An efficient cyber defense framework for UAV-Edge computing network

Hichem Sedjelmaci, Aymen Boudguiga, Inès Ben Jemaa, Sidi Mohammed Senouci

An efficient cyber defense framework for UAV-Edge computing network

Hichem Sedjelmaci [a,*], Aymen Boudguiga [b], Inès Ben Jemaa [c], Sidi Mohammed Senouci [d]

[a] Orange Labs, 4 Avenue de la République, Châtillon 92320, France
[b] CEA, 8 Avenue de la Vauve, Palaiseau 91120, France
[c] IRT SystemX, 8 Avenue de la Vauve, Palaiseau 91120, France
[d] DRIVE EA 1859, Univ. Bourgogne Franche Comté, Nevers F58000, France

Mobile Edge Computing (MEC) is usually deployed in energy and delay constrained networks, such as internet of things networks and transportation systems to address the issues of energy consumption, computation capacity and network delay. In this work, we focus on a special case, which is Unmanned Aerial Vehicle Edge Computing (UEC) network. Addressing the security issues in UAV-Edge Computing network is mandatory due to the criticality of UEC services, such as network traffic monitoring, or search and rescue operations. However, cyber defense and protection of UEC network have not yet received sufficient research attention. Thereby, we propose and develop a cyber-defense solution based on a non-cooperative game to protect the UEC from network and offloading attacks, while taking into account nodes' energy constraints and computation overhead. Simulation results show that, the deployment of our cyber defense system in UEC network requires low energy consumption and low computation overhead to obtain a high protection rate.

1. Introduction

An Unmanned Aerial Vehicle (UAV) is a mobile aerial node that relies on two mobility modes: controlled or autonomous. In the controlled mode, a human pilot controls remotely the mobility of the UAV. When the quality of the link between a controller and the UAV is degraded due to signal loss for instance, the UAV switches from the controlled mode to an autopilot or autonomous mode [1,2]. In the autonomous mode, the UAV relies on a GPS module to determine the optimal path to reach the requested destination. UAV networks, formed by a fleet of drones, are mainly deployed in hostile and inaccessible areas, where classical and wired infrastructures cannot be deployed [3]. These networks serve to monitor, explore, collect and analyze information about these areas and report them to a remote infrastructure for further analysis. UAVs networks have mainly civilian and military applications. For example, drone networks can serve for parcel delivery, traffic monitoring, target tracking, etc.

One of the main challenges of UAVs are their energy consumption and computation overhead [4,5]. Indeed, collecting and analyzing a huge amount of data is an energy-consuming task for drones. Fortunately, the Mobile Edge Computing (MEC) is an emerging solution for mitigating energy and computation issues for mobile nodes and in particular drones. MEC appeared in 2014 as an alternative technology for cloud computing [6]. When a mobile node (e.g., UAV) notices that the local processing of the collected data increases its energy consumption or computation overhead, it delegates the data processing to an edge computing server. Data processing delegation is known as offloading. The main advantage of computation offloading to a MEC server is reducing the network latency [7]. The network architecture combining UAV network and MEC technology is defined as an UAV-Edge Computing (UEC) network [4,8].

Cybersecurity of UEC network has not yet received considerable research attention. However, a cyber-attack on this system could significantly degrade its performance and effectiveness. In this work, we investigate the detection of two major attacks on UEC: the offloading attack [9] and the Denial of Service attack (DoS). The offloading attack targets the quality of the link between an UAV node and an edge node. It aims at increasing the network latency by dropping the offloading data, and then forcing legitimate UAVs to retransmit their packets. Attackers may carry out an offloading attack just by spoofing or jamming the communication. DoS attacks [10] targets the energy resources and alters the critical information collected by UAVs.

* Corresponding author.
E-mail address: hichem.sedjelmaci@orange.com (H. Sedjelmaci).

In this work, we propose a cybersecurity game based on a Stackelberg approach to protect the UEC network from the aforementioned attacks. We model the problem as a Stackelberg game since this strategic game is used in a scenario where the players are competitors as in the cyber security context (attacker and defender). The proposed Stackelberg game has two kinds of players: a security agent and an attacker. They play the role of a leader and a follower, respectively. The security agent is installed at each UAV. Each player has a set of pure and mixed strategies (i.e., possible actions). The mixed strategies represent a probability distribution over pure strategies [11]. The interaction between the competitor players leverages players' payoffs, which increase and decrease with respect to the executed strategies. The followers monitor the leaders' optimal strategy and they replicate by executing their best strategies while taking into account the actions envisaged by the security agents. The objective of the leader and of the follower players is to increase their payoffs and to decrease the payoffs of their opponents. The decision of the security agents for the monitoring, the detection and the reaction to the suspected attacks on offloading links and to the suspected UAVs depends mainly on the energy constraints of UAVs and on the computation overhead. The optimal decision corresponds to the Stackelberg Equilibrium (SE). To the best of our knowledge, our proposed solution is the first intrusion detection framework adapted to UEC networks. It takes into account the energy consumption and the computation overhead during the detection and the decision-making operation by the security agents.

The remainder of this paper is organized as follows. In Section 2, we describe the considered network architecture. In Section 3, we summarize the most notable research works by highlighting their advantages and shortcomings. Section 4 presents the proposed hierarchical security framework for UEC network by detailing the game mathematical model. Simulation results and performance analysis are provided in Section 5. Finally, in Section 6, we present our conclusion and some future works.

2. UAV-Edge computing network model

We consider a UAV network in which each UAV has a set of missions to accomplish. For instance, a UAV may collect information about its environment and send them to a server in the cloud via a communication infrastructure. We consider a heterogonous architecture formed by UAVs and a wireless infrastructure network composed by Road Side Unit (RSU), eNodeB and MEC server as show in Fig. 1.

The UAV sends the collected information to the server whenever an infrastructure network is available i.e., in case a UAV has a connectivity with an eNodeB or a RSU. The UAV monitors continuously the network and once it detects a suspicious behavior, it transmits the information to its neighbors. All UAVs maintain a reputation metric regarding their neighbors. Each UAV can increase or decrease the reputation metric of its monitored neighbors with respect to their behaviors (normal or bad).

As a UAV has limited computation and energy resources, it may not be able to process its neighbor's data to detect suspicious nodes in the network. Fortunately, it can entrust the detection task either to other UAVs with no energy shortage or to an edge server with abundant computation resources.

A group of UAVs is connected through wireless media. Each UAV broadcasts beacons to inform its neighbors about its presence in the network. Each drone stores a set of information about its neighbors such as their position, their velocity, the remaining energy and the computation power. Each drone monitors the network by performing attack detection locally. If the drone does not have enough resources or suspects an attack, it has to offload the computation (detection) either to the UEC node or to the edge server.

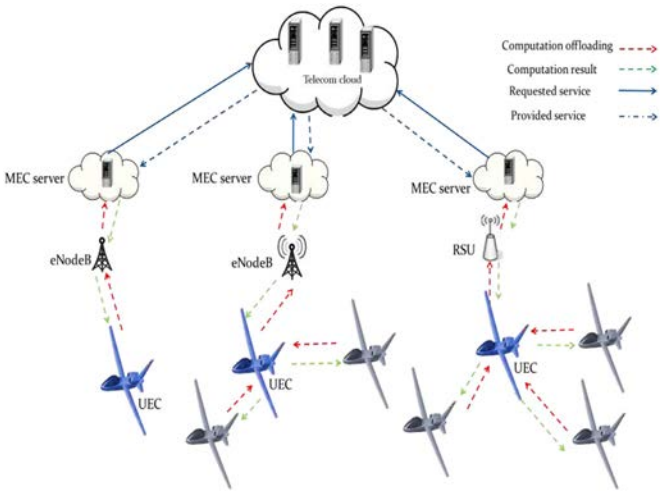


Fig. 1. UEC architecture.

The latter will be in charge of attacks detection as presented in Fig. 1. Sections 4.1 and 4.3 provide more details about the local and the offloading detections. The selection of an UAV as an UEC node, which can perform the detection task on behalf of other nodes, depends on its remaining energy, its distance to the edge server and its reputation. The UAV node that exhibits a good balance between energy, distance and reputation is selected as UEC node. The process of UEC node selection is repeated periodically during the network lifetime. UAVs have limited calculation resources. To decide whether to offload (to an UEC node or an edge server) a calculation for suspicious nodes detection, an UAV estimates the amount of the consumed energy and the computation overhead as explained in Section 4. Notice that when the UEC is not in the coverage of the eNodeB or the RSU, it has to store the packet until getting a connection to the edge server.

3. Related work

MEC technology improves the quality of service by reducing the latency in mobile networks [12]. Cyber security of MEC architecture is vital due to the critical data that mobile nodes offload to the edge servers. In [9], the authors aimed to secure the MEC architecture from the attackers that target the offloading link between the user equipment and the edge server. They relied on a non-cooperative game to study the interaction between the monitoring agents, the attackers and the user equipment. The equilibrium is reached when the monitoring agents detect the attackers with a high accuracy and when the user equipment ensures efficient offloading. The authors in [13] proposed a reconfigurable security framework for low resources Internet of Things (IoT) devices based on MEC architecture. The proposed security solution is based on a lightweight cryptography technique to ensure the privacy of the communication and the mobile node authentication. Their security solution exhibits a low computation overhead and is suitable to any heterogeneous IoT network. However, the major drawbacks of works [9] and [13] are their inability to detect the internal attacks that aware of the cryptography keys.

In [8,14–18], the authors tried to protect the intelligent transportation systems based on MEC architecture from offloading attacks resulting from jamming, GPS spoofing and DoS attacks targeting the edge links. In particular, the authors in [8] secured the information leakage of UAV-aided vehicular network. They designed a hierarchical edge computing architecture to address the UAV's energy consumption and computation overhead issues. Then, they developed a cyber-threat detection solution based on bloom filter to protect the information leakage from hackers that drop

packets or inject fake data (sent from vehicle to UAV and from UAV to edge server). In their simulation, they proved that the authentication, encryption and decryption require less computation overhead and storage to thwart the external threats. However, they did not take into account internal threats coming from malicious insiders, i.e. UAVs and vehicles. A cooperative cybersecurity framework for vehicular cloud computing is proposed in [14]. The authors presented a survey of vehicular cloud computing and the related applications, by highlighting the cyber security issues that this network encounters. They developed a cooperative security game to protect the heterogeneous and dynamic vehicles from the lethal attacks that target the cloud infrastructure (e.g., attacks targeting road safety and traffic management). However, the authors did not share key indicators about their simulated framework such as detection and false positive rates. In [15], a distributed reputation approach is proposed to protect the vehicular edge computing network against malicious behaviors. To obtain an accurate reputation history of each monitored vehicle, the authors run locally the reputation management at each edge-computing server. This approach can detect the misbehaving vehicles with a high accuracy. However, it does not allow detecting the other threats that target the vehicular edge computing such as DoS and offloading attacks. In [16], the authors developed an efficient and lightweight messages authentication scheme for vehicular edge computing network. To reduce the latency, the roadside unit plays the role of edge server and authenticates the messages transmitted by vehicles within its radio range, then disseminates the identities of the authenticated vehicles to other legitimate nodes (vehicles and roadside units). According to the simulation results, the proposed security scheme identifies promptly the valid and fake messages when the vehicular network is under attack. However, this solution is not scalable as it incurs a high communication overhead in a large scale network. In [17], the authors protect the UAVs network against jamming and spoofing attacks. These cyber-attacks aim mainly to hack transmission and offloading links. To protect the network from these attacks, the authors develop a deep reinforcement learning approach that uses a UAV power location as a main feature during the learning and detection process. The main issue of this work is that the energy constraint is not considered during the learning and the detection process. A cyber security framework based on cooperative and non-cooperative games is developed in [18] to protect the vehicular edge computing network from offloading attacks. The main idea of this work is to maximize the attack detection rate and minimize the latency when cyber-attacks occur. The security framework is based on a behavior game to study the interaction between three kinds of players: detection agents, vehicles and attackers. Then, it determines the optimal payoffs of each player. The simulation exhibits interesting results in terms of classification rate and delays.

Recently, other intrusion detection and prediction frameworks dedicated to secure the UAVs network were developed in [1,19–21]. These security frameworks present an interesting rate of attacks detection when taking into account the constraints related to network and drones. However, they all fail to detect the offloading attacks.

In this research work, we address the weakness mentioned above and propose a new cyber defense framework for UAV-Edge computing network for offloading and DoS attacks detection and prevention, while taking into account the energy consumption and the computation overhead of UAVs.

4. Hierarchical cyber security framework for UAV-Edge computing

In this section, we present an overview of the proposed cyber security game of UEC network. Afterward, we describe the considered attack model and finally we present the security solution based on a Stackelberg approach.

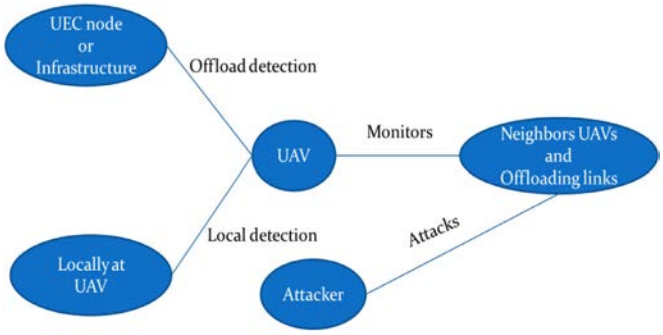


Fig. 2. Illustration of attack detection process in UEC network.

4.1. Cyber security game overview

The proposed security game involves two kinds of players, security agents (i.e., UAVs) and attackers. Security agents will protect the UAV-Edge computing network from attackers. Meanwhile, attackers will target energy resources, vital information of UAVs and the edge links of the network.

Security agents are installed at each UAV. They monitor the edge links and neighboring UAVs. When an attack is suspected, the security agent performs either a local or an offloading detection as shown in Fig. 2. During this monitoring process, we rely on rule-based detection, which is a lightweight detection approach since it generates a low energy consumption and low computation overhead. The local and offloaded detection (either to other UAVs or to Edge nodes) rely on a machine learning detection technique such as reinforcement learning or supervised learning algorithms [22,17]. This technique is accurate for attack detection, but it is heavy with respect to energy consumption and computation overhead. Thereby, the local or offloading detection approaches is launched only when the offloading links or/and UAV nodes are suspected to be attacked. For more details about rule-based detection and machine learning detection techniques applied for UAV networks, we refer the readers to [17,19,21,23].

We propose a zero-sum game based on a Stackelberg methodology to model the behavior game between the security agents that are installed at UAVs and the attackers that target the UEC network. In the Stackelberg game, we have two types of non-cooperative players: the leaders that are the security agents, and the followers that are the attackers. In this non-cooperative game, the security agents aim to protect the UAVs and all the communication links from attackers, namely UAV-to-UAV communications (U2U) and UAV-to Infrastructure communication (U2I).

During the monitoring, the security agents analyze the energy constraints of UAVs and the computation overhead of the network, and fetch the security breaches of UEC network. The followers probe the actions of the leader agents and perform malicious attacks with the goal of being undetected by the security agents. Here, the attackers will try to put out of service the UAVs network by flooding them with malicious messages. They can also target the link quality by dropping the packets orjamming and spoofing the communication.

The leader agents cooperate to protect the UAVs-Edge computing network. Meanwhile attackers are organized into groups to carry out massive cyber-attacks. The result of this behavior game is to achieve equilibrium, defined as a Stackelberg Equilibrium (SE). The equilibrium corresponds exactly to the strategy that the leader wishes, i.e., attack and monitor (detect) the same link and UAV node.

4.2. Attack model

In this work, we consider a Dolev and Yao attacker model [24] where the attacker is able to *Read, Drop* and *Send* valid messages. A *Read* action refers to receiving or intercepting messages. Meanwhile, a *Send* action refers to forging, injecting and replaying messages. A *Drop* action refers to message filtering or components isolation. The attacker can assail the UAV-Edge network either by targeting communication protocols, or by attacking the system hardware and software components. In this work, we are interested in the following class of attacks:

- *Offloading* refers to attacks that target the link between an UAV and an Edge server. These attacks are carried out by active hackers that will read and drop valid messages in the edge link. The attacker can also attempt jamming the radio link between the targeted UAV and the Edge server. In [9], the authors studied the security against the offloading attacks via eavesdropping or jamming,
- *DoS* refers to the class of attack aiming at putting an UAV node and the network out of service. An attacker can flood a targeted UAV with false information to exhaust its computation resources and power. It can run a GPS jamming attack against an UAV in autonomous mode and make the UAV waste its computation and power resources in finding a new path with noisy GPS data. Indeed, the attacker spam GPS signals when the Coarse/Acquisition codes are broadcasted from satellites [23]. DoS attacks against drones have been investigated in practice. In [25], authors showed how they compromised the AR.drone 2.0 using tools such as HPing. They reduced drastically the frame analysis rate of the attacked drone performance. In [26], authors studied deauthentication flood attacks impact on a UAV network.

In the following, we describe the attacks detection with respect to the consumed energy, *E*, and the computation overhead, *T*. *E* and *T* will be used for the attacker and security agent payoffs computation. The number of considered attackers will not exceed 30% of UAVs number.

4.3. Stackelberg game-based security framework

The game is divided into I stages and t denotes the time that the players spend during the interaction game at each stage. The Stackelberg security state at stage i is modeled as $\varphi_{i,t}=\{{\varphi^1}_{i,\,t},\ {\varphi^2}_{i,\,t},\ 1 \le i \le I\}$, where ${\varphi^1}_{i,\,t}$ refers to the UEC network as monitored by the leader (i.e. security agent) and ${\varphi^2}_{i,\,t}$ refers to the UEC network as monitored by the follower (i.e. attacker). We model ${\varphi^1}_{i,\,t}$ by $(\{{L^1}_k(t),\ 1 \le k \le K\},\ \{{N^1}_d(t),\ 1 \le d \le D\})$, where ${L^1}_k(t)$ represents the offloading links that a security agent can monitor. Meanwhile, ${N^1}_d(t)$ counts the neighbors of the considered security agent. K and D are the maximum number of offloading links and UAVs that security agent can monitor, respectively.

In addition, we set ${\varphi^2}_{i,\,t}= (\{{L^2}_k(t),\ 1 \le k \le K\},\ \{{N^2}_d(t),\ 1 \le d \le D\})$, where ${L^2}_k(t)$ and ${N^2}_d(t)$ are the offloading links and UAVs that could be targeted by attackers.

The pure strategies of a leader and a follower at stage i, ${S^1}_{i,\,t}$ and ${S^2}_{i,\,t}$ are expressed as ${S^1}_{i,\,t}({\varphi^1}_{i,\,t})=(\{{a^1}_{k,\,i} \in {L^1}_k(t)\}\cup\{{a^1}_{d,\,i} \in {N^1}_d(t)\})$ and ${S^2}_{i,\,t}({\varphi^2}_{i,\,t})=(\{{a^2}_{k,\,i} \in {L^2}_k(t)\}\cup\{{a^2}_{d,\,i} \in {N^2}_d(t)\})$. Action ${a^1}_{k,\,i}$ corresponds to protecting k suspected offloading links, while action ${a^1}_{d,\,i}$ corresponds to protecting d suspected UAVs. In addition, actions ${a^2}_{k,\,i}$ and ${a^2}_{d,\,i}$ corresponds to attacking k offloading links and d UAVs, respectively. Let, ${p^1}_k$ and ${p^1}_d$ be the probabilities of the leader to adopt respectively the strategies ${a^1}_{k,\,i}$ and ${a^1}_{d,\,i}$. ${p^2}_k$ and ${p^2}_d$ are the probability of the follower to adopt respectively the strategies ${a^2}_{k,\,i}$ and ${a^2}_{d,\,i}$. Here, we have $\sum_{k=1}^{K} {p^1}_k=1$, $\sum_{d=1}^{D} {p^1}_d=1$, $\sum_{k=1}^{K} {p^2}_k=1$ and $\sum_{d=1}^{D} {p^2}_d=1$.

4.3.1. Payoff

The payoffs of a leader and a follower are expressed in terms of reward and cost, i.e., payoff is equal to reward-cost. The leader's reward at each stage i depends on the number of offloading links and suspected UAVs that are monitored and protected by the security agent when attackers target them. That is, it depends on the detection rate. The leader's cost depends on the required computation overhead and energy consumption at UAV node to achieve a high correct detection rate. The follower's reward at each stage i depends on the number of infected offloading links and legitimate UAVs within attacker's neighborhood. Meanwhile, the follower's cost is the unsuccessful attack rate due to the correct detection of security agents.

4.3.1.1. Leader's payoff. As indicated in Section 4.1, during the detection process, the security agent performs either local computing or offloads the detection task to the remote UEC node or to the infrastructure. Each detection task j is represented by (M_j,T_j), where $1 \le j \le J$. M_j is the size of message delivered by rules based detection during the monitoring process, which contains a set of features related to a monitored target. T_j is the computation overhead or the required time for the machine learning algorithm to classify the suspected UAV or communication link as infected. T_j is required time for a leader to react against the cyber-attack.

The security agent estimates the energy consumption and T_j during the local and offloading detection and then takes its optimal choice. Inspired by the work [27], the energy consumption and required detection time during the local and offloading detection process, $({T^{Local}}_j,\ {E^{Local}}_j)$ and $({T^{Offload}}_j,\ {E^{Offload}}_j)$ are expressed as follows:

$${T^{Local}}_j = {D^{Local}}_j / {F^{Local}}_j \tag{1}$$

$${E^{Local}}_j = {D^{Local}}_j * {e^{Local}}_j \tag{2}$$

Where, ${D^{Local}}_j$ is the number of detection cycles required by UAV to identify an attack, ${F^{Local}}_j$ is the local CPU frequency and ${e^{Local}}_j$ corresponds to a local energy consumed per CPU cycle at UAV level.

$${T^{Offload}}_j = {D^{Offload}}_j / {F^{offload}}_j + {T^{link}}_j \tag{3}$$

Where, ${D^{Offload}}_j$ is the number of detection cycles required by an UEC node (or infrastructure) to categorize the suspected target as malicious or normal. Meanwhile, ${F^{offload}}_j$ is the frequency of infrastructure CPU or UEC node CPU, ${T^{link}}_j$ is the time elapsed between the task transmission time to the infrastructure (or UEC node) and the time of the reception of the decision result (i.e., monitored target is normal or malicious) at the UAV.

$${E^{Offload}}_j = {D^{Offload}}_j * ({e^{Offload}}_j + {e^{UEC}}_j) \tag{4}$$

Where, ${e^{Offload}}_j$ is the required energy to send the message with size M_j to infrastructure or UEC node through the cellular network or WIFI . The UAV is launched from some deployment areas to perform a couple of missions. They return to these departure points to recharge their batteries when their remaining energy is not sufficient to continue the mission. The energy of returning back to one of departure points is defined as e^{travel}. Here, ${e^{Offload}}_j$ is equal to the total energy of an UAV (before performing a mission) minus e^{travel}. ${e^{UEC}}_j$ corresponds to a local energy consumed per CPU cycle at UEC level. Note that we assume the infrastructure has no energy constraints and hence ${e^{infrastructure}}_j$ is equal to zero.

The cost function of a leader player is computed as the combination between the UAV's energy consumption and the leader's required time to react against the cyber-attack. The costs of local

detection and offloading detection are expressed as:

$$Cost^{Local} = \frac{\sum_{j=1}^{J}(\alpha_1 * T^{Local}{}_j + \beta_1 * E^{Local}{}_j)}{J}$$
$$Cost^{Offload} = \frac{\sum_{j=1}^{J}(\alpha_2 * T^{Offload}{}_j + \beta_2 * E^{Offload}{}_j)}{J} \quad (5)$$
$$Cost^1 = Cost^{Local} + Cost^{Offload}$$

Where, α_1, β_1, α_2, $\beta_2 \in [0,1]$ are weight parameters for computation overhead and energy consumption. Their values are flexibles and depend on security and network requirements. For instance, when the energy consumption of an UAV is high, the values of $\beta_1$and β_2 are increased. Whereas, when the lethal attacks against the UEC network are increased, $\alpha_1$and α_2 are decreased to react promptly against the attackers. In our mathematical model, Cost^1 $\in [0,1]$ describes the probability cost that a leader agent require to achieve a high level of protection.

The reward of security agent depends on the correct detection rate of infected edge links and attacked UAVs. The leader player's reward is expressed as:

$$Reward_j{}^1 = \alpha_3 . \frac{\sum_{k=1}^{K} D_k}{K} + \beta_3 . \frac{\sum_{d=1}^{D} D_d}{D},$$
$$Reward^1 = \frac{\sum_{j=1}^{J} Reward_j{}^1}{J} \quad (6)$$

Where $\frac{\sum_{k=1}^{K} D_k}{K}$ and $\frac{\sum_{d=1}^{D} D_d}{D}$ are the correct detection rates of infected offloading links and malicious UAVs, respectively. $Reward^1 \in [0,1]$ and α_3, $\beta_3 \in [0,1]$ are the weight parameters of the correct detection. The payoff function $\Psi_{i,t}^1$ of a leader agent is modeled by Eq. (7).

$$\Psi_{i,t}^1\left(S^1{}_{i,t}, S^2{}_{i,t}, \phi^1{}_{i,t}\right) = Reward_{i,t}{}^1\left(S^1{}_{i,t}, S^2{}_{i,t}, \phi^1{}_{i,t}\right) - Cost_{i,t}{}^1\left(S^1{}_{i,t}, S^2{}_{i,t}, \phi^1{}_{i,t}\right) \quad (7)$$

As shown in Eq. (8), the purpose of a leader player is to find the probabilities $p^1{}_k$ and $p^1{}_d$ of the optimal pure strategies $a^1{}_{k,i}$ and $a^1{}_{d,i}$ that maximize its payoff.

$$\max_{p^1{}_k, p^1{}_d} \sum_{k=1}^{K}\sum_{d=1}^{D} p^1{}_k * p^1{}_d * \Psi_{i,t}^1\left(S^1{}_{i,t}, S^2{}_{i,t}, \phi^1{}_{i,t}\right) \quad (8)$$

4.3.1.2. Follower's payoff. The main goal of attackers in UEC network is to maximize the number of attacked legitimate targets. They target the computation, communication and detection process by forcing the legitimate UAVs to exhaust their energies E_j^{attack} and so increase the computation overhead. The follower's reward is expressed as:

$$Reward^2 = \alpha_4 . \left(\sum_{k=1}^{K} T_k{}^{attack} + \sum_{d=1}^{D} T_d{}^{attack}\right) + \beta_4 . \sum_{k=1}^{K} E_k{}^{attack} \quad (9)$$

$T_{k,d}{}^{attack}$ is the computation overhead that UAV node (i.e., security agent) requires to protect itself and its neighboring UAVs from the follower. $E_k{}^{attack}$ is the energy consumption of an UAV targeted by a follower player. Where, α_4, $\beta_4 \in [0,1]$ are the weight parameters of computation overhead and energy consumption.

The cost of the follower player represents the detection rates of infected edge links and malicious UAVs, i.e., $Reward^1$. The follower's cost is expressed as:

$$Cost^2 = Reward^1 \quad (10)$$

The payoff function $\Psi_{i,t}^2$ of a follower player is modeled by Eq. (11).

$$\Psi_{i,t}^2\left(S^2{}_{i,t}, S^1{}_{i,t}, \phi^2{}_{i,t}\right) = Reward_{i,t}{}^2\left(S^2{}_{i,t}, S^1{}_{i,t}, \phi^2{}_{i,t}\right) - Cost_{i,t}{}^2\left(S^2{}_{i,t}, S^1{}_{i,t}, \phi^2{}_{i,t}\right) \quad (11)$$

As shown in Eq. (12), the follower agent aims at finding the optimal strategies $a^2{}_{k,i}$ and $a^2{}_{d,i}$ by computing the probabilities $p^2{}_k$ and $p^2{}_d$ that maximize its payoff $\Psi_{i,t}^2$.

$$\max_{p^2{}_k, p^2{}_d} \sum_{k=1}^{K}\sum_{d=1}^{D} p^2{}_k * p^2{}_d * \Psi_{i,t}^2\left(S^2{}_{i,t}, S^1{}_{i,t}, \phi^2{}_{i,t}\right) \quad (12)$$

4.3.2. Optimal stackelberg security equilibrium solution

The strategies of leader and follower players, $S^1{}_{i,t}(\varphi^1{}_{i,t}, a^1{}_{k,i}, a^1{}_{d,i})$ and $S^2{}_{i,t}(\varphi^2{}_{i,t}, a^2{}_{k,i}, a^2{}_{d,i})$ depend on the current and future Stackelberg security states $\varphi^1{}_{i,t}$ and $\varphi^2{}_{i,t}$. That is, the leader player considers the strategy of the follower at each stage i as well as the strategies of the subsequent I stages and vice versa. Therefore, the Stackelberg Equilibrium (SE) solution is computed recursively as demonstrated in [28]. The optimal payoffs of the leader and follower players at SE point is defined as shown in Eqs. (13) and (14).

$$\Psi_{i,t}^{*1}\left(S^{*1}{}_{i,t}, S^{*2}{}_{i,t}, \phi^{*1}{}_{i,t}\right) = \max_{p^1{}_k, p^1{}_d} \min_{p^2{}_k, p^2{}_d} P * \Psi_{i,t}^1\left(S^1{}_{i,t}, S^2{}_{i,t}, \phi^1{}_{i,t}\right) \quad (13)$$

$$\Psi_{i,t}^{*2}\left(S^{*2}{}_{i,t}, S^{*1}{}_{i,t}, phi^{*2}{}_{i,t}\right) = \max_{p^2{}_k, p^2{}_d} \min_{p^1{}_k, p^1{}_d} P * \Psi_{i,t}^2\left(S^2{}_{i,t}, S^1{}_{i,t}, \phi^2{}_{i,t}\right) \quad (14)$$

Where $P = \sum_{k=1}^{K}\sum_{d=1}^{D} p^1{}_k * p^1{}_d * p^2{}_k * p^2{}_d$;

The total payoffs of the leader and follower players in the Stackelberg security states $\varphi^1{}_{i,t}$ and $\varphi^2{}_{i,t}$ is the sum of their payoffs from stages i to I, which are computed as

$$\Psi_{Total}^1\left(S^1{}_{i,t}, S^2{}_{i,t}, \phi^1{}_{i,t}\right) = P * \left[\sum_{i=1}^{I} \Psi_{i,t+1}^{*1}\left(S^{*1}{}_{i,t+1}, S^{*2}{}_{i,t+1}, \phi^{*1}{}_{i,t+1}\right) + \sum_{i=1}^{I} \Psi_{i,t}^1\left(S^1{}_{i,t}, S^2{}_{i,t}, \phi^1{}_{i,t}\right)\right] \quad (15)$$

$$\Psi_{Total}^2\left(S^2{}_{i,t}, S^1{}_{i,t}, \phi^2{}_{i,t}\right) = P * \left[\sum_{i=1}^{I} \Psi_{i,t+1}^{*2}\left(S^{*2}{}_{i,t+1}, S^{*1}{}_{i,t+1}, \phi^{*2}{}_{i,t+1}\right) + \sum_{i=1}^{I} \Psi_{i,t}^2\left(S^2{}_{i,t}, S^1{}_{i,t}, \phi^2{}_{i,t}\right)\right] \quad (16)$$

The leader and follower players aim at maximizing their respective total payoffs by taking the best responses of their opponents as shown in Eqs. (17) and (18).

$$\begin{aligned}
&\forall S'^2{}_{i,t},\ p'^2{}_k \text{ and } p'^2{}_d \\
&\max_{p^1{}_k, p^1{}_d} \min_{p'^2{}_k, p'^2{}_d} \Psi_{Total}^1\left(S^1{}_{i,t}, S'^2{}_{i,t}, \phi^1{}_{i,t}\right) \\
&s.t. \\
&P * \left[\sum_{i=1}^{I} \Psi_{i,t+1}^{*1}\left(S^{*1}{}_{i,t+1}, S^{*2}{}_{i,t+1}, \phi^{*1}{}_{i,t+1}\right) + \sum_{i=1}^{I} \Psi_{i,t}^1\left(S^1{}_{i,t}, S^2{}_{i,t}, \phi^1{}_{i,t}\right)\right] \\
&< P' * \left[\sum_{i=1}^{I} \Psi_{i,t+1}^{*1}\left(S^{*1}{}_{i,t+1}, S^{*2}{}_{i,t+1}, \phi^{*1}{}_{i,t+1}\right) + \sum_{i=1}^{I} \Psi_{i,t}^1\left(S^1{}_{i,t}, S'^2{}_{i,t}, \phi^1{}_{i,t}\right)\right]
\end{aligned} \quad (17)$$

Where $P'=\sum_{k=1}^{K}\sum_{d=1}^{D} p^1{}_k * p^1{}_d * p'^2{}_k * p'^2{}_d, p'^2{}_k > p^2{}_k$ and $p'^2{}_d > p^2{}_d$

$$\forall S'^1{}_{i,t}, p'^1{}_k \text{ and } p'^1{}_d$$

$$\max_{p^2{}_k, p^2{}_d} \min_{p'^1{}_k, p'^1{}_d} \Psi^2_{Total}\left(S^2{}_{i,t}, S^1{}_{i,t}, \phi^2{}_{i,t}\right)$$

$$s.t.$$

$$P * \left[\sum_{i=1}^{I} \Psi^{*2}_{i,t+1}\left(S^{*2}{}_{i,t+1}, S^{*1}{}_{i,t+1}, \phi^{*2}{}_{i,t+1}\right) + \sum_{i=1}^{I} \Psi^2_{i,t}\left(S^2{}_{i,t}, S'^1{}_{i,t}, \phi^2{}_{i,t}\right)\right]$$

$$< P'' * \left[\sum_{i=1}^{I} \Psi^{*2}_{i,t+1}\left(S^{*2}{}_{i,t+1}, S^{*1}{}_{i,t+1}, \phi^{*2}{}_{i,t+1}\right) + \sum_{i=1}^{I} \Psi^2_{i,t}\left(S^2{}_{i,t}, S'^1{}_{i,t}, \phi^2{}_{i,t}\right)\right] \quad (18)$$

Where $P''=\sum_{k=1}^{K}\sum_{d=1}^{D} p'^1{}_k * p'^1{}_d * p^2{}_k * p^2{}_d, p'^1{}_k > p^1{}_k$ and $p'^1{}_d > p^1{}_d$

In this security game, each security agent determines the optimal strategy of the attacker by solving Eq. (17) for a given Stackelberg security states $\varphi^1{}_{i,t}$ and $\varphi^2{}_{i,t}$. As illustrated in Algorithm 1, the leader and the follower players compute their respective payoffs $\Psi^1_{i,t}$ and $\Psi^2_{i,t}$ at each Stackelberg security states $\varphi^1{}_{i,t}$ and $\varphi^2{}_{i,t}$ (during a period t) and attempt to predict the optimal strategies ($S^{*1}{}_{i,t+1}$, $S^{*2}{}_{i,t+1}$) that could occur in the subsequent state, i.e., at time $t+1$. These optimal players' strategies correspond to the actions that the security agent and attacker wishes, i.e., attack and monitor (detect) the same link and UAV node. In this case the payoff $\Psi^1_{i,t}$ of a leader agent reaches its optimal value (which corresponds to $\Psi^{*1}_{i,t}$) since almost of attackers are detected by the leader players.

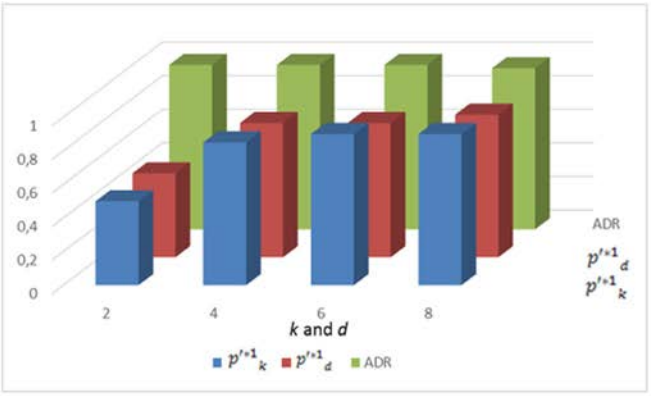
Algorithm 1 Attacks and cyber security process.

Begin:
Repeat:
For each *stage i:*
Leader L *Computes* $\Psi^1_{i,t}(S^1{}_{i,t}, S^2{}_{i,t}, \phi^1{}_{i,t})$,
Follower F *Computes* $\Psi^2_{i,t}(S^2{}_{i,t}, S^1{}_{i,t}, \phi^2{}_{i,t})$,
If $\Psi^1_{i,t} < \max_{p^1{}_k, p^1{}_d} \sum_{k=1}^{K}\sum_{d=1}^{D} p^1{}_k * p^1{}_d * \Psi^1_{i,t}$ ***Then***
L *Computes* $\sum_{i=1}^{I} \Psi^1_{i,t}$ and *Estimates* $\sum_{i=1}^{I} \Psi^{*1}_{i,t+1}$,
L *Computes* Ψ^1_{Total},
If $\Psi^1_{Total} \geq \max_{p^1{}_k, p^1{}_d} \min_{p'^2{}_k, p'^2{}_d} \Psi^1_{Total}$ ***Then***
If $p'^2{}_k > p^2{}_k$ ***Then***
Monitor and *Protect* the link *k*,
If $p'^2{}_d > p^2{}_d$ ***Then***
Monitor and *Protect* the UAV node *d*,
If $\Psi^2_{i,t} < \max_{p^2{}_k, p^2{}_d} \sum_{k=1}^{K}\sum_{d=1}^{D} p^2{}_k * p^2{}_d * \Psi^2_{i,t}$ ***Then***
F *Computes* $\sum_{i=1}^{I} \Psi^2_{i,t}$ and *Estimates* $\sum_{i=1}^{I} \Psi^{*2}_{i,t+1}$,
F *Computes* Ψ^2_{Total},
If $\Psi^2_{Total} \geq \max_{p^2{}_k, p^2{}_d} \min_{p'^1{}_k, p'^1{}_d} \Psi^2_{Total}$ ***Then***
If $p'^1{}_k > p^1{}_k$ ***Then***
Attacks the link *k*,
If $p'^1{}_d > p^1{}_d$ ***Then***
Attacks the UAV node *d*,
Until: The stage *I* of Stackelberg security game.

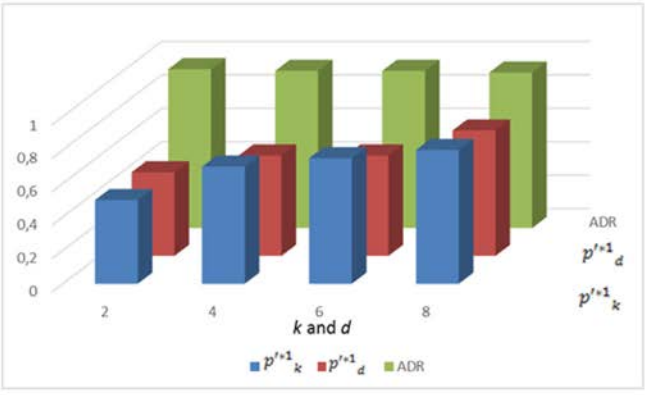
For each stage *i*, the leader and follower agents respectively protects and attacks the link and UAV node, where the maximum number of stages is equal to *I*. Hence, the complexity of the Algorithm 1 is equal to O (I*2).

5. Simulation results

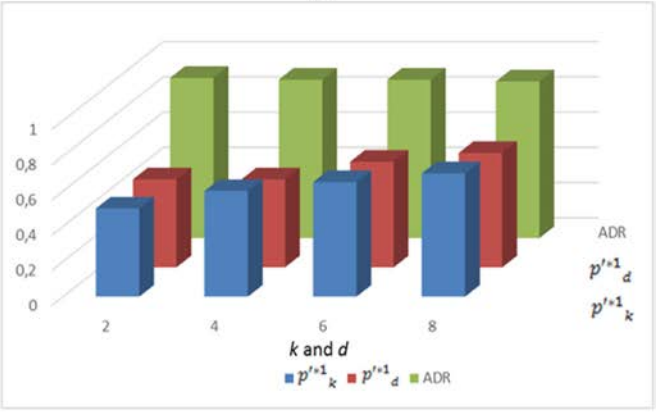
We evaluated the proposed cyber security framework for UAV-Edge computing with Network Simulator (NS3) [29]. First, we study the convergences of the optimal probabilities ($p^1{}_k$,$p^1{}_d$) that allow the leader player to detect the cyber-attacks while considering the computation overhead and the energy consumption of UAV (where the security agent is activated). We vary *d* and *k* which are the number of UAVs and edge links that are within the radio range of the leader agent L_l, where $1 \leq l \leq L$ and *L* is the maximum number of security agents in UAV-Edge computing network. Afterward, we compare the performance of the proposed security framework with state of the art intrusion detection frameworks developed for UAV and vehicular networks [18,19]. The main metrics that are analyzed in the simulations are:



(a)

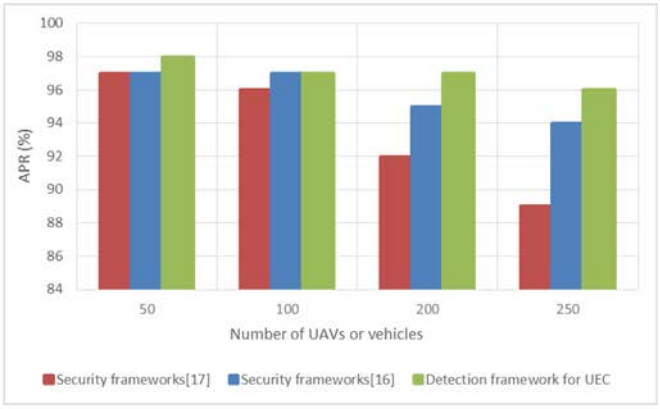


(b)

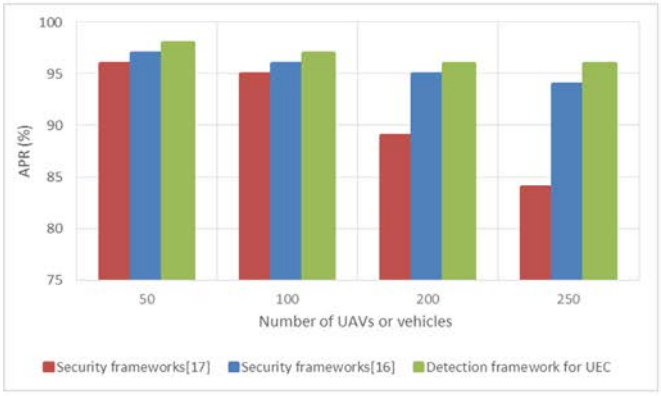


(c)

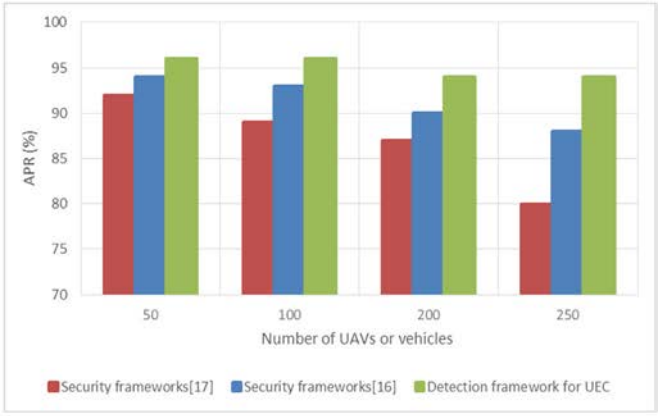
Fig. 3. Optimal protection probabilities where the number of attackers equal to (a) 5%, (b) 20% and (c) 30% of overall UAVs being attackers.



(a)



(b)



(c)

Fig. 4. Accuracy protection rate where the number of attackers equal to (a) 5%, (b) 20% and (c) 30% of overall UAVs (or vehicles) being attackers.

- The *attacks detection rate (ADR)*: Number of detected attacks over the total number of cyber-attacks against the network ($0 < \text{ADR} \leq 1$),
- The *false alarms rate (FAR):* Number of legitimate UAVs (and links) that are detected by the security agents as infected nodes (links) over the total number of UAVs and links ($0 < \text{FAR} \leq 1$),
- The *Average energy consumption (EC):* The sum of energy's percentage that each UAV consumes over the number of UAV nodes.

In this simulation, the mobility model of UAVs and vehicles is a deterministic mobility model [30], where the vehicles and the UAV nodes have their own mobility behavior and follow a deterministic path. We vary the number of nodes (UAV and vehicles) from 50 to 250 and the attacker from 5% to 30% of overall nodes. The main purpose of the offloading attacks is to increase the end-to-end delay of the network by attacking the offloading tasks requested by the mobile nodes [9,17]. Meanwhile, the purpose of attacks that target the UAVs such as DoS attacks is to increase the energy consumption and to drop the critical information of UAVs. The attackers are aware of the paths that legitimate UAVs follow and they can easily launch cyber-attacks against the targeted UAVs. The purpose of our simulations is to detect the attackers in order to prevent critical damages, e.g., UAV crash and altering critical information of UEC network. Table 1 summarizes the simulation parameters.

Table 1
Simulation setup.

Simulation parameters	Value (s)
Simulation time	5 min
IEEE 802.11	802.11 b and 802.11 p
Bandwidth	10 MHz
Transmission power	33 dB
Number of mobile nodes	From 50 to 250
Mobility model	Deterministic mobility model [7]
Speed	[30,...,90] Km/h
UDP flow rate	2 Mbps
Packet size	512 Bytes
UAV's range	80 m
Vehicle's range	300 m
Number of RSU and enodeB	20

5.1. Optimal probabilities ($p^1{}_k, p^1{}_d$)

As shown in Fig. 3, we vary the number of attackers from 5% to 30% of overall UAVs nodes, set the number of UAVs to 250 nodes and analyze the protection probabilities $p^1{}_k$ and $p^1{}_d$. The best protection probabilities that allow us to get a high attacks detection rate are selected. It's noted that, in our simulation, we compute the average values of the $p^1{}_k$and $p^1{}_d$, which are computed as: $p'^1{}_k = \frac{\sum_{l=1}^{L} pl^1{}_k}{L}$ and $p'^1{}_d = \frac{\sum_{l=1}^{L} pl^1{}_d}{L}$. Fig. 3(a)–(c) show that when the average number of neighbors, d and offloading links, k increase the average protection probabilities $p'^1{}_k$ and $p'^1{}_d$ increase. This is due to the fact that the security agents monitor all the links and the UAVs that are suspected to be an attractive target of attackers. Thus, the attack detection rate is higher than 0.85, even when d and k increase. We found that the detection rate decreases when the number of attackers is higher than 20% to save the energy consumption of UAVs. Indeed, the security agents are not activated simultaneously at each UAV and hence a certain number of infected UAVs and links are not monitored. However, as illustrated in Fig. 3, even in a worst case, i.e., when the number of attackers is above 30% of overall nodes, the detection rate still acceptable, i.e. it is higher than 0.85.

5.2. Accuracy protection

The accuracy protection rate (APR) is equal to ADR-FAR. We analyze it by varying the number of attackers and the number of UAVs and vehicles. The proposed detection framework for UEC network and the security framework for UAVs network proposed in [19] are analyzed by varying the number of UAVs. The security framework for vehicular edge computing proposed in [18] is analyzed by varying the number of vehicles. It's to be noted that, to allow the security frameworks [18,19] to detect the attacks that our framework is able to detect, we integrate in the security frameworks [18,19] the detection rules against the offloading and DoS attacks. As shown in Fig. 4(a)–(c), the accuracy protection of a

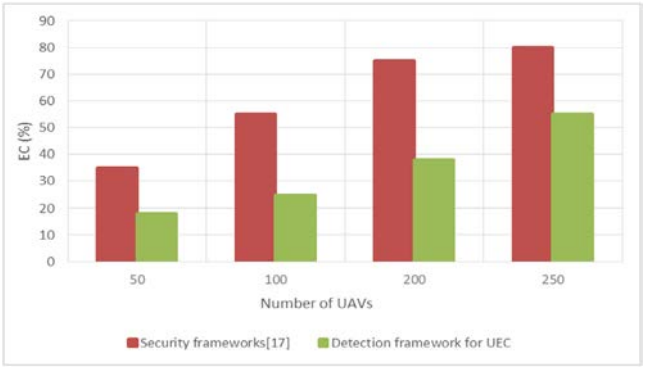


Fig. 5. Average percentage of UAVs' energy consumption.

detection framework for UEC is high and it overpasses the accuracy protection of the security framework [18]. This result is achieved even when the number of attackers reaches 30% of overall nodes. However, the accuracy protection of security framework [19] decreases, specifically when the number of attackers increases. The proposed detection framework for UEC achieves a high accuracy protection due to the following reason:

Cyber Stackelberg game: The proposed defense strategy exploits two phases of the attackers: thinking and learning. In the thinking phase, cyber game models how cyber-attacks are likely to attack, and in the learning phase, cyber game predict when and which legitimate UAVs and offloading links could be attacked. The Stackelberg game theory is used as a mathematical tool to study the interaction of conflict between the non-cooperative players, security agents and attackers. The goal of this study is to define the optimal decision of security agents to detect and predict the attacks occurred in UEC network by taking into account the computation overhead and energy consumption.

5.3. Energy consumption

Fig. 5 illustrates the average percentage of UAVs' consumption, which is defined as EC. In our analyses, we study the energy consumption of UAVs network when our detection framework for UEC and the security framework [19] are embedded. The percentage of UAVs' consumption is computed in a worst case scenario, i.e., the number of attackers is equal to 30% of overall nodes. As shown in Fig. 5, the proposed detection framework shows low energy consumption as compared to security frameworks [19]. This is mainly due to the fact that, in [19] at each UAV node, the detection agent activates its detection module to protect its neighbors nodes, which leads an increase on the communication overhead and hence the average percentage of UAVs' energy consumption will be increased. This is unlike our detection framework since the detection process is not activated simultaneously at each UAV node, where the detection process are launched only when a suspected links and UAVs are identified during the monitoring process.

6. Conclusion

In this work, we have addressed the tradeoff issue between energy consumption and computation overhead, and cyber defense in UEC network. This issue is formulated as a non-cooperative Stackelberg game between security agents and attackers that target the offloading links and UAVs nodes. In the proposed mathematical model, we define the optimal Stackelberg strategies of the opponent players, which correspond to scenario in which the security agent and attacker respectively protects and attacks the same offloading link and UAV node. Simulation results show that almost of 55% of required energy are used by UAVs to achieve a high accuracy protection rate, over 94%. This result is obtained when the number of UAV and attackers is high.

Conflict of Interest

None.

References

[1] D.P. Shepard, J.A. Bhatti, T.E. Humphreys, A.A. Fansler, Evaluation of smart grid and civilian UAV vulnerability to GPS spoofing attacks, in: Proc. ION GNSS Meeting, Nashville, TN, USA, 2012, pp. 1–15.
[2] M.A. Messous, S.-M. Senouci, H. Sedjelmaci, Network connectivity and area coverage for UAV fleet mobility model with energy constraint, IEEE Wireless Communications and Networking Conference, 2016.
[3] E. Yanmaz, R. Kuschnig, C. Bettstetter, Channel measurements over 802.11a-based UAV-to-ground links, in: Proc. IEEE Globecom Wi-UAV Workshop, Houston, TX, USA, 2011, pp. 1280–1284.
[4] M.A. Messous, A. Arfaoui, A. Alioua, S.-M. Senouci, A sequential game approach for computation-offloading in an UAV network, IEEE Global Communications Conference, 2017.
[5] Z. Zhou, J. Feng, B. Gu, B. Ai, S. Mumtaz, J. Rodriguez, M. Guizani, When mobile crowd sensing meets UAV: energy-efficient task assignment and route planning, IEEE Trans. Commun. 66 (11) (2018) 5526–5538.
[6] Y. Mao, C. You, J. Zhang, K. Huang, K.B. Letaief, A survey on mobile edge computing: the communication perspective, IEEE Commun. Surv. Tutor. 19 (4) (2017) 2322–2358.
[7] S. Chen, T. Zhao, H.-H. Chen, W. Meng, Downlink coordinated multi-point transmission in ultra-dense networks with mobile edge computing, IEEE Netw. (2018) 1–8.
[8] S. Garg, A. Singh, S. Batra, N. Kumar, L.T. Yang, UAV-Empowered edge computing environment for cyber-threat detection in smart vehicles, IEEE Netw. 32 (3) (2018) 42–51.
[9] L. Xiao, C. Xie, T. Chen, H. Dai, H.V. Poor, A mobile offloading game against smart attacks, IEEE Access 4 (2016) 2281–2291.
[10] G. Vasconcelos, C. Carrijo, R. Miani, J. Souza, V. Guizilini, The impact of DoS attacks on the AR.Drone 2.0, in: 13rd IEEE Latin American Robotics Symposium and 4th Brazilian Symposium on Robotics (LARS-SBR), Recife, Brazil, 2016, pp. 127–132.
[11] B. An, M. Tambe, Stackelberg security games (SSG) basics and application overview, in: Improving Homeland Security Decisions, Cambridge University Press, 2017, pp. 485–507.
[12] Z. Zhou, J. Feng, L. Tan, Y. He, J. Gong, An air-ground integration approach for mobile edge computing in iot, IEEE Commun. Mag. 56 (8) (2018) 40–47.
[13] R.-H. Hsu, J. Lee, T.Q.S. Quek, J.-C. Chen, Reconfigurable security: edge-computing-based framework for iot, IEEE Netw. 32 (5) (2018) 92–99.
[14] A. Alamer, Y. Deng, G. Wei, X. Lin, Collaborative security in vehicular cloud computing: a game theoretic view, IEEE Netw. 32 (3) (2018) 72–77.
[15] X. Huang, R. Yu, J. Kang, Y. Zhang, Distributed reputation management for secure and efficient vehicular edge computing and networks, IEEE Access 5 (2017) 25408–25420.
[16] J. Cui, L. Wei, J. Zhang, Y. Xu, H. Zhong, An efficient message-authentication scheme based on edge computing for vehicular ad hoc networks, IEEE Trans. Intell. Transp. Syst. (2018) 1–12.
[17] L. Xiao, C. Xie, M. Min, W. Zhuang, User-centric view of unmanned aerial vehicle transmission against smart attacks, IEEE Trans. Veh. Technol. 67 (4) (2017) 3420–3430.
[18] H. Sedjelmaci, I. Ben Jemaa, M. Hadji, A. Kaiser, Security framework for vehicular edge computing network based on behavioral game, IEEE Globecom, 2018.
[19] R. Mitchell, I.R. Chen, Adaptive intrusion detection of malicious unmanned air vehicles using behavior rule specifications, *IEEE Trans. Syst. Man Cybern.* 44 (5) (2014) 593–604.
[20] V. Dey, V. Pudi, A. Chattopadhyay, Y. Elovici, Security vulnerabilities of unmanned aerial vehicles and countermeasures: an experimental study, in: IEEE 31th International Conference on VLSI Design and 17th International Conference on Embedded Systems, Pune, India, 2018, pp. 398–403.
[21] H. Sedjelmaci, S.-M. Senouci, N. Ansari, A hierarchical detection and response system to enhance security against lethal cyber-attacks in UAV networks, IEEE Trans. Syst. Man Cybern. 48 (9) (2014) 1594–1606.
[22] A. Servin, D. Kudenko, Multi-agent reinforcement learning for intrusion detection, in: European Symposium on Adaptive and Learning Agents and Multi-Agent Systems, Springer, 2008, pp. 211–223.
[23] S.M. Giray, Anatomy of unmanned aerial vehicle hijacking with signal spoofing, in: 6th International Conference on Recent Advances in Space Technologies (RAST), Istanbul, Turkey, 2013, pp. 795–800.
[24] D. Dolev, A. Yao, On the security of public key protocols, IEEE Trans. Inf. Theory 29 (2) (1983) 198–208 March.
[25] G. Vasconcelos, G. Carrijo, R. Miani, J. Souza, V. Guizilini, The impact of DoS attacks on the AR.Drone 2.0, in: Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR), Recife, Brazil, 2016, pp. 127–132.

[26] C. Gudla, Md.S. Rana, A.H. Sung, Defense techniques against cyber attacks on unmanned aerial vehicles, in: International Conference on Embedded Systems, Cyber-physical Systems, & Applications, 2018, pp. 110–116.
[27] M.A. Messous, H. Sedjalmaci, H. Noureddine, S.M. Senouci, Computation offloading game for an UAV network in mobile edge computing, IEEE ICC, 2017.
[28] N. Abuzainab, W. Saad, Dynamic connectivity game for adversarial internet of battlefield things systems, IEEE Internet Things J. 5 (1) (2018) 378–390.
[29] Network simulator (ns-3). Available on http://www.nsnam.org.
[30] O. Bouachir, A. Abrassart, F. Garcia, N. Larrieu, A mobility model for UAV ad hoc network, in: International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 2014, pp. 383–388.