Security in networks of unmanned aerial vehicles for surveillance with an agent-based approach inspired by the principles of blockchain

Iván García-Magariño[a,b,*], Raquel Lacuesta[a,b], Muttukrishnan Rajarajan[c], Jaime Lloret[d]

[a]*Department of Computer Science and Engineering of Systems, University of Zaragoza, Escuela Universitaria Politécnica de Teruel, c/ Atarazana 2, 44003 Teruel, Spain*
[b]*Instituto de Investigación Sanitaria Aragón. University of Zaragoza. Zaragoza, Spain*
[c]*School of Mathematics, Computer Science and Engineering, City, University of London, London EC 1V 0HB, United Kingdom*
[d]*Integrated Management Coastal Research Institute, Universitat Politècnica de València, 46022, València, Spain*

Abstract

Unmanned aerial vehicles (UAVs) can support surveillance even in areas without network infrastructure. However, UAV networks raise security challenges because of its dynamic topology. This paper proposes a technique for maintaining security in UAV networks in the context of surveillance, by corroborating information about events from different sources. In this way, UAV networks can conform peer-to-peer information inspired by the principles of blockchain, and detect compromised UAVs based on trust policies. The proposed technique uses a secure asymmetric encryption with a pre-shared list of official UAVs. Using this technique, the wrong information can be detected when an official UAV is physically hijacked. The novel agent based simulator ABS-SecurityUAV is used to validate the proposed approach. In our experiments, around 90% of UAVs were able to corroborate information about a person walking in a controlled area, while none of the UAVs corroborated fake information coming from a hijacked UAV.

[*] Corresponding author, Telephone: +34 978645348, Fax: +34 978618104
Email addresses: ivangmg@unizar.es (Iván García-Magariño), lacuesta@unizar.es (Raquel Lacuesta), r.muttukrishnan@city.ac.uk (Muttukrishnan Rajarajan), jlloret@dcom.upv.es (Jaime Lloret)

Keywords: agent-based simulator, blockchain, multi-agent system, security, surveillance, unmanned aerial vehicle

1. Introduction

Vehicular ad-hoc networks (VANETs) are difficult to maintain because of the rapid and dynamic change of the network topology, the short connection durations, and the frequent disconnections [1]. The main challenges are in (a) the trust and information verification, (b) the key distribution for maintaining secure channels, and (c) the forwarding algorithms for finding the best route. Some of the most common attacks are (1) identity and geographical position revealing, (2) Denial of Service (DoS), (3) Sybil attack creating the illusion of several cars with the same ID, (4) Spam to increase the latency of network transmissions, (5) Man in the Middle (MiM), (6) black hole attack, by always declaring having the shortest path, and (7) fake location information. Unmanned aerial vehicles (UAVs) also usually use ad hoc networks in the absence of a network infrastructure.

Blockchain improves the security of distributed datasets by sharing and checking the information by the different implied parties [2]. In this context, blockchain is defined as a collaborative security technique that is used to guarantee the veracity of information. The survey in [3] describes different kinds of security threats in blockchain-based systems and security-related enhancement solutions for them.

Literature shows growing interest in surveillance with UAVs covering aspects such as optimal coverage [4] and continuous monitoring [5]. However, security in UAV network is not trivial as revealed in the detection of intrusions in UAVs [6] or in the improvement of secrecy rate [7].

In this context, the current work presents a novel solution for the accurate detection of intruders in a controlled area by a fleet of UAVs even when a minority of these have been physically hijacked. The proposed solution takes advantage of narrowing to this specific domain, by being based on the secure sharing of the UAVs that directly observed an intruder, which is a specific information in the context of surveillance. This approach is inspired by the principles of blockchain for maintaining a secure record of all the reported observations, which is novel to the best of our knowledge. In the current proposal, an asymmetric encryption ensures the authentication of the signature of the corresponding UAVs. In this manner, even if a minority of UAVs is physically hijacked, then their misinformation is easily tracked by trust heuristics. The current approach is validated using a novel agent-based simulator (ABS). Although these simulations were a simplified version of the reality (e.g. omitting the influence of wind in UAV movements or the effect of the rain in the wireless communications), these simulations were useful for understanding the repercussion of the proposed security protocol on the macro level composed by a group individual UAVs, which could be independently hacked for example.

The remainder of the paper is organized as follows. The next section presents related work. Section 3 presents the novel security method for surveillance from UAVs. It defends from official UAVs that may be officially compromised, and is illustrated with a novel ABS. Section 4 presents the validation of the current approach using the ABS. Concluding remarks are in section 5, including some possible future research directions.

2. Related work

The applications of UAVs are very varied, and these can be classified into civil and military ones. On the one hand regarding civil applications, the survey in [8] analyzed the existing application of UAVs for civil applications from a communication perspective, like natural disaster monitoring, border surveillance, emergency assistance, search and rescue operations, delivery of goods and construction, concluding that security was essential for guaranteeing proper communication among UAVs. Within civil applications, UAVs play a relevant role in communications, not only conforming UAV networks, but also supporting the connectivity of other kinds of VANETs, for example when these have non-cooperative vehicles [9]. In this line of research, Sbeiti et al. [10] analyzed the airborne network assisted applications based on the low-altitude UAVs combined with WLAN mesh networks (WMNs), they proposed the Position-Aware, Secure and Efficient mesh Routing approach (PASER) for avoiding routing attacks, and their experiments showed that their approach was secure from these attacks. UAVs have also used ad hoc networks connected to mountaineers' smartphones for supporting an emergency rescue system in critical areas without GSM cellular coverage [11]. Nevertheless, these works did not study the possible vulnerabilities raised by a physical hijacking of a UAV in the surveillance context, in which intruders and their collaborators may be so much interested in violating security measures for even taking the risk of exposing themselves by physically hijacking

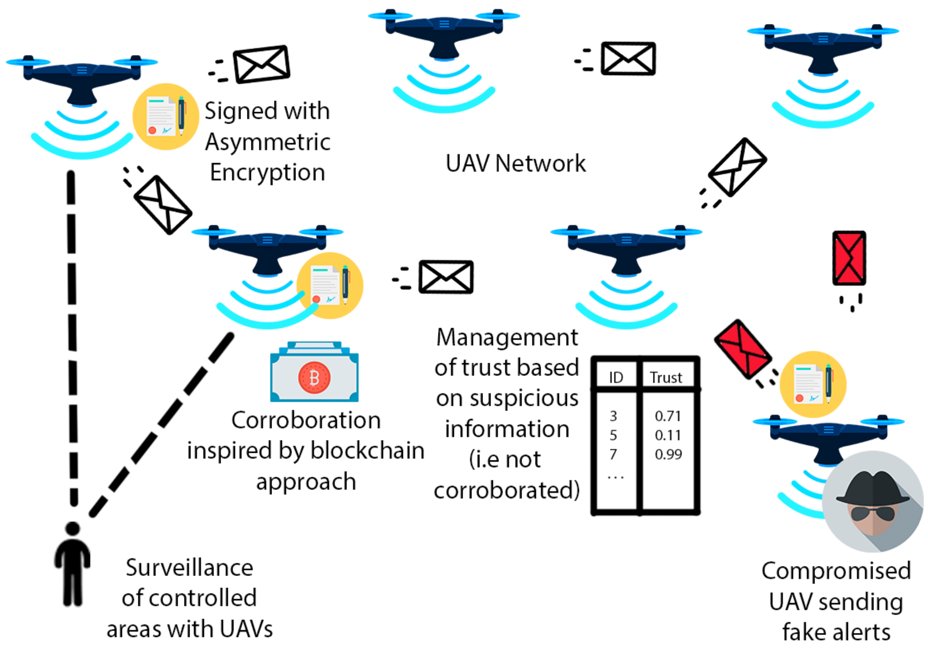
UAVs, which would be really weird in civil applications like rescue systems.

On the other hand concerning UAV military applications, reliable and secure communications are crucial in surveillance, in domains such as battlefield [12] and monitoring of borders. In surveillance, secure and reliable communications have relied on the proper encryption and lightweight transfer of data, like in the framework for IoT surveillance systems based on video summarization and image encryption [13], and the detection of malicious behaviors has been conducted in a large variety of UAV communication architectures [14]. In the field of surveillance with UAVs, [15] proposed a mechanism for achieving persistent surveillance considering dynamic aspects of the environment and being tolerant to UAV failures. Their approach decomposed the controlled area in cells, and each cell had an age, meaning the time elapsed since its last observation by any UAV. The goal of their algorithm was to minimize the maximum age of all the cells, by considering possible paths for covering the cells with maximum age, and selecting the one that minimized the estimated maximum age. In addition, [5] proposed an algorithm for maintaining a permanent and continuous surveillance infrastructure of UAVs, in which UAVs were coordinated for automatically charging and flying in a balanced way. In this context, [4] proposed a cooperative search and surveillance with UAVs, with a game-based approach with coordinated motion for optimal coverage, sensor observation, and cooperative information fusion. They used binary log-linear learning for the control of motion and information fusion to construct a probability map, proving the effectiveness of their approach with simulations. However, these approaches did not consider the security issues related to the fact that a UAV could be physically attacked and compromised for adopting a malicious behavior.

There are several works that focus on improving different aspects of communication security in UAVs, which are general and can be applied to different contexts. For example, [7] presented a mechanism for guaranteeing secure communications with an iterative convergent algorithm composed of two nested loops, in which the outer loop measured the difference of concave in order to increase the secrecy rate and the inner loop applied the ellipsoid method. They applied a water-filling-based solution to make the algorithm computationally efficient, proving the convergence to a Karush-Kuhn-Tucker point of the secrecy rate maximization problem, and their simulation results showed that their approach enhanced secrecy in comparison to an alternative static solution. In addition, [6] analyzed the intrusion detection in UAV networks reaching a balance between frequent monitoring and the network performance, focusing on the ejection of nodes with malware but trying to avoid as much as possible the wrong ejection of nodes without malware just because there were some occasional errors. They applied a Bayesian game model for accurately detecting attacks with low false positive rates, and they achieved a reliable detection accuracy according to their simulated results. Moreover, [16] studied the communication security in UAVs, presenting some low-cost implementations of the GPS spooling and WiFi attacks that effectively compromised some UAVs, and they proposed some solutions for defending against these attacks. Furthermore, [17] proposed to use trust management as an alternative to cryptography to avoid excessive energy and processing consumption, based on the assumption that there were network nodes with malicious behaviors behaving intelligently for not being detected, and their UAV-assisted detection mechanism was able to rapidly detect misbehaviors of network nodes. In addition, [18] proposed an authentication system for using an encrypted channel for protecting UAVs from cyber attacks, and they tested their approach with commercial UAVs showing its utility. However, these works did not guarantee security in surveillance in case an official UAV was compromised.

Agent-based simulators (ABSs) have been useful for testing security strategies in different network types, like ABS-TrustSDN [19], which allowed defining and assessing trust policies over network nodes in software-defined networks, and its experimentations showed that this tool was able to properly assess several strategies that obtained significantly different effectiveness results. However, this ABS was not able to simulate the surveillance by UAVs for detecting hijacked UAVs.

In summary, UAVs have a great diversity of applications ranging from civil to military applications and in some cases they need to rely on their own network built upon vehicular-to-vehicular (V2V) communications instead of vehicular-to-infrastructure (V2I) ones. Surveillance is one of the most common applications, and the literature agrees on the importance of security in UAV networks. However, to the best of our knowledge, the literature lacks the appropriate methods for preventing from physical hijacking of one official UAV in the context of distributed surveillance by UAVs, and the approach presented in the next section addresses this gap of the literature.



ID	Trust
3	0.71
5	0.11
7	0.99
...	

Figure 1: Overview of the detection of compromised UAVs in surveillance

3. Method for detecting compromised UAVs in surveillance

3.1. Overview and assumptions

In general, this approach is designed for borders with a low transit of people (e.g. in natural borders such as mountains). Figure 1 shows an overview of the current method for detecting compromised UAVs in the surveillance context. This approach addresses the distributed management of a UAV network for detecting people in the surveillance of a particular wide area. This method is based on the assumption that normally each person crossing a controlled area is observed by several UAVs, although this is not necessary. The observations from all the UAVs are propagated following the blockchain principles, and the information is stored distributedly in UAVs.

In blockchain, there is a list of records called blocks. The links between these records are secured by cryptography in which each block has a hash of the content of the previous block. In this way, a record cannot be altered without altering all the subsequent blocks. Notice that if a block was altered, the hash of the next block would need to be altered, and this change in the next block would require a change in the following block, and so on. All

]

the blocks are stored in a distributed manner in different nodes. Thus, a block could not be changed unless the majority would change it. In this way, the information is safely and permanently stored in a peer-to-peer (P2P) network. Attacks are very difficult to be perpetrated, as it would probably require to compromise the majority of nodes. The current approach is inspired by these principles, and each UAV has a list of blocks of the identifiers of the official UAVs that have reported the existence of a person based on direct observation. This information is propagated along the UAV network so most UAVs have this information in a secured and permanently manner. If a compromised UAV tried to change this blockchain introducing false information (e.g. inserting fake observer identifiers), this would be discarded by other UAVs given the P2P principles of blockchain, unless the compromised UAVs are majority, which is considered very improbable according to the common assumptions of blockchain. Another possible scenario is that a compromised UAV pretended to have observed a person. In this way, this information cannot be discarded by the P2P blockchain network, since it could have happened. In this way, this false information would be introduced in the blockchain distributed among UAVs. However, we included a trust policy to detect these cases, by identifying events (i.e. a person crossing the border) that are only reported by only one direct observer repetitively, based on the assumption that each event is normally observed by several UAVs. Another possibility is that a compromised UAV omits the detection of a person, however normally the person would probably also be observed by other UAVs.

In the current approach, all the UAVs should be officially registered before the UAV fleet starts the surveillance activity. Each UAV has a list of the public keys of all the UAVs for signing each message and securely sending it to all UAVs avoiding MiM attacks by compromised UAVs. In this way, the UAVs can sign their messages with asymmetric encryption. The messages are forwarded over the UAV network, and each UAV can know a list of UAVs that observed a particular person.

An official UAV could be physically captured and compromised. In this case, this UAV could send fake alerts properly signed in order to disturb the correct functioning of the UAV network. However, the compromised UAV cannot alter its identity for impersonating other agents, due to the required asymmetric encryption for authenticating senders.

Each UAV records the IDs of all the monitoring UAVs from the messages, and checks whether each intruder is corroborated by at least several UAVs that observed them. However, UAVs can have a distributed management of trust in each UAV. This trust would consider the percentage of time a UAV was the only one that had observed an intruder event, penalizing the trust on it. It also considers the number of times it sent corroborated information.

In very large areas, due to limited resources it is not always possible to have sufficient number of UAVs to provide seamless communications. The current approach assumes that communications are usually disrupted, in the sense that a UAV may need to wait after generating a message until it can actually send the message. In particular, each UAV will wait until another UAV is in close proximity to actually communicate with it. This approach also assumes that UAVs cannot perform long-distance communications due to energy constraints and related safety issues. In a similar way, when a UAV receives a message, it stores the received information for forwarding it to different UAVs for an established duration.

The current approach is validated using an ABS with several agent types. One agent type impersonates intruders. Another agent type is the UAVs. In addition, UAVs have an internal flag that determines whether they are compromised.

3.2. Internal functioning of the security approach illustrated with an ABS model

The current approach is illustrated with the novel ABS called ABS-SecurityUAV. This ABS was implemented with NetLogo for its support and utility for representing mobile ad hoc networks [20]. The model of this ABS was organized in three modules: the “Setup” methods (initially executed at the beginning of the simulation); the “Go” methods (periodically invoked in each frame of the simulation); and the “Measure” methods (used for updating the measures of the graphs). This structure of modules was designed considering the common metrics for evaluating agent-oriented architectures [21] for reducing the coupling between modules and increasing the cohesion inside each of them. In addition, this ABS was developed considering the principles of PEABS (a process for developing efficient agent-based simulators) [22] for achieving efficient simulations.

In the Setup methods, UAVs are initialized considering the number entered by the user. One or several of these are compromised taking into account the number indicated by the user. These UAVs are initially located in a different place from the other UAVs, simulating that these are physically hijacked. Then, intruders are initialized if the user indicates so.

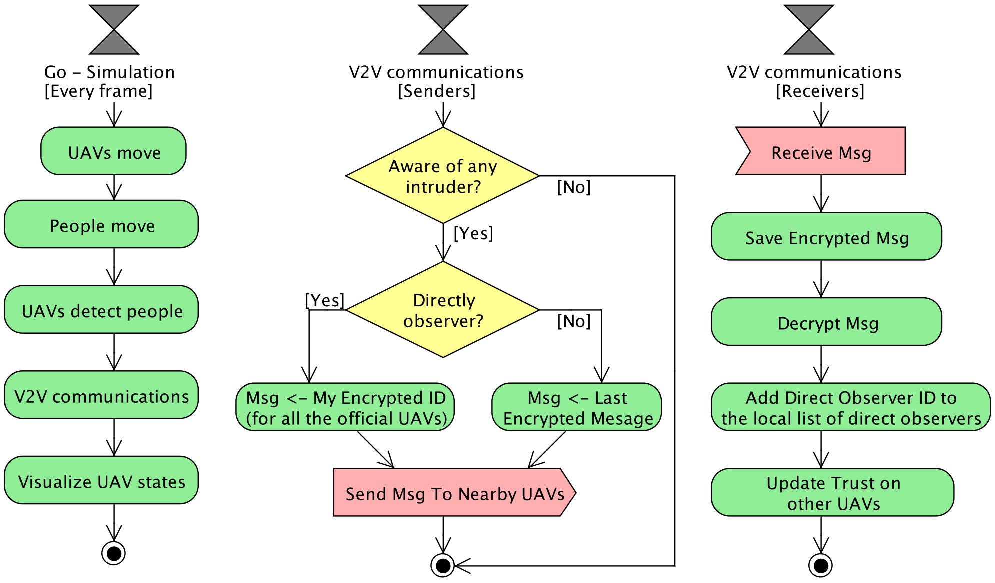


Figure 2: Block diagram of an excerpt of the Go methods of ABS-SecurityUAV

Regarding the Go methods, Figure 2 shows the block diagram of their most relevant part. The main method is shown in the left side. Firstly, both UAVs and people move. The former ones use a variable nondeterministic approach, while people mainly aim at crossing a controlled area following a specific direction with only slight variations. Then, UAVs detect whether any person is near. Following this, V2V communications are simulated. Finally, the UAV states are visualized considering the color notation described in the presentation of the user interface (UI) (see Appendix A) and showing some additional information in their labels. The block diagram also determines the V2V communications considering separately the perspectives of senders and receivers. UAVs communicate different messages regarding whether they have directly observed the intruder or they know it from the others. They respectively encrypt a message or forward the received encrypted message to all the nearby UAVs. The receptor UAVs save the encrypted message for later forwarding it, decrypt its content for updating their local list of direct UAV observers, and update their trust on each UAV.

Blockchain relies on P2P networks to store the information. In P2P, all nodes are both clients and servers so they communicate between peers.

Following the principles of P2P networks, in the current approach UAVs communicate as peers. Thus, all UAVs can adopt the both roles of sender and receiver determined in Figure 2. UAVs share information among each other about the direct observers of an event (i.e. a person crossing the border). In order to authenticate the information of the direct observer, the messages are signed using asymmetric encryption. All the official UAVs have the public keys of all the other official UAVs of the fleet, so they can check the identity of the direct observers. Thanks to the principles of asymmetric encryption, each UAV has its private key and only this UAV can sign their messages with its identifier. Thus, if a UAV observes a person, it can report its direct observation (similarly to a transaction in other blockchain scenarios). From this point, the UAV reports this information as a signed block into the blockchain system by sharing this information through P2P network. Once this information is included in the blockchain is difficult to change as it is shared through the permanent P2P storage. In addition, no UAV can report a block with a false identity due to the asymmetric encryption procedure. In case a UAV starts inserting false information signed by its real identifier (e.g. reporting observation of non-existent people), then the trust policy will detect its misinformation as shown in the experimentation of the current work.

Algorithm 1 shows the most relevant procedures for the detection of people and the coordination through V2V communications with a pseudo-language. The "main" procedure continuously (a) checks for intruders with computer vision, (b) shares the blockchain if the UAV has added a new block or there has been any update, and (c) manages the incoming messages if any. It proposes to use the open computer vision software OpenCV referred as "CV" for detecting any person on an image, by means of its pre-trained method Histogram of Oriented Gradients (HOG) with a Linear Support Vector Machine (SVM). When an intruder is detected, the UAV builds a new block filling the content with its ID, the intruder image, and the hash of the last block of the blockchain. The timestamp is also added by default when calling to the block builder. It mines the block for finding a hash that starts with a certain number of zeros determined with the "difficulty" parameter. In this way, breaking the blockchain would require to mine all the subsequent blocks, which by definition is considered very hard to break. Once the block is mined and added to the local copy of the blockchain, the UAV shares this version of the blockchain for reaching a consensus with P2P distribution scheme in which UAVs share this new blockchain version. Notice that the "manageIncomingMessages" method forwards the longest blockchain, and checks whether its block has been actually included when any. In the consensus phase, if the UAV receives a longer blockchain and its block is not present in this blockchain, then it adds the new block to the received blockchain for sharing it. For the distribution of the blockchain, each UAV shares the blockchain with the nearby UAVs (retrieved by the communications module with the operation denoted as "COM.nearby"), which dynamically change as these are moving. In this way, the current approach follows a blockchain approach conforming a chain about all the intruder observations with consensus.

As in any blockchain scenario, all nodes need to have the list of IDs of all the other authenticated nodes. Hence the list of IDs will be shared with all nodes in the distributed environment so that if any of the transactions are updated by any UAV then all the other UAVs will be able to apply the consensus mechanisms to approve or reject the transaction. The chain of IDs is distributed in various UAVs. Each block represents a reported observation of an intruder, including the ID of the UAV that reported it. Each block is linked with the next one with a cryptographic hash of the previous observation block. This list could not be altered unless altering all the subsequent blocks. Since the blockchain is shared like in a P2P network, it is really hard to alter any block, being considered secure by design. The distribution is handled by sharing the information among UAVs that are close enough to communicate when possible, propagating the information in the whole UAV network. Although the distribution may seem slow, it provides a reliable mechanism in disperse fleets of UAVs even when these cannot maintain seamless communications due to some long distances among each other when moving.

UAVs have a private key so they can authenticate their identity with asymmetric encryption. Each time a UAV observes an event (e.g. a person crossing the frontier), it communicates to its neighbors. Then, the neighbors transmit this information to other neighbors recursively and so on. Each UAV continues moving and keep transmitting the message to new UAVs for a specific period of time. The timeout of direct observers and the timeout of UAVs forwarding messages are established with two different input parameters. In order to simulate the timeouts, each UAV has two internal variables that determine the last times at which it observed a person and it was alerted by another UAV respectively.

In the trust management, if several UAV neighbors have observed the

Algorithm 1 Procedures for managing security in UAVs with a blockchain approach

```
procedure MAIN()
    while true do
        computerVision()
        if state = SharingBlockchain then
            shareBlockchain()
        manageIncomingMessages(myIncomingListMessages)
procedure COMPUTERVISION()
    image ← takeSnapshot()
    if CV.detectPerson(image) then
        intruderImage ← image
        onIntruderDetection(intruderImage)
procedure ONINTRUDERDETECTION(intruderImage)
    state ← MiningBlock
    myBlock ← new Block(myID, intruderImage, blockchain.lastBlock().hash)
    mineBlock(myBlock)
procedure MINEBLOCK(myBlock)
    myBlock.calculateHash()
    while myBlock.hash.substring(0, difficulty) <> zeros(difficulty) do
        myBlock.nonce ← myBlock.nonce + 1
        myBlock.calculateHash()
    myBlockchain.add(myBlock)
    state ← SharingBlockchain
procedure MANAGEINCOMINGMESSAGES(listMessages)
    for message ∈ listMessages do
        if message.type = Blockchain then
            blockchainReceived ← message.content
            if blockchainReceived.length > blockchain.length then
                myBlockchain ← blockchainReceived
                if myBlock = null then
                    state ← SharingBlockchain
                else if myBlockchain.contains(myBlock) then      ▷ My block is permanently saved
                    state ← Waiting
                    myBlock ← null
                else
                    mineBlock(myBlock)                           ▷ Mine again my block for adding it again
                    state ← SharingBlockchain
        if message.type = Validation then                        ▷ UAVs can check the validity of a block
            if myBlockchain.isValid(message.content)) then
                Send(message.sender, Approve)
            else
                Send(message.sender, Reject)
procedure SHAREBLOCKCHAIN()
    nearbyUAVs ← COM.nearby()
    for uav ∈ nearbyUAVs do
        send(uav,myBlockchain)
```

same intruder event, they can corroborate the information. The information that is only observed by one UAV is considered true but suspicious. The UAV that reports suspicious information is penalized, and the neighbors will gradually lose trust on this UAV. In this way, the neighbors of a compromised UAV normally detects its malicious behavior of sending false information, after an analysis period. In order to keep track of the original UAV observers from which a UAV has received messages, the latter keeps an updated list of the IDs of these original observers. It is worth mentioning that the user can enter an input parameter indicating the minimum number of direct observers for trusting the relevant information. This input parameter is set to two by default, but the user could change this value.

The trust on each UAV is managed by updating two different counter variables. The first variable (denoted as "s" of suspicious) counts the number of times that a UAV has reported an observation of an intruder and it is the only one reporting this after a time window (whose duration is referred as "w_s"). The other counter (referred as "c" of corroborated) counts the number of times that a UAV reported an observation of an intruder and any other UAV reported the same intruder considering an interval of "w_c" time. The trust is assigned to one (i.e. the maximum) if there is not a representative number of cases. Otherwise, the trust is the ratio of dividing the number of corroborated cases by the total number of cases (including suspicious ones) considering certain weighting factors. The following equation calculates the trust on each UAV:

$$t = \begin{cases} 1, & \text{if } (s+c) < m \\ (k_c \cdot c)/(k_s \cdot s + k_c \cdot c), & \text{otherwise} \end{cases} \tag{1}$$

where
s is the number of suspicious cases of a given UAV
c is the number of corroborated cases for a given UAV
k_s is the coefficient applied to the number of suspicious cases
k_c is the coefficient applied to the number of corroborated cases
m is the minimum number of cases to be considered representative

Algorithm 2 calculates the trust on a given UAV with a certain ID and a blockchain. Firstly, this algorithm counts the number of corroborations about each observation of a UAV by other UAVs. Then, it counts the number of suspicious cases (e.g. not corroborated by any other UAV when using the default threshold of corroborations) and the number of corroborated observations (at least corroborated by any other UAV if using this threshold). Finally, it calculates the trust with equation 1.

Algorithm 2 Algorithm for calculating the trust on a UAV given its ID and a blockchain

```
procedure CALCULATETRUST(id, blockchain)
    block ← blockchain.first()                              ▷ Count corroborations of each observation
    numCorroborations ← new Array(blockchain.size())
    observationIndex ← -1
    timeObsevation ← min_value
    while block <> null do
        if block.id = id then
            observationIndex ← observationIndex+1
            timeObservation[observationIndex] ← block.time
        else if (currentTime - timeObservation[observationIndex]) $\leq w_c$ then
            numCorroborations[observationIndex] ← numCorroborations[observationIndex] + 1
        block ← blockchain.next()
    s ← 0                          ▷ Count the suspicious and corroborated cases (with 's' and 'c')
    c ← 0
    thresholdCorroborations ← <Num. Direct Observers from User Interface> $-1$
    for i← 0; $i \leq$ observationIndex; i ← i+1 do
        if numCorroborations[i] $\geq$ thresholdCorroborations then
            c ← c + 1
        else if (currentTime-timeObservation[i]) $> w_s$ then
            s ← s + 1
    if $(s + c) < m$ then                                   ▷ Calculate the trust with the formula
        return 1
    else
        return $(k_c \cdot c)/(k_s \cdot s + k_c \cdot c)$
```

The coefficients k_s and k$_c$ allow one to weight how the trust is calculated indicating the relative importance of the number of suspicious cases and corroborated cases. The time window w_s is also important since it determines when considering a case as suspicious, and similarly for w_c in the corroborated cases.

Firstly, we calibrated w_s and w_c, by simulating a realistic number of UAVs with a realistic speed in a certain area. In these simulations, all the UAVs reported true observations, and we set a simulation time in which most intruders were observed at least twice. Then, we selected a value for w_c so that in most simulations the duration of intervals among detections were equals or lower than this value. After this, we assigned w_s to the same value as w_c, and we executed simulations with one hijacked UAVs reporting fake intruders, with a realistic frequency of intruders. We checked that the fake observations were not corroborated by any other UAV within the w_s time.

We assigned k_s and k_c weights for giving more importance to suspicious cases than corroborated ones. We executed different simulation scenarios combining different conditions about the existence of hijacked UAVs and/or intruders. We tried different values of k_s and k_c in order to find some values with which all correct UAVs were generally trusted with about 90% of trust or higher in long simulations.

In the UAV scenario, we need a mechanism to authorize a genuine UAV. To address this challenge, recently the Hyperledger permission blockchain was proposed (https://www.hyperledger.org/). It is open-source and based on standards, runs user-defined smart contracts, supports strong security and identity features, and uses a modular architecture with pluggable consensus protocols. Hyperledger Indy architecture is proposed to be used for establishing the identities in UAVs. For connecting to the UAV network every peer needs to obtain an enrollment certificate from an enrollment Certificate Authority (CA) that is part of the membership services. The CA will authorize a peer to connect to the network and to acquire the transaction certificates, which are needed to submit transactions. Notice that blockchain is really necessary because the presented reputation mechanism does not avoid "fake insertion" attack. In particular, if a hijacked UAV used different fake identities, then the trust mechanism would analyze this information as this UAV was several UAVs, and it would not consider malicious behaviors as representative for each fake ID. The advantage of using blockchain in a distributed environment is that compared to the traditional trust mechanisms, once the nodes receive their enrollment certificates from the CA and become part of the blockchain they don't need to re-authenticate themselves due to the inherent nature of the blockchain.

Given the assumption that there is not enough UAVs to cover all the area, we have decided to use a strategy that is difficult to be predicted by intruders. If UAVs moved deterministically, then the intruder could plan a route that avoids all the UAVs observation areas. Hence, we decided that in this approach UAVs move in a nondeterministic way, avoiding to be predicted by intruders. This UAV motion also has the advantage that each UAV has contact with many other UAVs. In this way, when a UAV starts having a malicious behavior for being hijacked, many other UAVs would notice conforming a distributed corroborated detection of the hijacked UAV, in order to exclude its information and alert the official services about it. The nondeterministic decisions were implemented following TABSAOND (a Technique for developing ABS Apps and Online tools with Nondeterministic

Decisions) [23]. In this way, a probability was assigned to the decision of changing the direction, and then this decision was simulated by comparing a random number with the threshold obtained from this probability. In addition, the rotation angle was calculated nondeterministically with certain limits. Sometimes these decisions were overridden due to the increase of area coverage or the avoidance of collisions as described later. Equation 2 calculates the default nondeterministic direction selection:

$$\alpha = \begin{cases} r_f(\beta) - (\beta/2), & \text{if } r \leq p_r \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where α is the angle of rotation (being zero going straight, a positive number turning right, and a negative turning left), β is the maximum angle of rotation, $r_f(x)$ is a random function that returns a real number between zero and the x parameter, r is a random real number in the [0, 1] interval, and p_r is the probability of changing the direction of the UAV.

In the motion of UAVs, we considered a two-dimensional space of the components x and y, since we considered the z component as a constant about the altitude regarding the terrain under surveillance. This assumption is common in the UAV literature [24]. In order to maintain area coverage and avoid collisions, the current approach uses different mechanisms of direction selection in some situations.

In order to guarantee the coverage, we discarded to use flock shapes common in some works of the UAV literature such as [24], because intruders could predict them and avoid these groups of UAVs. Instead, each UAV analyzes if there is any nearby area without UAVs, and heads this area if so. UAVs inform of their position to other UAVs so all nearby UAVs have the information about positions of each other. In this way, the current approach maintains the coverage of surveillance without leaving empty spaces for long periods. In this way, UAVs follow an implicit coordination regarding this matter.

In this approach, UAVs avoid collisions by explicit communications among UAVs. If two UAVs are too close to each other, they communicate so both adapt their direction to avoid a collision. Each UAV proposes to adjust their direction to minimize its direction change and to ensure the proper safe distance. In case two UAVs are exactly headed towards each other, then both UAVs turn right so these avoid the collision. In case, three or more UAVs are involved in a possible collision, all UAVs reduce their speeds and head to the direction with most available aerial space informing the other UAVs, so all UAVs can check there will be no collisions. Collisions rules have a higher priority than coverage rules since the negative impact of collisions is much higher.

The direction changes of each person was also calculated nondeterministically, but its maximum angle of rotation was much lower, so their path was almost straight. The speed of the person had a different value from the one for UAVs.

The methods of the Measure module allow the simulator to present the evolutions of respectively (a) the percentage of indirectly alerted UAVs considering all the UAVs (referred as a_p in equation 3), (b) the percentage of alerted UAVs that trust the messages considering only the alerted UAVs (denoted as t in equation 4), (c) the percentage of UAVs that directly observed an intruder (d_p in equation 5), and (d) the average number of direct observer UAV IDs stored locally in each alerted UAV (ids in equation 6). Equations 3-6 respectively define these metrics:

$$a_p = a/n \tag{3}$$

$$t = a_t/a \tag{4}$$

$$d_p = d/n \tag{5}$$

$$ids = \frac{\sum_{x \in A} |l_x|}{a} \tag{6}$$

where a is the number of UAVs alerted by other UAVs, n is the total number of UAVs, a_t is the number of UAVs that trust the information received by other UAVs about an intruder, d is the number of UAVs that directly observed and reported an intruder, A is the set of all the alerted UAVs, and l_x is the list of direct UAV observer IDs stored locally in the UAV x.

The hijacked UAVs move as any other UAVs. The only difference is that they report fake alerts of intruders. Their goal is to make the fleet of UAVs report false alarms, so that the system loses credibility and users may start ignoring it. In this way, a real intruder could go through the controlled area when UAV alarms are ignored.

The current framework is prepared to incorporate new attacking strategies. For this purpose, people can define new attacking strategies by creating new agent types with the "breed" NetLogo command, and implementing the necessary methods to communicate with UAV agents. If the behavior of the attacker is similar to other UAVs, then it can just introduce a new set of instructions with the behavior difference under an "if" conditional statement using "is-compromised" as condition. This should be placed within the corresponding method related to the UAV agent type. In this way, different researchers and students can test different attacking strategies in a game-like fashion. In this way, security strategies can become more robust by (1) being tested with different attacking strategies, and (2) being adapted for defending from these.

4. Experimentation

In order to assess the current approach, we performed several simulations with 100 UAVs with the novel ABS-SecurityUAV simulator presented in section 3.2. We selected 100 UAVs as this number is considered common for UAV networks as one can observe in the literature about UAV communication systems [25]. We set a time out duration of 1000 s for both V2V communications and for transmitting direct observations, since this time duration is commonly used in the literature about cooperative UAVs [26]. The trust threshold was two indicating that at least two UAVs were necessary for corroborating the information. Firstly, we run simulations for a scenario in which we assumed that a real person was crossing the controlled area. This scenario had one simulated person and zero compromised UAVs. In a second scenario, we simulated the existence of one compromised UAV sending false alerts without any person crossing the controlled area.

Figure 3 shows the results in the scenario in which a real person was crossing the controlled area. This figure shows the percentage of UAVs that were aware of this human intruder under the label "indirectly alerted". These UAVs did not directly observed the intruder, but they received the information. One can observe that this amount gradually increased for the occurring event, and reached high values in the interval 90-100%. Thus, the information spread worked properly in true positives (i.e. when an intruder entered the controller area) according to the results. The percentage of alerted UAVs that trusted this information reached initial values in the interval 40-80%, when there were several direct observers. The variability of initial period was probably due to the small sample of alerted UAVs, which reflected big changes with each change in a UAV. When the simulation continues, one

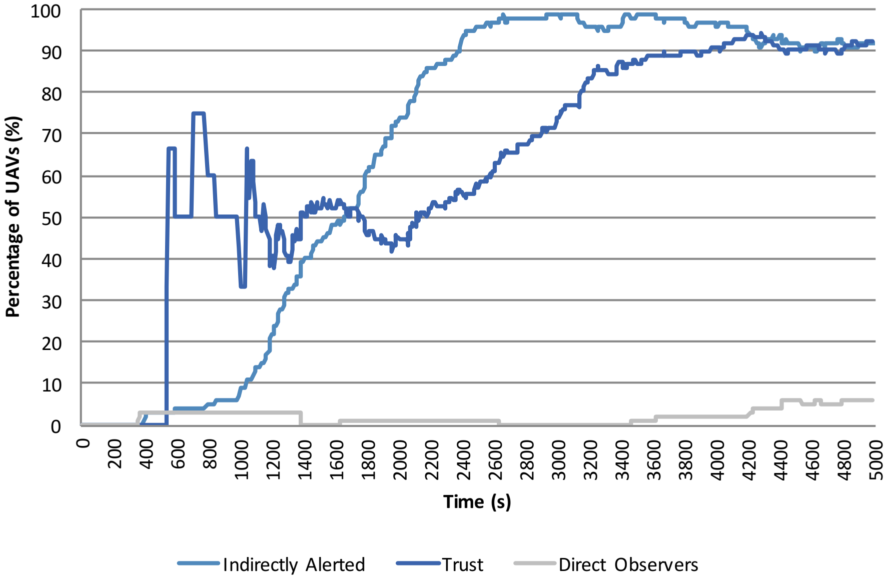


Figure 3: Results when a person was crossing the controlled area

can observe that trust increased and became stable around 90%. The figure also presents the percentage of UAVs that directly observed the intruder. One can observe that even with a relatively small percentage (i.e. in the 0-8% interval), the distributed trust properly coincided with the actual values showing a maximum around 90%.

Figure 4 shows the same information as in the previous figure, but in the scenario without any real intruder. A UAV simulated to be compromised, alerting about a false intruder detection. This false information was spread to the UAVs reaching the interval 90-100% of indirectly alerted UAVs at the end of the simulation. However, this figure shows that the trust remained as zero throughout the whole process. This information was never completely trusted, as it was never corroborated by any other direct observer. In fact, the figure also reveals that there was only one direct observer (the UAV with malicious behavior) throughout the simulation, as also observed across all the other simulations. Therefore, the current approach clearly detected the misbehavior of the compromised UAVs when alerting on a false event.

Figures 5 and 6 show the average number of the direct UAV observer IDs known by each alerted UAV. Since only the alerted UAVs were considered

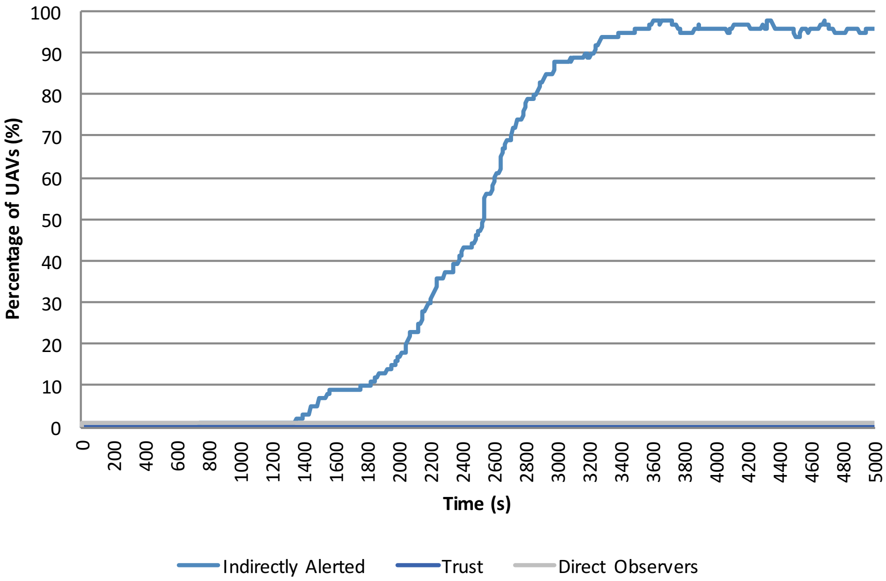


Figure 4: Results when a UAV was compromised

for this average, the least positive value was one because each UAVs was alerted at least by one. If there are no alerts, then the simulator presents a zero value. The difference between both figures is that Figure 5 presents the results of a simulation with a real intruder, while Figure 6 shows the results of a simulation without any real intruder and a compromised UAV faking alerts. One can observe that the real person was initially detected with the evolving average within [1.5, 2.0] interval. Then, when most UAVs were alerted, the propagation of the real observer alerts was spread, gradually increasing the number of UAV IDs. By contrast, in the case of fake alerts by a compromised UAVs, the number of alert IDs remained as one from the first alert. This allows the distributed system to detect the suspicious behavior over time and confirm its malicious behavior.

5. Conclusions and future work

This work has presented a security mechanism for detecting compromised UAVs in UAV networks for supporting surveillance. This mechanism is inspired by the principles of blockchain, and uses a trust policy. This work has considered different options about the behavior of malicious hijacked UAVs.

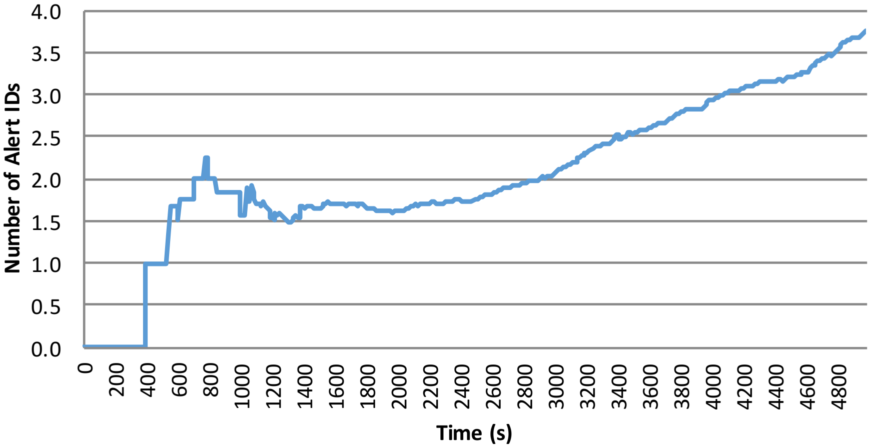


Figure 5: Average number of alert IDs when a person was crossing the controlled area

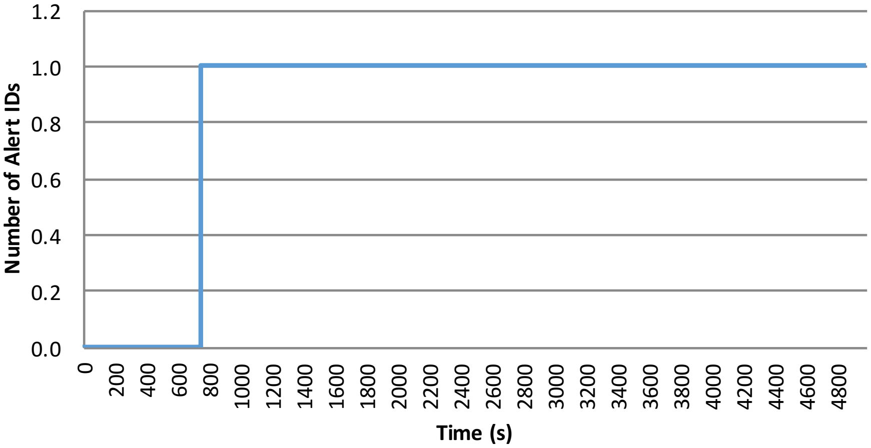


Figure 6: Average number of alert IDs when a UAV was compromised

More concretely, a compromised UAV has two options, which are either (1) to omit real observations or (2) to create noise with false alarms to discredit the system so that the real alarms are not taken seriously. If the compromised UAV omits real observations, the system is still considered reliable since other UAVs will probably detect it, assuming that only a minority of UAVs are compromised. The attacks related to the creation of false information have been studied more deeply. Signing false observation on behalf of other UAVs is not possible because this approach uses a secure signing system based on asymmetric encryption and blockchain records. A compromised UAVs could insert false information with its true identifier. This problem is addressed by a trust policy. To the best of the authors' knowledge, the current work is the first one that proposes a security mechanism for safely detecting intruders in surveillance with UAVs when a minority of the official UAVs is physically hijacked.

The experimentation of this current approach covered the simulation of cases when a person was crossing the border. One can observe that in these cases, most UAVs got informed, and the trust on the information increased, so the system was aware of and trusted this information. On the other hand, in case of having a compromised UAV without any people crossing the border, then the system realized that the compromised UAV was providing misinformation, which was reflected in the fact that the trust level kept being very low. Therefore, the current work presents a secure system for surveillance with UAVs even if a minority of these are compromised.

The current work is planned to be extended by testing this approach in real-world UAVs. In particular, we plan to apply this approach in the surveillance of schools for detecting bullying activities and reporting these to the school authorities. This work may also be tested for assisting military operations in detecting possible threats in critical borders where the terrain is difficult to monitor for military movement. The current work can also be extended by incorporating a trust policy for detecting compromised UAVs that omit to report the observed people. This trust policy could check the trajectories of UAVs in locations nearby where people were observed considering the coincidence in time. In this way, the system could detect this kind of compromised UAVs.

In addition, we plan to organize a game-like competition in which participants will be encouraged to define new attacking and defending strategies. In this competition, different combinations of attacking and defending strategies will be simulated together to rank both kinds of strategies. In this way, better defending strategies can be obtained. In addition, the resulting dataset of attacking strategies can help designers in defining more robust defending strategies afterwards.

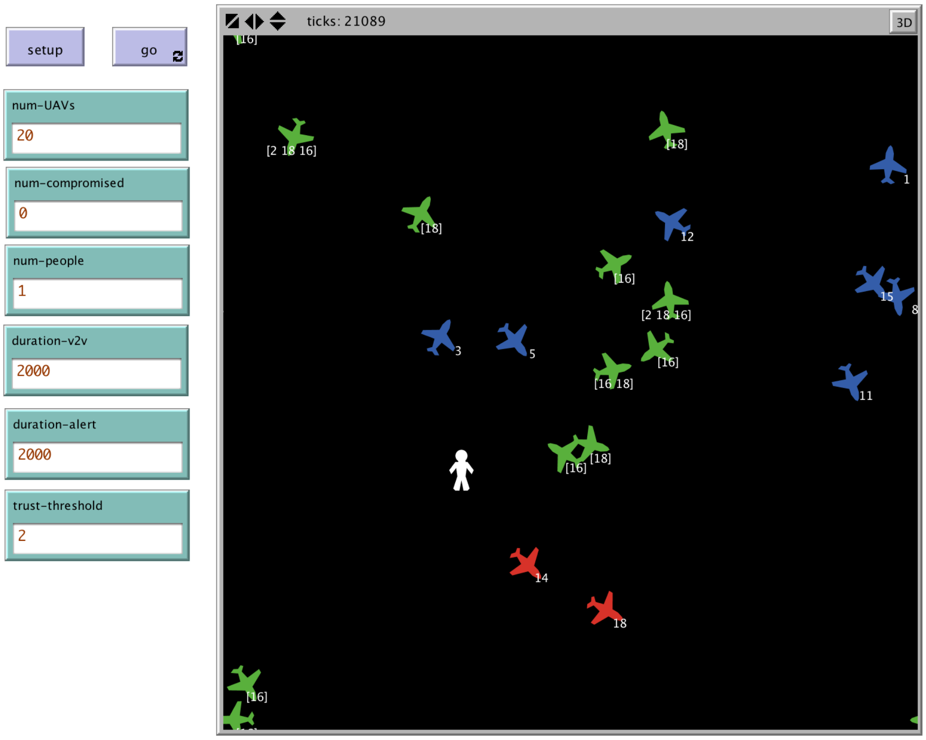
Acknowledgments

This work acknowledges the research project “Construcción de un framework para agilizar el desarrollo de aplicaciones móviles en el ámbito de la salud” funded by University of Zaragoza and Foundation Ibercaja with grant reference JIUZ-2017-TEC-03. We also acknowledge support from “Universidad de Zaragoza”, “Fundación Bancaria Ibercaja” and “Fundación CAI” in the “Programa Ibercaja-CAI de Estancias de Investigación” with reference IT1/18. This work was partially performed during the research stay of the first author in the Massachusetts General Hospital and Harvard University, funded by “Dpto. de Innovación, Investigación y Universidad del Gobierno de Aragón” through the program “FEDER Aragón 2014-2020 Construyendo Europa desde Aragón” (Ref: T49_17R). This work was also supported by the program “Estancias de movilidad en el extranjero José Castillejo para jóvenes doctores” funded by the Spanish Ministry of Education, Culture and Sport with reference CAS17/00005. We acknowledge the research project “Desarrollo Colaborativo de Soluciones AAL” with reference TIN2014-57028-R funded by the Spanish Ministry of Economy and Competitiveness. It has also been supported by “Organismo Autónomo Programas Educativos Europeos” with reference 2013-1-CZ1-GRU06-14277.

Appendix A. User interface of the novel ABS-SecurityUAV

In the left side of the UI of ABS-SecurityUAV (see Figure A.7), user can enter certain numeric input parameters in the corresponding input fields. The user can indicate the number of UAVs in the simulation, the number of compromised UAVs, the number of people crossing the controlled area, and the time-out duration for forwarding alert messages through V2V communications in the “duration-v2v” parameter, to test different scenarios. The “duration-alert” determines the time-out duration while a direct observer transmits its message to the nearby UAVs, and the “trust-threshold” parameter indicates the number of direct UAV observer IDs necessary for trusting the information. In the latter parameter, two would indicate that at least two IDs are necessary for corroborating the information. Notice that a sybil attack would require physical hijacking several official UAVs since these never share their private keys in asymmetric encryption. The trust threshold could be increased in some scenarios to augment security.

The UI has two buttons respectively labeled as “setup” and “go”. The former one allows users to establish the initial state of the simulation using



ticks: 21089

num-UAVs
20

num-compromised
0

num-people
1

duration-v2v
2000

duration-alert
2000

trust-threshold
2

Figure A.7: Main part of the user interface of ABS-SecurityUAV

the parameters indicated in the input number fields. The latter button allows both running and pausing the simulation evolution.

UI shows a graphical representation of the locations and information of the UAVs in a wide square area, as shown in the right side of Figure A.7. UAVs are represented with an airplane icon. The colors of UAVs represent different states. Blue represents the default state of flying without detecting any person. A red UAV means that it has directly observed a person. A green UAV represents that it has received a message of alert from another UAV regardless this was a direct observer or was indirectly alerted. In addition, each UAV shows a list of the direct UAV observer IDs from which it has received an alert.

In addition, ABS-SecurityUAV shows some graphs in the UI for representing the evolution of some global measurements in the simulation evolution. Figure A.8 shows these graphs for a simulation execution example. The upper graph shows the evolution of the percentage of direct observers reporting a person detection. It also presents the evolution of the percentage of UAVs indirectly alerted in the simulation. This graph also represents the evolution of average trust on a given person detection based on the local corroborations in each UAV. The lower graph represents the average number of alert IDs per each UAV that has been alerted.

References

[1] H. Hasrouny, A. E. Samhat, C. Bassil, A. Laouiti, VANet security challenges and solutions: A survey, Vehicular Communications 7 (2017) 7–20.

[2] M. Banerjee, J. Lee, K.-K. R. Choo, A blockchain future to Internet of Things security: A position paper, Digital Communications and Networks 4 (3) (2017) 149–160.

[3] X. Li, P. Jiang, T. Chen, X. Luo, Q. Wen, A survey on the security of blockchain systems, Future Generation Computer Systems (2017) https://doi.org/10.1016/j.future.2017.08.020.

[4] P. Li, H. Duan, A potential game approach to multiple UAV cooperative search and surveillance, Aerospace Science and Technology 68 (2017) 403–415.

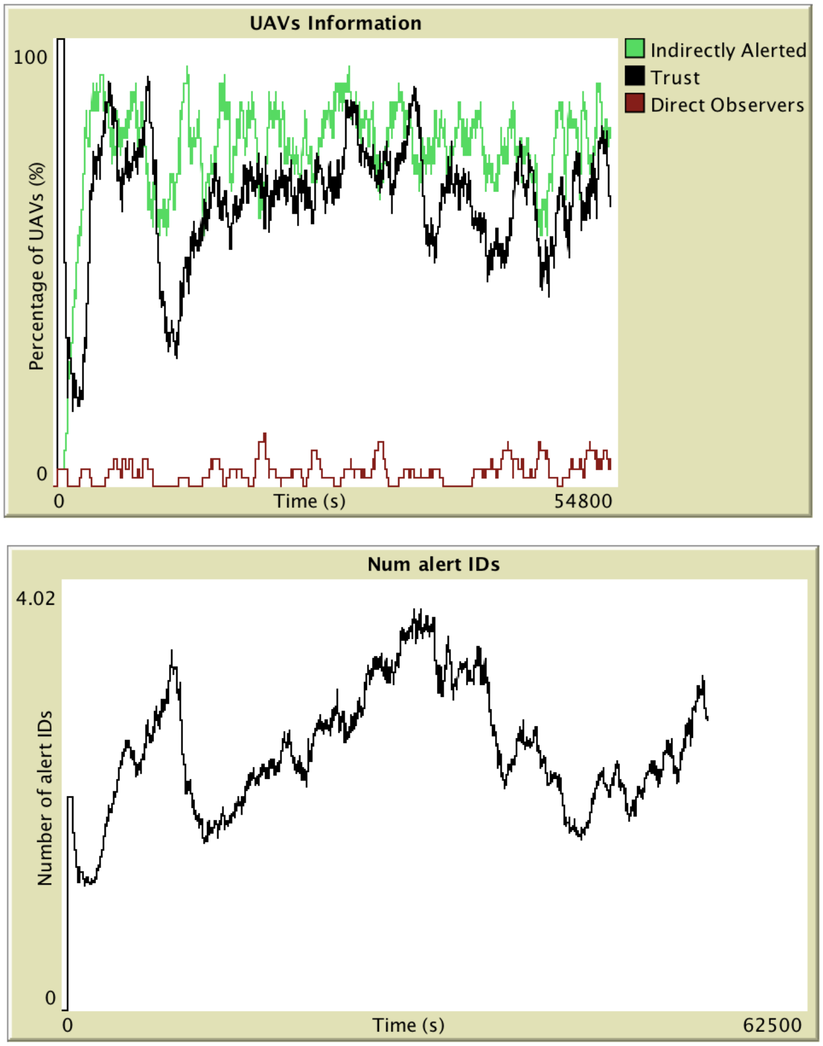


Figure A.8: Charts of the user interface of ABS-SecurityUAV

[5] M. Erdelj, O. Saif, E. Natalizio, I. Fantoni, UAVs that fly forever: Uninterrupted structural inspection through automatic UAV replacement, Ad Hoc Networks (2017) https://doi.org/10.1016/j.adhoc.2017.11.012.

[6] H. Sedjelmaci, S. M. Senouci, N. Ansari, Intrusion detection and ejection framework against lethal attacks in UAV-aided networks: A Bayesian game-theoretic methodology, IEEE Transactions on Intelligent Transportation Systems 18 (5) (2017) 1143–1153.

[7] Q. Wang, Z. Chen, W. Mei, J. Fang, Improving physical layer security using UAV-enabled mobile relaying, IEEE Wireless Communications Letters 6 (3) (2017) 310–313.

[8] S. Hayat, E. Yanmaz, R. Muzaffar, Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint, IEEE Communications Surveys & Tutorials 18 (4) (2016) 2624–2661.

[9] W. Fawaz, Effect of non-cooperative vehicles on path connectivity in vehicular networks: A theoretical analysis and UAV-based remedy, Vehicular Communications 11 (2018) 12–19.

[10] M. Sbeiti, N. Goddemeier, D. Behnke, C. Wietfeld, PASER: secure and efficient routing approach for airborne mesh networks, IEEE Transactions on Wireless Communications 15 (3) (2016) 1950–1964.

[11] C. Cambra, S. Sendra, J. Lloret, L. Parra, Ad hoc network for emergency rescue system based on unmanned aerial vehicles, Network Protocols and Algorithms 7 (4) (2016) 72–89.

[12] L. Wan, G. Han, L. Shu, N. Feng, C. Zhu, J. Lloret, Distributed parameter estimation for mobile wireless sensor network based on cloud computing in battlefield surveillance system, IEEE Access 3 (2015) 1729–1739.

[13] K. Muhammad, R. Hamza, J. Ahmad, J. Lloret, H. H. G. Wang, S. W. Baik, Secure Surveillance Framework for IoT systems using Probabilistic Image Encryption, IEEE Transactions on Industrial Informatics 14 (8) (2018) 3679–3689.

[14] I. Jawhar, N. Mohamed, J. Al-Jaroodi, D. P. Agrawal, S. Zhang, Communication and networking of UAV-based systems: Classification and

associated architectures, Journal of Network and Computer Applications 84 (2017) 93–108.

[15] N. Nigam, S. Bieniawski, I. Kroo, J. Vian, Control of multiple UAVs for persistent surveillance: algorithm and flight test results, IEEE Transactions on Control Systems Technology 20 (5) (2012) 1236–1251.

[16] D. He, S. Chan, M. Guizani, Communication security of unmanned aerial vehicles, IEEE Wireless Communications 24 (4) (2017) 134–139.

[17] C. A. Kerrache, A. Lakas, N. Lagraa, E. Barka, UAV-assisted technique for the detection of malicious and selfish nodes in VANETs, Vehicular Communications 11 (2018) 1–11.

[18] K. Yoon, D. Park, Y. Yim, K. Kim, S. K. Yang, M. Robinson, Security Authentication System Using Encrypted Channel on UAV Network, in: Robotic Computing (IRC), IEEE International Conference on, IEEE, 2017, pp. 393–398.

[19] I. García-Magariño, R. Lacuesta, ABS-TrustSDN: An agent-based simulator of trust strategies in software-defined networks, Security and Communication Networks 2017 (2017) Article ID 8575842, 9 pages, https://doi.org/10.1155/2017/8575842.

[20] M. Babiš, P. Magula, NetLogo - An alternative way of simulating mobile ad hoc networks, in: Wireless and Mobile Networking Conference (WMNC), 2012 5th Joint IFIP, IEEE, 2012, pp. 122–125.

[21] I. García-Magariño, M. Cossentino, V. Seidita, A metrics suite for evaluating agent-oriented architectures, in: Proceedings of the 2010 ACM Symposium on Applied Computing, ACM, 2010, pp. 912–919.

[22] I. García-Magariño, A. Gómez-Rodríguez, J. C. González-Moreno, G. Palacios-Navarro, PEABS: a process for developing efficient agent-based simulators, Engineering Applications of Artificial Intelligence 46 (2015) 104–112.

[23] I. García-Magariño, G. Palacios-Navarro, R. Lacuesta, TABSAOND: A technique for developing agent-based simulation apps and online tools with nondeterministic decisions, Simulation Modelling Practice and Theory 77 (2017) 84–107.

[24] M. De Benedetti, F. D'Urso, G. Fortino, F. Messina, G. Pappalardo, C. Santoro, A fault-tolerant self-organizing flocking approach for UAV aerial survey, Journal of Network and Computer Applications 96 (2017) 14–30.

[25] A. Agogino, C. HolmesParker, K. Tumer, Evolving large scale UAV communication system, in: Proceedings of the 14th annual conference on Genetic and evolutionary computation, Philadelphia, Pennsylvania, USA: ACM, 2012, pp. 1023–1030.

[26] A. E. Gil, K. M. Passino, S. Ganapathy, A. Sparks, Cooperative task scheduling for networked uninhabited air vehicles, IEEE transactions on aerospace and electronic systems 44 (2) (2008) 561–581.