IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 9, 2021, accepted May 19, 2021, date of publication June 7, 2021, date of current version June 15, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3087084

UAVouch: A Secure Identity and Location Validation Scheme for UAV-Networks

CARLOS FELIPE EMYGDIO DE MELO[1], TULIO DAPPER E SILVA[1], FELIPE BOEIRA[2], JORGITO MATIUZZI STOCCHERO[1], ALEXEY VINEL[3], (Senior Member, IEEE), MIKAEL ASPLUND[2], AND EDISON PIGNATON DE FREITAS[1], (Member, IEEE)

[1]Graduate Program in Electrical Engineering, Federal University of Rio Grande do Sul (UFRGS), Porto Alegre 90035-190, Brazil
[2]Department of Computer and Information Science, Linköping University, 58183 Linköping, Sweden
[3]School of Information Technology, Halmstad University, 30118 Halmstad, Sweden

Corresponding author: Carlos Felipe Emygdio de Melo (cfemelo@gmail.com)

This work was supported in part by the Knowledge Foundation (KKS) in the framework of "Safety of Connected Intelligent Vehicles in Smart Cities–SafeSmart" Project (2019–2023), in part by the Swedish Innovation Agency [Verket För Innovationssystem (VINNOVA)] in the framework of "Emergency Vehicle Traffic Light Pre-emption in Cities–EPIC" Project (2020–2022), in part by the Excellence Center Linköping–Lund in Information Technology (ELLIIT) Strategic Research Network, in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brazil (CAPES)—Finance Code 001, and in part by the Conselho Nacional de Desenvolvimento Científico e Tecnológico—Brazil (CNPq) under Project 309505/2020-8 and Project 420109/2018-8.

ABSTRACT Emerging surveillance applications of UAV teams rely on secure communication to exchange information, coordinate their movements, and fulfill mission objectives. Protecting the network by identifying malicious nodes that are trying to disturb the system is an important task, particularly in the military domain. This paper presents the design and evaluation of UAVouch, an identity and location validation scheme that combines a public-key based authentication mechanism with a movement plausibility check for groups of UAVs. The key idea of UAVouch is to supplement the authentication mechanism by periodically checking the plausibility of the locations of neighboring UAVs, allowing the detection of intruders that are unable to follow expected trajectories. The proposed solution was evaluated in a simulated military surveillance scenario in which it detected malicious nodes' position falsification attacks with an average accuracy of above 85%.

INDEX TERMS Distributed applications, inter-drone communications, Sybil attack, security protocols, unmanned aerial vehicles.

I. INTRODUCTION

In the last few years, Unmanned Aerial Vehicles (UAVs), also known as drones, have been used in several emerging applications in both civil and military domains. According to data from the Brazilian National Agency of Civil Aviation (ANAC), the number of registered drones for professional use grew by approximately 233% between 2017 and 2019 [1]. This number is even higher considering the market for drones worldwide [2]. Along with the growth in the number of drones, the number of applications for drones has also seen a significant increase. Some well-known applications of drone-based systems are surveillance, filmmaking, disaster management, and defense [3].

Although drones are becoming more common in civilian applications, military applications are still dominant, and drones represent an essential asset in the modern battlefield [4]. In applications such as surveillance or military reconnaissance, groups of UAVs can be used to provide awareness of threats ahead of the troop's line of sight. However, connecting multiple UAVs through ad hoc networks raises vulnerability issues, and enemy threats must be addressed in advance. For instance, attacks such as sinkhole, spoofing, eavesdropping, impersonation, and Sybil could potentially ruin a mission [5]–[7].

In an impersonation attack, the attacker manages to successfully masquerade itself as one of the legitimate parties [8]. A Sybil attack takes place when a malicious node impersonates or create multiple identities [9]. Impersonation and Sybil attacks, when successfully executed, give the intruder the ability to launch other kinds of attacks, such as information manipulation and Denial of Service (DoS) [10]. A Sybil node could be used to manipulate the position information exchanged among nodes in a vehicular or drone

The associate editor coordinating the review of this manuscript and approving it for publication was Shadi Aljawarneh.

network, for instance, in an attempt to cause a collision, or simply to separate a specific node from its network to steal its information or technology.

The canonical approach for controlling access to a protected network is through the use of public key infrastructure (PKI). In these schemes, a centralized entity can distribute certificates to legitimate users and devices, who can then use the certificates to authenticate themselves to other members of the network. This provides a basic level of security, but it does not protect the system against insider attacks, wherein the intruder has gained access to a valid certificate. Mechanisms like the ones presented in [11], [12] are capable of detecting malicious activity in vehicular networks, a domain that has been extensively studied with respect to security concerns [13], [14]. However, only a few works have covered the area of intrusion detection for drone networks [7], several of which present artificial intelligence or computer-vision based solutions, which tend to be resource consuming and are thus not ideal for resource-constrained drones.

This paper presents a novel approach - UAVouch - that combines the use of public-key authentication with location validation. The idea of using physical location and movement as an authentication mechanism is not in itself new: there are several works that make use of this idea in the vehicular networking domain [12], [15]. However, UAVouch, which is specifically designed for collaborative drone applications, provides some interesting properties that have not been previously studied and described in the literature. First, it presents a fully distributed group management mechanism in which an existing group (called a cell) collectively determines whether a joining node should be admitted. Second, once a cell has been formed, the nodes in the cell continue to control each other's movement patterns to ensure that everyone is behaving as expected. This position validation mechanism can be seen as a complement to cryptographic methods and as a form of anomaly detection (using node mobility as the feature set rather than data traffic, as is more common in the literature). Provided sufficiently complex mobility patterns, it will be hard for an attacker to guess where other nodes expect it to move to. Finally, the proposed approach also supports trusted communication between different cells.

To test the proposed scheme, an application scenario, illustrated in Figure 1, was designed using military units composed of an armored ground vehicle escorted by a number of drones. The purpose of the drones is to monitor an area that is out of sight from the ground vehicle. These units (the ground vehicles with their escorting drones) form cells. The drones fly around the armored ground vehicle at distances that maintain the wireless connection with the vehicle on the ground and an intermittent connection with one or more of the other drones, forming an ad hoc network to enable the exchange of data between themselves and with the vehicle on the ground.

The UAVouch proposal is evaluated using this setting in a simulation platform based on INET and OMNet++. Two attack scenarios are defined, one that considers an intruder within a cell, and another where the attack comes from a neighboring cell. A basic mobility model for the drones is considered, which is assumed not to be known to the attackers. The results show that under this assumption, UAVouch allows detection of the intruders with high accuracy. The location validation itself is very cost-effective since it does not require any computationally demanding operations. The main trade-off is associated with added messaging due to transmission of location messages among the nodes.

FIGURE 1. Illustration of the application scenario.

The key contributions of this work can be summarized as follows:

- A distributed identity and position validation mechanism that allows a cell to authenticate and verify the position of a drone without any infrastructure support;
- An assessment of the proposed mechanism for the application scenario that includes the detection performance and overhead analysis against impersonation and Sybil attacks using a realistic experimental setup in a network simulator.

The rest of this paper is organized as follows. Section II discusses related works. Section III formulates the problem, describing the attack scenarios. Section IV presents the overall UAVouch scheme, describing the proposed location validation mechanism in detail. Section V presents the experiments that were used to validate the proposed solution along with a discussion of the acquired results, while Section VI concludes the paper, providing insights for future works.

II. RELATED WORK

The open nature of wireless networks makes them more susceptible to cyberattacks than wired ones [16]. To mitigate this risk in mobile ad hoc networks, particularly in VANETs, different approaches exploring single or combined security mechanisms have been proposed. This section describes state-of-the-art research in this field, with particular attention to techniques applied to UAV networks. The literature review is organized into two major categories: the first addresses works about authentication mechanisms, and the second addresses works about position verification mechanisms. In each of these categories, the articles were also organized according to the type of network they targeted, from MANETs to FANETs. The end of this section presents a summarized comparison of UAVouch and the reviewed approaches.

A. REVIEWED AUTHENTICATION MECHANISMS

In [17], the authors proposed a novel technique called accurate prevention and detection of jelly-fish attack detection (APD-JFAD) in mobile ad-hoc networks (MANETs). The jelly-fish attack is a type of DoS attack, one of the most serious attack categories that affect the normal operation of MANETs. The proposal combines an authenticated routing-based framework and a Support Vector Machine (SVM) based technique to detect the malicious behavior of nodes by observing the quality of packets that reach the destination. APD-JFAD is tailored for MANETs, which are composed of nodes with lower speed and lower degrees of mobility than drones. This significantly affects the network topology and communication, resulting in a negative impact on the performance of the proposed mechanism.

The use of blockchain for an authentication mechanism was explored in [18]–[20] through the use of different versions of blockchain, such as public versions as Bitcoin [18] and Ethereum [20], and a private version named Hyperledge Fabric [19]. In the blockchain encryption scheme, techniques such as public-key cryptography and digital signatures are accepted means for proving the identities of specific agents in a swarm of robots [18] or in a swarm of UAVs [19], [20].

Although blockchain technology can provide data confidentiality and entity validation for a drone swarm, making it suitable for trust-sensitive applications has its limitations. If a large number of robots are deployed for a very long time, the blockchain could be expanded to a point where the agents would not be capable of maintaining a copy of the full ledger anymore. Also, the time to process a new block could take an average of 10 minutes, which would be not cost-efficient considering its usage in UAVs that have an average of around 25 minutes of flight autonomy.

The high mobility of flying nodes brings new challenges to the current security protocols applied in general mobile networks. In [21] the authors propose a fast and secure group key establishment protocol to facilitate forming groups and guaranteeing key freshness, key confidentiality, and member authentication. Their proposed protocol consists of two phases: initialization and post-deployment. During the initialization phase, individual security components are loaded into the UAVs, including their IDs, public and private keys, and their signatures. After that, an exchange of encrypted and signed request and joining messages is performed to allow a member to join a group by providing a group key through a secure and private channel. The authors have proven protocol robustness by a complete analysis of their proposed mechanism. However, it was not implemented in either a simulated or a real environment, thus failing to demonstrate whether or not the proposal is feasible for use in resource-constrained UAV-networks.

In [10], the authors focused their work on presenting an authentication mechanism that was designed to detect malicious nodes in Flying Ad-hoc Networks (FANETs). The malicious node used a Sybil attack to trigger a Distributed Denial of Service (DDoS) attack. During network initialization, the central unit controller (CUC) sends Internet Control Messages Protocol (ICMP) packets to all nodes. These nodes then reply to the ICMP packets and send information about their neighbors to the CUC, which starts analyzing it. If two nodes have the same identification, but different neighbors, then the CUC marks them as intruders and starts monitoring their identifications. The node that changes its identification will be marked as malicious and will also be held responsible for the DDoS attack. In simulations, the authors have shown that this method generates maximum throughput compared to other methods, while also generating less routing overhead and packet loss. Nevertheless, the paper lacks a complete explanation of the authentication mechanism, which makes the results difficult to replicate.

Securing a network of drones through the authentication mechanism is also addressed in [22]. The authors present *i*TCALAS, an improved scheme based on a temporal credential-based anonymous lightweight authentication scheme (TCALAS [23]) for the Internet of Drones (IoD). *i*TCALAS uses lightweight symmetric key primitives and temporal credentials. Despite promising results, the authentication schemes use a centralized ground station server, thereby exposing the solution to a single point of failure.

B. REVIEWED POSITION VERIFICATION MECHANISMS

Recently, there has been an increase in the number of location-based applications, many of which provide rewards to the user for visiting a specific venue. This also creates an incentive for dishonest users to falsify their position to get undeserved rewards. To solve this issue, the work reported in [24] proposed *SPARSE*, a distributed mechanism that provides secure and private Location Proof (LP) generation and verification for mobile users. In this mechanism, the system performs a witness selection mechanism by which some witnesses are chosen and qualified to generate LPs for a specific prover. The proof is then assessed and verified by an authorized entity known as the verifier.

A similar approach is presented in [25]. In this work, the authors propose a decentralized witness-based proof-of-location system for mobile devices. The system relies on different techniques for location estimation and on witness devices to confirm the presence of the user's device. The proposed solution was implemented in Huawei P9 Lite devices. Although the results were promising, both solutions are highly dependent on a high density of witnesses, which is not ideal for FANETs since they can potentially have a low density of nodes [26]. Moreover, the solution presented in [25] requires a considerable number of packets to accurately determine the node position, which is not suitable for a high mobility environment with sudden disconnections, packet loss, and permanent network partitioning [27].

In the context of VANETs, the work reported in [28] detailed the dangerous implications of a Sybil attack against a vehicular platoon. With this attack, a malicious node manages to introduce falsified vehicle identities into the platoon. In [11] the authors present a countermeasure named

Vouch, which is a proof-of-location mechanism tailored for VANETs. Vouch uses a centralized proof-of-location and plausibility system to detect a Sybil attack in a vehicular platoon. A vehicle that requires a proof of its location, called a prover, asks a Road Side Unit (RSU), which is called the proof provider, for a proof of location. Once the prover has received the signed proof from the proof provider, it then broadcasts it along with the position beacon to the other vehicles in the platoon. The other vehicles are called verifiers. The verifiers then use this proof of location to estimate the prover's location in subsequent beacons and verify if the position sent by the prover is plausible or not. The proposed solution is not ideal for the military domain, as addressed in this current paper, because it has a single point of failure due to the centralized approach based on the RSU. If the RSU is destroyed, the mechanism would not work. Furthermore, even with the assumption that the proof provider (RSU) cannot be destroyed, it cannot always be considered to be reliable because it could be compromised, in which case the entire system would be compromised.

In [12] *Vouch+* is introduced, which is an improvement on Vouch [11]. Instead of depending exclusively on previously installed roadside infrastructure (RSUs), Vouch+ presents a decentralized protocol for the obtention of the proof of location. In Vouch the only trusted proof provider is the RSU, but in Vouch+, along with the RSUs, a vehicle (proof provider) in the vicinity can also assess the location of the prover (the vehicle that asks for the proof of location). This decentralized approach allows vehicles to prove their locations to neighbors beyond their sensing range. Although the presented mechanism is an enhanced version of the Vouch, it does not eliminate the single point of failure related to the proof provider, because it also assumes that the entity (RSU or nearby vehicle) that will provide the proof of location is not compromised. Another important difference is that Vouch+ assumes that the proof provider vehicle has a certain type of sensor to assess the position of the prover.

The reviewed works presenting authentication and position verification mechanisms are not suitable for FANETs due to the combined high mobility, node density, and privacy constraints. To cope with these requirements, this work develops and assesses a FANET-tailored identity and location verification mechanism. UAVouch was designed to support these requirements without overloading the communication channel. The combination of these mechanisms in the proposed scheme is proven to effectively detect position falsification attacks. Table 1 summarizes the comparison of this proposal and the analyzed related work.

III. ATTACK SCENARIOS

The challenges related to the management of bandwidth, latency, and battery power restriction faced by employing resource-constrained devices, like drones, for real-time video stream applications such as surveillance or military reconnaissance missions, are extensively addressed in the literature. However, the vast majority of these works focus on solving the problems associated with these restrictions [30]–[35], leaving aside the security challenges in designing multi-UAV applications [36], [37]. Especially in military applications, securing the network is of prime importance. The security mechanism for this type of application must be efficient so that the resulting overhead does not negatively impact the performance of the ultimate mission goal, i.e., video streaming.

Consider a military reconnaissance mission performed by a military cell composed of an armored ground vehicle and several drones that are circulating around the armored vehicle. The line of sight of the crew inside the armored ground vehicle might be limited by different factors, such as vegetation, and uneven terrain topology.

When the drones are placed as shown in Figure 2a, they extend the crew's ability to monitor their surroundings. In this setup, the drones should fly at altitudes that, combined with their horizontal distance to the armored vehicle, keep them within the communication range of the armored ground vehicle. It is important to emphasize that they are completely independent of each other in their movement. Most of the time, they cannot communicate with all the other drones in the network, as illustrated in Figure 2b, which means that they mostly have intermittent connections with their direct neighbors. Exploring this setting, two attack scenarios are described in the following.

A. SCENARIO 1

In the first scenario, the threat model is composed of an attacker that impersonates an authentic drone in the target cell. The sequence of events in this attack is represented in Figure 3. First, the malicious drone approaches a distant drone in the cell as represented in Figure 3a. The legitimate drone is then captured through a physical attack and has its credentials stolen [5], as represented in Figure 3b. The malicious drone uses the stolen credentials to assume the identity of the legitimate drone, returning to the network to start disseminating deceitful information. However, it is assumed that the attacker will not be able to fully replicate the future mobility of the captured drone. Exactly replicating the mobility of a captured drone can be difficult due to physical characteristics of the device, or because the pattern itself is stored in volatile memory, so once the drone is down, this information is lost.

B. SCENARIO 2

The second scenario considers a more challenging situation involving more than one cell. While cell 1 is progressing from one location to another, it could encounter and interact with other cells to exchange information and expand their exploration and/or surveillance range as represented in Figure 4. The bridge formed by one drone from each cell is the communication path through which the information from cell 1 will be transmitted to cell 2 and vice versa. An attacker can take advantage of this feature to impersonate an entire cell and disseminate deceitful information. The attacked cell

TABLE 1. Summarization of authentication and position verification proposals.

Related works	Addressed problem	Network	Architecture	Proposed mechanism	Addressed attack
Islam*et al.* (2016) [21]	High degree of mobility and fast topology changes	FANETs	Centralized	Authentication	Sybil
Nosouhi *et al.* (2018) [24]	User's privacy preservation and position verification scheme design	MANET	Distributed	Position verification	-
Ferreira; Pardal (2018) [25]	Position verification scheme design	MANET	Decentralized	Position verification	-
Doss *et al.* (2018) [17]	Detection of one kind of DoS attack (Jelly-Fish)	MANETs	-	Authentication	Jelly Fish (DoS)
Boeira; Asplund; Barcellos (2018) [11]	High mobility and user's privacy preservation	VANETs	Centralized	Position verification	Sybil
Walia; Bathia; Kaur (2018) [10]	Detection of Sybil nodes	FANETs	Centralized	Authentication	Sybil and DDoS
Boeira; Asplund; Barcellos (2019) [12]	Location assurance in cooperative transportation systems	VANETs	Decentralized	Position verification	Sybil
Aggarwal *et al.* (2019) [20]	Privacy and security issues in the Internet of Drones (IoD)	FANETs	Decentralized	Authentication	-
Ferrer (2019) [18]	Trustful identification among swarm members	FANETs	Distributed	Authentication	-
Rodrigues *et al.* (2019) [29]	Security strategies for resource constraint devices	FANETs	-	Authentication	Sybil, DoS and impersonation
Ali *et al.* (2020) [22]	Securing drones and sensitive data collected in IoD	FANETs	Centralized	Authentication	Multiples
UAVouch This proposal	**Identification of malicious nodes access in UAV network**	**FANETs**	**Distributed**	**Authentication and Position validation**	**Sybil and impersonation**

could then be redirected into a trap due to the deceitful data and have its technology stolen. This scenario involves a Sybil attack, which is represented in Figure 5. In the Sybil attack, the malicious node would impersonate more than one drone. This is possible either in the case in which the drone has one or more additional radios or in the case in which the attacker sends packets that claim they come from different nodes.

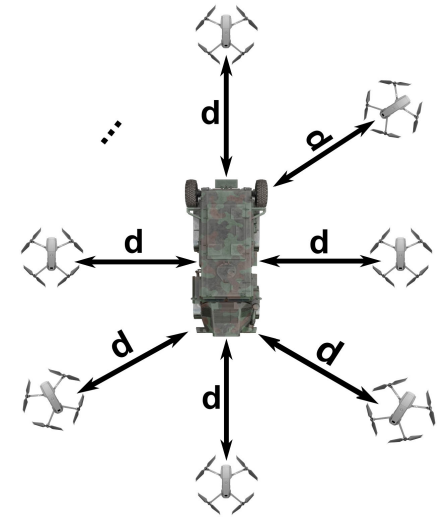
In this second scenario, a malicious drone takes advantage of the stolen identity of a drone from another cell, for instance, cell 2 in Figure 5, to approach cell 1, as represented in Figure 5a. The malicious drone then uses the stolen identity to authenticate itself with cell 1 and to get their session key, as illustrated in Figure 5b. After it manages to establish communication with cell 1, the malicious node makes it look like this cell is connected to the legitimate cell 2, so it impersonates all of the drones in cell 2 to make the attack more convincing, as represented in Figure 5c.

IV. DRONE IDENTITY AND LOCATION VALIDATION

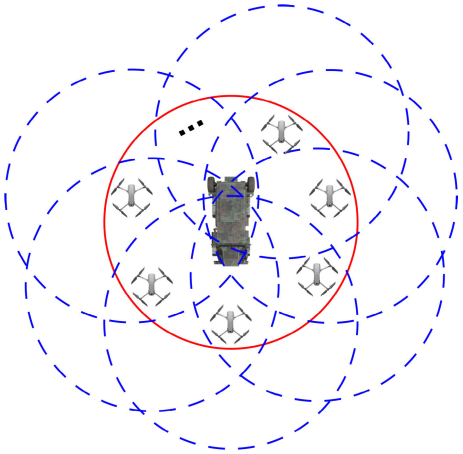
Security defense mechanisms are often classified into three categories, *prevention*, *detection*, and *response* (see e.g.,. [38]). Even though prevention strategies are necessary, attackers with enough resources can bypass these mechanisms. Thus, detection strategies are also needed to identify anomalous behavior and attacks on the system. Response mechanisms should be activated when an attack was successful, providing measures to mitigate the damages. This section describes the design of the proposed solution for drone identity and position validation. The solution is divided into a prevention strategy composed of a *public-key based authentication* **mechanism** and a detection strategy composed of a position validation **mechanism** that includes a **protocol** used for position validation, as well as a **classifier model** to detect inconsistencies in the movements of the nodes. This proposal is named *UAVouch*, which is both a reference to Vouch [11], an approach proposed to address Sybil attacks in a platoon of ground vehicles traveling on roads, and to the drones, as they are UAVs. The security features provided by this proposal are authentication and malicious node detection by position verification.

A. UAVouch SCHEME OVERVIEW

Figure 6 shows how entities interact with each other in the UAVouch scheme. Figure 6a illustrates the interaction among the entities in the authentication mechanism and Figure 6b specifically represents the interaction between the entities in

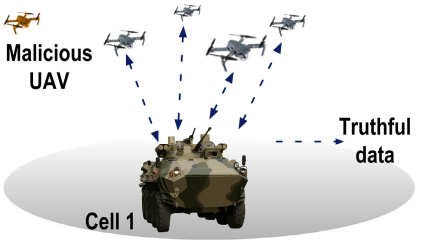


(a) Illustration of the drones positioning in relation to the armored vehicle

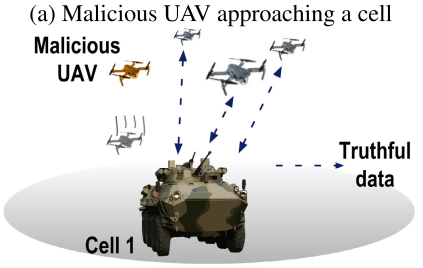


(b) Communication range illustration

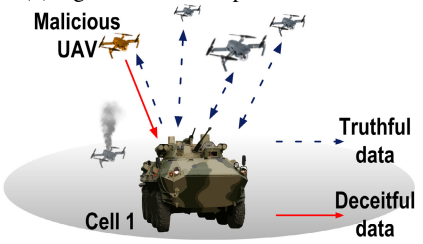
FIGURE 2. Scenario structure.



(a) Malicious UAV approaching a cell



(b) legitimate UAV captured/shot down



(c) Impersonation attack successfully concluded

FIGURE 3. Impersonation attack illustration.

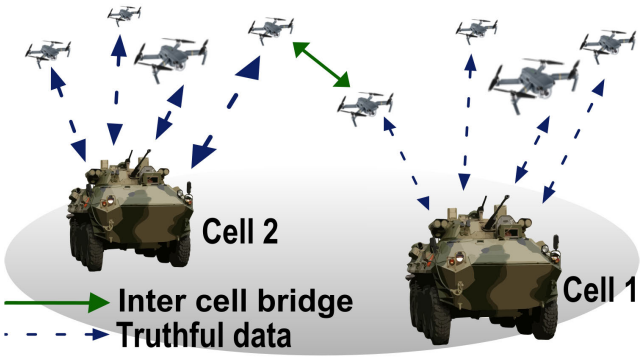
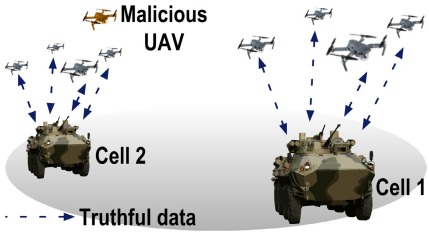


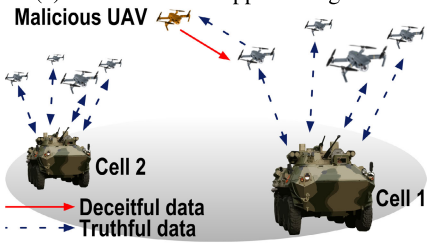
FIGURE 4. Communication between two cells.

the validation protocol, which is composed of a position validation mechanism together with a classifier model. Firstly, in the authentication mechanism, the *requester* is a drone that requests to join a cell of which it is currently not a member. The request to join the cell is received by one of several *verifiers*, which are the entities responsible for ensuring that the *requester* is authorized to join the network. The *verifier* that receives the request performs the authentication check and broadcasts its decision. This is received by the other verifiers in the cell, who also broadcast their own decisions. At the end of the chain, the *evaluator* is the entity responsible for counting the votes, and if the majority of the *verifiers* in the cell vote to admit the *requester* into the cell, the *evaluator* will send the session key to the *requester*, concluding the authentication mechanism. All drones are *verifiers* in their cell, but the drone that receives the request directly from the *requester* will also become an *evaluator*. The purpose of the authentication mechanism is to prevent intruders from entering the cell, and also to provide a secure way to identify a friendly cell traveling nearby. This is of paramount importance in a military scenario. If a cell erroneously connects to an enemy cell, mistaking it for a friendly one, the consequences could range from disclosing confidential intelligence information to losses of human lives.

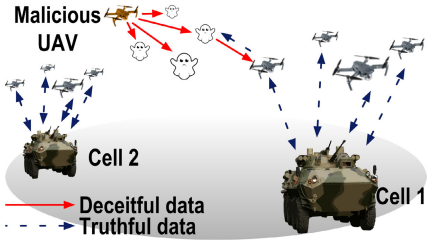
In the position validation mechanism, the UAVs continuously send their location information to each other through *pose packets*. In addition to a common header, the pose packet usually carries information about position coordinates and direction of movement (*pose* parameter), and can also carry other types of information such as speed and acceleration, depending on how the protocol was designed. A pose packet



(a) Malicious UAV approaching a cell

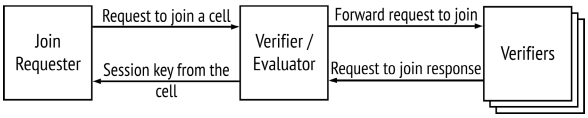


(b) Malicious UAV authenticates as a legitimate UAV

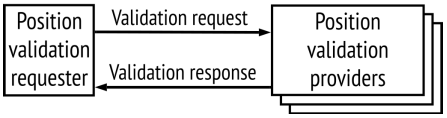


(c) Sybil attack successfully implemented

FIGURE 5. Sybil attack illustration.



(a) Entities involved in the authentication scheme



(b) Entities involved in the validation protocol

FIGURE 6. Relation between the entities in the UAVouch.

can optionally include a position validation request. If the pose packet includes a position validation request, this will trigger a position validation protocol. In this protocol, a *position validation provider* is responsible for validating position validation requests and replying with a verdict regarding whether the transmitted location was legitimate. All drones inside a cell are position validation providers for the other members of the same cell. Once the *position validation requester* has received more than 50% of the replies from the remaining drones in the cell, it will consider that everyone has computed and stored its position validation. If the pose packet does not include a position validation request, a classifier model is activated that judges whether the claimed location is plausible given the drone's previous locations. Both mechanisms will be described in more detail below.

B. PREMISES, ASSUMPTIONS, AND NOTATION

The proposed scheme was developed based upon a few premises and assumptions:

- *Security:* It is assumed that a drone receives its asymmetric key pair, all of the asymmetric public keys from all previously registered drones, and a unique session key from its cell in a secure environment during network initialization (e.g. during mission initialization at the base);
- *Inter-Cell Communication:* There is an exclusive communication channel between the armored ground vehicles that has a larger range than the channel between the drones and drone-to-armored vehicle, which is important to make possible the detection of attacks in which a malicious drone impersonates a whole cell (Section 3.2);
- *Intra-Cell Data Forwarding:* Every received packet is forwarded to the rest of the network. The number of hops is determined based on the number of drones and the topology of the network, thereby enabling the distributed position validation protocol;
- *Positioning:* It is assumed that the drones from each cell periodically receive the updated position of their cell's armored ground vehicle and the offset of the other drones from the ground vehicle. The position update rate is the same as the rate of position validation requests. This allows the UAVs to keep track of the movement of the whole cell;
- *Flight Pattern:* The drones exhibit flight patterns that are hard for an outsider to predict and mimic. This could, for example, be achieved through a combination of complex trajectories and specific physical dynamics of the drones. The complexity of the movement pattern is arbitrary, but the more complex it is, the harder it will be for an attacker to predict, even one employing advanced learning mechanisms, such as [39].

Regarding the notation used in this section, as presented in Table 2, the asymmetric public and private keys from an entity X are represented respectively as pk_X and sk_X, and the symmetric key from a given cell X is represented as k_X. The signature process is represented using sign(m, y), where m is the message, and the y is the key used to sign the message. The encryption operation is represented by aenc(x, y) for asymmetric encryption of data x with key y and senc(x, y) for symmetric encryption of data x with key y. Table 3 presents the cryptographic notations used in both authentication and

TABLE 2. Cryptographic notations.

Notation	Description
pk_X	asymmetric public key from entity X
sk_X	asymmetric private key from entity X
k_Y	symmetric key from cell Y
$\mathrm{sign}(m, y)$	signature process of data m using key y
$\mathrm{aenc}(x, y)$	asymmetric encryption of data x using key y
$\mathrm{senc}(x, y)$	symmetric encryption of data x using key y

TABLE 3. Cryptographic operations.

Symbols	Description
msgKind	The type of the message
nId_x	The unique identification of node x
t_x	Timestamp of entity x
seqNumber	Sequence number of the message
cell	The cell in which the drone is in
whosReq	Requester to join the network
isAuth	Authentication request response
pose	Quaternion containing coodinates x, y e z and orientation w
whosValReq	Position validation requester
valReply	Position validation reply
header	$<$ msgKind, nodeId, timestamp, seqNumber, cell $>$
idenHeader	$<$ pk_B, timestamp $>$
iden	$<$ idenHeader, sign(idenHeader, sk_B) $>$
idenResponse	$<$ aenc($<$ header, sign(header,sk_A) $>$, pk_B) $>$
reqJoin	$<$ aenc($<$ header, sign(header, sk_B) $>$, pk_A) $>$
reqJoinFwd	$<$ senc($<$ header, whosReq $>$, k_A) $>$
replyRequestJoin	$<$ senc($<$ header, whosReq, isAuth $>$, k_A) $>$
joinResponse	$<$ aenc($<$ header, k_A $>$, pk_B) $>$
posePkt	$<$ senc($<$ header, pose $>$, k_A) $>$
valReqReply	$<$ senc($<$ header, valReply, whosValReq $>$, k_A) $>$
verifySig()	Verify signature sign(y, x) of data using pk_B
voteCounting()	Authentication request response counting
validityCheck()	Execute the position validation calculation
valCounting()	Validation check responses counting
savePosVal()	Store valid position calculate

position validation mechanisms. Next, the proposed mechanisms are presented in detail.

C. AUTHENTICATION MECHANISM

To simplify the presentation of the scheme, consider a scenario in which a cell A enters the communication range of cell B. The authentication mechanism is triggered when a drone d_B belonging to cell B receives a message from another drone d_A that belongs to cell A. The drone d_B (*Join requester*) then sends an identification packet (**iden**) carrying its public key (pk_B) and a timestamp of the message. The message m is signed (sign($\langle m, \text{pk}_B \rangle$, sk_B)) using d_B's private key, sk_B. The signature is verified by d_A (*Verifier/Evaluator*), and if the signature is valid it sends a response packet (*idenResponse*) with all the header information signed using sk_A and encrypted using pk_B. The signature in *idenResponse* is then verified by d_B, and only if the signature is valid will it send a **reqJoin** packet to d_A requesting to join its network, as illustrated in Figure 7. The message is signed by d_B using sk_B and encrypted using the public key of d_A. Since d_A received the **reqJoin** packet directly from d_B, it should forward the packet (**reqJoinFwd**) to its cell, adding the *whosReq* parameter so that the other drones inside the cell know that they are not receiving that packet directly from d_B. The **reqJoinFwd** message is encrypted using the session key k_A.

Every drone in cell A (*Verifiers*) will verify if d_B is an authorized drone by checking the signature in **reqJoinFwd** using the pk_B key acquired during network initialization. They then broadcast their decisions to the network, sending a **reqJoinReply** packet, where the parameter *isAuth* states whether d_B is authorized or not. Due to the fact that at any given moment a cell could be handling multiple requests to join, the *whosReq* parameter in the **reqJoinReply** packet is used to identify whose request they are replying to.

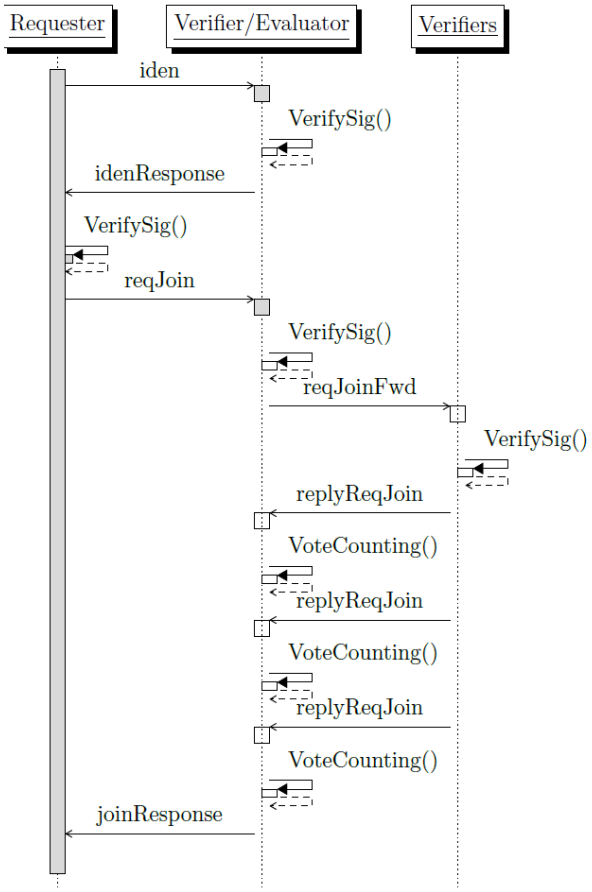


FIGURE 7. Interaction between entities in authentication model.

If more than 50% of the network confirms the legitimacy of d_B, then a **sessionKey** packet carrying k_A is sent to d_B, the one who requested to join the network, otherwise the node is ignored by cell A. As d_A is the closest drone to d_B, it will also act as an *evaluator*, which means that, after verifying that more than 50% of the cell A considers d_B legitimate, d_A will be responsible for sending the **sessionKey** packet to d_B. Considering that packet collision may occur during this process, if d_B has not been granted access to the network in cell A after a period of time t_{Auth}, it will resend its request to join the network. If d_A is compromised, then the whole authentication mechanism is also compromised. To avoid this problem, a position validation mechanism is used to identify the intruder and stop it before it can harm the network as described in the following subsection.

D. POSITION VALIDATION MECHANISM

The proposed position validation mechanism is composed of a validation protocol, which determines the interaction between the entities in the validation process, and a classifier model, which determines the position plausibility of pose packets that do not contain a validation request. The details of these two parts of the mechanism are as follows.

1) VALIDATION PROTOCOL

The validation protocol is illustrated in Figure 8. When a drone sends a pose packet (**posePkt**), it can also request validation of its location from the recipients. The *msgKind* parameter is used to identify whether or not a position validation was requested. If a position validation was requested in the *posePkt* (**posePkt$_{\mathbf{valReq}}$**), the other drones in the same cell verify the validity of the position based on the position ($\mathbf{av}_{pos}$) and heading angle θ of the armored vehicle ($\theta_{\mathbf{av}}$), on the offset (**OS**), and on the mobility model of the drone that asked for the position validation. After calculating the position validation, the drones then send a reply (**valReqReply**) containing information about the validity of the position (*valReply*) and from whom the position validation request came (*whosValReq*). The requester counts the votes, and if the majority of the network voted that the position is valid then the requester will consider that everyone has its position validation. Otherwise, the requester will send a new pose packet requesting a position validation. Packet collision occur happen during the voting process, so if the requester does not receive more than half the votes after a time $t_{valReply}$, it requests a new validation of its location in the next pose packet and this voting round is discarded. The mechanism was implemented in a way that the position validation is activated at a frequency equal to or less than the frequency of pose packets, which means that some position packets will not include a request for a validation of the sender's location. Every drone stores the calculated valid position to use it for the classifier model. When no position validation is requested in the pose packet, the classifier model will be responsible for determining if the position received is valid or not.

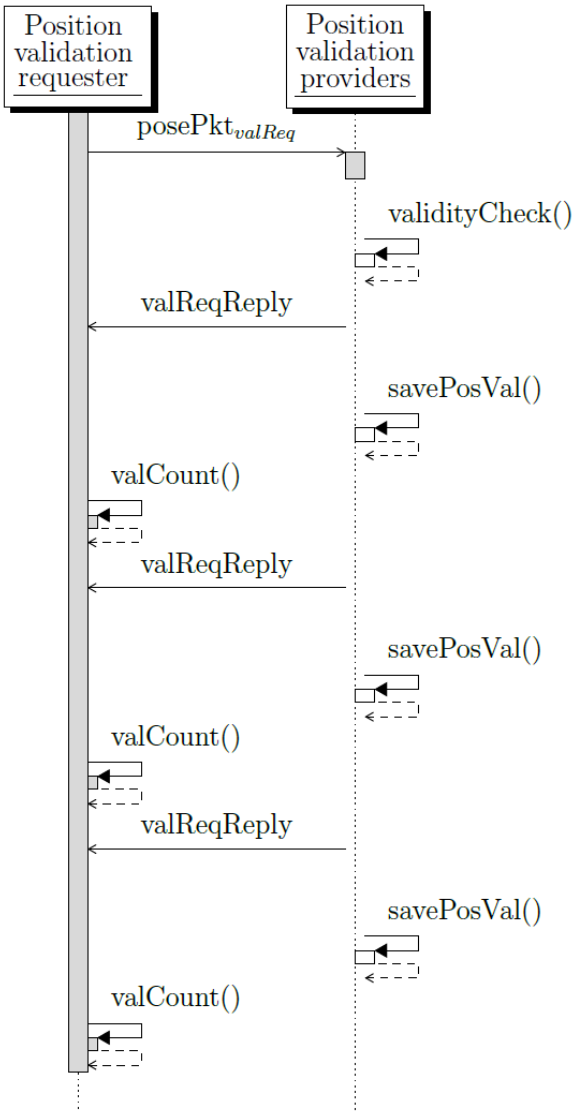


FIGURE 8. Interaction between entities in movement plausibility model.

2) CLASSIFIER MODEL

The classifier model, or position plausibility model, is activated when a drone receives a pose packet that does not include a position validation request. The plausibility of the position is calculated based on the last position validation computed for the drone that sent the position. Due to the assumed accelerated movement model, the position estimation is determined as presented in 1 and 2.

$$\boldsymbol{S}_{max} = \boldsymbol{S}_0 + \boldsymbol{v}\Delta t + \frac{1}{2}\boldsymbol{a}\Delta t^2 \tag{1}$$

$$\boldsymbol{S}_{min} = \boldsymbol{S}_0 + \boldsymbol{v}\Delta t - \frac{1}{2}\boldsymbol{a}\Delta t^2 \tag{2}$$

Δt is calculated using the time difference between the timestamp in the pose packet and the timestamp of the last calculated position validated. $\boldsymbol{a}$ and $\boldsymbol{v}$ are, respectively, the maximum acceleration and medium velocity passed as a parameter in the simulation. The actual acceleration in the analyzed period is unknown, therefore the precise position of the movement cannot be determined. Consequently, based on the acceleration parameter, it is possible to calculate a range in which the position should be if the drone moved using maximum acceleration or de-acceleration. The plausibility is then determined by checking if the position sent in the pose packet is within feasible boundaries. If the position is within feasible boundaries, it will be classified as *plausible*, otherwise it will be classified as *implausible*.

Regarding the communication between cells, every time a drone in a cell receives a position packet from another cell, it asks its armored vehicle for the position of the armored vehicle from the other cell. This allows drones from one cell to validate the position of drones from other cells.

E. REJOINING PROCESS

During a reconnaissance mission, a drone may leave the cell to execute a given task. This is the case, for instance, when it has to check a given event or object that is close to the cell, but out of the range of the other nodes in the cell. If the duration of this disconnection exceeds a preset amount of position packets (n_d), the drone will be considered disconnected from the cell, meaning that it will have to be

authenticated again when returning to the network. A long disconnection of a drone from the cell will also trigger the process of refreshing the session key. The old session key is not discarded, because the disconnected drone (d_d) will use it to communicate with its now former cell. When d_d returns, it will be placed in a quarantine period (Δt_q). It will only receive position updates from the cell's av, so it can position itself with the expected offset and resume with its movement pattern. During this period, its movement pattern will be analyzed by the cell members, based on the same parameters used for position validation, as presented on IV-D1, such as, $\mathbf{av}_{pos}$, $\theta_{\mathbf{av}}$, **OS**, and mobility model. The Δt_q is the time between a preset amount of position packets (n_q), where $n_q \geq n_d$. The disconnected drone will only receive the new session key if its movement pattern matches the movement pattern expected by the other members of the cell. Concerning the session key refreshing process, the armored ground vehicle is responsible for generating a new session key, which will then be sent to each drone encrypted using the drone's public key and signed by the armored vehicle.

F. SUPPORTING POSITION DATA ACQUISITION FROM ANOTHER CELL

As mentioned above, when 2 cells A and B are connected, if a drone in cell A (d_A) receives a position packet from a drone in cell B (d_B), d_A will need the position of the armored ground vehicle of cell B (av_B) to calculate the validity of the position of d_B. In order to get this data, drone d_A needs to request it from the ground vehicle in its cell (av_A), as it cannot directly communicate with av_B, and it cannot trust the drone (d_B) to provide this information as this drone may be compromised (i.e., it may be a malicious node).

Considering the assumption of an inter-cell exclusive communication channel between armored ground vehicles that was introduced in Section 3.2, d_A can obtain the position of av_B via av_A. This way, when d_A receives a position packet from d_B, it will ask its armored ground vehicle for the updated position of av_B. Only after receiving this information from av_A will d_A be able to calculate the validation of d_B's position.

V. EXPERIMENTAL SETTINGS AND EVALUATION

This section presents the experiments used to validate the proposal. Details about the simulation environment are presented, followed by the evaluation metrics. Then the specific parameters that were used in the simulation runs are presented, followed by a discussion of the acquired results.

A. SIMULATION ENVIRONMENT

The proposed scheme was evaluated by performing simulations in INET, an OMNet++ based framework. OMNet++ is a network simulator for implementing and testing novel networking solutions. By using the INET framework, it is possible to gather valuable results considering realistic mobility models and wireless communication constraints. As part of the solution, OpenSSL APIs were used to compute the required cryptographic operations. Figure 9 depicts the relationship between the elements included in the simulation environment, such as frameworks and libraries.

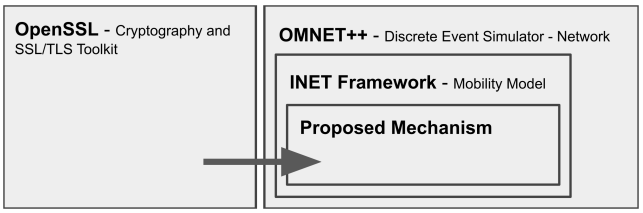


FIGURE 9. Simulation environment.

The simulation environment was implemented on a device with a Windows 8.1 × 64 operating system, an Intel Core processor (i7-4500U), 8GB of RAM, and 1TB of HDD.

B. EVALUATION METRICS

The evaluation of this proposal was performed based on the two different scenarios described in Section III. The first scenario focuses on evaluating the effectiveness of the scheme in detecting an intruder inside a cell. The second scenario focuses on evaluating the effectiveness of the scheme in detecting an intruder impersonating another cell, which means the attacker is outside the victim cell.

Two kinds of pose packets are considered (containing the position information of a drone): a *falsified pose packet*, which is a packet in which the position has been manipulated by an attacker; and a *correct pose packet*, which is a packet that was not manipulated. The position validation providers categorize each pose packet as being either *plausible* or *implausible*. In regard to the notation: a *true positive* (TP) is when a falsified pose packet is classified as implausible; a *true negative* (TN) is when a correct pose packet is classified as implausible; a *false negative* (FN) is when a falsified position packet is classified as plausible; and a *false positive* (FP) is when a correct position packet is classified as implausible. According to these definitions, the metrics used to evaluate UAVouch are as follows:

- *Retransmission Rate:* The percentage of retransmitted pose packets;

$$\text{Retransmission rate} = \frac{r}{s}, \tag{3}$$

  where r is the total number of pose packets resent and s is the total number of pose packets sent
- *Overhead:* The percentage of increase in packets sent in the network due to the application of the UAVouch scheme;

$$\text{Overhead} = \frac{\alpha - \beta}{\alpha}, \tag{4}$$

  where α is the total number of packets sent with UAVouch and β is the total number of packets sent without UAVouch
- *Accuracy:* The percentage of correctly classified pose packets;

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

- *True Negative Rate (TNR) or Specificity:* The percentage of correct pose packets correctly classified as plausible;

$$TNR = \frac{TN}{TN + FP} \tag{6}$$

- *True Positive Rate (TPR) or Sensitivity:* The percentage of falsified pose packets correctly classified as implausible.

$$TPR = \frac{TP}{TP + FN} \tag{7}$$

Specificity and sensitivity have an inherent trade-off between them. It is not possible to maximise both at the same time. The goal is to achieve a balance between these two metrics.

C. SIMULATION PARAMETERS

Table 4 presents the parameters that were considered in either scenario 1 or 2. For each combination of the presented parameters, 33 runs were executed using the simulator. A statistical power analysis (significance test) was conducted using Minitab software to validate that a sufficient number of simulations were run. With a standard deviation and a maximum difference between means of 4.8 taken from the analysis of the simulation data, a significance level of 0.05 ($\alpha = 0.05$), and a sample size of 33, with one sample for each run, the obtained power was 0.93. A commonly accepted value for statistical power is 0.9.

The drone mobility model chosen for the performed simulations was a circular mobility model. This model is combined with the accelerated linear mobility of the ground armored vehicle, which provides a spiral-like movement, as illustrated in Figure 10. This model was chosen because despite its trivial computation complexity, as defined in Equations 8 to 11, it is not completely trivial to mimic by a malicious node that does not know that this is the model being used and based only on visual observation of the movement. In a real scenario, a more elaborate mobility pattern could be utilized, but the focus here is on the general mechanism.

TABLE 4. Simulation parameters.

Parameter	Value
Drone mobility model	Circular mobility
Number of drones per cell	4
Communication range	$\approx$ 1 [km]
Asymmetric cryptography	RSA 2048-bit key
Maximum acceleration	2.5 [m/s^2]
Simulation time	200 [s]
Position noise mean/σ	0/0.5 [m]
Plausibility check threshold	$1\sigma, 2\sigma, 3\sigma, 4\sigma$ [m]
Position validation period	0.1, 0.2, 0.5, 1.0 [s]
Pose packet period	0.1 [s]
Packet maximum size	100 [bytes]
Armored vehicle velocity	20 [Km/h]
Attacker position offset	10 [m]

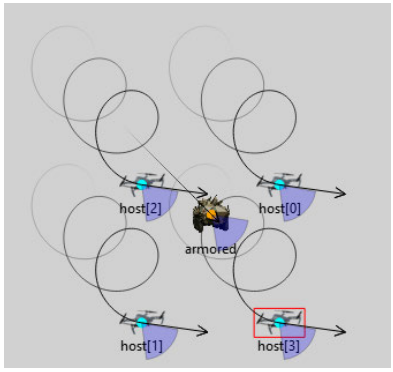


FIGURE 10. Screenshot of a simulation run showing the movement trail combining the circular and linear mobility models.

The center of the circular movement, represented by **cpos**, is calculated using matrix rotation. After calculating **cpos**, the distance between the position sent (*pose*), represented as **dpos**, and the center, represented as **r**, is calculated based on the distance between 2 points, as defined in Eq. 11. If **r** is inside the boundaries determined by the threshold for the radius of the circular movement, then the position is considered legitimate, otherwise it is considered false.

$$cpos_x = avpos_x + (OS_x cos(\theta) - OS_y sin(\theta)); \tag{8}$$

$$cpos_y = avpos_y + (OS_y cos(\theta) + OS_x sin(\theta)); \tag{9}$$

$$cpos_z = OS_z; \tag{10}$$

$$\delta_x = (cpos_x - dpos_x)^2$$

$$\delta_y = (cpos_y - dpos_y)^2$$

$$\delta_z = (cpos_z - dpos_z)^2$$

$$r = \sqrt{\delta_x + \delta_y + \delta_z}; \tag{11}$$

The simulation uses four drones, with one in front of the armored vehicle, another in back, one on the left side, and the last one on the right side. The small number of drones reduces the number of direct neighbors and consequently the number of connections, thus creating a more challenging environment for the experiments.

D. RESULTS AND DISCUSSION

The results from the simulation experiments for both scenarios are presented in the following.

1) SCENARIO 1

The purpose of scenario 1 is to evaluate the effectiveness of the solution in detecting an attack involving just one cell. At t = 30s, a drone inside the cell changes its settings and starts operating as the attacker, disseminating deceitful information and not being able to mimic the movement pattern. The error in the movement pattern is determined by the simulation parameter *attacker position offset*. The results are presented as follows.

a: TRUE NEGATIVE RATE (SPECIFICITY)

Figure 11 presents the percentage of the correct position that was correctly classified by the mechanism. This represents how effective the mechanism is at identifying legitimate drones. Effectiveness measurements, like the next ones, were taken based on the variations of the position validation and the plausibility check threshold. The position validation period is meant to evaluate the impact of using previously validated coordinates to classify a drone. The plausibility check threshold is meant to evaluate how resilient the mechanism can be regarding position errors and is based on the standard deviation (σ) of the position noise. It was expected that, with a shorter position validation period, the *TNR* would be better because the position plausibility mechanism would always have the drone's most recent position coordinates, therefore the error caused by using old position coordinates, as occurs with longer position validation periods, would be close to zero. It was also expected that with a shorter threshold, the plausibility model would have a higher sensitivity for error in the position coordinates, increasing the *FPR*, which means that the mechanism has incorrectly classified a legitimate drone as malicious. However, based on the simulation results presented in Figure 11, tiny fluctuations are noticeable in the true negative rates values resulting from variations in both threshold and position validation period values. This stems from the fact that the plausibility check designed for the circular mobility model has a very high rate of correctly classifying the position of the legitimate drones, as presented in Table 5. As a consequence, this metric should not be considered when deciding which parameter combination is the most efficient for this mechanism.

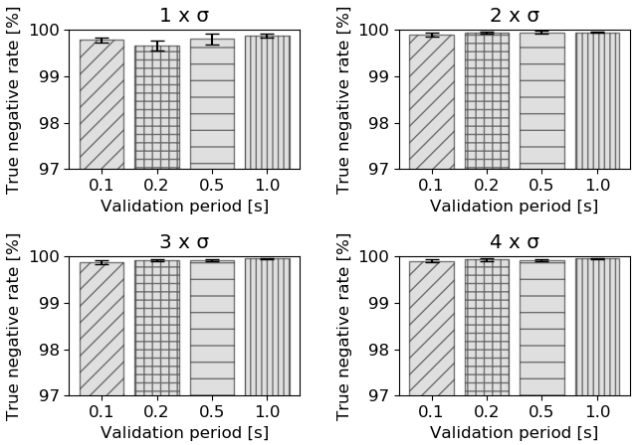


FIGURE 11. True negative rate.

TABLE 5. True negative rate from scenario 1.

Threshold \ Validation period	0.1	0.2	0.5	1.0
1σ	99.79	99.66	99.81	99.88
2σ	99.89	99.95	99.95	99.95
3σ	99.87	99.93	99.92	99.96
4σ	99.90	99.94	99.93	99.96

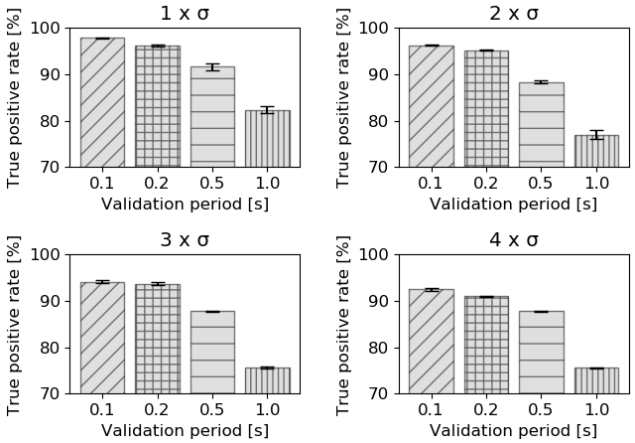


FIGURE 12. True positive rate.

b: TRUE POSITIVE RATE (SENSITIVITY)

Figure 12 presents the proportion of malicious nodes correctly classified by the mechanism. This represents how effective the proposed solution is in identifying a malicious node. As was expected with TNR, TPR is sensitive to variations in the threshold and position validation period values. As illustrated in Figure 12, it is noticeable that increasing the threshold, and consequently also the distance between the feasible boundaries, leads to an increase in the percentage of incorrect positions classified as correct, negatively impacting the performance of the proposed mechanism. The same negative impact occurs when there is an increase in the position validation period due to the use of old position coordinates as discussed above. Nevertheless, the mechanism achieved high true positives rates, above 90% for some combinations of threshold and position validation period values, showing that the position validation mechanism is a reliable and robust way to detect malicious drones.

c: ACCURACY

Figure 13 presents the graph of mechanism accuracy for the first scenario. Based on Eq. 5, it is expected that the accuracy

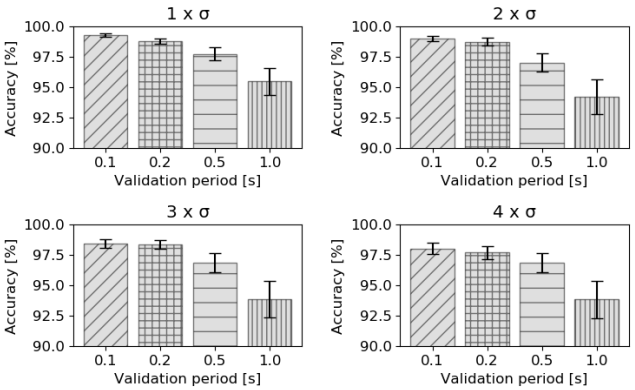


FIGURE 13. Accuracy.

would represent an approximated combination between both of the previously presented rates. Therefore, it is noteworthy that the position validation period and threshold parameters have a direct impact on the accuracy, as they impact both the TPR and TNR values. Considering only the metrics presented so far, the mechanism achieves a fairly high detection rate. For position validation periods of 0.1s and 0.2s, the detection rate was above 90% for 1, 2, and 3σ, and the overall accuracy was above 97.5%. This provides evidence of UAVouch's efficiency. The high accuracy values, combined with both high TNR and TPR values, demonstrates how good the proposed system is at correctly classifying a malicious drone as malicious and properly identifying a legitimate drone.

d: RETRANSMISSION RATE

In a distributed system, such as the one discussed here, the number of messages exchanged between the nodes is expected to be higher than in a centralized system. When dealing with wireless communications, this also leads to higher occurrences of interference and packet collisions. Figure 14 presents the retransmission rate measured in the first scenario. As expected, the retransmission rate is directly related to the validation frequency: as the rate of proofs increases, the number of transmitted packets also increases. As a consequence, the probability of packet collisions rises. On the other hand, varying the threshold value does not impact the probability of packet collisions, as it does not change the number of transmitted packets. It is clear that the packet retransmission rate metric impacts the voting system, as occasionally not all of the packets containing the votes are received by the position validation requester, thereby preventing it from reaching a decision and impacting the overall performance of the proposed solution. Table 6 shows the impact of this metric by measuring the percentage of validation requests that reached a decision. It is noticeable that the mechanism has a decision rate of 80% on average, meaning that a decision will be reached for 8 out of 10 requests.

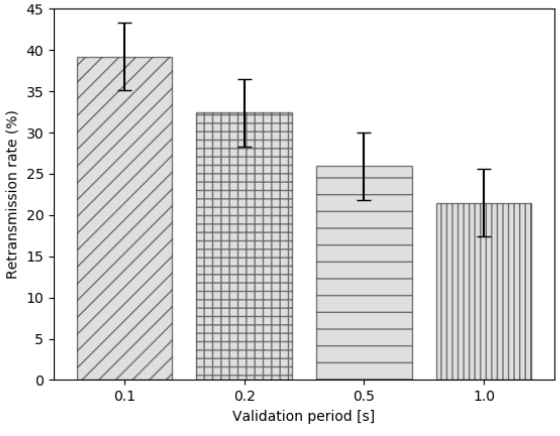


FIGURE 14. Retransmission rate.

TABLE 6. Decision rate - scenario 1.

Threshold \ Validation period	0.1	0.2	0.5	1.0
1σ	0.7990	0.8084	0.8084	0.7656
2σ	0.7991	0.7954	0.8136	0.8048
3σ	0.7987	0.7977	0.8150	0.8079
4σ	0.8012	0.8046	0.8134	0.8140

e: OVERHEAD

Figure 15 presents the overhead introduced into the system by the validation mechanism. It is evident that the mechanism overhead decreases as the position validation period increases. It is clear that varying the threshold value does not affect the number of packets being transmitted in the network, so since the overhead is computed based on the number of packets added to the network due to the use of UAVouch, the only parameter that affects the overhead is the validation period. Although the mechanism was responsible for a fairly high increase in the number of packets being transmitted, this number is reasonable in terms of bandwidth consumption. For the worst-case scenario, with a position validation request period of 0.1 s, and remembering that for each position validation request replies are expected from each drone in the network (3 replies in this case study), this would represent an increase of 30 packets per node in the network, thus, 120 packets in total. With a pose packet size of 60 bytes (on OMNet++), the data rate can be estimated at around 57,6 kbps, representing a very small bandwidth consumption when using technologies such as 4G and WiMax, for instance.

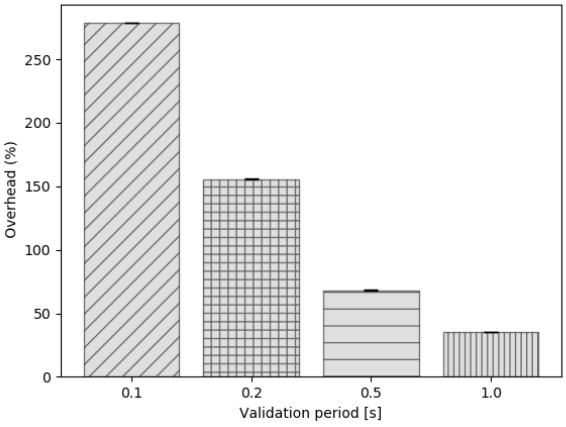


FIGURE 15. Overhead.

As mentioned in Section III, security mechanisms that are designed for military reconnaissance applications have to be efficient, avoiding negative impact on the performance of payload data transmission. With this requirement in mind, a trade-off must be made between detection performance and overhead to achieve the ideal combination of threshold and position validation period, such that the mechanism remains highly efficient, but without a significant increase in the imposed overhead.

Based on Figure 13, it is clear that the system's performance in detecting malicious drones is best for smaller

TABLE 7. Confusion matrix (1σ and 0.2s) - scenario 1.

		Predicted values	
		attacker	legitimate
Actual values	attacker	TP = 1662	FN = 40
	legitimate	FP = 7	TN = 1998

threshold values (1σ and 2σ) and smaller position validation periods (0.1 and 0.2). Taking the overhead into consideration as well, it also clear that the best combination of high detection rates and acceptable overhead is 1σ for the threshold and 0.2s for the position validation period. To better exemplify the performance for this particular combination of parameters, a confusion matrix is presented in Table 7.

2) SCENARIO 2

As in scenario 1, in scenario 2 the proposed solution's effectiveness in detecting an attacker was evaluated, but in this scenario the attacker is outside the cell. For these tests, as soon as the simulation starts the malicious node attempts to connect with cell 1, assuming the identity of a node from another cell. Once the malicious node manages to be authenticated and connect with cell 1, it starts sending manipulated position messages to cell 1, impersonating the other nodes in its *fake* network. The results obtained in the simulations for this scenario are presented in the following.

a: TRUE NEGATIVE RATE (SPECIFICITY)

Table 8 presents the *TNR* for scenario 2. As in scenario 1, tiny fluctuations in the TNR are noticeable as a result of variations in both threshold and position validation period values. Furthermore, even when the number of nodes was increased, there was only a tiny decrease in the TNR value from scenario 1 to scenario 2, demonstrating that the proposed scheme remained effective in identifying the legitimate drone correctly.

TABLE 8. True negative rate from scenario 2.

Threshold \ Validation period	0.1	0.2	0.5	1.0
1σ	99.76	99.60	99.85	99.87
2σ	99.92	99.96	99.94	99.96
3σ	99.91	99.96	99.95	99.96
4σ	99.91	99.96	99.95	99.96

b: TRUE POSITIVE RATE (SENSITIVITY)

Figure 16 presents the *TPR* from scenario 2. As in scenario 1, the *TPR* is noticeably affected by changes in the threshold. As presented in Table 9, increasing the threshold has a negative impact on the *TPR*. However, unlike in the first scenario, the *TPR* does not change with the variation of the position validation period. This happens because, in this scenario, the malicious drone is outside the cell network. Thus, every time a legitimate drone receives a request to check the malicious drone's position, it asks its armored vehicle

TABLE 9. True positive rate from scenario 2.

Threshold \ Validation period	0.1	0.2	0.5	1.0
1σ	97.55	97.50	97.52	97.57
2σ	95.03	95.05	95.13	95.13
3σ	92.58	92.61	92.59	92.63
4σ	90.14	90.16	90.20	90.17

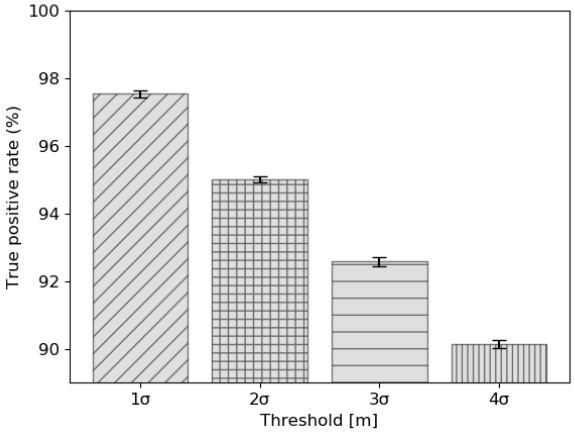


FIGURE 16. True positive rate.

for the position of the armored vehicle from the cell that the malicious drone is impersonating. Therefore, the position validation mechanism will always have updated information, and as a consequence the problem with stale information that happens in scenario 1 will never occur here. It is also noticeable that for all threshold values, with a position validation period value of 0.1s, the TPR is almost the same for both scenarios even though the number of drones has increased from the first to the second simulation.

c: ACCURACY

Figure 17 presents the overall accuracy of the proposed solution with regard to the Sybil attack simulated in the second scenario. As in the first scenario, the accuracy is dependent on the threshold. However, in the second scenario, the accuracy appears to be more sensitive to variance in the threshold, because in Figure 13, based on a position validation period of 0.1s, the accuracy difference between the first and the last graph is around 1.5%, but for the second scenario, this difference is more than 3.5%. This is related to the fact that more packets are introduced into the network, which increases the packet collisions, thus negatively affecting the efficiency. The other difference between the two scenarios is that in the second one, varying the position validation period does not have a significant impact on the accuracy. In the first scenario, the impact on the accuracy resulting from variation of the position validation period was because the true positive rate was affected by the position validation period variation. In the second scenario, since the true positive rate was not affected by the position validation period variation, this fact is reflected in the accuracy rate.

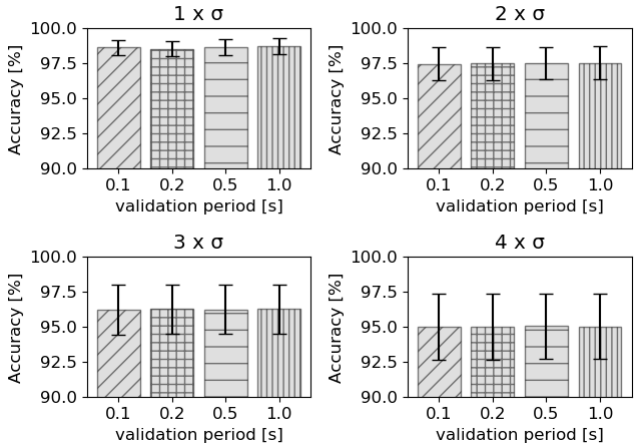


FIGURE 17. Accuracy.

d: RETRANSMISSION RATE

Figure 18 presents the retransmission rate measured in the second scenario. As expected, the retransmission rate is higher in the second scenario than it was in the first. The main reason is that with more drones in the network, more packets are exchanged. Therefore, there is a higher possibility of collisions occurring. Although only one additional drone is introduced from one scenario to the other, the extra one is acting as if it was 4 drones, and consequently it is like the network has doubled in size, from 4 to 8 drones. In numbers, from the first to the second scenario, the retransmissions increased by approximately 15%. As for the first scenario, the impact of the increase in retransmissions was analyzed regarding the decision rate, as shown in Table 10. The mechanism reached an average decision rate of 67%, meaning that about 7 out of 10 requests will result in a decision, one less than in the first scenario.

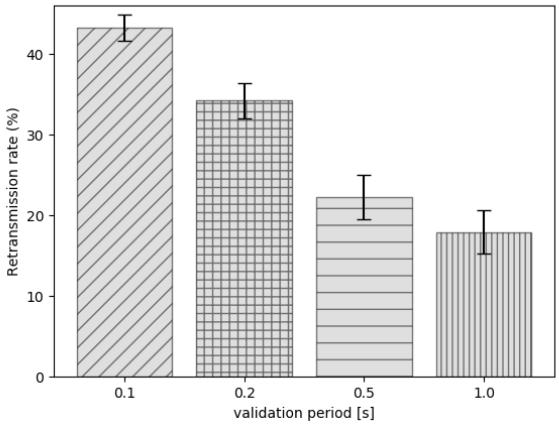


FIGURE 18. Retransmission rate.

e: OVERHEAD

Figure 19 presents the overhead introduced by the position validation mechanism. As in the first scenario, the overhead

TABLE 10. Decision rate - scenario 2.

Threshold \ Validation period	0.1	0.2	0.5	1.0
1σ	0.6342	0.6249	0.7168	0.6997
2σ	0.6266	0.6219	0.7124	0.7044
3σ	0.6323	0.6306	0.7214	0.7063
4σ	0.6364	0.6316	0.7184	0.7146

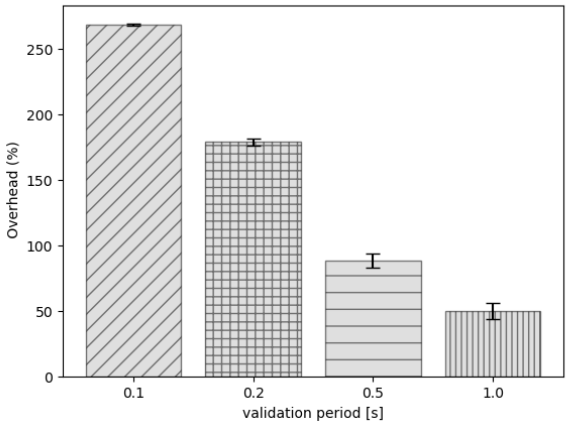


FIGURE 19. Overhead.

is noticeably susceptible to variation in the position validation period, but it is not affected by variation in the threshold. Comparing the overhead graphs for scenarios 1 and 2, it is also noticeable that there is a small increase in the overall overhead, which was expected due to a higher number of packets exchanged in the second scenario. As explained in the first scenario, although the second scenario also had high percentages of overhead, these numbers are completely acceptable given the data rates consumed by current widely used wireless technologies.

Based on the requirements discussed in scenario 1 and analyzing the results for scenario 2, it is clear that the best combination between high detection rates and acceptable overhead is a threshold of 1σ and a position validation period of 0.5s. To better exemplify the performance for this particular combination of parameters, a confusion matrix is presented in Table 11.

TABLE 11. Confusion matrix (1σ and 0.5 [s]) - scenario 2.

		Predicted values	
		attacker	legitimate
Actual values	attacker	TP = 1951	FN = 50
	legitimate	FP = 3	TN = 1997

VI. CONCLUSION

This paper presents a distributed scheme for identity and location validation that combines an asymmetric key-based authentication mechanism with a position validation mechanism for groups of drones. The proposal is evaluated using two attack scenarios, one for an impersonation attack, with

the intruder inside the cell, and the other for a Sybil attack, with the intruder outside the cell.

UAVouch was show to have high accuracy, above 90% in detecting the malicious node inside (scenario 1) and outside (scenario 2) its network. Due to the distributed nature of the protocol, evaluations of packet retransmission and the overhead of the mechanism were also presented. The results showed a retransmission rate below 50% for the worst-case scenario and an acceptable amount of overhead in all simulated conditions, which demonstrated the viability of the proposed scheme. Because of the voting system used in the proposed scheme, the number of times the system reached a decision was also evaluated. UAVouch achieved acceptable decision rates for both scenarios, with decision rates of 80% and 67% for scenarios 1 and 2 respectively.The difference is because in scenario 2 there is a higher number of devices exchanging messages, which increase the number of collisions, reducing the overall system decision rate, affecting on a minor scale the accuracy rate.

Regarding futures directions for this work, there are a few possibilities to explore that could improve the UAVouch scheme, particularly in its practical implementation, such as: *RSA key replacement*: Although RSA is widely used, RSA key size can be a problem for hardware limited systems such as the ones in most drones. Replacing RSA with an efficient algorithm, such as Elliptic Curves, could help to improve the system for real-world deployment. *Lower layers*: A more thorough investigation could be conducted of how a feasible long-range communication protocol, such as WiMax or LoRa, might affect the performance of the UAVouch mechanism. *Mobility model*: The mobility model has a significant impact on the design of the movement plausibility check. Further studies could be conducted to test the UAVouch position validation mechanism against other mobility models. Moreover, even with a complex mobility model, attackers could employ advanced learning mechanisms to predict it. More investigation about this topic needs to be conducted.

REFERENCES

[1] Agência Nacional de Aviacão Civil. (2019). *Registered Drones in Brazil From 2017 to 2019*. Accessed: Sep. 2019. [Online]. Available: https://www.anac.gov.br/assuntos/paginas-tematicas/drones/quantidade-de-cadastros

[2] W. Moskwa. (2016). *World Drone Market Seen Nearing $127 Billion in 2020, PWC Says*. Accessed: Sep. 2019. [Online]. Available: https://www.moneyweb.co.za/news/tech/world-drone-market-seen-nearing-127bn-2020-pwc-says/

[3] H. Shakhatreh, A. H. Sawalmeh, A. Al-Fugaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani, "Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges," *IEEE Access*, vol. 7, pp. 48572–48634, 2019.

[4] D. Orfanus, E. P. de Freitas, and F. Eliassen, "Self-organization as a supporting paradigm for military UAV relay networks," *IEEE Commun. Lett.*, vol. 20, no. 4, pp. 804–807, Apr. 2016.

[5] A. Fotouhi, H. Qiang, M. Ding, M. Hassan, L. G. Giordano, A. Garcia-Rodriguez, and J. Yuan, "Survey on UAV cellular communications: Practical aspects, standardization advancements, regulation, and security challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3417–3442, 2019.

[6] I. García-Magariño, R. Lacuesta, M. Rajarajan, and J. Lloret, "Security in networks of unmanned aerial vehicles for surveillance with an agent-based approach inspired by the principles of blockchain," *Ad Hoc Netw.*, vol. 86, pp. 72–82, Apr. 2019.

[7] R. Altawy and A. M. Youssef, "Security, privacy, and safety aspects of civilian drones: A survey," *ACM Trans. Cyber-Phys. Syst.*, vol. 1, no. 2, pp. 7:1–7:25, 2016.

[8] C. Adams, "Impersonation attack," in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg, Ed. Boston, MA, USA: Springer, 2005, p. 286.

[9] J. R. Douceur, "The sybil attack," in *Peer-to-Peer Systems*, P. Druschel, F. Kaashoek, and A. Rowstron, Eds. Berlin, Germany: Springer, 2002, pp. 251–260.

[10] E. Walia, V. Bhatia, and G. Kaur, "Detection of malicious nodes in flying ad-hoc networks (FANET)," *Int. J. Electron. Commun. Eng.*, vol. 5, pp. 6–12, Sep. 2018.

[11] F. Boeira, M. Asplund, and M. P. Barcellos, "Vouch: A secure proof-of-location scheme for VANETs," in *Proc. 21st ACM Int. Conf. Modeling, Anal. Simulation Wireless Mobile Syst. (MSWIM)*, New York, NY, USA, Oct. 2018, pp. 241–248.

[12] F. Boeira, M. Asplund, and M. Barcellos, "Decentralized proof of location in vehicular ad hoc networks," *Comput. Commun.*, vol. 147, pp. 98–110, Nov. 2019.

[13] G. Loukas, E. Karapistoli, E. Panaousis, P. Sarigiannidis, A. Bezemskij, and T. Vuong, "A taxonomy and survey of cyber-physical intrusion detection approaches for vehicles," *Ad Hoc Netw.*, vol. 84, pp. 124–147, Mar. 2019.

[14] S. Sharma and A. Kaul, "A survey on intrusion detection systems and honeypot based proactive security mechanisms in VANETs and VANET cloud," *Veh. Commun.*, vol. 12, pp. 138–164, Apr. 2018.

[15] X. Wang, A. Pande, J. Zhu, and P. Mohapatra, "STAMP: Enabling privacy-preserving location proofs for mobile users," *IEEE/ACM Trans. Netw.*, vol. 24, no. 6, pp. 3276–3289, Dec. 2016.

[16] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: Technical challenges, recent advances, and future trends," *Proc. IEEE*, vol. 104, no. 9, pp. 1727–1765, Sep. 2016.

[17] S. Doss, A. Nayyar, G. Suseendran, S. Tanwar, A. Khanna, L. H. Son, and P. H. Thong, "APD-JFAD: Accurate prevention and detection of jelly fish attack in manet," *IEEE Access*, vol. 6, pp. 56954–56965, 2018.

[18] E. C. Ferrer, "The blockchain: A new framework for robotic swarm systems," in *Proc. Future Technol. Conf. (FTC)*, K. Arai, R. Bhatia, and S. Kapoor, Eds. Cham, Switzerland: Springer, 2019, pp. 1037–1058.

[19] I. J. Jensen, D. F. Selvaraj, and P. Ranganathan, "Blockchain technology for networked swarms of unmanned aerial vehicles (UAVs)," in *Proc. IEEE 20th Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, 2019, pp. 1–7.

[20] S. Aggarwal, M. Shojafar, N. Kumar, and M. Conti, "A new secure data dissemination model in Internet of drones," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[21] N. Islam, M. K. Hossain, G. G. M. N. Ali, and P. H. J. Chong, "An expedite group key establishment protocol for flying ad-hoc network (FANET)," in *Proc. 5th Int. Conf. Informat., Electron. Vis. (ICIEV)*, May 2016, pp. 312–315.

[22] Z. Ali, S. A. Chaudhry, M. S. Ramzan, and F. Al-Turjman, "Securing smart city surveillance: A lightweight authentication mechanism for unmanned vehicles," *IEEE Access*, vol. 8, pp. 43711–43724, 2020.

[23] J. Srinivas, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, "TCALAS: Temporal credential-based anonymous lightweight authentication scheme for Internet of drones environment," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6903–6916, Jul. 2019.

[24] M. R. Nosouhi, S. Yu, M. Grobler, Y. Xiang, and Z. Zhu, "SPARSE: Privacy-aware and collusion resistant location proof generation and verification," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[25] J. Ferreira and M. L. Pardal, "Witness-based location proofs for mobile devices," in *Proc. IEEE 17th Int. Symp. Netw. Comput. Appl. (NCA)*, Nov. 2018, pp. 1–4.

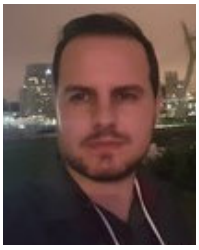
[26] M. H. Tareque, M. S. Hossain, and M. Atiquzzaman, "On the routing in flying ad hoc networks," in *Proc. Federated Conf. Comput. Sci. Inf. Syst. (FedCSIS)*, Sep. 2015, pp. 1–9.

[27] O. S. Oubbati, M. Atiquzzaman, P. Lorenz, M. H. Tareque, and M. S. Hossain, "Routing in flying ad hoc networks: Survey, constraints, and future challenge perspectives," *IEEE Access*, vol. 7, pp. 81057–81105, 2019.

[28] F. Boeira, M. P. Barcellos, E. P. de Freitas, A. Vinel, and M. Asplund, "Effects of colluding Sybil nodes in message falsification attacks for vehicular platooning," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Turin, Italy, Nov. 2017, pp. 53–60.
[29] M. Rodrigues, J. Amaro, F. S. Osorio, and B. R. L. J. C. Kalinka, "Authentication methods for uav communication," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2019, pp. 1210–1215.
[30] I. Zacarias, J. Schwarzrock, L. P. Gaspary, A. Kohl, R. Q. A. Fernandes, J. M. Stocchero, and E. P. de Freitas, "Employing sdn to control video streaming applications in military mobile networks," in *Proc. IEEE Int. Symp. Netw. Comput. Appl. (NCA)*, Oct. 2017, pp. 1–4.
[31] A. Sehrawat, T. A. Choudhury, and G. Raj, "Surveillance drone for disaster management and military security," in *Proc. Int. Conf. Comput., Commun. Autom. (ICCCA)*, May 2017, pp. 470–475.
[32] A. Y. Husodo, G. Jati, N. Alfiany, and W. Jatmiko, "Intruder drone localization based on 2D image and area expansion principle for supporting military defence system," in *Proc. IEEE Int. Conf. Commun., Netw. Satell. (Comnetsat)*, Aug. 2019, pp. 35–40.
[33] C. Paucar, L. Morales, K. Pinto, M. Sánchez, R. Rodríguez, M. Gutierrez, and L. Palacios, "Use of drones for surveillance and reconnaissance of military areas," in *Developments and Advances in Defense and Security*, Á. Rocha and T. Guarda, Eds. Cham, Switzerland: Springer, 2018, pp. 119–132.
[34] B. Engberts and E. Gillissen, *Policing From Above: Drone Use by Police*. The Hague, The Netherlands: T.M.C. Asser Press, 2016, pp. 93–113.
[35] A. Chowdhery and M. Chiang, "Model predictive compression for drone video analytics," in *Proc. IEEE Int. Conf. Sens., Commun. Netw. (SECON Workshops)*, Jun. 2018, pp. 1–5.
[36] C. Lin, D. He, N. Kumar, K.-K.-R. Choo, A. Vinel, and X. Huang, "Security and privacy for the Internet of drones: Challenges and solutions," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 64–69, Jan. 2018.
[37] R. Shakeri, M. A. Al-Garadi, A. Badawy, A. Mohamed, T. Khattab, A. K. Al-Ali, K. A. Harras, and M. Guizani, "Design challenges of multi-UAV systems in cyber-physical applications: A comprehensive survey and future directions," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3340–3385, 2019.
[38] J. Giraldo, E. Sarkar, A. A. Cardenas, M. Maniatakos, and M. Kantarcioglu, "Security and privacy in cyber-physical systems: A survey of surveys," *IEEE Des. Test. Comput.*, vol. 34, no. 4, pp. 7–17, Aug. 2017.
[39] H. Peng, A. Razi, F. Afghah, and J. Ashdown, "A unified framework for joint mobility prediction and object profiling of drones in UAV networks," *J. Commun. Netw.*, vol. 20, no. 5, pp. 434–442, Oct. 2018.

CARLOS FELIPE EMYGDIO DE MELO received the Computer Engineering degree from the Federal University of Itajubá (UNIFEI), in 2016, and the M.Sc. degree in electrical engineering from the Federal University of Rio Grande do Sul (UFRGS), in 2020. His current research interests include computer networks, wireless networks, flying *ad hoc* networks, cybersecurity, data science, and (multi)unmanned aerial vehicles.

TULIO DAPPER E SILVA graduated in automation and control engineering and received the M.Sc. degree in electrical engineering from the Federal University of Rio Grande do Sul (UFRGS), Brazil, in 2020. His research interests include autonomous and intelligent systems, data science, wireless networks, and software-defined networking.



FELIPE BOEIRA is currently pursuing the Ph.D. degree with the Computer Science Graduate School (CUGS), Linköping University, Sweden. He is also an Information Security Researcher with Linköping University. He worked with research topics related to mobile, wearable, and the IoT security, and his research is currently focused on vehicular *ad hoc* networks (VANETs) security.

JORGITO MATIUZZI STOCCHERO received the bachelor's degree in military sciences from AMAN, in 1990, the Communication Engineering degree from IME, in 1996, the M.Sc. degree in electrical engineering from COPPE/UFRJ, in 2004, the M.B.A. degree in politics and strategy from FGV/RJ, in 2016, and the master's degree in military sciences from ECEME, in 2016 and 2012. He is currently working on the development of scientific cooperation between the Brazilian Army and UFRGS.

ALEXEY VINEL (Senior Member, IEEE) received the Ph.D. degrees from the Institute for Information Transmission Problems, Moscow, Russia, in 2007, and from the Tampere University of Technology, Tampere, Finland, in 2013. He has been a Professor with the School of Information Technology, Halmstad University, Halmstad, Sweden, since 2015, and a Professor II with the Department of Electrical Engineering, Western Norway University of Applied Sciences, Bergen, Norway, since 2018. His areas of interests include wireless communications, vehicular networking, and cooperative autonomous driving.

MIKAEL ASPLUND received the M.Sc. degree in computer science and engineering and the Ph.D. degree in computer science from Linköping University, Sweden, in 2005 and 2011, respectively. His Ph.D. thesis focused on the design and analysis of partition-tolerant distributed systems, including the development of middleware services for maintaining consistency and information dissemination algorithms for disaster area networks. From 2011 to 2012, he was a Research Fellow with Trinity College Dublin, for one year. He is currently an Associate Professor with the Real-Time Systems Group, Linköping University. His current research interests include cyber-physical security, dependable distributed systems, and vehicular computing.

EDISON PIGNATON DE FREITAS (Member, IEEE) received the Computer Engineering degree from the Military Institute of Engineering, in 2003, the M.Sc. degree in computer science from the Federal University of Rio Grande do Sul (UFRGS), in 2007, and the Ph.D. degree in computer science and engineering from Halmstad University, Sweden, in 2011. He is currently an Associate Professor with UFRGS, acting in the Graduate Programs on Computer Science (PPGC) and Electrical Engineering (PPGEE), developing research mainly in the areas of computer networks, real-time systems, and (multi)unmanned aerial vehicles.