

 Open access • Journal Article • DOI:10.1016/J.ADHOOC.2016.11.008

Game theoretic path selection to support security in device-to-device communications — [Source link](#)

Emmanouil Panaousis, Eirini Karapistoli, Hadeer Elsemary, Tansu Alpcan ...+2 more authors

Institutions: University of Brighton, University of Göttingen, University of Melbourne, Queen Mary University of London ...+1 more institutions

Published on: 01 Mar 2017 - Ad Hoc Networks

Topics: Mobile malware, Link-state routing protocol, Routing protocol, Mobile edge computing and Malware

Related papers:

- [Differential Security Game in Heterogeneous Device-to-Device Offloading Network Under Epidemic Risks](#)
- [Controlling proximity-malware infection in diverse tactical mobile networks using K-distance pruning](#)
- [Secure message delivery games for device-to-device communications](#)
- [An optimal distributed malware defense system for mobile networks with heterogeneous devices](#)
- [MeDrone: On the use of a medical drone to heal a sensor network infected by a malicious epidemic](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/game-theoretic-path-selection-to-support-security-in-device-3mhnl28u68>

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

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

5 ^aUniversity of Brighton, UK

6 ^bCapritech Limited, UK

7 ^cUniversity of Gottingen, Germany

8 ^dUniversity of Melbourne, Australia

9 ^eQueen Mary University of London, UK

10 ^fUniversity of Macedonia, Greece

11 **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*.

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

14 **1. Introduction**

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

27 *1.1. Motivation*

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

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

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

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

65 1.2. Innovation

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

79 Our protocol, called *i*Routing (abbreviating “intelligent Routing”), is de-
80 signed upon the theoretical analysis of a simple yet illuminating two-player se-
81 curity game between the *Defender*, which abstracts a D2D network, and the
82 *Attacker*, which abstracts any adversarial entity that wishes to inject malware
83 into the D2D network. We have proven that the Defender’s *equilibrium strategies*
84 leave the network better off, in terms of *expected payoff*, which is a combination
85 of *security damage* and *malware detection cost* (i.e. cycles process units). Note
86 that *i*Routing can work on top of underlying physical and MAC layer protocols
87 [20, 21].

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

93 1.3. Progress beyond relevant work

94 This paper extends, in a significant manner, the results initially presented in
95 [24]. The exact differences are summarized below.

- 96 • [24] assumes a pure device-to-device network while in this paper the device-
97 to-device network has been enriched with a part of mobile edge comput-
98 ing. The network devices request services from the MEC server and multi-
99 hopping enables communication between the MEC server and the different
100 devices to overcome proximity issues due to the latter being outside the
101 transmission range of the server. In this paper, the security challenge is
102 how to safely utilise MEC services where a cluster-head (i.e. MEC server)
103 might be compromised by an adversary. Although this does not introduce
104 any new challenge in terms of malware detection and routing, it is an as-
105 sumption that places the idea of the paper within mobile edge computing
106 and 5G architectures.
- 107 • This paper assumes different mobile operating systems and these can be
108 infected with different types of malware as opposed to [24], which goes as
109 far as considering just a set of malicious messages that are sent from the
110 attacker’s device to infect the legitimate devices. This also has the effect of
111 defining, in this paper, the Malware Detection Game whereas in [24], the
112 defined game is called Secure Message Delivery Game.
- 113 • In [24], a confusion matrix is defined to determine how the different devices
114 of the network can detect malicious messages. In this paper here we take
115 a more realistic, in the terms of cyber security, approach where for each
116 device there is a probability to be compromised by malware. Therefore,

- 117 each route has, in turn, a penetration level, which is the probability the
 118 route to be compromised due to one or more devices on it being vulnerable.
- 119 • In [24], the details about the interdependencies of malicious message de-
 120 tectors is not discussed, while in our paper here we explicitly say that
 121 each control detects different signs of malware and *no interdependencies*, in
 122 terms of detection capabilities, are assumed, i.e. we have assumed that an
 123 anti-malware control is the minimal piece of software that detects certain
 124 malicious signs.
 - 125 • In [24], the Attacker is not assumed to monitor the network before launching
 126 a malware attack (no reconnaissance) while in our paper here the Attacker
 127 surveils the network before injecting malware giving us a Stackelberg game
 128 to study.
 - 129 • In [24], only Nash Equilibria (NE) and maximin strategies have been stud-
 130 ied. On the other hand, our paper here derives Strong Stackelberg Equi-
 131 libria (SSE) and shows the relationship among three of them; SSE, NE
 132 and maximin. Not only that, but this paper exhibits much larger depth of
 133 mathematical analysis referring also to best responses of players. Finally,
 134 it proves the equality of strategies of different games, such zero-sum and
 135 non-zero sum across all strategic types (Nash, Stackelberg, maximin).
 - 136 • Although Panaousis et al. [24] has investigated both zero sum and non-zero
 137 sum games, where in the latter the utility of the Attacker is a positive affine
 138 transformation (PAT) of the defender's utility, in this paper we go beyond
 139 that. We show the equality of the different strategies holds in a more generic
 140 (i.e. than the PAT case) payoff structure where the Attackers utility is a
 141 strictly positive scaling of the Defender's utility.
 - 142 • All simulations in [24] were numeric; as well as they do not compare the
 143 performance of the proposed routing protocol with other device-to-device
 144 routing protocols. For the purposes of our paper here we have undertaken
 145 a network simulation to compare the proposed protocol with legacy routing
 146 protocols using the OMNeT++ network simulator. In this way we have
 147 simulated physical and link-layer network characteristics.
 - 148 • In our paper here, we have considered, in our simulations, the efficacies of
 149 some of the most-recent real-world anti-malware controls against real-world
 150 malware types as opposed to the purely numeric assignment to the different
 151 variables.
 - 152 • In our simulations here, we have included a new Attacker type, called
 153 Weighted, which allows the adversary to distribute her resources propor-

154 tionally, over the different routes, aiming at the highest expected dam-
155 age. This type of Attacker was not simulated in [24].

156 1.4. Main assumptions

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

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

172 1.5. Outline

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

182 2. Related work

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

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

201 In a more security-oriented vein, Yu et al. [33] have used game theory to study
202 the dynamic interactions, in mobile ad hoc (device-to-device) networks, between
203 “good” nodes, which initially believe that all other nodes are not malicious, and
204 “adversaries”, which are aware of which nodes are good. They propose secure
205 routing and packet forwarding games that consist of 3 stages: route participa-
206 tion; route selection; and packet forwarding. In the first stage, a node decides
207 whether to be part of route or not; in the second phase, a node who wishes to
208 send a packet to a destination, after it discovers a *valid route* (called when all
209 nodes agree to be part of it), it either uses the discovered route or not; and, fi-
210 nally, in the third phase, each relay node decides to forward or not an incoming
211 packet. They have derived optimal defence strategies and studied the maximum
212 potential damage, which incurs when attackers find a route with maximum num-
213 ber of hops and they inject malicious traffic into it. The same authors also com-
214 bined this game with a secure routing game but without considering noise and
215 imperfect monitoring. Yu et al. [34] extended [33] and proposed a secure cooper-
216 ation game under noise and imperfect monitoring. Likewise, Yu and Liu tackled
217 the same challenge and presented a richer set of performance evaluation results in
218 [35]. The above publications do not tackle the same challenge with *i*Routing, as
219 they do not investigate the selection of a route among an available set of routes
220 to deliver packets from a source to a destination

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

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

232 3. System description and game model

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

240 3.1. System description

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

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

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

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

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

274 3.2. Adversarial model

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

289 3.3. Malware detection

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

303 3.4. Formulation

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

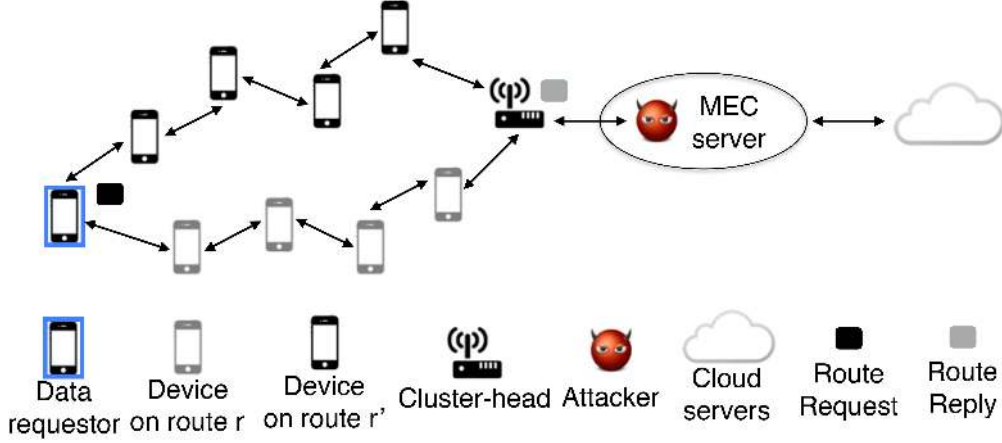


Figure 1: Investigated system model, where a device requests data, that the cluster devices do not possess, from the MEC server. The adversary has successfully launched a MITM attack controlling the communication between cluster-head and MEC server.

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

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

325 Let us consider Λ different mobile operating systems, and M_λ different mal-
 326 ware available to the Attacker to infect devices that run a mobile operating system
 327 $\lambda \in [\Lambda]$. Each device may run one or more anti-malware controls and for each λ
 328 we assume AM_λ anti-malware controls, which can mitigate malware that targets
 329 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 function² $\mathbf{1}_{[AM_\lambda^i]} : [AM_\lambda] \rightarrow \{0, 1\}$ defined as follows:

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

330 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

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

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

335 We define as

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

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

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

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

²this 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	\mathbf{C}	Cluster-head
\mathbf{Rqs}	Data requestor	\mathbf{D}	Requested data
$[R]$	Set of routes from \mathbf{C} to \mathbf{Rqs}	r_j	j -th route
\mathcal{S}_j	Set of devices on r_j	\mathbf{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	$\mathbf{p}(s_i)$	Vector of “failing-to-detect” probabilities of s_i for different malware
$p(r_j, m_l)$	Probability of \mathbf{Rqs} to be infected with malware m_l when \mathbf{D} is sent over r_j	$\mathbf{p}(r_j)$	Vector of infection probabilities for r_j and all malware types
$[M]$	Set of malware	ρ	Defender’s mixed strategy
μ	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 $\mathbf{p}(s_i)$ in r_j

347 3.5. Game model

348 Now that we have defined our system model by describing its components and
349 their relationship, in the rest of this section, we use game theory to investigate
350 the optimal strategic routing decisions of \mathbf{C} , the Defender, and the Attacker who
351 aims to infect one of the cluster devices with mobile malware. The Attacker’s
352 objective is to succeed an attack against \mathbf{Rqs} and the Defender must select a
353 route to deliver the **REPLY** to \mathbf{Rqs} .

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

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

364 as $U_d : (r_j, m_l) \rightarrow \mathbb{R}_-$ and $U_a : (r_j, m_l) \rightarrow \mathbb{R}_+$ the payoff functions of the Defender
365 and Attacker, respectively, when the pure strategy profile (r_j, m_l) is played. Ac-
366 cording to [44], we define a *preference relation* \succsim , when m_l is chosen by the
367 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 gen-
368 eral, given the set $[R]$ of all available routes from \mathcal{C} to \mathbf{Rqs} , a rational Defender can
369 choose a route (i.e. pure strategy) r^* that is *feasible*, that is $r^* \in [R]$, and *optimal*
370 in the sense that $r^* \succsim r, \forall r \in [R], r \neq r^*$; alternatively she solves the problem
371 $\max_{r \in [R]} U_d(r, m_l)$, for a message $m_l \in [M]$. Likewise, we can define the prefer-
372 ence 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
373 a route $r_j \in [R]$.

374 MDG can be seen as a *game per session*, where the start of each session is
375 signified by the transmission of a new **REPLY** that the cluster-head will send to
376 **Rqs**; it is also realistic to assume that over a time period, there will be multi-
377 ple sessions. To derive optimal strategies for the Defender during the repetitions
378 of MDGs, we deploy the notion of *mixed strategies*. Since players act independ-
379 ently, we can enlarge their strategy spaces, so as to allow them to base their
380 decisions on the outcome of random events that create uncertainty to the op-
381 ponent about individual strategic choices maximising their payoffs. Hence, both
382 Defender and Attacker deploy randomised (i.e. mixed) strategies. The mixed
383 strategy ρ of the Defender is a probability distribution over the different routes
384 (i.e. pure strategies) from \mathcal{C} to **Rqs**, where $\rho(r_j)$ is the probability of delivering
385 a **REPLY** via r_j under mixed strategy ρ . We refer to a mixed strategy of the
386 Defender as a *Randomised Delivery Plan* (RDP). For the finite nonempty set
387 $[R]$, let $\Pi_{[R]}$ represent the set of all probability distributions over it, i.e.

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

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

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

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

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

401 The above definitions state that the subset of routes (resp. malware) that
 402 are assigned positive probability by the mixed strategy ρ (resp. μ) is called the
 403 *support* of ρ (resp. μ). Note that a pure strategy is a special case of a mixed
 404 strategy, in which the support is a single action.

405 Now that we have defined the mixed strategies of the players, we can define
 406 MDG as the finite strategic game $\Gamma = \langle (\text{Defender, Attacker}), \Pi_{[R]} \times \Pi_{[M]}, (U_d, U_a) \rangle$.
 407 For a given mixed strategy profile $(\rho, \mu) \in \Pi_{[R]} \times \Pi_{[M]}$, we denote by $U_d(\rho, \mu)$, and
 408 $U_a(\rho, \mu)$ the expected payoff values of the Defender and Attacker, where the ex-
 409 pectation is due to the independent randomisations according to mixed strategies
 410 ρ , and μ .

411 Formally

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

412 and similarly

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

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

416 **Definition 3.** *The Defender's (resp. Attacker's) best response to the mixed strat-*
 417 *egy μ (resp. ρ) of the Attacker (resp. Defender) is a RDP $\rho^{\text{BR}} \in \Pi_{[R]}$ (resp. $\mu^{\text{BR}} \in$*
 418 *$\Pi_{[M]}$) such that $U_d(\rho^{\text{BR}}, \mu) \geq U_d(\rho, \mu)$, $\forall \rho \in \Pi_{[R]}$ (resp. $U_a(\rho, \mu^{\text{BR}}) \geq U_a(\rho, \mu)$, $\forall \mu \in$*
 419 *$\Pi_{[M]}$).*

420 It is noteworthy to mention that the game theoretic solutions that we will
 421 propose, in the next section, involve *randomisation*. For instance, in a mixed equi-
 422 librium, each player's randomisation leaves the other *indifferent* across her ran-
 423 domisation support. These choices can be deliberately randomised or be taken by
 424 software agents that run in mobile devices (i.e. cluster-heads or adversaries). How-
 425 ever these are not the only equilibria interpretations. For instance, the probabili-
 426 ties over the pure actions (i.e. route or malware pure selections) can represent
 427 (i) time averages of an "adaptive" player, (ii) a vector of fractions of a "popula-
 428 tion", where each player type adopts pure strategies and, (iii) a "belief" vector
 429 that each player has about the other regarding their behaviour.

430 4. Game solutions

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

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

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

441 4.1. Nash mixed strategies

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

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

- 450 1. $\rho^{\text{NE}} \succsim \rho, \forall \rho \in \Pi_{[R]}$, when the Attacker chooses μ^{NE} , i.e.

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

- 451 2. $\mu^{\text{NE}} \succsim \mu, \forall \mu \in \Pi_{[M]}$, when the Defender chooses ρ^{NE} , i.e.

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

452 **Definition 5.** *The Nash Delivery Plan (NDP), denoted by ρ^{NE} , is the probability*
 453 *distribution over the different routes, as determined by the NE of the MDG.*

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

457 4.2. Maximin strategies

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

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

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

Table 2: A toy game example

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

463 A maximinimiser for the Defender is an RDP that maximises the payoff that
 464 the Defender can *guarantee*. In other words, ρ^\dagger guarantees (i.e. “secures”) the
 465 Defender at least her maximin payoff regardless of μ , as ρ^\dagger solves the problem
 466 $\max_{\rho} \min_{\mu} U_d(\rho, \mu)$. That is why ρ^\dagger is also called *security strategy*.

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

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

469 4.3. Stackelberg mixed strategies

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

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

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

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

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

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

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

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

502 1. for any $\rho \in \Delta_{[R]}$, the Attacker plays a best-response $\mu^{\text{BR}}(\rho) \in \Delta_{[M]}$ that
 503 is,

$$U_a(\rho, \mu^{\text{BR}}(\rho)) \geq U_a(\rho, \mu(\rho)), \forall \mu(\rho) \neq \mu^{\text{BR}}(\rho); \quad (12)$$

504 2. for any $\rho \in \Delta_{[R]}$, the Attacker breaks ties in favour of the Defender, that
 505 is, when there are multiple best responses to ρ , the Attacker plays the best
 506 response $\mu^{\text{SSE}}(\rho) \in \Delta_{[M]}$ that maximises the Defender's payoff:

$$U_d(\rho, \mu^{\text{SSE}}(\rho)) \geq U_d(\rho, \mu^{\text{BR}}(\rho)), \quad (13)$$

$\forall \mu^{\text{BR}}$ best response to ρ ;

507 3. the Defender plays a best-response $\rho^{\text{SSE}} \in \Delta_{[R]}$, which maximises her payoff
 508 given that the Attacker's strategies are given by the first two conditions
 509 (i.e. the Attacker always plays best response with tie-breaking in favour of
 510 the Defender [38],[45]):

$$U_d(\rho^{\text{SSE}}, \mu^{\text{SSE}}(\rho^{\text{SSE}})) \geq U_d(\rho, \mu^{\text{SSE}}(\rho)), \forall \rho \neq \rho^{\text{SSE}}. \quad (14)$$

511 5. Optimality analysis

512 For the purpose of analysis, we consider *complete information* Nash MDGs,
 513 according to which both players know the game matrix, which contains the util-
 514 ities of both players for each pure strategy profile. The utility function of the
 515 Defender is determined by the probability of failing to detect a route and the
 516 overall performance cost, which is imposed on the devices of the j -th route when
 517 undertaking malware detection. We denote by $c(s_i)$ the performance cost imposed
 518 on each $s_i \in \mathcal{S}_j$ and therefore the overall performance cost over a route r_j equals
 519 $\sum_{s_i \in \mathcal{S}_j} c(s_i)$.

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

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

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

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

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

541 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
 542 have that

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

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

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

544 As Γ_0 is a zero sum game, the Attacker’s utility is given by $U_a^{\Gamma_0}(\boldsymbol{\rho}, \boldsymbol{\mu}) =$
 545 $-U_d^{\Gamma_0}(\boldsymbol{\rho}, \boldsymbol{\mu})$. Since the Defender’s equilibrium strategies maximise her utility,
 546 given that the Attacker maximises her own utility, we will refer to them as *optimal*
 547 *strategies*.

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

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

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

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

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

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

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

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

575 Formally, the Defender seeks to solve the following LP:

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

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

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

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

581 *Proof.* First, notice that U_d , as defined in (15), is a linear function in $\boldsymbol{\rho}(r_j)$ that
 582 is, for any two RDPs $\boldsymbol{\rho}_1$ and $\boldsymbol{\rho}_2$ and any number $\theta \in [0, 1]$ we have $U_d(\theta \boldsymbol{\rho}_1 + (1 -$
 583 $\theta) \boldsymbol{\mu}) = \theta U_d(\boldsymbol{\rho}_1) + (1 - \theta) U_d(\boldsymbol{\rho}_2)$. Then, for the sake of contradiction, assume
 584 there exists a route $r'_j \in \text{supp}(\boldsymbol{\rho}^{\text{NE}})$ selection that is not a best response to
 585 $\boldsymbol{\mu}^{\text{NE}}$. Due to the linearity of U_d in $\boldsymbol{\rho}^{\text{NE}}(r_j)$, the Defender can increase her payoff
 586 by transferring probability from $\boldsymbol{\rho}(r'_j)$ to a route selection that is a best response
 587 to $\boldsymbol{\mu}^{\text{NE}}$, creating a new mixed strategy $\boldsymbol{\rho}^* \succ \boldsymbol{\rho}^{\text{NE}}$. However, this contradicts the
 588 assumption that $\boldsymbol{\rho}^{\text{NE}}$ is the strategy of the Defender at the NE, as the Defender
 589 prefers to deviate from $\boldsymbol{\rho}^{\text{NE}}$ to gain a higher payoff, by playing $\boldsymbol{\rho}^*$. The second
 590 part of the lemma can be proven in the same way. \square

591 Let us now assume a non-zero sum MDG, denoted by Γ , with the same
 592 strategy spaces with Γ_0 , in which the Defender's utility is the same as in Γ_0 ,
 593 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 At-
 594 tacker's utility is (strictly positive) scaling of the security loss $S(r_j, m_l)$ of the
 595 Defender upon a successful attack. This is to say that the performance cost of
 596 the Defender is only important to her as the Attacker is only after compromising
 597 Rqs. Therefore, given a pure strategy profile (r_j, m_l) , the utility of the Attacker,
 598 in Γ , is defined as:

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

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

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

600 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

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

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

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

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

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

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

606 Here is the observation:

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

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

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

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

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

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

608 But these are exactly the conditions describing a NE of Γ_0 . Therefore $\Omega_\Gamma^{\text{NE}} =$
 609 $\Omega_{\Gamma_0}^{\text{NE}}$. \square

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

612 *Proof.* (\Rightarrow) Since Γ_0 is a two person zero-sum game, we know that the set of NE
 613 and Maximin strategies of the Defender are the same, i.e. $\Omega_{\Gamma_0}^{\text{NE}} = \Omega_{\Gamma_0}^{\text{maximin}}$. Let
 614 $(\boldsymbol{\rho}^{\text{NE}}, \boldsymbol{\mu}^{\text{NE}}) \in \Omega_\Gamma^{\text{NE}}$ then based on Lemma 2 we have that $(\boldsymbol{\rho}^{\text{NE}}, \boldsymbol{\mu}^{\text{NE}}) \in \Omega_{\Gamma_0}^{\text{NE}}$. Since
 615 Γ_0 is zero-sum, $\boldsymbol{\rho}^{\text{NE}} \in \Omega_{\Gamma_0}^{\text{maximin}}$. But the strategy spaces and the utility of the De-
 616 fender are the same in both Γ and Γ_0 . Hence the conditions for a mixed strategy to
 617 be a Defender's Maximin is the same in both games. Therefore, $\boldsymbol{\rho}^{\text{NE}} \in \Omega_\Gamma^{\text{maximin}}$,
 618 i.e. $\Omega_\Gamma^{\text{NE}} \subseteq \Omega_\Gamma^{\text{maximin}}$.

619 (\Leftarrow) The argument goes in the other direction as well: consider $\boldsymbol{\rho}^{\text{NE}} \in \Omega_\Gamma^{\text{maximin}}$. Since
 620 the utility of the Defender and the strategy spaces are the same across the two
 621 games, for the same strategy $\boldsymbol{\rho}^{\text{NE}}$, we have that $\boldsymbol{\rho}^{\text{NE}} \in \Omega_{\Gamma_0}^{\text{maximin}}$. Since Γ_0 is two-
 622 player zero-sum, there exists $\boldsymbol{\mu}^{\text{NE}}$ such that $(\boldsymbol{\rho}^{\text{NE}}, \boldsymbol{\mu}^{\text{NE}}) \in \Omega_{\Gamma_0}^{\text{NE}}$. From Lemma 2,
 623 this means $(\boldsymbol{\rho}^{\text{NE}}, \boldsymbol{\mu}^{\text{NE}})_\Gamma \in \Omega_\Gamma^{\text{NE}}$. Hence, *Maximin strategies of the Defender are*
 624 *also part of her NE strategies in Γ , i.e. $\Omega_\Gamma^{\text{maximin}} \subseteq \Omega_\Gamma^{\text{NE}}$. Putting the two together*
 625 $\Omega_\Gamma^{\text{NE}} = \Omega_\Gamma^{\text{maximin}}$. \square

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

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

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

- 639 (i) $\boldsymbol{\rho}^{\text{NE}} \in \arg \max_{\boldsymbol{\rho} \in \Delta_{[R]}} U_d(\boldsymbol{\rho}, \boldsymbol{\mu}^{\text{BR}}(\boldsymbol{\rho}))$ where;
640 (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;
641 (iii) for any $\boldsymbol{\rho} \in \Delta_{[R]}$:

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

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

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

In short, condition (ii) is equivalent to:

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

643 This makes condition (iii) irrelevant. But conditions (i) and (iv) exactly describe
644 a Maximin strategy of the Defender. Therefore we have proved that $\Omega_{\Gamma}^{\text{SSE}} \subseteq$
645 $\Omega_{\Gamma}^{\text{maximin}}$. (\Leftarrow) The argument can be established identically in reverse direction,
646 starting from a Maximin strategy of the Defender. So given conditions (i) and
647 (iv) we must prove that conditions (ii) and (iii) are true. Let $\boldsymbol{\rho}^{\text{NE}} \in \Omega_{\Gamma}^{\text{maximin}}$ be
648 a Maximin strategy of the Defender. Then by definition, $\boldsymbol{\rho}^{\text{NE}}$ is (i) an optimal
649 strategy of the Defender given that (iv) the Attacker is minimising Defender's
650 utility. We see that condition (ii) is true if and only if condition (iv) is true. Since
651 the Maximin strategy $\boldsymbol{\rho}^{\text{NE}}$ makes condition (iv) true, it will also make condition

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

$$\begin{aligned}
 \mu^{\text{BR}}(\rho) \notin \arg \max_{\mu \in \arg \max_{\mu} U_a(\rho, \mu)} U_a(\rho, \mu) &\iff \\
 \mu^{\text{BR}}(\rho) \notin \arg \max_{\mu \in \arg \max_{\mu} U_a(\rho, \mu)} \{-S(\rho, \mu) - k(\rho)\} &\iff \\
 \mu^{\text{BR}}(\rho) \notin \arg \min_{\mu \in \arg \max_{\mu} U_a(\rho, \mu)} \{S(\rho, \mu) + k(\rho)\} &\iff \quad (28) \\
 \mu^{\text{BR}}(\rho) \notin \arg \min_{\mu \in \arg \max_{\mu} U_a(\rho, \mu)} S(\rho, \mu) &\iff \\
 \mu^{\text{BR}}(\rho) \notin \arg \min_{\mu \in \arg \max_{\mu} U_a(\rho, \mu)} U_a(\rho, \mu), &
 \end{aligned}$$

655 which leads to a contradiction. Therefore condition (3) holds, and putting
 656 together all three conditions (1), (2), and (3), we have that ρ^{NE} , which is a
 657 Maximin strategy of the Defender it is also an SSE strategy, i.e. $\Omega_{\Gamma}^{\text{maximin}} \subseteq$
 658 $\Omega_{\Gamma}^{\text{SSE}}$. Putting the two proofs together we have that $\Omega_{\Gamma}^{\text{maximin}} = \Omega_{\Gamma}^{\text{SSE}}$. \square

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

662 *Proof.* Trivially, from Lemmas 3 and 4 we have that $\Omega_{\Gamma}^{\text{NE}} = \Omega_{\Gamma}^{\text{maximin}} = \Omega_{\Gamma}^{\text{SSE}}$. Since
 663 Γ_0 is a two person zero-sum game, we know that all NE are interchangeable
 664 [48]. From Lemma 2 the NE of Γ_0 are the NE of Γ and vice-versa. We also see
 665 that the utility of the Defender is the same across Γ and Γ_0 . Therefore the utility
 666 of the Defender in all NE of our original game is the same, which also implies
 667 that all NE of our original game are interchangeable. \square

668 The above lemma establishes that the Defender, regardless of whether the At-
 669 tacker conducts surveillance, she plays optimally when she randomises according
 670 to her NE strategy.

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

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

674 *the Defender plays optimally by choosing any strategy $\rho \in \Omega_{\Gamma_0}^{\text{NE}}$.*

675 *Proof.* By combining 2 and 1, we have that $\Omega_{\Gamma_0}^{\text{NE}} = \Omega_{\Gamma}^{\text{NE}} = \Omega_{\Gamma}^{\text{maximin}} = \Omega_{\Gamma}^{\text{SSE}}$,
 676 which proves the theorem. \square

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

681 6. *i*Routing

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

687 Within the realm of Mobile Edge Computing (MEC), devices of the cluster
 688 request services from the cluster-head (denoted by \mathcal{C}) imposing the need for estab-
 689 lishing an end-to-end path between the requestor (i.e. destination device denoted
 690 by \mathcal{R}_{qs}) and \mathcal{C} . Each time data must be delivered to \mathcal{R}_{qs} , \mathcal{C} has to compute the
 691 NDP by solving an MDG for this destination. To do this, following the route
 692 discovery, \mathcal{C} uses its latest information about the malware detection capabilities
 693 of all possible routes to \mathcal{R}_{qs} , along with their inspection costs (i.e. malware detec-
 694 tion costs to perform, for example, intrusion classification). Data is then relayed
 695 and collaboratively inspected by the devices on its way to \mathcal{R}_{qs} . Overall, the ob-
 696 jective of \mathcal{C} (i.e. the Defender) is to select the route that can correctly detect
 697 and filter out malicious data before they infect \mathcal{R}_{qs} by making sure that it is not
 698 crafted with malware. We assume that each device must use its data inspection
 699 capabilities at the maximum possible degree..

700 *i*Routing has characteristics of *reactive route selection protocols*, meaning that
 701 it takes action and starts computing routing paths that have not been previously
 702 computed when a request for data delivery to \mathcal{R}_{qs} is issued. *i*Routing requires to
 703 obtain information about the malware inspection capabilities and the associated
 704 computational cost of devices, in routes from \mathcal{C} to \mathcal{R}_{qs} .

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

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

Algorithm 1 Seeking routes to destination Rqs.

```
1: procedure iROUTING_REQUEST( $s, \text{Rqs}, \mathcal{S}_j$ )
2:    $s$  seeks routes to Rqs by broadcasting  $\text{RREQ}_{\text{Rqs}}$ ;
3:   if a device  $s_i$  receives  $\text{RREQ}_{\text{Rqs}}$  then
4:      $\mathcal{S}_j \cup \{s_i\}$ ;
5:     if  $s_i \neq \text{Rqs}$  then
6:        $s_i$  executes iROUTING_REQUEST( $s_i, \text{Rqs}, \mathcal{S}_j$ );
7:     else
8:        $L \leftarrow |\mathcal{S}_j|, n \leftarrow 0, \mathcal{T}_j \leftarrow \emptyset, \mathcal{C}_j \leftarrow \emptyset$ ;
9:       iROUTING_RESPONSE( $n, L, \mathcal{T}_j, \mathcal{C}_j, \mathcal{S}_j, \text{Rqs}$ );
10:      break;
11:    end if
12:  end if
13: end procedure
```

Algorithm 2 Responding to a cluster-head with a route to Rqs.

```
1: procedure iROUTING_RESPONSE( $n, L, \mathcal{T}_j, \mathcal{C}_j, \mathcal{S}_j, s$ )
2:    $s$  sends  $\text{RREP}_{\text{Rqs}}$  to the  $(L - n)$ -th device of  $\mathcal{S}_j$ , let it be  $s_i$ ;
3:   if  $s_i \neq \mathbf{C}$  then
4:      $\mathcal{T}_j \cup \mathbf{p}(s_i), \mathcal{C}_j \cup c(s_i), n \leftarrow n + 1$ ;
5:     iROUTING_RESPONSE( $n, L, \mathcal{T}_j, \mathcal{C}_j, \mathcal{S}_j, s_i$ );
6:   else
7:     Execute iROUTING( $\text{Rqs}, \mathbf{D}, \mathcal{S}_j, \mathcal{T}_j, \mathcal{C}_j$ );
8:     break;
9:   end if
10: end procedure
```

716 the set \mathcal{T}_j of vectors of “failing-to-detect” probabilities, for different malware,
717 of devices in r_j ; and (iii) the set \mathcal{C}_j of computational malware inspection costs
718 $c(s_i)$ of devices in r_j . These values are updated while the RREP_{Rqs} is traveling
719 back to \mathbf{C} . When each device (e.g. s_i) that is involved in the route response
720 phase, receives the RREP_{Rqs} , it updates \mathcal{T}_j and \mathcal{C}_j . Within the time period T_{req} , \mathbf{C}
721 aggregates RREP_{Rqs} messages and updates its routing table with information that
722 can be used to derive the *optimal routing strategy*, as dictated by Theorem 2.

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

Algorithm 3 Delivering data to Rqs.

```
1: procedure iROUTING(Rqs, D,  $\mathcal{S}_j, \mathcal{T}_j, \mathcal{C}_j$ )
2:   C derives the Nash Delivery Plan,  $\rho^{\text{NE}}$  using  $\mathcal{S}_j, \mathcal{T}_j, \mathcal{C}_j$ ;
3:   C chooses  $r^*$  probabilistically as dictated by  $\rho^{\text{NE}}$ ;
4:   C delivers D to Rqs over  $r^*$ ;
5:   Each device  $s_i \in r^*$  performs data inspection;
6:   if D found to carry malware then
7:      $s_i$  drops D;
8:      $s_i$  notifies C by sending a notification message along the reverse path;
9:     C blacklists the device that sent, through the cloud, D consisting of
malware;
10:  else
11:     $s_i$  forwards D to Rqs;
12:  end if
13: end procedure
```

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

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

745 The communications complexity of the *i*Routing protocol measured in terms
746 of number of messages exchanged in performing route discovery is $\mathcal{O}(2N)$, where
747 N is the number of devices in the D2D network. As a reactive routing protocol,
748 *i*Routing has higher storage complexity than conventional routing protocols, but
749 it supports multiple-path routing and QoS routing making malware detection
750 optimal, as shown in section 5. Finally, *i*Routing has a time complexity equal to
751 $\mathcal{O}(2D)$, 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

752 7. Simulations

753 7.1. Network setup

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

765 7.2. Security controls and malware

766 Simulations consider one adversary who is injecting a sequence of consecutive
767 malicious replies with the aim to infect Rqs. We assume that the Attacker chooses
768 to inject one of $[M] = \{\text{Keylogger, SMS spam, Rootkit iSAM, Spyware, iKee-B,}$
769 $\text{Premium-Rate calls}\}$ malware types (i.e. pure strategies of the Attacker). We
770 have also assumed the anti-malware controls, SMS Profiler, iDMA, iTL, and
771 Touchstroke, along with their detection rates, as published in [49]. Each mobile
772 device is equipped with at least one and up to three anti-malware controls.

773 7.3. Attackers

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

- 776 • *Uniform*: the Attacker chooses each malware type from the set with equal
777 probability. For example for the set we have used here, there is a probability
778 $\frac{1}{6} = 0.1667$ the Attacker to choose any of the malware types of $[M]$;

- 779 • *Weighted*: the Attacker chooses a malware type with probability derived
780 by the following algorithm:
- 781 1. find the average utility value of the Attacker for each column of the
782 game matrix;
 - 783 2. add the average utility values of the Attacker for all columns to get
784 the combined sum;
 - 785 3. for each malware type, derive the probability of a malware type to be
786 chosen by dividing its average utility value, found in step 1, by the
787 sum derived in step 2.
- 788 • *Nash*: the Attacker plays according to her Nash strategy μ^{NE} .

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

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

796 7.4. Experiments

797 We have considered 5 *Cases* each referring to different simulation times: 10,
798 20, 40, 60, and 120 mins. For each Case we have simulated 1,000 replies, which
799 are UDP messages of length 512 bytes with delay limit 100 seconds, for a fixed
800 network topology. Yet we refer to the run of the code for the pair $\langle \text{Case}, \# \text{replies} \rangle$
801 by the term *Experiment*. We have repeated each Experiment for 10 independent
802 network topologies to get a clear idea of the results' trend. We do that for all 5
803 Cases and each type of Attacker profile. Thus we simulate, in total: 5 Cases \times
804 1,000 replies \times 10 network topologies = 50,000 replies.

805 7.5. Comparisons

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

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

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

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

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

811 According to the results illustrated in Figures 2 - 4, *i*Routing consistently out-
812 performs the rest of the protocols, in terms of both Defender's *expected utility* and
813 *average detection rate*, for all different simulation times and Attacker types. The
814 results show that *i*Routing achieves its highest average malware detection rate
815 ($\sim 65\%$) against a Uniform Attacker (non-strategic Attacker), and its worst rate
816 against a Weighted Attacker. In the case of a Nash Attacker, *i*Routing has almost
817 22% higher detection rate than PR, 6% than DSR, while it is twice more efficient
818 (i.e. $\sim 11\%$) than AODV. For a Weighted Attacker, PR behaves differently as it
819 achieves approximately 6% lower average detection rate than *i*Routing, in con-
820 trast to DSR and AODV, which perform worse, as opposed to the Nash Attacker
821 case, since the difference of their average detection rate compared to *i*Routing
822 becomes double (i.e. $\sim 12\%$ for DSR and 24% for AODV). Finally, for a Uniform
823 Attacker, the difference, in terms of detection rate, compared to *i*Routing, is
824 almost the same for both DSR and PR, which is approximately equivalent to
825 8%. AODV still has the worst average detection rate among all protocols by
826 having 24% worse rate than *i*Routing.

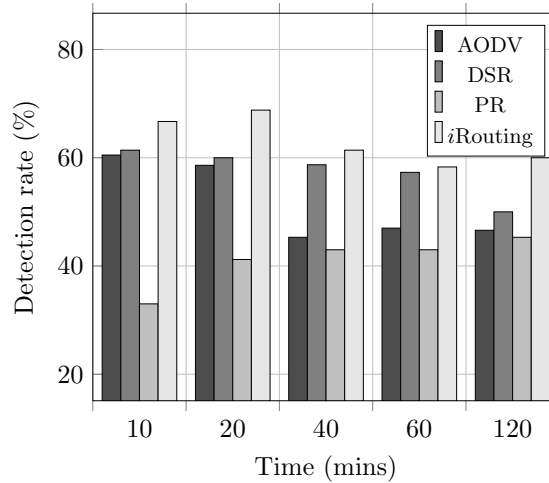


Figure 2: Malware detection rate in presence of a Nash attacker.

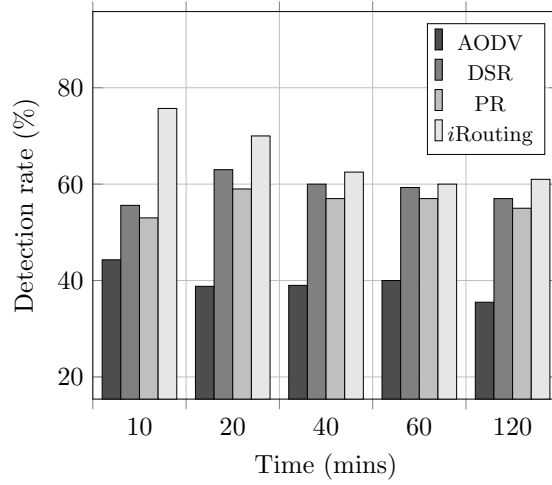


Figure 3: Malware detection rate in presence of a Uniform attacker.

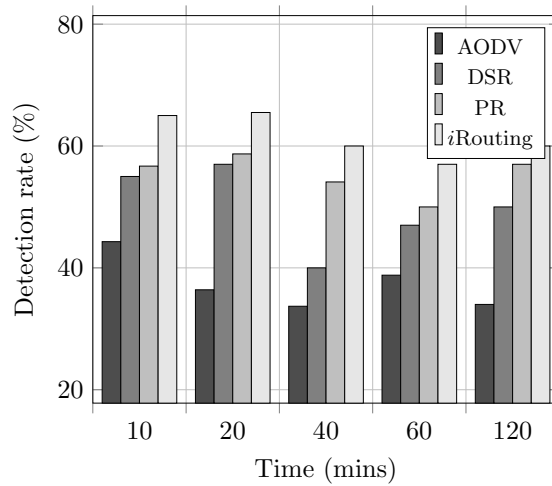


Figure 4: Malware detection rate in presence of a Weighted attacker.

827 According to Figures 5 - 7, *iRouting* achieves the best performance in terms of
 828 average expected utility among all protocols. More specifically, *iRouting* improves
 829 the average expected utility, in the case of a Nash Attacker, by, in average, 49%,
 830 17%, and 7% compared to PR, AODV, and DSR, respectively. We notice that
 831 the Defender's utility in *iRouting* is similar to the one achieved when DSR is
 832 used. The reason for this is that DSR improves computational cost as opposed
 833 to *iRouting* more than AODV and PR while exhibiting the best detection rate
 834 among AODV and PR. Average improvement values are slightly more pronounced

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

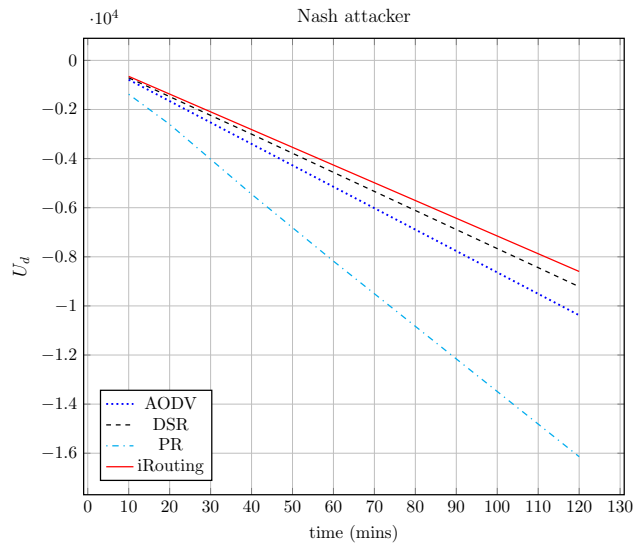


Figure 5: Utility of the Defender in presence of a Nash attacker.

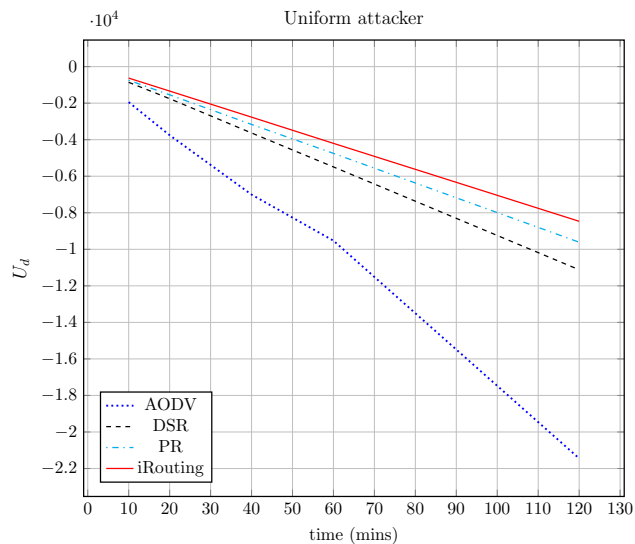


Figure 6: Utility of the Defender in presence of a Uniform attacker.

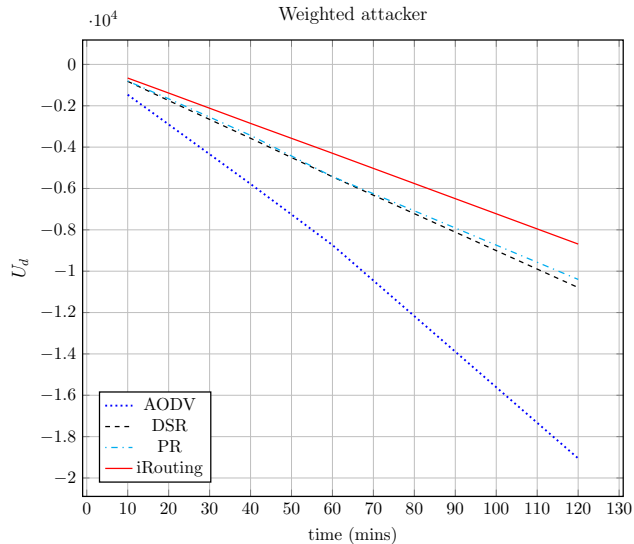


Figure 7: Utility of the Defender in presence of a Weighted attacker.

840 8. Conclusion

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

856 9. References

- 857 [1] D. Feng, L. Lu, Y. Yuan-Wu, G. Ye Li, S. Li, G. Feng, Device-to-device communications
 858 in cellular networks, *IEEE Commun. Mag.* 52 (4) (2014) 49–55.
 859 [2] H. Nishiyama, M. Ito, N. Kato, Relay-by-smartphone: realizing multihop device-to-device
 860 communications, *IEEE Commun. Mag.* 52 (4) (2014) 56–65.

- 861 [3] M. Tehrani, M. Uysal, H. Yanikomeroglu, Device-to-device communication in 5G cellular
862 networks: challenges, solutions, and future directions, *IEEE Commun. Mag.* 52 (5) (2014)
863 86–92.
- 864 [4] G. Fodor, E. Dahlman, G. Mildh, S. Parkvall, N. Reider, G. Miklos, Z. Turanyi, Design
865 aspects of network assisted device-to-device communications, *IEEE Commun. Mag.* 50 (3)
866 (2012) 170–177.
- 867 [5] K. Doppler, M. Rinne, C. Wijting, C. Ribeiro, K. Hugl, Device-to-device communication
868 as an underlay to LTE-advanced networks, *IEEE Commun. Mag.* 47 (12) (2009) 42–49.
- 869 [6] C. A. Ardagna, M. Conti, M. Leone, J. Stefa, An anonymous end-to-end communication
870 protocol for mobile cloud environments, *IEEE Trans. Serv. Comput.* 7 (3) (2014) 373–386.
- 871 [7] C. Perkins, E. Belding-Royer, S. Das, Ad hoc on-demand distance vector (AODV) routing,
872 RFC 3561 (Jul. 2003).
- 873 [8] D. Johnson, Y. Hu, D. Maltz, The Dynamic Source Routing protocol (DSR) for mobile ad
874 hoc networks for IPv4, RFC 4728 (Feb. 2007).
- 875 [9] T. Clausen, P. Jacquet, Optimized Link State Routing Protocol (OLSR), RFC 3626 (Oct.
876 2003).
- 877 [10] T. Ramrekha, E. Panaousis, C. Politis, Standardisation advancements in the area of routing
878 for mobile ad-hoc networks, *J. of Supercomputing* 64 (2) (2013) 409–434.
- 879 [11] G. Suarez-Tangil, J. E. Tapiador, P. Peris-Lopez, A. Ribagorda, Evolution, detection and
880 analysis of malware for smart devices, *IEEE Communications Surveys Tutorials* 16 (2).
- 881 [12] M. Khouzani, S. Saswati, E. Altman, Maximum damage malware attack in mobile wireless
882 networks, *IEEE/ACM Trans. Netw.* 20 (5) (2012) 1347–1360.
- 883 [13] R. Heartfield, G. Loukas, A taxonomy of attacks and a survey of defence mechanisms for
884 semantic social engineering attacks, *ACM Computing Surveys (CSUR)* 48 (3) (2016) 37.
- 885 [14] M. La Polla, F. Martinelli, D. Sgandurra, A survey on security for mobile devices, *IEEE*
886 *Commun. Surveys Tuts.* 15 (1) (2012) 446–471.
- 887 [15] T. Alpcan, T. Basar, *Network security: a decision and game-theoretic approach*, Cambridge
888 University Press, 2010.
- 889 [16] M. Naserian, K. Tepe, Game theoretic approach in routing protocol for wireless ad hoc
890 networks, *Ad Hoc Netw.* 7 (3) (2009) 569 – 578.
- 891 [17] Y. Xiao, K.-C. Chen, C. Yuen, Z. Han, L. A. DaSilva, A bayesian overlapping coalition
892 formation game for device-to-device spectrum sharing in cellular networks, *IEEE Transactions*
893 *on Wireless Communications* 14 (7) (2015) 4034–4051.
- 894 [18] C. Long, Q. Chi, X. Guan, T. Chen, Joint random access and power control game in ad
895 hoc networks with noncooperative users, *Ad Hoc Netw.* 9 (2) (2011) 142–151.
- 896 [19] F. Wang, O. Younis, M. Krunz, Throughput-oriented mac for mobile ad hoc networks: A
897 game-theoretic approach, *Ad Hoc Netw.* 7 (1) (2009) 98 – 117.
- 898 [20] Y. Jianting, M. Chuan, Y. Hui, Z. Wei, Secrecy-based access control for device-to-device
899 communication underlying cellular networks, *IEEE Commun. Mag.* 17 (11) (2013) 2068–
900 2071.
- 901 [21] Z. Daohua, A. Swindlehurst, S. Fakoorian, X. Wei, Z. Chunming, Device-to-device com-
902 munications: The physical layer security advantage, *IEEE Int. Conf. on Acoust., Speech,*
903 *Signal Process.* (2014) 1606–1610.
- 904 [22] L. Abusalah, A. Khokhar, M. Guizani, A survey of secure mobile ad hoc routing protocols,
905 *IEEE Commun. Surveys Tuts.* 10 (4) (2008) 78–93.
- 906 [23] S. Gupte, M. Singhal, Secure routing in mobile wireless ad hoc networks, *Ad Hoc Netw.*
907 1 (1) (2003) 151–174.
- 908 [24] E. Panaousis, T. Alpcan, H. Fereidooni, M. Conti, Secure message delivery games for
909 device-to-device communications, in: R. Poovendran, W. Saad (Eds.), *Decision and Game*
910 *Theory for Security*, Vol. 8840 of *Lecture Notes in Computer Science*, Springer International
911 Publishing, 2014, pp. 195–215.

- 912 [25] A. Patcha, J. M. Park, A game theoretic approach to modeling intrusion detection in mobile
913 ad hoc networks, in: Proc. 5th Annu. SMC Information Assurance Workshop, 2004, pp.
914 280–284.
- 915 [26] Y. Liu, C. Comaniciu, H. Man, A bayesian game approach for intrusion detection in
916 wireless ad hoc networks, in: Proc. 2006 workshop on Game Theory for Communications
917 and Networks, 2006, pp. 1–12.
- 918 [27] Y. Liu, C. Comaniciu, H. Man, Modelling misbehaviour in ad hoc networks: a game
919 theoretic approach for intrusion detection, *Int. J. of Security and Netw.* 1 (7) (2006) 243–
920 254.
- 921 [28] N. Marchang, R. Tripathi, A game theoretical approach for efficient deployment of intrusion
922 detection system in mobile ad hoc networks, in: Proc. 2007 Int. Conf. on Advanced
923 Computing and Communications, 2007, pp. 460–464.
- 924 [29] H. Otrok, M. Debbabi, C. Assi, P. Bhattacharya, A cooperative approach for analyzing
925 intrusions in mobile ad hoc networks, in: Proc. 27th Int. Conf. on Distributed Computing
926 Systems Workshops, 2009, pp. 985–992.
- 927 [30] N. Santosh, R. Saranyan, K. Senthil, V. Vetrivel, Cluster based co-operative game theory
928 approach for intrusion detection in mobile ad-hoc grid, in: Proc. of the International
929 Conference on Advanced Computing and Communications (ADCOM), 2008, pp. 273–278.
- 930 [31] J. Cho, I. Chen, P. Feng, Effect of intrusion detection on reliability of mission-oriented
931 mobile group systems in mobile ad hoc networks, *IEEE Trans. Rel.* 59 (1) (2010) 231–241.
- 932 [32] M. Felegyhazi, L. Buttyan, J. Hubaux, Nash equilibria of packet forwarding strategies in
933 wireless ad hoc networks, *IEEE Trans. Mobile Comput.* 5 (5) (2006) 463–476.
- 934 [33] W. Yu, K. Liu, Game theoretic analysis of cooperation stimulation and security in au-
935 tonomous mobile ad hoc networks, *IEEE Trans. Mobile Comput.* 6 (5) (2007) 507–521.
- 936 [34] W. Yu, Z. Ji, K. Liu, Securing cooperative ad-hoc networks under noise and imperfect
937 monitoring: strategies and game theoretic analysis, *IEEE Trans. Inf. Forensics Security*
938 2 (2) (2007) 240–253.
- 939 [35] W. Yu, K. Liu, Secure cooperation in autonomous mobile ad-hoc networks under noise and
940 imperfect monitoring: a game-theoretic approach, *IEEE Trans. Inf. Forensics Security* 3 (2)
941 (2008) 317–330.
- 942 [36] E. Panaousis, C. Politis, A game theoretic approach for securing AODV in emergency
943 mobile ad hoc networks, in: Proc. 34th IEEE Conf. on Local Computer Networks, 2009,
944 pp. 985–992.
- 945 [37] D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, M. Tambe, Stackelberg vs. Nash in secu-
946 rity games: An extended investigation of interchangeability, equivalence, and uniqueness,
947 *J. Artif. Intell. Res.* 41 (2011) 297–327.
- 948 [38] M. Tambe, *Security and game theory: algorithms, deployed systems, lessons learned*, Cam-
949 bridge University Press, 2011.
- 950 [39] A. Wang, Y. Cai, W. Yang, Z. Hou, A Stackelberg security game with cooperative jamming
951 over a multiuser OFDMA network, in: Proc. 2013 IEEE Wireless Communications and
952 Networking Conference, 2015, pp. 4169–4174.
- 953 [40] D. Kar, F. Fang, F. Delle Fave, N. Sintov, M. Tambe, A Game of Thrones: when human be-
954 havior models compete in repeated stackelberg security games, in: Proc. 2015 International
955 Conference on Autonomous Agents and Multiagent Systems, 2015, pp. 1381–1390.
- 956 [41] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the Internet of
957 Things, in: Proc. 1st MCC Workshop on Mobile Cloud computing, 2012, pp. 13–16.
- 958 [42] A. Asadi, Q. Wang, V. Mancuso, A survey on device-to-device communication in cellular
959 networks, *Communications Surveys & Tutorials*, *IEEE* 16 (4) (2014) 1801–1819.
- 960 [43] D. Dolev, A. Yao, On the security of public key protocols, *IEEE Trans. Inf. Theory* 29 (2)
961 (1983) 198–208.
- 962 [44] M. J. Osborne, A. Rubinstein, *A course in game theory*, MIT press, 1994.

- 963 [45] C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordóñez, M. Tambe, Computing optimal
964 randomized resource allocations for massive security games, in: Proceedings of The 8th
965 International Conference on Autonomous Agents and Multiagent Systems-Volume 1, Inter-
966 national Foundation for Autonomous Agents and Multiagent Systems, 2009, pp. 689–696.
- 967 [46] J. Von Neumann, O. Morgenstern, Theory of games and economic behavior (60th anniver-
968 sary commemorative edition), Princeton university press, 2007.
- 969 [47] J. Nash, Equilibrium points in n-person games., in: Proc. of the National Academy of
970 Sciences, 1950, pp. 48–49.
- 971 [48] T. Basar, G. J. Olsder, Dynamic noncooperative game theory, London Academic press,
972 1995.
- 973 [49] D. Damopoulos, G. Kambourakis, G. Portokalidis, The best of both worlds: a framework
974 for the synergistic operation of host and cloud anomaly-based ids for smartphones, Proc.
975 7th European Workshop on System Security.