GAME OF DECOYS: Optimal Decoy Routing Through Game Theory

Milad Nasr
College of Information and Computer Sciences
University of Massachusetts Amherst
milad@cs.umass.edu

Amir Houmansadr
College of Information and Computer Sciences
University of Massachusetts Amherst
amir@cs.umass.edu

ABSTRACT

Decoy routing is a promising new approach for censorship circumvention that relies on traffic re-direction by volunteer autonomous systems. Decoy routing is subject to a fundamental censorship attack, called routing around decoy (RAD), in which the censors re-route their clients' Internet traffic in order to evade decoy routing autonomous systems. Recently, there has been a heated debate in the community on the real-world feasibility of decoy routing in the presence of the RAD attack. Unfortunately, previous studies rely their analysis on heuristic-based mechanisms for decoy placement strategies as well as ad hoc strategies for the implementation of the RAD attack by the censors.

In this paper, we perform the first systematic analysis of decoy routing in the presence of the RAD attack. We use game theory to model the interactions between decoy router deployers and the censors in various settings. Our game-theoretic analysis finds the *optimal* decoy placement strategies—as opposed to heuristic-based placements—in the presence of RAD censors who take their *optimal* censorship actions—as opposed to some ad hoc implementation of RAD. That is, we investigate the *best* decoy placement given the *best* RAD censorship.

We consider two business models for the real-world deployment of decoy routers: a central deployment that resembles that of Tor and a distributed deployment where autonomous systems individually decide on decoy deployment based on their economic interests. Through extensive simulation of Internet routes, we derive the optimal strategies in the two models for various censoring countries and under different assumptions about the budget and preferences of the censors and decoy deployers. We believe that our study is a significant step forward in understanding the practicality of the decoy routing circumvention approach.

1. INTRODUCTION

Decoy routing is a promising approach to censorship circumvention, first proposed in the three independent studies of DR [16], Telex [26], and Cirripede [12]. Decoy routing presents a new paradigm in censorship circumvention: instead of the traditional approach of mounting circumvention at the edges of the Internet [2,7,11,13,15], i.e., on computer proxies, decoy routing suggests to deploy circumvention in the middle of the Internet, i.e., on Internet routers called *decoy routers*. Decoy routers are deployed by (friendly) Internet autonomous systems (AS) that install the decoy routing functionality on their existing Internet routers, e.g., on their border routers. To circumvent censorship using a decoy routing system, a censored user will need to generate traffic towards benign, non-blocked Internet destinations in such a way that it contains some covert signal for the decoy routers. If the user's traffic is intercepted by some decoy router on the path to its non-blocked (overt) destination, the intercepting decoy router will re-direct the client's traffic towards the forbidden Internet destination requested by the user.

The decoy routing approach aims at defeating the standard censorship techniques deployed against traditional proxy-based circumvention systems like Tor. Particularly, decoy routing defeats the widely used IP address-based censorship as clients' connections will appear to the censors to be destined to some overt non-forbidden IP addresses. However, the end-to-middle behavior of decoy routing circumvention enables a unique routing-based censorship attack, called *routing around decoys (RAD)* [23]. In this attack, first introduced by Schuchard et al. [23], the censors modify the BGP routing policies of the ASes they control in order to prevent their Internet users from reaching the ASes that deploy decoy routers. This presents a *fundamental attack* on decoy routing systems [8,12,16,25,26], regardless of their specific system design. In some sense, RAD on decoy routing is equivalent to IP address filtering on proxy-based circumvention systems.

The literature has recently seen a heated debate on the impact of RAD attack on decoy routing systems. While Schuchard et al. [23] present RAD as a fundamental weakness of decoy routing systems, Houmansadr et al. [14] argue RAD to be prohibitively expensive to the censors (both socially and economically) given specific decoy placement strategies. Unfortunately, existing work on decoy routing misses to perform a fundamental analysis on the feasibility of RAD. Instead, previous work, either magnifying or downplaying RAD, evaluate the attack only against *heuristic-based decoy placement* strategies and *ad hoc deployments of RAD*. Particularly, Schuchard et al. [23] demonstrate the attack only against *some*, but not the *best*, decoy placement

CCS'16, October 24 - 28, 2016, Vienna, Austria

ISBN 978-1-4503-4139-4/16/10...$15.00

DOI: http://dx.doi.org/10.1145/2976749.2978367

strategy, and Houmansadr et al. [14] evaluate the costs of RAD to the censors for a *specific* deployment of RAD, but not the *best* deployment of RAD. Cesareo et al. [4] analyze the optimum placement strategy for decoy routers—but in the absence of a RAD adversary—and Kim et al. [18] derive some sub-optimal decoy placement strategies without considering the economical and social costs of the RAD attack as demonstrated by Houmansadr et al. [14].

In this paper, we take the first systematic approach to the problem of decoy placement in decoy routing systems under adversarial settings. We use game theory to model the interactions between the parties involved in a decoy routing setting, particularly the censors and decoy routing deployers. We quantify the benefits and costs to each of the players due to their different strategies, and formulate their best responses under various settings. We use game theoretic algorithms to derive the best strategies for the placement of decoy routers as well the best strategies for the implementation of the RAD attack by the censors for different types of censors and for various decoy deployment budgets.

This work is also the first to study the potential business models for the real-world deployment of decoy routing systems. We consider two business models in our analysis: First, we consider a *central deployment* strategy in which an organization with a finite monetary budget pays autonomous systems for their deployment of decoy routers. This model is in some sense similar to systems like Tor and VPNs in which a central organization collects funding (either through donations or by charging the users) and pays the expenses of deploying the circumvention system (e.g., through running cloud servers). Our game-theoretic analysis derives the optimal set of ASes for decoy deployment, i.e., one that optimizes censorship resistance given the deploying organization's finite monetary budget.

The second business model we study is an *autonomous deployment* of decoy routers. In this model, Internet ASes decide *individually* whether or not to join a decoy routing system. ASes will receive monetary incentives for deploying decoy routers, e.g., by charging the clients, but also risk losing transit traffic if they get blocked by the RAD censors. Our game theoretic analysis finds the best response for each Internet AS given the strongest deployment of RAD optimized per Internet route by the censors.

We simulate the two decoy deployment models and derive the optimal decoy placement strategies and optimal censorship actions for various nation-state censors and under various assumptions about budget and censorship preferences. We show that *the optimal decoy placement strategies we derive significantly outperform the (heuristics-based) strategies of previous work [14]* in defeating censorship.

2. BACKGROUND

2.1 Decoy Routing Approach

Decoy routing circumvention was first proposed by the three independent works of DR [16], Telex [26], and Cirripede [12]. Unlike traditional circumvention systems [2,7,11, 13,15] in which circumvention is deployed on Internet endpoints, i.e., on computer proxies, decoy routing systems are implemented by some Internet autonomous systems (ASes), called **decoy ASes**, who mount circumvention functionality on their Internet routers, i.e., **decoy routers**.

In order to use decoy routers for circumvention, a censored client will need to establish one or multiple TLS [6] connections to arbitrary, *non-blocked* Internet destinations (e.g., non-forbidden HTTPS websites) in such a way that her traffic gets routed by at least one decoy router. The client, then, will send a covert signal to the intercepting decoy router stating her request for censorship circumvention. Finally, if the decoy router accepts to serve that client, she will deflect the client's traffic to the censored destinations requested by the client. This defeats the standard IP-based censorship as the censored client's network packets will have an IP address that corresponds to an overt non-forbidden TLS destination—but not the actual forbidden (covert) destination.

The specific design of decoy routing systems varies from one design to the other [8,12,16,25,26]. For instance, Cirripede [12] uses the initial sequence number of TCP connections for sending the covert signals to decoy routers, in contrast to Telex [26]'s use of the TLS ClientHello random nonce. Also, while DR [16] and Telex install all of the decoy routing operations on Internet routers, Cirripede uses some external servers for processing the deflected traffic. We refer the reader to the original design papers [8,12,16,25,26] for further details. **Our analysis in this paper is independent of the specific design of a decoy routing system**, and therefore applies to all decoy routing systems.

2.2 Routing Around Decoys (RAD)

A fundamental attack on the decoy routing approach is the *routing around decoys* (RAD) attack, first investigated by Schuchard et al. [23]. In this attack, a censoring country modifies the BGP routing decisions made by its ASes in order to render decoy routers unusable by its censored citizens. More specifically, if the standard (i.e., best) BGP route from a censored client to an Internet destination contains a decoy AS, the censoring country will discard that (best) BGP route and instead use another decoy-free route that she knows to that Internet destination (Schuchard et al. [23] demonstrate that censors can identify decoy ASes by sending particular probes). We refer to the alternative route chosen by a censoring AS as an **RBGP route**, in contrast to the normal **BGP route**.

RAD is a *fundamental attack* on decoy routing systems [8,12,16,25,26] regardless of their specific system designs. In some sense, RAD on decoy routing is equivalent to IP address filtering on proxy-based circumvention systems. However, performing RAD may impose various costs to the RAD censors, as discussed in previous work [14,23]:

1. Unreachable Destinations (C^1_{censor}). It is possible that for some Internet destinations the censor can not find any decoy-free routes. If the censor decides to deploy RAD on that destination, the destination will become unreachable to the censor's clients. C^1_{censor} is the fraction of such destination ASes.

2. Unreachable Domains (C^2_{censor}). Some of a censoring country's Internet domains may be hosted outside the censor's network territories, e.g., on international web hosting services. The use of RBGP may make some of such domains unreachable to the censors' users. C^2_{censor} is the fraction of such domains.

3. Increased Route Length (C^3_{censor}). The alternative RBGP routes are likely to be longer than the standard BGP

routes, therefore, degrading QoS due to failures, increased latency, etc. C^3_{censor} is the fraction of the censor's Internet routes that have become longer due to the RAD attack.

4. Expenses of Non-Valley-Free Paths (C^4_{censor}). The RAD attack may require the censoring adversary to create non-valley-free (NVF) routes between its ASes [14]. Such NVF routes impose extra monetary expenses to the censor. C^4_{censor} quantifies the fraction of a censor's paths that switch to the expensive NVF routes.

5. Less-Preferred Routes (C^5_{censor}). The use of RBGP may force a censoring AS to route some packets via a less-preferred route, e.g., by switching from a route through a peer AS to a route through a provider AS. This increases the monetary expenses of routing on such routes for the censor. C^5_{censor} is the fraction of the censor's Internet paths that switch to less-preferred routes due to RAD.

6. New Transit ASes (C^6_{censor}). For the RAD attack to be more effective, Schuchard et al. [23] suggest that censoring ASes may share their decoy-free routes. However, this may require the censor to transform some of its stub ASes into transit ASes, as discussed by Houmansadr et al. [14]. C^6_{censor} quantifies the fraction of such new transit ASes.

2.3 Past Studies on Decoy Placement

As noted earlier, Schuchard et al. [23] and Houmansadr et al. [14] analyzed the—non-optimal—placement of decoy routers in the presence of a RAD adversary. Earlier, Houmansadr et al. [12], Cesareo et al. [4], and Kim et al. [18] studied the placement of decoy routers in a non-adversarial setting, i.e., ignoring RAD. Our work is *the first* to study the optimal decoy placement strategies in the presence of RAD considering the costs to *both* censors and decoy deployers. We are also *the first* to study the optimal implementation of the RAD attack; unlike previous studies of RAD [14, 23], we consider a censor who optimizes the RAD attack by making the routing decisions *per individual Internet paths*.

3. NOTATIONS AND DEFINITIONS

Table 1 summarizes the main notations used in this paper. Our game-theoretic analysis consists of three types of players, `censor`, `decoy`, and $\texttt{AS}_k$, as introduced later. AS_i denotes the i-th autonomous system, and $\mathcal{A}_{cens}$ and $\mathcal{A}_{free}$ are the set of censor-controlled and free Internet ASes, respectively (obviously, they differ for different state censors).

We define every *ordered pair* of ASes, $P_{i,j} = (AS_i, AS_j)$, as an Internet **path** from the first AS to the second AS. Therefore, $P_{i,j} \neq P_{j,i}$. By contrast, we define $R_{i,j} = R(P_{i,j})$ as the BGP **route** (simply, route[1]) for path $P_{i,j}$, i.e., the BGP route from AS_i to AS_j. A route, if exists, will be an ordered list of ASes, $R_{i,j} = (AS_i, AS^T_1, ..., AS^T_k, AS_j)$, where each AS^T_* is a transit AS (if the route does not exist, $R_{i,j} = \emptyset$). We call the route $R_{i,j}$ to be a **decoyed route** if at least one of the ASes on the route is a decoy AS, otherwise we call it a **decoy-free route**.

For each path, $P_{i,j}$, the censor will decide one of the two actions of a^{BGP} or a^{RBGP} in order to route the packets on that path, as described later. $A_c = (a(P_{i,j})|\forall P_{i,j} \in \mathcal{P})$ is the strategy of `censor`, where $\mathcal{P}$ is the set of all paths. In our first game, `decoy`'s strategy, A_d, is the set of ASes it selects for decoy deployment. In our second game, each $\texttt{AS}_k$ decides individually from $\{a^{Deploy}, a^{NotDeploy}\}$ to whether or not deploy decoy routers.

[1]This path vs. route definition is unique to our work; others may use them interchangeably.

Censorship Metric: The goal of a RAD-capable censor is to minimize the number of the Internet paths in $\mathcal{P}$ that are served through decoyed routes (and therefore can be used by the censor's clients for circumvention). We scale each of these routes with the size of its origin (censoring) AS since this represents the number of potential censored clients who can use that decoyed route. We additionally scale each decoyed route with the size of its destination AS: as discussed by Houmansadr et al. [12], the more overt IP addresses available to the decoy users the better unobservability the system will offer to its users. We therefore formulate the effectiveness of censorship by a RAD-capable censor with a *censorship metric* defined as:

$$S = 1 - \frac{\sum_{AS_i \in \mathcal{A}_{cens}} \sum_{AS_j \in \mathcal{A}_{free}} ||AS_i|| \cdot ||AS_j|| \cdot \delta(R_{i,j})}{\sum_{AS_i \in \mathcal{A}_{cens}} \sum_{AS_j \in \mathcal{A}_{free}} ||AS_i|| \cdot ||AS_j||} \quad (1)$$

where we define $\delta(\cdot)$ as

$$\delta(R_{i,j}) = \begin{cases} 1 & R_{i,j} \text{ has at least one decoy} \\ 0 & R_{i,j} \text{ has no decoy ASes or if } R_{i,j} = \emptyset \end{cases} \quad (2)$$

Note that if the censor makes a path unreachable (i.e., $R_{i,j} = \emptyset$) we still consider that a successful censorship, though it will be at the cost of unreachability.

4. GAME ONE: CENTRAL DECOY DEPLOYMENT

In this game, we consider a central decision maker for the decoy routing system. That is, we consider a *decoy routing system* player, `decoy`, who selects the Internet ASes to deploy decoy routing. The `decoy` player has a finite monetary *budget*, which she uses to pay the selected ASes for their deployment of decoy routers. The fees paid to each AS will be proportional to their importance in the operation of decoy routing (e.g., their sizes).

This model captures the financial model behind existing centrally-operated circumvention systems like Tor [7], in which a circumvention organization collects monetary funding for the operation of the circumvention system (in Tor the funding goes to paying staff and running some servers). As shown by Houmansadr et al. [14], the location of decoy routers is critical in resisting RAD. Therefore, the **goal** of `decoy` in our game is to choose the *optimal* set of ASes for decoy deployment such that this maximizes resistance to the RAD attack given `decoy`'s finite monetary budget.

4.1 Players and Their Actions

Our game is played between `decoy` and a censoring adversary, `censor`. We consider both `censor` and `decoy` to be *rational*, i.e., each make their decisions in a way to maximize their utility functions, as defined later.

1. Central Decoy Placement System (`decoy`): The `decoy` player is the single entity who chooses a set of Internet ASes and pays them money to mount decoy routing on their border routers. The objective of the `decoy` player is to provide uncensored Internet access to a large fraction of users censored by `censor`.

Recall that $\mathcal{A}_{free}$ is the set of all Internet ASes outside the censorship region, and $\mathcal{A}_{cens}$ is the set of ASes controlled

Table 1: The list of the notations used in the paper

Notation	Description
`censor`, `decoy`, and $\texttt{AS}_k$	The players in the two games, as described later
AS_i	The i-th autonomous system (AS)
$\mathcal{A}_{cens}$	Set of all ASes in the censoring country
$\mathcal{A}_{free}$	Set of all ASes out of the censoring country
$\|\|AS_i\|\|$	Size of the AS AS_i (number of IP addresses)
$P_{i,j} = (AS_i, AS_j)$	The Internet **path** from AS_i to AS_j. Note that $P_{i,j} \neq P_{j,i}$.
$R_{i,j} = R(P_{i,j})$ $= (AS_i, AS_1^T, ..., AS_k^T, AS_j)$ or $\emptyset$	The BGP **route** from AS_i to AS_j
$\mathcal{P}$	Set of all paths from $\mathcal{A}_{cens}$ to $\mathcal{A}_{free}$
$a(P_{i,j}) \in \{a^{BGP}, a^{RBGP}\}$	The action that `censor` takes for path $P_{i,j}$. Action a^{BGP} is to use BGP do derive $R(P_{i,j})$, whereas a^{RBGP} is to use RBGP.
$A_c = (a(P_{i,j}) \| \forall P_{i,j} \in \mathcal{P})$	Vector of `censor`'s routing decisions (i.e., `censor`'s strategy)
$A_d \subseteq \mathcal{A}_{free}$	Set of decoy ASes (which is `decoy`'s strategy)
$a_k \in \{a^{Deploy}, a^{NotDeploy}\}$	The strategy of the $\texttt{AS}_k$ player in game two. a^{Deploy} is to deploy decoy routing and $a^{NotDeploy}$ is otherwise.
$\boldsymbol{\beta} = (\beta_0, ..., \beta_6)$	`censor`'s profile

by `censor`. `Decoy`'s *action space* is the set of all subsets of $\mathcal{A}_{free}$. That is, $A_d = \{AS_1, ..., AS_k | 0 \leq k \leq ||\mathcal{A}_{free}||\} \subseteq \mathcal{A}_{free}$ is a possible action for `decoy` in the game. Taking the $A_d = \{AS_1, ..., AS_k\}$ action by `decoy` means that she will pay the ASes AS_1,...,AS_k to deploy decoy routers.

We consider `decoy` to have a finite monetary budget, F, to pay the selected ASes in A_d for decoy deployment. That is, if $C(AS_i)$ is the fee `decoy` pays to AS_i for decoy deployment, we have that $\sum_{i=1}^{||A_d||} C(AS_i) \leq F$.

2. Censorship entity (`censor`): The `censor` player is a nation-state that censors the Internet access of its citizen Internet users. The main objective of `censor` in our game is to interfere with the operation of `decoy`, i.e., prevent censored users from using the decoy routing system for getting around censorship. The main technique used by `censor` to do so is the RAD attack described in Section 2.2.[2]

For each Internet path $P_{s,d} = (AS_s, AS_d)$, where AS_s is an AS controlled by `censor` (i.e., a Chinese AS) and AS_d is a non-censored AS, `censor` takes one of the following two actions: if $a(P_{s,d}) = a^{BGP}$ `censor` will use the standard BGP protocol to find the BGP route from AS_s to AS_d, and if $a(P_{s,d}) = a^{RBGP}$, `censor` will use RBGP (as described in Section 2.2) to find the route (as discussed in Section 2.2, an RBGP route may not exist, i.e., $R_{s,d} = \emptyset$). The actions taken by `censor` for different Internet paths are taken individually; `censor`'s strategy in the game is $A_c = (a(P_{i,j}) | \forall P_{i,j} \in \mathcal{P})$.

4.2 Utility Functions

We derive the game-theoretic utility functions of the two players.

4.2.1 *The* `Decoy` *Player*

Benefit: The goal of `decoy` is to maximize the censorship resistance offered to the censored users. We therefore quantify her benefit with the censorship metric of (1):

$$B_{decoy} = 1 - S \tag{3}$$

[2]RAD is the core attack known against decoy routing systems. Our model can be extended to consider other attacks proposed in the future.

Cost: `Decoy`'s cost is the sum of the monetary fees she has to pay to the selected ASes for decoy deployment:

$$C_{decoy} = \sum_{AS_i \in A_d} C(AS_i) \tag{4}$$

where $C(AS_i)$ is the deployment fee charged by AS_i, which includes the installation and operational costs of decoy routing. Therefore, we estimate $C(AS_i)$ based on the size of AS_i, as well as the economic/political relationship between AS_i's country and `censor`, i.e.,

$$C(AS_i) = \rho_0 ||AS_i|| \cdot Relation(AS_i) \tag{5}$$

where $Relation(AS_i)$ represents AS_i's relationship with `censor`. ρ_0 transforms the cost to dollar values. We will discuss the choice of these parameters in Section 6.3.

We assume that `decoy` uses all of its budget, F, for decoy deployment. Therefore, we quantify the utility of `decoy` as

$$\begin{aligned} U_{decoy} &= B_{decoy} \\ \text{subject to } C_{decoy} &\leq F \end{aligned} \tag{6}$$

where F is `decoy`'s finite budget. `Decoy`'s objective is to maximize U_{decoy} subject to the $C_{decoy} \leq F$ constraint. Intuitively, the larger the budget, the stronger the censorship resistance offered by `decoy`.

4.2.2 *The* `Censor` *Player*

Benefit: The main objective of `censor` is to prevent its Internet users from using the decoy routing system for circumvention. Therefore, we quantify its benefit with the censorship metric defined in (1):

$$B_{censor} = S \tag{7}$$

Cost: As discussed in Section 2.2, switching to RBGP instead of the standard BGP for an Internet path may impose two types of costs on the `censor` player. First, it can degrade the quality-of-service for `censor`'s benign (i.e., non-circumvention) Internet users, potentially causing collateral damage. Second, it may incur higher monetary costs to `censor`'s ASes for routing packets. We quantify the utility function of `censor` based on the cost metrics $C^i_{censor} (i = 1, ..., 6)$

Algorithm 1 Finding `decoy`'s best response (game one)

$A_d \leftarrow \{\}$
while $C_{decoy} <$ F **do**
  Sort ASes : For each AS_i compute its benefit contribution: $B^{AS_i}_{decoy} = \sum_{(AS_s,AS_d)\in\mathcal{P}} ||AS_s|| \cdot ||AS_d|| \mathbb{1}\{AS_i \in R_{s,d}\}$ Sort ASes by their benefits.
  Pick ASes: Choose the AS with the highest benefit and add to A_d.
  Update: Remove all the routes that contain the selected AS.
$A'_d \leftarrow \{\}$
while $C_{decoy} <$ F **do**
  Sort ASes : For each AS_i compute its benefit contribution: $B^{AS_i}_{decoy} = (\sum_{(AS_s,AS_d)\in\mathcal{P}} ||AS_s|| \cdot ||AS_d|| \mathbb{1}\{AS_i \in R_{s,d}\})/C(AS_i)$ Sort ASes by their benefits.
  Pick ASes: Choose the AS with the highest benefit and add to A'_d.
  Update: Remove all the routes that contain the selected AS.
if $U_{decoy}(A_d) < U_{decoy}(A'_d)$ **then**
  Return A'_d
Return A_d

defined in Section 2.2:

$$U_{censor} = \beta_0 B_{censor} - \sum_{i=1}^{6} \beta_i C^i_{censor} \quad (8)$$

We define $\beta = (\beta_0, \beta_1, ..\beta_6)$ as `censor`'s *profile*, which demonstrates how much she cares about each of the cost metrics with respect to the enforced censorship. In Section 6, we perform our analysis for different censor profiles. The objective of `censor` in the game is to maximize U_{censor}.

4.3 Mechanism Design

We consider the set of actions available to the `censor` and `decoy` players as well as their corresponding utility functions (as derived in Sections 4.1 and 4.2), to be *public knowledge*. Therefore, we model this game as a "complete information" game [21]. This allows each player to simulate the game in order to find her own best response. That is, `decoy` can use our analysis to identify the location of optimum decoy ASes given her finite budget, and `censor` can use the analysis to decide her optimum routing actions for its Internet paths.

Best response of `decoy`. The best response of `decoy` given `censor`'s action A_c is the set of ASes chosen for decoy deployment that maximizes `decoy`'s utility function of (6) given her known finite budget:

$$A^*_d = \underset{A_d}{\text{argmax}}\ U_{decoy}|A_c \quad (9)$$

This problem can be interpreted as a "budgeted maximum coverage" problem [17], which is known to be NP-hard. However, we are able to converge to the solution through approximation. As we prove in Lemma 2, `decoy`'s utility function is a monotone *submodular* function [9]. This allows us to use Leskovec et al.'s greedy algorithm [20] to find a *sub-optimal* best response for `decoy` within $\mathcal{O}(1-\frac{1}{e})$. Algorithm 1 sketches our greedy algorithm.

Best response of `censor`. The best response for `censor` given `decoy`'s action A_d is an A_c vector (as defined in Section 4.1) that maximizes the utility function of (8):

$$A^*_c = \underset{A_c}{\text{argmax}}\ U_{censor}|A_d \quad (10)$$

We consider the elements of A_c to be independent, i.e., the actions for different paths are taken independently by `censor`. This is because (a) the benefit function of (7) can be decomposed into the independent elements of $||A_i|| \cdot ||A_j||(1-\delta(R_{i,j}))$, and, (b) each of the cost components in (8) can be represented as the sum of the costs across all paths.[3] Using this independence assumption, we decompose the maximization problem of (10) into $||\mathcal{P}||$ independent maximization problems, which is solvable in polynomial time (i.e., with $2||\mathcal{P}||$ computations since there are two potential actions for each path).

Finding the Equilibrium: The goal of our analysis is to find an equilibrium point for the game, which is a pair (A^*_d, A^*_c) that satisfies both (9) and (10) concurrently. In a real-world deployment of decoy routers, we expect the players reach such an equilibrium point after potentially changing their strategies for several times.

In our analysis we only seek a pure Nash equilibrium, but not a mixed one [21]. This is because mixed strategies do not have real-world interpretations in scenarios where players' actions are observable to each other (see the network design problem [1] and facility location game [24] as examples). In our game also the `censor` and `decoy` players can observe the actions taken by each other: Schuchard et al. [23] demonstrate that censors can infer the identities of decoy ASes with very high confidence. Also, `decoy` can observe `censor`'s routing decisions by sending decoy routing traffic from inside the censorship region to the decoy routers.

However, not every game has a pure Nash equilibrium [24] (or if it does, finding it is not necessarily straightforward). We therefore solve the game by finding a pure $\varepsilon-$Nash equilibrium through empirical analysis.[4] Particularly, we use the *best-response dynamics* algorithm [19] for converging to an $\varepsilon-$Nash equilibrium. The algorithm is run in multiple iterations until no player can improve its utility beyond a threshold ε by changing strategy.

5. GAME TWO: AUTONOMOUS DECOY DEPLOYMENT

In this game, we consider the financial model where autonomous systems decide *individually* whether or not to deploy a decoy routing system. Unlike game one in which a central `decoy` player pays ASes to deploy decoy routing, in this game each AS individually decides to deploy or not based on her own *economic interests*.

5.1 Players and Their Actions

This game is played between $n+1$ players: a `censor` player, which is the same as the one in game one, and n autonomous system players, which are the ASes in the free world, i.e., $n = ||\mathcal{A}_{free}||$. From a high level, the objective of each AS player is to maximize her monetary revenue, and the goal of `censor` is to optimize the enforced censorship.

1. Censorship entity (`censor`): This player is exactly the same as the `censor` player of game one (Section 4.1).

[3] In reality, if changing the route of some path significantly changes the load on some Internet links, this may increase the latency for other Internet routes as well, however, we assume that in the long run Internet transit ASes will catch up with the changes to their transit traffic volumes.

[4] In an $\varepsilon-$Nash equilibrium point, changing one player's strategy may increase its utility by a maximum of ε.

2. n **AS Players** ($\texttt{AS}_k$ for $1 \leq k \leq n = ||\mathcal{A}_{free}||$): This includes all ASes not-controlled by `censor`. Each `AS` player will decide autonomously whether or not to deploy decoy routing based on her economic interests. We show $\texttt{AS}_k$'s action as $a_k \in \{a^{Deploy}, a^{NotDeploy}\}$, where a^{Deploy} is the decision to deploy decoy routing and $a^{NotDeploy}$ is otherwise.

5.2 Utility Functions

We quantify the utility functions of the players as follows.

5.2.1 *The* `Censor` *Player*

This is the same as what we derived in (8) for game one.

5.2.2 `AS` *Players*

For each AS player $\texttt{AS}_k$, we define S_{AS_k} as the set of all routes in $\mathcal{P}$ (from $\mathcal{A}_{cens}$ to $\mathcal{A}_{free}$) that go through $\texttt{AS}_k$:

$$S_{AS_k} = \{R_{s,t} | AS_s \in \mathcal{A}_{cens}, AS_t \in \mathcal{A}_{free}, AS_k \in R_{s,t}\} \tag{11}$$

For each route $R_{s,t} \in S_{AS_k}$, $\texttt{AS}_k$ will earn monetary revenue for transiting traffic on $R_{s,t}$ if $\texttt{AS}_k$ is a "provider" [14] on $R_{s,t}$. The action that $\texttt{AS}_k$ chooses from $\{a^{Deploy}, a^{NotDeploy}\}$ in the game, combined with the actions of the other $n-1$ `AS` players and `censor`, may impact $\texttt{AS}_k$'s revenue (i.e., utility) for each route $R_{s,t} \in S_{AS_k}$ as follows:

- If `censor` takes the a^{RBGP} action for $R_{s,t}$, $\texttt{AS}_k$ will lose the transit revenue (if any) she would normally receive for that route.
- If `censor` takes the a^{BGP} action for $R_{s,t}$, $\texttt{AS}_k$ will keep the transit revenue for that route. Additionally, $\texttt{AS}_k$ will earn extra monetary revenue for serving the decoy routing users who use $R_{s,t}$ *only if* (1) $\texttt{AS}_k$ decides to be a decoy ($a_k = a^{Deploy}$) and (2) $\texttt{AS}_k$ is the first decoy AS on $R_{s,t}$. In a practical deployment, this revenue may come from different sources: decoy clients can pay directly for the service they receive (similar to paid VPN services), pro-freedom NGOs can pay decoy routers per the volume of decoy traffic they serve, etc.

Note that the actions of each `AS` player may impact the utility of other `AS` players as well, even if they choose the $a^{NotDeploy}$ action. For each route $R_{s,t}$, we quantify the monetary revenue of transit traffic as well as decoy traffic to be proportional to the volume of the traffic routed on that route. For the lack of comprehensive data on traffic loads across Internet routes, we use $||AS_s|| \cdot ||AS_t||$ as an estimate. Therefore, we quantify $\texttt{AS}_k$'s utility as:

$$U_{AS_k} = \rho_1 \sum_{R_{s,t} \in S_{AS_k}} ||AS_s|| \cdot ||AS_t|| (1 + \gamma.\sigma(R_{s,t}, AS_k)) \tag{12}$$

where the first term in the summation corresponds to the transit traffic revenue and the second term is the decoy routing revenue. $\sigma(R_{s,t}, AS_k) = 1$ if $\texttt{AS}_k$ is a decoy *and* it is the first decoy AS on $R_{s,t}$, otherwise $\sigma(R_{s,t}, AS_k) = 0$. We use γ to scale the decoy routing revenue with respect to the transit revenue. We call γ as the *service fee* of decoy ASes. ρ_1 is a scaling factor to convert to the dollar value.

5.3 Mechanism Design

Similar to the game one, we consider the set of actions available to each of the `censor` and `AS` players as well as their corresponding utility functions to be public knowledge. Therefore, our game is a "complete information" game [21],

Algorithm 2 Finding equilibrium in game two

```
A'_d ← {}
A_d ← {}
A_c ← CBGP
while True do
    A_d ← AS actions (Step 1)
    ratio ← |A_d ∩ A'_d| / |A_d ∪ A'_d|
    A_c ← Censor Best Response (Step 2)
    if ratio ≥ τ then
        EQ ← refineEQ(A_d ∩ A'_d, A_c)(Step 3)
        return EQ
    A'_d ← A_d

Def refineEQ(ASes,R):
    Selected ← ASes
    A_d ← ASes
    A_c ← R
    Converged ← False
    History = {ASes}
    while !Converged do
        A_d ← AS actions (Step 1)
        A_c ← Censor Best Response (Step 2)
        if Selected ∩ A_d ≠ Selected then
            Selected = A_d ≠ Selected
            History = {Selected}
        if A_d ∈ History then
            Converged ← True
        History.append(A_d)
    Return Selected, A_c
```

which enables `censor` and each `AS` player to simulate the game and find their own best responses.

Best response of `censor`. We derive this as:

$$A_c^* = \underset{A_c}{\text{argmax}}\ U_{censor} | a_k (1 \leq k \leq n) \tag{13}$$

where U_{censor} is given by (8), and a_k is the action of $\texttt{AS}_k$.

Best response of `ASes`. The best response for an AS player, $\texttt{AS}_k$, is the action $a_k^* \in \{a^{Deploy}, a^{NotDeploy}\}$ that maximizes her utility U_{AS_k} derived in (12) given the actions of the other $n-1$ `AS` players and `censor`:

$$a_k^* = \underset{a_k \in \{a^{Deploy}, a^{NotDeploy}\}}{\text{argmax}}\ U_{AS_k} | A_c, a_j (1 \leq j \leq n, j \neq k) \tag{14}$$

Finding the Equilibrium: The game's equilibrium is (A_d^*, A_c^*), where $A_d^* = \{AS_i | a_i^* = a^{Deploy}\}$. (A_d^*, A_c^*) concurrently satisfies (14) for all n `AS`es and (13) ($n+1$ equations). Similar to the previous game, we only consider the pure equilibrium, since mixed strategies do not have real-world interpretations in the decoy application.

Finding the equilibrium in this game poses as a complex computational problem as it involves $n+1$ equations. To be able to solve this in polynomial time, we translate the game into a game played in three steps in multiple rounds. Algorithm 2 summarizes how we simulate the game.

Step 1 (n `ASes` taking actions): In this step, each of the n `AS` players take an action that maximizes their utility in (14) (i.e., a_k^* for $\texttt{AS}_k$) given a fixed, known `censor` action, $\mathcal{A}_c$, which may or may not be `censor`'s equilibrium action. In other words, given `censor`'s known routing decisions for all Internet routes, each `AS` decides to whether or not deploy decoy routing based on her utility function. Each `AS`'s action can potentially impact the utility of other `AS` players. Therefore, we model step 1 as a "non-cooperative facility location" game [24]. A facility location game is composed of multiple service providers as well as multiple users seeking

service, where the game aims at finding the best locations for servers to maximize their individual benefits.

We use the *best response dynamics* [21] to find the equilibrium of this game, which is known as a standard mechanism for facility location games [24] and is shown to be computable in polynomial time by Vetta [24]. The outcome of this step is a A_d^* set, containing the set of decoy ASes.

Step 2 (censor taking actions): Assuming that censor knows the actions taken by all AS players, i.e., he knows A_d, censor will derive his best response A_c^* given in (13). We use the mechanism described in Section 4.3 (under "Best response of censor") to derive A_c^*, which is a polynomial time algorithm as discussed in that section.

Step 3 (Refineing): If the pair (A_d^*, A_c^*), derived in steps 1 and 2, is an $\varepsilon-$Nash equilibrium point we terminate the algorithm, otherwise we go to step 1. If an $\varepsilon-$Nash equilibrium has been achieved, the improvement in each player's utility function will be smaller than ε if the players decide to change their actions.

Our game is composed of n AS players. As n is a large number (e.g., $n = 49,745$ for China) some of the AS players may take longer to reach an equilibrium state in our simulations. We, therefore, terminate the game when a large ratio ($\tau = 0.9$) of AS players have reached their equilibrium, i.e., when A_d does not change significantly between the consecutive iterations.

6. SIMULATIONS

6.1 Experimental Setup and Datasets

We use C-BGP [22],[5] a widely used Internet route simulator, to infer BGP routes between Internet ASes in our experiments. We also code an RBGP simulator in Python that implements the RAD attack as described before [14,23]. Our RBGP simulator uses C-BGP as its engine to find the routes. We run our computation-intensive simulations (i.e., routes between over 50,000 Internet ASes in different settings) on Google Compute Engine.[6]

Our game-theoretic algorithms (e.g., to find equilibrium) are implemented in Python and are run on a Linux box with 64GB of memory and 3.5GHz Xeon(R) CPU. We use multiple optimization techniques to speed up the algorithms, including *dynamic programming*, *lazy evaluation*, and *in-memory databases*. Particularly, we use the Redis[7] in-memory data structure to manage the tremendous volume of routing records, DNS records, and the other datasets used in our experiments.

We make use of the following datasets in our experiments:

- We use CAIDA's "Jan 2016" AS relationships database [3] to model the business relationship between ASes.
- We use GeLite2 geo-location database[8] for the mapping between IP addresses and geographical locations.
- We use CAIDA's AS rank dataset[9] to map between ASes and IP address ranges, also for AS sizes.
- We use the OEC[10] dataset [10] to infer the economical relationship between countries.
- We use the DNS Census dataset[11] composed of about 2.5 billion DNS records to find the domain names belonging to various censoring countries. This is particularly used in evaluating C_{censor}^2.

6.2 Profiles for the Censor Player

As described earlier, we characterize each real-world censor with a profile, $\boldsymbol{\beta} = (\beta_0, ..., \beta_6)$, which represents how much that censor cares about the cost metrics compared to the achieved censorship. More specifically, β_0 represents the importance of censorship effectiveness to censor, whereas β_1 to β_6 show how much censor wants to avoid the collateral censorship costs, as defined in Section 2.2.

Intuitively, the game's outcome highly depends on the censor profile, i.e., $\boldsymbol{\beta}$ impacts censor's best routing strategy as well as the best decoy placement strategy. We therefore perform our analysis for different censor profiles. We particularly use the representative profiles shown in Table 2 in our analysis. In the real world, a nation-state censor may change its profile from time to time, e.g., may take a harsher profile during a political unrest.

A "wealthy" censor can sustain large monetary expenses, therefore, her profile has low values for β_4 to β_6, which scale the monetary cost metrics of C_{censor}^4 to C_{censor}^6. By contrast, a "poor" censor has high values for β_4 to β_6. A "QoS-cautious" censor strongly desires to avoid degradation in Internet quality-of-service for its citizens, so she has high values for β_1 to β_3, which scale the QoS cost metrics of C_{censor}^1 to C_{censor}^3. This is in contrast to a "QoS-ignorant" censor who does not care about the QoS cost metrics. As an extreme censor profile, the "irrational" censor aims to deploy the strongest form of censorship with little concern about QoS and with ample monetary resources (a high β_0 shows the censor's desire for the strongest censorship). We also define the "reachability-cautious" and "domain-cautious" profiles, that aim to preserve specific QoS properties, e.g., T_4 is highly concerned about Internet reachability for its users.

In our analysis, we use the values in Table 3 as high and low values for each β_i. Note that what is important to our analysis is not the absolute values of β_i's, but how they compare relatively.

6.3 Other Parameters and Settings

Relation($\mathbf{AS_i}$): The $Relation(AS_i)$ function represents the economic relationship between AS_i and the censoring country, and is part of the decoy deployment cost function in (5). The stronger this relationship, the higher the economical costs of decoy deployment for AS_i (i.e., the less likely for AS_i to deploy decoy routers). We quantify $Relation(AS_i)$ based on the economic relationship of AS_i's country with censor. Specifically, we estimate $Relation(AS_i)$ with the sum of the import and export rates between the two countries, derived from the OEC dataset [10]. Also, as suggested previously [14, 23], we consider the *ring ASes* of a censoring country to be extremely unlikely to deploy decoy routers due to their direct business interaction with the censoring ASes. We therefore assign significantly large values to

[5] We do understand that the C-BGP simulator has inaccuracies in inferring paths between ASes, but since such inaccuracies are uniform across all routes we expect them not to impact our analysis significantly.
[6] https://cloud.google.com/compute/
[7] http://redis.io/
[8] http://www.maxmind.com
[9] http://as-rank.caida.org/
[10] http://atlas.media.mit.edu/en/resources/data/
[11] https://dnscensus2013.neocities.org/index.html

Table 2: Various `censor` profiles (H:High, L:Low)

Feature	β_0	β_1	β_2	β_3	β_4	β_5	β_6
Impacts	Cens	QoS	QoS	QoS	$	$	$
T_1: QoS-cautious, wealthy	L	H	H	H	L	L	L
T_2: QoS-cautious, poor	L	H	H	H	H	H	H
T_3: QoS-ignorant, poor	H	L	L	L	H	H	H
T_4: Reachability-cautious, wealthy	H	H	L	L	L	L	L
T_5: Domain-cautious, wealthy	H	L	H	L	L	L	L
T_6: Irrational	H	L	L	L	L	L	L

Table 3: H and L values

Coefficient	High	Low
β_0	50	1
β_1	10	0.0
β_2	10	0.0
β_3	10	0.0
β_4	10	0.0
β_5	10	0.0
β_6	10	0.0

Table 4: Number of ASes in Each Country

Country	ASes	#Rings
China	573	858
Syria	4	5
Saudi Arabia	107	176
Venezuela	44	835

$Relation(AS_i)$ for all ring ASes of `censor`, effectively causing no ring AS to deploy decoy routers in our simulations.

F, ρ_0, ρ_1, and γ: We will present our simulation results for different values of F/ρ_0 in game one, and for different values of γ in game two. F/ρ_0, which is the budget ratio, represents the monetary budget of the central decoy deployer in game one. Also, the service fee γ represents the monetary fee to be charged by decoy ASes for decoy traffic in game two. The value of the ρ_1 parameter in (12) does not matter as it does not impact the maximization problem of (14).

C^i_{censor}: We derive these as defined in Section 2.2.

Mapping ASes to countries: We use the GeLite2 and CAIDA AS rank datasets to map the 52,351 Internet ASes to countries. Some ASes are assigned to multiple countries in the two datasets. If an AS is assigned to both a censoring country and a free country, we consider it as a censoring AS to make the simulations in `censor`'s favor. We also identify each country's ring ASes as the ASes with a direct link to the country's ASes. Table 4 shows the number of ASes and ring ASes for the censoring countries of our experiments.

On the optimality of the results: As discussed earlier, our algorithms find the near-optimal strategies for decoy placement and censorship, i.e., our results are ε-Nash equilibria with $\varepsilon = 0.1$ in most cases. To find the most conservative decoy placement, we always let `censor` make the final move in the games.

6.4 Game One Experiments

We use the game theoretic analysis of Section 4 to converge to the *optimal decoy placement strategies* when the censors are taking *optimal censorship actions*. We perform our simulations for the four censoring countries in Table 4, which represent censors with various Internet connectivity (and therefore different censorship capability). We start our evaluation by discussing the results for China, which is the strongest censoring country among the list due to its highly well-connected network. Next, we will compare the results across the censoring countries.

How the routes change. The routing decisions made by `censor` in the game impacts the Internet routes of the `censor`'s Internet users. For instance, as we see in Figure 1,

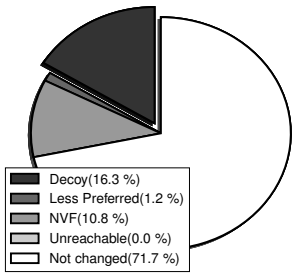

Figure 1: *Central decoy deployment (game one)*: Impact on Chinese routes for a budget ratio of $F/\rho_0 = 5 * 10^6$ and a T_1 censor profile.

if `decoy` has a budget ratio of $F/\rho_0 = 5 * 10^6$, the optimal routing decision of the Chinese `censor` with a profile T_1 causes 10.8% of the routes to become NVF, and 1.2% to go through less-preferred routes. For this particular censor profile, even the best censorship strategy leaves 16.3% of the routes available to the Chinese users for decoy routing.

Impact of decoy budget. As intuitively expected, `decoy`'s budget significantly impacts the success of decoy routing circumvention in game one. That is, with more budget `decoy` can choose more ASes and/or more effective ASes for decoy deployment. As noted earlier, we use the budget ratio (F/ρ_0) as a metric to represent the monetary budget available to the central decoy deployer, `decoy`. Figure 2 shows how `decoy`'s budget impacts the utility of the decoy routing system as well as the censorship metric for China as the `censor` with a T_1 (QoS-cautious, wealthy) profile. As expected, increasing `decoy`'s budget improves censorship resistance, i.e., reduces the censorship metric S, as it allows the central decoy deployment organization to install decoy routing on more Internet ASes, and/or on the more effective ASes (e.g., those on more routes). The figure also shows how different cost metrics for `censor` change with `decoy`'s budget. As can be seen, the monetary cost metrics (C^4_{censor} to C^6_{censor}) first show an increase with F/ρ_0 but at some point start declining since switching to more expensive routes will no longer improve the censorship success for `censor`.

Note that our evaluation is with respect to the budget ratio parameter, F/ρ_0, but not the actual budget F. One can infer the actual dollar budget by investigating the real-world deployment cost of decoy routers in (5), therefore estimating and eliminating ρ_0 from F/ρ_0. We leave this to future work.

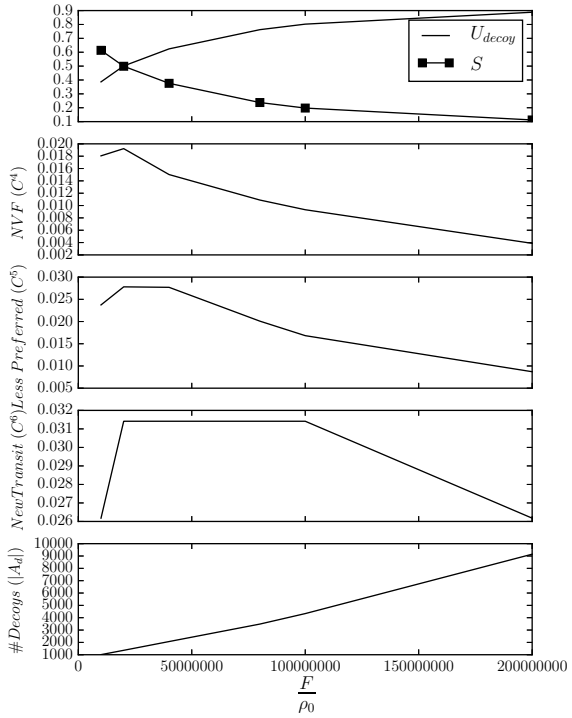

Figure 2: Impact of `decoy`'s budget ratio (F/ρ_0) on the censorship metric and censorship costs (`censor` is China with a T_1 profile).

Impact of censor profiles. The optimal censorship strategy depends on the censor's profile, i.e., censors will make routing decisions based on their preferences on the cost metrics with respect to the censorship metric. This in turn will impact the optimal set of decoy ASes for the central decoy deployer. Table 5 compares the cost metrics as well as the censorship metric for China for different censorship profiles. We see that the censorship costs as well as the achieved censorship largely varies across the profiles. For instance, comparing T_2 (QoS-cautious, poor) and T_3 (QoS-ignorant, poor) we see that if the Chinese censor does not care about the QoS offered to its clients, it can improve the censorship metric from 0.229 to 0.949 (given the same monetary resources for the censor). This, however, comes at the price of making 70.3% of all Internet routes becoming unreachable to the Chinese users, potentially causing collateral damage and unrest. On the other hand, we see that if China is committed to preserving QoS, i.e., not to cause collateral damage, even if they spend significant amounts of money (i.e., profiles T_1 and T_4) the censorship metric they can achieve will be around 0.27. This is a **promising finding for the decoy routing approach** indicating that if there is enough monetary budget for decoy deployment, "rational" censors will not be able to evade it even if they spend big money. On the other hand, we see if China acted irrationally (profile T_6) they can achieve perfect censorship at the price of significant QoS degradation to their users. This is not specific to decoy routing systems; irrational censors can always achieve

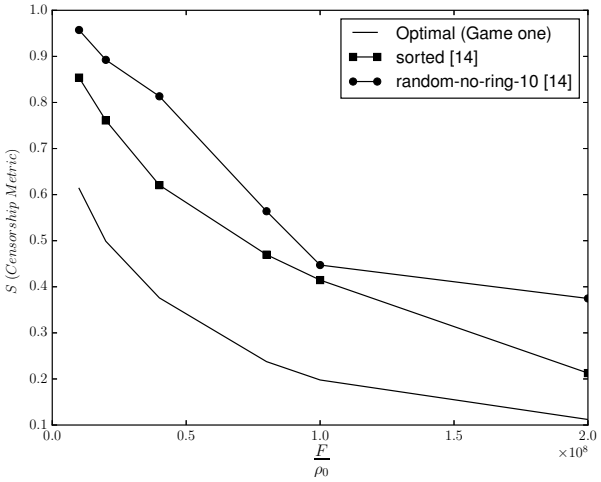

Figure 3: Comparing our central decoy placement strategy (game one) with state-of-the-art mechanisms [14].

complete censorship, e.g., they can entirely disconnect their country from the Internet as witnessed in Egypt [5].

Comparison with previous work: We compare our (near-)optimal decoy placement mechanism with the (non-optimal, heuristics-based) decoy placements proposed in previous work. Specifically, we compare our decoy placement with the state-of-the-art decoy placement mechanisms proposed by Houmansadr et al. [14], i.e., the "sorted" and "random" placements. This is shown in Figure 1 for different budget ratios. As can be seen, **our decoy placement mechanism is significantly more effective than previous mechanisms in defeating censorship**. That is, given the same budget for decoy deployment our decoy placement results in better censorship circumvention performance, i.e., a smaller censorship metric—and this is given the "best" censorship strategy, not just some strategy as in prior work [14]. For instance, for the budget of $F/\rho = e^8$ our decoy placement achieves a censorship metric of 0.2 as opposed to 0.42 for "sorted" [14].

Comparing censoring countries. We perform our simulations for the 4 representative censoring countries in Table 4. The countries represent various kinds of censoring countries regarding their Internet connectivity. China has the best connectivity to the Internet with the largest number of ASes and ring ASes, whereas Syria poses a weak connectivity. Table 6 compares the censorship performance and cost metrics for the four countries for the censor profile T_1 and given a fixed decoy deployment budget. We have sorted the countries in the descending order of the censorship metric. As can be seen, China poses as the strongest censor because of its highly connected Internet, while Syria is the weakest censor because of its poorly-connected Internet. We also see that even though Venezuela has less ASes than Saudi Arabia, it is a slightly stronger censor in this setting, presumably because of its larger number or ring ASes. Finally, we see that given the same budget, the optimal set of decoy ASes differ for different countries, as evident from the number of decoy ASes in each case.

Table 5: *Central decoy deployment (game one)*: Comparing the impact of different `censor` profiles on various cost metrics and the censorship success. The `censor` is China and `decoy`'s budget ratio is $F/\rho_0 = 10^7$.

Profile	Unreachable (C_1)	Unreach. Domains (C_2)	Inc. Length (C_3)	NVF (C_4)	Less Pref. (C_5)	New Transit (C_6)	*CenMetric*	*#decoy*	ε
QoS-caut.,wealthy (T_1)	0.000	0.000	0.000	0.026	0.030	0.031	0.277	2789	0.1
QoS-caut.,poor (T_2)	0.000	0.000	0.000	0.000	0.009	0.000	0.229	2706	0.1
QoS-ignr.,poor (T_3)	0.703	0.079	0.000	0.000	0.009	0.000	0.949	2710	0.3
Reach-caut.,wealthy (T_4)	0.000	0.000	0.001	0.011	0.022	0.031	0.262	3059	0.1
Domain-caut.,wealthy (T_5)	0.706	0.004	0.001	0.024	0.036	0.031	0.995	2635	0.1
Irrational (T_6)	0.717	0.079	0.003	0.029	0.033	0.031	1.000	2702	N/A

Table 6: *Central decoy deployment (game one)*: Comparing various censoring countries with profile T_1 and the same decoy budget ratio of $F/\rho_0 = 10^7$.

Country	Unreachable(C_1)	Blocked Domains(C_2)	Increased Length (C_3)	NVF (C_4)	Less Preferred (C_5)	New Transit (C_6)	*CenMetric*	*#decoy*	ε
China	0.000	0.000	0.000	0.026	0.030	0.031	0.277	2789	0.1
Venezuela	0.000	0.000	0.002	0.028	0.084	0.015	0.210	1450	0.1
Saudi Arabia	0.000	0.000	0.002	0.040	0.030	0.091	0.197	1853	0.1
Syria	0.000	0.000	0.003	0.009	0.002	0.000	0.101	544	0.1

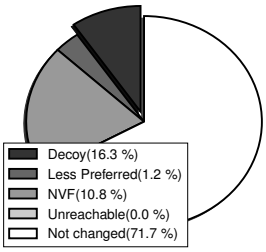

Figure 4: *Autonomous decoy deployment (game two)*: Impact on Chinese routes for a decoy service fee of $\gamma = 5.0$ and a T_1 censor profile.

6.5 Game Two Experiments

We use our analysis of Section 5 to converge to the *optimal decoy placement strategies* when the censors are taking *optimal censorship actions*. Similar to the previous game, we have performed our analysis for the four censoring countries in Table 4. We start our evaluation by discussing the results for China, followed by comparison of the results across countries.

How the routes change. As in game one, the routing decisions of the `censor` impact the Internet routes available to the `censor`'s users. For instance, as we see in Figure 4, for a service fee of $\gamma = 5.0$ charged by decoy ASes, the optimal routing strategy of the Chinese `censor` with a profile T_1 causes 19.9% of the routes to become NVF, and 3.5% to go through less-preferred routes. For this particular censor profile, even the best censorship strategy leaves 9.6% of the routes available to the Chinese users for decoy routing.

Impact of decoy service fee. Intuitively, higher service fees for decoy routing will incentivize more ASes to deploy decoy routing despite the risk of losing transit traffic due to `censor`'s re-routing actions. Figure 5 shows the impact of the service fee γ on the censorship metric and the costs to the Chinese `censor` with a T_1 (QoS-cautious, wealthy) profile. As expected, increasing the service fee improves censorship resistance as more ASes will decide to deploy decoy routers. Using this figure, one can estimate the achieved censorship resistance given particular values for the service fees charged by decoy ASes. Also, note that γ scales an AS's revenue from

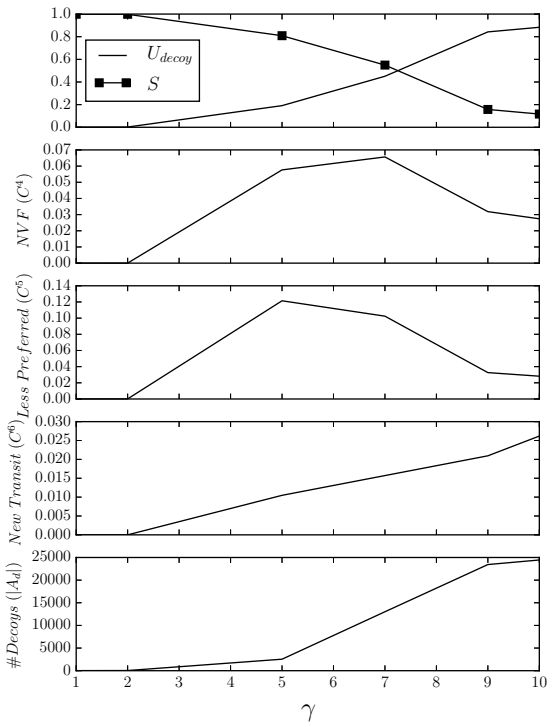

Figure 5: Impact of decoy service fee (γ) on the censorship metric and censorship costs (`censor` is China with a T_1 profile).

decoy routing with respect to its revenue from transit traffic. Therefore, a $\gamma = 5$ means that the decoy AS will charge for every MB of decoy traffic five times what she charges per MB of transit traffic.

Impact of censor profiles. As in game one, the optimal censorship strategy depends on the censor's profile, i.e., the preferences of `censor` between censorship performance and various cost metrics. Table 7 compares the cost metrics as well as the censorship metric for China based on different censorship profiles. Like game one, we see that the censor-

ship costs as well as the achieved censorship significantly vary across the profiles. For instance, comparing T_2 (QoS-cautions, poor) and T_3 (QoS-ignorant, poor) we see that if the Chinese `censor` does not care about the QoS offered to its clients, she can improve the censorship metric from 0.456 to 0.871. This, however, comes at the price of making 18.4% of all Internet routes becoming unreachable to the Chinese users.

Similar to the game one, an irrational censor (profile T_6) can achieve perfect censorship. In contrast to game one, in this case there will be no QoS degradation since no single AS will decide to deploy decoy routers as they will *all* be blocked by the irrational censor who does not care about QoS. By contrast, in game one the central decoy deployer always uses its decoy budget, therefore the irrational censor will suffer from significant QoS degradation.

Comparing censoring countries. Table 8 compares the censorship performance and cost metrics for the four censorship countries of Table 4, given a T_1 censor profile and a fixed decoy service fee. We have sorted the countries in the descending order of the censorship metric. Similar to the game one, the censorship enforced by various nation-state censors is proportional to their Internet connectivity.

7. FUTURE DIRECTIONS

We believe that our game-theoretic analysis is a significant step towards understanding the real-world feasibility of the recently proposed decoy routing circumvention approach. Our study can be extended in various directions. First, our analysis derived optimal decoy placements against individual nation-state censors. An interesting extension would be to derive the set of optimal decoy ASes that maximizes censorship circumvention against a collection of nation-state censors (as shown through our experiments, the optimal set of decoys differs for various nation-state censors).

A second direction for future work is to expand our game-theoretic model by considering the censored clients as players in the game, i.e., they can decide to whether or not use a decoy routing system comparing factors like the service fees and the offered QoS with other circumvention technologies like the paid VPNs.

Third, in our analysis we used comparative metrics to represent the monetary expenses of decoy deployment (i.e., F/ρ and γ). An interesting extension would be translate such metrics into actual dollar values based on real-world information on the costs of equipment, transit traffic, etc.

8. ACKNOWLEDGMENTS

We would like to thank anonymous reviewers and Reza Shokri for their feedback. This work was supported in part by NSF CAREER grant CNS-1553301.

9. REFERENCES

[1] E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. *SIAM Journal on Computing*, 38(4):1602–1623, 2008.

[2] J. Boyan. The Anonymizer: Protecting User Privacy on the Web. *Computer-Mediated Communication Magazine*, 4(9), Sept. 1997.

[3] The CAIDA UCSD [AS Relationships] - [Jan 2016]. http://www.caida.org/data/as-relationships/.

[4] J. Cesareo, J. Karlin, J. Rexford, and M. Schapira. Optimizing the placement of implicit proxies. https://www.cs.princeton.edu/~jrex/papers/decoy-routing.pdf, 2012.

[5] J. Cowie. Egypt Leaves the Internet. http://www.renesys.com/blog/2011/01/egypt-leaves-the-internet.shtml, January 2011. Online Article.

[6] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) protocol — version 1.2. Internet RFC 5246, 2008.

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

[8] D. Ellard, C. Jones, V. Manfredi, W. T. Strayer, B. Thapa, M. Van Welie, and A. Jackson. Rebound: Decoy routing on asymmetric routes via error messages. In *IEEE LCN*, 2015.

[9] S. Fujishige. *Submodular functions and optimization*, volume 58. Elsevier, 2005.

[10] R. Hausmann, C. A. Hidalgo, S. Bustos, M. Coscia, A. Simoes, and M. A. Yildirim. *The atlas of economic complexity: Mapping paths to prosperity*. MIT Press, 2014.

[11] A. Houmansadr, C. Brubaker, and V. Shmatikov. The Parrot is Dead: Observing Unobservable Network Communications. In *IEEE S&P*, 2013.

[12] A. Houmansadr, G. T. Nguyen, M. Caesar, and N. Borisov. Cirripede: circumvention infrastructure using router redirection with plausible deniability. In *ACM CCS*, 2011.

[13] 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.

[14] A. Houmansadr, E. L. Wong, and V. Shmatikov. No Direction Home: The True Cost of Routing Around Decoys. In *NDSS*, 2014.

[15] J. Jia and P. Smith. Psiphon: Analysis and Estimation. http://www.cdf.toronto.edu/~csc494h/reports/2004-fall/psiphon_ae.html, 2004.

[16] J. Karlin, D. Ellard, A. W. Jackson, C. E. Jones, G. Lauer, D. P. Mankins, and W. T. Strayer. Decoy routing: Toward unblockable internet communication. In *USENIX FOCI*, 2011.

[17] S. Khuller, A. Moss, and J. S. Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999.

[18] D. Kim, G. R. Frye, S.-S. Kwon, H. J. Chang, and A. O. Tokuta. On combinatoric approach to circumvent internet censorship using decoy routers. In *MILCOM*, 2013.

[19] J. Kleinberg and É. Tardos. *Algorithm design*. Pearson Education India, 2006.

[20] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *ACM SIGKDD*, 2007.

[21] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani. *Algorithmic game theory*, volume 1. Cambridge University Press Cambridge, 2007.

[22] B. Quoitin and S. Uhlig. Modeling the routing of an autonomous system with c-bgp. *Network, IEEE*, 19(6):12–19, 2005.

[23] M. Schuchard, J. Geddes, C. Thompson, and N. Hopper. Routing around decoys. In *ACM CCS*, 2012.

[24] A. Vetta. Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions. In *Foundations of Computer Science*, 2002.

[25] E. Wustrow, C. M. Swanson, and J. A. Halderman. TapDance: End-to-middle anticensorship without flow blocking. In *USENIX Security*, 2014.

[26] E. Wustrow, S. Wolchok, I. Goldberg, and J. A. Halderman. Telex: Anticensorship in the network infrastructure. In *USENIX Security*, 2011.

Table 7: *Autonomous decoy deployment (game two)*: Comparing the impact of different `censor` profiles on various cost metrics and the censorship success. Our `censor` is China and the service fee of decoy ASes is $\gamma = 7$.

Profile	Unreachable (C_1)	Unreach. Domains (C_2)	Inc. Length (C_3)	NVF (C_4)	Less Pref. (C_5)	New Transit (C_6)	*CenMetric*	*#decoy*
QoS-caut.,wealthy (T_1)	0.000	0.000	0.000	0.066	0.102	0.016	0.549	13018
QoS-caut.,poor (T_2)	0.000	0.000	0.000	0.000	0.078	0.000	0.456	13130
QoS-ignr.,poor (T_3)	0.184	0.013	0.000	0.000	0.146	0.000	0.871	92
Reach-caut.,wealthy (T_4)	0.000	0.000	0.002	0.067	0.128	0.016	0.579	12959
Domain-caut.,wealthy (T_5)	0.727	0.008	0.002	0.043	0.068	0.031	0.991	2769
Irrational (T_6)	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0

Table 8: *Autonomous decoy deployment (game two)*: Comparing various censoring countries with profile T_1 and the same decoy service fee of $\gamma = 7$.

Type	Unreachable(C_1)	Blocked Domains(C_2)	Increased Length (C_3)	NVF (C_4)	Less Preferred (C_5)	New Transit (C_6)	*CenMetric*	*#decoy*
China	0.000	0.000	0.000	0.066	0.102	0.016	0.549	13018
Saudi Arabia	0.000	0.000	0.000	0.174	0.105	0.114	0.310	11906
Venezuela	0.000	0.000	0.000	0.032	0.179	0.000	0.265	10840
Syria	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1058

APPENDIX

LEMMA 1. *The $\delta(\cdot)$ function defined in (2) is monotonic with the size of decoy set. That is, for two decoy placement sets $X, Y \subseteq \mathcal{A}_{free}$, if $X \subseteq Y$ then $\forall S_i, S_j : \delta_Y(R(S_i, S_j)) \geq \delta_X(R(S_i, S_j))$.*

PROOF. *We prove this by contradiction. Suppose there exists S_i, S_j such that $\delta_Y(R(S_i, S_j)) < \delta_X(R(S_i, S_j))$. Since $\delta(\cdot)$ only takes two values, this means that $\delta_X(R(S_i, S_j)) = 1$ and $\delta_Y(R(S_i, S_j)) = 0$. $\delta_X(R(S_i, S_j)) = 1$ means that*

$$X \cap R(S_i, S_j) \neq \emptyset. \tag{15}$$

On the other hand, $\delta_Y(R(S_i, S_j)) = 0$ means that

$$Y \cap R(S_i, S_j) = \emptyset$$

therefore:

$$\begin{aligned} X \cap Y \cap R(S_i, S_j) &= \emptyset \cap X \\ (X \cap Y) \cap R(S_i, S_j) &= \emptyset \ , \ X \subseteq Y \\ X \cap R(S_i, S_j) &= \emptyset \end{aligned} \tag{16}$$

which is in contradiction with (15) □

LEMMA 2. *U_{decoy} in (6) is a monotone submodular function [9].*

PROOF. *Since $U_{decoy} = B_{decoy}$, it suffices to prove that B_{decoy} is monotone submodular.*

We first show that B_{decoy} is submodular. We need to show [9] that for every two sets X and Y such that $X, Y \subseteq A_{free}$,$X \subseteq Y$ we have:

$$\forall x \in A_{free} \backslash Y : T \geq 0 \tag{17}$$

where $T = B_{decoy}(X \cup \{x\}) - B_{decoy}(X) - B_{decoy}(Y \cup \{x\}) + B_{decoy}(Y)$.

Using (3) we can write:

$$\begin{aligned} T = \sum_{AS_i \in \mathcal{A}_{cens}} \sum_{AS_j \in \mathcal{A}_{free}} ||AS_i|| \times ||AS_j|| \times \\ \big(\delta_{X\cup\{x\}}(R_{i,j}) - \delta_X(R_{i,j}) - \ \delta_{Y\cup\{x\}}(R_{i,j}) + \delta_Y(R_{i,j})\big) \end{aligned} \tag{18}$$

To prove that T in (18) is non-negative, it is enough to show that for each $R_{i,j}$, $L = \delta_{X\cup\{x\}}(R_{i,j}) - \delta_X(R_{i,j}) - \delta_{Y\cup\{x\}}(R_{i,j}) + \delta_Y(R_{i,j})$ is non-negative. From (2), $\delta(\cdot)$ can only be 0 or 1. Therefore, L can be negative only in two cases: $\delta_X(\cdot) = 1$ or $\delta_{Y\cup\{x\}}(\cdot) = 1$.

Case 1 ($\delta_X(\cdot) = 1$): We have that:

$$X \subseteq X \cup \{x\} \tag{19}$$

$$X \subseteq Y \subseteq Y \cup \{x\}. \tag{20}$$

Since $\delta(\cdot)$ is a monotone function (Lemma 1) we get:

$$\delta_X \leq \delta_{X\cup\{x\}} \tag{21}$$

$$\delta_X \leq \delta_Y \leq \delta_{Y\cup\{x\}}. \tag{22}$$

Therefore:

$$if \ \delta_X = 1 \Rightarrow \delta_{X\cup\{x\}} = \delta_Y = \delta_{Y\cup\{x\}} = 1 \tag{23}$$

$$\delta_{X\cup\{x\}}(.) - \delta_X(.) - \delta_{Y\cup\{x\}}(.) + \delta_Y(.) = 0 \tag{24}$$

so L is non-negative in case 1.

Case 2 ($\delta_{Y\cup\{x\}}(\cdot) = 1$): Given case 1, the only way L can be negative is that $\delta_{Y\cup\{x\}}(R_{i,j}) = 1$ and $\delta_{X\cup\{x\}}(R_{i,j}) = \delta_X(R_{i,j}) = \delta_Y(R_{i,j}) = 0$. We show by contradiction that this is not possible either. $\delta_{Y\cup\{x\}}(R_{i,j}) = 1$ means $((Y \cup \{x\}) \cap R_{i,j}) \neq \emptyset$. We also know $(Y \cap R_{i,j}) = \emptyset$ due to $\delta_Y(R_{i,j}) = 0$. Therefore, AS x is the decoy AS on the route. But, we have $\delta_{X\cup\{x\}}(R_{i,j}) = 0$, which is in contradiction with x being a decoy AS.

This concludes proving that $L \geq 0$, which in turn proves (17). Therefore, B_{decoy} is submodular.

Now, we prove that B_{decoy} is also a monotone function. From Lemma 1:

$$\forall S_i, S_j : \delta_Y(R_{i,j}) \geq \delta_X(R_{i,j}). \tag{25}$$

Therefore:

$$\sum_{AS_i \in \mathcal{A}_{cens}} \sum_{AS_j \in \mathcal{A}_{free}} \delta_Y(.) \geq \sum_{AS_i \in \mathcal{A}_{cens}} \sum_{AS_j \in \mathcal{A}_{free}} \delta_X(.) \tag{26}$$

Since $\forall S_i, S_j : ||AS_i|| \cdot ||AS_j|| \geq 0$, we can multiply it at both sides of (26):

$$\begin{aligned} \sum_{AS_i \in \mathcal{A}_{cens}} \sum_{AS_j \in \mathcal{A}_{free}} ||AS_i|| \cdot ||AS_j|| \delta_Y(.) \geq \\ \sum_{AS_i \in \mathcal{A}_{cens}} \sum_{AS_j \in \mathcal{A}_{free}} ||AS_i|| \cdot ||AS_j|| \delta_X(.) \end{aligned} \tag{27}$$

which proves that B_{decoy} is monotone. □