# Self-Defense: Optimal QIF Solutions and Application to Website Fingerprinting

Andreas Athanasiou, Konstantinos Chatzikokolakis, Catuscia Palamidessi

# Self-Defense: Optimal QIF Solutions and Application to Website Fingerprinting

Andreas Athanasiou
*INRIA, École Polytechnique*
Palaiseau, France
andreas.athanasiou@inria.fr

Konstantinos Chatzikokolakis
*National and Kapodistrian University of Athens*
Athens, Greece
kostasc@di.uoa.gr

Catuscia Palamidessi
*INRIA, École Polytechnique*
Palaiseau, France
catuscia@lix.polytechnique.fr

***Abstract*—Quantitative Information Flow (QIF) provides a robust information-theoretical framework for designing secure systems with minimal information leakage. While previous research has addressed the design of such systems under hard constraints (e.g. application limitations) and soft constraints (e.g. utility), scenarios often arise where the core system's behavior is considered fixed. In such cases, the challenge is to design a new component for the existing system that minimizes leakage without altering the original system.**

**In this work we address this problem by proposing optimal solutions for constructing a new row, in a known and unmodifiable information-theoretic channel, aiming at minimizing the leakage. We first model two types of adversaries: an *exact-guessing* adversary, aiming to guess the secret in one try, and a *s-distinguishing* one, which tries to distinguish the secret $s$ from all the other secrets. Then, we discuss design strategies for both fixed and unknown priors by offering, for each adversary, an optimal solution under linear constraints, using Linear Programming.**

**We apply our approach to the problem of website fingerprinting defense, considering a scenario where a site administrator can modify their own site but not others. We experimentally evaluate our proposed solutions against other natural approaches. First, we sample real-world news websites and then, for both adversaries, we demonstrate that the proposed solutions are effective in achieving the least leakage. Finally, we simulate an actual attack by training an ML classifier for the s-distinguishing adversary and show that our approach decreases the accuracy of the attacker.**

***Index Terms*—Quantitative Information Flow, Website Fingerprinting, Leakage, Smallest Enclosing Ball**

## I. Introduction

A fundamental link in the chain of secure system designing is the quantification of the protection its users enjoy, or conversely, determining how much information is *leaked* to an adversary. A system is traditionally described by a channel $C$, which is a probability distribution over its observable outputs. The adversary is assumed to have some *prior* knowledge before the system's execution, and the goal is to determine how much information the adversary has afterwards, due to the system's leakage during execution.

Quantitative Information Flow (QIF) [1] studies measures of information *leakage*. The main idea is simple: it quantifies a system's leakage by *comparing* the secret's *vulnerability* or *risk* before (prior) and after (posterior) executing the system.

When designing a system from scratch, one can use QIF to create a channel that minimizes leakage. However here we consider a different scenario where a system and its channel are already given, and we just want to add a new component. We assume that we can design this new component in any way we want (or perhaps under some constraints, which we discuss later on), but we cannot modify anything in the existing system.

For example consider the following problem: the administrator of a website $s$ wants to prevent Website Fingerprinting (WF); that is, prevent an adversary from inferring which site a user visits, over an encrypted connection, from observable information such as the page size. To achieve this goal, the administrator wants to make his website similar to a set $\mathbb{S} \setminus \{s\}$ of other websites. The behavior of those websites is completely known: for each $t \in \mathbb{S} \setminus \{s\}$, the distribution $C_t$ of observations produced by a visitor of $t$ is given. Note that $t$ may or may not be using obfuscation to prevent fingerprinting; this is reflected in the distribution $C_t$. The administrator of $s$ cannot possibly modify the *other* websites, but he has full control over *his own*. For instance, he can pad his site to make the page size match any other known page. Moreover, this can be done probabilistically, by constructing an arbitrary distribution $q$ of observations to be produced by the visitors of $s$.

We call this scenario *self-defense*, since the administrator of the website $s$ tries to protect his site using his own means. The fundamental question then is: *which distribution $q$ provides the best self-defense*?

Note that, in this work, we assume that the administrator cannot rely on the cooperation of any other websites. This is a crucial assumption, that significantly affects the optimal solution. Indeed, it is well-known in game theory that cooperative and non-cooperative games have very different outcomes. For instance, employing a common defense strategy can benefit all parties involved, while if only one site employs that particular strategy and the others do not, that site may become more vulnerable. Take, for instance, padding: if all administrators pad their website to the same value $v$, they will generate the same observation. However, if only one website implements the padding to $v$ while the others do not, the attacker will single out that website more easily.

The work of Andreas Athanasiou and Konstantinos Chatzikokolakis was supported by the project CRYPTECS, funded by the ANR (project number ANR-20-CYAL-0006) and by the BMBF (project number 16KIS1439). The work of Catuscia Palamidessi was supported by the project HYPATIA, funded by the ERC (grant agreement number 835294).

To state the problem formally in the language of QIF, consider a system described by the channel $C : \mathbb{S} \to \mathcal{D}(\mathbb{O})$. All rows $C_t, t \neq s$ are fixed, but we can freely choose the row $C_s$. Concretely, we want to construct a new channel $C^q : \mathbb{S} \to \mathcal{D}(\mathbb{O})$, such that $C^q_t = C_t$ for all $t \neq s$, while the row $C^q_s$ can be set to a new distribution $q \in \mathcal{D}(\mathbb{O})$. The question now becomes: *how to choose $q$ to minimize the leakage of $C$?*

Of course this problem is not specific to website fingerprinting; it arises in any QIF scenario in which each secret corresponds to a different *user* or *entity*, which has full control over his own part of the system, but no control over the remaining system. Note that the knowledge of $C_t$ is vital for choosing $q$; we want to make our secret as similar as possible to the other ones.

One may think of choosing $q$ so to replicate a specific site $t'$, or to emulate the "average" website. These naive approaches, however, fall short of providing an optimal solution. In this work, we tackle this problem by using QIF, rather than resorting to an ad-hoc solution. In particular, we will use the version of QIF known as $g$-leakage [2], which allows us to express the capabilities and goals of a large class of adversaries. We consider two kinds of adversaries: one that tries to exactly guess the secret in one try (*exact-guessing*) and one that tries to distinguish the secret $s$ from all the other secrets (*$s$-distinguishing*).

Until now we have assumed complete freedom in designing the new row. In some situations, however, such freedom could be unrealistic: there may be technical constraints, which of course will increase the complexity of the problem. For instance, in the WF example, one can easily increase the page size by padding but cannot reduce it (at least, not to an arbitrary value, as infinite compression is not possible). Therefore, we also consider the problem of finding the optimal solution in the presence of constraints on the new row $C_s$.

We summarize our contributions as follows:

- For a fixed prior, we show how to optimally create $q$ for both the exact-guessing and $s$-distinguishing adversary (Section IV).
- When the prior is unknown, we first discuss how to optimally create $q$ for the exact-guessing adversary, and then explore the $s$-distinguishing adversary (Section V). We demonstrate that the latter reduces to the problem of the Smallest Enclosing Ball (Section VI), offering an optimal solution and discussing computationally lighter alternatives.
- In all cases we demonstrate how to incorporate the constraints in the proposed solutions using Linear Programming (LP).
- Finally, we evaluate our proposed solutions by simulating a website fingerprinting attack and compare them with other conventional approaches for both adversaries (Section VII). The experiments confirm that our approaches offer indeed the minimum leakage and decrease the accuracy of the attack.

### A. Related Work

The works by Simon et al. [3] and Reed et al. [4] were motivated by a similar traffic analysis attack; they explored how a server can efficiently pad its files in order to minimize leakage. In their setting, a user selects a file from the server which is padded according to some probabilistic mechanism. Meanwhile, an adversary, observing the encrypted network, attempts to guess the selected file. Our case differs as we strive to make a website indistinguishable from others (which are considered fixed), rather than making the pages of the website indistinguishable among themselves.

On the other hand, designing a channel to minimize leakage by being able to manipulate all it's rows (instead of only one, as in our case) under some constraints, has been studied in the literature [5], [6]. In fact, by properly selecting these constraints one can fix the value of all rows but one, essentially mimicking our scenario. Using the techniques from [5] under such constraints, one can arrive at optimization programs similar to those of Prop. 2 and 4. However, studying in depth the specific problem of single row optimization is interesting since, first, we can give direct solutions such as Prop. 3 and second, we can give capacity solutions for an s-distinguishing adversary, which is outside the scope of [5] and a main contribution of our work.

P. Malacaria et al. [7] discussed the complexity of creating a deterministic channel aimed at minimizing leakage for a given leakage metric and a specific prior while satisfying a set of constraints. They first showed that this is an NP-hard problem and then they proposed a solution for a particular class of constraints, by introducing a greedy algorithm. The authors show that this algorithm offers optimal leakage across most entropy measures used in the literature.

Moreover, one could opt to explore the problem within the popular framework of Differential Privacy (DP) [8]. DP aims to protect the privacy of an individual within a statistical database when queried by an analyst (considered the adversary) seeking aggregated information. A trusted central server aggregates users' data and introduces noise before publishing the noisy result to the analyst, making it hard for her to distinguish an individual's value. The (central) DP solution however is not suitable in our case, because there is no central server, and the attacker can observe directly the obfuscated version of the secret (the output of the channel).

In contrast, Local Differential Privacy (LDP) [9] removes the need for a central entity. Instead, users autonomously inject noise into their data. Notably, all users employ the same probabilistic mechanism to perturb their data without knowledge of other users' values and perturbations. LDP is more suitable for the situation we are considering. However, our scenario involves only one secret having the ability to be modified, and assumes knowledge of how the other secrets are treated by the system. Furthermore, LDP (as well as DP) is a worst-case metric that, for every observable, requires a certain level of indistinguishability between our secret $s$ (corresponding to the row that we want to modify) and every other secret

(row) $t$. This may be impossible to satisfy, as it would require that all the rows are almost indistinguishable as well, but as we already explained, the other rows are already given and cannot be modified.

Another metric proposed to measure the vulnerability of a secret is the multiplicative Bayes risk leakage $\beta$ [10], which was inspired by the cryptographic notion of *advantage*. $\beta$ corresponds to the (multiplicative) $g$-leakage in the case of a Bayesian attacker. In [11], the authors showed that $\beta$ can also be used to quantify the risk for the two most vulnerable secrets.

The related problem of *location privacy* has also been studied in the QIF literature using game theory and LP approaches. For instance, [12] discussed the optimal user strategy when a desired Quality of Service (QoS) and a target level of privacy are given, taking into account the prior knowledge of the adversary. Moreover, [13], motivated by DP and distortion-privacy (i.e. inference error), provided a utility-maximizing obfuscation mechanism with formal privacy guarantees, aiming to solve the trade-off between QoS and privacy. Their mechanism is based on formulating a Stackelberg game and solving it via LP, using the DP guarantee as a constraint. They claim their approach is utility-wise superior to DP while offering the same privacy guarantees. Furthermore, [14] studied how to maximize the QoS while achieving a certain level of geo-indistinguishability using LP, and discussed methods to reduce the constraints from cubic to quadratic, significantly reducing the required computation time. Note that an extension of this problem, which aims to protect entire location trajectories rather than individual locations (known as *trajectory privacy*), has also been studied using similar techniques [15].

## II. Preliminaries

The basic building blocks of QIF are summarized in the following sections.

*Prior vulnerability and risk:* A natural framework for expressing vulnerability is in terms of *gain* functions. Let $\mathbb{S}$ be the set of all possible secrets (e.g. all valid passwords); the adversary does not fully know the secret, but he possesses some *probabilistic knowledge* about it, expressed as a *prior distribution* $\pi : \mathcal{D}(\mathbb{S})$. In order to exploit this knowledge, the adversary will perform some *action* (e.g. make a guess about the password in order to access the user's system); the set of all available actions is denoted by $\mathcal{W}$. Often we take $\mathcal{W} = \mathbb{S}$, meaning that the adversary is trying to guess the exact secret; however expressing certain adversaries might require a different choice of $\mathcal{W}$ (see §III for an example).

A gain function $g(w, s)$ expresses the gain obtained by performing the action $w \in \mathcal{W}$ when the secret is $s \in \mathbb{S}$. The *expected gain* (wrt $\pi$) of action $w$ is $\sum_s \pi_s g(w, s)$. Being rational, the adversary will choose the action that maximizes his expected gain; we naturally define *$g$-vulnerability* $V_g(\pi)$ as the expected gain of the *optimal* action:

$$V_g(\pi) \;=\; \max_w \sum_s \pi_s g(w, s) \;.$$

Alternatively, it is often convenient to measure the adversary's *failure* (instead of his success). This can be done in a dual manner by expressing the *loss* $\ell(w, s)$ occurred by the action $w$. The adversary tries to minimize his loss, hence *$\ell$-risk* is defined as the expected loss of the optimal $w$.

$$R_\ell(\pi) \;=\; \min_w \sum_s \pi_s \ell(w, s) \;.$$

Note that $V_g(\pi)$ is a *vulnerability* function, expressing the adversary's *success* in achieving his goal, while $R_\ell(\pi)$ is a *risk* function (also known as *uncertainty* or *entropy*), expressing the adversary's failure.

*Posterior Vulnerability and Leakage:* So far we modeled the adversary's success or failure *a-priori*, that is given only some prior information $\pi$, before executing the system of interest. Given an output $o$ of a system $C : \mathbb{S} \to \mathbb{O}$, the adversary applies Bayes law to convert his prior $\pi$ into a *posterior* knowledge $\delta$, which is exactly what causes information leakage. Hence, $V_g(\delta)$ expresses the adversary's success *after observing* $o$ (and similarly for $R_\ell(\delta)$). Since outputs are selected randomly, and each produces a different posterior, we can intuitively define the *posterior* $g$-vulnerability $V_g(\pi, C)$ and the posterior $\ell$-risk $R_\ell(\pi, C)$,[1] as the expected value of $V_g(\delta)$ and $R_\ell(\delta)$ respectively:

$$\begin{aligned} V_g(\pi, C) \;&=\; \mathrm{E}[V_g(\delta)] \;=\; \textstyle\sum_o \max_w \sum_s \pi_s C_{s,o} g(w, s) \;, \\ R_\ell(\pi, C) \;&=\; \mathrm{E}[R_\ell(\delta)] \;=\; \textstyle\sum_o \min_w \sum_s \pi_s C_{s,o} \ell(w, s) \;. \end{aligned}$$

An adversary might succeed in his goal – say, to guess the user's password – due to two very distinct reasons: either because the user is choosing weak passwords (that is, the *prior* vulnerability is high), or because the *system* is leaking information about the password, causing the *posterior* vulnerability to increase. When studying the privacy of a system we want to focus on the information leak caused by the system *itself*. This is captured by the notion of *leakage*, the fundamental quantity in QIF, which simply compares the vulnerability before and after executing the system. The *multiplicative* leakage of $C$ wrt $\pi$ and $g/\ell$ is defined as:

$$\mathcal{L}_g(\pi, C) \;=\; \tfrac{V_g(\pi, C)}{V_g(\pi)} \;, \qquad \mathcal{L}_\ell(\pi, C) \;=\; \tfrac{R_\ell(\pi)}{R_\ell(\pi, C)} \;.$$

A leakage of 1 means *no leakage* at all: the prior and posterior vulnerability (or risk) are exactly the same. On the other hand, a *high leakage* means that, after observing the output of the channel, the adversary is more successful in achieving his goal than he was before. Note that $V_g$ always *increases* as a result of executing the system, while $R_\ell$ always *decreases*, this is why the fractions in the definitions of $\mathcal{L}_g$ and $\mathcal{L}_\ell$ are reversed.[2] We use $h$ to denote a function that can be either a gain or a loss function, which is useful to treat both types together

[1] $R_\ell(\pi, C)$ in often called *Bayes risk* (wrt $\ell$).

[2] In [16], Bayes *security* is defined as $\beta(\pi, C) = 1/\mathcal{L}_\ell(\pi, C)$; in this paper we use leakage, which is the standard notion in the QIF literature; all results can be trivially translated to $\beta(\pi, C)$.

*Guessing the exact secret:* The simplest and arguably most natural adversary is the one that tries to guess the *exact secret* in one try. We call this adversary *exact-guessing*; he can be easily expressed by setting $\mathcal{W} = \mathbb{S}$, and defining gain and loss as follows:

$$g_{\mathrm{x}}(w,x) \;=\; \mathbf{1}_{\{w\}}(x)\;, \quad \ell_{\mathrm{x}}(w,x) \;=\; 1-\mathbf{1}_{\{w\}}(x)\;,$$

where $\mathbf{1}_S(x)$ is the indicator function (equal to 1 iff $x \in S$ and 0 otherwise). For this adversary, vulnerability and risk reduce to the following expressions:

$$V_{g_{\mathrm{x}}}(\pi) = \max_s \pi_s\;, \qquad V_{g_{\mathrm{x}}}(\pi,C) = \textstyle\sum_y \max_s \pi_s C_{s,o}\;,$$
$$R_{g_{\mathrm{x}}}(\pi) = 1 - V_{g_{\mathrm{x}}}(\pi)\;, \quad R_{\ell_{\mathrm{x}}}(\pi,C) = 1 - V_{g_{\mathrm{x}}}(\pi,C)\;.$$

$V_{g_{\mathrm{x}}}$ is known as *Bayes vulnerability*, while $R_{\ell_{\mathrm{x}}}$ as *Bayes error* or *Bayes risk*.[3]

*Capacity:* Finally, the notion of *$g/\ell$-capacity* is simply the worst-case leakage of a system wrt all priors. Bounding the capacity of a system means that its leakage will be bounded independently from the prior available to the adversary.

$$\mathcal{ML}_h(C) \;=\; \max_\pi \mathcal{L}_h(\pi,C)\;, \quad h \in \{g,\ell\}\;.$$

Denote by $\mathbf{u}$ the uniform prior, and by $\mathbf{u}^{s,t}$ the prior that is uniform among these two secrets, i.e. it assigns probability $1/2$ to them. For an exact-guessing adversary (i.e. for $g_{\mathrm{x}}, \ell_{\mathrm{x}}$), the following results are known:

**Theorem 1.** *$\mathcal{ML}_{g_{\mathrm{x}}}(C)$ is equal to:*

$$\mathcal{ML}_{g_{\mathrm{x}}}(C) \;=\; \mathcal{L}_{g_{\mathrm{x}}}(\mathbf{u},C) \;=\; \textstyle\sum_o \max_s Cs,o\;.$$

In the following, we use the $L_1$-metric $\|q-q'\|_1$ as the distance between probability distributions $q, q'$; note that this is equal to twice their *total variation* distance, since we only consider discrete distributions. Denote by $d_{\max}(S,T)$ the maximum distance between elements of sets $S,T$, and by $\mathrm{diam}(S) = d_{\max}(S,S)$ the *diameter* of $S$. Finally, given a subset of secrets $P \subseteq \mathbb{S}$, we denote by $C_P \subset \mathcal{D}(\mathbb{O})$ the set of rows of $C$ indexed by $P$ (note that rows are probability distributions); the set of all rows will be $C_{\mathbb{S}}$ while the $s$-th row will be denoted by $C_s$.

**Theorem 2** ( [16]). *$\mathcal{ML}_{\ell_{\mathrm{x}}}(C)$ is equal to:*

$$\mathcal{ML}_{\ell_{\mathrm{x}}}(C) \;=\; \mathcal{L}_{\ell_{\mathrm{x}}}(\mathbf{u}^{s,t},C) \;=\; \frac{1}{1-\frac{1}{2}\mathrm{diam}(C_{\mathbb{S}})}\;.$$

*where $s,t \in \mathbb{S}$ realize $\mathrm{diam}(C_{\mathbb{S}})$.*

## III. Guessing predicates and distinguishing a specific secret

Although systems are commonly evaluated against the exact-guessing adversaries $g_{\mathrm{x}}, \ell_{\mathrm{x}}$, there are many practical scenarios in which the adversary is not necessarily interested in guessing the *exact* secret. A general class of such adversaries are those aiming at guessing only a *predicate* $P$ of the secret, for instance *is the user located close to a hospital?* or *is the sender a male?*. We call this adversary *$P$-guessing*.

A predicate can be described by a subset of secrets $P \subseteq \mathbb{S}$, those that satisfy the predicate; the rest are denoted by $\neg P = \mathbb{S} \setminus P$. A $P$-guessing adversary can be easily captured using gain/loss functions, by setting $\mathcal{W} = \{P, \neg P\}$ and defining

$$g_P(w,s) \;=\; \mathbf{1}_w(s)\;, \quad \ell_P(w,s) \;=\; 1-\mathbf{1}_w(s)\;,$$

for $w \in \mathcal{W}$. Then $\mathcal{L}_{g_P}$ and $\mathcal{L}_{\ell_P}$ measure the system's leakage wrt an adversary trying to guess $P$.

Although this class of adversaries was discussed in [2], it has received little attention in the QIF literature. In this section, we first show that $\mathcal{L}_{g_P}$ can be expressed as the Bayes leakage $\mathcal{L}_{g_{\mathrm{x}}}$ of a properly constructed pair of prior/channel (and similarly for $\mathcal{L}_{\ell_P}$). Then, we study the capacity problem for this adversary.

*Expressing $\mathcal{L}_{g_P}$ as $\mathcal{L}_{g_{\mathrm{x}}}$:* Since every secret belongs to either $P$ or $\neg P$, given a prior $\pi \in \mathcal{D}(\mathbb{S})$ we can construct a *joint distribution* between secrets $s$ and classes $w$, where $\Pr(w,s) = \pi_s \cdot \mathbf{1}_w(s)$ gives the probability to have both $w$ and $s$ together. This joint distribution can be factored into a *marginal* distribution $\rho^\pi \in \mathcal{D}(\mathcal{W})$, and *conditional* distributions – i.e. a *channel* – $Q^\pi : \mathcal{W} \to \mathbb{S}$:[4]

$$\rho^\pi_w \;=\; \Pr(w) \;=\; \textstyle\sum_{s\in w} \pi_w\;, \tag{1}$$
$$Q^\pi_{w,s} \;=\; \Pr(s \mid w) \;=\; \frac{\pi_s \cdot \mathbf{1}_w(s)}{\rho^\pi_w}\;. \tag{2}$$

In the notation we emphasize that both $\rho^\pi, Q^\pi$ depend on $\pi$.[5]

This construction allows us to express $g_P$-leakage as $g_{\mathrm{x}}$-leakage, for a different pair of prior/channel, as stated in the following lemma.

**Lemma 1.** *Let $\pi \in \mathbb{S}, P \subseteq \mathbb{S}$ and define $\rho^\pi, Q^\pi$ as in (1),(2). Then for all channels $C$ it holds that:*

$$\mathcal{L}_{g_P}(\pi,C) \;=\; \mathcal{L}_{g_{\mathrm{x}}}(\rho^\pi, Q^\pi C)\;,$$
$$\mathcal{L}_{\ell_P}(\pi,C) \;=\; \mathcal{L}_{\ell_{\mathrm{x}}}(\rho^\pi, Q^\pi C)\;.$$

*Capacity:* Lem. 1 can be used to obtain capacity results for any $P$-guessing adversary.

**Theorem 3.** *For any $C$ and $P \subseteq \mathbb{S}$ it holds that*

$$\mathcal{ML}_{g_P}(C) \;=\; \mathcal{L}_{g_P}(\mathbf{u}^{s,t},C) \;=\; 1 + \frac{1}{2}\, d_{\max}(C_P, C_{\neg P})\;,$$
$$\mathcal{ML}_{\ell_P}(C) \;=\; \mathcal{L}_{\ell_P}(\mathbf{u}^{s,t},C) \;=\; \frac{1}{1-\frac{1}{2}\, d_{\max}(C_P, C_{\neg P})}\;,$$

*for $s \in P, t \in \neg P$ realizing $d_{\max}(C_P, C_{\neg P})$.*

### A. Distinguishing a specific secret

As discussed, an adversary of particular interest is one that tries to *distinguish a secret $s$* from all other secrets, that is to answer the question *is the secret $s$ or not?*. We call this adversary *$s$-distinguishing*.

[3] The term Bayes risk has been used in the literature to describe both $R_{\ell_{\mathrm{x}}}$ and $R_\ell$ (for arbitrary $\ell$).

[4] In case $\rho^\pi_w = 0$ we can define the row $Q^\pi_w$ arbitrarily.

[5] They also depend on $P$, which is clear from the context hence omitted for simplicity.

Clearly, this is a special case of a $P$-guessing adversary, for $P = \{s\}$. In this case, we simply write $s, \neg s$ instead of $P, \neg P$ respectively. Hence $g_s = g_P, \ell_s = \ell_P$ are the gain/loss functions modeling an $s$-distinguishing adversary, and $\mathcal{L}_{g_s}, \mathcal{L}_{\ell_s}$ measure the system's leakage wrt this adversary.

For this choice of $P$, the binary channel $Q^\pi C : \{s, \neg s\} \to \mathbb{O}$ (see (2)) has a clear interpretation. Its first row is simply $Q^\pi_s C = C_s$; more interestingly, its second row $Q^\pi_{\neg s} C$ models the behavior of an *average* secret *other than* $s$: choosing randomly (wrt $\pi$) a secret $t \neq s$ and then executing $C$ on $t$ produces outputs distributed according to $Q^\pi_{\neg s} C$. The usefulness of this distribution will become apparent in the following sections.

Finally Thm. 3 provides a direct solution of the capacity problem for $s$-distinguishing adversaries. The capacity will be given by the maximum distance $d_{\max}(C_s, C_{\neg s})$ between the row $C_s$ and all other rows of the channel.

## IV. Optimizing $q$ for a fixed prior $\pi$

We return to the problem discussed in the introduction, that is choosing the distribution of observations $q$ produced by our secret of interest $s$. The QIF theory gives us a clear goal for this choice: we want to find the distribution $q \in \mathcal{F}$ that *minimizes the leakage* of the channel $C^q$.

Here $\mathcal{F} \subseteq \mathcal{D}(\mathbb{O})$ denotes the set of *feasible* solutions for $q$; this set can be arbitrary, and is typically determined by practical aspects of each application (e.g. it might be possible to increase the page size by padding, but not to decrease it). The only assumption we make in this paper is that $\mathcal{F}$ can be expressed in terms of *linear inequalities*. A solution $q^\star$ will be called *optimal* iff it is both feasible and minimizes leakage among all feasible solutions.

We start with the case when we have a specific prior $\pi$ modeling the system's usage profile, and we want to minimize leakage wrt that prior. Since the behavior $C_t$ of all secrets $t \neq s$ is considered fixed, and their relative probabilities are dictated by our fixed $\pi$, we know exactly how an "average" secret other than $s$ behaves: it produces observables with probability $Q^\pi_{\neg s} C$ (see §III-A). Conventional wisdom dictates that we should try to make $s$ similar to an average non-$s$ secret, that is choose

$$q^\star \;=\; Q^\pi_{\neg s} C$$

as our output distribution. Note that this $q^\star$ is a *convex combination* of $C$'s rows (other than $s$), the elements of $Q^\pi_{\neg s}$ being the convex coefficients.

We first show that this intuitive choice is indeed meaningful in a precise sense: it minimizes leakage[6] wrt an *$s$-distinguishing* adversary (§III-A). In fact, for this adversary the resulting channel has no leakage at all; the adversary is trying to distinguish $s$ from $\neg s$, but the two cases produce identical observations.

[6]Note that, for fixed $\pi$, optimizing $\mathcal{L}_g, \mathcal{L}_\ell$ is equivalent to optimizing $V_g, R_\ell$ since the prior vulnerability/risk is constant.

**Proposition 1.** *Let $\pi \in \mathcal{D}(\mathbb{S})$, $C : \mathbb{S} \to \mathbb{O}$, $s \in \mathbb{S}$ and let $q^\star = Q^\pi_{\neg s} C$. The channel $C^{q^\star}$ has no leakage wrt an $s$-distinguishing adversary, that is:*

$$\mathcal{L}_h(\pi, C^{q^\star}) \;=\; 1 \;, \quad h \in \{g_s, \ell_s\} \;.$$

An immediate consequence of Prop. 1 is that, if $q^\star = Q^\pi_{\neg s} C \in \mathcal{F}$, then it is optimal for an $s$-distinguishing adversary. However, it could very well be the case that this construction is infeasible, in which case finding the optimal feasible solution is more challenging.

Moreover, somewhat surprisingly, it turns out that even when $q^\star = Q^\pi_{\neg s} C$ is feasible, it does *not* minimize leakage wrt an exact-guessing adversary. Consider the following example:

$$\pi \;=\; (0.47, 0.29, 0.24) \qquad C^q \;=\; \begin{bmatrix} q_1 & q_2 \\ 0.05 & 0.95 \\ 0.58 & 0.42 \end{bmatrix}$$

For this channel and prior we get $q^\star = Q^\pi_{\neg s} C = (0.29, 0.71)$. Consider also the distribution $q = (0.42, 0.58)$. We have that

$$\begin{aligned} \mathcal{L}_{\ell_x}(\pi, C^{q^\star}) \;&\approx\; 1.1 \;, & \mathcal{L}_{\ell_s}(\pi, C^{q^\star}) \;&=\; 1 \;, \\ \mathcal{L}_{\ell_x}(\pi, C^{q}) \;&=\; 1 \;, & \mathcal{L}_{\ell_s}(\pi, C^{q}) \;&\approx\; 1.01 \;. \end{aligned}$$

We see that $q^\star$ does indeed minimize leakage for an $s$-distinguishing adversary (in fact, the leakage becomes 1), but it does not minimize the exact-guessing leakage; the latter is minimized by $q$.

Although the simple construction $q^\star = Q^\pi_{\neg s} C$ does not always produce an optimal solution, as discussed above, finding an optimal one is still possible for all adversaries and arbitrary $\mathcal{F}$, via linear programming.

**Proposition 2.** *The optimization problem*

$$q^\star \;:=\; \operatorname*{arg\,min}_{q \in \mathcal{F}} \mathcal{L}_h(\pi, C^q) \;, \quad h \in \{g_x, g_s, \ell_x, \ell_s\} \;, \tag{3}$$

*can be solved in polynomial time via linear programming.*

For instance, the optimal solution for the $g_x$ adversary is given by the following linear program (recall that we assumed $q \in \mathcal{F}$ to be expressible in terms of linear inequalities):

$$\begin{array}{lll} \text{minimize} & \sum_{o \in \mathbb{O}} z_o & \\ \text{subject to} & q \in \mathcal{F} & \\ & z_o \geq \max\limits_{t \in \mathbb{S} \setminus \{s\}} \pi_t C_{t,o} & \forall o \in \mathbb{O} \\ & z_o \geq \pi_s q_o & \forall o \in \mathbb{O} \end{array}$$

While for $g_s$, the program is similar:

$$\begin{array}{lll} \text{minimize} & \sum_{o \in \mathbb{O}} z_o & \\ \text{subject to} & q \in \mathcal{F} & \\ & z_o \geq \sum_{t \in \mathbb{S} \setminus \{s\}} \pi_t C_{t,o} & \forall o \in \mathbb{O} \\ & z_o \geq \pi_s q_o & \forall o \in \mathbb{O} \end{array}$$

## V. Optimizing $q$ for an unknown prior

In the previous section we discussed finding the optimal $q$ then the prior $\pi$ (i.e. the user profile) is fixed. In practice, however, we often do not know $\pi$ or we do not want to restrict to a specific one. The natural goal then is to design our system wrt the *worst* possible prior. The QIF theory will again offer guidance in selecting $q$; this time we choose the one that minimizes $C^q$'s *capacity*, i.e. we minimize its maximum leakage wrt all priors.

### A. Exact-guessing adversary

Starting with the problem of optimizing $q$ wrt an exact-guessing adversary, in this section we make two observations. First, we show that finding an optimal $q$ is possible either via simple convex combinations of rows (provided that such solutions are feasible), or via linear programming. Second, and more important, we show that selecting $q$ solely wrt this adversary is a poor design choice, since many values are simultaneously optimal, although their behavior is not equivalent for other adversaries.

Recall that the natural choice of $q$ for a fixed prior (§IV) was $q^\star = Q^\pi_{\neg s} C$, which is a convex combination of the rows $C_{\neg s}$. It turns out that for an unknown prior and an exact-guessing adversary we can choose much more freely: *any* convex combination of the rows $C_{\neg s}$ minimizes capacity. To understand this fact, consider $g_\mathrm{x}$-capacity and recall that $\mathcal{ML}_{g_\mathrm{x}}(C^q)$ is given by the sum of the column maxima of the channel (Thm. 1). Adding a new row cannot decrease the column maxima, hence $\mathcal{ML}_{g_\mathrm{x}}(C^q) \geq \mathcal{ML}_{g_\mathrm{x}}(C^q_{\neg s})$. Moreover, achieving equality is trivial: setting $q$ to any convex combination of rows means that no element of $q$ can be strictly greater than all corresponding elements of $C^q_{\neg s}$, hence $C^q$ and $C^q_{\neg s}$ will have the exact same column maxima. This brings us to the following result:

**Proposition 3.** *For all $C$, any $q^\star \in \mathrm{ch}(C_{\neg s})$ minimizes capacity for exact-guessing adversaries, that is*

$$\mathcal{ML}_h(C^{q^\star}) \leq \mathcal{ML}_h(C^q) \quad \forall q \in \mathcal{D}(\mathbb{O}), h \in \{g_\mathrm{x}, \ell_\mathrm{x}\} .$$

*Moreover, it holds that $\mathcal{ML}_h(C^{q^\star}) = \mathcal{ML}_h(C_{\neg s})$.*

A direct consequence of Prop. 3 is that any convex combination of the rows $C_{\neg s}$ that happens to be feasible, that is any $q^\star \in \mathcal{F} \cap \mathrm{ch}(C_{\neg s})$, is an optimal solution for an exact-guessing adversary. Note that this is a sufficient but not necessary condition for optimality. In §V-B we see that solutions outside the convex hull can be also optimal.

On the other hand, there is no guarantee that any such solution exists, it could very well be the case that no convex combination of rows is feasible. In this case, we can still compute an optimal solution, as follows:

**Proposition 4.** *The optimization problem*

$$q^\star := \operatorname*{arg\,min}_{q \in \mathcal{F}} \mathcal{ML}_h(C^q) , \quad h \in \{g_\mathrm{x}, \ell_\mathrm{x}\} , \tag{4}$$

*can be solved in polynomial time via linear programming.*

The linear program for $g_\mathrm{x}$ is given below:

$$\begin{array}{lll} \text{minimize} & \sum_{o \in \mathbb{O}} z_o & \\ \text{subject to} & q \in \mathcal{F} & \\ & z_o \geq \max_{t \in \mathbb{S} \setminus \{s\}} C_{t,o} & \forall o \in \mathbb{O} \\ & z_o \geq q_o & \forall o \in \mathbb{O} \end{array}$$

Prop. 3 states that a large set of choices for $q$ are all equivalent from the point of view of the exact-guessing capacity. However, this is not true wrt other types of adversaries, as the following example demonstrates:

$$C^q = \begin{array}{c|cc} & o_1 & o_2 \\ \hline s & q_1 & q_2 \\ s_1 & 1 & 0 \\ s_2 & 0 & 1 \end{array}$$

Here $C_{\neg s}$ is a deterministic channel with only 2 secrets that are completely distinguishable. For instance, two websites, without any obfuscation mechanism, having distinct page sizes. From Thm. 1 we can compute $\mathcal{ML}_{g_\mathrm{x}}(C^q_{\neg s}) = 2$. Since the rows $s_1, s_2$ are already maximally distant, the choice of $q$ is irrelevant. For *any* $q$, the rows $s_1, s_2$ will still be maximally distant, giving $\mathcal{ML}_{g_\mathrm{x}}(C^q) = 2$.

Although $q$ does not affect $\mathcal{ML}_{g_\mathrm{x}}$, this does not mean that the choice of $q$ is irrelevant for the website $s$. Setting $q = (1, 0)$ we make $s$ indistinguishable from $s_1$ but completely distinguishable from $s_2$. Conversely, $q = (0, 1)$ makes $s$ indistinguishable from $s_2$ but completely distinguishable from $s_1$. Finally, $q = (1/2, 1/2)$ makes $s$ somewhat indistinguishable from both $s_1$ and $s_2$; intuitively the latter seems to be a preferable choice, but *why*?

To better understand how $q$ affects the security of this channel we should study the difference between an exact-guessing and an $s$-distinguishing adversary. For the former, recall that $\mathcal{ML}_{g_\mathrm{x}}$ is always given by a uniform prior $\mathbf{u}$. For such a prior, an adversary who guesses the secret after observing the output will always guess $s_1$ after seeing $o_1$ (because $s_1$ always produces $o_1$) and $s_2$ after seeing $o_2$, independently from the values of $q$. So intuitively $q$ does not affect this adversary at all, which is the reason why $\mathcal{ML}_{g_\mathrm{x}}(C^q) = 2$ for any $q$.

However, for an $s$-distinguishing adversary, the situation is very different. When $q = (1, 0)$ we can use Thm. 3 to compute $\mathcal{ML}_{g_s}(C^q) = 2$, given by the prior $\mathbf{u}^{s,s_2}$; for this prior the adversary can trivially infer whether the secret is $s$ or $\neg s$ after the observation. But for $q' = (1/2, 1/2)$ we get $\mathcal{ML}_{g_s}(C^{q'}) = 3/2$, realized by both $\mathbf{u}^{s,s_1}$ and $\mathbf{u}^{s,s2}$. The system provides non-trivial privacy even in the worst case; $s$ and $\neg s$ can never be fully distinguished.

The discussion above suggests that maximizing the exact-guessing capacity by itself does not fully guide us in choosing $q$; it is meaningful to also optimize wrt an $s$-distinguishing adversary. In fact, in the next section we see that optimizing wrt *both* $s$-distinguishing and exact-guessing adversaries simultaneously is sometimes possible.

### B. $s$-distinguishing adversary

We turn our attention to optimizing $q$ wrt an $s$-distinguishing adversary for an unknown prior. We already know from Thm. 3 (for $P = \{s\}$) that both $g_s$ and $\ell_s$-capacities depend on the maximum distance $d_{\max}(C_s, C_{\neg s})$ between the row $C_s$ and all other rows of the channel. In other words, the capacity is related to the radius of the smallest $L_1$-ball centered at $C_s$ that contains $C_{\mathbb{S}}$.

This gives us a direct way of optimizing $q$ wrt $g_s, \ell_s$-capacity by a solving a *geometric* problem known as the *smallest enclosing ball* (SEB): find a vector that minimizes its maximal distance to a set of known vectors, or equivalently find the smallest ball that includes this set.

Interestingly, it turns out that in one particular case the SEB solution is guaranteed to be simultaneously optimal wrt an exact-guessing adversary. This happens in the *unconstrained* case $\mathcal{F} = \mathcal{D}(\mathbb{O})$, that is when any solution $q$ is feasible, as stated in the following result.

**Theorem 4.** *For all $C : \mathbb{S} \to \mathbb{O}$, any distribution given by*

$$q^\star \in \arg\min_{q \in \mathcal{F}} d_{\max}(q, C_{\neg s})$$

*gives optimal capacity for $s$-distinguishing adversaries:*

$$\mathcal{ML}_h(C^{q^\star}) \leq \mathcal{ML}_h(C^q) , \quad h \in \{g_s, \ell_s\}, q \in \mathcal{F} .$$

*Moreover, if $\mathcal{F} = \mathcal{D}(\mathbb{O})$, then $q^\star$ is simultaneously optimal for exact-guessing adversaries, i.e. for $h \in \{g_{\mathrm{x}}, \ell_{\mathrm{x}}\}$.*

Note that the solution $q^\star$ obtained from the above result might lie *outside* the convex hull of $C_{\neg s}$. So, in the unconstrained case, the simple optimality conditions of Prop. 3 are *not* met, yet the resulting solution is still guaranteed to be optimal also for exact-guessing adversaries.

The smallest enclosing ball problem is discussed in the next section, showing that it can be solved in linear time on $|\mathbb{S}|$ for fixed $|\mathbb{O}|$, or in polynomial time on $|\mathbb{S}| \cdot |\mathbb{O}|$.

## VI. The SEB problem

The following is known as the smallest enclosing ball (SEB) problem [17]: given a finite subset $S \subseteq M$ of some metric space $(M, d)$, find the smallest ball $B_r(x), x \in M, r \in \mathbb{R}$ that contains $S$. In this paper, the goal is to solve the problem for $M = \mathcal{F}$ and $d = L_1$ (see §V).

*Euclidean norm:* The problem is well-studied for $(\mathbb{R}^m, L_2)$ [18]. It has been shown that the solution is always unique and belongs to the convex hull of $S$. For any fixed $m$, it can be found in linear time on $n = |S|$. However, the dependence on $m$ is exponential[7]. Nonetheless, as we discuss below, approximation algorithms also exist that can compute a ball of radius at most $(1 + \epsilon)r^*$, where $r^*$ is the optimal radius and $\epsilon > 0$, in time linear on both $m$ and $n$.

[7]A sub-exponential algorithm does exist, but still its complexity is larger than any polynomial.

*Manhattan norm:* For $(\mathbb{R}^m, L_1)$ the problem is much less studied. The solution is *no longer unique*, due to the fact that $L_1$-balls have straight-line segments in their boundary. Moreover, somewhat surprisingly, *none* of the solutions is guaranteed to be in the convex hull of $S$.

Similarly to the Euclidean case, for fixed $m$ the problem can be solved in time linear on $n$, using the isometric embedding of $(\mathbb{R}^m, L_1)$ into $(\mathbb{R}^{2^m}, L_\infty)$. In contrast to the Euclidean case, however, the problem can be solved in polynomial time on both $n$ and $m$ via linear programming.

*Probability distributions:* Our case of interest is $(\mathcal{F}, L_1)$ for $\mathcal{F} \subseteq \mathcal{D}(\mathbb{O})$, the set of constrained probability distributions over some set finite $\mathbb{O}$, under the $L_1$-distance. Note that $\mathcal{D}(\mathbb{O})$ is a subset of $\mathbb{R}^m$ for $m = |\mathbb{O}|$. However, solving the $(\mathbb{R}^m, L_1)$-SEB problem does not immediately yield a solution for $(\mathcal{F}, L_1)$-SEB, since the center of the optimal ball might lie outside $\mathcal{F}$, or even outside $\mathcal{D}(\mathbb{O})$. This problem is studied in the following sections.

### A. Linear time solution for fixed $m$

We start from the fact that the $(\mathbb{R}^d, L_\infty)$-SEB problem admits a direct solution: given a set $S \subset \mathbb{R}^d$, denote by $S^\top, S^\bot \in \mathbb{R}^d$ the vectors of component-wise maxima and minima:

$$S_i^\top = \max_{x \in S} x_i \qquad S_i^\bot = \min_{x \in S} x_i \qquad i \in \{1, \dots, d\} . \quad (5)$$

It is easy to see that the optimal radius is $r^* = \frac{1}{2}\|S^\top - S^\bot\|_\infty$, and the (non-unique) optimal center is $x^* = \frac{1}{2}(S^\top + S^\bot)$.

Moving to the $(\mathbb{R}^m, L_1)$-SEB problem, we use a well-known embedding $\varphi : \mathbb{R}^m \to \mathbb{R}^{2^m}$ for which it holds that $\|\varphi(x) - \varphi(x')\|_\infty = \|x - x'\|_1$. Using the fact that $\varphi$ is invertible, an optimal solution $(x^*, r^*)$ for $(\mathbb{R}^{2^m}, L_\infty)$-SEB can be directly translated to an optimal solution $(\varphi^{-1}(x^*), r^*)$ for $(\mathbb{R}^m, L_1)$-SEB.

Turning our attention to our problem of interest, the $(\mathcal{F}, L_1)$-SEB case is a bit more involved. We can still use the same embedding $\varphi$, but an optimal solution $(x^*, r^*)$ for $(\mathbb{R}^{2^m}, L_\infty)$-SEB cannot be translated to our problem since $\varphi^{-1}(x^*)$ is not guaranteed to be a probability distribution; in fact no solution of radius $r^*$ is guaranteed to exist at all. Writing $\varphi(S)$ for $\{\varphi(x) \mid x \in S\}$, essentially what we need is to solve the $(\varphi(\mathcal{F}), L_\infty)$-SEB problem; in other words to impose that the solution is the translation of a feasible probability distribution.

The first step is to compute the vectors $\varphi(S)^\top, \varphi(S)^\bot$ of component-wise maxima and minima for each translated vector. Although we cannot directly construct the solution from these vectors (as we did for $(\mathbb{R}^d, L_\infty)$-SEB), the key observation is that these two vectors alone represent the maximal distance to the whole $S$, because $\forall y \in \mathbb{R}^m$:

$$\begin{aligned} \max_{x \in S} \|y - x\|_1 &= \max_{x \in S} \|\varphi(y) - \varphi(x)\|_\infty \\ &= \max\{\|\varphi(y) - \varphi(S)^\top\|_\infty, \|\varphi(y) - \varphi(S)^\bot\|_\infty\} . \end{aligned}$$

Then, we exploit the fact that $\varphi$ is a *linear map*; more precisely

$$\varphi(x) = x\Phi ,$$

where $\Phi$ is a $m \times 2^m$ matrix, having one column for each bitstring $b$ of size $m$, defined as $\Phi_{i,b} = (-1)^{b_i}$. This allows us to solve the $(\mathcal{F}, L_1)$-SEB problem via *linear programming*: we use $x \in \mathbb{R}^m$ as variables, imposing the linear constraints $x \in \mathcal{F}$. Moreover, we ask to minimize the $L_\infty$-distance between $x\Phi$ and $\varphi(S)^\top, \varphi(S)^\perp$, two vectors that we have computed in advance. The program can be written as:

$$\begin{aligned} \text{minimize} \quad & z \\ \text{subject to} \quad & x \in \mathcal{F} \\ & z \geq \quad \varphi(S)_b^\top - (x\Phi)_b \qquad \forall b \in \{0,1\}^m \\ & z \geq -\varphi(S)_b^\perp + (x\Phi)_b \qquad \forall b \in \{0,1\}^m \end{aligned}$$

Note that $\varphi(S)^\top, \varphi(S)^\perp$ can clearly be computed in $O(n)$ time. Given these vectors, the whole linear problem does not depend on $n$ (because it does not involve $S$). For fixed $m$, solving the linear program takes constant time, which implies the following result.

**Theorem 5.** *The $(\mathcal{D}(\mathbb{O}), L_1)$-SEB problem can be solved in $O(n)$ time for any fixed $m$.*

### B. Polynomial time solution for any dimension

In contrast to the Euclidean case, the dependence on $m$ for $(\mathcal{F}, L_1)$-SEB is polynomial. This is because the objective function $\max_{y \in S} \|x - y\|_1$, can be turned into a linear one using auxiliary variables. We use variables $w_{y,i}$ to represent $|x_i - y_i|$, and a variable $z$ to represent $\max_{y \in S} \sum_i |x_i - y_i| = \max_{y \in S} \sum_i w_{y,i}$.

The linear program can be written as:

$$\begin{aligned} \text{minimize} \quad & z \\ \text{subject to} \quad & x \in \mathcal{F} \\ & w_{y,i} \geq x_i - y_i \qquad \forall y \in S, i \in \{1, \ldots, m\} \\ & w_{y,i} \geq y_i - x_i \qquad \forall y \in S, i \in \{1, \ldots, m\} \\ & z \geq \sum_i w_{y,i} \qquad \forall y \in S \end{aligned}$$

**Theorem 6.** *The $(\mathcal{F}, L_1)$-SEB problem can be solved in polynomial time, using a linear program with $O(nm)$ variables and $O(nm)$ constraints.*

### C. Approximate solutions

The $(\mathcal{F}, L_1)$-SEB problem can be approximated by solving the $(\mathbb{R}^m, L_2)$-SEB problem for which several algorithms exist, and then projecting the solution to $\mathcal{F}$.

For an exact (Euclidean) solution, there are known algorithms claimed to handle dimensions up to several thousands [19]. Note that an exact solution is unique and is guaranteed to lie within the convex hull of $S$. Hence, when applied to distributions, the solution is guaranteed to be a distribution. Moreover, Prop. 1 guarantees that if the solution is feasible, it will be optimal for an exact-guessing adversary, although it will not be optimal for an $s$-distinguishing one.

Furthermore, there are several approximate algorithms that run in linear time or even better (see [20] for a recent work which provides several references). Their solution is not guaranteed to be a probability distribution, so a projection to $\mathcal{F}$ will be needed.

In the experiments of §VII, we call *SEB exact* the solution of Thm. 6, and *SEB approx* the solution obtained via a linear-time approximation algorithm.

## VII. Use-case: Website Fingerprinting

In this section we apply the optimization methods described in previous sections to defend against Website Fingerprinting (WF) attacks and evaluate their performance. In a WF attack the adversary observes the encrypted traffic pattern between a user and a website and tries to infer which website the user is visiting. This is a particularly interesting attack when the adversary cannot directly observe the sender of the intercepted packets, for example when traffic is sent through the Tor network [21] for which a series of WF attacks have been proposed [22], [23].

### A. Setup

Consider a news website covering "controversial" topics, prompting an adversary to target it and attempt to identify its readers via WF. To defend against such an attack, the administrator of this website would like to make its responses as indistinguishable as possible from other news sites, so that WF becomes harder.

For simplicity, in our evaluation we consider that only one request is intercepted by the adversary and only the size of the encrypted response is observed. The administrator's goal is to try to imitate other websites by producing pages that are similar in size. Such target websites produce responses according to distributions which are assumed to be known both to the administrator and to the adversary.

For our evaluation, we started by identifying the top 5 (in traffic) news website from 40 countries[8], leading to a total of 200 sites, of which one is selected as the defended site $s$. Then, we crawled these sites and measured the size of the received pages, rounded to the closest KB, creating a distribution over the page sizes. The biggest page size is 300KB, hence the output space of the distribution is 1KB, 2KB,.., 300KB.

Note that obtaining an accurate distribution of an *average* request requires knowledge of how the traffic is distributed across the different pages of the site. For instance, the probability of visiting a page typically decreases as the user navigates deeper into the site: it is more likely for users to read the headlines on the homepage than to access a page several links deeper.

For our evaluation, we simulate visitors by randomly following 10 links of every page up to a maximum depth of 4. A probability distribution over the page sizes is constructed by assuming that a visitor access each depth with the following probabilities:

- home page: 0.3
- 1-click depth: 0.25

[8]According to www.similarweb.com.

- 2-clicks depth: 0.2
- 3-clicks depth: 0.15
- 4-clicks depth: 0.1

while the probability of accessing pages within the same layer is uniform.

Note that defending against WF when we can control only a single site is quite challenging: the selected 200 sites are quite different from each other, so we cannot imitate all of them simultaneously. Instead, we assume that the administrator of $s$ selects a moderate subset of 19 sites close in distance to $s$, leading to a system of 20 secrets that we try to minimize its leakage.[9]

To successfully hide $s$ among the other 19 sites, the administrator needs to find $q$ (i.e. a distribution over the page sizes) in order to reduce the leakage (or capacity) of $C$. Then, he needs to modify his site so that the served pages follow this distribution.

Note, however, that not all distributions $q$ are feasible for the defended site since the administrator still needs to respect the site's existing content. While increasing a page size is usually straightforward via padding [24], decreasing it may not be feasible without affecting the content. In the following experiments we take this issue into account by enforcing a *non-negative padding* constraint $\mathcal{F}$, which is discussed below.

*Priors:* Let us first describe the priors that we are about to use in the following experiments:

- *Uniform*: **u**, that is probability $1/20$ for each site.
- *Traffic*: Based on the monthly visits of each website.

In practice, the adversary may have suspicions regarding the user's location. For example, a user living in the EU would probably not have a regular interest in reading the daily news from Brazil. Instead, she may frequently visit news websites from their own country or neighboring countries.

To simulate this, considering that $s$ is a Romanian site, we create the following priors:

- *Eastern*: Proportional to the country's population if the country is in the Eastern Bloc of EU, 0 otherwise
- *Ro-Slo*: Uniform if the site is hosted in either Romania or Slovakia, 0 otherwise.
- *Ro-Hu*: Uniform if the site is hosted in either Romania or Hungary, 0 otherwise.

*Baseline Methods:* We are about to compare our proposed solutions with the following natural approaches:

- *No Defense*: Setting $q = C_s$.
- *Average*: For each page size, calculate the average probability across all the other sites. Used only in the experiments for unknown prior.
- *Weighted* Average *on prior* $\pi$: Similar to *Average*, but now assign weights to each row based on the prior.
- *Copy*: Emulates another site, i.e. setting $q = C_{t'}$ for some site $t'$. If $\pi$ is known choose the $t'$ with the biggest $\pi_{t'}$. Otherwise, choose the $t'$ that offers the minimum capacity.
- *Pad*: Pad each page size deterministically to the next multiple of 5KB.

[9]The selected sites can be found in the Appendix D.

*Feasible Solutions:* As previously discussed, when searching for an optimal distribution $q$ of page sizes, we need to take into account that not all distributions are feasible in practice, which is expressed by enforcing feasibility constraints $\mathcal{F}$. In our use case, the constraints arise from the fact that page size can be easily increased via padding, but not decreased. For example, say that $q$ dictates that we should produce a page size of 5KB with probability 0.2. If all actual pages of our site are 10KB or larger, then producing a page of 5KB with non-zero probability is impossible.

To apply the optimization methods of previous sections we need to express $\mathcal{F}$ in terms of linear inequalities, which is done as follows. Let $\hat{q}$ be the site's *original* size distribution, without applying any defense. To express our non-negative padding constraint, we create a matrix $T_{o,o'}$, denoting the probability that we move from $o$ to $o'$ for all pairs $o, o' \in \mathbb{O}$. Then, a solution $q$ is feasible iff it can obtained from $\hat{q}$ via a transformation table $T$ that only moves probabilities from smaller to larger observables, which is expressed by the following linear constraints:

$$\begin{aligned} T_{o,o'} &\geq 0 && \forall o, o' \\ T_{o,o'} &= 0 && \forall o > o' \\ \hat{q}_o &= \textstyle\sum_{o'} T_{o,o'} && \forall o \\ q_o &= \textstyle\sum_{o'} T_{o',o} && \forall o \end{aligned}$$

This approach not only expresses the feasibility of $q$, but also provides the matrix $T$ which can be directly used as a padding strategy. When we receive a request for a page with size $o$, we can use the row $T_{o,\cdot}$ (after normalizing it), as a distribution to produce a padded page. The constraints guarantee that all sizes produces by that distribution will not be smaller than $o$.

### B. Experiments for a fixed prior $\pi$

This section evaluates the solutions discussed in Section IV.

Starting from the *s-distinguishing adversary*, recall that the weighted (by the prior) average of the other rows $q = Q^{\pi}_{\neg s} C$ is guaranteed to have no leakage (Prop. 1). In our experiment, however, this simple solution cannot be directly applied since it violates the feasibility constraints. Still, we can use the LP solution of Prop. 2 to obtain the optimal feasible solution. The results are shown in Table I; we can see that, although $q = Q^{\pi}_{\neg s} C$ itself is not feasible, we can actually find a feasible solution with no leakage at all in all cases, except from Ro-Hu in which the leakage is slightly larger than 1.

Moving to the *exact-guessing adversary*, we again use the LP of Prop. 2 to find the optimal solution (since the weighted

TABLE I: Leakages for each prior using Proposition 2 (known $\pi$, s-distinguishing adversary)

| Prior | Leakage |
|---|---|
| Uniform | 1 |
| Traffic | 1 |
| Eastern | 1 |
| Ro-Slo | 1 |
| Ro-Hu | 1.03 |

TABLE II: Leakage and posterior vulnerability (in parenthesis) for each method and prior (known $\pi$, exact-guessing adversary)

| Method | Uniform prior | Traffic prior | Eastern prior | Ro-Slo prior | Ro-Hu prior |
|---|---|---|---|---|---|
| Optimal (Prop. 2) | 8.78 (0.44) | 2.29 (0.61) | 1.01 (0.44) | 1.75 (0.58) | 1.04 (0.52) |
| No Defense | 9.16 (0.46) | 2.29 (0.61) | 1.69 (0.74) | 2.40 (0.80) | 1.76 (0.88) |
| Weighted Average | 9.02 (0.45) | 2.29 (0.62) | 1.21 (0.53) | 1.90 (0.63) | 1.04 (0.52) |
| Copy | 8.8 (0.44) | 2.29 (0.61) | 1.43 (0.63) | 1.78 (0.59) | 1.04 (0.52) |
| Pad | 9.26 (0.46) | 2.29 (0.61) | 1.86 (0.81) | 2.55 (0.85) | 1.89 (0.94) |

average $q = Q^{\pi}_{\neg s}C$ is neither feasible nor optimal). Table II shows the leakage for each prior and method, as well as the *posterior vulnerability* (in parenthesis) which is helpful to interpret the results. For instance, the optimal solution for the Ro-Slo prior gives a leakage of 1.75 and posterior vulnerability of 0.58, meaning that the adversary can guess the secret with probability $0.58/1.75 = 0.33$ a priori, and his success probability increases to $0.58$ after observing the output of the system. Note that this adversary is much harder to address (by controlling only a single row of the channel) than the s-distinguishing, hence it is impossible to completely eliminate leakage in most cases, but we can still hope for a substantial improvement compared to having no defense at all.

The results show that our approach offers the least leakage and the smallest posterior vulnerability across all priors. Among the other options, the natural choice of (projected) *Weighted Average* offers comparable results for some priors (Traffic, Ro-Hu) , but notably worse for others (for example $\approx 20\%$ more leakage on the Eastern prior and $\approx 10\%$ more leakage on the Ro-Slo prior).

*Copy* seems to be an interesting alternative for some priors but offers a solution with increased leakage on the case of the Eastern prior (1.43 over 1.01 of Proposition 2). This prior is the only one in which $s$ has a significantly larger probability (0.43) than any other site $t$; in this case, if we can make the evidence of any observation $o$ smaller than our a priori belief, that is $C_{t,o}/C_{s,o} \leq \pi_s/\pi_t$, then the rational choice would be to guess $s$ for any observation $o$, and the system would have no leakage at all. This is indeed achieved by the optimal solution. If we choose to *Copy*, however, the site $t$ with the largest (other than $s$) prior, we might not necessarily achieve this goal. This solution does make $s$ and $t$ indistinguishable, but we might now produce certain observations with very small probabilities, making $s$ and $t$ distinguishable from some of the the remaining sites, which explains why *Copy* performs worse than *Optimal* for this prior.

Another somewhat surprising observation is that padding turns out to be more *harmful* than no defense at all. The reason is that padding relies on every site using it simultaneously, so that different size observations become identical when mapped to the same padded value. However, if only $s$ pads, then its observations can become even more distinguishable than before, since only that site will always report sizes that are multiples of 5KB.

### C. Experiments for an unknown prior $\pi$

This section aims to assess the solutions presented in Section V.

*exact-guessing adversary:* We discussed that any convex combination of the remaining rows of $C$ will yield the same capacity, provided that the solution is feasible, since each column maximum is being retained. But a convex combination that complies with the constraint might not exist at all. To overcome the hurdle, we can use LP (Proposition 4).

Table III shows that the *Optimal (Prop. 4)* offers the best capacity of 8.78. In fact, any other method that is a convex combination of the remaining rows (i.e. *Average*, *Copy*) would have had the same result, but the constraints led to a slightly increased capacity. Observe that for the exact guessing adversary this capacity is the same as the leakage on the uniform prior (Table II), which is the worst prior (for the defender) for the exact guessing adversary, as discussed in Section II.

Nonetheless, Section V-B discussed that we could also use the solution of SEB (for the unconstrained case). Even though constraints do exist in this experiment, Table III shows that we get the same capacity as the *Optimal (Prop. 4)*. Interestingly, neither SEB nor Proposition 4 provide a solution that belongs to the convex hull of $C$.

TABLE III: Capacity for each method (unknown $\pi$, exact-guessing adversary)

| Method | Capacity |
|---|---|
| Optimal (Prop. 4) | 8.78 |
| SEB exact | 8.78 |
| No Defense | 9.14 |
| Average | 9.02 |
| Copy | 8.8 |
| Pad | 9.25 |

*s-distinguishing adversary:* Recall that we discussed why the problem reduces to SEB, offering two solutions, an exact but slow one (*SEB exact*) and a faster approximation (*SEB approx.*) resp. in Section VI-B and Section VI-C.

Table IV shows that *SEB exact* offers the best capacity: it decreases the capacity from $\approx 1.8$ (*No Defense*) to $1.56$. *Average* does show some improvement over *No Defense*, while *Pad* yields results that are too easily distinguishable. On the other hand, *SEB approx.* is technically the second-best choice, although the result is nearly identical to *Average*.

### D. Attack Simulation

In this section we simulate an actual attack, to estimate the attacker's accuracy, using the s-distinguishing adversary, as it is directly applicable in the WF scenario. In a real-world attack

TABLE IV: Capacity for each method (unknown $\pi$, s-distinguishing adversary)

| Method | Capacity |
|---|---|
| SEB exact | 1.56 |
| SEB approx. | 1.653 |
| No Defense | 1.79 |
| Average | 1.659 |
| Copy | 1.79 |
| Pad | 1.9 |

the defender will probably be oblivious for the attacker's $\pi$ and hence we use the solutions for the unknown prior.

We train a Random Forest classifier to estimate the target site among the others based on the observed page size, sampling pages from each site $t$ according to $C_t$. While this scenario involves an unknown prior, it is essential to establish one solely for the attacker, in order to decide how many pages to sample from each site. In other words, for each site $t$ the number of sampled pages depends on $\pi_t$. For $s$, we set $\pi_s = 0.5$ for all the following experiments, to capture the boolean nature of this adversary. Note that $\pi$ is only known to the adversary; it is not disclosed to the defender, who for that reason uses prior-agnostic methods.

In order to train the classifier for $s$, we begin by requesting page sizes according to $C_s$ and record the actually reported page sizes from the server (derived from $T$), which are then used in the training process. Essentially, this simulates the behavior of the server of $s$; a user requests a page (selecting not uniformly randomly, but according to $C_s$), then the server calculates the page size $i$, pads it according to $T_i$, and finally sends the padded page back to the user.

Finally, as per standard procedure, $80\%$ of the data is used for training and the remaining $20\%$ for testing.

Note that having a prior where $\pi_s > 0.5$ would essentially decrease the attacker's gains in information. In that case, the leakage of the system is not particularly interesting to the attacker, who already has enough information (as an example, consider the extreme $\pi_s = 1$).

Let us first examine the worst possible case, involving the highest possible posterior vulnerability, derived from the capacity. Intuitively, the worst case will be when the prior is evenly divided between two sites: $s$ and the one that differs the most from $s$.

For this omnipotent adversary equipped with the worst (for the defender) prior, Table V shows his accuracy [10]. The *No Defense* gives a $89\%$ accuracy while *SEB exact* offers an $11\%$ decrease in accuracy. All other methods offer at best only about half of that, with the best possible alternative to be *Average* with an accuracy of $83\%$. *Pad* is again worse than *No Defense*, offering a staggering $95\%$ accuracy to the attack.

TABLE V: Accuracy of the WF attack, for each method, when an s-distinguishing adversary possesses the worst (for the defender) possible $\pi$

| Method | Accuracy | Recall | F1 score |
|---|---|---|---|
| SEB exact | 0.78 | 0.8 | 0.8 |
| SEB approx. | 0.83 | 0.78 | 0.82 |
| No Defense | 0.89 | 0.88 | 0.89 |
| Average | 0.83 | 0.8 | 0.83 |
| Copy | 0.89 | 0.89 | 0.9 |
| Pad | 0.95 | 0.99 | 0.95 |

While in this first experiment we allocated the remaining $50\%$ of the prior to the most different site to capture the worst possible case, it will also be interesting to explore other ways to distribute it. We examine two cases in Figure 3 where we also include as a baseline the optimal solution for a particular prior (Proposition 2) but recall that it cannot be used in this scenario by the defender who does not have access to $\pi$.

First, we do a *1-on-1* comparison for each site $t \in C \setminus \{s\}$ by creating a prior $\mathbf{u}^{s,t}$ (i.e. we split the prior evenly between $s$ and $t$). Figure 1a shows that *SEB exact* offers worse accuracy for the attacker compared to the *Average* on every such 1-on-1 comparison. *Copy* is close to *No Defense* while *Pad* is again the worst option in every case; the classifier can easily distinguish a site that produces page sizes that are multiples of 5KB.

It will also be interesting to keep $\pi_s = 0.5$ and split the remaining $50\%$ to some other $n$ sites, not uniformly but according to their traffic, making a *1-on-n* comparison. Figure 1b shows that *SEB exact* performs better in the worst case. Note that when $n = 13$, the site with the most visits comes into play, notably affecting the prior [11].

On the other hand, Figure 1b shows that *Average* performs slightly better than *SEB exact* when $n$ is increased. To understand this, recall that in this scenario the optimal choice would be a Weighted Average based on the specific prior. But *Average* essentially assigns the same weight to each row of $C$, regardless of $n$, as it is prior-agnostic. However, as $n$ increases, more rows have a non-zero prior, favoring *Average* since its uniform weights happen to work well with this particular prior. Also, keep in mind that the same heuristics were used to sample each site as discussed in Section VII-A. This gives another advantage to *Average*; consider, informally, that it attempts to find the average of similar-looking items. In reality, the way a user explores a site might differ for each site, making it difficult to generalize the performance of *Average*, in contrast to *SEB exact* which we proved to be the best option for the worst possible prior.

### E. Comparison of SEB exact and SEB approx.

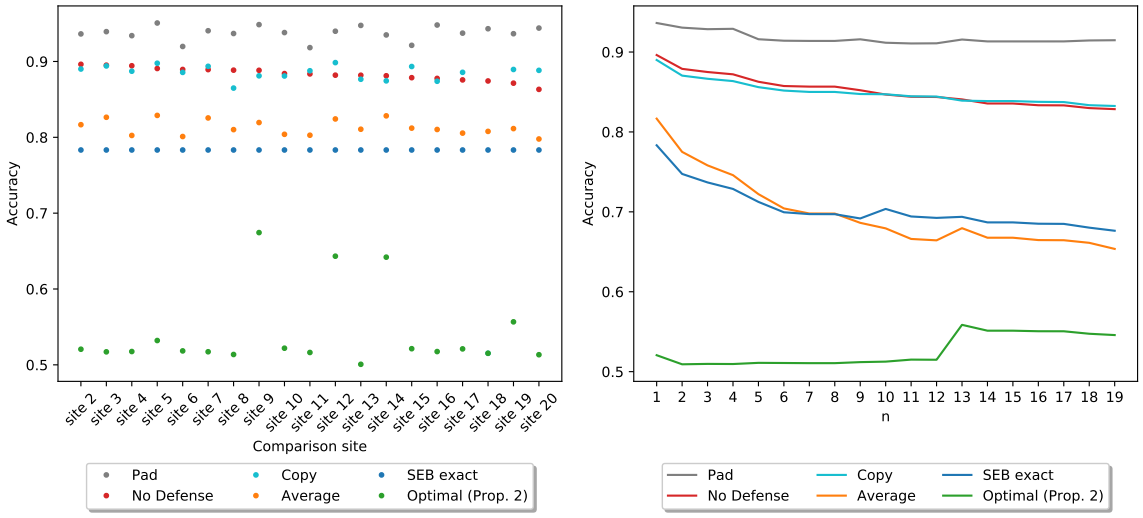
Previously, we discussed the trade-off between the two methods: the first offers optimal results at a high computational cost, while the latter provides approximated results quickly.

To illustrate the comparison, we conduct an experiment to measure the performance and runtime of the two methods as the size of the channel increases [12]. In this experiment, we

[10] Note that the accuracy could have also been calculated directly from Table IV by simply multiplying the capacity by $\pi_s = 0.5$, as the capacity captures the worst possible leakage.

[11] That is because the entire prior is normalized each time to ensure that its sum is 1.

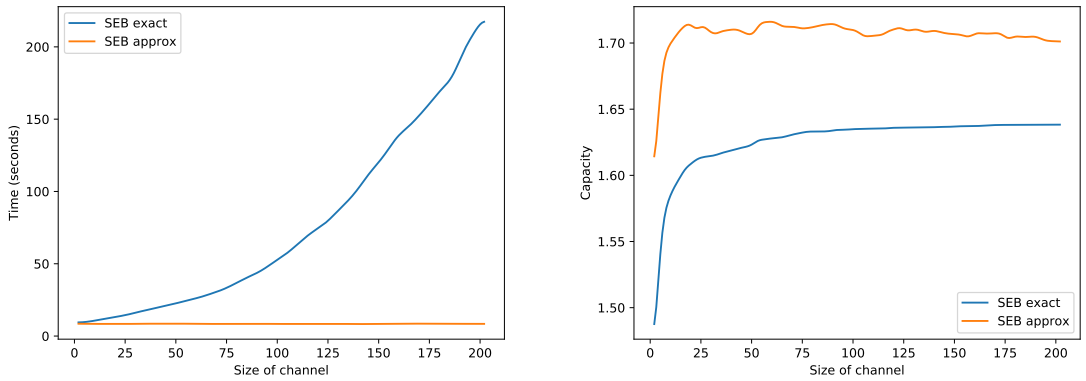
[12] The system specifications used in the experiments can be found in Appendix C.




(a) 1-on-1: prior is 0.5 on $s$ and 0.5 on another site

(b) 1-on-n: prior is 0.5 on $s$ and the remaining 0.5 is divided across $n$ sites according to their traffic

Fig. 1: Attacker's accuracy




(a) Runtime

(b) Capacity

Fig. 2: Comparison of *SEB exact* and *SEB approx.*

employ all 200 sampled sites, instead of only the 20 sites used in the previous experiments.

Figure 2a shows that *SEB approx.* always completes almost instantly. On the contrary, Figure 2b shows that as the channel size increases, *SEB approx.* achieves a capacity near 1.7, compared to 1.6 for *SEB exact*. On the other hand, the runtime of *SEB exact* scales polynomially, even taking more than 3 minutes to complete when all the 200 sites are featured.

For smaller channel sizes, such as the one used in the previous experiments, choosing *SEB exact* seems rather obvious; the runtime delay is insignificant while the improvement in capacity is remarkable.

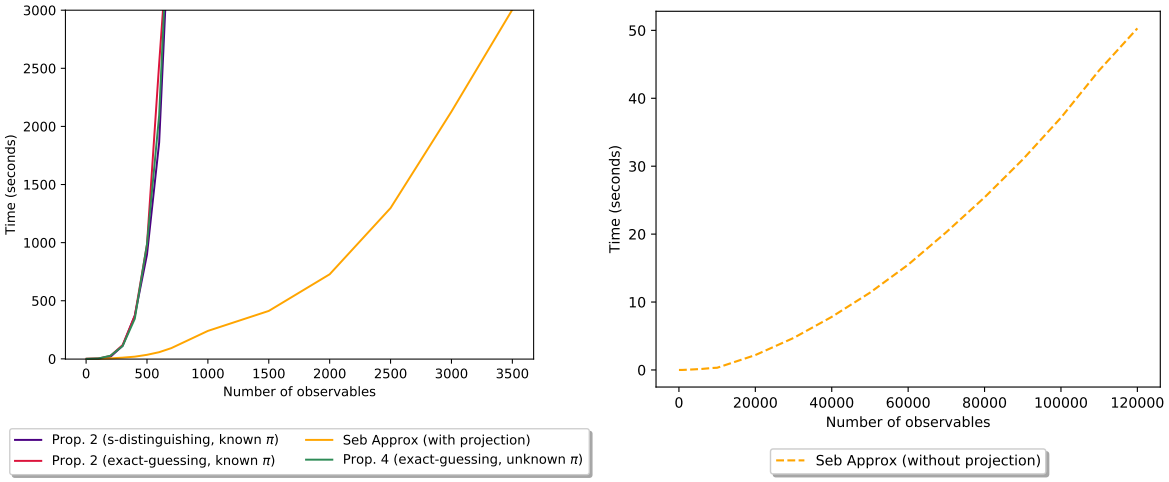
### *F. Scalability of Proposed Solutions*

After showing the better scalability of *SEB approx.* (compared to *SEB exact*), the next natural question is how the other proposed solutions scale. Indeed, this might be a primary concern for the defender if they wish to apply these solutions in a scenario with more observables.

Figure 3a shows the computation time of the corresponding LPs of Proposition 2, Proposition 4, and *SEB approx.*; the latter is scaling much better than the first two, which behave similarly.

Recall, however, that *SEB approx.* consists of two steps: a) finding the approximated solution and b) projecting it into $\mathcal{F}$ via LP. Intuitively, the second step is more costly, which raises the question about the performance of *SEB approx.* in the unconstrained case. Figure 3b shows that not computing the projection makes *SEB approx.* scale even better, as it computes a solution in less than a minute for even 120.000 observables.

This can be useful to the defender if they are completely free to design their own row (e.g. creating a website from




(a) Prop. 2, Prop. 4 and SEB approx.

(b) SEB approx. without computing the projection to $\mathcal{F}$

Fig. 3: Computation time of the proposed solutions as a function of observable attributes

scratch). Moreover, if the defender can sacrifice some level of protection, they can boost their performance by computing the unconstrained case (i.e. without performing the costly projection) and then produce a result by sampling iteratively from the distribution until the constraints are met. For instance, if the defender has a page that is 100KB, they can iteratively sample until they get a number larger than 100, since everything less than 100 should be discarded (assuming that the page cannot be further compressed).

## VIII. DISCUSSION AND CONCLUSIONS

In this work, we explored methods for adding a new row to an existing information channel under two distinct types of adversaries for both known and unknown priors.

When the prior is known, we discussed that the natural approach for an s-distinguishing adversary, namely an adversary that tries to distinguish if the secret is $s$ or not, falls short of achieving optimal results for the exact-guessing adversary (which tries to guess the exact secret in one try). In that case, LP can be used to provide an optimal solution.

We argued however that in real-world applications, the prior information available to an adversary is often difficult to estimate. Thus, when designing secure systems, one should also consider the worst-case scenario, or equivalently, in QIF terms, one should seek to minimize the capacity. Note that, when we use capacity for the comparison, the improvement is usually smaller than in the case of the leakage for a known prior. This is because the capacity represents the maximum leakage over all priors. Hence the capacities of the other approaches are "squeezed" between our capacity and the maximum possible capacity of a channel (system) with the same number of secrets and observables as the given one.

Therefore, guided by minimizing capacity, we showed that for an unknown prior, any convex combination of the remaining rows is sufficient for the exact-guessing adversary. However, for the s-distinguishing adversary, solving the SEB (Smallest Enclosing Ball) problem is necessary to find a solution with optimal capacity, although it requires polynomial time.

Furthermore, we explained how our techniques can be applied to defend against website fingerprinting, specifically by discussing how a site can pad its responses to comply with our proposed solutions. We conducted experiments demonstrating that our approach can significantly reduce the leakage, compared to other natural methods. Then, we simulated an actual attack by training an ML classifier for the s-distinguishing adversary and an unknown prior.

Our experiments confirm that solving the SEB problem ensures the lowest accuracy for the s-distinguishing adversary in the worst-case scenario compared to all the other prior-agnostic methods. Still the accuracy appears to be high because we considered an attacker who already knows that $s$ has a probability of $0.5$ to be the secret, in order to capture the capacity. This type of attacker is arguably uncommon in real-world situations. Although the abundance of information in today's information age makes it nearly impossible to know the attacker's prior knowledge, with our approach the defender can be prepared for the worst-case scenario.

We remark, however, that the results presented in Section VII depend on the assumptions taken during the design of the WF (Website Fingerprinting) attack simulation. The probability of each page at a given click-depth determines the behavior of each site. Informally, the more distinguishable the sites are, the harder it is to find a solution, and the benefit of the optimal solution, compared to other naive methods, may be reduced. Consider, for example, the extreme case where each website contains only a single page (e.g. appears with probability 1), which is unique in size. In this scenario, hiding s among a set of completely distinguishable sites becomes even more challenging.

Another practical aspect of the problem is the constraints faced when designing a defense against WF attacks. In this work, we considered only the page size, showing how linear

constraints can be used. These constraints can similarly be applied to the packet size, which has been shown to be the most valuable information for a WF attack [25]. However in real-world applications, one might want to include other parameters as well to increase the level of protection. Some natural choices are packet numbers, timing, and burst sizes, but an adversary can boost their WF attack by gathering information from other attributes, which can be as many as 35683 [25]. This plethora of attributes raises the question of whether all of them can be expressed via linear constraints. In cases where this is not possible, one potential approach would be to first find an unrestricted solution and then try to project it into the subspace of $L_1$ where the constraints are met. This approach was discussed in the "Feasible Solutions" paragraph of Section VII-A and we intend to explore it formally in future research.

Future work should also continue by measuring the efficiency of our approaches in complex real-world WF attacks, such as those studied in the literature for the Tor Network. Additional use cases could also be explored, as the scope of applications extends to any scenario in which a new user joins a fixed system and seeks privacy by designing their own responses based on the (fixed) responses of others. Finally, another potential research direction involves searching for a solution that is simultaneously optimal for both adversaries in the unknown prior setting while respecting any class of constraints.

## References


[1] *The Science of Quantitative Information Flow*, ser. Information Security and Cryptography. United States: Springer, Springer Nature, 2020.

[2] M. S. Alvim, K. Chatzikokolakis, C. Palamidessi, and G. Smith, "Measuring information leakage using generalized gain functions," in *Proceedings of the 25th IEEE Computer Security Foundations Symposium (CSF)*. IEEE, 2012, pp. 265–279. [Online]. Available: http://hal.inria.fr/hal-00734044/en

[3] S. Simon, C. Petrui, C. Pinzón, and C. Palamidessi, "Minimizing information leakage under padding constraints," 2022.

[4] A. C. Reed and M. K. Reiter, "Optimally hiding object sizes with constrained padding," 2021.

[5] M. H. R. Khouzani and P. Malacaria, "Leakage-minimal design: Universality, limitations, and applications," in *30th IEEE Computer Security Foundations Symposium, CSF 2017, Santa Barbara, CA, USA, August 21-25, 2017*. IEEE, 2017, pp. 305–317. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/CSF.2017.40

[6] M. Khouzani and P. Malacaria, "Optimal channel design: A game theoretical analysis," *Entropy*, vol. 20, no. 9, 2018. [Online]. Available: https://www.mdpi.com/1099-4300/20/9/675

[7] A. Americo, M. Khouzani, and P. Malacaria, "Deterministic channel design for minimum leakage," in *2019 IEEE 32nd Computer Security Foundations Symposium (CSF)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2019, pp. 428–42 813. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CSF.2019.00036

[8] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014. [Online]. Available: http://dx.doi.org/10.1561/0400000042

[9] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. (2010) What can we learn privately?

[10] G. Cherubin, "Bayes, not naïve: Security bounds on website fingerprinting defenses," *Proc. Priv. Enhancing Technol.*, vol. 2017, no. 4, pp. 215–231, 2017. [Online]. Available: https://doi.org/10.1515/popets-2017-0046

[11] K. Chatzikokolakis, G. Cherubin, C. Palamidessi, and C. Troncoso, "The bayes security measure," *CoRR*, vol. abs/2011.03396, 2020. [Online]. Available: https://arxiv.org/abs/2011.03396

[12] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. L. Boudec, "Protecting location privacy: optimal strategy against localization attacks," in *Proceedings of the 19th ACM Conference on Computer and Communications Security (CCS 2012)*, T. Yu, G. Danezis, and V. D. Gligor, Eds. ACM, 2012, pp. 617–627.

[13] R. Shokri, "Privacy games: Optimal user-centric data obfuscation," *Proceedings on Privacy Enhancing Technologies*, vol. 2015, no. 2, pp. 299–315, 2015.

[14] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Optimal geo-indistinguishable mechanisms for location privacy," in *Proceedings of the 21th ACM Conference on Computer and Communications Security (CCS 2014)*, 2014.

[15] G. Theodorakopoulos, R. Shokri, C. Troncoso, J. Hubaux, and J. L. Boudec, "Prolonging the hide-and-seek game: Optimal trajectory privacy for location-based services," *CoRR*, vol. abs/1409.1716, 2014. [Online]. Available: http://arxiv.org/abs/1409.1716

[16] K. Chatzikokolakis, G. Cherubin, C. Palamidessi, and C. Troncoso, "Bayes security: A not so average metric," in *36th IEEE Computer Security Foundations Symposium, CSF 2023, Dubrovnik, Croatia, July 10-14, 2023*. IEEE, 2023, pp. 388–406. [Online]. Available: https://doi.org/10.1109/CSF57540.2023.00011

[17] J. J. Sylvester, "A question in the geometry of situation," *Quarterly Journal of Pure and Applied Mathematics*, vol. 1, p. 79, 1857.

[18] D. Cheng, X. Hu, and C. Martin, "On the smallest enclosing balls," *Communications in Information and Systems*, vol. 6, 01 2006.

[19] K. Fischer, B. Gärtner, and M. Kutz, "Fast smallest-enclosing-ball computation in high dimensions," in *Proc. 11th European Symposium on Algorithms (ESA*. SpringerVerlag, 2003, pp. 630–641.

[20] H. Ding, "Minimum enclosing ball revisited: Stability and sub-linear time algorithms," *CoRR*, vol. abs/1904.03796, 2019. [Online]. Available: http://arxiv.org/abs/1904.03796

[21] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *In Proceedings of the 13 th Usenix Security Symposium*, 2004.

[22] G. Cherubin, R. Jansen, and C. Troncoso, "Online website fingerprinting: Evaluating website fingerprinting attacks on tor in the real world," in *31st USENIX Security Symposium (USENIX Security 22)*. Boston, MA: USENIX Association, Aug. 2022, pp. 753–770. [Online]. Available: https://www.usenix.org/conference/usenixsecurity22/presentation/cherubin

[23] M. A. I. Mohd Aminuddin, Z. F. Zaaba, A. Samsudin, F. Zaki, and N. B. Anuar, "The rise of website fingerprinting on tor: Analysis on techniques and assumptions," *Journal of Network and Computer Applications*, vol. 212, p. 103582, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804523000012

[24] G. Cherubin, J. Hayes, and M. Juarez, "Website fingerprinting defenses at the application layer," *Proceedings on Privacy Enhancing Technologies*, vol. 2, pp. 165–182, 2017.

[25] J. Yan and J. Kaur, "Feature selection for website fingerprinting," *Proceedings on Privacy Enhancing Technologies*, vol. 2018, pp. 200–219, 10 2018.


## Appendix

### A. Proofs of Section III

We start with two auxiliary lemmas.

**Lemma 2.** *For any $S, T \subseteq \mathbb{R}^n$, it holds that*

$$d_{\max}(\mathrm{ch}(S), \mathrm{ch}(T)) \;=\; d_{\max}(S, T)\ ,$$

*where distances are measured wrt any norm $\|\cdot\|$.*

*Proof.* Let $d = d_{\max}(S, T)$. Since $S \subseteq \mathrm{ch}(S)$ and $T \subseteq \mathrm{ch}(T)$ we clearly have $d \leq d_{\max}(\mathrm{ch}(S), \mathrm{ch}(T))$, the non-trivial part is to show that $d \geq d_{\max}(\mathrm{ch}(S), \mathrm{ch}(T))$.

We first show that

$$\forall s \in S, t \in \mathrm{ch}(T) : \|s - t\| \leq d\ . \tag{6}$$

Let $s \in S, t \in \mathrm{ch}(T)$ and denote by $B_d[s]$ the closed ball of radius $d$ centered at $s$. Since $d \geq \|s'-t'\|$ for all $s' \in S, t' \in T$ it holds that

$$B_d[s] \;\supseteq\; T \;,$$

and since balls induced by a norm are convex:

$$B_d[s] \;=\; \mathrm{ch}(B_d[s]) \;\supseteq\; \mathrm{ch}(T) \;,$$

which implies $\|s-t\| \leq d$, concluding the proof of (6).

Finally we show that

$$\forall s \in \mathrm{ch}(S), t \in \mathrm{ch}(T) : \|s-t\| \leq d \;.$$

Let $s \in \mathrm{ch}(S), t \in \mathrm{ch}(T)$, from (6) we know that $B_d[t] \supseteq S$, and since balls are convex we have that $B_d[t] = \mathrm{ch}(B_d[t]) \supseteq \mathrm{ch}(S)$, which implies $\|s-t\| \leq d$. □

Note that $\mathrm{diam}(S) = d_{\max}(S,S)$, hence Lem. 2 directly implies that $\mathrm{diam}(S) = \mathrm{diam}(\mathrm{ch}(S))$.

**Lemma 3.** *Let $C : \mathbb{S} \to \mathbb{O}$ be a binary channel, with $\mathbb{S} = \{s_1, s_2\}$.*

$$\mathcal{ML}_{g_{\mathrm{x}}}(C) \;=\; \mathcal{L}_{g_{\mathrm{x}}}(\mathbf{u}, C) \;=\; 1 + \frac{1}{2}\|C_{s_1} - C_{s_2}\|_1 \;,$$
$$\mathcal{ML}_{\ell_{\mathrm{x}}}(C) \;=\; \mathcal{L}_{\ell_{\mathrm{x}}}(\mathbf{u}, C) \;=\; \frac{1}{1 - \frac{1}{2}\|C_{s_1} - C_{s_2}\|_1} \;.$$

*Proof.* The result for $\mathcal{ML}_{\ell_{\mathrm{x}}}(C)$ follows directly from Thm. 2. Define

$$\Sigma_\top \;=\; \textstyle\sum_o \max_s C_{s,o} \;, \quad \Sigma_\bot \;=\; \sum_o \min_s C_{s,o} \;.$$

We obtain the following equalities:

$$\begin{aligned} \Sigma_\top &= \mathcal{ML}_{g_{\mathrm{x}}}(C) \;, && \text{(Thm. 1)} \\ \Sigma_\top + \Sigma_\bot &= 2 \;, && \text{(Rows sum to 1)} \\ \Sigma_\top - \Sigma_\bot &= \|C_{s_1} - C_{s_2}\|_1 \;. && \text{(Def. of } \|\cdot\|_1\text{)} \end{aligned}$$

Adding the last two and substituting in the first gives the required result. □

**Theorem 3.** *For any $C$ and $P \subseteq \mathbb{S}$ it holds that*

$$\mathcal{ML}_{g_P}(C) \;=\; \mathcal{L}_{g_P}(\mathbf{u}^{s,t}, C) \;=\; 1 + \frac{1}{2}\, d_{\max}(C_P, C_{\neg P}) \;,$$
$$\mathcal{ML}_{\ell_P}(C) \;=\; \mathcal{L}_{\ell_P}(\mathbf{u}^{s,t}, C) \;=\; \frac{1}{1 - \frac{1}{2}\, d_{\max}(C_P, C_{\neg P})} \;,$$

*for $s \in P, t \in \neg P$ realizing $d_{\max}(C_P, C_{\neg P})$.*

*Proof.* Starting from $\mathcal{ML}_{g_P}(C)$, let $\pi \in \mathcal{D}(\mathbb{S})$, define $\rho^\pi, Q^\pi$ as in (1),(2) and let

$$A \;=\; Q^\pi C \;.$$

Note that $A : \{P, \neg P\} \to \mathbb{O}$ is a binary channel; its rows $A_P$ and $A_{\neg P}$ express the behavior of an "average" (wrt $\pi$) secret of $C$ among those in $P$ and $\neg P$ respectively.

Note that $P$ represents a *set* of secrets of $C : \mathbb{S} \to \mathbb{O}$, but a *single* secret of $A : \{P, \neg P\} \to \mathbb{O}$. Hence, with a slight abuse of notation, $C_P$ denotes a *set of rows* of $C$, while $A_P$ denotes a *single row* of $A$.

We have that

$$\begin{aligned} & \mathcal{L}_{g_P}(\pi, C) \\ = \;& \mathcal{L}_{g_{\mathrm{x}}}(\rho^\pi, A) && \text{“Lem. 1”} \\ \leq \;& \mathcal{ML}_{g_{\mathrm{x}}}(A) && \text{“Def. of } \mathcal{ML}_{g_{\mathrm{x}}}\text{”} \\ = \;& 1 + \frac{1}{2}\|A_P - A_{\neg P}\|_1 && \text{“Lem. 3, } A \text{ is binary”} \end{aligned}$$

Notice that since $Q^\pi$ is a channel, the rows of $A$ are *convex combinations* of those of $C$. More precisely, since $Q^\pi_{w,s} = 0$ whenever $s \notin w$, the rows $A_P$ and $A_{\neg P}$ are convex combinations of the sets of rows $C_P$ and $C_{\neg P}$ respectively. Continuing the previous equational reasoning:

$$\begin{aligned} & 1 + \frac{1}{2}\|A_P - A_{\neg P}\|_1 \\ \leq \;& && \text{“}A_P \in \mathrm{ch}(C_P), A_{\neg P} \in \mathrm{ch}(C_{\neg P})\text{”} \\ & 1 + \frac{1}{2} d_{\max}(\mathrm{ch}(C_P), \mathrm{ch}(C_{\neg P})) \\ = \;& 1 + \frac{1}{2} d_{\max}(C_P, C_{\neg P}) && \text{“Lem. 2”} \\ = \;& && \text{“Let } s \in P, t \in \neg P \text{ be those realizing } d_{\max}(C_P, C_{\neg P})\text{”} \\ & 1 + \frac{1}{2}\|C_s - C_t\|_1 \;. \end{aligned}$$

This holds for all $\pi$, hence $\mathcal{ML}_{g_P}(C) \leq 1 + \frac{1}{2}\|C_s - C_t\|_1$. Taking $\pi = \mathbf{u}^{s,t}$ we get $\rho^\pi = \mathbf{u}$, $A_P = C_s$ and $A_{\neg P} = C_t$, hence

$$\mathcal{L}_{g_P}(\mathbf{u}^{s,t}, C) = 1 + \frac{1}{2}\|C_s - C_t\|_1 \;.$$

So the upper bound of $\mathcal{ML}_{g_P}(C)$ is attained, concluding the proof.

For $\mathcal{ML}_{\ell_P}(C)$ the proof is similar. □

### B. Proofs of Section V

**Proposition 3.** *For all $C$, any $q^\star \in \mathrm{ch}(C_{\neg s})$ minimizes capacity for exact-guessing adversaries, that is*

$$\mathcal{ML}_h(C^{q^\star}) \;\leq\; \mathcal{ML}_h(C^q) \quad \forall q \in \mathcal{D}(\mathbb{O}), h \in \{g_{\mathrm{x}}, \ell_{\mathrm{x}}\} \;.$$

*Moreover, it holds that $\mathcal{ML}_h(C^{q^\star}) = \mathcal{ML}_h(C_{\neg s})$.*

*Proof.* Starting from $\ell_{\mathrm{x}}$, note that $\mathrm{diam}(S) = d_{\max}(S,S)$. Hence Lem. 2 directly implies that $\mathrm{diam}(S) = \mathrm{diam}(\mathrm{ch}(S))$, that is taking convex combinations does not affect the diameter of a set. As a consequence $\mathrm{diam}(C^{q^\star}) = \mathrm{diam}(C_{\neg s})$. Then $\mathcal{ML}_{\ell_{\mathrm{x}}}(C^{q^\star}) = \mathcal{ML}_{\ell_{\mathrm{x}}}(C_{\neg s})$ follows from Thm. 2.

Similarly, for $g_{\mathrm{x}}$ the result follows from Thm. 1 and the fact that convex combinations do not affect the column maxima. □

Recall the notation $S^\bot, S^\top$ from (5). We also denote by $\preccurlyeq$ the partial order on $\mathbb{R}^n$ defied as $x \preccurlyeq y$ iff $x_i \leq y_i$ for all $i \in 1..n$.

**Lemma 4.** *Let $S \subseteq \mathcal{D}(\mathbb{O})$ and let*

$$q^\star \in \operatorname*{arg\,min}_{q \in \mathcal{D}(\mathbb{O})} d_{\max}(q, Q)$$

*be a solution to the $(\mathcal{D}(\mathbb{O}), L_1)$-SEB problem for $S$. Then*

$$S^\bot \;\preccurlyeq\; q^\star \;\preccurlyeq\; S^\top \;.$$

*Proof.* We show that $q^\star \preccurlyeq S^\top$, the proof of $S^\perp \preccurlyeq q^\star$ is similar. Assume that $q^\star_{o_1} > S^\top_{o_1}$ for some $o_1$, that is $q^\star_{o_1} > x_{o_1}$ for all $x \in S$. Select some

$$0 \;<\; \epsilon \;<\; \min_{\substack{o\in\mathbb{O}, x\in S \\ q^\star_o \neq x_o}} |q^\star_o - x_o|$$

and define $q \in \mathcal{D}(\mathbb{O})$ as

$$q_o \;=\; \begin{cases} q^\star_o - \epsilon & o = o_1 \\ q^\star_o + \frac{\epsilon}{|\mathbb{O}|-1} & \text{otherwise} \end{cases} .$$

In the following, we show that this construction moves $q^\star$ *strictly closer* to *all* elements $x \in S$ simultaneously.

Fix some arbitrary $x \in S$; the choice of $\epsilon$ is such that the relative order of $q^\star_o$ and $x_o$ is not affected by adding or subtracting $\epsilon$. Since $q^\star_{o_1} > x_{o_1}$ it follows that $q_{o_1} > x_{o_1}$ which it turn implies that in the $o_1$ component we moved $q^\star$ closer to $x$ by exactly $\epsilon$:

$$|q_{o_1} - x_{o_1}| \;=\; |q^\star_{o_1} - x_{o_1}| - \epsilon \; . \tag{7}$$

From the triangle inequality we get that in all other components, we moved $q^\star$ away from $x$ by at most $\frac{\epsilon}{|\mathbb{O}|-1}$:

$$|q_o - x_o| \;\leq\; |q^\star_o - x_o| + \tfrac{\epsilon}{|\mathbb{O}|-1} \; , \qquad \forall o \neq o_1 \; . \tag{8}$$

Moreover, since both vectors sum to 1 and $q^\star_{o_1} > x_{o_1}$ there must by some $o_2$ such that $q^\star_{o_2} < x_{o_2}$. By the choice of $\epsilon$ we get that $q_{o_2} < x_{o_2}$, hence:

$$|q_{o_2} - x_{o_2}| \;<\; |q^\star_{o_2} - x_{o_2}| + \tfrac{\epsilon}{|\mathbb{O}|-1} \; . \tag{9}$$

Summing over all $o$ using (7), (8) and (9) we get that

$$\|q - x\|_1 \;<\; \|q^\star - x\|_1 \; .$$

Since this happens for all $x \in S$, it contradicts the fact that $q^\star$ is a solution to the SEB problem, concluding the proof. □

**Theorem 4.** *For all $C : \mathbb{S} \to \mathbb{O}$, any distribution given by*

$$q^\star \;\in\; \arg\min_{q\in\mathcal{F}} \; d_{\max}(q, C_{\neg s})$$

*gives optimal capacity for s-distinguishing adversaries:*

$$\mathcal{ML}_h(C^{q^\star}) \;\leq\; \mathcal{ML}_h(C^q) \; , \quad h \in \{g_s, \ell_s\}, q \in \mathcal{F} \; .$$

*Moreover, if $\mathcal{F} = \mathcal{D}(\mathbb{O})$, then $q^\star$ is simultaneously optimal for exact-guessing adversaries, i.e. for $h \in \{g_{\mathrm{x}}, \ell_{\mathrm{x}}\}$.*

*Proof.* The minimization of $\mathcal{ML}_{g_s}, \mathcal{ML}_{\ell_s}$ is a direct consequence of Thm. 3, since both capacities are increasing functions of $d_{\max}(C^q_s, C^q_{\neg s}) = d_{\max}(q, C_{\neg s})$.

For $\mathcal{ML}_{\ell_{\mathrm{x}}}$, let $t \neq s$. Since $q = q^\star$ is a better choice than $q = C_t$ (and $q = C_t$ is feasible since $\mathcal{F} = \mathcal{D}(\mathbb{O})$) we have that:

$$d_{\max}(q^\star, C_{\neg s}) \;\leq\; d_{\max}(C_t, C_{\neg s}) \;\leq\; \mathrm{diam}(C_{\neg s}) \; . \tag{10}$$

As a consequence, for any $q \in \mathcal{D}(\mathbb{O})$ it holds that

$$\begin{aligned} & \mathrm{diam}(C^{q^\star}_{\mathbb{S}}) \\ = \;& \max\{\mathrm{diam}(C_{\neg s}), d_{\max}(q^\star, C_{\neg s})\} \\ = \;& \mathrm{diam}(C_{\neg s}) \\ \leq \;& \mathrm{diam}(C^q_{\mathbb{S}}) \; . \end{aligned}$$

The result follows from Thm. 2, since $\mathcal{ML}_{\ell_{\mathrm{x}}}$ is an increasing function of $\mathrm{diam}(C_{\mathbb{S}})$.

The last case $\mathcal{ML}_{g_{\mathrm{x}}}$. From Lem. 4 (which is applicable since $\mathcal{F} = \mathcal{D}(\mathbb{O})$) we get that the elements of $q^\star$ cannot be greater than the column maxima of $C_{\neg s}$. As a consequence, $C^{q^\star}$ and $C_{\neg s}$ have exactly the same column maxima, which from Thm. 1 implies that for any $q$:

$$\mathcal{ML}_{g_{\mathrm{x}}}(C^{q^\star}) \;=\; \mathcal{ML}_{g_{\mathrm{x}}}(C_{\neg s}) \;\leq\; \mathcal{ML}_{g_{\mathrm{x}}}(C^q) \; .$$

The last inequality comes from the fact that adding a row can only increase the column maxima. □

### C. System Specifications used in the experiments

The system specifications are crucial for determining the runtime of *SEB exact* and *SEB approx.* and the scalability of the proposed solutions.

We implemented the experiments in Python 3.6.9 using the qif library and run them in the following system:

- CPU: 2 Quad core Intel Xeon E5-2623
- GPU: NVIDIA GP102 GeForce GTX 1080 Ti
- RAM: 16x 16GB DDR4

### D. Selected Sites

In our experiments, out of the 200 sites we selected the 20 closest (in total variation distance) to the one we defend. The list is shown Table VI.

TABLE VI: Monthly Visits (in millions) of the selected sites (source: similarweb.com)

| Site Name | Visits |
|---|---|
| mediafax.ro ($s$) | 1.145 |
| wort.lu | 0.177 |
| sapo.pt | 13.6 |
| sabah.com.tr | 61.4 |
| lavanguardia.com | 41.5 |
| e24.no | 0.8 |
| cbc.na | 17.4 |
| news.com.au | 8.3 |
| cnnbrasil.com.br | 32.3 |
| slobodnadalmacija.hr | 1.3 |
| primicia.com.ve | 7.6 |
| sme.sk | 7.4 |
| topky.sk | 4.8 |
| oe24.at | 1.2 |
| ilfattoquotidiano.it | 12.7 |
| meinbezirk.at | 2.6 |
| voxeurop.eu | 0.019 |
| the-european-times.com | 0.004 |
| index.hu | 9.09 |
| hespress.com | 5.4 |