet ecosystem, search engines, websites and Internet Service Providers (ISP) capture a wealth of personal data [3]. Due to Senate Joint Resolution 34, 2017, any ISP is permitted to record and sell consumer browsing activity [17]. The private browsing modes afforded by the four most popular web browsers are anything but private and suffer from security vulnerabilities [42].

Both the individual and commercial entity demand a certain level of digital privacy. At the personal level, more than half of Americans do not approve of corporate mining of their online data [5]. In an business setting, if a corporation queries a database for historical bond yields or regarding a certain patent technology, the database manager may deduce the company's next financial or scientific strategy [13]. Obfuscation-based private web search (OB-PWS) is one method that permits freedom of inquiry online while concealing user preferences.

What is unique about OB-PWS is that it provides a semantic as opposed to a network solution to private browsing via the generation of strategic false searches. The goal of the noisy searches is to prevent an adversary from easily distinguishing a user's actual searches from synthetic ones [35].

The purpose of this study is to review the privacy guarantees of Stephen Smith's "ISP Pollution" software [12]. The significance of this investigation is to verify known limitations of data pollution and confirm the properties of privacy guarantees by noise.

We will explore many of the topics related to preserving privacy when using services providing broadband Internet access service or online search. The next section will provide an high level look at digital privacy. Section 3 reviews known issues of networked solutions to privacy. Section 4 provides a methodology for measuring the privacy guarantees of an OB-PWS. Section 4 explores theoretical grounding for obfusacation mechanisms. Section 5 is a literature review of noise injection software. In Section 6 and 7, we perform a machine learning attack on ISP Pollution [12] and present our findings.

2 Primer on Privacy

In the literature, there is no universal consensus on the formal definition of privacy. Intuitively, individuals may have substantially different tolerances to the disclosure of certain aspects of their digital activity [32].

Internet service providers have the capability to classify users by their activity, interests and preferences. By construction, a web search engine with the capability to track its usage requires storing user data. In turn, this data allows developers to produce models to profile behavior in order to best serve advertiser and user demands [40]. The custom content that users enjoy as a result of this auditing comes at the cost of privacy [2]. Businesses are motivated to spend money on services such as Google's PageRank algorithm to increase their search rankings and advertisement reach to target demographics [47].

Both the search engine and Internet Service provider possess distinct advantages as data collection entities. The search engine is privy to a catalogue of the user's desire for information via web search. At a minimum, an ISP can observe encrypted traffic endpoints and thus knows the user's website activity. The implementation of advanced techniques degrade encrypted privacy further [24].

At present, the major online search providers claim to purge or anonymize their data stores of user queries routinely, but the time frame under which these deletions occur is dubious[30]. Increasingly, one's digital footprint reveals more than is comfortable. As demonstrated by the *Doe vs. Netflix* civil suit, the many streams of digital use are accompanied by serious releases of sensitive information. In the litigation, it was argued that the movie data history exposes highly sensitive interests including "sexuality, mental illness, recovery from alcoholism, and victimization from incest" [46]. In practice, it is not plausible to expect that Internet-based businesses can provide total protection to their customers although in theory, server-side guarantees of privacy would be most robust [32]. Many solutions to protecting privacy from data mining attacks necessitate networked solutions that are prone to failure and insider-exploitation.

We present an abstract model to quantify privacy loss in the context of Internet traffic logging. In this paper, we use the terms obfuscation mechanisms and obfuscator interchangeably. We also consider terms noisy, dummy, false, and fake equivalent when describing to queries and activity. Let there

be a user *Alice* petitioning a series of online activity to a web search engine observed by an honest-but-curious ISP *Eve* [6]. We will call the finite sequence of her real activity emitted by a fixed device an *activity stream*. The *digital profile* generated by the device's activity stream will be denoted as $\mathcal{A}$, a multinomial distribution of elements a_i, each representing the allocation of searches with respect to a semantic category i. By convention, $a_i \in [0, 1]$ is calculated as a representation of the fraction of Alice's searches by topic divided by her total searches [6].

The adversary Eve composes an observed profile $\mathcal{A}'$ from the record of activity on Alice's device. In effect, the observed profile is not a reflection per se of the individual(s) using a device, but oftentimes can serve as a strong enough proxy to raise serious privacy concerns. Because of the increasing prevalence of encrypted internet traffic, the service provider often cannot directly observe the contents of Alice's queries. Nevertheless, the quantity, timing and size of messages, as well as the recipient and sender are available and exploitable [6].

We note that $\mathcal{A}$ has no prescribed topical composition since the topics of interest will vary by use case. For example, a law enforcement agency would seek high granularity of categories related to crime while a clothing companys marketing team would desire categorical information relating to user fashion trends. WordNet [26], for example, proposes forty-four lexical categories. See Cheng et al. for more on heuristics for ontological categorization [9].

Individual digital profiles are in fact an ensemble deriving from a cast of internet-connected devices e.g. laptops, tablet computers, and IoT devices. For a person's k internet devices, their *composite digital profile* $\overline{\mathcal{A}}$ is represented by a weighted average of finite sequences of queries across k activity streams $\{\mathcal{A}_1, ..., \mathcal{A}_k\}$. In Section 3, we provide a theoretical framework for understanding privacy guarantees of obfuscators across $\overline{\mathcal{A}}$.

3 Networked Solutions Obstacles

Crowds [38] was one the first distributed proposals in which nodes pass unencrypted traffic randomly within a network until surfacing onto the Internet. Some peer-to-peer designs that followed such as Tarzan[18] and MorphMix[39] required that all nodes relay and generate throughput with layered encryption. These systems intend to deliver privacy by anonymity of the emitted traffic out of the network. Crowds also spawned a successor,

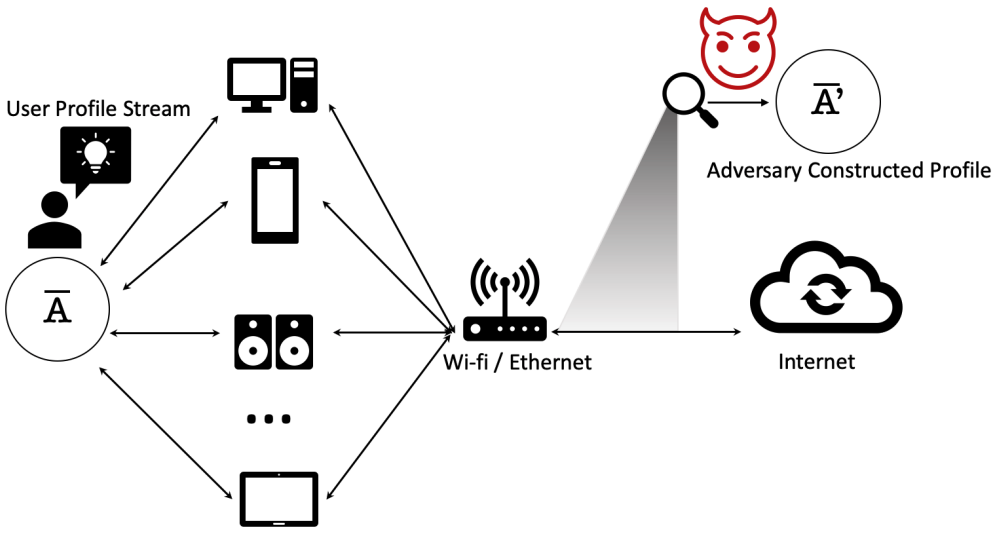

Figure 1: Threat Model

Hordes [15], that incorporated multicast responses as a privacy mechanism. Further distributed work would involve layered public-key encryption known as *onion layering*. Tor[49] is a mature implementation of a onion-layered anonymity.

For those users not participating in distributed anonymity systems, when the Domain Name System (DNS) provided by a broadband internet access provider is used to convert domain names to IP addresses, the end users privacy is compromised [12]. Although the full address of the search is unknown, domain names are enough to reveal significant profile activity. The following domain name visits, for example: “aidsinfo.nih.gov“, “redcrossblood.org“, and “doctor.webmd.com“ could suggest a user's concern about their HIV status.

Many major broadband internet access services record DNS query information for administrative purposes such as tracking users whose activity suggests malware infection [43] . From a privacy perspective, publishing domain name queries alone to an adversary amounts to significant exposure. A Virtual Private Network (VPN) serves as an intermediary between ISP servers and the request destination, partially concealing the final destination from an ISP observer [14]. Another development that aims to increase the

privacy of the internet is the increasing adoption of Hypertext Transfer Protocol Secure (HTTPs) [52]. In our examination of private web search, we first outline a shortlist of current technical obstacles to digital privacy.

3.1 VPN Pitfalls

In a VPN setting, in which a client trusts a service provider to redirect traffic blindly, recording the throughput traffic is a treasure trove of private information. Audits by the VPN host known as "no log" audits are often sponsored by the company themselves. Worse still, even after such an audit, logging can ostensibly resume [20]. There is also no guarantee a government agency will not subpoena the VPN provider to record user data anyway. Many VPNs are poorly configured, and clients often still defer to their ISP DNS server in an event known as a DNS leak [20]. Furthermore, correlation attacks capitalize on the temporal correlations between a web clients inbound request to a server and the resulting outbound request. Leveraging network traffic patterns and query distributions, an adversary snooping VPN activity can execute a correlation attack to infer user queries [8].

3.2 Shortcomings of Tor

At over two million users, Tor is the largest scale privacy enhancement tool online. The anonymity Tor provides, however, requires higher latency than direct internet access. Because a majority of Tor nodes are located in Germany, the Tor infrastructure does not at present support true geographic diversity [25]. In addition to slower Internet speeds, many websites do not offer full functionality to Tor exit nodes. Even more problematic, many Tor exit nodes conduct malicious activity on certain ports. Supporting a Tor node puts the end user at risk to sponsor criminal activity travelling through her IP address [25]. Tor also has systematic flaws that reduce its privacy guarantees. Under certain conditions, an adversary can successfully execute a website fingerprinting attack on Tor [22]. Nasr et al. showed that by commandeering several Tor nodes, an advanced deep learning algorithm could deanonymize 80% of traffic [31].

3.3 Lack of ubiquitous encryption

Unencrypted web requests weaken privacy by exposing the user to malicious scripts or legal corporate actors intending to profile a user's search preferences. Among the top fifty Alexa sites for news, shopping and health, more than 85% do not request fully encrypted browsing by default [52]. As of Q3 2018, nearly 21% of the Alexa Top 100,000 websites did not even use HTTPS [21]. A robustly encrypted Internet would certainly decrease sensitive browsing exposure, but in many cases is not sufficient protection.

An encrypted connection between a web server and client still leaks sensitive metadata. Cai et al [51] demonstrated that these features alone expose the end user to a website fingerprinting attack. Network operators can train models to identify web page requests or otherwise infer the traffic contents [16]. As online encryption continues to become more popular, there will be monetary incentives to extract as much information from users as possible to offset the loss of access to more detailed unencrypted data.

The capability of ISPs to observe client traffic and the potential security leaks of current network privacy solutions suggests that alternative methods or combinations thereof might be optimally suited for providing digital privacy. Data obfuscation is a mechanism that does not have the same latent issues as purely network-based privacy software [28].

4 Queries in the Mist

In a nutshell, data obfuscation is a privacy mechanism that consumes a sequence of real user queries R and produces a sequence of noisy queries $Q : M(R) \in Q$ for some mapping M. When M modifies a request r before submitting it to the web, the data obfuscation algorithm incurs a utility loss. The user's original searches are no longer sent, but instead modified for the sake of hiding their intent.

The drawback of such systems is that the result of searching for $M(r)$ may not contain exactly what the user wants. Conversely, with $M(r) = r$, there is a potential for privacy loss because the user's request is sent "as-is" amidst the noise in Q. In both cases, the privacy guarantees of data obfuscation are weaker in principle than networked solutions. Sensitive user activity or a modified version thereof will still be contained in Q. In conjunction with networked solutions to privacy, the obfuscator can increase a system's privacy

[32].

The upshot of providing privacy by noise injection is that an obfuscator offers privacy at the cost of increased network traffic and energy consumption [32]. Ideally, the generation of noisy activity in parallel to user activity does not produce significant network slowdown [28]. The success of the noise generated depends on how well an adversary receiving the stream of activity can parse out real requests from false ones. Precisely measuring the success of an obfuscation mechanism is a difficult problem, but a differentially private method is a useful perspective.

4.1 Composition of Differentially Private Obfuscators

Differential privacy is a process whereby accurate statistics are extracted from a dataset without infringing on individual privacy. The canonical example examines the privacy of two databases differing by exactly one row. Natural extensions include privacy guarantees for k-size groups or protection across several databases[11]. From machine learning to honest auction bidding, differentially private mechanisms have many applications, one of which is to quantify the security offered by an online activity obfuscating mechanism. This section intends to outline a theoretical groundwork for modeling obfuscation mechanisms. The definitions and theorems are a review of the work of [6] and proofs due to [11]. The differential privacy of an obfuscator is formulated as follows:

Let r and r' be sequences of real activity from the universe R of possible user generated sequences. Let q be a sequence of real and noisy queries from the universe of possible sequences Q an obfuscation mechanism Ω generates. Let S be a subset of sequences q. We fix a pair of finite real activity streams and argue that the output for both is almost equally likely with high probability.

Definition 1 ϵ-LOCAL DIFFERENTIAL PRIVACY. An obfuscation mechanism Ω is ϵ-locally differentially private provided if:

$$\sup_{S \in Q, r, r' \in R} [\ln \frac{\Omega(S|r)}{\Omega(S|r')}] \leq \varepsilon \tag{1}$$

The conventional information theoretic assumptions hold. An obfuscator is said to be *perfectly distinguishable* when $\Omega(S|r) = \Omega(S|r') = 0$, and thus we have $\varepsilon = 0$. Moreover, an obfuscator is *perfectly indistinguishable* when

$(\Omega(S|r) = 0) \oplus (\Omega(S|r') = 0)$, and thus we have $\varepsilon = \infty$. In the former case, the obfuscator produces an identical distribution given any real query stream r and r'. In the latter case, one or more q_i exist such that q_i unequivocally originates in either r or r' since the mechanism maps one sequence to a non-zero distribution that the other sequence maps to zero.

The statement in (1) is also a formulation of the *Max Divergence* [11] of S conditioned on r and r', and has utility as an instrument for hypothesis testing under a user's real searches r. The ratio represents the probability gain ratio of r against r' as the hypothesized argument for the obfuscation mechanism. A suitable property of Max Divergence for differential privacy is that $\varepsilon \geq 0$.

In the case where an adversary has access to many streams emitted by an individual, a natural question is whether a privacy obfuscation mechanism on one activity stream continues to satisfy ε-local differential privacy in spite of the adversary's access to other activity streams. Indeed, a differentially private obfuscator is resilient to post-processing [11].

Theorem 1 POST-PROCESSING. Let $\Omega(r_1)$ be a ε-locally d.p. obfuscator. Let $B(\Omega(r_1), r_2, ..., r_k)$ be an algorithm taking $\Omega(r_1)$ as input, as well as other data.

Proof. Let B be a deterministic function mapping $Q \to Q'$. Fix any sequences r and r', and fix any sequence $\beta \subseteq Q'$. Let $\mathrm{T} = \{q \in Q : B(q) \in \beta\}$. We then have:

$$\begin{aligned} \sup_{\beta \in Q', r, r' \in R} \frac{B(\Omega(\beta|r))}{B(\Omega(\beta|r'))} &= \sup_{\substack{\beta \in B(T) \\ r, r' \in R}} \frac{\Omega(T|r)}{\Omega(T|r')} \\ &\leq e^{\varepsilon} \end{aligned}$$

which is what we wanted.

Such a mathematically stringent requirement for the capabilities of Ω may be from an information theoretic standpoint be intractable. A given activity stream only containing an permutations of an individual's social security number, home address and full name, for example, may be infeasible to mask in practice. Therefore, we provide a relaxation of ε-local differential privacy as follows to allow for low probability edge cases the obfuscator will not handle.

Definition 2 (ϵ, δ)-LOCAL DIFFERENTIAL PRIVACY. An obfuscator is

(ε, δ)-locally differentially private if:

$$\sup_{S \in Q, r, r' \in R} [\ln \frac{\Omega(S|r) - \delta}{\Omega(S|r')}] \leq \varepsilon \tag{2}$$

This inequality is also representative of the δ-Approximate Max Divergence between r and r'. In order to better illustrate the relationship between the approximate Max Divergence and Max Divergence provides insight to its meaning. It will be useful to examine another metric known as the statistical distance of a profile obfuscation mechanism. Intuitively, statistical distance quantifies the greatest difference between the probabilities that the obfuscator can assign to any semantic category for two fixed activity streams.

$$\Delta(\Omega(R), \Omega(R')) := \sup_{S \in Q, r, r' \in R} |\Omega(S|r) - \Omega(S|r')|$$

By convention, let R and R' is said to be δ-*close* if $\Delta(\Omega(R), \Omega(R')) \leq \delta$. In turn, an obfuscation mechanism is (ε, δ)-differentially private for any input streams R and R' if and only if there exists a real input stream T such that T and R are δ-close and $\Omega(T) \leq e^{\varepsilon}\Omega(R')$.

Theorem 2 Obfuscation Repetition. Let $\Omega_1 : \mathrm{R} \to Q_1$ be a ε_1-locally d.p. algorithm, and let $\Omega_2 : \mathrm{R} \to Q_2$ be a ε_2-locally d.p. algorithm. Then their combination, defined to be $\Omega_{1,2} : \mathrm{R} \to Q_1 \times Q_2$ by the mapping: $\Omega_{1,2}(r) = (\Omega_1(r), \Omega_2(r))$ is $(\varepsilon_1 + \varepsilon_2)$-differentially private.

Proof. With $S \in Q_1 \times Q_2$, fix any r,r' $\in R_1 \times R_2$. Then:

$$\begin{aligned} \frac{\Omega_{1,2}(S|(r_1, r_2))}{\Omega_{1,2}(S|(r_1', r_2'))} &= \frac{\Omega_1(S|r_1)}{\Omega_1(S|r_1')} \frac{\Omega_2(S|r_2)}{\Omega_2(S|r_2')} \\ &\leq e^{\varepsilon_1} e^{\varepsilon_2} \\ &= e^{(\varepsilon_1 + \varepsilon_2)} \end{aligned}$$

Corollary 1 Let $\Omega_i : R \to Q_i$ be an $(\varepsilon_i, 0)$-locally d.p. algorithm for $i \in [k]$. Then if $\Omega_{[k]} : R \to \prod_{i=1}^{k} Q_i$ is defined to be $\Omega_{[k]}(x) = (\Omega_1(x), ..., \Omega_k(x))$, then $\Omega_{[k]}$ is $(\sum_{i=1}^{k} \varepsilon_i, 0)$-locally differentially private.

Corollary 2 It follows from Theorem 2 and Corollary 1 that if $\Omega_i : R \to Q_i$ is an $(\varepsilon_i, \delta_i)$-locally d.p. algorithm for $i \in [k]$ then $\Omega_{[k]} : R \to \prod_{i=1}^{k} Q_i$ is $(\sum_{i=1}^{k} \varepsilon_i, \sum_{i=1}^{k} \delta_i)$-locally differentially private.

The increasing value of ε and δ captures the privacy loss incurred from multiple applications of the same obfuscator on the same query stream. From the definition of ε-differential privacy, obfuscators compose in a satisfactory manner, admitting straightforward composition. For example, the composition of three ε-differentially private obfuscation mechanisms is (3ε)-differentially private. There must be caution taken in selecting the proper δ such that the exposure afforded by the relaxation is not too great. For many cases, δ should be worst-case $O(\sqrt{n})$ in order to be less than the sampling noise, where n is the number of elements in an activity stream [11].

One further relaxation of the privacy obfuscation mechanism is motivated by the intractability of protecting extremely dissimilar interest distributions. In Definition 1, any two activity streams must have a bounded indistinguishability. In Definition 2, we allow for a small probability of failure captured by δ. A more nuanced view of the obfuscator's privacy performance takes into account the distance ℓ between any activity streams r and r' covered by the obfuscator. One such measurement tool for ℓ could be the Jaccard Similarity [33]. As an example, the user may wish to hide interest in "army fatigues" or "Halal meats", but not worry about concealing her alma mater or favorite movies.

Definition 3 (ϵ, δ, ℓ)-LOCAL DIFFERENTIAL PRIVACY. An obfuscator is $(\varepsilon, \delta, \ell)$-locally differentially private provided that all pairs of input sequences r and r' such and any subset of S over the range of output sequences Q satisfies the following inequality:

$$\sup_{S \in Q, r, r' \in R} [\ln \frac{\Omega(S|r) - \delta}{\Omega(S|r')}] \leq \varepsilon \cdot \ell(r, r') \tag{3}$$

The parameter ℓ can be viewed as making guarantees for a subset of distributions of real activity issued. The magnitude of ℓ does not indicate the type of obfuscation offered by the mechanism, but it does allow the designer to quantify the privacy guarantee for a subset of activity streams intended to be protected with respect to ℓ. Indeed, such *"distance-based relaxation enables designers to navigate trade-offs between a limited budget of resources for obfuscation (e.g., bandwidth) and privacy"* [6].

As outlined in Figure 1, many digital profiles are composed of activity streams from multiple devices, motivating the quantification of privacy guar-

antees on the behavior of a differentially private obfuscator across *different* activity streams.

Theorem 3 K-Fold Composition In order to guarantee $(\tilde{\varepsilon}, k\delta + \tilde{\delta}, \tilde{\ell})$ cumulative privacy loss over k d.p.-mechanisms given $\tilde{\delta}, \delta \in [0, 1]$ and $\tilde{\varepsilon} > 0$, and

$$\tilde{\varepsilon} \cdot \tilde{\ell} = \sqrt{2k\ln(\tilde{\delta}^{-1})} \cdot \varepsilon \cdot \ell + (e^{\varepsilon \cdot \ell} - 1) \cdot \varepsilon \cdot \ell$$

each obfuscator should be worst-case $(\varepsilon, \delta, \ell)$-differentially private. Now, we are able to reason about the privacy degradation contributed by Alice's k activity streams. Recalling that randomized response with a fair coin is $ln(3) \approx 1$-differentially private, it is fair to examine the special case in which $\varepsilon \in [0, 1)$ and thus, the privacy loss grows in $O(\frac{1}{\sqrt{k}})$, which is a much better degradation than the one given by Theorem 2 [11]. With Theorem 3 in hand, we can better understand privacy requirements of noise injection software on multiple devices owned by one user.

Remark* A tight bound for *k-fold* composition of differentially private mechanisms was given by [23].

5 Data Obfuscation Software

5.1 Query Perturbation

GooPIR[19] is a mechanism for enhancing query privacy by the injection of words into user queries matching the search term frequency. By the addition of several noise words into each search, the system increases the entropy of the user's searches. For example, a search for "Samsung" kicks off a lookup for words having similar popularity of search from a fixed library. If the result of the search indicates that terms "milk" and "linens" have equal popularity, the strings are randomly appended to the original search term, resulting in the submission of "milk Samsung linens" as one search.

From an information theoretic standpoint, the goal of GooPIR is to ensure that for every real query r, $k-1$ noise queries are added such that the entropy $H(R) = \ln(k)$ for a real user search sequence R.

The underlying assumption that weakens the efficacy of GooPIR is that user search terms are uncorrelated. In practice, user behavior is highly correlated and exploitable for statistical regression. The following example {...,"cars governor mint", "flowers bodyshop corn", "carburetor boron

pony",...} illustrates an activity stream evidently related to automobiles where noisy information does not help against temporal correlation reverse engineering. The strategy sacrifices utility for privacy because no untarnished query can be submitted, so the user's original desired content may not be found. Moreover, Internet search frequencies are dynamic, so any frequency library will decay in accuracy. GooPIR is an obfuscation strategy that relies on the false premise that the conditional probability of one search to the next is near-zero. It serves as a ground-zero baseline to compare other query perturbation tools.

Sánchez et al. [41] quantify a query's information content and calculate an equivalent version by increasing the abstraction of certain terms. Concretely, the search "apples basketball" might be transformed into "fruit sports" via association in established semantic hierarchy libraries such as OpenNLP Maxent or WordNet. The mechanism is given by three steps: (i) syntax analysis over on the unfiltered query in order to break each noun into its semantic unit (ii) in order to capture the most information-rich elements of the query, the information content of each semantic unit is calculated and evaluated vis-à-vis other semantic units (iii) the final step deploys a linguistic hierarchy database to generate semantically similar queries modulo a configured privacy parameter. Namely, the greater semantic distance between the original query and the new query, the greater privacy achieved.

The natural question that Sánchez aims to answer is to what extent the ontological reduction modulo the privacy parameter diminishes utility of search. In order to measure the success of a transformed search, Sánchez et. al calculate the information content as the fraction of web hits of the original divided by the magnitude of all web hits possible. The utility of the search mechanism, then, is derived from the change in the information content between the original and distorted queries. They find their tool has high utility, but is slow in practice.

PrivacySearch [40] requires that ontology categorization happens on-the-fly. This OB-PWS is a query perturbation mechanism that generates privacy through an ontological mechanism. Namely, an algorithm known as *PrivacySearch* employs WordNet as a hierarchical natural language categorizer [26] to reduce words into their ontological root. Next, all permutations of the ontological roots are computed and quantified with respect to the original query by measuring the mutual amount of existing hyperonyms. The reduced query in the permutation with the least distance to the original is selected and submitted. Ginés underscores that ontological query perturbation

is often too slow to be practical.

Transforming a real query into a semantically similar, but reduced query is often computationally costly. Moreover, it is hard to guarantee that the product of such computation will yield the desired search results online. Most crucially, pure semantic reduction cannot offer sufficient privacy and utility simultaneously [40].

Distortion Search [29] is similar to TrackMeNot, but implements a different heuristic for producing false queries. Mivule presents a a universe of five elemental query types: that are used to systematically obfuscate the real query.

The steps are as follows: (i) a query is distilled into its ontological root words (ii) if any verbs appear in the query, they are piped into further processing (iii) verbs that are one or two degrees separated from the root verbs are generated (iv) Highly visible dummy keywords that would return highly relevant search results are identified (v) The original query and resultant dummy keywords are categorized into one of five classes: informational, transactional, natural language, temporal and navigational. (vi) Permutations of the query types are generated from steps (i-iv) (vii) A batch of searches is executed based on concatenating of subset of the permutations in (vi).

Discretionary click-through was performed for a subsample of the search queries with the goal of simulating a user's single interest in "purchasing a Toyota" [29]. As a result, the noisy queries generated significant change in targeted ads within a week's time with 93% of advertisements no longer related to the simulated user profile. The classification universe was intended to maintain a certain level of relevance with respect to the original search. Distortion Search also showed an improvement on TrackMeNot, but a more systematic look is required since the machine learning learning was trained on less than five hundred of the over 600,000 query rows [4].

5.2 Query Pollution

Noise Injection for Search Privacy Protection[53]

Ye et al. provide an information-theoretic lower bound for the expected number of false queries to achieve plausible deniability from a search engine adversary. The paper proposes minimizing the mutual information, defined as the intersection of the entropy of an activity stream and the entropy of the same activity stream injected with noise queries. Selecting the right kind of noise is hard, but given the right kind of noise, a server side observer

cannot distinguish between real and false queries. The framework of mutual privacy allows for total plausible deniability given a search mechanism emits a sufficient number of false activity.

Formally, if the probability that a real activity $|Q_u|$ is emitted by an obfusactor is ε, the expected noise queries N_q required for perfect protection is given by the tight bound:

$$E(|Q_n|) = \frac{1-\varepsilon}{\varepsilon}|Q_u| \leq (N_q - 1)|Q_n|$$

Given enough noise, it is theoretically possible to offer high degree of privacy. Another result of Ye's paper is that fewer noise calls are necessary if the noise depends on the distribution of the user's searches, but no implementation is given for such a solution as it would require a method for locally categorizing user activity.

TrackMeNot [50] is a web browser extension that uses a customizable RSS feed to generate dummy queries based on trending keywords. The queries are generated by sampling tokens from article titles listed in the RSS feed. In addition to the option to adjust the RSS feed, the software provides users the options to adjust the frequency at which the dummy queries are submitted, including a "Burst Mode" that produces a high volume of activity intermittently. Nevertheless, the semantics of these false queries are susceptible to natural language processing attacks.

If the RSS feed is known beforehand, TrackMeNot offers significantly reduced privacy [6]. Richard Chow et al. showed that the timing of the dummy query generation also increased an attacker's ability to separate real from fake queries [10]. Petit et al. [34] leveraged a similarity metric to the Urls generated by RSS seed links to classify TMN noisy Urls with greater accuracy than pure machine learning techniques.

Smith's ISP Pollution ISP Pollution is based on an information theoretic argument that with sufficient quantity of noise, users will enjoy increased privacy from an ISP observer. The process begins by downloading words sampled from a random word bank and initializing a set of biasing links rich in links to popular sites. As a precautionary measure, the algorithm downloads

the Shalla Black list [44] to filter which sights can be viewed.

Algorithm 1: ISP Data Pollution

```
input: B: set of blacklisted domains
u ← randomQuerySeed() ;
u ← biasLinks();
while true do
    if U(0,1) < 0.005 then
        resetuseragent();
    if u.linkcount < 2000 then
        u ← u ∪ randomQuerySeed();
        u ← u ∪ biasLinks();
    url = drawRandom(u);
    if url ∉ B then
        goTo(url);
        u ← u ∪ randomSampleLinksFrom(url);
    sleep(χ²(1/2, 1/5) seconds);
```

First, the biasing links are added to a cache of seed Urls. While the number of seed Urls is less than two thousand, the word bank is used to generate seed queries. The queries are fed into one of four search engines: Yahoo!, Google, Bing and DuckDuckGo with a uniform probability. A sample of the outgoing links returned by the query are then added to the seed Urls. The pollution continues at every step popping off a link randomly from the cache until the cache exceeds two thousand. So long as the cache size exceeds two thousand, the search engines are not queried by the word bank again. For every Url, the crawler scrapes a subsample of all links returned by upon visiting and adds the links to the ongoing cache. The algorithm is given by ISP Pollution

Because this mechanism is the subject of our experiments, we present a deep dive into other details involved in spoofing the polluted activity user agent. As an example, the following header is used to spoof Safari use on an iPad. "`The 'Mozilla/5.0 (iPad; CPU OS 6_1 like Mac OS X) AppleWebKit/536.26 (KHTML, like Gecko) Version/6.0 Mobile/10B141 Safari/8536.25`"

Other features such as DoNotTrack (DNT) requests, operating system, personal computer and touch cabale parameters modulate the polluted activity given in Table 1. Smith suggests that advanced users tailor the parameters

according to household usage rates. Logging a Wi-Fi router to collect such statistics is outside the scope of this paper.

DNT	True: 0.8
	False: 0.2
OS	'Mac *OS': 3, 'iOS': 6,
	'Linux': 1, 'Windows': 1,
	'noneoftheabove': 1
is_pc	True: 4
	False: 6
is_touch_capable	True: 6
	False: 4

Table 1: browser params

By design, this mechanism will fail to protect a user actively trying to engage with online material contained in the following blacklisted categories: *downloads* (filesharing, p2p, wallpapers and torrent), *drugs*, *hacking*, *gamble*, *porn*, *spyware*, *updatesites* (utility download hosting for vendors), *urlshortener*, *violence*, *warez* (cracked software) and *weapons* [44].

A summarizing table for Section 5 and 6 is found in Appendix A. In the next section, we examine the capacity of an adversary to correctly parse the false output of the ISP Pollution software embedded in a user's activity stream.

6 Methodology

Many data obfuscation mechanisms privacy guarantees for protecting user queries or online activity. ISP Pollution is a mechanism that by virtue of its query mechanism in effect makes privacy claims for both categories. Since seed links are generated through Google and co., user queries in the activity stream visiting major search engines produced by this mechanism are either a product of genuine user activity or randomized word search. We examine the theoretical result of Ye et al. through the lens of a Logistic Regression with L2 normalization on queries generated by AOL users and queries generated by ISP Pollution. Modulating the amount of noise with respect to the user queries, F-score and accuracy are compared.

In the second experiment, we train a Logistic Regression with L2 normalization on the domain name activities of AOL users and the ISP Pollution generated websites. We extract features from the text of each domain using a word segmentation algorithm as well as an n-gram set [7].

6.1 Data Sets

The AOL search keywords dataset released in 2006 is often used in benchmarking tests of obfuscator efficacy to simulate real user searches [4]. The release of this data became the subject of a class action lawsuit as a result of the privacy exposure caused by only anonymizing the users identity [27]. Although contemporary internet usage has notable differences, many search behaviors are still the same as they were in 2006. Another argument for its utility is that sensitive searches relating to taboo content are presented unfiltered e.g. "nude celebrity images″, "lexapro not working after 2 weeks″ & "how to kill and not get court″ . In its entirety, the dataset contains nearly twenty million entries for over 650,000 users collected from March 1 to May 30 in 2006. The columns are as follows:

1. AnonID - anonymous user identifier
2. Query - search terms entered by the user with some special characters removed
3. QueryTime - a timestamp for the query submission
4. ItemRank - the rank of the search result clicked, if clicked
5. ClickURL - the forwarding URL domain, if clicked

Although temporal correlation attacks on query identification may work in principle, [32] finds no significant utility in the temporal information of the AOL data set, so we drop the QueryTime column. We do not at present have the ItemRank information for the ISP Pollution data, so we drop it as well.

Some users in the AOL dataset have too few searches to properly analyze. In this special case, we consider these users highly amenable to make use of obfuscation techniques. Users with fewer than twenty searches have been ommitted for analysis, and the remaining users divided into quintiles partitioned on users possessing queries in the quintiles $\{29, 45, 73, 142\}$ from

here on known as AOLQ1, AOLQ2, AOLQ3, AOLQ4 and AOLQ5. Our view of the *cold-start problem* will partition user activity based on a chronological threshold [47]. We foist each user into the following scenario with respect to ISP Pollution deployment: after two months on the internet, the user kicks off ISP Pollution. Therefore, we partition the training and test data with respect to the start of May 1, 2006. Furthermore, we did not consider users with fewer than five entries in either the test or training periods.

The ISP Pollution dataset is composed of two columns. The first column contains urls accessed, and the second are timestamps logged by ISP Pollution v2.0.1 over the course of a week. See Figure 2 for a sample of the hourly throughput incurred by running the program.

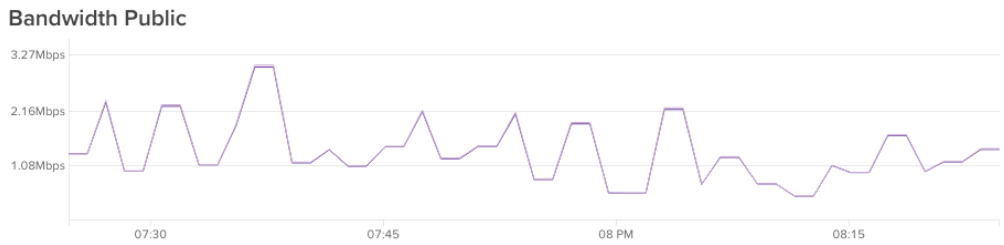

Figure 2: Sample Hourly Throughput

6.2 Adversaries and Threat Models

We consider two adversaries in our experiments: the search engine host and internet service provider.

The *search engine adversary* has access to search queries and ClickURL but not activity after the outgoing link. We note that DuckDuckGo[1] in principle does not track user queries, but for the purpose of analysis, we assume all click through activity originates from a non-private search engine.

ISP Pollution does not perform many searches relative to its url requests. In the AOL dataset, 38.30% of queries involved click-through as opposed to 13.43% in the ISP Pollution data set.

In order to retrofit the AOL data to the ISP Pollution data, we now assume that the search engine distribution in the AOL data equals that of the ISP Pollution. Therefore, every URL search will be uniformly transformed from an AOL search to a Yahoo!, Google, Bing or DuckDuckGo search.

The *ISP adversary* cannot make a distinction beforehand between client-side HTTP requests and real user activity. Although the ISP adversary does not have direct access to web page contents, it keeps track of the domain names accessed. Comparatively, the ISP Pollution algorithm has a higher output of direct link visits than the AOL data set. Since the ISP adversary knows in principle just the domain name visited by user, the query-only activity from the AOL data set is at a baseline interpreted as *aol.com*. Therefore, we consider only the domain names from AOL ClickUrl column.

In both scenarios, we trained a binary classifier on equal partition of AOL and ISP Pollution data for 500 AOL users. Next, we tested the classifier on varying ratios of ISP Pollution noise relative to the amount of user data, given by N1, N2, N5, N10, N20.

7 Results

Our results indicate that privacy diminishes almost monotonically on increasing quintiles. Accuracy naturally increases as the noise ratio increases as more and more activity is correctly identified as false relative to the real activity. Initial findings determined that the out-of-the-box RandomTreeClassifier and SupportVectorMachine were suboptimal at this classification problem.

For the search engine adversary, we demonstrate that the noise added by ISP Pollution does not degrade a classifier's ability to identify user queries. We encounter a powerful structural flaw in the ISP Pollution algorithm evidenced by these results. The twenty-to-one noise to real ratio suggested is not sufficient enough to lower the F-score of a search engine adversary. In the literature, a fundamental weakness of TrackMeNot arises if the adversary knows the user's RSS feed that generates the dummy searches. By the same token, if the wordbank generating the searches is known, the efficacy of the search noise drops even further. To optimize the privacy utility, we recommend switching the default search engines to a customized blend of searches via DuckDuckGo [1], Searx [45] and Qwant [37] instead.

For the ISP adversary, we confirm the theoretical work in [53] empirically by showing that increasing noise of user queries decreased the fraction identified in the polluted activity stream. Moreover, we affirm the findings in [7] suggesting that the English word substrings of a Url have strong predictive power. Overall, n-gram Url features were best at distinguishing real from

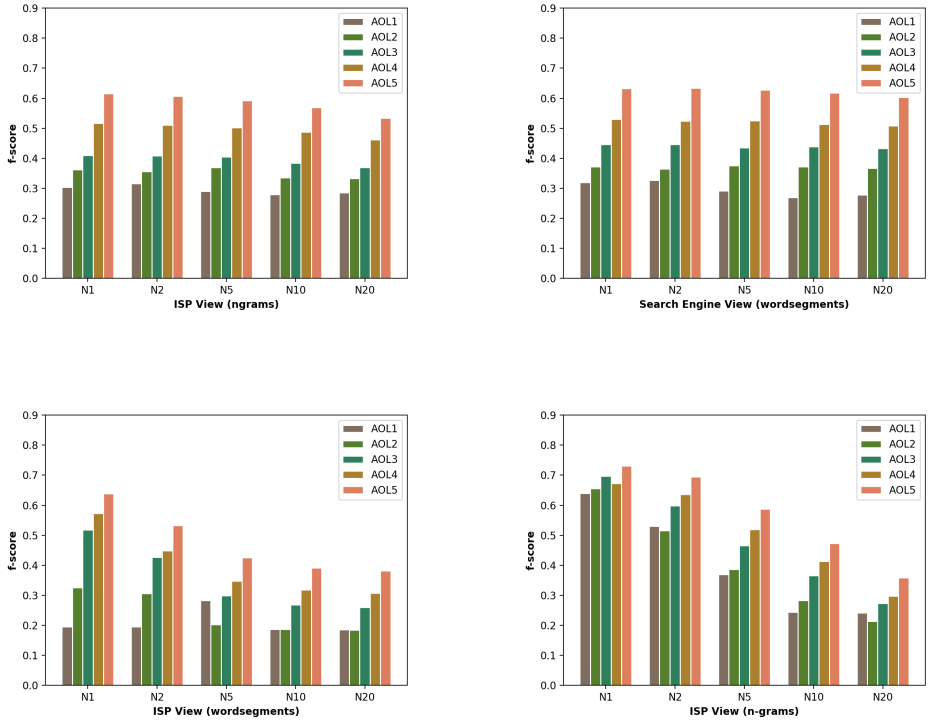

Figure 3: Adversary F-Score Against Noise

false activity, but in the N20 category performed slightly worse than word segmentation for AOL5.

8 Discussion

This paper examines a recent open-source solution to privacy protection via data pollution and verifies known limitations of naively constructed network noise to defend against an honest-but-curious adversary. In addition, we summarize theoretical frameworks for differentially private obfuscation.

Without proper regular expressions, Smith's data polluting agent has the potential to engage with malicious content. While aggregating the dataset for ISP Pollution, average cumulative size of files downloaded per hour was 70MB including pdf, executable, compressed and spreadsheet formats. It is

not assumed that the ISP crawler will avoid downloading potentially malicious files, yet by avoiding downloading these files, the user behavior on downloads will not be properly masked. More sophisticated attacks may better identify real queries with exogenous data about websites or queries. Temporal correlation attacks also have the potential to better filter dummy activity.

Data obfuscation as a privacy preserving mechanism hides user activity in plain sight. Therefore, by preventing the web crawler from accessing illicit or illegal websites, this obfuscation software can make no guarantees about protecting certain types of sensitive online activity. Nevertheless, the underlying principle given by Ye et al. that from an information-theoretic standpoint, sufficient noise can offer plausible deniability. The bottom line is that the blacklist underscore a shortcoming of the semantic approach to digital privacy: all activity, fake or otherwise, will be published to the adversary's view.

Data obfuscation also faces adoption challenges in the cyber-security technology sphere. By design, privacy guaranteed by polluting networks is an antagonistic method when compared to networked privacy solutions such as Crowds, Tor and virtual proxy networks.

Future work in quantifying differentially private obfuscation mechanisms on empirical semantic categories is needed to evaluate the practicality of protecting users in the wild by testing known mechanisms and tuning relaxation hyperparameters. Systems such as Privacy, Efficient, and Accurate Web Search (PEAS)[35] have already suggested the potential of obfuscation mechanisms as cyber-security tool. The strategic transmission of digital noise has promising functionality as a privacy-enhancing primitive for the stealth community.

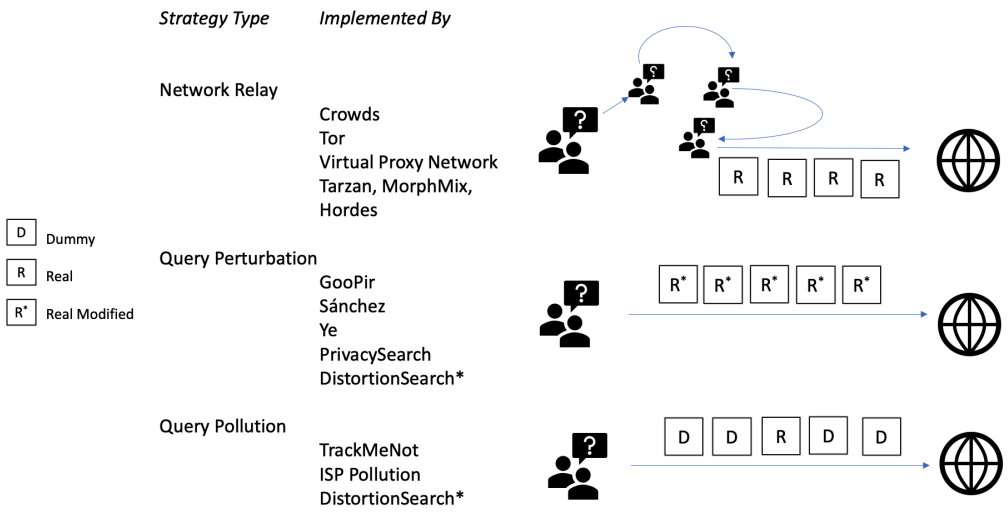

Figure 4: Appendix A: Privacy Mechanism Shortlist

References

[1] "About DuckDuckGo." DuckDuckGo, https://duckduckgo.com/about. Accessed 29 Apr. 2020.

[2] Aktolga, Elif et al. Building Rich User Search Queries Profiles. UMAP (2013).

[3] Ali, Muhammad, et al. "Ad Delivery Algorithms: The Hidden Arbiters of Political Messaging." ArXiv:1912.04255 [Cs], Dec. 2019. arXiv.org, http://arxiv.org/abs/1912.04255.

[4] AOL (2006), "AOL keyword searches" (online), available at: http://dontdelete.com/default.asp. Accessed April 29 2020.

[5] Auxier, Brooke, Lee Rainie, Monica Anderson, Andrew Perrin, Madhu Kumar and Erica Turner. "Key Takeaways on Americans Views about Privacy, Surveillance and Data-Sharing." Pew Research Center, https://www.pewresearch.org/fact-tank/2019/11/15/key-takeaways-on-americans-views-about-privacy-surveillance-and-data-sharing/. Accessed 29 Apr. 2020.

[6] Balsa, Ero. "Chaff-based profile obfuscation," PhD thesis, KU Leuven, C. Diaz, and B. Preneel (promoters), 2019.

[7] Berardi, Giacomo Esuli, Andrea Fagni, Tiziano Sebastiani, Fabrizio. (2015). Classifying websites by industry sector: a study in feature design. 1053-1059. 10.1145/2695664.2695722.

[8] Cai, Xiang, et al. "A Systematic Approach to Developing and Evaluating Website Fingerprinting Defenses." Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security - CCS 14, ACM Press, 2014, pp. 22738. DOI.org (Crossref), doi:10.1145/2660267.2660362

[9] Cheng, Ching Kang et al. Ontology-Based Semantic Classification of Unstructured Documents. Adaptive Multimedia Retrieval (2003).

[10] Chow, Richard & Golle, Philippe. (2009). Faking contextual data for fun, profit, and privacy. Proceedings of the ACM Conference on Computer and Communications Security. 105-108. 10.1145/1655188.1655204.

[11] Cynthia Dwork and Aaron Roth (2014), "The Algorithmic Foundations of Differential Privacy", Foundations and Trends in Theoretical Computer Science: Vol. 9: No. 34, pp 211-407.

[12] "Essandess/Isp-Data-Pollution." GitHub, https://github.com/essandess/isp-data-pollution. Accessed 29 Apr. 2020.

[13] Domingo-Ferrer, Josep et al. "h(k)-private Information Retrieval from Privacy-uncooperative Queryable Databases." Online Information Review 33 (2009): 720-744.

[14] Erdoan, Ayhan & Yzb, Dz. (2008). Virtual Private Networks (VPNs): A Survey.

[15] Levine, Brian Neil, and Clay Shields. "Hordes: A Multicast Based Protocol for Anonymity1." Journal of Computer Security, vol. 10, no. 3, July 2002, pp. 21340. DOI.org (Crossref), doi:10.3233/JCS-2002-10302.

[16] Feghhi, Saman, and Douglas J. Leith. "A Web Traffic Analysis Attack Using Only Timing Information." ArXiv:1410.2087 [Cs], July 2016. arXiv.org, http://arxiv.org/abs/1410.2087.

[17] Flake, Jeff. S.J.Res.34 - 115th Congress (2017-2018): A Joint Resolution Providing for Congressional Disapproval under Chapter 8 of Title 5, United States Code, of the Rule Submitted by the Federal Communications Commission Relating to "Protecting the Privacy of Customers of Broadband and Other Telecommunications Services". 3 Apr. 2017, https://www.congress.gov/bill/115th-congress/senate-joint-resolution/34.

[18] Freedman, Michael & Sit, Emil & Cates, Josh & Morris, Robert. (2002). Introducing Tarzan, a Peer-to-Peer Anonymizing Network Layer. Peer-to-Peer Systems, Lecture Notes in Computer Science. 2429. 10.1007/3-540-45748-8_12.

[19] GooPIR. https://unescoprivacychair.urv.cat/goopir.php. Accessed 29 Apr. 2020.

[20] Imana, Basileal, Aleksandra Korolova, and John Heidemann. "Enumerating privacy leaks in DNS data collected above the recursive." NDSS: DNS Privacy Workshop. 2018.

[21] Internet Security Report - Q3 2018 — WatchGuard Technologies. https://www.watchguard.com/wgrd-resource-center/security-report-q3-2018. Accessed 29 Apr. 2020.

[22] Juarez, Marc, et al. "A critical evaluation of website fingerprinting attacks." Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. 2014.

[23] Kairouz, Peter, Sewoong Oh, and Pramod Viswanath. "The composition theorem for differential privacy." IEEE Transactions on Information Theory 63.6 (2017): 4037-4049.

[24] Li, Feng & Chung, Jae & Claypool, Mark. (2018). Silhouette: Identifying YouTube Video Flows from Encrypted Traffic. 19-24. 10.1145/3210445.3210448.

[25] Mccoy, Damon & Bauer, Kevin & Grunwald, Dirk & Kohno, Tadayoshi & Sicker, Douglas. (2008). Shining Light in Dark Places: Understanding the Tor Network. 63-76. 10.1007/978-3-540-70630-4_5.

[26] Miller, George A., et al. "Introduction to WordNet: An On-Line Lexical Database *." International Journal of Lexicography, vol. 3, no. 4, 1990, pp. 23544. DOI.org (Crossref), doi:10.1093/ijl/3.4.235.

[27] Mills, Elinor (September 25, 2006). "AOL sued over Web search data release". CNET. Accessed April 29, 2020.

[28] Mivule, Kato. "Utilizing noise addition for data privacy, an overview." arXiv preprint arXiv:1309.3958 (2013).

[29] Mivule, Kato & Hopkinson, Kenneth. (2017). Distortion Search A Web Search Privacy Heuristic.

[30] Moawad, Ibrahim & Talha, Hanaa & Hosny, Ehab & Hashem, Mohammed. (2012). Agent-based web search personalization approach using dynamic user profile. Egyptian Informatics Journal. 10.1016/j.eij.2012.09.002.

[31] Nasr, Milad, et al. "DeepCorr: Strong Flow Correlation Attacks on Tor Using Deep Learning." Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2018, pp. 196276. DOI.org (Crossref), doi:10.1145/3243734.3243824.

[32] Petit, Albin. "Introducing Privacy in Current Web Search Engines", PhD-Thesis, Universitt Passau 2017

[33] Niwattanakul, Suphakit Singthongchai, Jatsada Naenudorn, Ekkachai Wanapu, Supachanun. (2013). Using of Jaccard Coefficient for Keywords Similarity.

[34] Petit, Albin, et al. "SimAttack: Private Web Search under Fire." Journal of Internet Services and Applications, vol. 7, no. 1, Dec. 2016, p. 2. DOI.org (Crossref), doi:10.1186/s13174-016-0044-x.

[35] Petit, Albin & Cerqueus, Thomas & Mokhtar, Sonia & Brunie, Lionel & Kosch, Harald. (2015). PEAS: Private, Efficient and Accurate Web Search. 571-580. 10.1109/Trustcom.2015.421.

[36] Quantifying Differential Privacy in Continuous Data Releases under Temporal Conditions https://arxiv.org/pdf/1711.11436.pdf

[37] "Qwant." Qwant, https://www.qwant.com/. Accessed 29 Apr. 2020.

[38] Reiter, Michael K., and Aviel D. Rubin. "Crowds: Anonymity for Web Transactions." ACM Transactions on Information and System Security, vol. 1, no. 1, Nov. 1998, pp. 6692. DOI.org (Crossref), doi:10.1145/290163.290168.

[39] Rennhard, Marc. (2002). MorphMix: Peer-to-Peer based Anonymous Internet Usage with Collusion Detection.

[40] Rodrigo-Gins, F. J., Parra-Arnau, J., Meng, W., & Wang, Y. (2018). PrivacySearch - An end-user and query generalization tool for privacy enhancement in web search. In M. H. Au, X. Luo, J. Li, K. Kluczniak, S. M. Yiu, C. Wang, & A. Castiglione (Eds.), Network and System Security (pp. 304-318). Springer. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 11058 https://doi.org/10.1007/978-3-030-02744-5_23

[41] Sánchez, David et al. "Knowledge-based scheme to create privacy-preserving but semantically-related queries for web search engines." Inf. Sci. 218 (2013): 17-30.

[42] Satvat, Kiavash & Forshaw, Matt & Hao, Feng & Toreini, Ehsan. (2014). On the Privacy of Private Browsing A Forensic Approach. Journal of Information Security and Applications. 19. 10.1016/j.jisa.2014.02.002.

[43] Shaikh A., R. Tewari and M. Agrawal, "On the effectiveness of DNS-based server selection," Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213), Anchorage, AK, USA, 2001, pp. 1801-1810 vol.3.

[44] Shalla Secure Services KG. http://www.shallalist.de/. Accessed 29 Apr. 2020.

[45] Searx - Privacy-Respecting Metasearch Engine. https://searx.me/. Accessed 29 Apr. 2020.

[46] Singel, Ryan. Netflix Spilled Your Brokeback Mountain Secret, Lawsuit Claims. Wired, Dec. 2009. www.wired.com, https://www.wired.com/2009/12/netflix-privacy-lawsuit/.

[47] Stamou, S., Ntoulas, A.: Search Personalization through Query and Page TopicalAnalysis. User Modeling and User-Adapted Interaction (February 2009).

[48] Tor Blog. "Traffic Correlation Using Netflows". Tor Blog. https://blog.torproject.org/traffic-correlation-using-netflows. Accessed 29 Apr. 2020.

[49] Tor Browser user manual. In The Tor Project at https://tbmanual.torproject.org/. Last accessed on April 29 2020.

[50] TrackMeNot. http://trackmenot.io/. Accessed 29 Apr. 2020.

[51] Wang, Tao et al. "Effective Attacks and Provable Defenses for Website Fingerprinting." USENIX Security Symposium (2014).

[52] What ISPs Can See. https://www.upturn.org/reports/2016/what-isps-can-see/#four-key-technical-clarifications. Accessed 29 Apr. 2020.

[53] Ye S., F, Wu, R. Pandey and H. Chen, "Noise Injection for Search Privacy Protection," 2009 International Conference on Computational Science and Engineering, Vancouver, BC, 2009, pp. 1-8.