Game Theoretic Path Selection to Support Security in Device-to-Device Communications

Emmanouil Panaousis[a], Eirini Karapistoli[b], Hadeer Elsemary[c], Tansu Alpcan[d], MHR Khuzani[e], Anastasios A. Economides[f]

[a] *University of Brighton, UK*
[b] *Capritech Limited, UK*
[c] *University of Gottingen, Germany*
[d] *University of Melbourne, Australia*
[e] *Queen Mary University of London, UK*
[f] *University of Macedonia, Greece*

Abstract

[1] Device-to-Device (D2D) communication is expected to be a key feature supported by 5G networks, especially due to the proliferation of Mobile Edge Computing (MEC), which has a prominent role in reducing network stress by shifting computational tasks from the Internet to the mobile edge. Apart from being part of MEC, D2D can extend cellular coverage allowing users to communicate directly when telecommunication infrastructure is highly congested or absent. This significant departure from the typical cellular paradigm imposes the need for decentralised network routing protocols. Moreover, enhanced capabilities of mobile devices and D2D networking will likely result in proliferation of new malware types and epidemics. Although the literature is rich in terms of D2D routing protocols that enhance quality-of-service and energy consumption, they provide only basic security support, e.g., in the form of encryption. Routing decisions can, however, contribute to collaborative detection of mobile malware by leveraging different kinds of anti-malware software installed on mobile devices. Benefiting from the cooperative nature of D2D communications, devices can rely on each other's contributions to detect malware. The impact of our work is geared towards having more malware-free D2D networks. To achieve this, we designed and implemented a novel routing protocol for D2D communications that optimises routing decisions for explicitly improving malware detection. The protocol identifies optimal network paths, in terms of malware mitigation and energy spent for malware detection, based on a *game theoretic model*. Diverse capabilities of

[1] ©⟨2016⟩. This manuscript version is made available under the CC-BY-NC-ND 4.0 license http://creativecommons.org/licenses/by-nc-nd/4.0/.
DOI: 10.1016/j.adhoc.2016.11.008.

network devices running different types of anti-malware software and their potential for inspecting messages relayed towards an intended destination device are leveraged using game theoretic tools. An optimality analysis of both Nash and Stackelberg security games is undertaken, including both zero and non-zero sum variants, and the Defender's equilibrium strategies. By undertaking network simulations, theoretical results obtained are illustrated through randomly generated network scenarios showing how our protocol outperforms conventional routing protocols, in terms of expected payoff, which consists of: *security damage inflicted by malware* and *malware detection cost*.

Keywords: Device-to-Device (D2D) communications, iRouting protocol, Malware detection games, Game theory.

---

1. Introduction

Demand for anytime-anywhere wireless broadband connectivity and increasingly stringent Quality of Service (QoS) requirements pose new research challenges. As mobile devices are capable of communicating in both cellular (e.g. 4G) and unlicensed (e.g. IEEE 802.11) spectrum, the Device-to-Device (D2D) networking paradigm has the potential to bring several immediate gains. Networking based on D2D communication [1, 2, 3, 4, 5] not only facilitates wireless and mobile peer-to-peer services, but also provides energy efficient communications, locally offloading computation, offloading connectivity, and high throughput. The most emerging feature of D2D is the establishment and use of multi-hop paths to enable communications among non-neighbouring devices. In multi-hop D2D communications, data are delivered from a source to a destination via intermediate (i.e. relaying) devices, independently of operators' networks.

1.1. Motivation

To motivate the D2D communication paradigm, we emphasise the need for *localised applications*. These run in a collaborative manner by groups of devices at a location where telecommunications infrastructures: (i) are not present at all, e.g. underground stations, airplanes, cruise ships, parts of a motorway, and mountains; (ii) have collapsed due to physical damage to the base stations or insufficient available power, e.g. areas affected by a disaster such as earthquake; or (iii) are over congested due to an extremely crowded network, e.g. for events in stadiums, and public celebrations. Furthermore, relay by device can be leveraged for commercial purposes such as advertisements and voucher distributions for instance in large shopping centres. This is considered a more efficient way of promoting businesses than other traditional methods such as email broadcasting and SMS messaging due to the immediate identification of the clients in a

surrounding area. Home automation and building security are another two areas that multi-hop data delivery using D2D communications is likely to overtake our daily life in the near future while multi-hop D2D could be also leveraged towards the provision of anonymity against cellular operators [6].

A key question related to multi-hop D2D networks is, *which route should the originator of some data choose to send it to an intended destination?*. This has been exhaustively investigated in the literature of wireless and mobile ad hoc routing with well-known protocol to be among others AODV [7], DSR [8], and OLSR [9]. A thorough survey of standardisation efforts in this field has been published by Ramrekha et al. [10].

Due to the myriad number of areas D2D communications are applicable to, devices are likely to be an ideal target for attackers who aim to infect devices with malware. Authors in [11] point out that malware in current smartphones and tablets have recently rocketed and established its presence through advanced techniques that bypass security mechanisms of devices. Malware can spread, for instance, through a Multimedia Messaging System (MMS) with infected attachments, or an infected message received via Bluetooth aiming at stealing users' personal data or credit stored in the device. An example of a well-known worm that propagates through Bluetooth was Cabir, which consists of a message containing an application file called `caribe.sis`. Apart from malware infection, Khuzani et al. [12] have investigated outbreaks of malware (i.e. malware epidemics) mainly by adopting the notion of D2D communication. Finally, social engineering attacks against mobile phones is one of the most serious threats, as presented in a relevant survey here [13]. For thorough surveys on mobile malware one may refer to [11, 14].

1.2. Innovation

In a nutshell,this paper presents a novel routing protocol, for D2D communications, that supports malware detection in an optimal way by using non-cooperative *game theoretic* tools, which have been extensively used in the security literature (e.g. [15]) and in D2D routing (e.g. [16]). Game theory has also been used for other than routing purposes [17], [18, 19] in D2D networks. In this paper we only focus on security games and we tackle a decision-making routing challenge, in D2D networks, in presence of an adversary who injects malware into the network, after she has compromised a gateway that connects the D2D network with the cloud. This assumption is fairly realistic given the vast power attackers have in their hands these days to successfully exploit vulnerabilities of modern gateways. Our underlying network has been inspired by the *Mobile Edge Computing* (MEC) (also refer to as Fog Computing) paradigm as a step towards addressing security within the realm of an increasingly important area of 5G.

Our protocol, called *i*Routing (abbreviating "intelligent Routing"), is designed upon the theoretical analysis of a simple yet illuminating two-player security game between the *Defender*, which abstracts a D2D network, and the *Attacker*, which abstracts any adversarial entity that wishes to inject malware into the D2D network. We have proven that the Defender's *equilibrium strategies* leave the network better off, in terms of *expected payoff*, which is a combination of *security damage* and *malware detection cost* (i.e. cycles process units). Note that *i*Routing can work on top of underlying physical and MAC layer protocols [20, 21].

It is worth noting that this paper does not tackle secure routing issues in traditional ways. For a survey of secure routing protocols for wireless ad hoc networks, see [22, 23]. Such protocols mainly aim at enabling confidentiality, and integrity of the communicated data and they do not consider underlying collaborative malware detection.

1.3. Progress beyond relevant work

This paper extends, in a significant manner, the results initially presented in [24]. The exact differences are summarized below.

- [24] assumes a pure device-to-device network while in this paper the device-to-device network has been enriched with a part of mobile edge computing. The network devices request services from the MEC server and multi-hopping enables communication between the MEC server and the different devices to overcome proximity issues due to the latter being outside the transmission range of the server. In this paper, the security challenge is how to safely utilise MEC services where a cluster-head (i.e. MEC server) might be compromised by an adversary. Although this does not introduce any new challenge in terms of malware detection and routing, it is an assumption that places the idea of the paper within mobile edge computing and 5G architectures.

- This paper assumes different mobile operating systems and these can be infected with different types of malware as opposed to [24], which goes as far as considering just a set of malicious messages that are sent from the attacker's device to infect the legitimate devices. This also has the effect of defining, in this paper, the Malware Detection Game whereas in [24], the defined game is called Secure Message Delivery Game.

- In [24], a confusion matrix is defined to determine how the different devices of the network can detect malicious messages. In this paper here we take a more realistic, in the terms of cyber security, approach where for each device there is a probability to be compromised by malware. Therefore,

each route has, in turn, a penetration level, which is the probability the route to be compromised due to one or more devices on it being vulnerable.

- In [24], the details about the interdependencies of malicious message detectors is not discussed, while in our paper here we explicitly say that each control detects different signs of malware and *no interdependencies*, in terms of detection capabilities, are assumed, i.e. we have assumed that an anti-malware control is the minimal piece of software that detects certain malicious signs.

- In [24], the Attacker is not assumed to monitor the network before launching a malware attack (no reconnaissance) while in our paper here the Attacker surveils the network before injecting malware giving us a Stackelberg game to study.

- In [24], only Nash Equilibria (NE) and maximin strategies have been studied. On the other hand, our paper here derives Strong Stackelberg Equilibria (SSE) and shows the relationship among three of them; SSE, NE and maximin. Not only that, but this paper exhibits much larger depth of mathematical analysis referring also to best responses of players. Finally, it proves the equality of strategies of different games, such zero-sum and non-zero sum across all strategic types (Nash, Stackelberg, maximin).

- Although Panaousis et al. [24] has investigated both zero sum and non-zero sum games, where in the latter the utility of the Attacker is a positive affine transformation (PAT) of the defender's utility, in this paper we go beyond that. We show the equality of the different strategies holds in a more generic (i.e. than the PAT case) payoff structure where the Attackers utility is a strictly positive scaling of the Defender's utility.

- All simulations in [24] were numeric; as well as they do not compare the performance of the proposed routing protocol with other device-to-device routing protocols. For the purposes of our paper here we have undertaking a network simulation to compare the proposed protocol with legacy routing protocols using the OMNeT++ network simulator. In this way we have simulated physical and link-layer network characteristics.

- In our paper here, we have considered, in our simulations, the efficacies of some of the most-recent real-world anti-malware controls against real-world malware types as opposed to the purely numeric assignment to the different variables.

- In our simulations here, we have included a new Attacker type, called Weighted, which allows the adversary to distribute her resources propor-

tionally, over the different routes, aiming at the highest expected damage. This type of Attacker was not simulated in [24].

1.4. Main assumptions

Our analysis assumes that each device has some malware detection capabilities (e.g. anti-malware software). Therefore, a device is able to detect malicious application-level events. In other words, each device has its own detection rate which contributes towards the overall detection rate of the routes that this device is part of. In order to increase malware detection, the route with the highest detection capabilities must be selected to relay the message to the destination.

However, due to the different malware types available to attackers, these days, such a decision is not trivial. One could argue that if we know the probability of a malware type to be chosen, we can develop a proportional routing strategy, which will distribute security risks across the different routes by choosing routes in a proportional, to their malware detection capabilities, manner. Since this knowledge can not be taken for granted in addition to the volatile nature of such statistics, in this paper we use game theory to optimise routing decisions to support malware detection in D2D networks, regardless of the probability of the different malware to be used by the Attacker.

1.5. Outline

The remainder of this paper is organised as follows: In Section 2, we review related work with more emphasis to be given in papers at the intersection of game theory, security, and routing for wireless ad hoc networks (i.e. prominent example of D2D networking). In Section 3, we present the system and game models, while in Section 4, we devise game solutions. In Section 5, we undertake optimality analysis which leads to a list of theoretic contributions. Section 6 describes, in detail, the *i*Routing protocol, and in Section 7, we compare *i*Routing against other routing protocols. Finally, Section 8 provides concluding remarks and points towards future research.

2. Related work

In this section, we briefly review the state-of-the-art, in chronological order, in terms of game theoretic approaches at the intersection of three fields: security, routing, and device-to-device networks. Another set of game theoretic works that focus on optimising intrusion detection strategies per se than adjusting routing decisions to optimally support intrusion detection, consist of papers such as [25], [26], [27], [27], [28], [29], [30], and [31]. Our work is complementary to this literature as it optimises end-to-end path selections, in terms of malware detection efficacy and computational effort.

Looking more into decision regarding packet forwarding by using game theoretic tools and without incentive mechanisms in place, Felegyhazi et al. [32] have studied the Nash equilibria of packet forwarding strategies with tit-for-tat punishment strategy in an iterative game. In each stage (i.e. time slot) of the game, each device selects its cooperation level based on the normalised throughput it experienced in the previous stage. As opposed to *i*Routing, the authors do not propose a new end-to-end routing protocol; instead they consider a shortest path algorithm. Also, they assume the existence of internal malicious or selfish nodes in contrast to our work here, which models an adversary outside of the D2D cluster, who aims to infect legitimate devices with malware.

In a more security-oriented vein, Yu et al. [33] have used game theory to study the dynamic interactions, in mobile ad hoc (device-to-device) networks, between "good" nodes, which initially believe that all other nodes are not malicious, and "adversaries", which are aware of which nodes are good. They propose secure routing and packet forwarding games that consist of 3 stages: route participation; route selection; and packet forwarding. In the first stage, a node decides whether to be part of route or not; in the second phase, a node who wishes to send a packet to a destination, after it discovers a *valid route* (called when all nodes agree to be part of it), it either uses the discovered route or not; and, finally, in the third phase, each relay node decides to forward or not an incoming packet. They have derived optimal defence strategies and studied the maximum potential damage, which incurs when attackers find a route with maximum number of hops and they inject malicious traffic into it. The same authors also combined this game with a secure routing game but without considering noise and imperfect monitoring. Yu et al. [34] extended [33] and proposed a secure cooperation game under noise and imperfect monitoring. Likewise, Yu and Liu tackled the same challenge and presented a richer set of performance evaluation results in [35]. The above publications do not tackle the same challenge with *i*Routing, as they do not investigate the selection of a route among an available set of routes to deliver packets from a source to a destination

Finally, in [36], Panaousis and Politis present a routing protocol that respects the energy spent by intrusion detection on each route and therefore prolonging network lifetime. This paper takes a simple approach, according to which the attacker either attacks or not a route, and the Defender, likewise, decides whether to allocate resources to defend or not.

None of the aforesaid protocols consider the propagation of malware within the network and none of these works investigates Stackelberg games, which basically assume that the Attacker conducts surveillance before deciding upon her strategy. This is a reasonably realistic assumption when looking at the intelligence of cyber hackers and it is a conventional decision in other security related fields [37, 38, 39, 40].

3. System description and game model

This section presents our underlying system model along with its components. Mobile-edge computing (MEC) is an emerging paradigm that allows mobile applications to offload computationally intensive workloads to a MEC server. This introduces a new network architecture concept that provides cloud-computing capabilities at the edge of the mobile network. The MEC server is likely to be setup by a service provider to ensure that it can provide a service environment with very low latency and high-bandwidth.

3.1. System description

We use a motivational paradigm demonstrating how D2D communication can be combined with a MEC architecture [41], as depicted in Fig. 1. In our model, MEC is an intermediate layer between a *D2D cluster* and the *cloud*, aiming at *low-latency service delivery* from the latter to the former, and it can serve users by using local short-distance high-rate connections. The intermediate layer can contain a number of deployed MEC servers aiming to handle the localised requests issued by cluster users.

We assume that devices within a cluster can communicate in a D2D manner: directly or by using multi-hop routes. The cluster is formed based on discovery protocols that run in each device. These allow to sense the environment and create a list of one-hop neighbours in order to be able to communicate should any request to forward data or a direct request be sent. We also assume no cellular infrastructure within the cluster, which means that devices can only communicate in a device-to-device fashion.

It is envisaged that such scenarios will be very common in 5G ecosystems where heterogeneous wireless technologies (e.g. NB-LTE, WiFi, ZigBee, Bluetooth) will facilitate D2D communication [3]. For example, a device that seeks some data, can request this from other devices in its cluster, and if the Request cannot be served the MEC servers must be contacted to assist with the discovery of this data.

The idea here is that a MEC server is dedicated to provide predefined service applications to cluster users without the need to communicate with the cloud so that it accelerates responses while *"pushing" the cloud away of the user*. We assume that each D2D cluster has a *cluster-head* [42], which is a device that communicates with the MEC servers. The main functionalities of a cluster-head are (i) to forward the Request of a device to the MEC servers, and (ii) upon its response, to transmit the Reply back to the requestor. In this work, the cluster-head can be any device of the cluster. The MEC server is expected to talk to both the cloud servers and the cluster-head to handle functionalities such as device identifier allocation, call establishment, UE capability tracking, service support, and mobility tracking. Note that the election of the cluster-head is not

investigated in this paper and also this paper is not concerned about deciding the nature of the cluster-head.

3.2. Adversarial model

As any open wireless environment, akin to one described in this paper, can be a target of adversaries. More specifically, in this paper, we assume the existence of a malicious device, called *the Attacker*, that can launch a Man-In-the-Middle (MITM) attack by *hijacking the link* between the cluster-head and MEC servers. Our analysis adopts the Dolev-Yao model [43]. According to this, the D2D network, along with its established connection with the MEC servers, is represented as a *set of abstract entities* that exchange messages. Yet, the adversary is capable of overhearing, intercepting, and synthesising any message and she is only limited by the constraints of the deployed cryptographic methods. We enrich this adversarial model by considering "compromised MEC servers". This is to say that the adversary per se could be inside a *legitimate MEC server* interacting with the cluster-head by using valid credentials and having *privileged access* to MEC servers. In this way, the adversary can inject a fake Reply, crafted with *malware*, and send it back to the data requestor aiming at infecting her device.

3.3. Malware detection

In this adversarial environment, we envisage the use of anti-malware controls running in each device. These can be responsible for *scanning network traffic* for patterns to detect known malicious attempts. Each device may even respond to newly detected attack methods (anomaly-based detection). Upon detection, devices can block messages that are likely to consist of insecure content preventing, in this way, the spread of malware to other devices within their cluster. This assumption can be seen as an advanced application of the *next-generation firewalls* to mobile devices. Although in this paper we assume that any detected malice is blocked by the device that has successfully undertaken the inspection, our work can be extended to support collaborative (e.g. reputation-based) filtering towards blocking messages that end up having a bad reputation. Such an approach can take advantage of *learning* techniques and its investigation will be part of our future work.

3.4. Formulation

Let us assume a cluster of N devices. We denote by `C` its cluster-head, and by `Rqs` the requestor of some data. Henceforth we will refer to this data as `D`. If the latter can not be found within the cluster itself, `Rqs` must seek `D` hosted by the MEC servers of its cluster. Thus, `C` receives a Request from `Rqs`, and it then queries the MEC server.

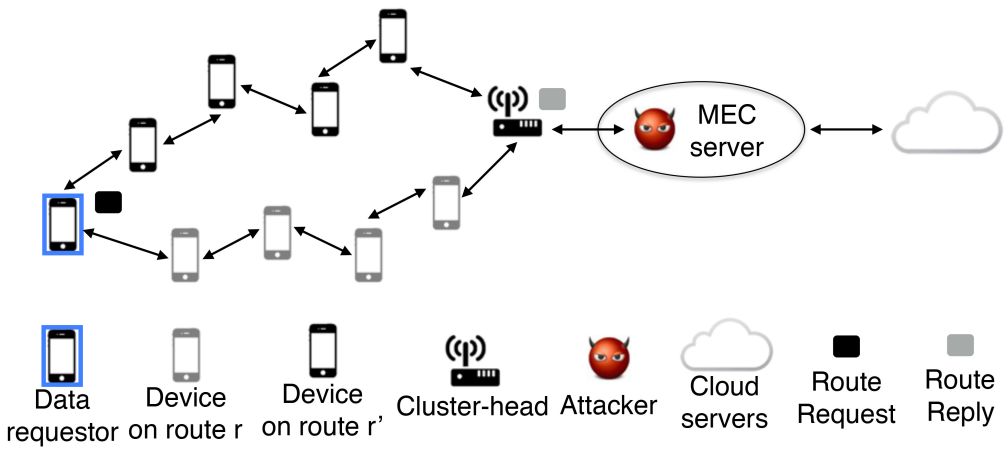

Figure 1: Investigated system model, where a device requests data, that the cluster devices do not possess, from the MEC server. The adversary has successfully launched a MITM attack controlling the communication between cluster-head and MEC server.

When `C` receives back a Reply from the MEC server and `Rqs` is not within its transmission range, a route r must be established to deliver `D` from `C` to `Rqs`. Therefore, there is a need for the devices to relay `D` towards `Rqs`, but before that, `C` *must decide upon* r. We assume R routes available between `C` and `Rqs`, we denote by $r_j \in [R]$, the jth route, and the set of devices that constitute r_j are expressed by $\mathcal{S}_j$. Note that we use the notation $[\Xi]$ to represent the set of Ξ elements.

Although the route selection can be entirely taken based on quality-of-service parameters optimising network delay and jitter, the presence of an Attacker, let it be `A`, introduces uncertainty with regards to the malice of the data conveyed toward `Rqs`. For instance, if `A` controls the link `C` $\Longleftrightarrow$ MEC, then `D` can be anything including malware. If this is the case, `Rqs`, which trusts `C`, is very likely to be infected by this malware. In this paper, the infection risk depends on the likelihood the malware to be collaboratively detected prior to the data being used by `Rqs`. This detection relies on devices that forward packets to `Rqs`, as these are also inspecting the incoming and outgoing network traffic.

Let us consider Λ different mobile operating systems, and M_λ different malware available to the Attacker to infect devices that run a mobile operating system $\lambda \in [\Lambda]$. Each device may run one or more anti-malware controls and for each λ we assume AM_λ anti-malware controls, which can mitigate malware that targets devices running λ.

Let us also assume S devices and a device $s_i \in [S]$, which runs λ, might have available a combination of anti-malware controls given by the set $[AM_\lambda^i] \subseteq$

$[AM_\lambda]$. We use the characteristic function[2] $\mathbf{1}_{[AM^i_\lambda]} : [AM_\lambda] \to \{0,1\}$ defined as follows:

$$\mathbf{1}_{[AM_\lambda]}(a_z) := \begin{cases} 1, & \text{if } a_z \in [AM_\lambda], \\ 0, & \text{if } a_z \notin [AM_\lambda]. \end{cases} \tag{1}$$

to express whether a control a_z is installed in s_i or not.

We express by $d(m_l, a_z) \in [0,1)$ the effectiveness of anti-malware control a_z in mitigating $m_l \in [M_\lambda]$. As a device can run one or more anti-malware controls, and each control a_z has $1 - d(m_l, a_z)$ probability of failing to detect m_l, the probability of s_i failing to detect m_l equals

$$p(s_i, m_l) := \prod_{a_z \in [AM_\lambda] : \mathbf{1}_{[AM_\lambda]}(a_z)=1} \left[1 - d(m_l, a_z)\right]. \tag{2}$$

Note that each control detects different signs of malware and *no interdependencies*, in terms of detection capabilities, are assumed in this paper. To put it differently, we have assumed that an anti-malware control is the minimal piece of software that detects certain malicious signs.

We define as

$$\boldsymbol{p}(s_i) := [p(s_i, m_l)]_{m_l \in [M_\lambda]} \in [0,1]^{M_\lambda}. \tag{3}$$

the vector of *failing detection probabilities*, which captures the *effectiveness* of s_i on detecting malware of the set $[M_\lambda]$. One challenge here is to be able to derive these probabilities in practice. This, for instance, can be done by undertaking thorough penetration tests (i.e. ethical hacking) to assess the efficacy of each anti-malware control. These tests can be performed offline for individual software components and then their combinations can be deployed on the devices. As a result of this we can derive the probability of m_l to infect `Rqs`, when `C` uses the jth route for data delivery, as follows:

$$p(r_j, m_l) := \prod_{s_i \in \mathcal{S}_j} p(s_i, m_l). \tag{4}$$

Thus, we define as $\boldsymbol{p}(r_j) := [p(r_j, m_l)]_{m_l \in [M]}$ the vector of probabilities r_j to be infected by the different malware. For more convenience, Table 1 summarizes the notation used in this paper.

[2]this is a function defined on a set X that indicates membership of an element in a subset X' of X, having the value 1 for all elements of X' and the value 0 for all elements of X not in X'.

Table 1: List of Symbols

Symbol	Description	Symbol	Description
$[N]$	Set of N devices	`C`	Cluster-head
`Rqs`	Data requestor	`D`	Requested data
$[R]$	Set of routes from `C` to `Rqs`	r_j	j-th route
$\mathcal{S}_j$	Set of devices on r_j	`A`	Attacker
$[\Lambda]$	Set of mobile operating systems	λ	Operating system
$[M_\lambda]$	Set of malware that can infect λ	$[AM_\lambda]$	Set of anti-malware controls for λ
$[S]$	Set of devices	s_i	i-th device
m_l	l-th malware	$d(m_l, a_z)$	Effectiveness a_z in mitigating m_l
$p(s_i, m_l)$	Probability of s_i failing to detect m_l	$\boldsymbol{p}(s_i)$	Vector of "failing-to-detect" probabilities of s_i for different malware
$p(r_j, m_l)$	Probability of `Rqs` to be infected with malware m_l when `D` is sent over r_j	$\boldsymbol{p}(r_j)$	Vector of infection probabilities for r_j and all malware types
$[M]$	Set of malware	$\boldsymbol{\rho}$	Defender's mixed strategy
$\boldsymbol{\mu}$	Attacker's mixed strategy	$S(r_j, m_l)$	Expected security damage on route r_j when relaying m_l
$c(s_i)$	Malware detection cost on s_i	$C(r_j)$	Malware detection cost on r_j
$H(m_l)$	Security loss inflicted by m_l	L	path length
$\mathcal{C}_j$	Set of computational malware inspection costs $c(s_i)$ in r_j	$\mathcal{T}_j$	Set of malware inspection capabilities $\boldsymbol{p}(s_i)$ in r_j

3.5. Game model

Now that we have defined our system model by describing its components and their relationship, in the rest of this section, we use game theory to investigate the optimal strategic routing decisions of `C`, the Defender, and the Attacker who aims to infect one of the cluster devices with mobile malware. The Attacker's objective is to succeed an attack against `Rqs` and the Defender must select a route to deliver the Reply to `Rqs`.

We define the *Malware Detection Game* (MDG) between Defender and Attacker, as an *one-shot, bimatrix* game of *complete information* played for each requestor that seek some data. The set of pure strategies of the Defender consists of all possible routes, $r_j \in [R]$, from `C` to `Rqs`. On the other hand, the pure strategies of the Attacker are the different malware $m_l \in [M]$ that can be injected into the D2D network in the form of a Reply. Thus, in MDG a pure strategy profile is a pair of Defender and Attacker actions, $(r_j, m_l) \in [R] \times [M]$ giving a pure strategy space of size $R \times M$. For the rest of the paper, the convention is adopted where the Defender is the row player and the Attacker is the column player.

Each player's preferences are specified by her *payoff function*, and we define

as $U_d : (r_j, m_l) \to \mathbb{R}_-$ and $U_a : (r_j, m_l) \to \mathbb{R}_+$ the payoff functions of the Defender and Attacker, respectively, when the pure strategy profile (r_j, m_l) is played. According to [44], we define a *preference relation* $\succsim$, when m_l is chosen by the Attacker, by the condition $r_x \succsim r_y$, if and only if $U_d(r_x, m_l) \geq U_d(r_y, m_l)$. In general, given the set $[R]$ of all available routes from `C` to `Rqs`, a rational Defender can choose a route (i.e. pure strategy) r^* that is *feasible*, that is $r^* \in [R]$, and *optimal* in the sense that $r^* \succsim r,\ \forall\ r \in [R],\ r \neq r^*$; alternatively she solves the problem $\max_{r \in [R]} U_d(r,\ m_l)$, for a message $m_l \in [M]$. Likewise, we can define the preference relation for the Attacker, where $m_x \succsim m_y \iff U_a(r_j, m_x) \geq U_a(r_j, m_y)$, for a route $r_j \in [R]$.

MDG can be seen as a *game per session*, where the start of each session is signified by the transmission of a new Reply that the cluster-head will send to `Rqs`; it is also realistic to assume that over a time period, there will be multiple sessions. To derive optimal strategies for the Defender during the repetitions of MDGs, we deploy the notion of *mixed strategies*. Since players act independently, we can enlarge their strategy spaces, so as to allow them to base their decisions on the outcome of random events that create uncertainty to the opponent about individual strategic choices maximising their payoffs. Hence, both Defender and Attacker deploy randomised (i.e. mixed) strategies. The mixed strategy $\boldsymbol{\rho}$ of the Defender is a probability distribution over the different routes (i.e. pure strategies) from `C` to `Rqs`, where $\boldsymbol{\rho}(r_j)$ is the probability of delivering a Reply via r_j under mixed strategy $\boldsymbol{\rho}$. We refer to a mixed strategy of the Defender as a *Randomised Delivery Plan* (RDP). For the finite nonempty set $[R]$, let $\Pi_{[R]}$ represent the set of all probability distributions over it, i.e.

$$\Pi_{[R]} := \{\boldsymbol{\rho} \in \mathbb{R}^{+R} | \sum_{r_j \in [R]} \boldsymbol{\rho}(r_j) = 1\}. \tag{5}$$

Therefore a member of $\Pi_{[R]}$ is a mixed strategy of the Defender.

Likewise, the Attacker's mixed strategy is a probability distribution over the different available malware. This is denoted by $\boldsymbol{\mu}$, where $\boldsymbol{\mu}(m_l)$ is the probability of choosing m_l under mixed strategy $\boldsymbol{\mu}$. We refer to a mixed strategy of the Attacker as the *Malware Plan* (MP). Similarly with (5), we express by $\Pi_{[M]}$ the set of all probability distributions over the set of all Attacker's pure strategies given by $[M]$. Thus, a member of $\Pi_{[M]}$ is as a mixed strategy of the Attacker. From the above, the set of mixed strategy profiles of MDG is the Cartesian product of the individual mixed strategy sets, $\Pi_{[R]} \times \Pi_{[M]}$.

Definition 1. *The support of RDP $\boldsymbol{\rho}$ is the set of routes $\{r_j | \boldsymbol{\rho}(r_j) > 0\}$, and it is denoted by supp($\boldsymbol{\rho}$).*

Definition 2. *The support of MP $\boldsymbol{\mu}$ is the set of malware $\{m_l | \boldsymbol{\mu}(m_l) > 0\}$, and it is denoted by supp($\boldsymbol{\mu}$).*

The above definitions state that the subset of routes (resp. malware) that are assigned positive probability by the mixed strategy $\boldsymbol{\rho}$ (resp. $\boldsymbol{\mu}$) is called the *support* of $\boldsymbol{\rho}$ (resp. $\boldsymbol{\mu}$). Note that a pure strategy is a special case of a mixed strategy, in which the support is a single action.

Now that we have defined the mixed strategies of the players, we can define MDG as the finite strategic game $\Gamma = \langle(\text{Defender, Attacker}),\ \Pi_{[R]}\times\Pi_{[M]},\ (U_d, U_a)\rangle$. For a given mixed strategy profile $(\boldsymbol{\rho}, \boldsymbol{\mu}) \in \Pi_{[R]}\times\Pi_{[M]}$, we denote by $U_d(\boldsymbol{\rho}, \boldsymbol{\mu})$, and $U_a(\boldsymbol{\rho}, \boldsymbol{\mu})$ the expected payoff values of the Defender and Attacker, where the expectation is due to the independent randomisations according to mixed strategies $\boldsymbol{\rho}$, and $\boldsymbol{\mu}$.

Formally

$$U_d(\boldsymbol{\rho}, \boldsymbol{\mu}) := \sum_{r_j\in[R]} \sum_{m_l\in[M]} U_d(r_j, m_l)\, \boldsymbol{\rho}(r_j)\, \boldsymbol{\mu}(m_l). \tag{6}$$

and similarly

$$U_a(\boldsymbol{\rho}, \boldsymbol{\mu}) := \sum_{r_j\in[R]} \sum_{m_l\in[M]} U_a(r_j, m_l)\, \boldsymbol{\rho}(r_j)\, \boldsymbol{\mu}(m_l). \tag{7}$$

By using the preference relation we can say that, for an Attacker's mixed strategy $\boldsymbol{\mu}$, the Defender prefers to follow the RDP $\boldsymbol{\rho}$ as opposed to $\boldsymbol{\rho}'$ (i.e. $\boldsymbol{\rho} \succsim \boldsymbol{\rho}'$), if and only if $U_d(\boldsymbol{\rho}, \boldsymbol{\mu}) \geq U_d(\boldsymbol{\rho}', \boldsymbol{\mu})$.

Definition 3. *The Defender's (resp. Attacker's) best response to the mixed strategy $\boldsymbol{\mu}$ (resp. $\boldsymbol{\rho}$) of the Attacker (resp. Defender) is a RDP $\boldsymbol{\rho}^{\mathrm{BR}} \in \Pi_{[R]}$ (resp. $\boldsymbol{\mu}^{\mathrm{BR}} \in \Pi_{[M]}$) such that $U_d(\boldsymbol{\rho}^{\mathrm{BR}}, \boldsymbol{\mu}) \geq U_d(\boldsymbol{\rho}, \boldsymbol{\mu})$, $\forall\ \boldsymbol{\rho} \in \Pi_{[R]}$ (resp. $U_a(\boldsymbol{\rho}, \boldsymbol{\mu}^{\mathrm{BR}}) \geq U_d(\boldsymbol{\rho}, \boldsymbol{\mu})$, $\forall\ \boldsymbol{\mu} \in \Pi_{[M]}$).*

It is noteworthy to mention that the game theoretic solutions that we will propose, in the next section, involve *randomisation*. For instance, in a mixed equilibrium, each player's randomisation leaves the other *indifferent* across her randomisation support. These choices can be deliberately randomised or be taken by software agents that run in mobile devices (i.e. cluster-heads or adversaries). However these are not the only equilibria interpretations. For instance, the probabilities over the pure actions (i.e. route or malware pure selections) can represent (i) time averages of an "adaptive" player, (ii) a vector of fractions of a "population", where each player type adopts pure strategies and, (iii) a "belief" vector that each player has about the other regarding their behaviour.

4. Game solutions

Now that we have defined MDG along with its components, in this section we concentrate in deriving optimal strategies for the Defender. First, we investigate

the problem of determining best RDPs and MPs (i.e. mixed strategies), for the Defender and the Attacker respectively, when both parties are rational decision-makers and they play simultaneously. Note that a *game solution* is a prediction of how rational players may take decisions.

As we have not explicitly defined the *strategic type* of Attacker, we consider different types of solutions based on various Attacker behaviours. This analysis will allow us to draw robust conclusions regarding the *overall optimal* Defender strategy, which will minimise expected damages *regardless of the Attacker type.*

4.1. Nash mixed strategies

The most commonly used solution concept in game theory is that of *Nash Equilibrium* (NE). This concept captures a steady state of the play of the MDG in which Defender and Attacker hold the correct expectation about the other players' behaviour and they act rationally. In other words, an NE dictates optimal responses to each other's actions, keeping the others' strategies fixed, i.e. strategy profiles that are resistant against unilateral deviations of players.

Definition 4. *In any Malware Detection Game (MDG), a mixed strategy profile* $(\boldsymbol{\rho}^{\mathrm{NE}}, \boldsymbol{\mu}^{\mathrm{NE}})$ *of* Γ *is a mixed NE if and only if*

1. $\boldsymbol{\rho}^{\mathrm{NE}} \succsim \boldsymbol{\rho},\ \forall \boldsymbol{\rho} \in \Pi_{[R]}$, *when the Attacker chooses* $\boldsymbol{\mu}^{\mathrm{NE}}$, *i.e.*

$$U_d(\boldsymbol{\rho}^{\mathrm{NE}}, \boldsymbol{\mu}^{\mathrm{NE}}) \geq_{\forall \boldsymbol{\rho} \in \Pi_{[R]}} U_d(\boldsymbol{\rho}, \boldsymbol{\mu}^{\mathrm{NE}}); \tag{8}$$

2. $\boldsymbol{\mu}^{\mathrm{NE}} \succsim \boldsymbol{\mu},\ \forall \boldsymbol{\mu} \in \Pi_{[M]}$, *when the Defender chooses* $\boldsymbol{\rho}^{\mathrm{NE}}$, *i.e.*

$$U_a(\boldsymbol{\rho}^{\mathrm{NE}}, \boldsymbol{\mu}^{\mathrm{NE}}) \geq_{\forall \boldsymbol{\mu} \in \Pi_{[M]}} U_a(\boldsymbol{\rho}^{\mathrm{NE}}, \boldsymbol{\mu}). \tag{9}$$

Definition 5. *The Nash Delivery Plan (NDP), denoted by* $\boldsymbol{\rho}^{\mathrm{NE}}$, *is the probability distribution over the different routes, as determined by the NE of the MDG.*

For instance, a NDP $(0.7, 0.3)$ dictates that 70% of the Replys will be sent over r_1, and 30% over r_2. Note that this distribution does not determine which Reply is sent over which route, as this decision is probabilistic.

4.2. Maximin strategies

We say that the Defender maximinimizes if she chooses an RDP that is best for her on the assumption that whatever she does, the Attacker will choose an MP to cause the highest possible damage to her.

Definition 6. *A Randomised Delivery Plan* $\boldsymbol{\rho}^{\dagger} \in \Pi_{[R]}$ *is a maximin strategy of the Defender, if and only if*

$$\min_{\boldsymbol{\mu} \in \Pi_{[M]}} U_d(\boldsymbol{\rho}^{\dagger}, \boldsymbol{\mu}) \geq \min_{\boldsymbol{\mu} \in \Pi_{[M]}} U_d(\boldsymbol{\rho}, \boldsymbol{\mu}), \forall \boldsymbol{\rho} \in \Pi_{[R]}. \tag{10}$$

Table 2: A toy game example

	m	m'
r	-3,1	-1,0
r'	-4,0	-2,1

A maximinimiser for the Defender is an RDP that maximises the payoff that the Defender can *guarantee*. In other words, $\boldsymbol{\rho}^\dagger$ guarantees (i.e. "secures") the Defender at least her maximin payoff regardless of $\boldsymbol{\mu}$, as $\boldsymbol{\rho}^\dagger$ solves the problem $\max_{\boldsymbol{\rho}} \min_{\boldsymbol{\mu}} U_d(\boldsymbol{\rho}, \boldsymbol{\mu})$. That is why $\boldsymbol{\rho}^\dagger$ is also called *security strategy*.

Definition 7. *A Malware Plan* $\boldsymbol{\mu}^\dagger \in \Pi_{[M]}$ *is a maximin strategy of the Attacker, if and only if*

$$\min_{\boldsymbol{\rho} \in \Pi_{[R]}} U_a(\boldsymbol{\rho}, \boldsymbol{\mu}^\dagger) \geq \min_{\boldsymbol{\rho} \in \Pi_{[R]}} U_a(\boldsymbol{\rho}, \boldsymbol{\mu}), \forall \boldsymbol{\mu} \in \Pi_{[M]}. \tag{11}$$

4.3. Stackelberg mixed strategies

A *two-player Stackelberg game* involves one player (leader) to commit to a strategy before the other player (follower) moves. In a Stackelberg model the *commitment of the leader is absolute*, that is the leader cannot back-track on her commitment. On the other hand, the follower sees the strategy that the leader committed to, before she chooses a strategy.

In an Stackelberg MDG, the Attacker *conducts surveillance* before she attacks and therefore she is aware of the Defender's RDP. For completeness, we consider that this best-response is expressed also in mixed strategies.

In general, Stackelberg and Nash games *do not have the same equilibria*. For instance, let us consider the normal-form MDG in Table 2, where the Defender has only two routes (r, r') available and the Attacker can choose between two malware types (m, m'). We see that if this is a Nash game, r is a strictly dominant strategy for the Defender, as it gives her a higher payoff value than r'. As we have assumed that this is a complete information game, the Attacker knows that r is preferable for the Defender and she chooses m, which rewards her with 1 as opposed to m', which gives payoff value 0. Therefore the NE of the game (in pure strategies) is (r, m).

If we now consider this game as Stackelberg, the Defender (leader) can commit to a strategy before the Attacker (follower) chooses her strategy. If the Defender commits to r then the Attacker will play m, but if the Defender commits to r' then the Attacker will choose m'. The second pure strategy profile, i.e. (r', m') gives higher payoff to the Defender (-2 as opposed to (r, m), which gives -3) and therefore the Defender is better-off in the Stackelberg game compared to the Nash game, where her payoff equals -3 < -2.

Definition 8. *A Reply Delivery Plan (RDP) is optimal if it maximises the Defender's payoff given that the Attacker will always play a best-response strategy with tie-breaking in favour of the Defender.*

Definition 9. *A Malware Plan is a best response if it maximises the Attacker's payoff, taking the Defender's Reply Delivery Plan as given.*

A commonly used notion of a solution in Stackelberg games is the Strong Stackelberg Equilibrium (SSE), defined in MDG as follows.

Definition 10. *At the Strong Stackelberg Equilibrium of the MDG:*

1. *for any* $\boldsymbol{\rho} \in \Delta_{[R]}$*, the Attacker plays a best-response* $\boldsymbol{\mu}^{\mathrm{BR}}(\boldsymbol{\rho}) \in \Delta_{[M]}$ *that is,*
$$U_a(\boldsymbol{\rho},\boldsymbol{\mu}^{\mathrm{BR}}(\boldsymbol{\rho})) \geq U_a(\boldsymbol{\rho},\boldsymbol{\mu}(\boldsymbol{\rho})),\ \forall \boldsymbol{\mu}(\boldsymbol{\rho}) \neq \boldsymbol{\mu}^{\mathrm{BR}}(\boldsymbol{\rho}); \tag{12}$$
2. *for any* $\boldsymbol{\rho} \in \Delta_{[R]}$*, the Attacker breaks ties in favour of the Defender, that is, when there are multiple best responses to* $\boldsymbol{\rho}$*, the Attacker plays the best response* $\boldsymbol{\mu}^{\mathrm{SSE}}(\boldsymbol{\rho}) \in \Delta_{[M]}$ *that maximises the Defender's payoff:*
$$\begin{aligned} U_d(\boldsymbol{\rho},\boldsymbol{\mu}^{\mathrm{SSE}}(\boldsymbol{\rho})) \geq U_d(\boldsymbol{\rho},\boldsymbol{\mu}^{\mathrm{BR}}(\boldsymbol{\rho})), \\ \forall \boldsymbol{\mu}^{\mathrm{BR}} \text{ best response to } \boldsymbol{\rho}; \end{aligned} \tag{13}$$
3. *the Defender plays a best-response* $\boldsymbol{\rho}^{\mathrm{SSE}} \in \Delta_{[R]}$*, which maximises her payoff given that the Attacker's strategies are given by the first two conditions (i.e. the Attacker always plays best response with tie-breaking in favour of the Defender [38],[45]):*
$$U_d(\boldsymbol{\rho}^{\mathrm{SSE}},\boldsymbol{\mu}^{\mathrm{SSE}}(\boldsymbol{\rho}^{\mathrm{SSE}})) \geq U_d(\boldsymbol{\rho},\ \boldsymbol{\mu}^{\mathrm{SSE}}(\boldsymbol{\rho})),\ \forall\ \boldsymbol{\rho} \neq \boldsymbol{\rho}^{\mathrm{SSE}}. \tag{14}$$

5. Optimality analysis

For the purpose of analysis, we consider *complete information* Nash MDGs, according to which both players know the game matrix, which contains the utilities of both players for each pure strategy profile. The utility function of the Defender is determined by the probability of failing to detect a route and the overall performance cost, which is imposed on the devices of the j-th route when undertaking malware detection. We denote by $c(s_i)$ the performance cost imposed on each $s_i \in \mathcal{S}_j$ and therefore the overall performance cost over a route r_j equals $\sum_{s_i \in \mathcal{S}_j} c(s_i)$.

We consider two different MDGs; (i) a *zero sum* MDG, where the Attacker's utility is the opposite of the Defender's utility and (ii) a *non-zero sum* MDG, where the Attacker's utility is a strictly positive scaling of the Defender's utility.

The rationale behind the zero sum game is that when there are clear winners (e.g. the Attacker) and losers (e.g. the Defender), and the Defender is uncertain about the Attacker type, she considers the *worst case scenario*, which can be formulated by a zero sum game where the Attacker can cause her *maximum damage*. While in most security situations the interests of the players are neither in strong conflict nor in complete identity, the zero sum game provides important insights into the notion of "optimal play", which is closely related to the *minimax theorem* [46].

In the zero sum MDG, $\Gamma_0 = \langle \{d, a\}, [R] \times [M], \{U_d, -U_d\} \rangle$ (for clarity d has been used for the Defender and a for the Attacker), the Attacker's gain is equal to the Defender's security loss, and vice versa. We define the utility of the Defender in Γ_0 as

$$U_d^{\Gamma_0}(r_j, m_l) := -w_H \, p(r_j, m_l) \, H(m_l) - w_C \sum_{s_i \in \mathcal{S}_j} c(s_i). \tag{15}$$

The first term of (15) is the expected security loss of the Defender inflicted by the Attacker when attempting to infect `Rqs` with m_l, while the second term expresses the aggregated message inspection cost imposed on all devices of r_j, irrespective of the attacking strategy. Note that $w_H, w_C \in [0, 1]$ are importance weights, which can facilitate the Defender with setting her preferences in terms of security loss, and computational detection cost, accordingly.

By setting $S(r_j, m_l) = w_H \, p(r_j, m_l) \, H(m_l)$, and $C(r_j) = w_C \sum_{s_i \in \mathcal{S}_j} c(s_i)$, we have that

$$U_d^{\Gamma_0}(r_j, m_l) := -S(r_j, m_l) - C(r_j). \tag{16}$$

For a mixed profile $(\boldsymbol{\rho}, \boldsymbol{\mu})$, the utility of the Defender equals

$$\begin{aligned} U_d^{\Gamma_0}(\boldsymbol{\rho}, \boldsymbol{\mu}) &\overset{(6)}{=} \sum_{r_j \in [R]} \sum_{m_l \in [M]} U_d^{\Gamma_0}(r_j, m_l) \boldsymbol{\rho}(r_j) \, \boldsymbol{\mu}(m_l) \\ &\overset{(16)}{=} \sum_{r_j \in [R]} \sum_{m_l \in [M]} [-S(r_j, m_l) - C(r_j)] \, \boldsymbol{\rho}(r_j) \, \boldsymbol{\mu}(m_l) \\ &= - \sum_{r_j \in [R]} \sum_{m_l \in [M]} S(r_j, m_l) \, \boldsymbol{\rho}(r_j) \, \boldsymbol{\mu}(m_l) \\ &\quad - \sum_{r_j \in [R]} C(r_j) \, \boldsymbol{\rho}(r_j). \end{aligned} \tag{17}$$

As Γ_0 is a zero sum game, the Attacker's utility is given by $U_a^{\Gamma_0}(\boldsymbol{\rho}, \boldsymbol{\mu}) = -U_d^{\Gamma_0}(\boldsymbol{\rho}, \boldsymbol{\mu})$. Since the Defender's equilibrium strategies maximise her utility, given that the Attacker maximises her own utility, we will refer to them as *optimal strategies*.

As Γ_0 is a two-person zero sum game with finite number of actions for both

players, according to Nash [47], it admits at least a NE in mixed strategies, and saddle-points correspond to Nash equilibria as discussed in [15] (p. 42). The following result from [48], establishes the existence of a saddle (equilibrium) solution in the games we examine and summarizes their properties.

Definition 11 (Saddle point of the MDG)**.** *The Γ_0 Malware Detection Game (MDG) admits a saddle point in mixed strategies, $(\boldsymbol{\rho}^{\mathrm{NE}}_{\Gamma_0}, \boldsymbol{\mu}^{\mathrm{NE}}_{\Gamma_0})$, with the property that*

- $\boldsymbol{\rho}^{\mathrm{NE}}_{\Gamma_0} = \arg\max_{\boldsymbol{\rho}\in\Delta_{[R]}} \min_{\boldsymbol{\mu}\in\Delta_{[M]}} U_d^{\Gamma_0}(\boldsymbol{\rho}, \boldsymbol{\mu})$, $\forall \boldsymbol{\mu}$, *and*
- $\boldsymbol{\mu}^{\mathrm{NE}}_{\Gamma_0} = \arg\max_{\boldsymbol{\mu}\in\Delta_{[M]}} \min_{\boldsymbol{\rho}\in\Delta_{[R]}} U_a^{\Gamma_0}(\boldsymbol{\rho}, \boldsymbol{\mu})$, $\forall \boldsymbol{\rho}$.

Then, due to the zero sum nature of the game, the minimax theorem [46] holds, i.e. $\max_{\boldsymbol{\rho}\in\Delta_{[R]}} \min_{\boldsymbol{\mu}\in\Delta_{[M]}} U_d^{\Gamma_0}(\boldsymbol{\rho}, \boldsymbol{\mu}) = \min_{\boldsymbol{\mu}\in\Delta_{[M]}} \max_{\boldsymbol{\rho}\in\Delta_{[R]}} U_d^{\Gamma_0}(\boldsymbol{\rho}, \boldsymbol{\mu})$.

The pair of saddle point strategies $(\boldsymbol{\rho}^{\mathrm{NE}}_{\Gamma_0}, \boldsymbol{\mu}^{\mathrm{NE}}_{\Gamma_0})$ are at the same time security strategies for the players, i.e. they ensure a minimum performance regardless of the actions of the other. Furthermore, if the game admits multiple saddle points (and strategies), they have the ordered interchangeability property, i.e. the player achieves the same performance level independent from the other player's choice of saddle point strategy.

The minimax theorem [46] states that for zero sum games, NE and minimax solutions coincide. Therefore, $\boldsymbol{\rho}^{\mathrm{NE}}_{\Gamma_0} = \arg\min_{\boldsymbol{\rho}\in\Delta_{[R]}} \max_{\boldsymbol{\mu}\in\Delta_{[M]}} U_a^{\Gamma_0}(\boldsymbol{\rho}, \boldsymbol{\mu})$. This means that regardless of the strategy the Attacker chooses, the Nash Delivery Plan (NDP) is the Defender's security strategy that guarantees a minimum performance.

We can convert Γ_0 into a Linear Programming (LP) problem and make use of some of the powerful algorithms available for LP to derive the equilibrium. For a given mixed strategy $\boldsymbol{\rho}$ of the Defender, we assume that the Attacker can cause maximum damage to `Rqs` by injecting a message $\widehat{m}$ into the cluster network.

Formally, the Defender seeks to solve the following LP:

$$\max_{\boldsymbol{\rho}\in\Delta_{[R]}} \min_{\boldsymbol{\mu}\in\Delta_{[M]}} U_d^{\Gamma_0}(\boldsymbol{\rho}, \widehat{m}\,)$$
$$\text{subject to} \begin{cases} U_d^{\Gamma_0}(\boldsymbol{\rho}, m_1) - \min_{\boldsymbol{\mu}\in\Delta_{[M]}} U_d^{\Gamma_0}(\boldsymbol{\rho}, \widehat{m})e \geq 0 \\ \qquad\qquad \vdots \\ U_d^{\Gamma_0}(\boldsymbol{\rho}, m_M) - \min_{\boldsymbol{\mu}\in\Delta_{[M]}} U_d^{\Gamma_0}(\boldsymbol{\rho}, \widehat{m})e \geq 0 \\ \boldsymbol{\rho}e = 1 \\ \boldsymbol{\rho} \geq 0. \end{cases} \tag{18}$$

In this problem, e is a vector of ones of size M.

Lemma 1. *A mixed strategy profile $(\boldsymbol{\rho}^{\mathrm{NE}}, \boldsymbol{\mu}^{\mathrm{NE}}) \in \Pi_{[R]} \times \Pi_{[M]}$ in Γ_0, is a mixed strategy NE if and only if*

1. *every route* $r_j \in supp(\boldsymbol{\rho}^{\mathrm{NE}})$ *selection is a best response to* $\boldsymbol{\mu}^{\mathrm{NE}}$ *and,*
2. *every malware* $m_l \in supp(\boldsymbol{\mu}^{\mathrm{NE}})$ *selection is a best response to* $\boldsymbol{\rho}^{\mathrm{NE}}$*.*

Proof. First, notice that U_d, as defined in (15), is a linear function in $\boldsymbol{\rho}(r_j)$ that is, for any two RDPs $\boldsymbol{\rho}_1$ and $\boldsymbol{\rho}_2$ and any number $\theta \in [0,1]$ we have $U_d(\theta\,\boldsymbol{\rho}_1 + (1-\theta)\,\boldsymbol{\mu}) = \theta\, U_d(\boldsymbol{\rho}_1) + (1-\theta)\, U_d(\boldsymbol{\rho}_2)$. Then, for the sake of contradiction, assume there exists a route $r'_j \in supp(\boldsymbol{\rho}^{\mathrm{NE}})$ selection that is not a best response to $\boldsymbol{\mu}^{\mathrm{NE}}$. Due to the linearity of U_d in $\boldsymbol{\rho}^{\mathrm{NE}}(r_j)$, the Defender can increase her payoff by transferring probability from $\boldsymbol{\rho}(r'_j)$ to a route selection that is a best response to $\boldsymbol{\mu}^{\mathrm{NE}}$, creating a new mixed strategy $\boldsymbol{\rho}^* \succsim \boldsymbol{\rho}^{\mathrm{NE}}$. However, this contradicts the assumption that $\boldsymbol{\rho}^{\mathrm{NE}}$ is the strategy of the Defender at the NE, as the Defender prefers to deviate from $\boldsymbol{\rho}^{\mathrm{NE}}$ to gain a higher payoff, by playing $\boldsymbol{\rho}^*$. The second part of the lemma can be proven in the same way. □

Let us now assume a non-zero sum MDG, denoted by Γ, with the same strategy spaces with Γ_0, in which the Defender's utility is the same as in Γ_0, i.e. $U_d^{\Gamma}(\boldsymbol{\rho},\boldsymbol{\mu}) = U_d^{\Gamma_0}(\boldsymbol{\rho},\boldsymbol{\mu}) = -S(r_j,m_l) - C(r_j)$. On the other hand, the Attacker's utility is (strictly positive) scaling of the security loss $S(r_j,m_l)$ of the Defender upon a successful attack. This is to say that the performance cost of the Defender is only important to her as the Attacker is only after compromising `Rqs`. Therefore, given a pure strategy profile (r_j,m_l), the utility of the Attacker, in Γ, is defined as:

$$U_a^{\Gamma}(r_j,m_l) := \Xi\, S(r_j,m_l), \text{ for } \Xi > 0. \tag{19}$$

For a mixed profile $(\boldsymbol{\rho},\boldsymbol{\mu})$ the utility of the Attacker is given by

$$\begin{aligned} U_a^{\Gamma}(\boldsymbol{\rho},\boldsymbol{\mu}) &\overset{(7)}{=} \sum_{r_j\in[R]} \sum_{m_l\in[M]} U_a^{\Gamma}(r_j,m_l)\,\boldsymbol{\rho}(r_j)\,\boldsymbol{\mu}(m_l) \\ &\overset{(19)}{=} \sum_{r_j\in[R]} \sum_{m_l\in[M]} \Xi\, S(r_j,m_l)\,\boldsymbol{\rho}(r_j)\,\boldsymbol{\mu}(m_l). \end{aligned} \tag{20}$$

Hence, due to $U_d^{\Gamma}(\boldsymbol{\rho},\boldsymbol{\mu}) = U_d^{\Gamma_0}(\boldsymbol{\rho},\boldsymbol{\mu})$, from (17) and (20) we have that

$$\begin{aligned} U_d^{\Gamma}(\boldsymbol{\rho},\boldsymbol{\mu}) &= -\frac{1}{\Xi} U_a^{\Gamma}(\boldsymbol{\rho},\boldsymbol{\mu}) - \sum_{r_j\in[R]} C(r_j)\,\boldsymbol{\rho}(r_j) \\ &= -\frac{1}{\Xi} U_a^{\Gamma}(\boldsymbol{\rho},\boldsymbol{\mu}) - k(\boldsymbol{\rho}), \end{aligned} \tag{21}$$

where $\frac{1}{\Xi} > 0$, and $k(\boldsymbol{\rho})$ is an expression that does not depend on $\boldsymbol{\mu}$. That is, the best response of the Defender to any given malware plan, also yields the utility for the Defender at the worst case scenario.

Lemma 2. *NE strategies of the Defender in* Γ *are equivalent of the NE strategies of the Defender in* Γ_0*. Formally,* $\Omega_{\Gamma}^{\mathrm{NE}} = \Omega_{\Gamma_0}^{\mathrm{NE}}$.

Proof. By definition, a strategy profile $(\boldsymbol{\rho}^{\mathrm{NE}}, \boldsymbol{\mu}^{\mathrm{NE}})$ is NE of Γ if and only if:

$$S(\boldsymbol{\rho}^{\mathrm{NE}}, \boldsymbol{\mu}^{\mathrm{NE}}) + k(\boldsymbol{\rho}^{\mathrm{NE}}) \leq S(\boldsymbol{\rho}, \boldsymbol{\mu}^{\mathrm{NE}}) + k(\boldsymbol{\rho}), \forall \boldsymbol{\rho} \in \Delta_{[R]}, \tag{22a}$$

$$\Xi \cdot S(\boldsymbol{\rho}^{\mathrm{NE}}, \boldsymbol{\mu}^{\mathrm{NE}}) \geq \Xi \cdot S(\boldsymbol{\rho}^{\mathrm{NE}}, \boldsymbol{\mu}), \forall \boldsymbol{\mu} \in \Delta_{[M]}. \tag{22b}$$

Here is the observation:

$$\begin{aligned} \Xi \cdot S(\boldsymbol{\rho}^{\mathrm{NE}}, \boldsymbol{\mu}^{\mathrm{NE}}) \geq \Xi \cdot S(\boldsymbol{\rho}^{\mathrm{NE}}, \boldsymbol{\mu}), \forall \boldsymbol{\mu} \in \Delta_{[M]} \iff \\ \Xi \cdot [S(\boldsymbol{\rho}^{\mathrm{NE}}, \boldsymbol{\mu}^{\mathrm{NE}}) + k(\boldsymbol{\rho}^{\mathrm{NE}})] \geq \\ \Xi \cdot [S(\boldsymbol{\rho}^{\mathrm{NE}}, \boldsymbol{\mu}) + k(\boldsymbol{\rho}^{\mathrm{NE}})], \forall \boldsymbol{\mu} \in \Delta_{[M]}. \end{aligned} \tag{23}$$

Since $\Xi > 0$, the latter condition is satisfied if and only if:

$$S(\boldsymbol{\rho}^{\mathrm{NE}}, \boldsymbol{\mu}^{\mathrm{NE}}) + k(\boldsymbol{\rho}^{\mathrm{NE}}) \geq S(\boldsymbol{\rho}^{\mathrm{NE}}, \boldsymbol{\mu}) + k(\boldsymbol{\rho}^{\mathrm{NE}}), \forall \boldsymbol{\mu} \in \Delta_{[M]}. \tag{24}$$

In short, $(\boldsymbol{\rho}^{\mathrm{NE}}, \boldsymbol{\mu}^{\mathrm{NE}})$ is a NE of Γ, if and only if it satisfies:

$$S(\boldsymbol{\rho}^{\mathrm{NE}}, \boldsymbol{\mu}^{\mathrm{NE}}) + k(\boldsymbol{\rho}^{\mathrm{NE}}) \leq S(\boldsymbol{\rho}, \boldsymbol{\mu}^{\mathrm{NE}}) + k(\boldsymbol{\rho}), \forall \boldsymbol{\rho} \in \Delta_{[R]}, \tag{25a}$$

$$S(\boldsymbol{\rho}^{\mathrm{NE}}, \boldsymbol{\mu}^{\mathrm{NE}}) + k(\boldsymbol{\rho}^{\mathrm{NE}}) \geq S(\boldsymbol{\rho}^{\mathrm{NE}}, \boldsymbol{\mu}) + k(\boldsymbol{\rho}^{\mathrm{NE}}), \forall \boldsymbol{\mu} \in \Delta_{[M]}. \tag{25b}$$

But these are exactly the conditions describing a NE of Γ_0. Therefore $\Omega_{\Gamma}^{\mathrm{NE}} = \Omega_{\Gamma_0}^{\mathrm{NE}}$. □

Lemma 3. *In* Γ*, the set of NE and Maximin strategies of the Defender are equivalent, i.e.* $\Omega_{\Gamma}^{\mathrm{NE}} = \Omega_{\Gamma}^{\mathrm{maximin}}$.

Proof. ($\Rightarrow$) Since Γ_0 is a two person zero-sum game, we know that the set of NE and Maximin strategies of the Defender are the same, i.e. $\Omega_{\Gamma_0}^{\mathrm{NE}} = \Omega_{\Gamma_0}^{\mathrm{maximin}}$. Let $(\boldsymbol{\rho}^{\mathrm{NE}}, \boldsymbol{\mu}^{\mathrm{NE}}) \in \Omega_{\Gamma}^{\mathrm{NE}}$ then based on Lemma 2 we have that $(\boldsymbol{\rho}^{\mathrm{NE}}, \boldsymbol{\mu}^{\mathrm{NE}}) \in \Omega_{\Gamma_0}^{\mathrm{NE}}$. Since Γ_0 is zero-sum, $\boldsymbol{\rho}^{\mathrm{NE}} \in \Omega_{\Gamma_0}^{\mathrm{maximin}}$. But the strategy spaces and the utility of the Defender are the same in both Γ and Γ_0. Hence the conditions for a mixed strategy to be a Defender's Maximin is the same in both games. Therefore, $\boldsymbol{\rho}^{\mathrm{NE}} \in \Omega_{\Gamma}^{\mathrm{maximin}}$, i.e. $\Omega_{\Gamma}^{\mathrm{NE}} \subseteq \Omega_{\Gamma}^{\mathrm{maximin}}$.
($\Leftarrow$) The argument goes in the other direction as well: consider $\boldsymbol{\rho}^{\mathrm{NE}} \in \Omega_{\Gamma}^{\mathrm{maximin}}$. Since the utility of the Defender and the strategy spaces are the same across the two games, for the same strategy $\boldsymbol{\rho}^{\mathrm{NE}}$, we have that $\boldsymbol{\rho}^{\mathrm{NE}} \in \Omega_{\Gamma_0}^{\mathrm{maximin}}$. Since Γ_0 is two-player zero-sum, there exists $\boldsymbol{\mu}^{\mathrm{NE}}$ such that $(\boldsymbol{\rho}^{\mathrm{NE}}, \boldsymbol{\mu}^{\mathrm{NE}}) \in \Omega_{\Gamma_0}^{\mathrm{NE}}$. From Lemma 2, this means $(\boldsymbol{\rho}^{\mathrm{NE}}, \boldsymbol{\mu}^{\mathrm{NE}})_{\Gamma} \in \Omega^{\mathrm{NE}}$. Hence, *Maximin strategies of the Defender are also part of her NE strategies in* Γ, i.e. $\Omega_{\Gamma}^{\mathrm{maximin}} \subseteq \Omega_{\Gamma}^{\mathrm{NE}}$. Putting the two together $\Omega_{\Gamma}^{\mathrm{NE}} = \Omega_{\Gamma}^{\mathrm{maximin}}$. □

This lemma establishes that the Defender can randomise according to her NE and, in expectation, be guaranteed at least the expected utility prescribed by the NE, irrespective of the mixed strategy of the Attacker. To put it differently, the Defender can play her pessimistic maximin strategy, but she does not lose anything in expectation by not playing a NE strategy. It is worth stressing that this property only holds for the NE strategy of the Defender and not of the Attacker.

Lemma 4. *In Γ, the set of Maximin and SSE strategies of the Defender are the same, i.e.* $\Omega_\Gamma^{\text{maximin}} = \Omega_\Gamma^{\text{SSE}}$.

Proof. ($\Rightarrow$) Let $\boldsymbol{\rho}^{\text{NE}} \in \Omega_\Gamma^{\text{SSE}}$ be a SSE strategy of the Defender. Then by definition, $\boldsymbol{\rho}^{\text{NE}}$ is (i) an optimal strategy of the Defender given that (ii) the Attacker is best-responding to it but by (iii) breaking ties in favour of the Defender. That is:

(i) $\boldsymbol{\rho}^{\text{NE}} \in \arg\max_{\boldsymbol{\rho}\in\Delta_{[R]}} U_d(\boldsymbol{\rho}, \boldsymbol{\mu}^{\text{BR}}(\boldsymbol{\rho}))$ where;

(ii) for any $\boldsymbol{\rho} \in \Delta_{[R]}$, $\boldsymbol{\mu}^{\text{BR}}(\boldsymbol{\rho}) \in \arg\max_{\boldsymbol{\mu}\in\Delta_{[M]}} U_a(\boldsymbol{\rho}, \boldsymbol{\mu})$ and;

(iii) for any $\boldsymbol{\rho} \in \Delta_{[R]}$:

$$\boldsymbol{\mu}^{\text{BR}}(\boldsymbol{\rho}) \in \arg \max_{\boldsymbol{\mu}\in\arg\max_{\boldsymbol{\mu}\in\Delta_{[M]}} U_a(\boldsymbol{\rho},\boldsymbol{\mu})} U_d(\boldsymbol{\rho}, \boldsymbol{\mu}). \tag{26}$$

Let us examine condition (ii): for any $\boldsymbol{\rho} \in \Delta_{[R]}$:

$$\begin{aligned} \boldsymbol{\mu}^{\text{BR}}(\boldsymbol{\rho}) \in \arg \max_{\boldsymbol{\mu}\in\Delta_{[M]}} \Xi \cdot S(\boldsymbol{\rho}, \boldsymbol{\mu}) \iff \\ \boldsymbol{\mu}^{\text{BR}}(\boldsymbol{\rho}) \in \arg \max_{\boldsymbol{\mu}\in\Delta_{[M]}} \Xi \cdot [S(\boldsymbol{\rho}, \boldsymbol{\mu}) + k(\boldsymbol{\rho})] \\ \boldsymbol{\mu}^{\text{BR}}(\boldsymbol{\rho}) \in \arg \max_{\boldsymbol{\mu}\in\Delta_{[M]}} S(\boldsymbol{\rho}, \boldsymbol{\mu}) + k(\boldsymbol{\rho}). \end{aligned} \tag{27}$$

In short, condition (ii) is equivalent to:

$$\text{(iv) For any } \boldsymbol{\rho} \in \Delta_{[R]}, \boldsymbol{\mu}^{\text{BR}}(\boldsymbol{\rho}) \in \arg \min_{\boldsymbol{\mu}\in\Delta_{[M]}} U_d(\boldsymbol{\rho}, \boldsymbol{\mu}).$$

This makes condition (iii) irrelevant. But conditions (i) and (iv) exactly describe a Maximin strategy of the Defender. Therefore we have proved that $\Omega_\Gamma^{\text{SSE}} \subseteq \Omega_\Gamma^{\text{maximin}}$. ($\Leftarrow$) The argument can be established identically in reverse direction, starting from a Maximin strategy of the Defender. So given conditions (i) and (iv) we must prove that conditions (ii) and (iii) are true. Let $\boldsymbol{\rho}^{\text{NE}} \in \Omega_\Gamma^{\text{maximin}}$ be a Maximin strategy of the Defender. Then by definition, $\boldsymbol{\rho}^{\text{NE}}$ is (i) an optimal strategy of the Defender given that (iv) the Attacker is minimising Defender's utility. We see that condition (ii) is true if and only if condition (iv) is true. Since the Maximin strategy $\boldsymbol{\rho}^{\text{NE}}$ makes condition (iv) true, it will also make condition

(ii). To prove that $\boldsymbol{\rho}^{\mathrm{NE}}$ is an SSE, we also need to prove condition (iii). Let us assume that the condition is not true. This means that there is a best-response of the Attacker that does not break ties in favour of the Defender. Formally,

$$\begin{aligned}
\boldsymbol{\mu}^{\mathrm{BR}}(\boldsymbol{\rho}) \notin \arg \max_{\boldsymbol{\mu} \in \mathrm{argmax}_{\boldsymbol{\mu}} U_a(\boldsymbol{\rho},\boldsymbol{\mu})} U_d(\boldsymbol{\rho},\boldsymbol{\mu}) \Longleftrightarrow \\
\boldsymbol{\mu}^{\mathrm{BR}}(\boldsymbol{\rho}) \notin \arg \max_{\boldsymbol{\mu} \in \mathrm{argmax}_{\boldsymbol{\mu}} U_a(\boldsymbol{\rho},\boldsymbol{\mu})} \{-S(\boldsymbol{\rho},\boldsymbol{\mu}) - k(\boldsymbol{\rho})\} \Longleftrightarrow \\
\boldsymbol{\mu}^{\mathrm{BR}}(\boldsymbol{\rho}) \notin \arg \min_{\boldsymbol{\mu} \in \mathrm{argmax}_{\boldsymbol{\mu}} U_a(\boldsymbol{\rho},\boldsymbol{\mu})} \{S(\boldsymbol{\rho},\boldsymbol{\mu}) + k(\boldsymbol{\rho})\} \Longleftrightarrow \\
\boldsymbol{\mu}^{\mathrm{BR}}(\boldsymbol{\rho}) \notin \arg \min_{\boldsymbol{\mu} \in \mathrm{argmax}_{\boldsymbol{\mu}} U_a(\boldsymbol{\rho},\boldsymbol{\mu})} S(\boldsymbol{\rho},\boldsymbol{\mu}) \Longleftrightarrow \\
\boldsymbol{\mu}^{\mathrm{BR}}(\boldsymbol{\rho}) \notin \arg \min_{\boldsymbol{\mu} \in \mathrm{argmax}_{\boldsymbol{\mu}} U_a(\boldsymbol{\rho},\boldsymbol{\mu})} U_a(\boldsymbol{\rho},\boldsymbol{\mu}),
\end{aligned} \tag{28}$$

which is leads to a contradiction. Therefore condition (3) holds, and putting together all three conditions (1), (2), and (3), we have that $\boldsymbol{\rho}^{\mathrm{NE}}$, which is a Maximin strategy of the Defender it is also an SSE strategy, i.e. $\Omega_{\Gamma}^{\mathrm{maximin}} \subseteq \Omega_{\Gamma}^{\mathrm{SSE}}$. Putting the two proofs together we have that $\Omega_{\Gamma}^{\mathrm{maximin}} = \Omega_{\Gamma}^{\mathrm{SSE}}$. □

Theorem 1. *In Γ, the set of NE, Maximin and SSE strategies of the Defender are the same, i.e. $\Omega_{\Gamma}^{\mathrm{NE}} = \Omega_{\Gamma}^{\mathrm{maximin}} = \Omega_{\Gamma}^{\mathrm{SSE}}$. Besides, all NE are interchangeable, in Γ, and all yield the same utility for the defender.*

Proof. Trivially, from Lemmas 3 and 4 we have that $\Omega_{\Gamma}^{\mathrm{NE}} = \Omega_{\Gamma}^{\mathrm{maximin}} = \Omega_{\Gamma}^{\mathrm{SSE}}$. Since Γ_0 is a two person zero-sum game, we know that all NE are interchangeable [48]. From Lemma 2 the NE of Γ_0 are the NE of Γ and vice-versa. We also see that the utility of the Defender is the same across Γ and Γ_0. Therefore the utility of the Defender in all NE of our original game is the same, which also implies that all NE of our original game are interchangeable. □

The above lemma establishes that the Defender, regardless of whether the Attacker conducts surveillance, she plays optimally when she randomises according to her NE strategy.

Theorem 2. *Regardless of the type of malware detection game played, i.e.*

1. *a zero sum or a non-zero sum malware detection game,*
2. *a Nash or a Stackelberg malware detection game,*

the Defender plays optimally by choosing any strategy $\boldsymbol{\rho} \in \Omega_{\Gamma_0}^{\mathrm{NE}}$.

Proof. By combining 2 and 1, we have that $\Omega_{\Gamma_0}^{\mathrm{NE}} = \Omega_{\Gamma}^{\mathrm{NE}} = \Omega_{\Gamma}^{\mathrm{maximin}} = \Omega_{\Gamma}^{\mathrm{SSE}}$, which proves the theorem. □

The above theorem demonstrates that it is computationally efficient for the Defender to derive her optimal strategy by solving the LP represented by (18). It is worth noting that a similar result but for different problem has been published in [37].

6. *i*Routing

In this section, we present the *i*Routing protocol, which stands for *intelligent Routing* and whose routing decisions are made according to the *Nash Delivery Plan* (NDP). *i*Routing has been designed based on the mathematical findings of the MDG analysis, presented in previous sections, and its main goal is to maximise the utility of the Defender in the presence of a "rational" Attacker.

Within the realm of Mobile Edge Computing (MEC), devices of the cluster request services from the cluster-head (denoted by `C`) imposing the need for establishing an end-to-end path between the requestor (i.e. destination device denoted by `Rqs`) and `C`. Each time data must be delivered to `Rqs`, `C` has to compute the NDP by solving an MDG for this destination. To do this, following the route discovery, `C` uses its latest information about the malware detection capabilities of all possible routes to `Rqs`, along with their inspection costs (i.e. malware detection costs to perform, for example, intrusion classification). Data is then relayed and collaboratively inspected by the devices on its way to `Rqs`. Overall, the objective of `C` (i.e. the Defender) is to select the route that can correctly detect and filter out malicious data before they infect `Rqs` by making sure that it is not crafted with malware. We assume that each device must use its data inspection capabilities at the maximum possible degree..

*i*Routing has characteristics of *reactive route selection protocols*, meaning that it takes action and starts computing routing paths that have not been previously computed when a request for data delivery to `Rqs` is issued. *i*Routing requires to obtain information about the malware inspection capabilities and the associated computational cost of devices, in routes from `C` to `Rqs`.

*i*Routing consists of *three main phases*, which we describe in more detail in the remainder of this section. In the first phase of the protocol (described in Algorithm 1), `C` *broadcasts* a Route REQuest ($\texttt{RREQ}_{\texttt{Rqs}}$) to discover routes towards `Rqs`. Each device that receives the $\texttt{RREQ}_{\texttt{Rqs}}$), acts similarly by broadcasting it towards `Rqs`. After `C` sends a $\texttt{RREQ}_{\texttt{Rqs}}$, it has to await for some timeout T_{req}, which is set equal to the Net Traversal Time (NetTT), as in AODV [7].

The second phase of the protocol starts when the receiving device is `Rqs`. Then, this device does not forward the request any further. Instead, it prepares a Route REPly ($\texttt{RREP}_{\texttt{Rqs}}$), and sends it back towards `C` by using the reverse route, which is built during the delivery of $\texttt{RREQ}_{\texttt{Rqs}}$, as described by Algorithm 2. Each $\texttt{RREP}_{\texttt{Rqs}}$ carries information about: (i) the set $\mathcal{S}_j$ of devices that comprise a route; (ii)

Algorithm 1 Seeking routes to destination $\texttt{Rqs}$.

procedure iROUTING_REQUEST$(s, \texttt{Rqs}, \mathcal{S}_\texttt{j})$
  s seeks routes to $\texttt{Rqs}$ by broadcasting $\texttt{RREQ}_\texttt{Rqs}$;
  if a device s_i receives $\texttt{RREQ}_\texttt{Rqs}$ **then**
    $\mathcal{S}_j \cup \{s_i\}$;
    if $s_i \neq \texttt{Rqs}$ **then**
      s_i executes iROUTING_REQUEST$(s_i, \texttt{Rqs}, \mathcal{S}_j)$;
    else
      $L \leftarrow |\mathcal{S}_j|$, $n \leftarrow 0, \mathcal{T}_j \leftarrow \emptyset, \mathcal{C}_j \leftarrow \emptyset$;
      iROUTING_RESPONSE$(n, L, \mathcal{T}_j, \mathcal{C}_j, \mathcal{S}_j, \texttt{Rqs})$;
      break;
    end if
  end if
end procedure

Algorithm 2 Responding to a cluster-head with a route to $\texttt{Rqs}$.

procedure iROUTING_RESPONSE$(n, L, \mathcal{T}_j, \mathcal{C}_j, \mathcal{S}_j, s)$
  s sends $\texttt{RREP}_\texttt{Rqs}$ to the $(L-n)$-th device of $\mathcal{S}_j$, let it be s_i;
  if $s_i \neq \texttt{C}$ **then**
    $\mathcal{T}_j \cup \boldsymbol{p}(s_i)$, $\mathcal{C}_j \cup c(s_i)$, $n \leftarrow n+1$;
    iROUTING_RESPONSE$(n, L, \mathcal{T}_j, \mathcal{C}_j, \mathcal{S}_j, s_i)$;
  else
    Execute iROUTING$(\texttt{Rqs}, \texttt{D}, \mathcal{S}_\texttt{j}, \mathcal{T}_\texttt{j}, \mathcal{C}_\texttt{j})$;
    break;
  end if
end procedure

the set $\mathcal{T}_j$ of vectors of "failing-to-detect" probabilities, for different malware, of devices in r_j; and (iii) the set $\mathcal{C}_j$ of computational malware inspection costs $c(s_i)$ of devices in r_j. These values are updated while the $\texttt{RREP}_\texttt{Rqs}$ is traveling back to $\texttt{C}$. When each device (e.g. s_i) that is involved in the route response phase, receives the $\texttt{RREP}_\texttt{Rqs}$, it updates $\mathcal{T}_j$ and $\mathcal{C}_j$. Within the time period T_{req}, $\texttt{C}$ aggregates $\texttt{RREP}_\texttt{Rqs}$ messages and updates its routing table with information that can be used to derive the *optimal routing strategy*, as dictated by Theorem 2.

In the third phase of the protocol, described in Algorithm 3, $\texttt{C}$ uses its routing table to solve the MDG by computing the *Nash Delivery Plan*, denoted by $\boldsymbol{\rho}^{\text{NE}}$, which has a lifetime T. Then, $\texttt{C}$ probabilistically selects a route according to $\boldsymbol{\rho}^{\text{NE}}$ to deliver the requested data to $\texttt{Rqs}$. The chosen route is denoted by r^*. Note that for the same $\texttt{Rqs}$ and before T expires, $\texttt{C}$ uses the same $\boldsymbol{\rho}^{\text{NE}}$ to derive r^*, upon a new REQUEST.

Algorithm 3 Delivering data to `Rqs`.

procedure iROUTING($\texttt{Rqs}, \texttt{D}, \mathcal{S}_\texttt{j}, \mathcal{T}_\texttt{j}, \mathcal{C}_\texttt{j}$)
  `C` derives the *Nash Delivery Plan*, $\boldsymbol{\rho}^{\text{NE}}$ using $\mathcal{S}_j, \mathcal{T}_j, \mathcal{C}_j$;
  `C` chooses r^* probabilistically as dictated by $\boldsymbol{\rho}^{\text{NE}}$;
  `C` delivers `D` to `Rqs` over r^*;
  Each device $s_i \in r^*$ performs data inspection;
  if `D` found to carry malware **then**
    s_i drops `D`;
    s_i notifies `C` by sending a notification message along the reverse path;
    `C` blacklists the device that sent, through the cloud, `D` consisting of malware;
  else
    s_i forwards `D` to `Rqs`;
  end if
end procedure

Also, the third phase focuses on detecting malware injected along with the requested data (denoted by `D`) to prevent the infection of `Rqs`. While `D` is delivered to `Rqs` over r^*, the relay devices, on r^*, perform data inspection auditing `D` for malware. Upon successful detection, the device that detects the malware, first drops `D`, and then notifies `C` that `D` was crafted with malware. The notification message is sent along the reverse path. When receiving this, `C` blacklists the device that has originally sent `D` (this device is assumed that has hijacked the communication link between MEC server and the cluster-head). This can be seen as the first step towards mitigating the investigated attack model and anything beyond that is out of the scope of this paper.

While each data `D` is collaboratively inspected by the devices on its way to `Rqs`, the derivation of the *optimal routing strategy*, i.e. the Nash Delivery Plan (NDP), is computed only by `C` through solving a Malware Detection Game (MDG) for this specific destination `Rqs`. Therefore, even if the other devices are aware of the existence of some infected data, it is only `C` that isolates the Attacker (i.e. data source) towards mitigating future malware infection risks.

The communications complexity of the iRouting protocol measured in terms of number of messages exchanged in performing route discovery is $\mathcal{O}(2N)$, where N is the number of devices in the D2D network. As a reactive routing protocol, iRouting has higher storage complexity than conventional routing protocols, but it supports multiple-path routing and QoS routing making malware detection optimal, as shown in section 5. Finally, iRouting has a time complexity equal to $\mathcal{O}(2D)$, where D is the diameter of the D2D network.

Table 3: Simulation parameter values

Parameter	**Value**
Number of nodes	20
Mobility model	Linear Mobility
Mobility Speed	10 m/s
Mobility Update Interval	0.1 s
Packet size	512 bytes
Packet generation rate	2 packets/s
Simulation time	600 s

7. Simulations

7.1. Network setup

We have conducted a series of simulations to evaluate the performance of the optimal strategies in D2D networks. Devices have been randomly deployed inside a rectangular area of 1000m x 1000m. For each device, the transmission power is fixed, and the maximum transmission range is 200m, while two devices can directly communicate with each other only if they are in each others transmission range. We have performed the simulations using the OMNeT++ network simulator and INET framework. We have simulated the IEEE 802.11 MAC layer protocol and devices send UDP traffic. In the simulations, the requestor of some data is chosen randomly, and the total number of devices of a *cluster* is set to be 20. The total simulation time varies (10, 20, 40, 60, 120 seconds) to confirm the consistency of results. Table 3 summarizes the simulation parameters.

7.2. Security controls and malware

Simulations consider one adversary who is injecting a sequence of consecutive malicious replies with the aim to infect `Rqs`. We assume that the Attacker chooses to inject one of $[M] = \{$Keylogger, SMS spam, Rootkit iSAM, Spyware, iKee-B, Premium-Rate calls$\}$ malware types (i.e. pure strategies of the Attacker). We have also assumed the anti-malware controls, SMS Profiler, iDMA, iTL, and Touchstroke, along with their detection rates, as published in [49]. Each mobile device is equipped with at least one and up to three anti-malware controls.

7.3. Attackers

We have simulated 3 different Attacker types; namely *Uniform*, *Weighted*, and *Nash* Attacker:

- *Uniform*: the Attacker chooses each malware type from the set with equal probability. For example for the set we have used here, there is a probability $\frac{1}{6} = 0.1667$ the Attacker to choose any of the malware types of $[M]$;

- *Weighted*: the Attacker chooses a malware type with probability derived by the following algorithm:
  1. find the average utility value of the Attacker for each column of the game matrix;
  2. add the average utility values of the Attacker for all columns to get the combined sum;
  3. for each malware type, derive the probability of a malware type to be chosen by dividing its average utility value, found in step 1, by the sum derived in step 2.
- *Nash*: the Attacker plays according to her Nash strategy $\boldsymbol{\mu}^{NE}$.

Per Reply, the simulator chooses an attack sample from the attack probability distribution which is determined by the Attacker profile.

We have introduced different probability distributions for each Attacker type, only for testing purposes. Nevertheless, *i*Routing is optimal regardless of the probability distribution of a malware type to be chosen by the Attacker; a petition that is formally consolidated by the mathematical results presented in sections 4 and 5 as well as the simulation results uncovered in this section.

7.4. Experiments

We have considered 5 *Cases* each referring to different simulation times: 10, 20, 40, 60, and 120 mins. For each Case we have simulated 1,000 replies, which are UDP messages of length 512 bytes with delay limit 100 seconds, for a fixed network topology. Yet we refer to the run of the code for the pair ⟨Case, #replies⟩ by the term *Experiment*. We have repeated each Experiment for 10 independent network topologies to get a clear idea of the results' trend. We do that for all 5 Cases and each type of Attacker profile. Thus we simulate, in total: 5 Cases $\times$ $1,000$ replies $\times$ 10 network topologies $= 50,000$ replies.

7.5. Comparisons

We compare *i*Routing against AODV, DSR, and custom-made routing protocol called *Proportional Routing* (PR), for different Attacker types.

PR is computed as follows. First, by using the game matrix, the Defender computes the average utility value for each row, let it be

$$\hat{U}_d(r_j) = \frac{\sum_{m_l=1}^{M} U_d(r_j, m_l)}{M}, \ \forall \ r_j \in [R]. \tag{29}$$

Then, the probability of route r_j to be chosen equals:

$$1 - \frac{\hat{U}_d(r_j)}{\sum_{r=1}^{R} \hat{U}_d(r)}. \tag{30}$$

According to the results illustrated in Figures 2 - 4, *i*Routing consistently outperforms the rest of the protocols, in terms of both Defender's *expected utility* and *average detection rate*, for all different simulation times and Attacker types. The results show that *i*Routing achieves its highest average malware detection rate (∼65%) against a Uniform Attacker (non-strategic Attacker), and its worst rate against a Weighted Attacker. In the case of a Nash Attacker, *i*Routing has almost 22% higher detection rate than PR, 6% than DSR, while it is twice more efficient (i.e. ∼11%) than AODV. For a Weighted Attacker, PR behaves differently as it achieves approximately 6% lower average detection rate than *i*Routing, in contrast to DSR and AODV, which perform worse, as opposed to the Nash Attacker case, since the difference of their average detection rate compared to *i*Routing becomes double (i.e.∼12% for DSR and 24% for AODV). Finally, for a Uniform Attacker, the difference, in terms of detection rate, compared to *i*Routing, is almost the same for both DSR and PR, which is approximately equivalent to 8%. AODV still has the worst average detection rate among all protocols by having 24% worse rate than *i*Routing.

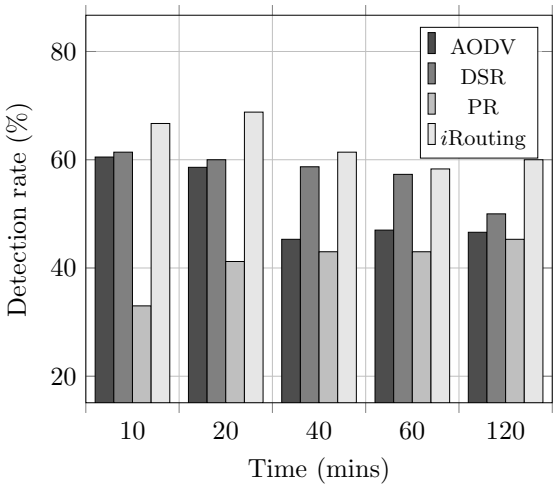

Figure 2: Malware detection rate in presence of a Nash attacker.

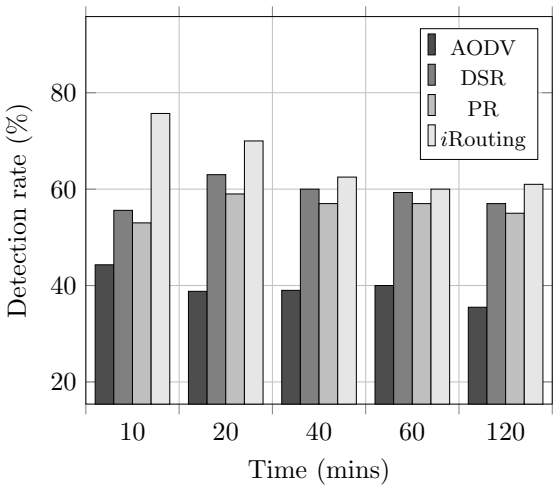

Figure 3: Malware detection rate in presence of a Uniform attacker.

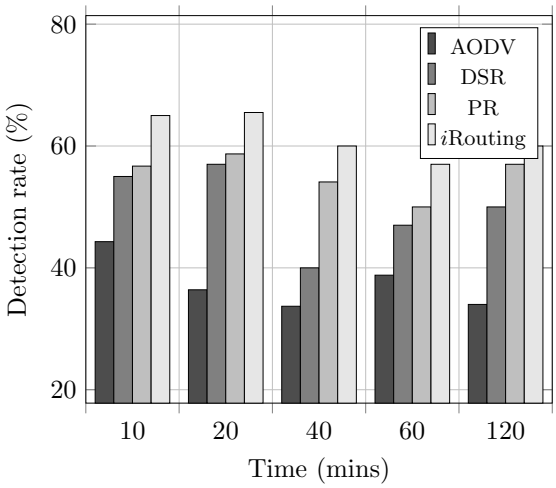

Figure 4: Malware detection rate in presence of a Weighted attacker.

According to Figures 5 - 7, *i*Routing achieves the best performance in terms of average expected utility among all protocols. More specifically, *i*Routing improves the average expected utility, in the case of a Nash Attacker, by, in average, 49%, 17%, and 7% compared to PR, AODV, and DSR, respectively. We notice that the Defender's utility in *i*Routing is similar to the one achieved when DSR is used. The reason for this is that DSR improves computational cost as opposed to *i*Routing more than AODV and PR while exhibiting the best detection rate among AODV and PR. Average improvement values are slightly more pronounced

for a non-strategic Uniform Attacker; 16%, 68%, and 37%, as opposed to the same protocols. The situation is similar for a Weighted Attacker, in which case the corresponding improvement values are 18%, 53%, and 20%. We also notice that the behaviour of all protocols but iRouting is stochastic despite of iRouting having steadily the best performance.

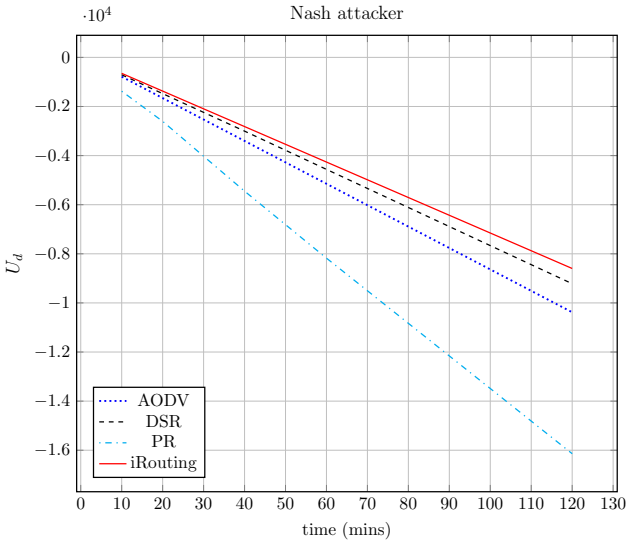

Figure 5: Utility of the Defender in presence of a Nash attacker.

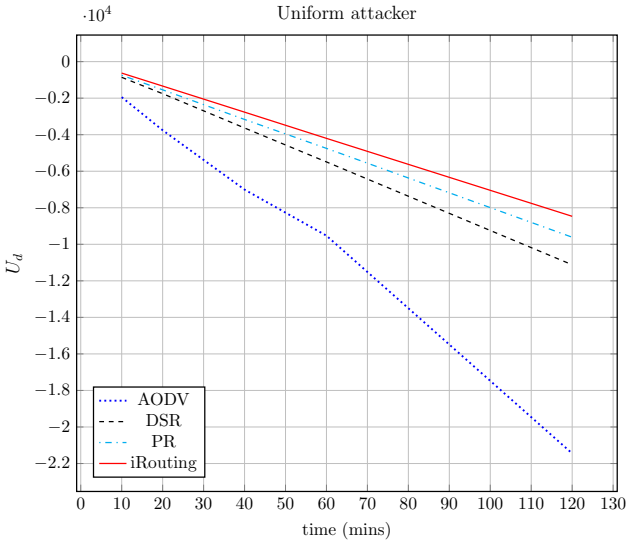

Figure 6: Utility of the Defender in presence of a Uniform attacker.

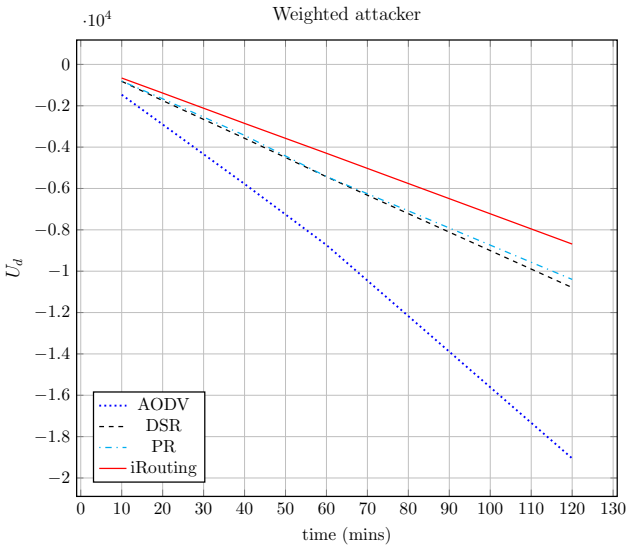

Figure 7: Utility of the Defender in presence of a Weighted attacker.

8. Conclusion

In this paper, we have formally investigated how to select an end-to-end path to deliver data from a source to a destination in device-to-device networks under a game theoretic framework. We assume the presence of an external adversary who aims to infect "good" network devices with malware. First, a simple yet illuminating two-player security game, between the network (the Defender) and an adversary, is studied. To devise optimal routing strategies, optimality analysis has been undertaken for different types of games to prove, *in theory*, that there is a Nash equilibrium strategy that always makes the Defender better-off. The analysis has shown that the expected security damage that can be inflicted by the *Attacker* is bounded and limited when the proposed strategy is used by the Defender. Network simulation results have also illustrated, *in practice*, that the proposed strategy can effectively mitigate malware infection. In future work, we intend to investigate machine learning algorithms (e.g. boosting) to convert weak learners (e.g. devices with limited number of anti-malware controls) to strong ones.

9. References

[1] D. Feng, L. Lu, Y. Yuan-Wu, G. Ye Li, S. Li, G. Feng, Device-to-device communications in cellular networks, IEEE Commun. Mag. 52 (4) (2014) 49–55.

[2] H. Nishiyama, M. Ito, N. Kato, Relay-by-smartphone: realizing multihop device-to-device communications, IEEE Commun. Mag. 52 (4) (2014) 56–65.

[3] M. Tehrani, M. Uysal, H. Yanikomeroglu, Device-to-device communication in 5G cellular networks: challenges, solutions, and future directions, IEEE Commun. Mag. 52 (5) (2014) 86–92.
[4] G. Fodor, E. Dahlman, G. Mildh, S. Parkvall, N. Reider, G. Miklos, Z. Turanyi, Design aspects of network assisted device-to-device communications, IEEE Commun. Mag. 50 (3) (2012) 170–177.
[5] K. Doppler, M. Rinne, C. Wijting, C. Ribeiro, K. Hugl, Device-to-device communication as an underlay to LTE-advanced networks, IEEE Commun. Mag. 47 (12) (2009) 42–49.
[6] C. A. Ardagna, M. Conti, M. Leone, J. Stefa, An anonymous end-to-end communication protocol for mobile cloud environments, IEEE Trans. Serv. Comput. 7 (3) (2014) 373–386.
[7] C. Perkins, E. Belding-Royer, S. Das, Ad hoc on-demand distance vector (AODV) routing, RFC 3561 (Jul. 2003).
[8] D. Johnson, Y. Hu, D. Maltz, The Dynamic Source Routing protocol (DSR) for mobile ad hoc networks for IPv4, RFC 4728 (Feb. 2007).
[9] T. Clausen, P. Jacquet, Optimized Link State Routing Protocol (OLSR), RFC 3626 (Oct. 2003).
[10] T. Ramrekha, E. Panaousis, C. Politis, Standardisation advancements in the area of routing for mobile ad-hoc networks, J. of Supercomputing 64 (2) (2013) 409–434.
[11] G. Suarez-Tangil, J. E. Tapiador, P. Peris-Lopez, A. Ribagorda, Evolution, detection and analysis of malware for smart devices, IEEE Communications Surveys Tutorials 16 (2).
[12] M. Khouzani, S. Saswati, E. Altman, Maximum damage malware attack in mobile wireless networks, IEEE/ACM Trans. Netw. 20 (5) (2012) 1347–1360.
[13] R. Heartfield, G. Loukas, A taxonomy of attacks and a survey of defence mechanisms for semantic social engineering attacks, ACM Computing Surveys (CSUR) 48 (3) (2016) 37.
[14] M. La Polla, F. Martinelli, D. Sgandurra, A survey on security for mobile devices, IEEE Commun. Surveys Tuts. 15 (1) (2012) 446–471.
[15] T. Alpcan, T. Basar, Network security: a decision and game-theoretic approach, Cambridge University Press, 2010.
[16] M. Naserian, K. Tepe, Game theoretic approach in routing protocol for wireless ad hoc networks, Ad Hoc Netw. 7 (3) (2009) 569 – 578.
[17] Y. Xiao, K.-C. Chen, C. Yuen, Z. Han, L. A. DaSilva, A bayesian overlapping coalition formation game for device-to-device spectrum sharing in cellular networks, IEEE Transactions on Wireless Communications 14 (7) (2015) 4034–4051.
[18] C. Long, Q. Chi, X. Guan, T. Chen, Joint random access and power control game in ad hoc networks with noncooperative users, Ad Hoc Netw. 9 (2) (2011) 142–151.
[19] F. Wang, O. Younis, M. Krunz, Throughput-oriented mac for mobile ad hoc networks: A game-theoretic approach, Ad Hoc Netw. 7 (1) (2009) 98 – 117.
[20] Y. Jianting, M. Chuan, Y. Hui, Z. Wei, Secrecy-based access control for device-to-device communication underlaying cellular networks, IEEE Commun. Mag. 17 (11) (2013) 2068–2071.
[21] Z. Daohua, A. Swindlehurst, S. Fakoorian, X. Wei, Z. Chunming, Device-to-device communications: The physical layer security advantage, IEEE Int. Conf. on Acoust., Speech, Signal Process. (2014) 1606–1610.
[22] L. Abusalah, A. Khokhar, M. Guizani, A survey of secure mobile ad hoc routing protocols, IEEE Commun. Surveys Tuts. 10 (4) (2008) 78–93.
[23] S. Gupte, M. Singhal, Secure routing in mobile wireless ad hoc networks, Ad Hoc Netw. 1 (1) (2003) 151–174.
[24] E. Panaousis, T. Alpcan, H. Fereidooni, M. Conti, Secure message delivery games for device-to-device communications, in: R. Poovendran, W. Saad (Eds.), Decision and Game Theory for Security, Vol. 8840 of Lecture Notes in Computer Science, Springer International Publishing, 2014, pp. 195–215.

[25] A. Patcha, J. M. Park, A game theoretic approach to modeling intrusion detection in mobile ad hoc networks, in: Proc. 5th Annu. SMC Information Assurance Workshop, 2004, pp. 280–284.
[26] Y. Liu, C. Comaniciou, H. Man, A bayesian game approach for intrusion detection in wireless ad hoc networks, in: Proc. 2006 workshop on Game Theory for Communications and Networks, 2006, pp. 1–12.
[27] Y. Liu, C. Comaniciu, H. Man, Modelling misbehaviour in ad hoc networks: a game theoretic approach for intrusion detection, Int. J. of Security and Netw. 1 (7) (2006) 243–254.
[28] N. Marchang, R. Tripathi, A game theoretical approach for efficient deployment of intrusion detection system in mobile ad hoc networks, in: Proc. 2007 Int. Conf. on Advanced Computing and Communications, 2007, pp. 460–464.
[29] H. Otrok, M. Debbabi, C. Assi, P. Bhattacharya, A cooperative approach for analyzing intrusions in mobile ad hoc networks, in: Proc. 27th Int. Conf. on Distributed Computing Systems Workshops, 2009, pp. 985–992.
[30] N. Santosh, R. Saranyan, K. Senthil, V. Vetriselvi, Cluster based co-operative game theory approach for intrusion detection in mobile ad-hoc grid, in: Proc. of the International Conference on Advanced Computing and Communications (ADCOM), 2008, pp. 273–278.
[31] J. Cho, I. Chen, P. Feng, Effect of intrusion detection on reliability of mission-oriented mobile group systems in mobile ad hoc networks, IEEE Trans. Rel. 59 (1) (2010) 231–241.
[32] M. Felegyhazi, L. Buttyan, J. Hubaux, Nash equilibria of packet forwarding strategies in wireless ad hoc networks, IEEE Trans. Mobile Comput. 5 (5) (2006) 463–476.
[33] W. Yu, K. Liu, Game theoretic analysis of cooperation stimulation and security in autonomous mobile ad hoc networks, IEEE Trans. Mobile Comput. 6 (5) (2007) 507–521.
[34] W. Yu, Z. Ji, K. Liu, Securing cooperative ad-hoc networks under noise and imperfect monitoring: strategies and game theoretic analysis, IEEE Trans. Inf. Forensics Security 2 (2) (2007) 240–253.
[35] W. Yu, K. Liu, Secure cooperation in autonomous mobile ad-hoc networks under noise and imperfect monitoring: a game-theoretic approach, IEEE Trans. Inf. Forensics Security 3 (2) (2008) 317–330.
[36] E. Panaousis, C. Politis, A game theoretic approach for securing AODV in emergency mobile ad hoc networks, in: Proc. 34th IEEE Conf. on Local Computer Networks, 2009, pp. 985–992.
[37] D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, M. Tambe, Stackelberg vs. Nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness, J. Artif. Intell. Res. 41 (2011) 297–327.
[38] M. Tambe, Security and game theory: algorithms, deployed systems, lessons learned, Cambridge University Press, 2011.
[39] A. Wang, Y. Cai, W. Yang, Z. Hou, A Stackelberg security game with cooperative jamming over a multiuser OFDMA network, in: Proc. 2013 IEEE Wireless Communications and Networking Conference, 2015, pp. 4169–4174.
[40] D. Kar, F. Fang, F. Delle Fave, N. Sintov, M. Tambe, A Game of Thrones: when human behavior models compete in repeated stackelberg security games, in: Proc. 2015 International Conference on Autonomous Agents and Multiagent Systems, 2015, pp. 1381–1390.
[41] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the Internet of Things, in: Proc. 1st MCC Workshop on Mobile Cloud computing, 2012, pp. 13–16.
[42] A. Asadi, Q. Wang, V. Mancuso, A survey on device-to-device communication in cellular networks, Communications Surveys & Tutorials, IEEE 16 (4) (2014) 1801–1819.
[43] D. Dolev, A. Yao, On the security of public key protocols, IEEE Trans. Inf. Theory 29 (2) (1983) 198–208.
[44] M. J. Osborne, A. Rubinstein, A course in game theory, MIT press, 1994.

[45] C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordóñez, M. Tambe, Computing optimal randomized resource allocations for massive security games, in: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1, International Foundation for Autonomous Agents and Multiagent Systems, 2009, pp. 689–696.

[46] J. Von Neumann, O. Morgenstern, Theory of games and economic behavior (60th anniversary commemorative edition), Princeton university press, 2007.

[47] J. Nash, Equilibrium points in n-person games., in: Proc. of the National Academy of Sciences, 1950, pp. 48–49.

[48] T. Basar, G. J. Olsder, Dynamic noncooperative game theory, London Academic press, 1995.

[49] D. Damopoulos, G. Kambourakis, G. Portokalidis, The best of both worlds: a framework for the synergistic operation of host and cloud anomaly-based ids for smartphones, Proc. 7th European Workshop on System Security.