Randomized Security Patrolling for Link Flooding Attack Detection

Xiaobo Ma, Bo An, Mengchen Zhao, Xiapu Luo, Lei Xue, Zhenhua Li, Tony T. N. Miu, and Xiaohong Guan, *Fellow, IEEE*

Abstract—With the advancement of large-scale coordinated attacks, the adversary is shifting away from traditional distributed denial of service (DDoS) attacks against servers to sophisticated DDoS attacks against Internet infrastructures. Link flooding attacks (LFAs) are such powerful attacks against Internet links. Employing network measurement techniques, the defender could detect the link under attack. However, given the large number of Internet links, the defender can only monitor a subset of the links simultaneously, whereas any link might be attacked. Therefore, it remains challenging to practically deploy detection methods. This paper addresses this challenge from a game-theoretic perspective, and proposes a randomized approach (like security patrolling) to optimize LFA detection strategies. Specifically, we formulate the LFA detection problem as a Stackelberg security game, and design randomized detection strategies in consideration of the adversary's behavior, where best and quantal response models are leveraged to characterize the adversary's behavior. We employ a series of techniques to solve the nonlinear and nonconvex NP-hard optimization problems for finding the equilibrium. The experimental results demonstrate the necessity of handling LFAs from a game-theoretic perspective and the effectiveness of our solutions. We believe our study is a significant step forward in formally understanding LFA detection strategies.

Index Terms—Internet security, link flooding attack, security patrolling

1 Introduction

With the advancement of large-scale coordinated attacks (e.g., botnets), the adversary is shifting away from traditional distributed denial of service (DDoS) attacks against specific victim servers [1], [2], [3], [4], [5], [6], [7], [8], [9] to sophisticated DDoS 7attacks against critical Internet infrastructures. Link flooding attacks (LFAs) are such advanced DDoS attacks against critical links on the Internet [10], which have recently come into practice after attracting the academia for years. They are extremely powerful due to their ability in paralyzing a large regional network by congesting critical links surrounding it. For example, to degrade the connectivity of the anti-spam service Spamhaus, LFAs were employed by the adversary to flood a few links of major Internet exchange points in Europe and Asia, threatening to break down core Internet infrastructure with up to 300 Gb/s attack traffic [11].

As compared to traditional DDoS attacks against victim servers, LFAs exhibit new features. Specifically, LFAs provide an indirect, stealthy yet powerful choice for the adversary to scale attack traffic to unattainable levels to flood *all possible* links (and thus degrade the connectivity of nearby networks) [12]. LFAs are indirect in the sense that the adversary never directly attacks any host in a network, while degrading the connectivity of the network (or the host). To achieve this, the adversary coordinates enormous traffic flows, originating from a large number of distributed compromised machines (e.g., bots) to many different machines (e.g., publicly-accessible servers) near or within the target network, so that all traffic flows cross and congest only a few selected links surrounding the network. LFAs are stealthy and undetectable by the target network because individual traffic flows (e.g., accessing a publicly-accessible server) are indistinguishable from legitimate ones.

Due to the new features, the detection of LFAs differs from that of traditional DDoS attacks, which relies on server-side passive traffic monitoring. To defend against such attacks, several router-based approaches have been proposed [13], [14], [15], [16]. Despite the promising prospects, their effectiveness may be limited because they cannot be widely deployed to the Internet immediately. In contrast, non-cooperative measurement techniques deployed at terminal hosts can be explored to detect LFAs via active probing [17], without the need to modify current Internet infrastructures (e.g., configuring routers) requiring

- *X. Ma is with the MOE Key Lab for Intelligent Networks and Network Security, School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China. He is also with Shaanxi Province Key Laboratory of Computer Networks, Xi'an Jiaotong University, China. E-mail: xma.cs@xjtu.edu.cn.*
- *B. An and M. Zhao are with the School of Computer Engineering, Nanyang Technological University, 639798, Singapore. E-mail: boan@ntu.edu.sg, zhao0204@e.ntu.edu.sg.*
- *X. Luo and L. Xue are with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. E-mail: {csxluo, cslxue}@comp.polyu.edu.hk.*
- *Z. Li is with the School of Software, Tsinghua University, Beijing 100084, China. E-mail: lizhenhua1983@gmail.com.*
- *T.T.N. Miu is with NexusGuard Limited, Hong Kong. E-mail: tony.miu@nexusguard.com.*
- *X. Guan is with Shenzhen Research School, Xi'an Jiaotong University, Shenzhen, China. He is also with the MOE Key Lab for Intelligent Networks and Network Security, School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China. E-mail: xhguan@xjtu.edu.cn.*

Manuscript received 27 Dec. 2016; revised 9 Nov. 2018; accepted 28 Dec. 2018. Date of publication 14 Jan. 2019; date of current version 9 July 2020. (Corresponding author: Xiapu Luo.)
Digital Object Identifier no. 10.1109/TDSC.2019.2892370

administrative privileges. Apparently, they would be a more practical countermeasure in immediate response to LFAs.

These non-cooperative techniques actively measure the network performance (e.g., packet loss rate, RTT, available bandwidth) along a path (i.e., a sequence of links) originating from a source host (where a probing agent is deployed) to a destination host (e.g., a publicly-accessible server), and the abnormal performance degradation along the path (e.g., high packet loss rate, abrupt changes) will indicate the occurrence of LFAs targeting at least one link along this path. To further detect the specific links that account for the performance degradation, hop-by-hop measurement techniques for localizing a path's bottleneck can be leveraged [18]. Detecting the paths and the links suffering from LFAs whenever possible enables upstream providers to launch responsive countermeasures to mitigate LFAs.

Despite the capability of employing non-cooperative measurement techniques to detect LFAs along a specific path, we are facing a tremendous challenge when deploying these techniques. That is, we cannot predict the target links the adversary selects to attack (and correspondingly where to deploy detection methods), given that all links under protection might be targeted by LFAs [12]. At first glance, this challenge can be easily addressed by deploying detection methods covering all the paths (links). Unfortunately, this is *non-trivial* because of the practical barriers below. First, to get accurate measurements, the probing of path (link) performance originating from a single host should not concurrently cover many paths [19], due to the potential mutual interference. For example, two packet trains for measuring the available bandwidth of two different paths, if concurrently issued from the same host, would be interleaved with each other, thereby damaging the back-to-back nature of a packet train and causing mutual interference. As pointed out and verified by Croce et al. in [20], such mutual interference between concurrent packet trains may lead to rather inaccurate measurement. Second, bursts of a large probing outdegree (resp. indegree) from a source and (resp. towards a destination) may result in self-congestion [21], or trigger security alerts from existing monitoring systems [22], consequently making the probing activities inaccurate or even blocked. Third, to cover more paths (links), the defender has to deploy probing agents in more machines that are geographically dispersed. However, such a strategy is cost prohibitive, not to mention that the agents cannot be deployed arbitrarily due to resource constraints [23], [24].

As a result of the above barriers, the defender needs to initiatively avoid mutual interference [25] and takes into account the constraint of limited resources. Consequently, the defender can only monitor a subset of all the paths and the links during a period. On the other side, since the adversary might *illegally* possess abundant compromised machines all over the network, he[1] has the agility to coordinate massive traffic flows originating from these machines to attack many (and potentially different) links (at different time periods) [12], [26]. Therefore, to make the LFA detection methods deployable, it is highly desirable to design optimal detection strategies subject to practical resource constraints. Such optimal detection strategies are expected to have the following properties. First, they should be dynamic rather than static because if static strategies are adopted, a number of links would be left unprotected, and consequently being attacked without the defender's awareness. Nevertheless, adopting dynamic strategies can deter the adversary, and let him feel that the detection might be everywhere at the time of attacks. Second, dynamic strategies should be designed to maximize the defender's expected utility with limited resources and take into account the adversary's behavior.

With these expected properties in mind, we propose a randomized game-theoretic approach working like real-life security patrolling to study LFA detection strategies. Achieving this approach entails addressing several key issues. First, a mathematical model is desired to describe the confrontation between the defender and the adversary. Second, we need to find a way capable of quantifying the adversary's behavior to a wide extent. Third, once successfully formulated, the problem should be effectively solved.

Our major contributions are summarized below.

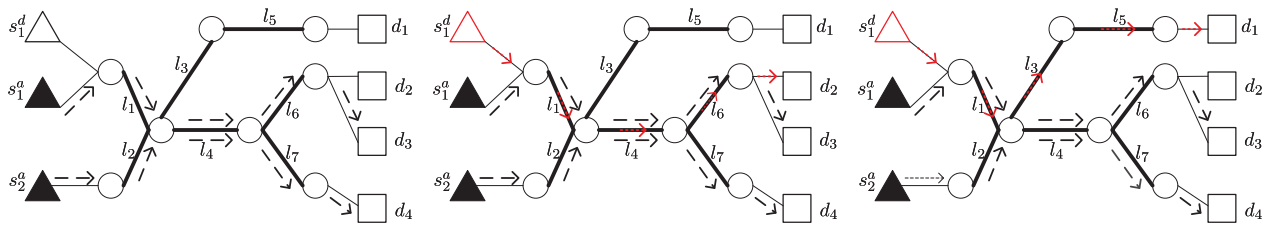
- We are the *first* to formulate the problem of optimal LFA detection as a Stackelberg security game and explore the LFA detection strategies. We consider a well-informed adversary who can perfectly observe the detection strategies, or diligently learn the detection strategies after conducting surveillance. The threat model of an agile and well-informed adversary makes the defender robust against a strong adversary by design. (Section 4)
- We design a randomized mixed-detection strategy in consideration of the adversary's behavior to maximize the defender's expected utility. Meanwhile, best and quantal response models are leveraged to characterize the adversary's behavior in response to the defender's strategy, allowing us to characterize a continuous wide range of the adversary's rationality from zero to infinity. (Sections 5 and 6)
- To tackle the NP-hard nonlinear and nonconvex optimization problems raised from our model for finding the equilibrium, we transform them into standard mixed-integer linear programming problems using techniques such as conditional judgment elimination, binary search and piecewise linear approximation. Both synthetic and real-world experiments demonstrate that our randomized mixed-detection strategy, compared to uniform-detection and best-detection strategies, can effectively maximize detection utility given limited security resources. (Section 7)

Roadmap. Section 2 illustrates LFAs. Section 3 describes the problem. We formulate the problem in Section 4, and solve it in Sections 5 and 6. We experiment in Section 7, discuss in Section 8, review the literature in Section 9, and finally conclude in Section 10.

2 Background of Link Flooding Attack

Fig. 1a demonstrates a simplified example of LFAs. In this example, triangles s_1^a and s_2^a are compromised machines (i.e., bots), and the adversary wants to attack target link l_4. To this end, he instructs s_1^a and s_2^a to send traffic to publicly-accessible servers d_3 and d_4, respectively. In practice, he can instruct

1. We use "he/him/his" and "she/her/her" to represent the adversary/attacker and the defender respectively for ease of representation.



(a) link flooding attack (b) successful detection (c) unsuccessful detection

Fig. 1. Simplified examples demonstrating: (a) an LFA against l_4, (b) a successful detection scenario, and (c) an unsuccessful detection scenario. Continuous thick lines from l_1 to l_7 are all possible links under attack. The hollow triangle s_1^d denotes a source host owned by the defender, while the black shaded ones s_1^a and s_2^a owned by the adversary. Square boxes from d_1 to d_4 denote publicly-accessible servers. The dotted black arrows beside links denote traffic flows congesting l_4 issued by the adversary, while the dotted red arrows over links denote the probing from the defender.

more bots to send traffic flows that cross l_4 to more public servers, rendering all the traffic flows aggregated at target link l_4 and in turn congesting l_4. From the standpoint of d_3 and d_4, the attacking traffic flows originating from s_1^a and s_2^a are indistinguishable from legitimate ones [10]. Therefore, LFAs targeting l_4 cannot be perceived by d_3 and d_4, not to mention other nodes (e.g., d_2, s_1^d) where attacking traffic flows neither arrive nor originate. However, the network area covering d_2, d_3 and d_4 will be unreachable once l_4 is congested, hence making resources of d_2, d_3 and d_4 unavailable to their intended users (e.g., s_1^d) on the Internet.

The above example demonstrates that LFAs can effectively cut off the connections of a target network, without being detected by passive traffic monitoring at terminal hosts or the perimeter of the target network. Specifically, the adversary first selects persistent links that connect the target network area to the Internet, and then instructs massive bots to generate legitimate traffic flows between bots and a bunch of publicly-accessible servers, for crossing and thus congesting the selected links. If the paths among bots cover the selected links, the adversary can also coordinate traffic flows among these bots for the same goal.

To diagnose the disconnection and performance degradation of a network, non-cooperative measurement methods based on active probing were proposed to detect LFAs. As Fig. 1b demonstrates, the defender can probe the path (i.e., a sequence of links) originating from s_1^d to d_2, to detect whether LFAs occur along the path, by performing end-to-end path measurement (e.g., packet loss rate, RTT, available bandwidth) [17]. The anomaly (e.g., abrupt degradation) of the path performance indicates the occurrence of LFAs targeting at least one link (i.e., l_1, l_4, l_6) along this path. To further locate the specific links that account for the performance degradation along the path, hop-by-hop measurement techniques for detecting the location of a path's bottleneck can be leveraged [18]. In our example, to find the link whose performance degrades the most, s_1^d can choose to perform hop-by-hop path performance measurement that covers paths l_1, $l_1 \rightarrow l_4$, and $l_1 \rightarrow l_4 \rightarrow l_6$, respectively. Interested readers can refer to [17] for technical details.

3 Problem Description

Let $G = (V, E)$ be a network topology *visible to both* the adversary and the defender. Nodes in V denote end hosts, public servers or routers, and edges in E denote IP links. We next detail nodes and edges as demonstrated in Fig. 1.

The set of nodes can be decomposed into four types of non-empty sets: (1) S^d is the set of source nodes owned by the defender (e.g., the end host $s_1^d \in S^d$). Probing traffic flows in charge of LFA detection originate from source nodes in S^d; (2) S^a is the set of source nodes under the control of the adversary (e.g., end hosts $s_1^a, s_2^a \in S^a$). Attack traffic flows for congesting a target link originate from source nodes in S^a; (3) D is the set of destination nodes (e.g., public servers $d_1, \ldots, d_4 \in D$ with open ports), where both probing traffic flows and attack traffic flows are directed towards ($S^d \cap S^a \cap D = \emptyset$); (4) All remaining nodes in the network (i.e., $V \setminus \{S^d \cup S^a \cup D\}$), including routers (i.e., circles) and end hosts that do not belong to $S^d \cup S^a \cup D$.

The set of edges can be decomposed into two non-empty sets: (1) The set of links adjacent to at least one end host (e.g., edges with continuous thin lines), which we call *terminal links*; (2) The set of links only adjacent to routers (e.g., edges with continuous thick lines), which we call *target links* and denote the set of these links by L. LFAs make a feature of attacking target links, rather than terminal links targeted by traditional DDoS attacks. Therefore, L is the protection space where the defender performs security patrolling, and meanwhile the attack space which the adversary targets.

When the adversary attempts to congest a target link $l \in L$ by LFAs, he chooses a set of nodes in S^a and a set of nodes in D. Then, he coordinates massive traffic flows originating from nodes in S^a to nodes in D. All these traffic flows cross target link l, and consequently l would be congested. The traffic flows are routed in G by underlying protocols (e.g., BGP, OSPF), whereby the path from any node in S^a to any node in D can be determined. Existing tools like `Paris Traceroute` could get the paths between the source and the destination [27]. Therefore, to congest l, the adversary would choose node pairs between S^a and D based on the routing protocol. For example, the adversary in Fig. 1a attacks l_4 using node pairs (s_1^a, d_3) and (s_2^a, d_4). Note that, to evade detection, LFAs may limit the period of flooding l to avoid triggering route changes [10].

To detect LFAs, the defender chooses a set of nodes in S^d and a set of nodes in D. Then, for each node $s^d \in S^d$, she deploys a probing agent, and assigns at most $M(s^d)$ nodes in D to probe during a period. From each node s^d, at most $M(s^d)$ node pairs are formed. Each node pair typically corresponds to a path. Through the path, the defender can detect LFAs by sending probing traffic flows originating from s^d. Since different node pairs may have the same paths (e.g., (s_1^d, d_2) and (s_1^d, d_3) in Fig. 1a) that cover the same sequence of target links, the number of paths originating from node s^d is upper bounded by the number of corresponding node pairs. The reasons that each node in S^d is assigned a limited

TABLE 1
Summary of Notations

Notation	Definition
l_i	the ith target link
L	set of target links
I	total number of target links
S^d	set of probing sources of the defender
s^d	probing source $s^d \in S^d$
S^a	set of sources (i.e., bots) of the adversary
s^a	source $s^a \in S^a$ of the adversary
D	set of destinations (i.e., publicly-accessible servers)
d	destination $d \in D$
N_i	number of paths crossing l_i
$M(s_d)$	maximum number of probing paths from s^d
$M(d)$	maximum number of probing paths to d
R_i^d	defender's reward on protecting l_i if attacked
P_i^d	defender's penalty on not protecting l_i if attacked
R_i^a	adversary's reward on successfully attacking l_i
P_i^a	adversary' penalty on unsuccessfully attacking l_i
J	total number of pure strategies of the defender
Γ_j	the jth defender's pure strategy
$\|\Gamma\|_{\max}$	maximum number of links covered by $\Gamma_j, \forall j$
a_j	probability that the defender chooses Γ_j
$\mathbf{a}$	$(a_1, a_2, \ldots, a_J)$, mixed-detection strategy
B_i	probability that the best response adversary attacks l_i
λ	quantal response adversary's rationality, $\lambda \in [0, +\infty)$
$Q_i(\lambda)$	probability that the quantal response adversary attacks l_i
A_{ij}	coverage indicator of Γ_j on l_i; $A_{ij} = 1$ or $A_{ij} = 0$
x_i	marginal probability that defender protects l_i
$U_i^d(x_i)$	defender's expected detection utility on protecting l_i
$U_i^a(x_i)$	adversary's expected attack utility on attacking l_i
U^d	defender overall utility $\sum B_i U_i^d(x_i)$ or $\sum Q_i(\lambda) U_i^d(x_i)$
U^a	adversary overall utility $\sum B_i U_i^a(x_i)$ or $\sum Q_i(\lambda) U_i^a(x_i)$

number of nodes in D during a period has been described (e.g., mutual interference of concurrent measurement sessions, self-congestion, security alerts) in Section 1. Similarly, each node $d \in D$ is assigned at most $M(d)$ nodes in S^d during a period. However, the number of destinations publicly accessible by the defender (i.e., $|D|$) is far larger than that of probing agents deployed by the defender (i.e., $|S^d|$). This makes the constraint regarding $M(d)$ *ignorable* in most cases. One can simply remove those probing strategies in which constraints regarding $M(d)$ are not satisfied. The example in Fig. 1b shows a successful detection scenario since the path (l_1, l_4, l_6) covers l_4, enabling the performance measurement of l_4, while Fig. 1c depicts an unsuccessful one because the path (l_1, l_3, l_5) fails to cover l_4.

Since the adversary often illegally controls a large-scale compromised machines (e.g., bots) in G, any of the target links in L might be attacked by LFAs. These machines can be instructed to send massive traffic flows to public servers in D to congest the target links. Besides publicly-accessible servers in D, the adversary may also deploy his own "public" servers to coordinate the massive traffic flows [28]. On the other hand, it is hard, if not impossible, for the defender to cover all target links in L, due to the practical resource constraints, meaning that the defender can only cover a subset of paths (and links) during a period.

Taking advantage of limited security resources, the defender seeks to perform security patrolling using randomized detection strategies. She first deploys the detection strategy in advance, and then waits for attacks from the adversary. The deployed strategy is desired to be optimally randomized, relying on the design by the defender in consideration of the adversary's response. Meanwhile, to design the optimal attack strategy, the adversary's response also needs to take into account the defender's deployed randomized detection strategy, which may be exposed to the adversary with surveillance capability. Therefore, to design better detection and attack strategies, the defender and the adversary have to take into account each other's behavior.

This paper focuses on theoretically investigating LFA detection deployment strategies rather than designing new detection schemes against LFAs along a specific path. We, therefore, consider the underlying detection scheme as a blackbox that can be directly called by security patrolling with the guidance of our carefully designed strategies. Table 1 summarizes major notations throughout our paper.

4 A Game-Theoretic Formulation

The problem of optimal LFA detection strategies can be modeled as "attacker-defender" security games (a.k.a. Stackelberg security game) [29], [30], [31], [32]. The defender acts as the leader, while the adversary is the follower. The leader commits to a strategy, and the follower then optimizes his reward considering the strategy chosen by the leader.

4.1 Threat Model: Agile and Well-Informed Adversary

We consider a threat model wherein the adversary is agile and "well-informed". The former means the adversary has the agility to attack all possible (potentially different) links (at different time periods by shifting the target links), while the latter allows the adversary to perfectly observe the detection strategies, or learn them by conducting surveillance. This threat model with a strong adversary is the *worst case* for the defender, where the adversary could react based on the acquired knowledge of the detection strategies. Note that it is our design choice to consider very strong adversary capability. Below we will justify the design.

Specifically, the well-informed adversary has two major capabilities. First, he has full knowledge of the network topology. Discovering network topologies has been widely studied, and can be conducted in a lot of well-established passive and active ways [33], [34]. For example, iPlane, a scalable service providing accurate predictions of Internet path performance, performs traceroutes from PlanetLab nodes daily to map the Internet's topology [35], [36]. Second, the adversary can perfectly observe the defender's strategies after conducting surveillance, or learn the defender's strategies. In our context, the adversary can learn the probability distribution of the defender choosing pure strategies, but cannot know exactly which pure strategy will be chosen at the time of scheduled attacks. This coincides with the situation that the adversary faces, i.e., the adversary can hardly determine whether a link is under the defender's protection during a certain period, even though he knows the probability that the defender protects the link.

Our assumption that the adversary is able to perfectly observe the defender's strategies after conducting surveillance, or learn the defender's strategies, seems to be too idealistically at first glance. Such an assumption is reasonable in our problem for three reasons.

First, our primary goal is to design a system defending Internet links against a sophisticated adversary. Hence, assuming an adversary with strong capability allows us to achieve our primary goal.

Second, the adversary normally can only acquire *limited knowledge* about the defender's strategies in practice, not strictly consistent with our assumption that the adversary can perfectly know the defender's strategies. Nevertheless, Blum et al. have analytically demonstrated that, in zero-sum Stackelberg security games, the defender that simply keeps optimizing against the adversary with full knowledge of the defender's strategies, is meanwhile almost optimal against the adversary with limited surveillance, i.e., sampling a reasonable (and usually very small) number of observations [37]. Precisely, according to [37], when optimizing against the (ideal) adversary with full knowledge of the defender's strategies, a probing agent could achieve a sub-optimal utility that is at most $O(\sqrt{\ln(dk)/k})$ less than the utility when optimizing against the (actual) adversary with limited surveillance, where the utility specifically refers to the (normalized) relative utility ranging between -1 and 1 (see Section 7.3 for details). Here, d denotes the number of paths that the probing agent can cover simultaneously, and k is the number of observations that the adversary actually has. Since d is small for accurate non-cooperative measurement, such utility loss decreases rather quickly as k grows.

Third, for the adversary who knows the defender adopting security games for detecting LFAs, he also has an opportunity to mathematically solve (instead of observing) the defender's strategies from the defender's perspective via security games. To solve the defender's strategies, the adversary requires the topology and the link importance associated with the target network under protection, which are not difficult to acquire.

Generally, the adversary could conduct surveillance in two ways:

Passively Observing. First, since probing techniques have unique patterns [38], the adversary can set up decoy servers or compromise routers to reveal ongoing probing. Second, public measurement platforms [39], [40], [41], [42] usually provide detailed information of their infrastructures and the measurement results online, thus easing the surveillance. Third, there is a possibility of invading the defender's server for the adversary (though *pretty small*), thereby leaking the full knowledge of the defender's strategies.

Actively Testing. By active attack testing, the adversary can learn from the defender's reaction, such as route changes. However, such active attack testing would render the adversary always one step behind the defender, because it is too late for the adversary to respond after route changes. The only way to take advantage of active attack testing is to cross-validate the defender's strategies learned from passively observing beforehand so to improve the way that the defender performs passively observing.

In our model, to deal with the agile and well-informed adversary, the defender takes into account the adversary's awareness of her detection strategies by design. Such a design makes the defender robust against a strong adversary.

4.2 Player Strategies

In the Stackelberg security game, the defender commits to a randomized *mixed-detection* strategy, wherein the defender chooses a pure detection strategy from her strategy set with a certain probability distribution. Essentially, a mixed strategy is an assignment of a probability to each pure strategy. The adversary then conducts surveillance of the mixed-detection strategy, and in turn responds with a pure attack strategy from his strategy set. Here, a pure strategy provides a complete definition of how a player would play a game for any situation he or she could face, and a player's strategy set is the set of pure strategies.

4.2.1 Defender

The defender's strategy set is a set of pure strategies, each of which, say Γ, depends on the mapping relation between probing sources S^d and destinations D. Specifically, each probing source $s^d \in S^d$ can choose to probe $M(s^d)$ destinations at most, and each destinations $d \in D$ can serve up to $M(d)$ probing sources. Consequently, different choices regarding which destinations to probe (subject to connection degree constraints $M(s^d)$ and $M(d)$) of s^d in S^d result in different defender's pure strategies (and thus different sets of links). Therefore, a defender's pure strategy Γ is a set of links covered by one mapping relation between S^d and D.

4.2.2 Adversary

The adversary aggregates traffic flows to attack a few selected target links each time. A pure strategy of the adversary is a set of target links that the adversary may select to attack. For simplicity, we consider that each pure strategy contains one link $l \in L$. Note it can be easily extended to the case where each pure strategy contains a few more links when the number of core links is relatively small (see Section 8).

When the adversary decides to attack l, the probability that the defender can detect LFAs against l depends on the marginal probability that the defender's pure strategies cover l, denoted by x. The larger x is, the more likely for the defender to capture the abrupt (abnormal) changes of link performance that indicate the presence of LFAs against l, hence more chances to detect the attack. The probability of detecting LFAs against l can be measured by x. For example, when the defender plays a pure strategy Γ with probability one and the adversary performs LFAs against l, the attack can be detected with probability one whenever $l \in \Gamma$, and the adversary succeeds in the attack when $l \notin \Gamma$.

4.3 Utility Functions

Consider that target link l_i is attacked. If l_i is covered by the defender's (adopted) pure strategy, the defender receives reward R_i^d. Otherwise, the defender receives penalty P_i^d. Similarly, the adversary receives penalty P_i^a in the former case, and reward R_i^a in the latter case. In practice, the reward that the defender protects a specific link can be assigned based on the asset importance of the link (e.g., path coverage, capacity), and so does the penalty that the adversary attacks a link [12], [43]. Selecting different asset importance measures does not affect the basic game-theoretic model.

We further consider the marginal probability x_i that the defender's pure strategies cover l_i. That is, x_i denotes the defenders' probability to protect l_i. The defender's expected utility on protecting l_i can be expressed as:

$$U_i^d(x_i) = x_i R_i^d + (1 - x_i) P_i^d, \tag{1}$$

and the adversary's expected utility on attacking l_i is:

$$U_i^a(x_i) = x_i P_i^a + (1 - x_i) R_i^a. \quad (2)$$

Properties $R_i^d > P_i^d$ and $R_i^a > P_i^a$ for $\forall i$ hold because they respectively guarantee that $U_i^d(x_i)$ and $U_i^a(x_i)$ strictly increase and decrease w.r.t. x_i, ensuring that investing security resources benefits the defender and restricts the adversary.

Let Γ_j denote the jth pure defender strategy, and A_{ij} denotes the coverage indicator of Γ_j on l_i, where $A_{ij} = 1$ when $l_i \in \Gamma_j$, and $A_{ij} = 0$ otherwise. Let I be the total number of target links, J be that of the pure strategies of the defender. We denote the probability of the defender choosing Γ_j by a_j, and $\sum_{j=1}^{J} a_j = 1$. The marginal probability x_i for the defender to protect l_i can be calculated by:

$$x_i = \sum_{j=1}^{J} a_j A_{ij}, \quad i = 1, 2 \ldots, I. \quad (3)$$

The summation of x_i denotes the expectation of the number of links that the defender can protect simultaneously, upper bounded by $|\Gamma|_{\max}$, i.e., the maximum number of links covered by Γ_j, $\forall j$. We denote $(a_1, a_2, \ldots, a_J)$ by $\mathbf{a}$, and $(x_1, x_2, \ldots, x_I)$ by $\mathbf{x}$, where $\mathbf{x}$ is determined by $\mathbf{a}$.

Since the defender's utility considers the adversary's response models, solving the optimal strategy relies on the adversary's response models. Meanwhile, the adversary with different response models maximizes his utility after observing the defender's strategy. Eventually, the defender 7and the adversary reach an *equilibrium* where each player's strategy is optimal given the strategy of the other player.

To find the equilibrium, in Sections 5 and 6, we will formally define the players' objective functions. Particularly, the defender will design the mixed-detection strategy (i.e., $\mathbf{a}$) maximizing her utility, as is formally presented below.

> Mixed-Detection Strategy: play Γ_j with probability a_j where $a_j \in \mathbf{a}$ is the solution maximizing the defender's utility (defined in problem P1/P3 in Sections 5/6), $j = 1, 2, \ldots, J$

4.4 Justification for Using Stackelberg Security Game

The Stackelberg security game is played sequentially between two players, i.e., the defender and the adversary. The defender, as the leader, moves first and commits to a mixed strategy. The adversary, as the follower, then observes the defender's strategy and optimizes his reward. Such a security game model fits our problem for three reasons.

First, detecting LFAs requires the defender to persistently and routinely perform monitoring before the attack occurs. Therefore, the sequential game where the defender moves before the adversary meets our requirement. However, the defender has to figure out its (optimal) strategies before moving (i.e., performing security patrolling), in consideration of its strategies (once adopted) being observed by the adversary. In other words, it is the defender herself rather than the adversary who needs "solving the defender's strategies", and it is the adversary rather than the defender who needs to observe the defender's strategies. Note that "observing the defender's strategies" means that the adversary knows the probabilities of the defender' choosing a pure strategy, yet without the knowledge of which specific strategy the defender would adopt at the time of an adversary's scheduled attack.

Therefore, "solving the defender's strategies" and "observing the defender's strategies" are conducted by the defender and the adversary respectively. Moreover, the defender solves its strategies by maximizing its utility function against the adversary who perfectly observes its strategies (i.e., strong surveillance capability), thereby achieving mathematically reliable strategies. By doing so, the defender can accomplish almost optimal utility in zero-sum Stackelberg security games, even when the adversary has limited surveillance capability [37].

Second, due to resource constraints, the defender cannot execute all pure strategies simultaneously. Moreover, she cannot know which target the adversary would attack. Thus, the defender typically selects one pure strategy with a probability distribution over all pure strategies, termed as a mixed strategy. In this way, even if the adversary can learn the probability distribution of the defender choosing pure strategies, he cannot know exactly which pure strategy the defender will choose at the time of scheduled attacks, thereby deterring the adversary. Also, it has been proved in [44] that if the commitment to a mixed strategy is possible, then the (optimal) commitment never hurts the leader (i.e., the defender), and often helps.

Third, as a best response to the defender committing to a mixed strategy, the adversary has two choices to maximize his utility. One is to do the mixed strategy as well, i.e., selecting one pure strategy (e.g., attacking one link) with a probability distribution over all pure strategies. The other is to select one pure strategy among all pure strategies. At first glance, the former would be better for the adversary since it is randomized. However, if a mixed strategy is a best response, then each of the pure strategies involved in the mixed strategy must itself be a best response. Particularly, each must yield the same expected utility. Otherwise, the mixed strategy would not be a best response if the probabilities of selecting pure strategies with lower expected utilities are dropped and assigned to the probabilities of selecting pure strategies with higher expected utilities, hence resulting in a contradiction. Interested readers could refer to [45] for a detailed proof. Such a lesson indicates that, as a best response, the adversary only needs to find a pure strategy maximizing his expected utility.

One may argue that, once the defender exactly predicts the pure strategy that the adversary would adopt, she can deterministically select a pure strategy that can detect the attack by violating her commitment to a mixed strategy. Nevertheless, violating the commitment and deterministically selecting pure strategies would make the defender fail to be better off in the long run, since in this case the adversary could evade the detection.

The expected utilities of both the defender and the adversary are calculated according to the probabilities of the adversary attacking each link, the probabilities of the defender protecting each link, and link importance. For the defender

committing to a mixed strategy and the best response adversary selecting one pure strategy (i.e., one link) to perform the attack, both of their expected utilities are determined by the attacked link, the importance of the attacked link, and the probability of the defender protecting the attacked link. The defender does not benefit from protecting the links that are not attacked, while the adversary only benefits from attacking the selected link. Please refer to (6) in Section 5 for how to calculate the defender's expected utility, where the adversary's expected utility can be calculated through replacing R_i^d and P_i^d by P_i^a and R_i^a, respectively. In the Stackelberg security game, a mixed strategy equilibrium is guaranteed to exist in finite games [44].

When conducting security patrolling against LFAs in practice, we can re-initiate the Stackelberg security game if no attack is detected in the last round. In each round, the defender chooses a pure strategy from her strategy set according to the probability distribution of the mixed strategy. The time duration of each round is lower bounded by the time for the defender to execute her selected pure strategy, i.e., the time to measure the paths (or links) belonging to the pure strategy. Moreover, the defender enters the next round to select a new pure strategy to execute upon finishing the current one if no attack is detected. If an attack is detected, the defender will concentrate on tracking the attack. An alternative approach is to invoke an attack tracking module and meanwhile the defender keeps patrolling.

5 Optimal Detection Strategies against A Best Response Adversary

Let us first consider a perfectly rational adversary who takes the best response strategy. Such an adversary will attack the target link maximizing his own expected utility. Thus, the probability that he chooses to attack target l_i is:

$$B_i = \begin{cases} 1 & U_i^a(x_i) \geq U_j^a(x_j) \text{ for } \forall j = 1, 2, \ldots, I, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Simultaneously, the defender selects an optimal mixed (i.e., randomized) strategy in consideration of the perfectly rational adversary's best response behavior model, where the adversary breaks ties in favor of the defender [29]. The defender maximizes her detection utility as:

$$\max_{\mathbf{a}} \sum_{i=1}^{I} B_i U_i^d(x_i). \tag{5}$$

Substituting (1), we rewrite (5) as:

$$\max_{\mathbf{a}} \sum_{i=1}^{I} B_i \big(x_i R_i^d + (1 - x_i) P_i^d\big). \tag{6}$$

In the LFA detection problem, there are constraints for the defender to assign security resources. As interpreted in previous sections, not all possible subsets of links can be feasible strategies of the defender, and the marginal coverage ratio x_i on target l_i is also restricted. To calculate the defender's optimal detection strategy against a best response adversary with resource assignment constraints, the following optimization problem P1 needs to be solved.

$$\mathbf{P1}: \begin{cases} \max_{\mathbf{a}} \sum_{i=1}^{I} B_i \big(x_i R_i^d + (1 - x_i) P_i^d\big), \\ \text{subject to } x_i = \sum_{j=1}^{J} a_j A_{ij} \text{ for } \forall i, \\ \quad \sum_{j=1}^{J} a_j = 1, \\ \quad 0 \leq a_j \leq 1 \text{ for } \forall j. \end{cases} \tag{7}$$

P1 *cannot* be directly solved since B_i in (4) contains the "if-then" logical relationship (i.e., if $U_i^a(x_i) \geq U_j^a(x_j)$ then $B_i = 1$; otherwise $B_i = 0$). After eliminating the relationship, we can equivalently transform P1 into problem P2.

$$\mathbf{P2}: \begin{cases} \max_{\mathbf{a}} V, \\ \text{subject to} \quad x_i = \sum_{j=1}^{J} a_j A_{ij} \text{ for } \forall i, \\ \quad \sum_{j=1}^{J} a_j = 1, \\ 0 \leq a_j \leq 1 \text{ for } \forall j, B_i \in \{0, 1\} \text{ for } \forall i, \\ V - \big(x_i R_i^d + (1 - x_i) P_i^d\big) \leq (1 - B_i) Z \text{ for } \forall i, \\ K - \big(x_i P_i^a + (1 - x_i) R_i^a\big) \leq (1 - B_i) Z \text{ for } \forall i, \\ K - \big(x_i P_i^a + (1 - x_i) R_i^a\big) \geq 0 \text{ for } \forall i. \end{cases} \tag{8}$$

Proof of the Equivalence of P1 and P2. The aim of P1 is to maximize $\sum_{i=1}^{I} B_i (x_i R_i^d + (1 - x_i) P_i^d)$, which equals $(x_{i_{\text{attack}}} R_{i_{\text{attack}}}^d + (1 - x_{i_{\text{attack}}}) P_{i_{\text{attack}}}^d)$ according to (4). Here, i_{attack} is the index of the target link that the best response adversary selects to attack, and thus we have $B_{i_{\text{attack}}} = 1$ and $U_{i_{\text{attack}}}^a(x_{i_{\text{attack}}}) \geq U_j^a(x_j)$ for $\forall j = 1, 2, \ldots, I$. Therefore, P1 is equivalent to ① ensuring that the adversary attacks $l_{i_{\text{attack}}}$ and achieves the largest attack utility; ②when$l_{i_{\text{attack}}}$ is attacked, maximizing the detection utility $(x_{i_{\text{attack}}} R_{i_{\text{attack}}}^d + (1 - x_{i_{\text{attack}}}) P_{i_{\text{attack}}}^d)$.

In P2, the last two constraints jointly ensure ①. Specifically, the last constraint in P2 means that, for $\forall i$, $\big(x_i P_i^a + (1 - x_i) R_i^a\big)$ is upper bounded by K. The penultimate constraint in P2 indicates that, when the adversary selects to attack a link $l_{i_{\text{attack}}}$, i.e., $B_{i_{\text{attack}}} = 1$, we have $(x_{i_{\text{attack}}} R_{i_{\text{attack}}}^a + (1 - x_{i_{\text{attack}}}) P_{i_{\text{attack}}}^a) \geq K$. This means that the adversary achieves the largest attack utility K by attacking $l_{i_{\text{attack}}}$ and thus would select to attack $l_{i_{\text{attack}}}$. Meanwhile, from the penultimate constraint in P2, we derive $(x_i P_i^a + (1 - x_i) R_i^a) \geq K - Z$ for $\forall i (i \neq i_{\text{attack}})$, where Z is a positive constant satisfying $K - Z < K$. This ensures that the adversary achieves the attack utility lower than K by attacking links other than $l_{i_{\text{attack}}}$.

On the other hand, the antepenultimate constraint in P2 ensures ②. More precisely, we have $(x_{i_{\text{attack}}} R_{i_{\text{attack}}}^d + (1 - x_{i_{\text{attack}}}) P_{i_{\text{attack}}}^d) \geq V$, when the adversary selects to attack $l_{i_{\text{attack}}}$. Maximizing V would result in a maximized detection utility $(x_{i_{\text{attack}}} R_{i_{\text{attack}}}^d + (1 - x_{i_{\text{attack}}}) P_{i_{\text{attack}}}^d)$.

6 Optimal Detection Strategies against A Quantal Response Adversary

We next consider an adversary with tunable rationality. In real-world situations, the adversary is unlikely to be perfectly

rational due to the subjective nature of human behavior, indicating a very low probability for him to choose the strategy with maximum utility. Therefore, we need to extend the best response behavior model of the adversary.

When selecting strategies, it is hard, if not impossible, for the adversary to be perfectly rational or completely irrational. In other words, decision noises will be added to characterize the rationality of the adversary. Despite decision noises, the adversary generally chooses better strategies more frequently. However, the capability of the adversary choosing better strategies decreases as the noises increase.

To describe such strategy selection behavior, we employ the quantal response model for the adversary [46]. Specifically, the probability $Q_i(\lambda)$ that a quantal response adversary chooses target l_i can be calculated by:

$$Q_i(\lambda) = \frac{e^{\lambda U_i^a(x_i)}}{\sum_{i=1}^{I} e^{\lambda U_i^a(x_i)}}, \tag{9}$$

where $\lambda \in [0, \infty)$ is a non-negative parameter characterizing the rationality of the adversary. As λ increases, the adversary becomes rational. Particularly, the adversary becomes completely irrational when $\lambda = 0$, and perfectly rational as $\lambda \to \infty$. From (9), we see that a completely irrational adversary chooses any link l_i uniformly at random; whereas a perfectly rational adversary chooses the strategy with the largest utility, indicating (9) is equivalent to (4) as $\lambda \to \infty$.

Given that the adversary follows the quantal response model, the defender aims to find the optimal detection strategy that maximizes the expected utility as follows:

$$\max_{\mathbf{a}} \sum_{i=1}^{I} Q_i(\lambda) U_i^d(x_i). \tag{10}$$

Substituting (9), we rewrite (10) as:

$$\max_{\mathbf{a}} \sum_{i=1}^{I} \frac{e^{\lambda U_i^a(x_i)}}{\sum_{i=1}^{I} e^{\lambda U_i^a(x_i)}} U_i^d(x_i). \tag{11}$$

To calculate the defender's optimal strategy against a quantal response adversary, problem P3 needs to be solved.

$$\mathbf{P3}: \begin{cases} \max_{\mathbf{a}} \sum_{i=1}^{I} \frac{e^{\lambda U_i^a(x_i)}}{\sum_{i=1}^{I} e^{\lambda U_i^a(x_i)}} U_i^d(x_i), \\ \text{subject to} \quad x_i = \sum_{j=1}^{J} a_j A_{ij} \text{ for } \forall i, \\ \qquad\quad \sum_{j=1}^{J} a_j = 1, \\ \qquad\quad 0 \le a_j \le 1 \text{ for } \forall j. \end{cases} \tag{12}$$

Solving P3 requires solving nonlinear, nonconvex NP-hard optimization problems. Substituting (1) and (2), we rewrite the objective function in P3 as follows:

$$\max_{\mathbf{a}} \sum_{i=1}^{I} \frac{e^{\lambda R_i^a} e^{-\lambda(R_i^a - P_i^a)x_i} \left((R_i^d - P_i^d)x_i + P_i^d\right)}{\sum_{i=1}^{I} e^{\lambda R_i^a} e^{-\lambda(R_i^a - P_i^a)x_i}}. \tag{13}$$

The above equation is a function w.r.t. x_i. Let $e^{\lambda R_i^a} = \mathcal{A}_i$, $\lambda(R_i^a - P_i^a) = \mathcal{B}_i$, and $R_i^d - P_i^d = \mathcal{C}_i$. The above equation can be further rewritten as:

$$\max_{\mathbf{a}} \frac{\sum_{i=1}^{I} \mathcal{A}_i \mathcal{C}_i x_i e^{-\mathcal{B}_i x_i} + \sum_{i=1}^{I} \mathcal{A}_i P_i^d e^{-\mathcal{B}_i x_i}}{\sum_{i=1}^{I} \mathcal{A}_i e^{-\mathcal{B}_i x_i}}. \tag{14}$$

We use $\mathscr{N}(\mathbf{x})$ and $\mathscr{D}(\mathbf{x})$ to denote the numerator and denominator of (14). Recall that $x_i = \sum_{j=1}^{J} a_j A_{ij}$, $\forall i$. Then, (14) can be equivalently simplified as:

$$\max_{\mathbf{a}} \frac{\mathscr{N}(\mathbf{x})}{\mathscr{D}(\mathbf{x})}. \tag{15}$$

To optimize (15), we employ a *binary search* strategy. Specifically, we first estimate the initial upper bound (UB) (e.g., maximum R_i^d) and lower bound (LB) of the defender's expected utility (e.g., minimum R_i^d) in (15), then calculate the mean value, denoted as m, of UB and LB. If $\exists \mathbf{x}$ such that $\mathscr{N}(\mathbf{x})/\mathscr{D}(\mathbf{x}) \ge m$, we update LB as m. Otherwise, we update UB as m. We iteratively conduct the updating until UB and LB are sufficiently close (i.e., smaller than an arbitrarily small value ϵ).

A key step in the above iterations is to determine if $\exists \mathbf{x}$ such that $\mathscr{N}(\mathbf{x})/\mathscr{D}(\mathbf{x}) \ge m$. To achieve this, we can minimize the objective function $m\mathscr{D}(\mathbf{x}) - \mathscr{N}(\mathbf{x})$ to see if its optimal value is smaller than or equal to zero. If so, we can conclude that $\exists \mathbf{x}$ such that $\mathscr{N}(\mathbf{x})/\mathscr{D}(\mathbf{x}) \ge m$. Otherwise, we have $\mathscr{N}(\mathbf{x})/\mathscr{D}(\mathbf{x}) < m$ for $\forall \mathbf{x}$. The objective function to minimize can be further expanded as:

$$\min_{\mathbf{a}} \sum_{i=1}^{I} \mathcal{A}_i (m - P_i^d) e^{-\mathcal{B}_i x_i} - \sum_{i=1}^{I} \mathcal{A}_i \mathcal{C}_i x_i e^{-\mathcal{B}_i x_i}. \tag{16}$$

We then convert P3 to P4.

$$\mathbf{P4}: \begin{cases} \min_{\mathbf{a}} \sum_{i=1}^{I} \mathcal{A}_i (m - P_i^d) e^{-\mathcal{B}_i x_i} - \sum_{i=1}^{I} \mathcal{A}_i \mathcal{C}_i x_i e^{-\mathcal{B}_i x_i}, \\ \text{subject to} \quad x_i = \sum_{j=1}^{J} a_j A_{ij} \text{ for } \forall i, \\ \qquad\quad \sum_{j=1}^{J} a_j = 1, \\ \qquad\quad 0 \le a_j \le 1 \text{ for } \forall j. \end{cases} \tag{17}$$

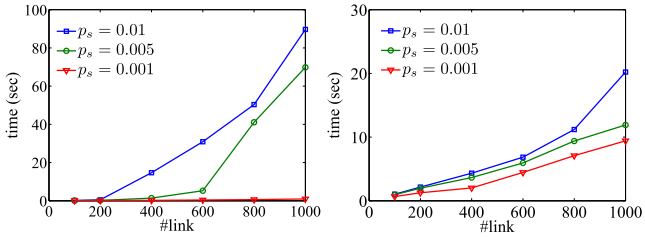
Let $\mathcal{N}'(\mathbf{x})$ and $\mathcal{D}'(\mathbf{x})$ be the piecewise linear approximation of $\mathcal{N}(\mathbf{x})$ and $\mathcal{D}(\mathbf{x})$, respectively. Using piecewise linearization $e^{-\mathcal{B}_i x_i} = 1 + \sum_{w=1}^{W} \gamma_{iw} x_{iw}$ and $x_i e^{-\mathcal{B}_i x_i} = \sum_{w=1}^{W} \mu_{iw} x_{iw}$ [47], we have $\mathcal{N}'(\mathbf{x}) = \sum_{i=1}^{I} \mathcal{A}_i P_i^d (1 + \sum_{w=1}^{W} \gamma_{iw} x_{iw}) + \sum_{i=1}^{I} \mathcal{A}_i \mathcal{C}_i \sum_{w=1}^{W} \mu_{iw} x_{iw}$ and $\mathcal{D}'(\mathbf{x}) = \sum_{i=1}^{I} \mathcal{A}_i (1 + \sum_{w=1}^{W} \gamma_{iw} x_{iw})$. Therefore, (16) is approximated as

$$\min_{\mathbf{a}} \sum_{i=1}^{I} \mathcal{A}_i (m - P_i^d) \left(1 + \sum_{w=1}^{W} \gamma_{iw} x_{iw}\right) - \sum_{i=1}^{I} \mathcal{A}_i \mathcal{C}_i \sum_{w=1}^{W} \mu_{iw} x_{iw}, \tag{18}$$

where W is the number of pieces that uniformly partition $[0, 1]$, x_{ik} is the portion of x_i in each piece (i.e., $x_i = \sum_{k=1}^{W} x_{ik}$), γ_{iw} and μ_{iw} are the slope of each piece.

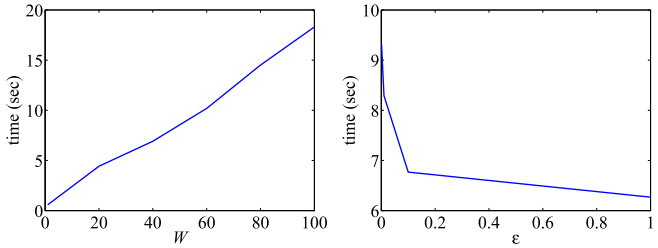
The minimization problem for determining if $\exists \mathbf{x}$ such that $\mathscr{N}(\mathbf{x})/\mathscr{D}(\mathbf{x}) \ge m$ is presented in problem P5, a mixed-integer linear programming formulation.

$$\mathbf{P5}: \begin{cases} \min_{\mathbf{a}} \sum_{i=1}^{I} \mathcal{A}_i (m - P_i^d) \left(1 + \sum_{w=1}^{W} \gamma_{iw} x_{iw}\right) - \sum_{i=1}^{I} \mathcal{A}_i \mathcal{C}_i \sum_{w=1}^{W} \mu_{iw} x_{iw}, \\ \text{subject to} \quad x_i = \sum_{j=1}^{J} a_j A_{ij} \text{ for } \forall i, \\ \qquad\quad \sum_{j=1}^{J} a_j = 1, \\ \qquad\quad 0 \le a_j \le 1 \text{ for } \forall j, \\ 0 \le x_{iw} \le \frac{1}{W} \text{ for } \forall i, w = 1, \ldots, W, \\ z_{iw} \frac{1}{W} \le x_{iw} \text{ for } \forall i, w = 1, \ldots, W, \\ z_{i(w+1)} \le z_{iw}, z_{iw} \in \{0, 1\} \text{ for } \forall i, w = 1, \ldots, W. \end{cases} \tag{19}$$



(a) Best response (b) Quantal response

Fig. 2. Time consumption versus the number of links in random network topologies.



(a) W (b) ϵ

Fig. 3. Time consumption versus W and ϵ.

The Optimal Solution of P5. For assisting the analysis, we define two constants $C_1 = e^{\bar{\mathcal{B}}}(\bar{\mathcal{C}} + \bar{\mathcal{B}}(\bar{R}^d + \bar{P}^d))\bar{\mathcal{A}}/\underline{\mathcal{A}}$ and $C_2 = 1 + e^{\bar{\mathcal{B}}}\bar{\mathcal{A}}/\underline{\mathcal{A}}$, where $\bar{\mathcal{B}} = \max_{i=1}^{I}\mathcal{B}_i$, $\bar{\mathcal{C}} = \max_{i=1}^{I}\mathcal{C}_i$, $\bar{\mathcal{A}} = \max_{i=1}^{I}\mathcal{A}_i$, $\underline{\mathcal{A}} = \min_{i=1}^{I}\mathcal{A}_i$, $\bar{R}^d = \max_{i=1}^{I}|R_i^d|$, and $\bar{P}^d = \max_{i=1}^{I}|P_i^d|$.

First, the error of the piecewise linearization is bounded and decreases as W increases, because we have $|\mathcal{D}'(\mathbf{x}) - \mathcal{D}(\mathbf{x})| \leq \bar{\mathcal{A}}I\bar{\mathcal{B}}/W$ and $|\mathcal{N}'(\mathbf{x}) - \mathcal{N}(\mathbf{x})| \leq \bar{\mathcal{A}}I(\bar{\mathcal{C}} + \bar{\mathcal{B}}\bar{P}^d)/W$. This ensures that the piecewise linearization can well approximate the original problem. $|\mathcal{D}'(\mathbf{x}) - \mathcal{D}(\mathbf{x})| \leq \bar{\mathcal{A}}I\bar{\mathcal{B}}/W$ can be proved as follows. We have $|\mathcal{D}'(\mathbf{x}) - \mathcal{D}(\mathbf{x})| = \sum_{i=1}^{I}\mathcal{A}_i(1 + \sum_{w=1}^{W}\gamma_{iw}x_{iw}) - \sum_{i=1}^{I}\mathcal{A}_i e^{-\mathcal{B}_i x_i} = \sum_{i=1}^{I}\mathcal{A}_i(1 + \sum_{w=1}^{W}\gamma_{iw}x_{iw} - e^{-\mathcal{B}_i x_i})$. Since the maximal error of the piecewise linear approximation cannot exceed the function value variation within any piece, we derive $1 + \sum_{w=1}^{W}\gamma_{iw}x_{iw} - e^{-\mathcal{B}_i x_i} \leq \mathcal{B}_i e^{-\mathcal{B}_i x_i}/W \leq \mathcal{B}_i/W$. Note $\mathcal{B}_i e^{-\mathcal{B}_i x_i}$ is deduced by finding the derivative of $e^{-\mathcal{B}_i x_i}$ w.r.t. x_i and then taking the positive value. Thus, we obtain $|\mathcal{D}'(\mathbf{x}) - \mathcal{D}(\mathbf{x})| \leq \sum_{i=1}^{I}\mathcal{A}_i\mathcal{B}_i/W \leq \bar{\mathcal{A}}I\bar{\mathcal{B}}/W$. One can prove $|\mathcal{N}'(\mathbf{x}) - \mathcal{N}(\mathbf{x})| \leq \bar{\mathcal{A}}I(\bar{\mathcal{C}} + \bar{\mathcal{B}}\bar{P}^d)/W$ similarly.

Second, the defender's expected utility computed by P5 can be arbitrarily close to the global optimal solution with sufficiently large W and sufficiently small ϵ, because we have $0 \leq U_*^d - U_{\mathbf{a}}^d \leq 2C_1/W + (C_2 + 1)\epsilon$. Here, U_*^d is the global optimum of the defender's expected utility, $\mathbf{a}$ denotes the defender's strategy computed by P5, and $U_{\mathbf{a}}^d$ represents the defender's expected utility calculated by (15) when the defender adopts $\mathbf{a}$. It is easy to know $0 \leq U_*^d - U_{\mathbf{a}}^d$ because U_*^d is the global optimal solution maximizing the defender's expected utility. Please refer to Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TDSC.2019.2892370, for the proof of $U_*^d - U_{\mathbf{a}}^d \leq 2C_1/W + (C_2 + 1)\epsilon$.

7 Experimental Evaluation

First, we evaluate the time consumption of our algorithms using synthetic data. Then, we demonstrate the effectiveness of our security patrolling for LFAs using a real-world network topology and a testbed.

7.1 Benchmark Detection Strategies

Before reporting experiment results, we define two benchmark strategies, namely *uniform-detection* and *best-detection*. The uniform-detection strategy allows the defender to select one pure strategy uniformly at random, while the best-detection strategy allows the defender to select one constant pure strategy that contains the links contributing the largest reward or penalty (if attacked). These benchmark strategies are formally defined below.

> Uniform-Detection Strategy: play Γ_j with probability $a_j = \frac{1}{J}$ where J is the total number of pure strategies $j = 1, 2, \ldots, J$

> Best-Detection Strategy: play Γ_{j^*} with probability 1 where Γ_{j^*} is the pure strategy that contains the link with the largest reward if attacked

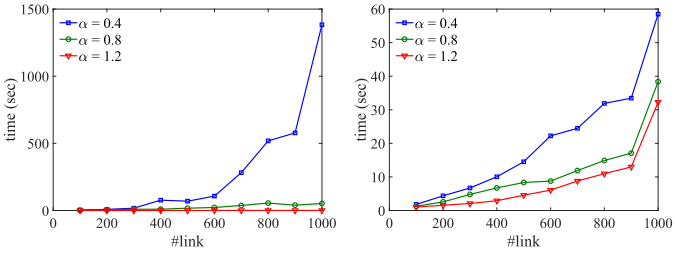
7.2 Time Consumption Evaluation

To evaluate the time complexity of solving our optimization problems, we show the time consumption of our algorithms using the off-the-shelf optimizer, i.e., the actual time spent by our algorithms to figure out the optimal solution under different parametric settings. Specifically, we use the CPLEX optimizer, a high-performance mathematical programming solver for linear programming, mixed integer programming, and quadratic programming [48]. To this end, we execute our algorithms by varying the parameters with synthetic data, on a normal PC with 16 GB RAM and a quad-core Intel E5-1650 3.20 GHz CPU.

We first randomly generate networks with the number of links varying from 100 to 1,000. We consider each link as an adversary's pure strategy, and thus the number of the adversary's pure strategies equals the number of links. Meanwhile, we set the defender's pure strategies two times of the number of links. To generate each defender's pure strategy Γ_j, we set the sampling probability as 0.001, 0.005, 0.01, denoted as p_s, to randomly select links so to generate A_{ij}, i.e., the belonging relationship between a pure strategy Γ_j and a link l_i.

Fig. 2 depicts the time consumption over different number of links (i.e., #link). The results show that, for the best/quantal response adversary (we set $\lambda = 1$), the time consumption increases as the number of links increases. Also, for a certain number of links, the time consumption increases as p_s increases. More importantly, the overall growth rate of the time consumption increases as p_s increases. This indicates that designing small-sized pure strategies for the defender benefits the scalability w.r.t. the number of links (i.e., the number of the adversary's pure strategies).

Fig. 3 shows the time consumption with varying parameters W and ϵ, where W denotes the number of pieces of linear



(a) Best response (b) Quantal response

Fig. 4. Time consumption versus the number of links in network topologies with link importance following the Zipf-Mandelbrot distribution.

functions used to approximate the nonlinear objective function in (16), and ϵ is a small value for determining the convergence of the binary search method optimizing (15). Fig. 3a shows that the time consumption increases roughly linearly as W increases, meaning our algorithms are scalable w.r.t. W. Fig. 3b reveals that the time consumption decreases with increased ϵ (i.e., decreased approximation accuracy). Particularly, when ϵ reaches around 0.1, the time consumption drops drastically and then decreases linearly.

To further evaluate the time consumption over network topologies conforming to real-world characteristics, we generate networks with link importance following the Zipf-Mandelbrot distribution, which has demonstrated its capability of accurately modeling real-world characteristics of IP links [12], [49]. More precisely, we normalize link importance (i.e., the number of paths crossing the link) by the total number of paths, and the normalized link importance follows the Zipf-Mandelbrot distribution, namely, $f(k) = 1/(k+\beta)^{\alpha}$. Here, k denotes the rank of a link w.r.t. link importance, α is the exponent of the power law distribution, β represents the fitting parameter, and $f(k)$ stands for the normalized link importance. We set the values of α as 0.4, 0.8, 1.2, and the value of β as 0.5. Note that the higher α, the sharper the concentration of paths in a few links. We then generate the networks with link importance following the Zipf-Mandelbrot distribution under these settings to evaluate the time consumption, with the number of links varying from 100 to 1,000. Fig. 4 depicts the time consumption over different number of links (i.e., #link). We again observe that the runtime increases as the number of links grows. More importantly, for a certain number of links, the runtime decreases as α increases (i.e., as the concentration of paths in a few links becomes sharper).

7.3 Evaluation over A Real-World Topology

We report experimental results on a real-world topology surrounding a regional network in Taiwan. The network topology was collected by tracerouting the paths between different geographically dispersed machines and the machines inside the regional network. Our data is available at [50]. As shown in Fig. 5, the topology contains 70 paths and 336 links, where the defender's *probing agent* is deployed at the top node (i.e., $s_d \in S^D$). Note that a node's succeeding (downstream) nodes, if no further branching or merging exists, are all removed for simplifying the visualization. The annotations in color are presented to intuitively demonstrate detection strategies against a best response adversary (see Section 7.4).

Following the real-world LFA detection scenario, we set up the probing agent to monitor one or two paths (i.e., $M(s^d) = 2$) in each round of the game, accounting for less than 3 percent of all paths and hence consistent with the scenario that the defender has limited security resources. Thus, the defender can perform 1-path (one path only) and 2-path (two paths concurrently) probing as needed.

We consider that the defender and the adversary play a zero-sum game. That is, each player's gains (or losses) of the utility is exactly balanced by the losses (or gains) of the utility of the other player(s). In a two-player zero-sum game, the equilibrium of the Stackelberg security game is equivalent to the Nash equilibrium [29]. Formally, we set $R_i^d + P_i^a = 0$, $R_i^a + P_i^d = 0$, and consequently we have $U^d + U^a = 0$. For the ease of demonstration, we employ one of the metrics quantifying the asset importance of a link, i.e., *the number of paths crossing link* l_i, denoted by N_i, to measure the reward and penalty of attacking or protecting l_i. Specifically, in cases where l_i is attacked without successful detection, the defender receives penalty $P_i^d = -N_i$ and the adversary receives reward $R_i^a = N_i$; and in cases where l_i is attacked

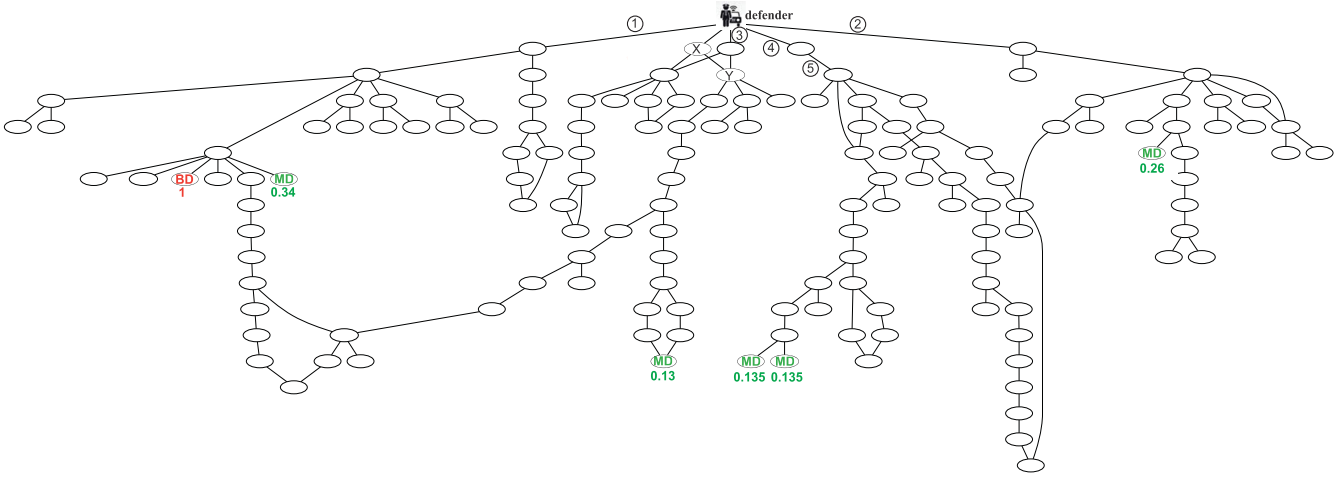


Fig. 5. The defender's detection strategies against a best response adversary, including best-detection and mixed-detection strategies, under 1-path constraint. The defender following the best-detection strategy probes the path heading the node BD with probability 1, while the defender adopting the mixed-detection strategy probes the paths heading the nodes MD with the probabilities equal to the green numbers aside that sum up to 1. In response to the best-detection (or uniform-detection) defender, the best response adversary selects to attack the link ④ or ⑤. In response to the mixed-detection defender, the best response adversary selects to attack one link among ①, ②, ③, ④, or ⑤.

TABLE 2
The Defender's Expected Utility against a Best Response Adversary When Adopting Different Detection Strategies

utility / strategy / constraint	mixed-detection	uniform-detection	best-detection
1-path	−7.2828	−8.6857	−16.0000
2-path	−3.0101	−4.6584	−15.0000

with successful detection, the defender receives reward $R_i^d = N_i$ and the adversary receives penalty $P_i^a = -N_i$.

Utility Definition. After defining the reward and the penalty, the defender's expected detection utility is then calculated as $U^d = \sum_{i=1}^{I} B_i U_i^d$ (best response adversary) or $U^d = \sum_{i=1}^{I} Q_i(\lambda) U_i^d$ (quantal response adversary). Combining (1) and (2), we rewrite the defender utility as:

$$U^d = \sum_{i=1}^{I} \{B_i, Q_i(\lambda)\} x_i N_i - \sum_{i=1}^{I} \{B_i, Q_i(\lambda)\}(1 - x_i) N_i, \quad (20)$$

where $\{B_i, Q_i(\lambda)\}$ equals B_i for the best response adversary and $Q_i(\lambda)$ for the quantal response adversary.

Physical Intuition. The defender's expected detection utility has a meaningful physical intuition. That is, when the adversary attacks a single link, the difference between the expected asset importance of the links *successfully* protected by the defender (i.e., encountering the attack whereas the attack is detected) and the expected asset importance of the links *unsuccessfully* protected by the defender (i.e., suffering from the attack and the attack is not detected). The former is calculated as $\sum_{i=1}^{I} \{B_i, Q_i(\lambda)\} x_i N_i$, where the attack against the link is detected by the defender. The latter can be derived by $\sum_{i=1}^{I} \{B_i, Q_i(\lambda)\}(1 - x_i) N_i$, where the attack against the link is *not* detected by the defender. Apparently, the larger the difference between the former and the latter is, the more utility the defender gains.

Since in our context the asset importance of link l_i is measured by N_i, i.e., the number of paths crossing l_i. It is easy to know $U^d \in [-p_{\max}, p_{\max}]$, where $p_{\max}$ is the maximum number of paths crossing a link. Meanwhile, the adversary's expected utility U^a is the opposite, i.e., $U^a = -U^d \in [-p_{\max}, p_{\max}]$. In our topology, $p_{\max}$ equals 22. To enhance the utility representation, we normalize U^d and U^a by $p_{\max}$ to deduce *relative utility* for the adversary and the defender, which lies in the interval of $[-1, 1]$.

Note that the values of the detection utility are negative when the (defender's security resource) investment is limited. Specifically, in the face of a best response adversary, we have $U^d = \sum_{i=1}^{I} B_i [x_i N_i - (1 - x_i) N_i] = \sum_{i=1}^{I} B_i N_i (2x_i - 1)$. Recall the definition of B_i in (4), we derive $U^d = N_{i\_\text{attack}}(2x_{i\_\text{attack}} - 1)$, where i_attack is the index of the link that the best response adversary chooses to attack. Therefore, if $x_{i\_\text{attack}} \geq 1/2$ holds (the marginal probability of protecting link i_attack is not less than 1/2), U^d would be non-negative; otherwise negative. In the face of a quantal response adversary, we have $U^d = \sum_{i=1}^{I} Q_i(\lambda)[(2x_i - 1) N_i]$. In two extreme cases, such as $x_i < 1/2$ (limited investment) and $x_i > 1/2$ (sufficient investment) for $i = 1, 2, \ldots, I$, U^d would be negative and positive, respectively. The boundary of x_i that leads to $U^d = 0$ depends on the distribution of $Q_i(\lambda)$ and N_i. This suggests that, to gain a positive utility, the defender should invest more security resources (e.g., deploying more probing servers to increase the marginal probability of protecting each link).

Other Metrics. The defender utility above is an *overall* metric integrating both successful and unsuccessful detection of the defender. We can also use two separate metrics to jointly describe the defender's performance, namely, patrolling efficiency index (PEI) and attack mitigation ratio (AMR). They are formally defined as follows:

$$\begin{aligned} PEI &= \sum_{i=1}^{I} \{B_i, Q_i(\lambda)\} x_i N_i / \sum_{i=1}^{I} x_i N_i, \\ AMR &= \sum_{i=1}^{I} \{B_i, Q_i(\lambda)\} x_i N_i / \sum_{i=1}^{I} \{B_i, Q_i(\lambda)\} N_i. \end{aligned} \quad (21)$$

Physical Intuition. PEI represents the ratio of the expected asset importance of the links successfully protected by the defender (i.e., $\sum_{i=1}^{I} \{B_i, Q_i(\lambda)\} x_i N_i$) to the expected asset importance of the links that the defender patrols (i.e., $\sum_{i=1}^{I} x_i N_i$). AMR measures the ratio of the expected asset importance of the links successfully protected by the defender to the expected asset importance of all the links suffering from the attack (i.e., $\sum_{i=1}^{I} \{B_i, Q_i(\lambda)\} N_i$). These two metrics will be demonstrated in Section 7.5.

7.3.1 A Best Response Adversary

We first consider a best response adversary who attacks the link that maximizes his expected utility. The defender can adopt mixed-detection, uniform-detection and best-detection strategies. Table 2 reports the detection performance when the defender adopts different strategies.

We observe that, under different constraints, the defender adopting the mixed-detection strategy achieves the largest expected utility. Compared to the best-detection strategy, the uniform-detection strategy performs better.

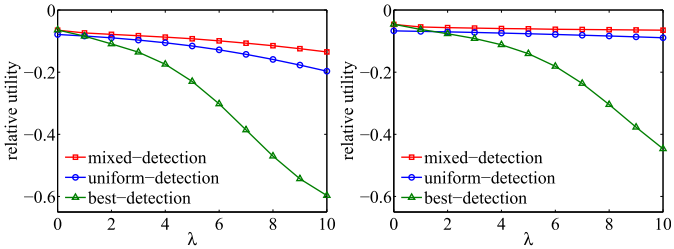
Insight. To handle a best response adversary, the defender constantly protecting the most important link (i.e., with the largest reward if attacked) loses the most, whereas our mixed-detection strategy can assist the defender to gain the highest utility.

We also observe that, for a certain detection strategy, the defender gains more utility when she can probe more path simultaneously. This is not difficult to understand. However, it is interesting to see that, even when the defender adopting the best-detection strategy can probe two paths simultaneously, her utility (i.e., -15) is still much lower than that (i.e., -7.2828) of the defender that adopts the mixed-detection strategy and can only probe one path each time.

Insight. Our carefully designed mixed-detection strategy can achieve much higher detection utility with only half resource investment compared to the best-detection strategy.

7.3.2 A Quantal Response Adversary

Next, we consider a quantal response adversary who attacks links with bounded rationality. Recall λ denotes the rationality of the adversary, where larger values of λ mean more rationality. Fig. 6 presents the defender's expected *relative* utility against a quantal response adversary with varying rationality when adopting mixed-detection, uniform-detection,



(a) 1-path (b) 2-path

Fig. 6. The defender's expected relative detection utility against a quantal response adversary with varying rationality (λ).

and best-detection strategies, under 1-path and 2-path constraints. We have three observations below.

First, the mixed-detection strategy achieves the largest detection utility, the best-detection strategy performs the worst, while the uniform-detection strategy has moderate performance, except at the point $\lambda = 0$ where the adversary is completely irrational. In the exceptional point, the uniform-detection strategy performs the worst, while the mixed-detection and best-detection strategies achieve equal performance better than that of the uniform-detection strategy. In particular, the mixed-detection strategy *degenerates* to the best-detection strategy when $\lambda = 0$. This interesting finding reveals the following insight.

Insight. For a completely irrational adversary (attacking target links uniformly at random), the defender's optimal strategy is to select one constant pure strategy with the largest asset importance.

Second, whichever strategy the defender adopts, the defender's expected utility decreases as the adversary's rationality grows. Meanwhile, as the adversary becomes rational, there is a significant tendency that the utility of the best-detection strategy decreases drastically, meaning that constantly protecting the most important link could be easily defeated by a rational adversary. On the contrary, when the defender adopts randomized strategies (i.e., mixed-detection and uniform-detection), such a tendency becomes much less significant, indicating the robustness of randomized strategies against the adversary with increased rationality. Again, the mixed-detection strategy is the best.

Insight. As the adversary becomes rational, the defender should adopt randomized strategies so to achieve better utility, wherein the mixed-detection strategy outperforms the uniform-detection strategy. Otherwise, constantly protecting the most important link could be easily defeated.

Last, compared to the 1-path constraint, the defender's expected utility is larger under the 2-path constraint, for any given detection strategy at a particular value of λ.

Insight. All other things being equal, more security resource investment results in better detection utility.

7.4 Case Study

To get intuitive understanding, we detail detection strategies against a best response adversary and a quantal response adversary, under 1-path constraint.

In Fig. 5, we label detection strategies against a best response adversary. Specifically, we label mixed-detection and best-detection strategies based on the nodes that the defender's probing packets head. Specifically, nodes labeled with MD and the corresponding numbers aside represent the nodes that the defender's probing packets head and the probabilities in the mixed-detection strategy, respectively. The nodes in the best-detection strategy are labeled with BD. Meanwhile, we label the links that the best response adversary attacks by ①, ②, ③, ④, and ⑤.

In Fig. 7, we label detection strategies against a quantal response adversary with $\lambda = 3$, along with the links that the quantal response adversary attacks in response to the mixed-detection strategy. We use the same way to label detection strategies. Furthermore, we label the links that the quantal response adversary attacks using blue solid lines, where the thickness represents the probability of attacking the corresponding links, and all probabilities sum up to one.

Uniform-Detection. The uniform-detection defender randomly probes all the paths with equal probability. To maximize the attack utility, the best response adversary attacks the link ④ or ⑤ with (equal) maximum attack utility.

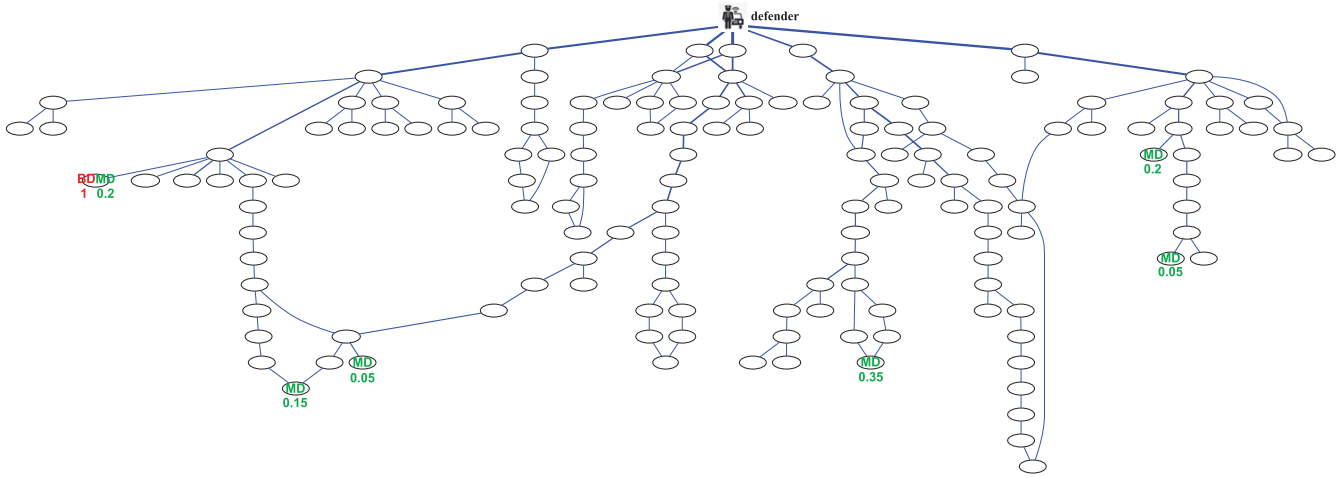


Fig. 7. The defender's detection strategies against a quantal response adversary with $\lambda = 3$, including best-detection and mixed-detection strategies, under 1-path constraint. The defender following the best-detection strategy probes the path heading towards the node BD with probability 1, while the defender adopting the mixed-detection strategy probes the paths heading towards the nodes MD with the probabilities equal to the green numbers aside that sum up to 1. In response to the mixed-detection defender, the quantal response adversary selects to attack one link among the blue links, where the thickness of the link is proportional to the attack probability.

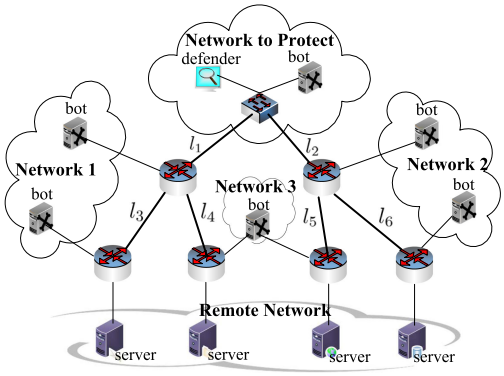


Fig. 8. The testbed.

Best-Detection. The best-detection defender selects one path that crosses the most important link (i.e., the link with the largest reward if attacked) to protect. Consequently, she deploys the probing along the path destined to the nodes labeled with BD. For the best response adversary, the attacked link is ④ or ⑤ for maximizing the attack utility.

Mixed-Detection. The mixed-detection defender probes the paths destined to the nodes labeled with MD, denoted as PATHS_MD, following the probabilities equal to the numbers aside that add up to one, so to maximize her detection utility against the best/quantal response adversary. The best response adversary attacks one of the links among ①, ②, ③, ④, and ⑤ that result in (equal maximum) attack utility.

In response to all the detection strategies above, the quantal response adversary might attack any link with a probability distribution maximizing his utility. For instance, as depicted in Fig. 7, in response to the mixed-detection defender, the quantal response adversary selects to attack one link among the blue links, where the thickness of the link is proportional to the attack probability.

Note that, when adopting the mixed-detection strategy, the defender only probes the paths heading towards the nodes labeled with MD, only a portion of all the path, in defending against both the best response adversary (Fig. 5) and the quantal response adversary (Fig. 7). This is also termed as *small support size observation* [51]. The reason is that probing those paths heading towards the nodes without MD, denoted as PATHS_NMD, would reduce the detection utility. Specifically, probing PATHS_NMD leads to the decline of the overall probability of probing PATHS_MD. Consequently, the probability of probing *at least* one path in PATHS_MD, say p, decreases. If so, the best/quantal response adversary would re-design his strategies, i.e., attacking the most important link along p, or raising the probability to do so, further increasing the attack utility while reducing the detection utility. For example, in Fig. 5, if the probability that the defender probes the leftmost node labeled with MD decreases from 0.34 to 0.30, the attack utility of attacking link ① would increase by 1.5172 from 7.2828 to 8.8, and accordingly the detection utility decreases by 1.5172.

Albeit the presence of paths (and links) that are not protected by the mixed-detection defender, the best/quantal response adversary does not benefit from attacking these unprotected paths (and links). Take the link between nodes X and Y in Fig. 5 as an example. Although not protected, the link induces an attack utility only equal to 5, lower than the attack utility 7.2828 by attacking link ④ that the defender protects with probability 0.27.

TABLE 3
The Defender's Expected Detection Utility under Different Detection and Response Strategies over the Testbed

utility: defender / adversary	mixed-detection	uniform-detection	best-detection
best response	−16.2162	−30.0000	−50.0000
quantal response ($\lambda = 3$)	−13.3365	−16.0060	−36.7702
quantal response ($\lambda = 0$)	3.3333	−12.5000	3.3333

7.5 Evaluation over a Testbed

Besides simulation experiments over real-world topologies, we perform experiments to demonstrate a scenario where link importance can be measured by capacity bandwidth. To make capacity bandwidth under control, we deploy a testbed as shown in Fig. 8. More importantly, the controlled testbed allows us to launch real attacks against links and meanwhile perform security patrolling, while avoiding ethical issues of launching real attacks against Internet links serving a large region.

Fig. 8 illustrates the testbed. To emulate LFAs, we deploy a bot near each router for flexibly coordinating attack traffic towards any link. Each bot is equipped with D-ITG (Distributed Internet Traffic Generator) [52] to generate UDP-based attack traffic flows, allowing us to produce traffic at packet level with pre-specified distributions of both packet size and packet inter-departure time. The defender resides within the network under protection, where she could probe the paths surrounding the network by sending measurement packets to servers in the remote network.

We set the capacity bandwidth of l_1, l_2, l_3, l_4, and l_5 as 100, 50, 60, 40, 30 and 20 Mbps, respectively. The asset importance of a link is measured by the capacity bandwidth, meaning that N_i *in all formulas will represent the capacity bandwidth of link* l_i. We run the games by attacking the links. The games are repeated 100 rounds, each of which lasts for 10 seconds. During each round, the adversary attacks one link by congesting it (thus resulting in near-zero available bandwidth), while the defender probes one path by measuring the available bandwidth of the links along the path. Once near-zero available bandwidth of a link is observed, the defender successfully detects the attack.

Table 3 shows the defender's expected detection utility under different strategies, where the reward and penalty of attacking and protecting each link are measured by its capacity bandwidth. The observations are consistent with those derived in Section 7.3. For example, when the adversary attacks each link uniformly at random (i.e., quantal response with $\lambda = 0$), the mixed-detection strategy *degenerates* to the best-detection strategy, thus having equal utility. In all remaining cases, the mixed-detection performs the best.

Table 4 illustrates the defender's detection performance measured by metrics other than utility, including PEI and AMR, under different detection and response strategies over the testbed. The results demonstrate that the mixed-detection strategy is the best choice against the adversary with different levels of rationality. Note that, in the face of a

TABLE 4
The Defender's Detection Performance Measured by PEI (Patrolling Efficiency Index) and AMR (Attack Mitigation Ratio), under Different Detection and Response Strategies over the Testbed

(PEI, AMR) adversary \ defender	mixed-detection	uniform-detection	best-detection
best response	(13.40%, 28.15%)	(10.23%, 21.00%)	(0, 0)
quantal response ($\lambda = 3$)	(16.15%, 34.66%)	(12.36%, 29.43%)	(0.76%, 3.09%)
quantal response ($\lambda = 0$)	(16.67%, 53.33%)	(16.67%, 37.50%)	(16.67%, 53.33%)

quantal response adversary with $\lambda = 0$, two edge cases exist. First, the values of (PEI, AMR) are the same for both the mixed-detection defender and the best-detection defender, since in this case the mixed-detection strategy *degenerates* to the best-detection strategy. Second, all the values of PEI across different detection strategies are equal to 16.67 percent, because in (21) we have $\{B_i, Q_i(\lambda)\} = Q_i(0)$ that is a constant 0.1667. However, in the second edge case, the mixed-detection (and the best-detection) defender achieves the largest value of AMR. Therefore, to handle a quantal response adversary with $\lambda = 0$, the mixed-detection defender and the best-detection defender achieve equal performance, which is better than that of the uniform-detection defender.

Recall that PEI represents the ratio of the expected asset importance of the links successfully protected by the defender to the expected asset importance of the links that the defender patrols, and AMR measures the ratio of the expected asset importance of the links successfully protected by the defender to the expected asset importance of all the links suffering from the attack. Since in our scenario the asset importance of a link is measured by its capacity bandwidth, as compared to the defender adopting other strategies, the mixed-detection defender could detect the links under attack with more capacity bandwidth (thus accomplishing larger PEI) when patrolling links with a certain amount of capacity bandwidth; and that among all the links under attack, the mixed-detection defender can also identify the links under attack with more capacity bandwidth (thus having larger AMR).

8 Discussion

We have demonstrated the effectiveness of our carefully designed LFA detection strategies. There remain a few practical issues to discuss when security patrolling is carried out.

8.1 Path Instability

In our experiments, we assume stable paths for the network routing topology. This assumption, though not always true, enables us to concentrate on designing models for the network routing topology that is stable. The designed models are also applicable in real networks for four reasons. First, to successfully coordinate attack traffic flows towards target links, the adversary only requires the paths to be stable in a relatively short period, which is true in practice since IPv4 (IPv6 resp.) paths just have 0.13 (0.27 resp.) changes per day on average [53]. Second, the adversary tends to attack critical links, and these critical links are normally stable because they are traversed by a large number of paths (thus naturally immune to path instability). Third, the adversary would like to perform LFAs on the premise of keeping control plane of the attacked link unaffected so that dynamic re-routing cannot be initiated [10], hence not introducing additional path instability. Last, our model is also compatible with path instability, in the sense that path instability primarily influences the way to calculate link importance. Since link importance is typically measured by the number of paths crossing a link, one could consider the situation where a pair of source and destination nodes constitute multiple paths when calculating link importance in the context of path instability.

8.2 The Adversary's Rationality

The rationality of a quantal response adversary (i.e., $\lambda \in [0, \infty)$) is a parameter that needs to be determined empirically. As λ increases, the adversary becomes rational. In particular, the adversary becomes completely irrational when $\lambda = 0$, while perfectly rational as $\lambda \to \infty$. However, to issue a large-scale coordinated attack like LFAs that require tremendous resources, the adversary is unlikely to attack all links uniformly at random. Therefore, $\lambda = 0$ is not practical. Moreover, most adversaries may perform LFAs heuristically, without accurately maximizing their utility mathematically. Thus, $\lambda \to \infty$ may not hold in most cases. Consequently, λ should be positive and needs to be empirically determined based on historical attacks and typical LFAs behaviors.

8.3 The Adversary's Strategies

We consider that each adversary's pure strategy contains one link, meaning that the adversary attacks only one link each time. However, in LFA attacks, such as Crossfire [10], multiple links could be attacked concurrently to maximize the negative impact on the target network (or servers). Therefore, we need to enable our solution to handle the situation when multiple links are attacked concurrently. One intuitive approach is to enumerate all possible combinations of the set of target links, each of which constitutes a pure strategy. This approach is feasible when the number of target links is *not* large. However, as the number of links increases, it may make our solution mathematically intractable, because the number of pure strategies grows exponentially.

Fortunately, to disconnect the target network (or servers), the adversary usually just needs to flood a few core links, and thus the strategy space is narrowed down. These core links usually account for a relatively small proportion of all links [12]. To figure out the group of core links, one *cannot* simply list the top-ranked links based on their importance (e.g., the number of paths crossing a link). The reason is that different links may be overlapped with each other in terms of importance, which harms their overall importance as a group, i.e., *collective importance*. For example, for two links that are both important if separately measured, their collective importance may be just equivalent to the importance of one link, provided that the set of paths crossing one link is the same as those crossing the other. Given a set of links C, we define their collective importance as $F(C) = |\cup_{c \in C} Path(c)|$, where $Path(c)$ denotes the set of paths crossing a link $c \in C$, and $|\cdot|$ counts the number of elements within a set.

For the adversary, the set of core links are the set of links C maximizing $F(C)$ and carrying a large proportion of paths (i.e., $F(C)/F(L) \geq \theta$, where θ is a relatively large number between 0 and 1). However, the problem of maximizing $F(C)$ is NP-hard. One can solve it using the greedy algorithm (i.e., select a new link c' maximizing the reward gain, $\delta_{c'}(C) = F(C \cup c') - F(C)$, and insert c' into C in each round). This can derive a solution lower bounded by $1 - 1/e \approx 63\%$ of the global maximum, because of the non-increasing monotony and submodularity of $F(C)$ [54], [55]. The non-increasing monotony means that, for any two sets $C_1, C_2 \subseteq L$ and $C_1 \subseteq C_2$, we have $F(C_1) \leq F(C_2)$. The submodularity means that, for a non-decreasing function $F(C)$, given any new link $c' \in L \setminus C_2$, $F(C_1 \cup \{c'\}) - F(C_1) \geq F(C_2 \cup \{c'\}) - F(C_2)$ holds, i.e., smaller sets have more function value increment when a new link is added. It is easy to prove these two properties.

Suppose that the number of core links is N_{bt}. To elude security patrolling, it is possible for the adversary to flood all combinations of the core links. Thus, the total number of the adversary's pure strategies equals $2^{N_{bt}} - 1$. If N_{bt} is relatively small, e.g., $N_{bt} = 10$, there are 1,023 strategies, which can be easily tractable. However, for relatively large values of N_{bt}, the number of strategies would become extremely large. For example, using the greedy algorithm, our real-world large-scale topology experiment reveals that 200 links (56,039 links in total) carry around 80 percent of all 875,861 paths, when probing five networks in Switzerland, Hong Kong, Japan, Singapore, and Taiwan from 267 cities in 33 countries, resulting in the number of strategies (more than 1.6×10^{16}) intractable. The number of links becomes 400 when carrying roughly 90 percent of all paths, hence even having the number of strategies 1.6×10^{16} times larger.

The above analysis tells us that, to extend our work into the situation when multiple links are concurrently attacked, narrowing down the strategy space by finding out a group of core links is *necessary yet inadequate*, especially when the number of core links is relatively large. A *fundamental* approach is to divide the group of core links into different (tractable) subsets (e.g., dividing 200 links into 20 subsets), with each subset being assigned to one defender. Nevertheless, the links protected by different defenders are typically interdependent due to network externalities (e.g., the impact of attacking one link may be propagated to others, the links that two defenders patrol may have overlap). Thus, defenders may mutually affect each other's decision of resource assignment. This is further compounded by the fact that multiple links attacked by a single attacker might be under the protection of different defenders. Therefore, how to divide the group of core links into different subsets to minimize the interdependence and meanwhile simplify each defender's decision process to the utmost, and how to design a multi-defender model facilitating collaboration among defenders to optimally suppress attacks against different links remain new problems worthy of being systematically explored. We leave these problems as future work. Our current work is definitely the building block for future research in this direction.

9 Related Work

Recently, LFAs have gained attention in the literature. Kang et al. proposed LFAs that can effectively cut off the Internet connections of a target area *without* being detected [10], [12]. The Coremelt attack could be considered a special case of LFAs [28]. Since LFAs result in abnormal link performance, traditional active link (and path) measurement techniques, such as packet pair and packet train, could naturally facilitate the detection of LFAs. These conventional measurement techniques have been widely adopted [18], [56], hence practically deployable. However, they focus on the measurement accuracy, rather than the anomaly of link performance. Active measurement techniques in detecting network faults and connectivity problems [57], [58], [59], [60], [61], [62], [63] are also inapplicable to detect LFAs, because LFAs cause temporary (rather than persistent) congestion to avoid BGP changes, which would deter the attack by rerouting attack flows. To apply these techniques in detecting LFAs, LinkScope employs both end-to-end and hop-by-hop measurement to detect the links under such attacks [17]. However, how to schedule the probing in consideration of the adversary's behavior so to maximize the defender utility is not studied.

To defend against LFAs, Lee et al. designed a router-based approach named CoDef [14], and Gillani et al. proposed reallocating network resources dynamically through virtual networks to circumvent such attacks [64]. Kang et al. designed a software-defined network based system SPIFFY to tackle the root causes of LFAs (i.e, detection of bots participating in LFAs) by temporarily expanding link bandwidth [13]. Liaskos et al. performed continuous traffic re-routing at the level of traffic engineering, aiming at making the participation in LFAs improbable for benign sources and meanwhile forcing bots to adopt suspicious behaviors [65]. These studies require extensive deployment and fine-grained network reconfiguration unavailable to the Internet immediately. Thus, their effectiveness may be limited.

Game theory has been used to cope with traditional DDoS attacks. For example, Xu et al. proposed a game-theoretic model to protect web services from DDoS attacks [66]. The basic idea is to isolate legitimate traffic from a huge volume of attack traffic. Wu et al. focused on UDP-based DDoS attacks [67]. They addressed the challenge of determining optimal firewall settings to block attack traffic while protecting legitimate traffic. Yan et al. proposed a semi network-based game-theoretic framework that considers uncertain factors affecting the decision making of the players [68]. However, DDoS attacks in these studies usually target a single victim server, thereby inapplicable to LFAs by design. In addition, by deploying defense systems between the victim server and the network perimeter, as is required by [66], [67], [68], one cannot observe LFAs since traffic flows of LFAs are indistinguishable from legitimate ones.

Besides cyber security, national security has also been studied based on game theory. For example, to deter fare evasion, suppress urban crime and counter terrorism, TRUSTS was evaluated in the Los Angeles Metro system [69], and GUARDS was deployed for security inside the airport by the US Transportation Security Administration [70]. To randomize checkpoints within the airport terminals and on the roadways entering the airport, ARMOR has been deployed at the Los Angeles International airport [71]. A randomized deployment of the US Federal Air Marshal Service using game theory has also been adopted [72]. To conduct randomized patrolling for the US Coast Guard, PROTECT was deployed

[73]. Different from these studies protecting national security, we focus on cyber security.

10 Conclusion

Armed with powerful game-theoretic tools, we make the first effort towards detection strategies of LFAs, an emerging threat against Internet infrastructure. We formulate the problem as a Stackelberg security game, and design optimal detection strategies in consideration of the adversary's behavior, which we believe is a significant step forward in formally understanding LFA detection strategies. The proposed strategy is randomized (like security patrolling) to make the specific resource assignment unpredictable to the adversary at the time of scheduled attacks, while simultaneously optimizing the defender utility. Moreover, best and quantal response models are leveraged to characterize the adversary's behavior. We demonstrate that, compared with strategies such as uniform-detection (i.e., probing each path uniformly at random) and best-detection (i.e., constantly probing the path containing the most important link), the proposed strategy can maximize the detection utility. All the results suggest the necessity and effectiveness of our solutions in handling LFAs by taking into account the asset importance of links and the adversary's behavior.

Acknowledgments

This work is supported in part by National Natural Science Foundation (61602371, 61772411, U1736205, 61632013), CCF-NSFOCUS KunPeng Research Fund (2018002), Hong Kong ITF (No. UIM/285) and Hong Kong RGC Project No. PolyU5389/13E, PolyU152279/16E, Natural Science Basic Research Plan in Shaanxi Province (2016JQ6034), SZSTI JCYJ20170816100819428, Special Foundation of China Postdoctoral Science (2018T111065), China Postdoctoral Science Foundation (2015M582663), the Fundamental Research Funds for the Central Universities, Shaanxi Province Postdoctoral Science Foundation, of China.

References

[1] Y. Chen, K. Hwang, and W. S. Ku,"Collaborative detection of ddos attacks over multiple network domains," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 12, pp. 1649–1662, Dec. 2007.

[2] Z. Tan, A. Jamdagni, X. He, P. Nanda, R. P. Liu, and J. Hu, "Detection of denial-of-service attacks based on computer vision techniques," *IEEE Trans. Comput.*, vol. 64, no. 9, pp. 2519–2533, Sep. 2015.

[3] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 2046–2069, Oct.-Dec. 2013.

[4] J. Mirkovic, A. Hussain, S. Fahmy, P. Reiher, and R. K. Thomas, "Accurately measuring denial of service in simulation and testbed experiments," *IEEE Trans. Dependable Secure Comput.*, vol. 6, no. 2, pp. 81–95, Apr.-Jun. 2009.

[5] F. J. Ryba, M. Orlinski, M. Wählisch, C. Rossow, and T. C. Schmidt, "Amplification and drdos attack defense - A survey and new perspectives," *CoRR*, vol. abs/1505.07892, pp. 1–26, 2015.

[6] G. Pellegrino, C. Rossow, F. J. Ryba, T. C. Schmidt, and M. Wählisch, "Cashing out the great cannon? on browser-based ddos attacks and economics," in *Proc. USENIX WOOT*, 2015, pp. 1–8.

[7] C. Rossow, "Amplification hell: Revisiting network protocols for ddos abuse," in *Proc. NDSS*, 2014, pp. 1–15.

[8] M. Kührer, T. Hupperich, C. Rossow, and T. Holz, "Exit from hell? reducing the impact of amplification ddos attacks," in *Proc. USENIX Secur.*, 2014, pp. 111–125.

[9] M. Kührer, T. Hupperich, C. Rossow, and T. Holz, "Hell of a handshake: Abusing TCP for reflective amplification ddos attacks," in *Proc. USENIX WOOT*, 2014, pp. 4–4.

[10] M. Kang, V. Lee, and S. Gligor, "The crossfire attack," in *Proc. IEEE Symp. Secur. Privacy*, 2013, pp. 127–141.

[11] P. Bright, "Can a ddos break the internet?" 2013. [Online]. Available: http://goo.gl/oM6XJt

[12] M. S. Kang and V. D. Gligor, "Routing bottlenecks in the internet: Causes, exploits, and countermeasures," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2014, pp. 321–333.

[13] M. S. Kang, V. D. Gligor, and V. Sekar, "Spiffy: Inducing cost-detectability tradeoffs for persistent link-flooding attacks," in *Proc. NDSS*, 2016, pp. 1–15.

[14] S. Lee, M. Kang, and V. Gligor, "Codef: Collaborative defense against large-scale link-flooding attacks," in *Proc. 9th ACM Conf. Emerging Netw. Exp. Technol.*, 2013, pp. 417–428.

[15] S. Lee and V. Gligor, "FLoc: Dependable link access for legitimate traffic in flooding attacks," in *Proc. IEEE 30th Int. Conf. Distrib. Comput. Syst.*, 2010, pp. 327–338.

[16] A. Athreya, X. Wang, Y. S. Kim, Y. Tian, and P. Tague, "Resistance is not futile: Detecting ddos attacks without packet inspection," in *Proc. 14th Int. Workshop Inf. Secur. Appl.*, Aug. 2013, pp. 174–188.

[17] L. Xue, X. Luo, E. W. W. Chan, and X. Zhan, "Towards detecting target link flooding attack," in *Proc. 28th USENIX Conf. Large Installation Syst. Admin.*, 2014, pp. 81–96.

[18] N. Hu, L. E. Li, Z. M. Mao, P. Steenkiste, and J. Wang, "Locating internet bottlenecks: Algorithms, measurements, and implications," in *Proc. Conf. Appl. Technol. Archit. Protocols Comput. Commun.*, 2004, pp. 41–54.

[19] P. Calyam, C.-G. Lee, E. Ekici, M. Haffner, and N. Howes, "Orchestration of network-wide active measurements for supporting distributed computing applications," *IEEE Trans. Comput.*, vol. 56, no. 12, pp. 1629–1642, Dec. 2007.

[20] D. Croce, M. Mellia, and E. Leonardi, "The quest for bandwidth estimation techniques for large-scale distributed systems," *SIGMETRICS Perform. Evalation Rev.*, vol. 37, no. 3, pp. 20–25, Jan. 2010.

[21] X. Luo, E. Chan, and R. Chang, "Design and implementation of TCP data probes for reliable and metric-rich network path monitoring," in *Proc. Conf. USENIX Annu. Tech. Conf.*, 2009, p. 4.

[22] P. Wang, X. Guan, J. Zhao, J. Tao, and T. Qin, "A new sketch method for measuring host connection degree distribution," *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 6, pp. 948–960, Jun. 2014.

[23] A. Jaggard, S. Kopparty, V. Ramachandran, and R. Wright, "The design space of probing algorithms for network-performance measurement," in *Proc. ACM SIGMETRICS/Int. Conf. Meas. Model. Comput. Syst.*, 2013, pp. 105–116.

[24] E. Blanton, S. Fahmy, G. Frederickson, and S. Gangam, "On the cost of network inference mechanisms," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 4, pp. 662–672, Apr. 2011.

[25] S. Ekelin, A. Johnsson, and C. Flinta, "Scalability and dimensioning of network-capacity measurement system using reflecting servers," *CoRR*, vol. abs/1505.06310, 2015. [Online]. Available: http://arxiv.org/abs/1505.06310

[26] D. Gkounis, V. Kotronis, C. Liaskos, and X. Dimitropoulos, "On the interplay of link-flooding attacks and traffic engineering," *SIGCOMM Comput. Commun. Rev.*, vol. 46, no. 2, pp. 5–11, May 2016.

[27] Paris Traceroute. (2019). [Online]. Available: http://www.paris-traceroute.net/

[28] A. Studer and A. Perrig, "The coremelt attack," in *Proc. 14th Eur. Conf. Res. Comput. Secur.*, 2009, pp. 37–52.

[29] D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, and M. Tambe, "Stackelberg vs. nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness," *J. Artif. Intell. Res.*, vol. 41, no. 2, pp. 297–327, May 2011.

[30] M. Tambe, *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge, U.K.: Cambridge Univ. Press, 2011.

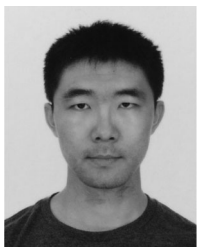
[31] P. Paruchuri, J. P. Pearce, J. Marecki, M. Tambe, F. Ordonez, and S. Kraus, "Playing games for security: An efficient exact algorithm for solving Bayesian Stackelberg games," in *Proc. 7th Int. Joint Conf. Auton. Agents Multiagent Syst.*, 2008, pp. 895–902.

[32] B. An, M. Tambe, F. Ordóñez, E. A. Shieh, and C. Kiekintveld, "Refinement of strong Stackelberg equilibria in security games," in *Proc. 25th AAAI Conf. Artif. Intell.*, 2011, pp. 587–593.

[33] A. Anandkumar, A. Hassidim, and J. Kelner, "Topology discovery of sparse random graphs with few participants," in *Proc. ACM SIGMETRICS*, 2011, pp. 253–264.
[34] B. Donnet and T. Friedman, "Internet topology discovery: A survey," *IEEE Commun. Surveys Tuts.*, vol. 9, no. 4, pp. 56–69, 2007.
[35] iplane: Datasets. (2018). [Online]. Available: http://iplane.cs.washington.edu/data/data.html
[36] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "iplane: An information plane for distributed services," in *Proc. 7th Symp. Operating Syst. Des. Implementation*, 2006, pp. 367–380.
[37] A. Blum, N. Haghtalab, and A. D. Procaccia, "Lazy defenders are almost optimal against diligent attackers," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 573–579.
[38] G. Karame, B. Danev, C. Bannwart, and S. Capkun, "On the security of end-to-end measurements based on packet-pair dispersions," *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 1, pp. 149–162, Jan. 2013.
[39] M-Lab. (2019). [Online]. Available: http://www.measurementlab.net/
[40] mPlane. (2019). [Online]. Available: http://www.ict-mplane.eu/
[41] RIPE Atlas. (2019). [Online]. Available: https://atlas.ripe.net/
[42] perfSONAR. (2019). [Online]. Available: http://www.perfsonar.net/
[43] M. Alenazi and J. Sterbenz, "Comprehensive comparison and accuracy of graph metrics in predicting network resilience," in *Proc. 11th Int. Conf. Des. Reliable Commun. Netw.*, 2015, pp. 157–164.
[44] V. Conitzer and T. Sandholm, "Computing the optimal strategy to commit to," in *Proc. 7th ACM Conf. Electron. Commerce*, 2006, pp. 82–90.
[45] Handout on mixed strategies. (2019). [Online]. Available: https://oyc.yale.edu/sites/default/files/mixed_strategies_handout_0_0.pdf
[46] R. McKelvey and T. Palfrey, "Quantal response equilibria for normal form games," *Games and Economic Behavior*, vol. 10, no. 1, pp. 6–38, 1995.
[47] M.-H. Lin, ohn Gunnar Carlsson, D. Ge, J. Shi, and J.-F. Tsai, "A review of piecewise linearization methods," *Math. Problems Eng.*, vol. 2013, 2013, Art. no. 101376.
[48] CPLEX Optimizer. (2019). [Online]. Available: https://www.ibm.com/analytics/cplex-optimizer
[49] L. Xue, X. Ma, X. Luo, E. W. W. Chan, T. T. N. Miu, and G. Gu, "Linkscope: Toward detecting target link flooding attacks," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 10, pp. 2423–2438, Oct. 2018.
[50] Traceroute Data. (2019). [Online]. Available: https://drive.google.com/open?id=1FW-p84uEtqsVAkqsa33CmQBak6rCV82s
[51] J. Gan and B. An, "Minimum support size of the defenders strong stackelberg equilibrium strategies in security games," in *Proc. AAAI Spring Symp. Appl. Comput. Game Theory*, 2013, pp. 1–7.
[52] A. Botta, A. Dainotti, and A. Pescapè, "A tool for the generation of realistic network workload for emerging networking scenarios," *Comput. Netw.*, vol. 56, no. 15, pp. 3531–3547, 2012.
[53] F. Golkar, T. Dreibholz, and A. Kvalbein, "Measuring and comparing internet path stability in ipv4 and ipv6," in *Proc. Int. Conf. Workshop Netw. Future*, 2014, pp. 1–5.
[54] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—I," *Math. Program.*, vol. 14, no. 1, pp. 265–294, 1978.
[55] A. Krause and D. Golovin, "Submodular function maximization," *Tractability: Practical Approaches Hard Problems*, vol. 3, no. 19, 2012, Art. no. 8.
[56] V. E. Paxson, "Measurements and analysis of end-to-end internet dynamics," Ph.D. dissertation, Comput. Sci. Division, Univ. California, Berkeley, Berkeley, CA, USA, 1998, uMI Order No. GAX98–03325.
[57] L. Quan, J. Heidemann, and Y. Pradkin, "Trinocular: Understanding internet reliability through adaptive probing," in *Proc. ACM SIGCOMM*, 2013, pp. 255–266.
[58] Y. Zhang, Z. Mao, and M. Zhang, "Detecting traffic differentiation in backbone ISPs with NetPolice," in *Proc. 9th ACM SIGCOMM Conf. Internet Meas.*, 2009, pp. 103–115.
[59] E. Katz-Bassett, H. Madhyastha, J. John, A. Krishnamurthy, D. Wetherall, and T. Anderson, "Studying black holes in the internet with hubble," in *Proc. 5th USENIX Symp. Netw. Syst. Des. Implementation*, 2008, pp. 247–262.
[60] Y. Liu, X. Luo, R. Chang, and J. Su, "Characterizing inter-domain rerouting by betweenness centrality after disruptive events," *IEEE J. Select. Areas Commun.*, vol. 31, no. 5, pp. 1147–1157, Jun. 2013.
[61] W. Fok, X. Luo, R. Mok, W. Li, Y. Liu, E. Chan, and R. Chang, "Monoscope: Automating network faults diagnosis based on active measurements," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage.*, 2013, pp. 210–217.
[62] Y. Zhang, Z. Mao, and M. Zhang, "Effective diagnosis of routing disruptions from end systems," in *Proc. USENIX NSDI*, 2008, pp. 219–232.
[63] X. Luo, L. Xue, C. Shi, Y. Shao, C. Qian, and E. Chan, "On measuring one-way path metrics from a web server," in *Proc. IEEE 22nd Int. Conf. Netw. Protocols*, 2014, pp. 203–208.
[64] F. Gillani, E. Al-Shaer, S. Lo, Q. Duan, M. Ammar, and E. Zegura, "Agile virtualized infrastructure to proactively defend against cyber attacks," in *Proc. IEEE INFOCOM*, 2015, pp. 729–737.
[65] C. Liaskos, V. Kotronis, and X. Dimitropoulos, "A novel framework for modeling and mitigating distributed link flooding attacks," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun.*, 2016, pp. 1–9.
[66] J. Xu and W. Lee, "Sustaining availability of web services under distributed denial of service attacks," *IEEE Trans. Comput.*, vol. 52, no. 2, pp. 195–208, Feb. 2003.
[67] Q. Wu, S. Shiva, S. Roy, C. Ellis, and V. Datla, "On modeling and simulation of game theory-based defense mechanisms against dos and ddos attacks," in *Proc. Spring Simul. Multiconference*, 2010, Art. no. 159.
[68] G. Yan, R. Lee, A. Kent, and D. Wolpert, "Towards a Bayesian network game framework for evaluating ddos attacks and defense," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2012, pp. 553–566.
[69] Z. Yin, A. Jiang, M. Johnson, M. Tambe, C. Kiekintveld, K. Leyton-Brown, T. Sandholm, and J. Sullivan, "Trusts: Scheduling randomized patrols for fare inspection in transit systems," in *Proc. AAAI Artif. Intell.*, 2012, pp. 2348–2355.
[70] J. Pita, M. Tambe, C. Kiekintveld, S. Cullen, and E. Steigerwald, "Guards: Game theoretic security allocation on a national scale," in *Proc. 10th Int. Conf. Auton. Agents Multiagent Syst. - Vol. 1*, 2011, pp. 37–44.
[71] J. Pita, M. Jain, J. Marecki, F. Ordóñez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus, "Deployed armor protection: The application of a game theoretic model for security at the los angeles international airport," in *Proc. 7th Int. Joint Conf. Autonomous Agents Multiagent Syst.: Ind. Track*, 2008, pp. 125–132.
[72] J. Tsai, S. Rathi, C. Kiekintveld, F. Ordonez, and M. Tambe, "Iris - a tool for strategic security allocation in transportation networks," in *Proc. 8th Int. Joint Conf. Auton. Agents Multiagent Syst.*, 2009, pp. 1–8.
[73] B. An, E. Shieh, M. Tambe, R. Yang, C. Baldwin, J. DiRenzo, B. Maule, and G. Meyer, "PROTECT - A deployed game theoretic system for strategic security allocation for the united states coast guard," *AI Mag.*, vol. 33, no. 4, pp. 96–110, 2012.

Xiaobo Ma received the PhD degree in control science and engineering from Xian Jiaotong University, and was a post-doctoral research fellow with The Hong Kong Polytechnic University. He is an associate professor with MOE Key Lab for Intelligent Networks and Network Security, School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China. He is also with Shaanxi Province Key Laboratory of Computer Networks, Xi'an Jiaotong University, China. His research interests include network security and privacy. He is a member of the IEEE.

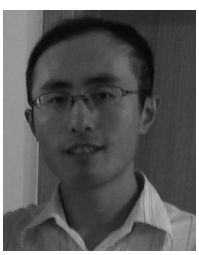
Bo An received the PhD degree in computer science from the University of Massachusetts, Amherst. He is a Nanyang associate professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His current research interests include artificial intelligence, multiagent systems, game theory, and optimization. He is a member of the editorial board of JAIR and an associate editor of JAAMAS. He was elected to the board of directors of IFAAMAS.



Mengchen Zhao received the BS degree in applied mathematics from Sun Yat-Sen University, China. He is working toward the PhD degree in the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include game theory, machine learning, reinforcement learning and their applications to real-world problems.

Xiapu Luo received the PhD degree in computer science from The Hong Kong Polytechnic University, and was a post-doctoral research fellow with the Georgia Institute of Technology. He is an assistant professor with the Department of Computing and an associate researcher with the Shenzhen Research Institute, The Hong Kong Polytechnic University. His research focuses on smartphone security and privacy, network security and privacy, and internet measurement.

Lei Xue received the PhD degree in computer science from The Hong Kong Polytechnic University. He is currently a post-doctoral research fellow with the Department of Computing, The Hong Kong Polytechnic University. His current research interests include network security, mobile security, and network measurement.



Zhenhua Li received the BSc and MSc degrees from Nanjing University, in 2005 and 2008, and the PhD degree from Peking University, in 2013, all in computer science and technology. He is an assistant professor with the School of Software, Tsinghua University. His research areas cover cloud computing/storage/download, big data analysis, content distribution, and mobile Internet.

Tony T. N. Miu received the BS degree in computer science from The Hong Kong Polytechnic University. He is currently a security researcher with Nexusguard Ltd. His current research interests include network security, especially DDoS detection, and defense.

Xiaohong Guan received the PhD degree in electrical engineering from the University of Connecticut, Storrs, in 1993. He has been with the Systems Engineering Institute, Xian Jiaotong University, where he is currently a Cheung Kong professor of systems engineering and the dean of the School of Electronic and Information Engineering. His research interests include allocation and scheduling of complex networked resources, and network security. Since 1995, he is also with the Department of Automation, Tsinghua National Laboratory for Information Science and Technology, and the Center for Intelligent and Networked Systems, TNLIST, Tsinghua University. He is a fellow of the IEEE and an academician of Chinese Academy of Sciences.