Enemy At the Gateways: Censorship-Resilient Proxy Distribution Using Game Theory

Milad Nasr
University of Massachusetts
Amherst
milad@cs.umass.edu

Sadegh Farhang
Pennsylvania State
University
farhang@ist.psu.edu

Amir Houmansadr
University of Massachusetts
Amherst
amir@cs.umass.edu

Jens Grossklags
Technical University
of Munich
jens.grossklags@in.tum.de

***Abstract*—A core technique used by popular proxy-based circumvention systems like Tor is to privately and selectively distribute the IP addresses of circumvention proxies among censored clients to keep them unknown to the censors. In Tor, for instance, such privately shared proxies are known as *bridges*. A key challenge to this mechanism is the *insider attack* problem: censoring agents can impersonate benign censored clients in order to learn (and then block) the privately shared circumvention proxies. To minimize the risks of the insider attack threat, in-the-wild circumvention systems like Tor use various *proxy assignment mechanisms* in order to minimize the risk of proxy enumeration by the censors, while providing access to a large fraction of censored clients.**

Unfortunately, existing proxy assignment mechanisms (like the one used by Tor) are based on ad hoc heuristics that offer no theoretical guarantees and are easily evaded in practice. In this paper, we take a systematic approach to the problem of proxy distribution in circumvention systems by establishing a game-theoretic framework. We model the proxy assignment problem as a game between circumvention system operators and the censors, and use game theory to derive the optimal strategies of each of the parties. Using our framework, we derive *the best (optimal)* proxy assignment mechanism of a circumvention system like Tor in the presence of *the strongest* censorship adversary who takes her best censorship actions.

We perform extensive simulations to evaluate our optimal proxy assignment algorithm under various adversarial and network settings. We show that the algorithm has superior performance compared to the state of the art, i.e., provides stronger resistance to censorship even against the strongest censorship adversary. Our study establishes a *generic framework* for optimal proxy assignment that can be applied to various types of circumvention systems and under various threat models. We conclude with lessons and recommendations for the design of proxy-based circumvention systems.

I. Introduction

The Internet plays a critical role in the open circulation of ideas and information across the world. Consequently, repressive regimes and totalitarian governments monitor and restrict their citizens' access to the Internet [46], [23], [27], [7], [12], [3], a phenomenon known as *Internet censorship*. The major techniques used by the censors to enforce censorship are IP address blacklisting, DNS interference, and deep-packet inspection [26], [50]. Practitioners and academics have designed and deployed various *censorship circumvention tools* to help censored users bypass censorship restrictions. Existing circumvention tools use techniques ranging from one-hop proxying [40], [2], [25], [37], [39] to more advanced mechanisms such as onion routing [8], [9], domain fronting [15], [18], [55], and decoy routing [20], [54], [36].

The Challenge of Proxy Distribution: The *core technique* used by the majority of in-the-wild circumvention tools, including Tor [9], Lantern [25], and Psiphon [40], is to run a number of circumvention *proxies* outside the censorship regions, and leverage these proxies to relay the traffic of censored users to censored Internet destinations. Unfortunately, once the censors learn the IP addresses of such proxies (e.g., by impersonating censored users and requesting proxies from circumvention operators) they can trivially block any access to the identified proxies, i.e., by IP blacklisting the identified proxy servers. Therefore, in-the-wild circumvention systems keep the IP addresses of their circumvention proxy servers private, and use various **proxy assignment mechanisms** to *strategically* distribute proxy information between the requesting clients (some of them could be censor-owned clients). For instance, Tor uses various mechanisms [47] to distribute its private proxies, known as *bridges* [8], among Tor clients. Therefore, the *goal* of a proxy assignment mechanism is to minimize the risks of proxy enumeration by the censors while providing proxy information to censored users.

Note that not all circumvention techniques require proxy assignment. For example, in domain fronting [15], a proxy server shares its IP address with other Internet services, making its censorship expensive to the censors. However, domain fronting is prohibitively expensive to be used for circumvention proxying at scale [32]. Therefore, recent proposals suggest to use domain fronting only for circumvention signaling, but not for proxying [44], [33], [25]. Other examples of circumvention systems that do not require proxy assignment include decoy routing [20], [54], [24], service tunneling [21], [22], [30], [6], and CDNBrowsing [18], [55]. All of these systems have remained *undeployed* in practice due to various

Network and Distributed Systems Security (NDSS) Symposium 2019
24-27 February 2019, San Diego, CA, USA
ISBN 1-891562-55-X
https://dx.doi.org/10.14722/ndss.2019.23xxx
www.ndss-symposium.org

practical issues such as low throughput or the need for adoption by major Internet operators. Therefore, *the proxy assignment problem continues to be the core challenge to all major in-the-wild circumvention systems* despite the recent advances in circumvention technologies.

Existing Approaches to Proxy Distribution. Designing effective proxy assignment[1] mechanisms is extremely challenging in practice: in-the-wild circumvention systems like Tor aim at serving the mass of censored Internet users with no restrictions, therefore they make their circumvention system and software available to the general public. Consequently, it is extremely difficult for such circumvention systems to reliably distinguish censoring agents (who request Tor for bridge proxies in order to block them) from genuinely censored clients, since the censoring agents reside in the same geographic region as censored clients and use the same mechanisms (e.g., email, software) to interact with circumvention systems. Existing proxy assignment mechanisms [47], [51], [29], [28] aim at reducing the disclosure of proxy information to the censors by deploying *heuristics-based techniques*. For instance, Tor restricts the number of bridge IPs shared with each censored client to three per request, and identifies different clients by their email addresses. Unfortunately, such heuristics-based assignment mechanisms are known to be easily defeatable in practice [53], [52], [10], e.g., the Chinese censors were able to enumerate *all* Tor bridges within a short period [53]. Applying more rigorous restrictions on proxy distribution will damage the usability of a circumvention system, i.e., censored users may not be able to obtain circumvention proxies. There also exist several academic mechanisms for proxy distribution [51], [29], [28]; such solutions are not deployed as they are designed for specific, impractical threat models. Additionally, such solutions are non-optimal as they are based on heuristics.

Our Approach. In this paper, we take a systematic, generic approach to the problem of proxy assignment by establishing a game-theoretic framework. We model the proxy assignment problem as a game between circumvention deployers and the censors, and use game theory to derive the optimal strategies of each of the parties. Specifically, we model the proxy distribution problem using a classic matching game called the *college admissions game* [16] whose goal is to admit students into colleges based on the rankings provided by the students and the colleges. We build a *proxy assignment game* by making an analogy between circumvention clients and students, as well as between proxies and colleges. We define various metrics based on the real-world constraints of circumvention systems to enable clients and proxies to rank each other in the proxy assignment game.

Using our framework, we derive *the best (optimal)* proxy assignment mechanism of a circumvention system like Tor in the presence of *the strongest, most strategic* censorship adversary who takes her best censorship actions. Therefore, the major contribution of our work is **deriving the optimal strategies of the censors and circumvention operators**, in contrast to prior work's heuristics-based proxy assignment mechanisms. We will demonstrate through simulations that our proxy assignment mechanism is significantly more resilient to censorship compared to prior solutions since we deploy optimal circumvention strategies as opposed to heuristics.

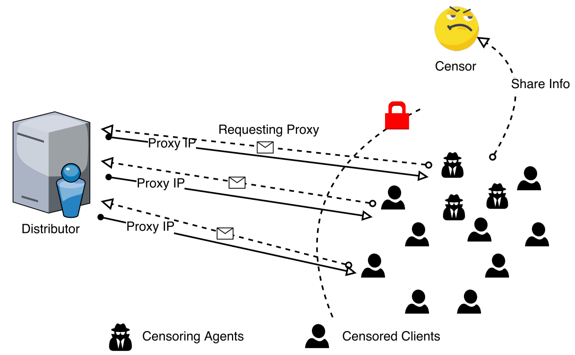

Fig. 1. Proxy distribution (assignment) in a circumvention system.

Moreover, our game-theoretic framework is *generic* and can be applied to *arbitrary* circumvention systems with various adversarial and network settings. This is in contrast to prior work [51], [29], [28] that consider only specific (and often unrealistic) adversarial and network settings. We perform extensive simulations to evaluate our optimal proxy assignment algorithm under a broad range of adversarial and network settings, and show that our optimal proxy assignment algorithm has superior performance compared to prior work, even against the strongest censorship adversary. We conclude with lessons learned for the design of proxy-based circumvention systems.

II. Preliminaries

A. Problem Statement: Proxy Distribution

Figure 1 illustrates the main setting of the proxy distribution problem, which consists of the following main entities: (1) the **Distributor** is a circumvention system operator who is in charge of distributing proxy information between clients. In the case of Tor, this entity is Tor's bridge distribution service. (2) The **Censored Clients** are the Internet users living within the censored regions. The censored clients request proxy information from the distributor in order to be able to use the underlying circumvention system and bypass censorship. (3) The **Censoring Agents** are censor-controlled entities who impersonate censored clients in order to obtain proxy information from the distributor. (4) The **Censor** is a central censorship authority who controls and commands all the censoring agents. She collects and combines the proxy information learned by its censoring agents and uses the collected information to block the circumvention system or identify circumvention clients.

In this setting, the objective of the distributor party is to distribute proxy information between clients (i.e., an undisclosed mixture of censored clients and censoring agents) in a way that maximizes the performance of the circumvention system for the censored clients. On the other hand, the censor's objective is to degrade the performance of the circumvention system by learning and (strategically) blocking circumvention proxies.

Significance of Proxy Distribution: Proxy-based circumvention continues to be the main technique used by major in-the-wild circumvention system like Tor, Psiphon, Lantern, and VPN services. As discussed in Section I, domain-fronted [15]

[1] We use proxy "distribution" and "assignment" interchangeably.

proxies do not require proxy assignment, however, such proxies cannot be used for large-scale proxying due to their prohibitive expenses [32] (instead, they are suggested to be used for circumvention signaling [44], [33], [25]). Decoy routing [20], [54], [24], service tunneling [21], [22], [30], [6], and CDNBrowsing [18], [55] are other circumvention approaches that do not require proxy distribution; however, such techniques are *not deployed* in-the-wild due to various practical issues such as low throughput or the need for adoption by Internet operators. To summarize, *the proxy assignment problem is a key challenge to all major in-the-wild circumvention systems* despite the recent advances in circumvention technologies.

B. Existing Proxy Distribution Mechanisms

Deployed Techniques. In-the-wild circumvention systems use *ad hoc heuristics* for proxy distribution. The Tor project offers three mechanisms to clients to obtain Tor bridge IP addresses[2]: (1) a user's Tor client software can directly obtain bridge IPs from Tor servers, (2) a user can visit Tor's bridge distribution webpage [5] and obtain bridge IPs after solving a CAPTCHA, and (3) a user can send an email to `bridges@bridges.torproject.org` from Gmail, Yahoo!, or Riseup! to receive bridge IPs via an email response. To prevent censors from bridge enumeration, Tor limits the number of proxy IPs returned to each client to three for specific time intervals. Tor distinguishes users based on their IP addresses and email addresses. Other in-the-wild circumvention systems use similar heuristics for proxy distribution. For instance, Psiphon [40] and Lantern [25] include proxy information in each software download.

Unfortunately, such CAPTCHA- and identity-based protection mechanisms are ineffective against a resourceful censor, i.e., one who can create large numbers of email accounts and connect from a diverse set of IP addresses, or use human-based CAPTCHA solving platforms. Previous work [14], [53] provides evidence that such approaches have limited impact against resourceful censors; this is confirmed by the fact that the Chinese censors were able to enumerate all Tor bridges over the course of a single month [53], [52], [10].

Strategic Distribution Proposals. Feamster et al. [14] propose a mechanism for proxy distribution that uses computational puzzles to prevent corrupt users from enumerating a large number of proxies. However, they demonstrate that this mechanism is not effective against a resourceful adversary who can do extensive computations, e.g., the adversary can discover 95% of 100,000 proxies by solving only 300,000 puzzles.

Sovran et al. [45] distribute proxy information among a small number of highly trusted users, known as forwarders, who will relay the traffic from other clients to the circumvention proxies. The addresses of such forwarders are distributed among clients through a random walk on social networks. The technique is not practical due to the churn of forwarders, the extreme bandwidth and computation overhead imposed on forwarders, and the real-world risks to forwarders due to facilitating circumvention.

Mahdian [28] takes an algorithmic approach to proxy distribution by deriving the lower bound on the number of required proxies given a known number of censoring agents. The scheme, however, is not practical, due to its various unrealistic (simplifying) assumptions, e.g., by assuming no limit on the capacity of proxies, and assuming the number of censoring agents to be known to the distributor.

Proximax [29] leverages social networks for proxy distribution. It identifies the most efficient social network channels for proxy distribution in order to maximize the overall usage of all proxies. rBridge [51] is the state-of-the-art proxy distribution system that outperforms prior systems like Proximax. rBridge uses client reputations to distribute bridge information among the clients. Each client will collect reputation credits over time based on the uptime of the bridges she has been provided with. A user will have to spend her collected reputation credits to request new proxies. Both Proximax and rBridge use heuristics to design proxy distribution mechanisms. Also, they both consider a non-optimal censoring adversary who does not use her best censorship strategy against the distributor.

C. Sketch of Our Approach

Existing proxy distribution mechanisms, reviewed above, suffer in two ways; First, they deploy heuristics-based mechanisms for proxy distribution that are non-optimal against censors with varying resources. For instance, the use of computational client puzzles is known to be ineffective against resourceful adversaries, and the state-of-the-art rBridge uses only a single feature (i.e., proxy uptimes) to strategically determine proxy distribution. Second, prior schemes consider simple and limited models for the censoring adversaries, which limit the applicability of derived proxy distribution schemes to in-the-wild systems. Even the state-of-the-art scheme of rBridge considers only 3 simple, specific censorship strategies, and they assume that the censoring agents act independently in blocking proxies.

In this paper, we take a systematic, generic approach to the problem of proxy assignment by establishing a game-theoretic framework. As opposed to designing heuristics-based proxy distribution mechanisms against simple, specific censoring adversaries, we derive *the best (optimal)* proxy assignment mechanism of a circumvention system like Tor in the presence of *the strongest, most strategic* censorship adversary who takes its best censorship actions. Our use of game theory allows us to incorporate *arbitrary features* into the design of proxy distribution schemes, therefore allowing it to get applied to diverse circumvention systems with various adversarial and network settings. This is in contrast to rBridge's use of a single feature, i.e., proxy uptime, in strategically determining proxy distribution. As demonstrated through simulations, our proxy assignment mechanism is more resilient to censorship compared to prior solutions since it is based on optimal circumvention strategies.

Note that game theory has been studied in various circumvention contexts. For example, Elahi et al. [13] use game theory to optimize traffic obfuscation in circumvention systems, and Nasr et al. [35] use game theory to optimize decoy placement in decoy routing circumvention systems. However, we are the first to apply game theory to the problem of proxy distribution.

[2] https://www.torproject.org/docs/bridges

D. Threat Model and Scope of Our Work

Like rBridge [51], we assume that the clients (including both censored clients and censoring agents) have no way of obtaining the proxy information other than requesting it from the distributor entity. Therefore, we do not consider "network-level" proxy identification attacks by the censors such as traffic fingerprinting attacks [19], [17] and probing attacks [10], [53], [52]. We emphasize that no proxy distribution mechanism will function in the presence of such network-level attacks, however, defending against such attacks is *orthogonal to our work*. Prior work has designed various traffic obfuscation techniques [38], [21], [11], [34] and client authentication mechanisms [43], [48] to defend against network-level proxy identification attacks.

Unlike the simplifying, unrealistic assumption made in prior works [51], [29], [28], we assume that the censoring agents are able to communicate among themselves and the central censorship authority makes the censorship decisions by aggregating collected information from all of its censoring agents. Therefore, the censor can strategically allow an identified proxy to remain unblocked for a while, or block it instantly depending on the importance of the discovered proxy as well as the reputation of the censoring agent who discovered that bridge.

Also, note that we do *not* attempt to design a Sybil defense mechanism. Resisting Sybil attacks is orthogonal to our work and is an active area of related research projects [14], [4]. Our goal is to, for a given fraction of Sybil clients (i.e., a given ratio of censoring agents to censored clients), derive the optimal proxy distribution mechanism that optimizes the utility of the circumvention proxies for censored clients.

Paper's Outline: We introduce the college admissions game in Section III, which is the core game-theoretic model we use. In Section IV, we present our generic game-theoretic framework for proxy distribution. In Section V, we show how our model can be applied to a specific circumvention system. We simulate and evaluate our optimal proxy distribution mechanism in Sections VI and VII, and conclude the paper in Section VIII with discussions and recommendations.

III. Building Blocks: College Admissions Game

In this section, we provide an overview of the *college admissions game* framework by Gale and Shapley [16], which is the foundation for our game-theoretic model. The college admissions game is also referred to as *deferred acceptance* algorithm, which we will use interchangeably throughout the paper. First, we describe the assignment criteria in the college admissions game. Second, we describe the college admissions algorithm and its characteristics.

A. The Assignment Criteria

In the college admissions game, there are n students and m colleges. Each college has a capacity of q_i students to admit. Every student ranks the colleges she is considering to attend based on her order of preferences (that is, a student only ranks the colleges that she has "applied to"). On the other hand, each college ranks the students who have applied to that school based on the college's preferences. The college first eliminates the students who will not be admitted under any circumstances even if the college does not reach its capacity. The college admissions algorithm then makes admission decisions for all colleges based on the rankings provided by colleges and students, as well as the capacity of the colleges. We are particularly interested in *stable* assignments, as defined below.

Definition 1. *An assignment of students to colleges is called **unstable** if there exist two students 1 and 2 who are assigned to colleges a and b, respectively, however, student 2 prefers college a to b and college a prefers student 2 to 1 [16]. Otherwise, we have a **stable** assignment.*

Note that it is possible that different stable assignments exist. Then, the question is how to choose among different stable assignments; ideally the optimal one. The following gives the definition of an *optimal* assignment in the college admissions game.

Definition 2. *A stable assignment is **optimal** if every student is just as well off under it as under any other stable assignment [16].*

It is worth mentioning that the optimal stable algorithm is *unique*. In other words, there exists (if it exists at all) only one assignment that is optimal and stable. Moreover, in the above definition, the assignment is optimal from the students' point of view. In the following subsection, we provide an overview of an algorithm that preserves these two features.

B. Deferred Acceptance Algorithm

In this subsection, we describe the *deferred acceptance* algorithm [16]. As mentioned in Section III-A, there exist some students that a college will not admit under any circumstances. Here, in the deferred acceptance algorithm, the assumption is made that these students are not allowed to apply for that college. By considering this assumption, the algorithm is as follows. First, all students apply to their top-choice colleges. A college with capacity of q students places the q students on the waiting list with the highest rank, or all the students who have applied if there are less than q applicants. The college rejects the remainder of the students. Then, the rejected students apply to their second choices and so on. In a similar way, each college selects the top q students from the students on its waiting list from a previous round and the new students who have applied to this college. The college chooses the top q students and rejects the rest. This procedure terminates once every student is on the waiting list of some college or has been rejected by all of the colleges he was permitted to apply to. Finally, each college admits the students on its waiting list.

The following two theorems describe the characteristics of the deferred acceptance algorithm.

Theorem 1. *There always exists a stable assignment in the deferred acceptance algorithm [16].*

Proof: See Appendix A.

Theorem 2. *Every student is at least as well off under the assignment given by the deferred acceptance algorithm as he would be under any other stable assignment [16].*

Proof: See Appendix B.

Why this game fits our problem. We use the college admissions game to establish an optimal mechanism to assign proxies to clients; we call our mechanism the *proxy assignment game*. On a conceptual level, we model the proxy assignment problem by using the college admission game as a foundation as follows: the clients (including the censoring agents) act as the students who are interested in learning the addresses of the proxies, and the proxies act as colleges as each of them has a finite (known) capacity for serving clients. Solving this game results in the assignment of proxies to clients.

It is important to note that the theory and concept of the college admissions game and the deferred acceptance algorithm is successfully applied in practice, and has stood the test of time [41]. Even predating the publication of the seminal paper by Gale and Shapley is the National Resident Matching Program used to place United States medical school students into residency training programs; today, over 40,000 applicants and 30,000 positions are part of the program on an annual basis. At the same time, over 60 matching programs for medical subspecialties follow similar processes [41]. More recently, the matching approach has also been used in the scenario of school choice in the New York and Boston public school systems [1]. Further, key insights of the general concept are used in diverse scenarios such as kidney exchanges [42]. Finally, the 2012 Nobel Memorial Prize in Economic Sciences was awarded to Roth and Shapley "for the theory of stable allocations and the practice of market design;" thereby validating the profound practical and theoretical impact of the work.

IV. Our Proxy Assignment Game

In this section, we introduce a generic framework for the proxy assignment problem by making an analogy with the college admissions game, introduced above. To do so, we model the *clients* (including censored clients and censoring agents) as the students in the college admissions game, and we model the *proxies* as the colleges, where a proxy's finite circumvention capacity is equivalent to a college's admission capacity. Solving this game results in the assignment of one (or no) proxies to each client. We run the game multiple times in a sequence if the distributor aims at assigning multiple proxies to each client. We will define utility functions for the clients and proxies for them to be able to rank each other. *While our game is generic and can be applied to various proxy-based circumvention systems, the utility functions will need to be tailored to the specific threat models and settings of specific circumvention systems*, as will be discussed later. Finally, we will use the deferred acceptance algorithm, introduced in Section III, to find the stable associations between users and proxies, i.e., the Nash equilibrium. This will enable our model to derive the optimal strategies for the censors and circumvention operators.

How Our Algorithm Is Deployed in Practice Note that, while we present our proxy assignment algorithm as a game between proxies, benign clients, and censor-controlled clients, in practice it is solely run by the circumvention system's proxy distribution system (e.g., by Tor's proxy distribution system) with no need for any cooperation from proxies/clients in running the algorithm. That is, our game is **virtual**: the circumvention distributor plays the game on behalf of all of the clients and proxies, and the central censorship authority makes decisions for all of the censoring agents. In order to run the game and make the assignments, the distributor does *not* need to know any prior information about individual clients, like their types (e.g., censoring agent or censored client) or their proxy preferences—in fact, typical real-world proxies have no particular preference between genuinely censored clients, and typical real-world censored clients have no preference between working proxies. The distributor derives the rankings of each client using utility metrics that are derived from the client's behavior and observable features. Finally, note that our algorithm uses equilibrium points *only to find* the optimal proxy assignments, however, a stable matching is not required for a real-world proxy distribution system.

A. The Game Setup

We start by presenting the main model of the proxy assignment game. Please refer to Table I for all the notations.

Players. In our model, there are n users (clients) $\mathbb{A} = \{a_1, a_2, ..., a_n\}$ who request proxies for circumvention. Therefore, a_i is the ID of the ith client. Among these n users, m of them are *censoring agents*, denoted by $\mathbb{J} = \{j_1, j_2, ..., j_m\}$, while the rest are genuine censored clients. All of the m censoring agents are controlled by a central censoring authority (e.g., the GFW) and take actions as instructed by the central censoring authority. We also consider l circumvention proxies in the system denoted by a set $\mathbb{P} = \{p_1, p_2, ..., p_l\}$. Per each request, a client is provided with the information (i.e., IP addresses) of k proxies by the distributor.

We divide the time dimension into intervals called *stages* denoted by t (the game starts at $t = 0$). Without loss of generality, in our model all the actions by the players, such as asking for new proxies, blocking proxies, providing proxies, etc., are performed at the end of the stages, not during the stages.

Scoring functions. A client will use a scoring (utility) function to score (and then rank) each of the proxies. We use $U^t_{a_i}(p_x)$ to indicate the score that user a_i gives to proxy p_x at stage t. Similarly, each proxy will use a utility function to score the clients, i.e., $U^t_{p_x}(a_i)$ is proxy p_x's score for the client a_i at stage t. As will be discussed later, the utility functions should be tailored to specific threat models and settings of different circumvention systems. However, that does not impact our game's general setting, presented here, nor its solution.

B. Ranking Mechanism

Each proxy (client) uses a preference relation $\succeq_{p_x}$ ($\succeq_{a_i}$) to rank clients (proxies). This relation is a binary relation that is complete, reflexive, and transitive [16]. By using these preference relations, proxies and users can rank each other. A proxy $p_x \in \mathbb{P}$ will rank all users making requests at stage t. In doing so, for any two users $a_i, a_{i'} \in \mathbb{A}$ ($i \neq i'$), we define the following preference relation for the proxy $p_x \in \mathbb{P}$:

$$a_i \succeq_{p_x} a_{i'} \;\Rightarrow\; U^t_{p_x}(a_i) \geq U^t_{p_x}(a_{i'}), \tag{1}$$

where $U^t_{p_x}(a_i)$ is user a_i's score by proxy p_x, as defined earlier. The user with the highest utility is the most preferred user for proxy p_x.

Similarly, each user $a_i \in \mathbb{A}$ uses the following preference relation $\succeq_{a_i}$ to rank proxies $p_x, p_{x'} \in \mathbb{P}$ $(x \neq x')$:

$$p_x \succeq_{a_i} p_{x'} \;\Rightarrow\; U^t_{a_i}(p_x) \geq U^t_{a_i}(p_{x'}), \tag{2}$$

where $U^t_{a_i}(p_x)$ is the score that user a_i give to proxy p_x.

Note that in the college admissions game, it is desirable to have strict preferences (denoted by $\succ$). Here, we assume that when a player is indifferent between two choices, then that player ranks these two choices randomly, e.g., by tossing a coin. Furthermore, in a college admissions game, each college can have a threshold for accepting new students. Here, we set a global threshold for all proxies. We define that the utility of each user should be at least more than η to be able to request a new proxy.

C. The Optimal Proxy Distribution Mechanism

Recall that our game is "virtual," therefore, the distributor plays the game on behalf of all clients and proxies. Note that, to do so, the distributor does *not* need to know the type of each client, nor their proxy preferences. Instead, the distributor evaluates the scoring functions and ranks parties (as described above) on behalf of the clients and proxies. Therefore, a (censoring or censored) client will petition the distributor for access to *some* proxy, and the rankings and assignments are performed locally by the distributor based on the observable features of the requesting clients.

At the end of each stage, t, the distributor plays the game on behalf of users and proxies by calculating the utility (scores) of all the proxies and all clients who have requested new proxies at stage t. Then, the distributor will use the deferred acceptance algorithm introduced earlier to assign proxies to the clients requesting new proxy addresses. Algorithm 1 summarizes the *optimal proxy assignment mechanism* used by the distributor to assign proxies to the requesting clients. If the distributor aims at assigning $k > 1$ proxies to each client, she will run the algorithm k times in a consecutive fashion, each time removing the proxies assigned to a client in the previous rounds from the client's ranked list.

Algorithm 1 Optimal proxy distribution.

```
Each new client is initially provided with $k$ arbitrary proxies
Clients who need proxies request new proxies from the distributor
For each client, the distributor builds a preference list based on (1). For each client, the distributor removes the proxies previously assigned to that client from her preference list.
For each proxy, the distributor builds a preference list based on (2) (each proxy rejects the users whose utilities are less than a threshold)
The distributor runs the deferred acceptance algorithm (Section III-B) to derive the stable assignment
The distributor assigns proxies to clients
```

Note that based on Theorem 1, the deferred acceptance algorithm provides a "stable" assignment of proxies to clients. Specifically, the deferred acceptance algorithm guarantees that in the resulting assignment at the end of each stage, there is no user, regardless of his type, who prefers another proxy where that proxy also prefers that user. Also, according to Theorem 2, the resulting assignment is "optimal" from the clients' perspective. Therefore, assuming the censored clients to be rational, none of them will have incentives to use the proxies assigned to other clients. Similarly, while the censoring agents can share the proxies they have obtained, they will have no incentive to do so as they have obtained their optimal proxies.

D. The Optimal Censorship Strategy

As discussed earlier, previous proxy distribution proposals evaluate against an unrealistically weak censoring adversary who takes heuristics-based censorship actions, as opposed to its optimal strategy. For instance the state-of-the-art rBridge considers only three ad hoc censorship strategies, and even assumes the censoring agents to act independently from each other. By contrast, we evaluate against the most strategic censor who implements optimal censorship strategies.

To decide her optimal strategy, the central censoring authority takes actions that maximize her *utility function*, U^t_C. The censor's utility function has two components:

- *Proxy discovery:* This represents the censor's capability in discovering circumvention proxies; quantified as $\sum_{a_i \in \mathbb{J}} U^t_C(a_i)$, where $\mathbb{J}$ is the set of censoring agents and $U^t_C(a_i)$ is the utility of the ith censoring agent (as defined in the following). A larger $U^t_C(a_i)$ increases a_i's chances of obtaining new proxies from the distributor.
- *Blocking impact:* This presents the censor's impact on blocking clients. We quantify this with the fraction of censored clients who are not able to obtain a working bridge from the distributor, $r_{Blocked}$.

Note that "proxy discovery" and "blocking impact" do not necessarily imply each other. A censor may leave a discovered proxy unblocked to identify and surveil clients. Therefore, we present the censor's utility function as:

$$U^t_C = \omega \sum_{a_i \in \mathbb{J}} U^t_C(a_i) + r_{Blocked} \tag{3}$$

where the ω coefficient shows the censor's relative preference regarding the two components.

We quantify $U^t_C(a_i)$, the utility of censoring agent a_i, with the average of a_i's scores by proxies, i.e., $\mathbb{E}_{p_x \in \mathbb{P}}[U^t_{p_x}(a_i)]$. Note that in practice, the censor can only compute $U^t_{p_x}(a_i)$ for the proxies she has discovered, therefore our model simulates the strongest possible adversary. If a censoring agent's utility falls below the acceptance threshold (η), she will not receive any proxies from the distributor (will be identified as a censoring agent by the distributor). This will cost the censor by losing that censoring agent, and we quantify it with ν, which is the cost of replacing a lost censoring agent. Therefore, we quantify $U^t_C(a_i)$ as:

$$U^t_C(a_i) = \begin{cases} \mathbb{E}_{p_x \in \mathbb{P}}[U^t_{p_x}(a_i)] & \mathbb{E}[U^t_{p_x}(a_i)] \geq \eta \\ -\nu & o.w. \end{cases} \tag{4}$$

TABLE I. NOTATIONS

Variable	Definition
t	Stage of the game
$\mathbb{A}$ ($\mathbb{P}$)	Set of all users (proxies)
$\mathbb{J}$ (m)	Set (Number) of censoring agents
n (l)	Number of users (proxies)
$U^t_{a_i}(p_x)$	Score that user a_i gives to proxy p_x at stage t
$U^t_{p_x}(a_i)$	Proxy p_x's score for the client a_i at stage t
$U^t_C(a_i)$	Utility of the ith censoring agent
$r_{Blocked}$	Fraction of circumvention clients who are not able to obtain a working bridge from the distributor
ν	Cost of losing a censoring agent
ω	Weighting factor in (3)
η	Acceptance utility threshold for new proxy request
$\gamma^t_{a_i}$	Blocked proxy usage
$T^t_{a_i,p_x}$	The time duration that a_i utilized proxy p_x
T_{a_i}	Proxy utilization
$R^t_{a_i}$	Number of requests for new proxy addresses
$\delta^t_{a_i}$	Number of blocked proxies that a user knows
d_{a_i,p_x}	Client locations
$B^t_{p_x}$	Number of users having the address of proxy p_i at stage t
$c^t_{p_x}$	The number of users connected to proxy p_x at stage t
$\tau^t_{p_x}$	Total time utilization of the proxy
$\mathbb{B}^t_{p_x}$	The set of user IDs who know proxy p_x
$(\beta_1, \beta_2, \beta_3, \beta_4)$	Weighting factors in (6)
$(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5)$	Weighting factors in (7)
$\overline{T}$	Maximum time a user can use proxy for utility improvement
μ_b, μ_s	Rate of new clients in birth interval (stable interval)
λ_b, λ_s	Rate of new proxies in birth interval (stable interval)
ρ	Ratio of the censoring agents to the total number of clients
p	Probability of blocking a proxy for a conservative censor

To summarize, the censor derives her optimal strategy by taking actions that maximize her utility function in (3).

V. CASE STUDY: DERIVING UTILITY FUNCTIONS

Our proxy assignment game in Section IV is a *generic* framework for deriving optimal proxy distribution mechanisms for proxy-based circumvention systems. In order to use this framework for a specific circumvention system, one needs to derive tailored utility functions (i.e., $U_{a_i}(p_x)$ and $U_{p_x}(a_i)$) based on the specific threat model and assumptions of the target circumvention system.

In this section, we demonstrate how such tailored utility functions can be derived for a hypothetical circumvention system, which we call HypoTor. Note that in this paper we *do not* aim at deriving the tailored utility functions for a real-world circumvention like Tor (but only for a hypothetical circumvention system). This is because deriving "accurate" utility functions for Tor (or any other real-world system) that everyone would agree on is not feasible with the current state of literature on *proxy-related decision-making* due to (1) the lack of extensive (or any) social studies on the behavior and preferences of censored users, and (2) the community's limited knowledge on the internal operations of the censors. We hope that future work (particularly by social scientists and measurement researchers) will help the community to reach a reliable understanding of censors and censored users.

In the following, we derive the utility functions for HypoTor by making hypothetical assumptions about the preferences and behavior of censors and clients in HypoTor's ecosystem. While we try to derive metrics that make sense based on real-world observations, we would like to re-emphasize that we are not trying to model a real-world system like Tor.

A. Possible Actions of the Players

In each stage of HypoTor's operation, we assume that a censored client can take one or both of the following actions: (1) use a proxy he has previously obtained to browse censored websites, or (2) issue a request for new proxies to the distributor if all his known proxies are blocked. This is shown in Algorithm 2. A censoring agent takes the same actions in order to impersonate benign clients. She additionally shares her obtained proxy addresses with the central censor entity, who will decide whether to block that proxy at that given stage.

Algorithm 2 A censored client's strategy

```
PC=$\{p_1, p_2, \ldots, p_k\}$: Set of known proxies to the client
while PC is not empty do
    Choose one of the proxies from PC by uniform distribution ($p_x$)
    if Can connect to $p_x$ then
        return
    else
        Remove $p_x$ from PC
if PC is empty then
    Request for new proxies
```

B. Metrics to Distinguish Censors from Clients

In the HypoTor ecosystem, the distributor uses the following metrics to distinguish censoring agents from censored clients (Table I summarizes the notations). Note a specific real-world circumvention scenario like Tor may deploy different metrics for this purpose.

Proxy utilization (T_{a_i}): We assume that a typical censoring agent of HypoTor may not utilize the proxies she has obtained from the distributor to the same degree as a benign client (due to various costs such as bandwidth overhead). We define $T^t_{a_i,p_x}$ to represent the time duration that a_i utilized proxy p_x. The metric $T_{a_i} = \sum_{p_x \in \mathcal{P}} T_{a_i,p_x}$ represents a_i's usage for all proxies.

Blocked proxy usage ($\gamma^t_{a_i}$): An aggressive censor may block the proxies she has identified right away, before even impersonating to use them like benign clients. $\gamma^t_{a_i}$ quantifies the number of proxies obtained by a_i that were blocked without being used by a_i.

Number of requests for new proxy addresses ($R^t_{a_i}$): We use $R^t_{a_i}$ to represent the number of requests that user a_i has made up to stage t for new proxies. A benign HypoTor client will ask for new proxies only once all of her proxies are blocked, whereas a censoring agent of HypoTor may request new proxies more frequently to expedite her proxy discovery.

Number of blocked proxies that a user knows ($\delta^t_{a_i}$): We use $\delta^t_{a_i}$ to represent the number of blocked proxies that user a_i has known up to stage t. In HypoTor, this metric is on average larger for censoring agents than censored clients.

Client locations (d_{a_i,p_x}): We use d_{a_i,p_m} to indicate the normalized distance of user a_i from proxy p_x (in practice, this distance may be estimated based on IP addresses). The distance metric increases the chances of assigning the same proxies to the clients in close proximity, which improves resilience against censors who run censoring agents within the same subnet (running censoring agents from various subnets is costlier).

C. Metrics to Rank Proxies

In HypoTor, the distributor uses the following metrics to compare the importance of proxies. Censors are interested in learning (and possibly blocking) the more important proxies, so the distributor should be more protective of those more valuable proxies.

Number of users who know the proxy ($B^t_{p_x}$)*:* The number of users having the address of proxy p_i until stage t is denoted by $B^t_{p_x}$.

Number of users connected to the proxy ($c^t_{p_x}$)*:* This is the number of users connected to proxy p_x at stage t.

Total time utilization of the proxy ($\tau^t_{p_x}$)*:* Another metric to quantify the importance of a proxy is the sum of time intervals it has been used by different users. We define this metric as

$$\tau^t_{p_x} = \sum_{a_i \in \mathbb{B}^t_{p_x}} T^t_{a_i,p_x}, \tag{5}$$

where $\mathbb{B}^t_{p_x}$ is the set of user IDs who know proxy p_x. A higher value of $\tau^t_{p_x}$ indicates that the proxy is more important for circumvention.

D. Utility Functions in HypoTor

Based on the metrics introduced above, we derive the following utility functions for the clients and proxies of HypoTor.

Client utility function ($U^t_{a_i}(p_x)$)**.** For a HypoTor client a_i, we define her utility from proxy p_x as:

$$U^t_{a_i}(p_x) = \left(\beta_1 B^t_{p_x} + \beta_2 c^t_{p_x} + \beta_3 \tau^t_{p_x} - \beta_4 d_{a_i,p_x}\right) \tag{6}$$

which is a weighted sum of the proxy importance factors introduced in Section V-C, i.e., $B^t_{p_x}$, $c^t_{p_x}$, $\tau^t_{p_x}$, and d_{a_i,p_x}. Therefore, the client's utility will be higher if he is assigned to the more important proxies (since they are more reliable). β_1, β_2, β_3, and β_4 are the scaling factors indicating the relative importance of each of the proxy metrics for a typical HypoTor client.

Proxy utility function ($U^t_{p_x}(a_i)$)**.** In HypoTor, the proxy distributor has three objectives: (1) assigning as many censoring agents as possible to the *same* set of proxies, (2) assigning censored clients to reliable (non-blocked) proxies, and (3) keeping the proxies alive as long as possible. Therefore, we define the utility of a HypoTor proxy p_x with respect to client a_i as:

$$\begin{aligned} U^t_{p_x}(a_i) = \big(&\alpha_1 \min(T_{a_i}, \overline{T}) - \alpha_2 R^t_{a_i} \\ &- \alpha_3 \gamma^t_{a_i} - \alpha_4 \delta^t_{a_i} - \alpha_5 d_{a_i,p_x}\big) \end{aligned} \tag{7}$$

which is a weighted sum of the client metrics defined in Section V-B to rank HypoTor clients. Note that we cap the proxy utilization time metric by $\overline{T}$ to prevent a censoring agent from receiving large utility bonuses by leaving only some discovered proxies unblocked. The scaling factors $(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5)$ weigh the significance of different metrics in HypoTor's ecosystem (we will show their impact through experiments).

TABLE II. DIFFERENT CENSORSHIP ECOSYSTEMS

World Name	μ_b	λ_b	μ_s	λ_s
Static	20	5	2	0
Slow	20	5	5	0.1
Alive	20	5	10	0.5–7.5
Popular	20	5	20	0.5–10

VI. Simulation Setup

We built an event-based simulator for proxy-based circumvention systems, which we use to simulate HypoTor. Our simulator can be used to simulate arbitrary proxy-based circumvention systems, in particular, when a more complete set of accurate, widely accepted metrics for such real-world systems (guided by social studies and measurements) is available.

A. Simulation Parameters

Whenever possible, we use real-world data for the present study, e.g., from Tor, the most popular proxy-based circumvention system, to derive selected features of HypoTor. Needless to say, the parameters can be adjusted to other systems and threat models.

1) System's lifespan: Figure 14 (Appendix C) presents the number of active Tor bridges per month over time (error bars show standard deviation during each month). The figure shows a monotonic increase in the number of bridges during the first few years of their inception, which has changed in recent years due to various social and political events like the post-Snowden effect. As indicated by this figure, we divide the lifespan of a circumvention system into two phases. The *birth interval* is the initial phase of the circumvention system's operation, i.e., until it reaches a stable rate of growth. The second phase is the *stable interval*, which starts right after the birth interval. In our simulator, we define each time unit as *one day* in the real world. We set the birth interval to 365, therefore, representing 1 year in the real world.

2) Ecosystem: We define the following parameters to model client and proxy churn in HypoTor's ecosystem:

- μ: is the rate of new clients per time unit. μ_b is the rate during the birth interval, and μ_s is the rate after the birth interval.
- λ: is the rate of new proxies per time unit. Similarly, λ_b is the rate during the birth interval, and λ_s is the rate after that.

Figures 14 and 15 (Appendix C) show the number of Tor bridges and bridge users per month over time, respectively. To consider different simulation settings, we define various ecosystems (Table II), each with different values of the aforementioned parameters.

We also define the parameter ρ to denote the ratio of the censoring agents to the total number of clients. When adding a new user to the system, she will be a censoring agent with probability ρ and a benign client otherwise.

TABLE III. DIFFERENT SCALING PROFILES FOR DISTRIBUTOR AND CENSOR (H=HIGH, M=MEDIUM, L=LOW).

Distributor Type	α_1	α_2	α_3	α_4	α_5	β_1	β_2	β_3	β_4
Strict - Balanced distribution	L	M	H	H	M	L	M	M	M
Strict - Sparse distribution	L	M	H	H	H	H	L	M	H
Kind - Balanced distribution	H	M	M	M	M	L	M	M	M
Kind - Sparse distribution	H	M	M	M	H	H	L	M	H

Censor's Preference	ω
Blocker (prefers blocking)	L
Surveillor (prefers surveillance)	H

3) The placement of censoring agents: We consider two different types of censors. An *omnipresent censor* is a resourceful censor who is able to run censoring agents at various geographic locations (therefore better imitating censored clients). On the other hand, a *circumscribed censor* is one who is running its censoring agents within a limited region, i.e., inside a single subnet.

4) Client locations: In our simulations, we model the world as an $X \times X$ rectangular map with coordinates from $(-\frac{X}{2}, -\frac{X}{2})$ to $(\frac{X}{2}, \frac{X}{2})$. The censorship region covers a rectangular region from $(-y, -y)$ to (y, y). Benign censored users are uniformly distributed within the censored region, and the proxies are uniformly distributed outside of the censorship region. For the omnipresent censor, the censoring agents are distributed similar to benign users (i.e., uniformly). For the circumscribed censor, the censoring agents are distributed uniformly in a rectangular from $(-y_1, -y_1)$ to (y_1, y_1), where $y_1 < y$. We set $X = 20000$, $y = 1000$, and $y_1 = 100$. We use the Euclidean distance to determine the distance between users and proxies. We normalize the distance metrics to the range $[0, 1]$.

5) Proxy parameters: Similar to Tor, the `HypoTor` distributor returns $k = 3$ proxies to each new client. Otherwise, for existing clients, the `HypoTor` distributor uses the game as discussed above to assign bridges. Also, without loss of generality, we set the capacity of each proxy to be 40 clients (which is the same as rBridge [51]).

6) Scaling factors: The scaling factors α_i and β_i, used in the utility functions (6) and (7), demonstrate how the distributor weighs different factors in ranking proxies and clients. While inferring such scaling factors for a real-world system like Tor is beyond the scope of this work, we build various distributor *profiles*, as shown in Table III, each representing a distributor with different preferences. Note that, as shown in Section VII-G, our proxy distribution algorithm *is not highly sensitive* to the values of the scaling factors; therefore, in practice, the proxy distribution system only needs to infer rough estimates for such factors based on the preferences of typical clients, volunteer proxy operators, and censors.

A *Strict* distribution profile punishes users who have engaged in suspicious actions (by using large values for α_3, α_4). In contrast, we have the *Kind* distribution profile, where the distributor tolerates more suspicious actions from users. A *Balanced* distributor places less weight on the location of requesting clients in assigning proxies, in contrast to a *Sparse* distributor.

As shown in Table III, we also consider various values for ω (the scaling factor in the censor utility function, (3)) to represent censors with various preferences regarding the importance of surveillance (higher ω) versus blocking (lower ω).

B. Evaluation metrics

In our experiments, we use the following three metrics to evaluate the performance of proxy assignment in each setting.

- **Connected Clients:** This measures the fraction of censored clients with access to some unblocked proxies.
- **Unblocked Proxies:** This measures the number/ratio of circumvention proxies that are not discovered by the censors.
- **Client Wait Time:** This metric shows how many rounds a censored client should wait to obtain unblocked proxies.

C. Censorship Strategies Evaluated

We evaluate and compare three censorship strategies in our simulations. The first strategy is the optimal strategy, as devised in this paper, while the other two strategies represent the censorship strategies considered in previous studies including the state-of-the-art rBridge [51].

- **Optimal censor**: This is the optimal game-theoretic censorship strategy derived in Section IV-D.
- **Aggressive censor**: In this model, each censoring agent will immediately block a new proxy that she has identified.
- **Conservative censor**: Each censoring agent keeps the proxies she has learned alive for a certain amount of time in order to increase her utility of the system. We use the same utility function (4) used in our game to model the utility of independent censoring agents. A censoring agent will block proxies after this time interval with probability p. If the censoring agents cannot increase their utility of (4) by waiting longer, they will simply block the proxies with probability 1.

Note that to be consistent with the model used in prior work [51], [29], [28], in both of the aggressive and conservative strategies, the censoring agents act and decide autonomously.

VII. SIMULATION RESULTS

We evaluate our proxy assignment mechanism based on the setup described in Section VI, and use the metrics defined in Section VI-B to evaluate the performance of our assignment. We study our mechanism for different censorship ecosystems (Table II) and against different blocking strategies and censor placements (Section VI-C). We also compare our proposal to prior mechanisms.

A. Static World

As shown in Table II, the "static world" is the censorship ecosystem in which no new proxies are added to the system over time. As intuitively expected, our experiments show that the circumvention system is inefficient as the censors can eventually discover a large fraction of the proxies. Figure 2 shows the performance metrics (Section VI-B) of our proxy assignment mechanism in the static world ecosystem for an aggressive censor (for different fractions of censoring clients, ρ). As can be seen, even for the non-optimal aggressive censor (which is the least strategic censor) and a "strict - balance distribution" profile, the circumvention system is not able to

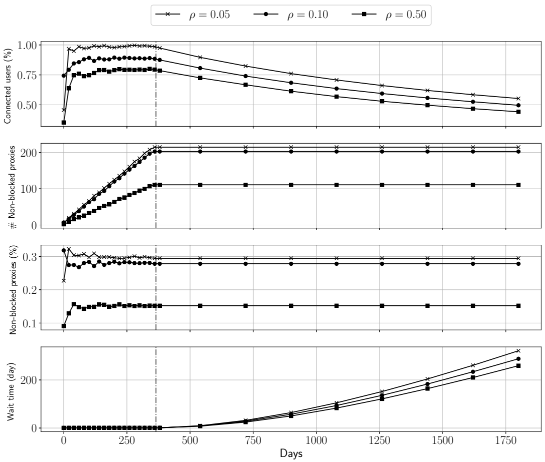

Fig. 2. Aggressive censor in a static world ecosystem.

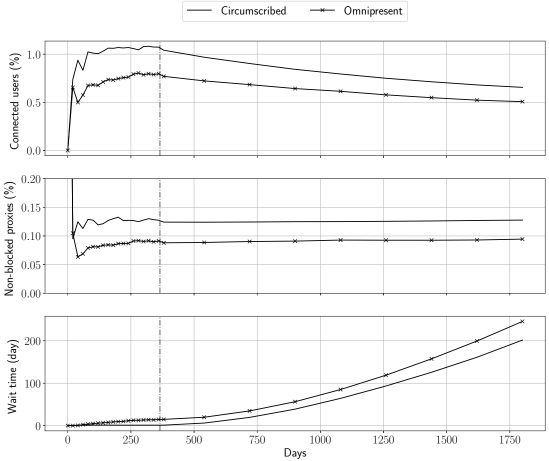

Fig. 3. Comparison between omnipresent and circumscribed censor

keep up: the fraction of connected clients decreases constantly over time.

• **Lesson One:** Independent of the censorship strategy, a circumvention system needs to add new proxies over time to be able to keep up with the censors.

B. Different Distributions of Censoring Agents

As mentioned earlier, we study two types of censoring agent distributions: *omnipresent* and *circumscribed*. Figure 3 compares the performance for these two distributions. As can be seen, the omnipresent censoring agents have a greater impact, since they can obtain a larger number of proxies due to their location diversity (which confirms (7)). Note, however, that it is costlier for the censors to distribute their censoring agents; therefore, the omnipresent censor represents a more resourceful censorship authority. In the rest of our experiments, we will use the omnipresent distribution as it represents a stronger censorship adversary.

• **Lesson Two:** Resourceful censors can increase their success by geographically distributing their censoring agents.

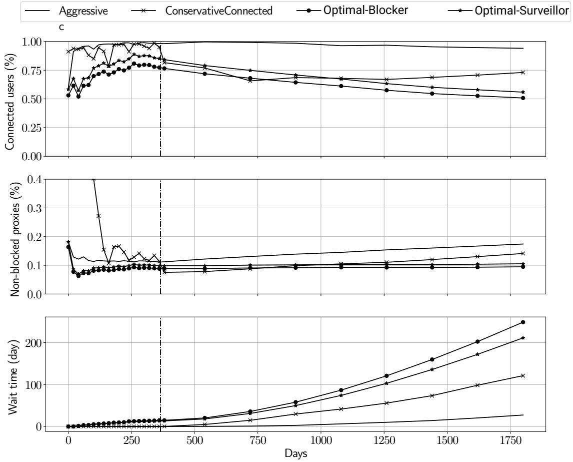

Fig. 4. Comparison of different censoring strategies in Slow world

C. Comparing Censorship Strategies

In Section VI-C, we introduced three strategies for the censors: optimal, as derived in our work, and the two strategies of aggressive and conservative, which represent the mechanisms used by prior work. Figure 4 compares the performance for these censoring strategies using the same configuration ($\rho = 0.1$, Slow world, Strict-Sparse distributor, Omnipresent censor). We observe that our proxy assignment algorithm works better against a censor with an aggressive blocking strategy than a conservative one. Comparing our optimal censorship strategy to conservative and aggressive strategies, we find it to be significantly stronger, i.e., it blocks proxies more successfully than the aggressive and conservative ad hoc mechanisms (which represent the censors modeled by prior work). Figure 4 also compares different profiles for the optimal censoring strategy. We see that changing the harshness of the censor (i.e., ω) impacts how many proxies a censor can block. Since the optimal strategy is the strongest, we will use it in the rest of experiments.

• **Lesson Three:** Censors can intensify their damage by applying strategic censorship mechanisms, as opposed to heuristics-based censorship mechanisms.

D. Different Distributor Profiles

We show that a distributor's profile (e.g., Table III) impacts the performance of the circumvention system. In particular, Figure 5 compares the performance of Kind and Strict distributors. We see that a Kind distributor connects a smaller fraction of censored users, but it provides a better wait time compared to the Strict distributor. Also, Figure 6 compares the Sparse and Balanced distribution profiles. We can see that both profiles have the same ratio of connected users, however, Sparse offers better wait times (since the censors cannot get access to all of the proxies). For the rest of the experiments, we will use the "Strict-Sparse" profile for the distributor entity.

• **Lesson Four:** The distributor's profile should be tailored for the desired trade-off between different circumvention factors.

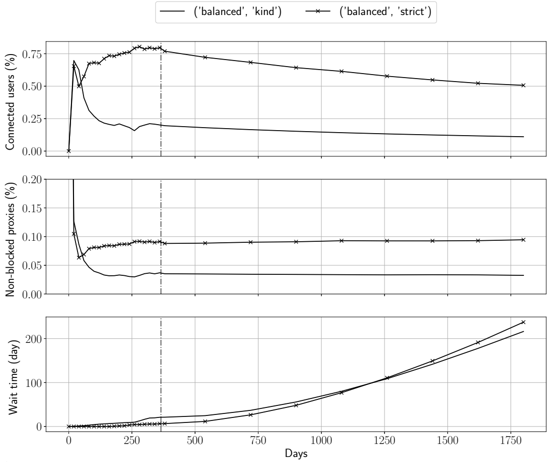

Fig. 5. Comparison of the kind to strict distribution profiles in slow world

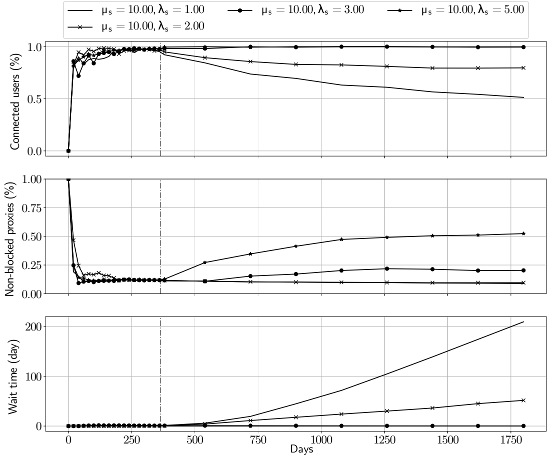

Fig. 7. Omnipresent censor with optimal blocking in *Alive* world and $\rho = 0.05$

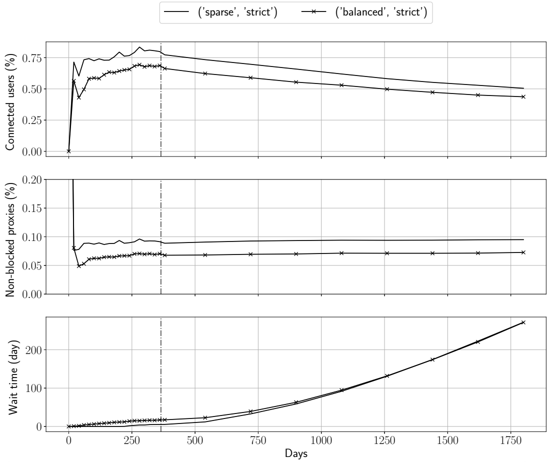

Fig. 6. Comparison of the sparse to balance distribution profiles in slow world

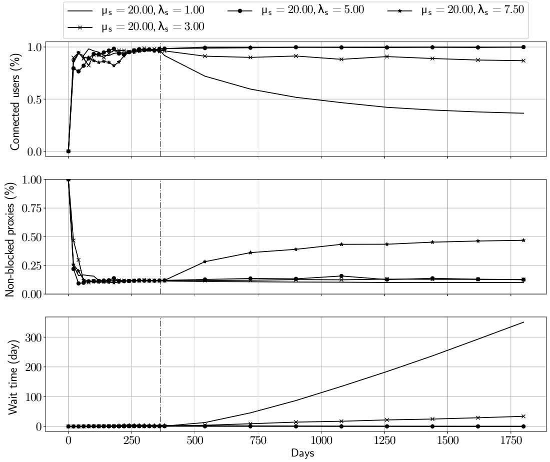

Fig. 8. Omnipresent censor with optimal blocking in *Popular* world and $\rho = 0.05$

E. Different Ecosystems

We also compare the different ecosystems defined in Table II, where each uses different rates for adding new users and proxies.

First, the rate of new proxies, λ_s, is a critical factor in the success of proxy distribution. Figures 7 and 8 compare the performance of the system for different values of λ_s for Alive and Popular ecosystems. We see that each ecosystem has an equilibrium point ($\hat{\lambda}_s \approx 3$ in our setting); if λ_s is less than $\hat{\lambda}_s$, the system will eventually get blocked over time (and the pace of blocking is inversely proportional to λ_s). On the other hand, if λ_s is much larger than $\hat{\lambda}_s$, the system will be underutilized, and therefore cost-ineffective. Also, comparing Figures 7 and 10 we see that the rate of new proxies (λ_s) should be chosen proportional to the rate of new clients (μ_s), i.e., an increase in μ_s requires the system to increase λ_s.

• **Lesson Five:** The rate of adding new proxies to a circumvention system is crucial for its effectiveness. The rate should be based on the capabilities of the censor as well as the circumvention ecosystem.

Second, protecting the system during the birth interval is crucial for its long-term operation. Specifically, if the censoring agents can get into the system at very high rates during the birth interval, the system will have a hard time to recover, regardless of the rate of new proxies added during the stable interval. This can be seen from Figures 10 and 9: despite the high rate of new proxies the system cannot recover since many benign clients have gotten flagged as censoring agents during the birth interval (since they had their assigned proxies blocked by the censors).

• **Lesson Six:** A proxy distribution system should bootstrap with trusted clients. Bootstrapping with a large fraction of malicious (censoring) clients can make the system unrecoverable.

Finally, Figure 10 shows the results for a high censoring agent rate of $\rho = 0.2$. We see that even for such a high rate of censoring agents, the circumvention system can survive by using a proper rate of new proxies (λ_s).

F. Comparison to rBridge

In Figure 11, we compare our proposed proxy assignment algorithm with rBridge [51], the state-of-the-art prior work. We use an aggressive censor for both of the systems for a fair comparison (rBridge does not consider an optimal censor).

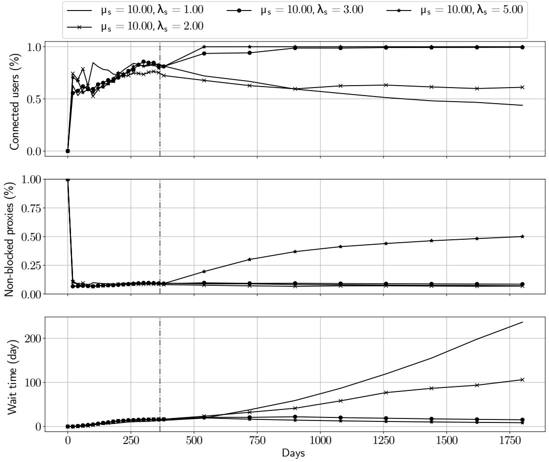

Fig. 9. Omnipresent censor with optimal blocking in *Alive* world and $\rho = 0.1$

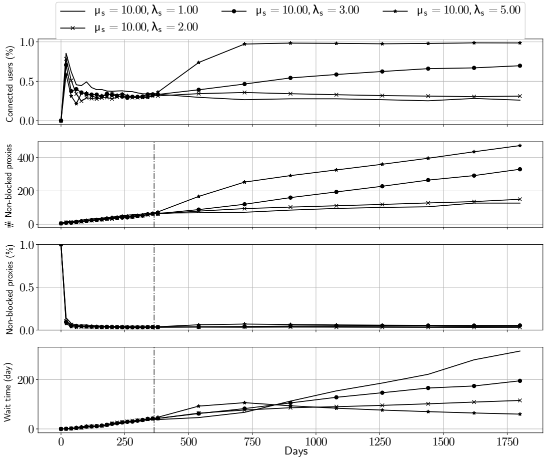

Fig. 10. Omnipresent censor with optimal blocking in *Alive* world and $\rho = 0.2$

We choose the parameters similar to values in [51], and we use $\rho = 0.05$, $\mu_s = 5$, $\lambda_s = 0.5$. As can be seen, our proxy distribution mechanism is significantly more resistant to censorship than rBridge (against the same censor attacker). That is, our mechanism can keep a larger number of censored clients unblocked.

- **Lesson Seven:** Our game-theoretic proxy distribution mechanism outperforms previous state-of-the-art mechanisms, as they are based on ad hoc approaches.

G. Sensitivity to Parameters

As explained earlier, for a proxy distribution system to deploy our algorithm in practice, it first needs to measure a set of parameters about the players' preferences, specifically, the scaling profiles of the censor, clients, and proxies in Table III. A practical proxy distribution system should *not* be sensitive to the values of these parameters, as these metrics represent the preference of typical clients and proxies, and therefore do not have accurate values. We evaluate the sensitivity of our results to small changes to these parameters. We see in Figure 12 that even a 10% change in the scaling parameters does not substantially impact the behavior of our proxy distribution algorithm.

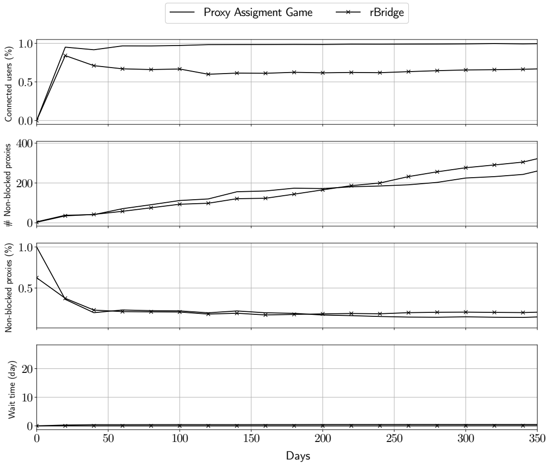

Fig. 11. Comparison with rBridge [51] using the same settings.

VIII. Discussion and Suggestions

One of the main goals of this paper is to analyze existing proxy distribution mechanisms in order to provide insights for designing stronger proxy assignment mechanisms. We have seen in practice that censoring countries are capable of blocking most Tor bridges in their purview [53]. Moreover, the number of Tor bridges has not seen proper increases in recent years—instead it has been decreasing (see Figure 14). According to our experiments and analysis, no distribution algorithm can protect proxies from the censoring agents as long as there is no influx of new proxies. Applied to Tor, this means that all bridges can/will eventually be blocked if the rate of new bridges continues to remain low (or negative with the current Tor setting). The reason for the lack of new bridges in the Tor ecosystem can be likely attributed to the fact that adding new bridges to the ecosystem is expensive (for both volunteers and Tor operators). In order to manage this situation and reduce the cost, Tor could change the IP addresses of bridges. Using expensive technologies like domain fronting [31], [15], [32] can be effective, but is likely impractical for cost reasons. Regardless, Tor as a censorship circumvention tool should employ a more strategic and principled mechanism for proxy distribution (as motivated in this paper) as opposed to its current ad hoc mechanisms. The results of our paper show the importance of central management to play the role of a single distributor in each jurisdiction. Further, our experiments and analysis corroborate the usefulness of our proposed proxy assignment game as a policy for this distributor. Another important factor derivable from our experiments and usable in the Tor ecosystem is that the rate of new bridges in Tor should be proportional to the ratio of the censoring agents (which is valid for any proxy system).

The next observation from our experiments is the importance of the birth interval. If the censor can corrupt a system during the birth interval, it is very hard for a distributor to

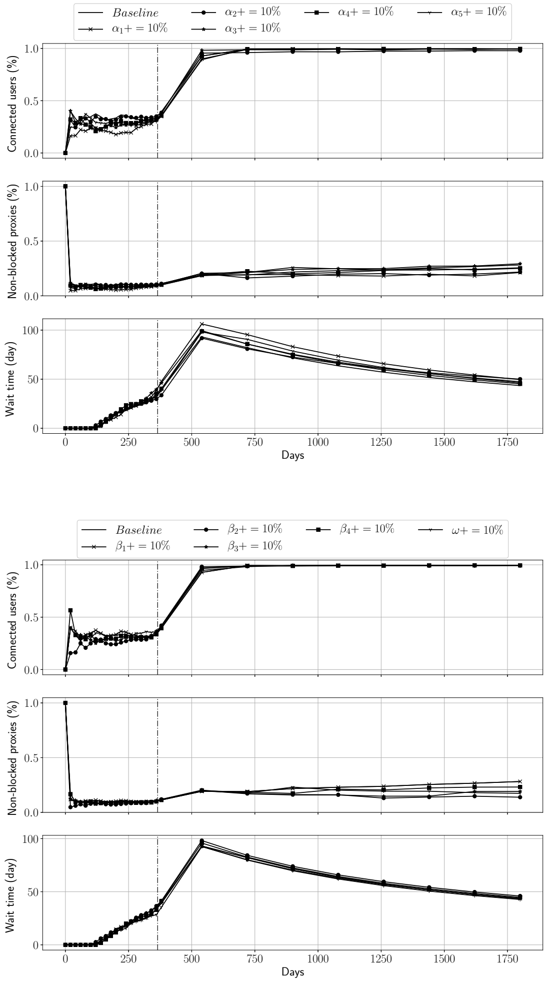

Fig. 12. Sensitivity of our algorithm's operation to small changes of the scaling factors.

recover the system according to our experiments. One of our main suggestions for a proxying system is to use a very restricted invitation system, such as [29], for the birth interval. After a while, i.e., during the stable interval, the system can transition to an open system, without a restricted invitation system. One of the main drawbacks of using an invitation system is that it cannot scale well to a large number of users. But, here, we merely propose using an invitation system in order to control the ecosystem in the birth interval to mitigate the invasion of censoring agents. Also, most of the invitation systems are capable of handling a fair amount of users in the birth interval. To evaluate our proposal, we designed an additional experiment where the ratios of censoring agents to the total number of agents are different in the birth interval and the stable interval. In the birth interval, we have $\rho = 0.02$. After the birth interval, i.e., stable interval, this ratio increases to $\rho = 0.1$ (for example, by changing to an open registration system). Figure 13 shows the outcome of our experiment. By comparing Figure 13 to Figure 9, we can observe an obvious difference: the system is able to defend itself during the birth interval, which also means that it is able to maintain its performance afterwards.

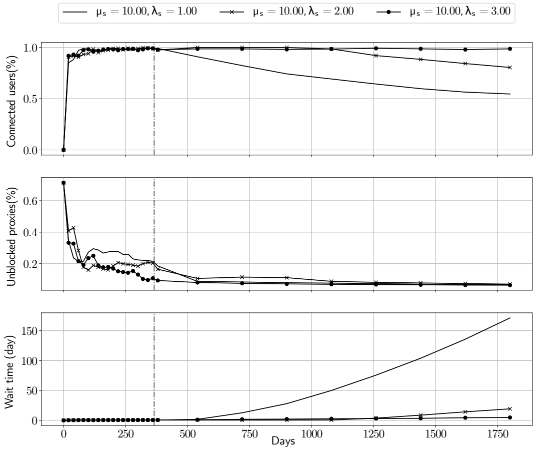

Fig. 13. Omnipresent censor with optimal blocking in *alive* world. We have $\rho = 0.02$ during the birth interval that can be construed as restricted invitation system. In the stable interval, we have $\rho = 0.1$, i.e., open world.

Finally, we would like to emphasize that, unlike previous works, our game-theoretic framework is *generic* and can be applied to *arbitrary* circumvention systems with various adversarial and network settings. As described in Section II-D, we consider network-level attacks [10], [53], [52] out of our scope as defenses to such attacks are orthogonal to proxy distribution [43], [48], [38], [21], [11], [34]. Nonetheless, we can extend our game-theoretic framework to model a proxy-based circumvention system with weaker protection against network-level attacks, as demonstrated in Appendix D.

Acknowledgments

We would like to thank anonymous reviewers for their feedback. This work was supported by the NSF CAREER grant CNS-1553301, and the German Institute for Trust and Safety on the Internet (DIVSI).

References

[1] A. Abdulkadiroğlu, P. Pathak, A. Roth, and T. Sönmez, "The Boston public school match," *American Economic Review*, vol. 95, no. 2, pp. 368–371, 2005.

[2] "Anonymizer," https://www.anonymizer.com/.

[3] "Joining China and Iran, Australia to Filter Internet," http://www.foxnews.com/scitech/2009/12/15/like-china-iran-australia-filter-internet.

[4] N. Borisov, "Computational puzzles as sybil defenses," in *Peer-to-Peer Computing*, 2006.

[5] "Tor's Bridge Distribution Webpage," https://bridges.torproject.org/.

[6] C. Brubaker, A. Houmansadr, and V. Shmatikov, "CloudTransport: Using Cloud Storage for Censorship-Resistant Networking," in *PETS*, 2014.

[7] "China blocks VPN services that let Internet users get around censorship," http://www.scmp.com/news/china/article/1689961/china-blocks-vpn-services-let-internet-users-get-\around-censorship, January 2015, online Article.

[8] R. Dingledine and N. Mathewson, "Design of a Blocking-Resistant Anonymity System," https://svn.torproject.org/svn/projects/design-paper/blocking.html.

[9] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second-generation Onion Router," in *USENIX Security*, 2004.

[10] "Ten Ways to Discover Tor Bridges," https://blog.torproject.org/blog/research-problems-ten-ways-discover-tor-bridges.

[11] K. Dyer, S. Coull, T. Ristenpart, and T. Shrimpton, "Protocol Misidentification Made Easy with Format-transforming Encryption," in *CCS*, 2013.

[12] "Egypt Leaves the Internet," http://www.renesys.com/blog/2011/01/egypt-leaves-the-internet.shtml.

[13] T. Elahi, J. A. Doucette, H. Hosseini, S. J. Murdoch, and I. Goldberg, "A framework for the game-theoretic analysis of censorship resistance," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 4, pp. 83–101, 2016.

[14] N. Feamster, M. Balazinska, W. Wang, H. Balakrishnan, and D. Karger, "Thwarting web censorship with untrusted messenger discovery," in *PETS*, 2003, pp. 125–140.

[15] D. Fifield, C. Lan, R. Hynes, P. Wegmann, and V. Paxson, "Blocking-resistant Communication through Domain Fronting," *Proceedings on Privacy Enhancing Technologies*, vol. 2015, no. 2, pp. 46–64, 2015.

[16] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *The American Mathematical Monthly*, vol. 69, no. 1, pp. 9–15, 1962.

[17] J. Geddes, M. Schuchard, and N. Hopper, "Cover Your ACKs: Pitfalls of Covert Channel Censorship Circumvention," in *CCS*, 2013.

[18] J. Holowczak and A. Houmansadr, "CacheBrowser: Bypassing Chinese Censorship without Proxies Using Cached Content," in *CCS*, 2015.

[19] A. Houmansadr, C. Brubaker, and V. Shmatikov, "The Parrot Is Dead: Observing Unobservable Network Communications," in *S&P*, 2013.

[20] A. Houmansadr, G. Nguyen, M. Caesar, and N. Borisov, "Cirripede: Circumvention Infrastructure Using Router Redirection with Plausible Deniability," in *CCS*, 2011.

[21] A. Houmansadr, T. Riedl, N. Borisov, and A. Singer, "I Want My Voice to Be Heard: IP over Voice-over-IP for Unobservable Censorship Circumvention," in *NDSS*, 2013.

[22] A. Houmansadr, W. Zhou, M. Caesar, and N. Borisov, "SWEET: Serving the Web by Exploiting Email Tunnels," in *PETS*, 2013.

[23] "How Iran Censors The Internet," http://www.popsci.com/technology/article/2013-03/how-iran-censors-internet-infographic.

[24] J. Karlin, D. Ellard, A. Jackson, C. Jones, G. Lauer, D. Mankins, and W. Strayer, "Decoy Routing: Toward Unblockable Internet Communication," in *FOCI*, 2011.

[25] "Lantern," https://getlantern.org/.

[26] C. S. Leberknight, M. Chiang, and F. Wong, "A Taxonomy of Censors and Anti-Censors Part II: Anti-Censorship Technologies," *International Journal of E-Politics*, vol. 3, no. 4, pp. 20–35, 2012. [Online]. Available: http://dx.doi.org/10.4018/jep.2012100102

[27] C. Lecher, "Internet censorship reaching dangerous levels in Turkey," http://www.todayszaman.com/national_internet-censorship-reaching-dangerous-levels-in-turkey_393727.html, July 2014, online Article.

[28] M. Mahdian, "Fighting censorship with algorithms," in *Fun with Algorithms*. Springer, 2010, pp. 296–306.

[29] D. McCoy, J. A. Morales, and K. Levchenko, "Proximax: A measurement based system for proxies dissemination," *Financial Cryptography and Data Security*, 2011.

[30] R. McPherson, A. Houmansadr, and V. Shmatikov, "CovertCast: Using Live Streaming to Evade Internet Censorship," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 3, pp. 212–225, 2016.

[31] "meek Pluggable Transport," https://trac.torproject.org/projects/tor/wiki/doc/meek.

[32] "[tor-project] summary of meek's costs, march 2017," https://lists.torproject.org/pipermail/tor-project/2017-April/001097.html.

[33] "Moat Integration," https://trac.torproject.org/projects/tor/ticket/23136.

[34] H. Moghaddam, B. Li, M. Derakhshani, and I. Goldberg, "SkypeMorph: Protocol Obfuscation for Tor Bridges," in *CCS*, 2012.

[35] M. Nasr and A. Houmansadr, "Game of Decoys: Optimal Decoy Routing Through Game Theory," in *CCS*, 2016.

[36] M. Nasr, H. Zolfaghari, and A. Houmansadr, "The Waterfall of Liberty: Decoy Routing Circumvention that Resists Routing Attacks," in *CCS*, 2017.

[37] D. Nobori and Y. Shinjo, "VPN Gate: A Volunteer-Organized Public VPN Relay System with Blocking Resistance for Bypassing Government Censorship Firewalls," in *NSDI*, 2014, pp. 229–241.

[38] "A Simple Obfuscating Proxy," https://www.torproject.org/projects/obfsproxy.html.en.

[39] V. Perta, M. Barbera, G. Tyson, H. Haddadi, and A. Mei, "A glance through the VPN looking glass: IPv6 leakage and DNS hijacking in commercial VPN clients," in *PETS*, 2015.

[40] "Psiphon," http://psiphon.ca/.

[41] A. Roth, "Deferred acceptance algorithms: History, theory, practice, and open questions," *International Journal of Game Theory*, vol. 36, no. 3, pp. 537–569, 2008.

[42] A. Roth, T. Sönmez, and U. Ünver, "Kidney exchange," *The Quarterly Journal of Economics*, vol. 119, no. 2, pp. 457–488, 2004.

[43] R. Smits, D. Jain, S. Pidcock, I. Goldberg, and U. Hengartner, "BridgeSPA: Improving Tor bridges with single packet authorization," in *WPES*, 2011, pp. 93–102.

[44] "SnowFlake Pluggable Transport," https://github.com/keroserene/snowflake.

[45] Y. Sovran, A. Libonati, and J. Li, "Pass it on: Social Networks Stymie Censors," in *IPTPS*, 2008.

[46] "Syria Tightens Control over Internet," http://www.thenational.ae/news/world/middle-east/syria-tightens-control-over-internet.

[47] "Tor: Bridges," https://www.torproject.org/docs/bridges.

[48] "Proposal 190: Password-based Bridge Client Authorization," https://lists.torproject.org/pipermail/tor-dev/2011-November/003060.html.

[49] "Tor Metrics," https://metrics.torproject.org/.

[50] M. Tschantz, S. Afroz, V. Paxson *et al.*, "SoK: Towards Grounding Censorship Circumvention in Empiricism," in *IEEE S&P*, 2016.

[51] Q. Wang, Z. Lin, N. Borisov, and N. Hopper, "rBridge: User Reputation Based Tor Bridge Distribution with Privacy Preservation," in *NDSS*, 2013.

[52] T. Wilde, "Knock Knock Knockin' on Bridges' Doors," https://blog.torproject.org/blog/knock-knock-knockin-bridges-doors, 2012.

[53] P. Winter and S. Lindskog, "How the Great Firewall of China Is Blocking Tor," in *FOCI*, 2012.

[54] E. Wustrow, S. Wolchok, I. Goldberg, and J. Halderman, "Telex: Anticensorship in the Network Infrastructure," in *USENIX Security*, 2011.

[55] H. Zolfaghari and A. Houmansadr, "Practical censorship evasion leveraging content delivery networks," in *CCS*, 2016.

Appendix

A. *Proof of Theorem 1*

Proof: Note that the deferred acceptance algorithm gives an iterative procedure for assigning students to schools. Here, we claim that the deferred acceptance algorithm is stable. In doing so, let us suppose that student a is not admitted by school b_1, but student a prefers school b_1 to his admitted school b_2. This means that student a has applied to school b_1 at some stage and student a has been rejected in favor of some students that school b_1 prefers. Then, it is obvious that school b_1 must prefer its admitted students compared to student a. Therefore, there is no instability in the deferred acceptance algorithm [16]. ■

B. *Proof of Theorem 2*

Proof: We prove the optimality by induction. Let us consider student a_1 and school b_1 with q capacity. If student a_1 is sent to a school under a stable assignment, we call that school a *possible* one for student a_1. Let us assume that at a stage in the deferred acceptance algorithm, no student has been rejected from a school which is possible for him. Let us

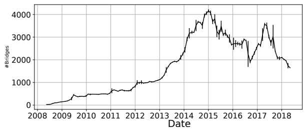

Fig. 14. Number of Tor bridges per month [49]

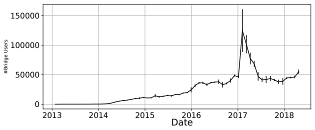

Fig. 15. Number of Tor bridge users per month [49]

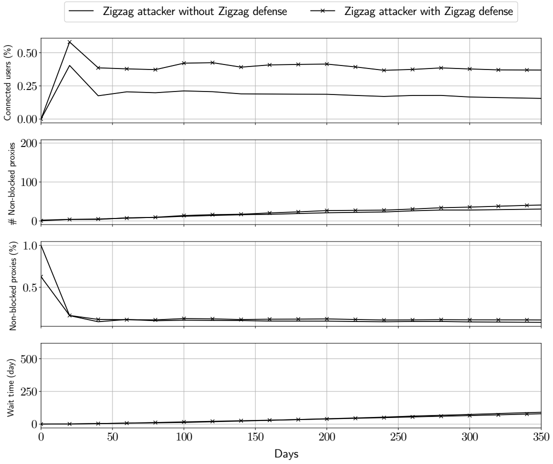

Fig. 16. Comparison of our proposed proxy assignment game with its enhanced version for zig-zag attack

also assume that school b_1 has received q applications from q students, i.e., a_2,..., a_{q+1}, who are more qualified compared to a_1. As a result, school b_1 rejects student a_1. We need to show that school b_1 is impossible for student a_1. Note that each student a_i, where $i \in \{2, ..., q+1\}$, prefers school b_1 to all of the other schools except those schools that they have previously applied to and have been rejected. In a hypothetical assignment, let us further assume that student a_1 is sent to school b_1 and all other students are sent to schools that are possible for them.

Note that there exists at least one student a_i where $i \in \{2, ..., q+1\}$ who has to go to a school which is a less desirable school compared to b_1. It is straightforward to see that this is an unstable assignment due to the fact that both the school and the student are dissatisfied about this assignment. This hypothetical assignment is unstable and school b_1 is impossible for student a_1. We can conclude that the deferred acceptance algorithm only rejects students who cannot be admitted in any stable assignment. Hence, the resulting assignment is optimal [16]. ■

C. Tor Statistics

Figures 14 and 15 show the number of Tor bridges and bridge users per month over time, respectively. The error bars show standard deviation during each month.

D. Modeling Zig-zag Attacks

We consider network-level attacks [10], [53], [52] out of our scope as defenses to such attacks are orthogonal to proxy distribution [43], [48], [38], [21], [11], [34]. Nonetheless, in this section we show how our game theoretic framework can be extended to model a proxy-based circumvention system unprotected against network-level attacks. We particularly demonstrating this for a particular type of network-level attacks called the *zig-zag* attack [10]. A censoring agent can learn the addresses of some proxies by requesting them from the proxy distributor. When a censoring agent knows some of the proxy addresses, she can observe who connects to those proxies. As those proxies are now blocked, a censored client will now request new proxies and connect to them. A censoring agent can watch those censored clients and see what other proxies they connect to. In that way, she can learn about new proxy addresses without requesting them herself from the distributor and she might even observe new users who connected to these new proxies. This process, which is called zig-zag attack, can be repeated until the censoring agent manages to block all proxies.

Our general framework in Section IV can also be used to address the *zig-zag* attack. To show the resiliency of our model with respect to zig-zag attacks, we will change our derived utility function in our `HypoTor` case study, which have been defined in Section V-D. Note that our client utility function does not change for this attack, but we define a new metric to protect the clients from zig-zag attack, which is also changing the proxy utility function.

The set of all proxies that has been used so far by all the clients, who connected to proxy p_x at stage t, is represented by $\mathbb{H}^t_{p_x}$. Let us define set $h^t_{a_i}$ as the set of all proxies that has been used by client a_i up to stage t. We update our utility function, i.e., Equation (7), as follows:

$$\begin{aligned} U^t_{p_x}(a_i) = \big(&\alpha_1 \min(T_{a_i}, \overline{T}) - \alpha_2 R^t_{a_i} - \alpha_3 \gamma^t_{a_i} \\ &-\alpha_4 \delta^t_{a_i} - \alpha_5 d_{a_i,p_x} - \alpha_6 \mid \mathbb{H}^t_{p_x} - h^t_{a_i} \mid\big), \end{aligned} \tag{8}$$

where $\mid \mathbb{H}^t_{p_x} - h^t_{a_i} \mid$ denotes the cardinality of set $\mathbb{H}^t_{p_x} - h^t_{a_i}$, which is called *zig-zag vulnerability parameter*. We add this parameter to our utility function to limit the number of proxies revealed during new proxy requests for those clients who were previously connected to the same proxy. In particular, when making a new proxy request, the client will receive a higher utility for those proxies where the currently connected clients have a similar recent history regarding proxy connections. In the following, we evaluate our proposed mechanism for zig-zag attack based on our simulation setup in Section VI.

In Figure 16, we evaluate our proposed proxy assignment algorithm for the zig-zag attack. In doing so, we compare our proposed method to prevent the zig-zag attack with our proxy assignment game without zig-zag consideration. For our evaluation, we use an *alive* censorship ecosystem with $\rho = 0.1$. As can be seen, the percentage of connected users is significantly higher when our proxy assignment game is zig-zag resilient, i.e., taking into account the zig-zag vulnerability parameter. Note that it results in a marginally lower wait time as well as a lower number of blocked proxies. Therefore, our proposed mechanism for proxy assignment can also be effective for the zig-zag attack.