Formal Analysis of $k-$Resiliency for Collaborative UAVs

A H M Jakaria and Mohammad Ashiqur Rahman
Department of Computer Science, Tennessee Tech University, Cookeville, TN, USA
Emails: ajakaria42@students.tntech.edu, marahman@tntech.edu

***Abstract*—Unmanned aerial vehicles (UAVs) are growingly used in different aspects of our lives. As the cost of UAVs is decreasing with novel technologies, their popularity is increasing rapidly in surveillance, disaster management, agriculture, military operations, etc. The recent trend of collaborative operations of a network of UAVs to achieve a common objective has attracted the researchers as well as commercial vendors. It has revolutionized the means of data collection to maximize mission performances. However, the collaborative UAVs, with respect to the mission objective, need to be resilient to the unavailability of one or more UAVs that can be caused due to cyberattacks or technical failures. As UAVs can easily be targeted by adversaries, these devices need to maintain safe communication with each other while avoiding fuel outage and mid-air collisions, thus reducing the possibilities of being hacked but remain undetected. In this work, we present a formal framework that takes different UAV parameters in a UAV network, resiliency requirements, and resource constraints as the input, and verifies for the resiliency threats. Each threat specifies if k or less number of UAVs fail, whether the rest of the network still can maintain successful communication. We illustrate the execution of this framework with an example case study. We evaluate the proposed framework to demonstrate the relationships between different parameters of a network and its resiliency. We also evaluate the framework in terms of the resiliency analysis performance and scalability.**

***Index Terms*—Collaborative UAV; UAV swarm; UAV network; formal modeling; resiliency verification**

I. Introduction

UAVs have revolutionized the collection of data from the aerial viewpoint. They are being used in civilian applications, as well as in the military, where the popularity of UAVs is increasing rapidly. They are successfully being used in applications, such as search and rescue, perimeter surveillance, situational awareness during natural disasters, and many other cases [1]. They are also being used in action-oriented applications, where tasks are typically costly, infeasible, dangerous, or monotonous for human pilots. The collaborative swarm approach is a recent trend of utilizing UAVs in such applications. For example, A fleet of cargo drones has been recently tested successfully in the Amazon rainforest in South America to deliver essential medical equipment to remote areas [2].

The UAV swarm is a network of UAVs that perform a common goal by communicating with each other. With all the recent endeavors of modern technology, UAVs are becoming easily affordable. To obtain precise and broader results, groups of UAVs are preferred these days. There may be heterogeneous types of UAVs in a swarm network where many UAVs perform the actual task (actuators) while some UAVs control and navigate the actuators (navigators) [3]. A UAV swarm can provide us with multi-fold benefits including a wide area of coverage as well as avoiding redundancy in collected data, or correct decisions about a consensus problem [4].

The mission of a UAV network depends on the availability of its UAVs, as they often rely on each other to perform the intended tasks. However, UAVs are vulnerable to cyberattacks, physical damages, or technical failures. A UAV can be compromised if it has inadequate collaboration with its neighbors. A compromised UAV can land in an unsafe zone giving sensitive information to the adversary, as well as they can be used to hack others in the network in a chaining fashion. While they are performing their jobs, they must always maintain a secure position that helps to avoid potential mid-air collisions, or fuel run-out. All UAVs in a network, hence, should always maintain a safe position with respect to others. In case some UAVs become unavailable, they should not affect the safety of others.

In this work, we propose a verification framework that automatically determines the resiliency of a UAV network, and can find out unsafe or vulnerable UAVs in terms of control or connectivity requirement, when a number of UAVs are unavailable (k), which can be due to accidents or cyberattacks. The framework takes necessary UAV parameters, such as current position, communication range, current velocity including direction, fuel levels, encryption capabilities, etc. as input. It formally models these parameters, as well as the requirements and constraints to maintain secure communication, and finally solves the model using a satisfiability modulo theory (SMT) based formal verification engine [5]. The results can determine whether the network of UAVs is resilient or not under the unavailability of a number of UAVs, and allows a navigator UAV to navigate others in a safer manner. In summary, the major contributions of this paper include:

1) Formal modeling of UAV parameters with respect to control requirements and resource constraints for analyzing the $k-$resiliency of a UAV network.
2) Prototype Implementation of the proposed framework and demonstration of the resiliency verification on an example case study.
3) Evaluation of the proposed framework tool in terms of the UAV and resiliency parameters, as well as the scalability.

The rest of this paper is organized as follows: Section II gives an overview of collaborative UAV networks, as well as the related works in this area. In Section III, we present

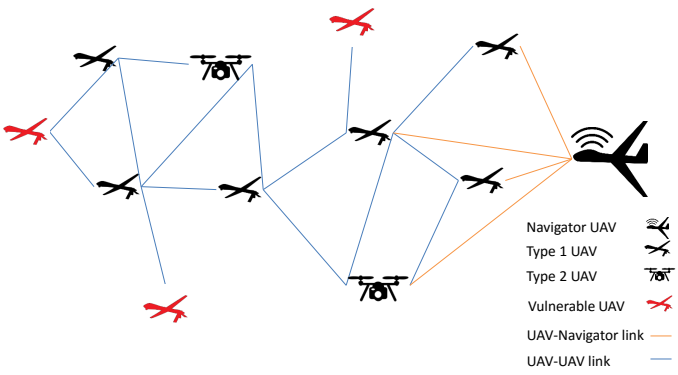


Fig. 1. A wireless network topology of UAVs.

the framework of the proposed verification tool. Section IV presents the formal model of the overall problem of resiliency analysis. The implementation of the proposed framework is briefly discussed in Section V. We demonstrate the resiliency verification on an example case study in Section VI. The evaluation results of the framework is presented in Section VII. We conclude the paper in Section VIII.

II. Background

This section briefly overviews the collaborative nature of UAVs, why it is important, and how it is achieved. Later, we present a comprehensive literature review on collaborative UAVs and existing models of UAV networks for performing a set of goals. Then we discuss our objective of the research.

A. Collaborative UAVs

Traditionally, individual UAVs are utilized in many different use cases, such as in infrastructure surveillance, target monitoring, rescue operation, etc. They are used to collect data as images, videos, or sensor data from an aerial perspective. They are very useful to get the jobs done where they are monotonous, dangerous, or too expensive for humans. However, if the task is multi-objective or distributed in nature, a single UAV may not be able to provide good results. For example, one UAV cannot perform well to detect a threat that needs tracking of multiple targets at a time or within a limited period. If we can use multiple UAVs that collaborate with each other and cooperatively performs a common goal, it provides much better results in terms of performance. They can cover a much larger area in a short time, as well as reduce the redundant nature in the collected data. With the recent advancements in modern technology, UAVs are becoming less expensive making the way to easily deploy collaborative UAVs.

A UAV network may consist of a variety of vehicles. There can be different roles of these different types of UAVs [6]. Often there is a navigator UAV to control such a network, which needs to communicate with each of the UAVs to perform the task successfully [3]. These communications are done in a wireless sensor network in ad-hoc fashion [4]. One challenge for such a navigator is to send commands to all the UAVs successfully to maintain a safe network that cannot be breached.

In a network of UAVs, the vehicles are dependent on each other in many cases. For example, they can depend on each other's location data if there is no GPS service, or if there is a chance of spoofed location assumption due to being compromised. They can also be dependent when a consensus problem is solved utilizing decisions from multiple UAVs. For example, a conclusion about a target to be correct can be drawn based on the decisions provided by a group of UAVs. While searching for a survival point for a battalion in warfare, a collaborative decision is very important.

However, in such a network, each UAV needs to be secured from being breached or isolated. A number of security issues for video feeds from UAVs, used by first responders in emergency situations, were studied by Rivera *et al.* [7]. In many cases, the data is transmitted by the UAVs without any encryption, which raises privacy and security issues. In a wireless network of UAVs, each UAV is dependent on its neighbors for being secured through successful communication. Given the communication ranges, as long as they have an adequate number of neighbors, they can safely reach the navigator and exchange data. The network should always maintain some requirements while satisfying constraints. For example, they have limited fuel, so they must return to a refueling station safely before they run out of fuel. After a certain level of fuel, they must head back to avoid unwanted break down on deserted or enemy regions.

Each UAV should maintain a safe distance from its neighbor UAVs to avoid mid-air collisions while remaining within the communication range with its neighbors. They can employ different radio frequency and encryption mechanisms to communicate with each other. When communicating, a common frequency band and encryption method are required between two UAVs. If there are different types of UAVs in the network, any communication between two UAVs of different types may require at least a certain level of encryption. A sample network of UAVs is presented in Fig. 1. The example consists of two types of UAVs and a navigator controlling them. The links are established dynamically according to the positions, communication properties, and ranges of the UAVs. Some UAVs in the network are considered vulnerable because of having a small number of neighbors or a low remaining flying time. These may not have been already compromised, but they are susceptible to easily be hacked by an adversary who is trying to disrupt or compromise the UAV network's mission. In this work, we consider UAV networks with the above-mentioned properties and formally model them to find out the resiliency of the network at the current instant.

B. Related Works

There are several works in the literature that provides solutions related to the management of collaborative UAVs, particularly the problems of path projection and path planning. For example, Tisdale *et al.* proposed a mechanism to examine control strategies for a team of UAVs performing cooperative

sensing [8]. The mechanism performs path planning minimizing the uncertainty in sensing missions by considering a blend of online sensor models and estimation objectives. Beard *et al.* described the design and implementation of semi-autonomous UAVs, as well as proposed their graph-based approach to path planning and trajectory generation [9]. Ceccarelli *et al.* developed a model for path planning of a micro UAV for reconnaissance, where the goal is to obtain video footage of a set of known targets [10]. They find the best selection of waypoints in achieving the goals in the presence of a constant wind. However, they did not consider a group of UAVs. Bortoff presented a two-step algorithm to obtain a path for UAVs traveling through enemy sites [11]. The technique performed a trade-off between the shortest path for traveling and avoiding radar ranges.

Kvarnstrom *et al.* presented a technique for mission planning that combines ideas from forward-chaining planning and partial order planning to meet the requirements of centralization, abstraction and distribution of task assignments [12]. A health management system for UAV teams that perform persistent surveillance was proposed by Bethke *et al.* [13]. They propose a methodology for planning missions that can anticipate the negative effects of various types of anomalies on future mission state, as well as choosing actions to mitigate the effects.

Kuiper *et al.* proposed a variation of mobility models to describe a pheromone model, where the movement of a group of UAVs is guided by pheromone [14]. This model performs well in reconnaissance. The authors in [15], [6] proposed an architecture that integrates cloud computing with UAV networks for better communication and easier developments of collaborative UAV applications. Teacy *et al.* developed a project named SUAAVE, where a fully automated sensing platform is used for multiple UAVs [1]. They aim to solve the complex dependencies between data fusion, ad-hoc wireless network issues and multi-agent coordination. But no formal verification of such network is performed. A model for assessing UAV system architectures was proposed by Renault [16]. This model only evaluates the functional and system architecture to reduce system ambiguity, as well as enhances the tangibility of the system. The model generates a score for given system architecture to determine its acceptability. However, it does not provide any verification of collaborative UAV networks. A communication model for collaborative UAVs was proposed by Borges *et al.* [17]. The model focuses on improving the stealth to collaboratively achieve a common goal. The model has mechanisms to detect and avoid threats by turning off radar communication with others.

There are few works that formally model the UAV network according to some requirements and constraints. Tsourdos *et al.* presented a Kripke model to cope with uncertainties and decision making involved in the process of collaborative achievement of reaching the goal while covering a certain area of interest avoiding obstacles on the path [18]. They modeled the collision avoidance with the help of two conditions: minimum separation and non-intersection of paths at equal length. The model finds the shortest path for each UAV, which guarantees fuel consumption. In [19], the authors used SMV model checker to verify the model against temporal properties expressed in computational tree logic (CTL). However, they did not consider the resiliency in terms of communication capabilities of the UAVs with each other while performing the model checking. Zhu *et al.* presented a model of collaborative UAV swarm towards coordination and control mechanisms [20]. They modeled a multi-agent system by constructing a model for an agent's behavior, constraints and uncertainties using a bottom-up approach. They did not model the resiliency in terms of position, communication, and fuel management.

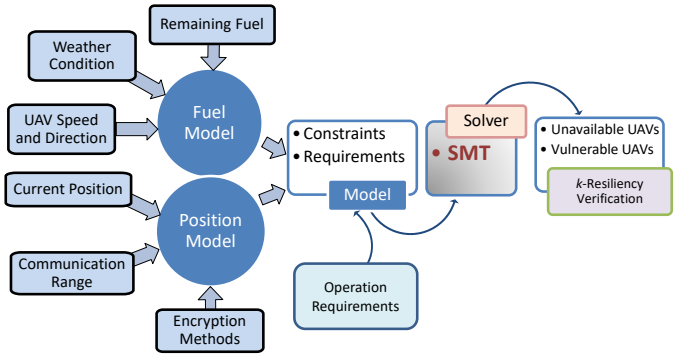


Fig. 2. The verification framework architecture.

C. Research Challenge and Our Objective

Analyzing of the resiliency of a UAV network is essential, especially from the navigation point of view. If there is any UAV in the network that is considered as vulnerable to cyberattacks, it needs to be instructed immediately to return to a safe territory. If the absence of a UAV makes other UAVs vulnerable in terms of communication with the navigator, it is considered as a threat. Identifying the resiliency threats in a UAV network is very important. However, satisfying all the requirements of the mission and constraints of the resources, given the UAV properties of a large network, forms a multi-objective combinatorially hard problem. Since, to the best of our knowledge, no other work in the literature solves a similar problem, we propose a formal method-based framework to solve it. Within this framework, we model the UAV network configurations and resiliency properties, encode the model into SMT logics, and solve it to identify the potential resiliency threats. The threats will allow the navigator to guide the vulnerable set of UAVs to safe positions to keep the network resilient.

III. Resiliency Verification Framework

Our proposed framework follows a top-down approach for the automatic verification of UAV network. Fig. 2 presents an overview of the framework. The framework formally models the UAV parameters such as remaining fuel, current position, their velocity and direction, communication ranges, encryption

mechanisms, and weather conditions. These are fed to the framework in an input file. The framework can solve the model and provide a solution that denotes whether a network is safe from cyberthreats or not.

We simply plot the current positions of all the UAVs on a Cartesian plane and calculate their distance from each other. We also calculate the isolation as a triangle formed by the trajectories of any two UAVs and themselves. Details on the isolation calculation are provided in Section IV-B. The UAVs create a mesh of ad-hoc wireless network. All the UAVs must be able to communicate with a navigator UAV either directly or through other UAVs. That means the UAVs may depend on their neighbors and the subsequent nodes on its path to the navigator. Our framework can find out if k number of UAVs are currently unavailable, whether it makes others unsafe/vulnerable or not.

For the communication between two UAVs, we ensure that there is a common encryption method between the two. We consider a wireless protocol that allows the UAVs to use different encryption methods for communicating with different neighbors, according to the neighbors' supported methods.

IV. Formal Model for Resiliency Verification

In this section, we present the formal model of the proposed verification framework. We first model the fuel usage constraints, then we model the position of the UAVs. We consider a network of UAVs on a Cartesian plane. Finally we model the reachability requirements given the communication possibilities between certain UAV pairs. The model takes a pre-specified resiliency time period, T, during which it makes sure that the network of UAVs is safe at different intervals of small time periods, t. We assume that during this time period the velocity and direction of the UAVs remain the same. Table I shows the different modeling variables used in this formal model.

TABLE I
Notation Table

Notation	Definition
$RemTime_{v,t}$	Remaining flying time of UAV v at time t
$RemFuel_{v,t}$	Remaining fuel of UAV v at time t
$\vec{S}_{v,t}$	Speed of UAV v with direction at time t
$\vec{W}_t$	Wind speed and direction at time t
F_v^{max}	Maximum fuel capacity of UAV v
T_{th}	Threshold time for all UAVs to return to refueling station or base
$SafeRemFuel_{v,t}$	Boolean denoting if UAV v has adequate fuel remaining at time t
$Iso_{v,v',t}$	Isolation of UAV v and v' at time t
I_{th}	Minimum required isolation between any two UAVs
$SafeDist_{v,t}$	Boolean denoting UAV v is at safe distance to avoid collision with anyone else
$Dist_{v,v',t}$	Scaler distance between two UAVs at time t
$Neighbor_{v,v',t}$	Boolean denoting if two UAVs are in range for communication at time t
$InRange_{v,t}$	Boolean denoting if UAV v has adequate number of neighbors within its range at time t
$SafePos_{v,t}$	Boolean denoting if UAV v is in a safe position at time t
$SafeRelPos_{v,v',t}$	Boolean denoting if two UAVs are in safe relative position at time t
$CommSec_{v,v'}$	Boolean denoting if two UAVs have common security measures at time t
$Com_{v,v',t}$	Boolean denoting if two UAVs can communicate according to their ranges and security measures
$Reachable_{v,\bar{v},t}$	Boolean denoting if UAV v is reachable to and from navigator $\bar{v}$ at time t
$Uav_{v,t}$	Boolean denoting if UAV v is alive at time t
$Safety_t$	Boolean denoting if the UAV network is safe at time t
$ResilientSafety$	Boolean denoting if the UAV network is resilient at any time, even if k number of UAVs fail

A. Fuel Model

Remaining flying time ($RemTime_{v,t}$) of a UAV is mainly a function of remaining fuel ($RemFuel_{v,t}$), maneuver (*e.g.*, speed with direction, $\vec{S}_{v,t}$), weather conditions (*e.g.*, wind speed and direction, $\vec{W}_t$) at time t. The flying time is calculated according to the following equation, where α, β, and γ represents the impacts of remaining fuel, speed of UAV, and speed of the wind, respectively. The denominator represents the resultant (intended) speed and direction of a UAV with respect to the speed and direction of the wind.

$$RemTime_{v,t} = \frac{\alpha \times RemFuel_{v,t}}{|\beta \times \vec{S}_{v,t} + \gamma \times \vec{W}_t|}$$

There are other parameters such as engine capacity, maneuvers, etc. which impact the remaining flying time of a UAV. However, these are out of the scope of this research. Our model can be easily extended according to any other parameters.

A UAV must maintain its own speed and direction according to the weather conditions to attain a final expected velocity. Our model takes the final expected velocity of the UAVs as input and calculates the actual required velocity. For the sake of simplicity, in this research, we only consider wind speed, but we can easily add any other weather phenomenon. We calculate the magnitude of the resultant velocity of a UAV, $\vec{V}$, with respect to wind speed and direction as follows, where θ is the angle between velocity of the UAV and wind:

$$V = \sqrt{|\vec{S}|^2 + |\vec{W}|^2 + |\vec{S}||\vec{W}|cos\theta}$$

The direction of the resultant UAV velocity is angle α, which is with respect to vector $\vec{S}$:

$$\alpha = tan^{-1}\frac{|\vec{W}|sin\theta}{|\vec{S}| + |\vec{W}|cos\theta}$$

All UAVs have a maximum capacity of fuel. Hence, the following is always true: $\forall_{v \in \mathbb{V}}\ RemFuel_v \leq F_v^{max}$

All UAVs must be able to return to safety before remaining flying time ends. The model makes sure that this is true for each small time interval t throughout T. T_{th} is the threshold time within which a UAV can safely return to a base or a refueling station. If the remaining flying time of a UAV is less than this threshold, then it is considered vulnerable.

$$\forall_{t \in \mathbb{T}} \forall_{v \in \mathbb{V}}\ SafeRemFuel_{v,t} \rightarrow RemTime_{v,t} \geq T_{th} \quad (1)$$

B. Position Model

Isolation: Any two UAVs should always maintain a safe distance to avoid mid-air collision. $Iso_{v,v',t}$ denotes the isolation

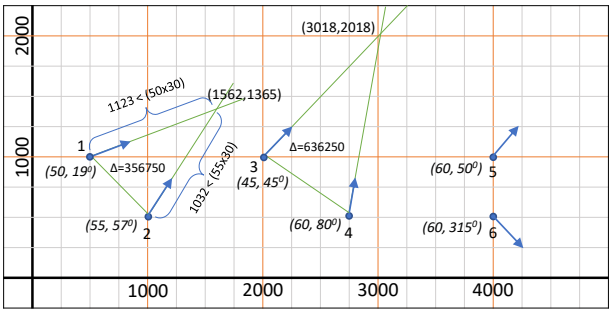


Fig. 3. Calculation of isolation between two UAVs.

of any two UAVs at time t. This is calculated as a triangle formed by the paths after a pre-specified time $\hat{t}$ from the current instance, according to the velocity of the UAVs, relative to wind speed and direction, and their current positions at time t. For example, in Fig. 3, 6 UAVs are shown on a Cartesian plane. UAV 1 is traveling at a speed of 50 m/s at an angle of 19° with positive x-axis. UAV 2 has a speed of 55 m/s and a direction of 57°. Their straight line trajectories meet at the intersection point of (1562,1365). If $\hat{t} = 30s$, after this time, the distance covered by both the UAVs will be 1500 m and 1650 m, respectively. The distances from their current positions and the intersection points are 1123 m and 1032 m, respectively, which are less than their projected covered distance in 30 s. The area of the triangle formed by the three points is 356,750 m^2, which is the calculated isolation for this pair. If we consider a threshold isolation of 500,000 m^2, then UAV 1 and UAV 2 are not at a safe distance given their trajectories. On the other hand, UAV 3 and 4 have an isolation of 636,250 m^2, which is greater than the threshold. So they are considered to be safe from each other. We will check this for every seconds of a total time period of 10 s, if $T = 10s$. UAV 5 and 6 have trajectories such that they will not meet within the next 30 s. Hence, they are safe, too.

$SafeDist_{v,t}$ is the variable denoting if a UAV is located at a safe distance from all other UAVs in the network at time t. The isolation constraints can be formed as:

$$\forall_{t\in\mathbb{T}}\forall_{v\in\mathbb{V}}\ SafeDist_{v,t} \rightarrow \neg\exists_{v'\in\mathbb{V},v\neq v'}\ Iso_{v,v',t} < I_{th} \quad (2)$$

If the coordinates of the two UAVs in a Cartesian system at time t are $(v_{x,t}, v_{y,t})$ and $(v'_{x,t}, v'_{y,t})$, and the intersection coordinate of the trajectories, if they intersect after $\hat{t}$ seconds, is $(\hat{v}_{x,\hat{t}}, \hat{v}_{y,\hat{t}})$, then we can calculate $Iso_{v,v',t}$ as follows:

$$Iso_{v,v',t} = \left|\frac{v_{x,t}(v'_{y,t} - \hat{v}_{y,\hat{t}}) + v'_{x,t}(\hat{v}_{y,\hat{t}} - v_{y,t}) + \hat{v}_{x,t}(v_{y,t} - v'_{y,\hat{t}})}{2}\right|$$

Range: Let $Neighbor_{v,v',t}$ denote whether UAV v and v' are neighbors within each others communication range at time t. If $Range_v$ is the range of UAV v, then following expression holds:

$$\forall_{t\in\mathbb{T}}\forall_{v,v'\in\mathbb{V},v\neq v'}\ Neighbor_{v,v',t} \rightarrow (Dist_{v,v',t} \leq Range_v) \wedge (Dist_{v,v',t} \leq Range_{v'}) \quad (3)$$

TABLE II
ENCRYPTION METHODS

ID (k)	Encryption Method	Encryption Decision (c_v^k)	Score (S_v^k)
1	AES 256	c_v^1	4
2	Wired Equivalent Privacy (WEP)	c_v^2	3
3	Wifi Protected Access (WPA)	c_v^3	2
4	Frequency-Hopping Spread Spectrum (FHSS)	c_v^4	1

The distance between two UAVs at time t, $Dist_{v,v',t}$ is calculated as follows:

$$Dist_{v,v',t} = \sqrt{(v_{x,t} - v'_{x,t})^2 + (v_{y,t} - v'_{y,t})^2}$$

One UAV is considered to be in range if it has some neighbors within its specified range. If N_{th} specifies the minimum number of neighbors required for safe communication for a UAV v, then the following is true:

$$\forall_{t\in\mathbb{T}}\forall_{v\in\mathbb{V}}\ InRange_{v,t} \rightarrow \sum_{v'} Neighbor_{v,v',t} \geq N_{th} \quad (4)$$

If two UAVs maintain a safe distance between them such that, at t, there is no possibility of future collision, however, the distance is within their communication range, they are in safe relative position. This needs to be true for all timesteps t throughout T.

$$\forall_{t\in\mathbb{T}}\forall_{v\in\mathbb{V}}\ SafePos_{v,t} \rightarrow SafeDist_{v,t} \wedge InRange_{v,t} \quad (5)$$

C. Security Model

Table II specifies different encryption mechanisms practiced by UAVs while communicating with each other. The table shows the scores of different methods, where the score is basically a function of the time taken to break the corresponding algorithm. In other words, for the same key length, the encryption method providing higher security obtains higher score. c_v^k is a boolean variable which denotes whether a UAV v uses encryption method k or not.

The minimum score of encryption between any two UAVs in the network should be greater than or equal to a certain threshold K. If $CommSec_{v,v'}$ represents a variable to denote common security/encryption measures between any two UAVs v and v', then the following should hold:

$$\forall_{v,v'\in\mathbb{V},v\neq v'}\ CommSec_{v,v'} \rightarrow \exists_k\ c_v^k \wedge c_{v'}^k \wedge (s_v^k \geq K) \wedge (s_{v'}^k \geq K) \quad (6)$$

Two UAVs are able to directly communicate at time t if they are in range with each other at that time and have common security methods. We also make sure that the score of the security profiles of all neighboring UAVs should be greater than a user specified threshold. Let $Com_{v,v',t}$ represent the direct communication between two UAVs at time t. It can be formalized as follows:

$$\forall_{t\in\mathbb{T}}\forall_{v,v'\in\mathbb{V},v\neq v'}\ Com_{v,v',t} \rightarrow Neighbor_{v,v',t} \wedge CommSec_{v,v'} \quad (7)$$

D. Reachability Model

Let $Uav_{v,t}$ denote the availability of a UAV v at time t. Let $\bar{v}$ be a UAV that navigates the other UAVs in the UAV set $\mathbb{V}$. That means, any UAV in the set should be able to reach the navigator at any instant t. We define this property as $Reachable_{v,\bar{v},t}$. It is required that there exists some paths from UAV v to the navigator UAV $\bar{v}$. All the UAVs on these paths should also be alive at time t.

While communicating with the navigator, the network should maintain the low latency required for the kinds of tasks the swarm performs. The navigator should also be able to send required commands within a certain threshold time. As the latency between a UAV and the navigator is directly proportional to the number of hops in the communication, the following should hold:

$$\forall_{t\in\mathbb{T}}\forall_{v\in\mathbb{V}}\ Reachable_{v,\bar{v},t} \rightarrow\ \exists_p(\forall_{l\in P_{v,\bar{v},p}}\ (i,j)\in l \\ \wedge\ Uav_{i,t}\wedge Uav_{j,t}\wedge Com_{i,j,t})\wedge(|P_{v,\bar{v},p}|\leq Hop_{th}) \tag{8}$$

E. Resiliency Model

A UAV is not safe at time t, if although being alive, it does not have safe remaining fuel to return to a refueling station, or does not maintain a safe position with respect to others, or cannot reach the navigator successfully. These UAVs are vulnerable to cyberattacks by adversaries.

$$\forall_{t\in\mathbb{T}}\forall_{v\in\mathbb{V},\,Uav_{v,t}}\ \neg Safety_{v,t}\ \rightarrow\ \neg SafeRemFuel_{v,t} \\ \vee\ \neg SafePos_{v,t}\ \vee\ \neg Reachable_{v,\bar{v},t} \tag{9}$$

$k-$Resilient Safety: We define a network of UAVs as $k-$resilient, if the system is safe for some UAVs, in spite of k number of unavailability of UAVs. That is, even if k number of UAVs become unavailable due to cyberattack or any other failure, a certain percentage r of the remaining UAVs can still create a safe network according to the constraints. If there exists any timestep t during which safety is not ensured for the remaining UAVs, we consider the network as non-resilient.

$$\neg ResilientSafety\ \rightarrow\ \exists_{t\in\mathbb{T}}\sum_{v}Uav_{v,t}\geq(|\mathbb{V}|-k) \\ \wedge\ \frac{\sum_{v\in(|\mathbb{V}|-k)}\neg Safety_{v,t}}{|\mathbb{V}|-k}\geq(1-r) \tag{10}$$

V. Prototype Implementation

The objective of our formal verification is to determine the resiliency of a UAV network satisfying various requirements as well as constraints. Thus, the verification problem is formalized as the satisfaction of the conjunction of all the constraints in Equations 1 through 10. We implement our model by encoding the network configuration and the constraints into SMT logics [5]. In this encoding purpose, we use Z3, an efficient SMT solver [21].

The solver checks the verification constraints and provides a satisfiable (SAT) result if all the constraints are satisfied. Otherwise, it returns an unsatisfiable (UNSAT) result. The SAT result provides a SAT instance, which represents the value assignments to the parameters of the model. According to our objective, we require the assignments to the following variables:

1) The decision variable referring to whether a network is resilient to cyberthreats, $ResilientSafety$.
2) Variables that determine the safety of the network at any particular time, *e.g.*, the boolean denoting whether a UAV is safe in terms of adequate fuel to return to safe location anytime, $SafeRemFuel_{v,t}$, the boolean denoting if all the UAVs hold safe positions to avoid mid-air collisions, as well as maintaining communication range with neighbors, $SafePos_{v,t}$, and the variable denoting the reachability of all the UAVs to and from a navigator, $Reachable_{v,\bar{v},t}$.
3) $Neighbor_{v,v',t}$ and $CommSec_{v,v'}$ provides the verification of whether two UAVs are within each other's range and whether they have common and sufficient security measures to communicate, respectively.
4) $Uav_{v,t}$ provides us with the information of whether a UAV is available (alive) currently or not.

TABLE III
Sample Code

```
1. assert(=> (SafeDist_1_1) (not (or
                                (not Isolation_1_2_1
                                 not Isolation_1_2_1
                                 ...
                                 not Isolation_1_10_1))))
2. assert(=> (SafePos_1_1) (and
                          (SafeDist_1_1 InRange_1_1)))
3. assert(=> (Com_1_2_1)
             (and (Neighbor_1_2_1 CommSec_1_2)))
4. assert(=> (Reachable_4_0_1) (and (
                          (and (Uav_4_2_1 Com_4_2_1))
                          (and (Uav_2_1_1 Com_2_1_1)))))
```

We model our solution as an assertion of non-resiliency of a UAV network. That means, a SAT result determines that the network is unsafe or non-resilient to cyberattacks. In other words, there exists at least one UAV in the network that is not safe at least one timestep throughout the user specified resiliency period. In this case, we print all the value assignments of the variables enumerated above in a text file. A 'true' value to any of these variables means the corresponding requirement/constraint is satisfied. An UNSAT result means that the network is resilient to cyberthreats throughout the resiliency period. Table III shows some SMT-Lib code snippet generated by the tool. For example, line 2 in the code shows that if at timestep 1, UAV 1 is considered in safe position, then it must be true that it has safe distance with all other UAVs, as well as it has adequate number of neighboring UAVs. Line 4 suggests that, if UAV 4 can be reachable to and from the navigator UAV 0, that means, all the links between these two should be up, and the UAVs on both ends of each link can communicate with each other.

TABLE IV
THE INPUT TO THE CASE STUDY (SECTION VI-A)

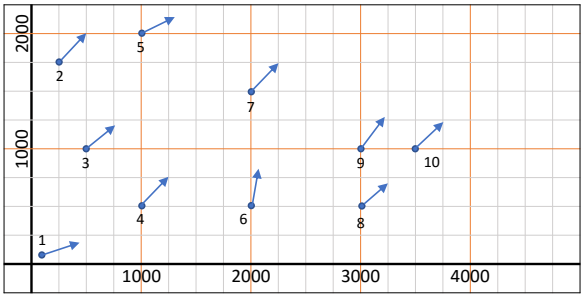
```
# Number of UAVs
10
# UAV x-coordinates
100 250 500 1000 1000 2000 2000 3000 3000 3500
# UAV y-coordinates
100 1750 1000 500 2000 500 1500 500 1000 1000
# UAV velocities (m/s)
50 55 60 45 55 60 50 55 50 60
# UAV angles (degree with x-axis)
30 45 40 45 35 85 45 40 50 45
# UAV ranges (m)
2000 1200 1500 3500 2500 2000 1800 2400 1100 1100
# Remaining fuel (l)
50 35 35 40 30 25 30 30 25 20
# UAV type and encryption methods
1 1 2 3
2 1 2 3 4
2 1 2 4
1 2 3 4
2 1 2 3
2 1 3 4
1 1 2 3 4
2 3 4
1 2 3 4
1 2 4
# Wind speed (m/s)
20
# Wind direction (degree with x-axis)
20
# Expected resiliency period (s)
10
# Number of unavailable UAVs
0
# Expected percentage of safe UAVs except the unavailable ones
100
```

VI. A SYNTHETIC CASE STUDY

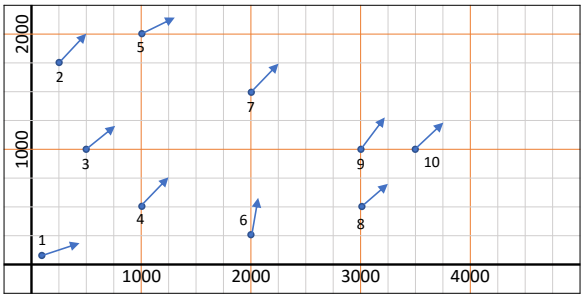
In this section, we discuss the model's execution with two example case studies. In both case studies, we consider a network of 10 UAVs which have positions defined on a Cartesian plane. Their properties, as well as the requirements are provided from an input file.

A. Example: Scenario 1

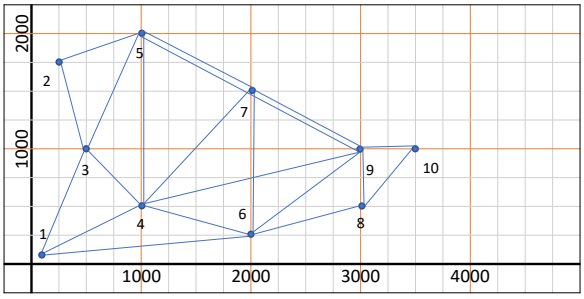
In the first case study, we consider 10 UAVs located at positions (100, 100), (250, 1750), (500, 1000), (1000, 500), and so on, on a Cartesian plane, as shown in Fig. 4(a). The corresponding input file is presented in Table IV. The velocity and the direction with the x-axis of each UAV are given for the current time. The velocity is the resultant velocity of the actual speed and direction of the UAV and wind. The framework calculates the actual speed and direction of each UAV which should be maintained to achieve the given resultant velocity. Hence, the remaining flying times of all the UAVs are calculated using the available fuel, which should always be greater than a given threshold. We also calculate the isolation of all the UAV pairs considering the velocity and direction of them. If two UAVs can meet within the next 30 seconds (a pre-specified time), we calculate the area of the triangle formed by the current positions of the UAVs and the potential intersection point. The area needs to be greater than a threshold to ensure avoiding mid-air collisions. The communication ranges of the UAVs, as well as their types and available encryption methods, are provided. For example, the



(a)



(b)

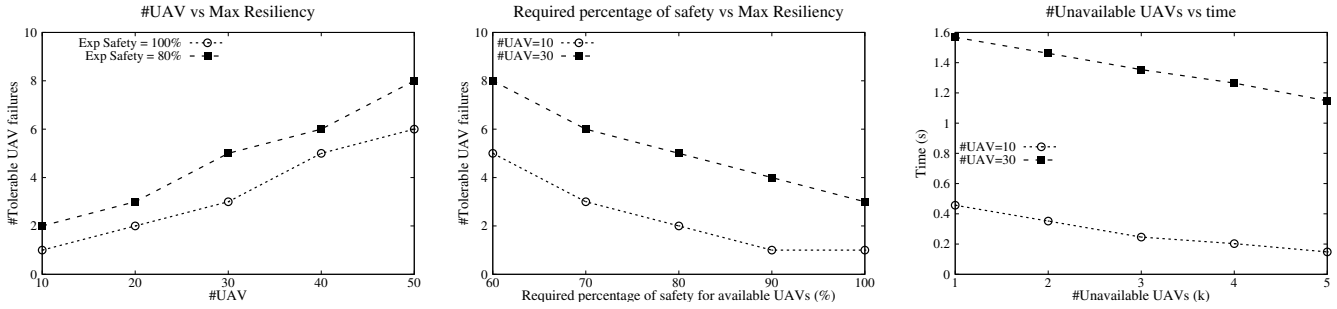


(c)

Fig. 4. (a) A network of UAVs plotted on a Cartesian plane and (b) The modified network of the UAVs to obtain safety, (c) Connectivity of UAVs according to communication range and security constraints.

first UAV is of type 1, and supports three different encryption mechanisms when communicating with the neighbors, while the second one is of type 2 and is capable of 4 different kinds of mechanisms. For this case, we consider that all the UAVs are functioning and available. Hence, the total number of unavailable UAVs is 0. We consider the UAV at point (100, 100) as the navigator for this case. This means, all the other UAVs must be able to communicate with the navigator residing at (100, 100). If all the preceding constraints and requirements are met, we consider the network as safe at the current instant. A resiliency period of 10 seconds is also specified in the input file, which means that the network needs to be safe at each second starting from the current time and ending at 10 s.

The framework returns a SAT result for this network, which means there is at least one UAV which is not safe and vulnerable to cyberattacks at one or more timesteps between 0 and 10 s. As a result, the network is not resilient to threats,



(a) (b) (c)

Fig. 5. (a) Number of tolerable UAV failures w.r.t. number of UAVs, (b) Number of tolerable UAV failures w.r.t. expected percentage of safety for available UAVs (for 20 UAVs), (c) Verification time w.r.t. number of unavailable UAVs.

despite the fact that all the UAVs are available and functioning. From the value assignments to the model variables, we observe that UAV 6 and 7 are not in isolation, which means that they have a chance to collide with each other within 30 seconds. This made the network non-resilient.

B. Example: Scenario 2

This case is presented in Fig. 4(b). In this case, we change the positions of the vulnerable UAVs. For example, UAV 6 is moved to x-coordinate of 250. If the wind speed and direction, as well as the velocity of all the UAVs, and all other properties remain the same, our framework provides an UNSAT result for this network. This means that there is no UAV in the network which is vulnerable to cyberattacks according to the model. In other words, the network is potentially safe. Fig. 4(c) shows the connectivity between UAVs in the network at the current instance according to the mutual communication range and security techniques.

However, if we start changing the number of unavailable UAVs in the input file, we get different results. The expected percentage of safe UAVs, in this case, is 100%. That is, if any of the UAVs fails, it must not impact the safety of any other UAVs in the network. In other words, all the available UAVs should have their safety ensured, even if some other UAVs fail. If the number of unavailable UAVs is changed to 1, we still get an UNSAT result, meaning all other 9 UAVs have their safety intact. If it is changed to 2, we get a SAT result for this network. This means, if two UAVs fail for some reason and becomes unavailable in the network, one or more other UAVs will start to have safety issues in terms of communication with the navigator. By analyzing the values of the model variables, we can see that if UAV 8 and 9 both are unavailable, then it impacts UAV 10's reachability from the navigator. Again, if UAV 6 and 9 both fail, UAV 9 and 10 both lose their reachability which makes them vulnerable. Our model can find out all possible combinations of unavailable UAVs that make the system unsafe. According to our definition, this network is a $1-$resilient UAV network and can tolerate up to 1 UAV failure. In other words, even if 1 UAV fails or becomes unavailable, all the other UAVs can operate safely. In case two UAVs fail together, they will impact the safety of at least one other UAV.

VII. Evaluation

In this section, we present the evaluation of our proposed verification framework. We first present the analysis on the relationships among different deployment parameters such as how the speed of the UAVs affect the resiliency verification results. Then we present the performance (*i.e.*, scalability) analysis of the tool.

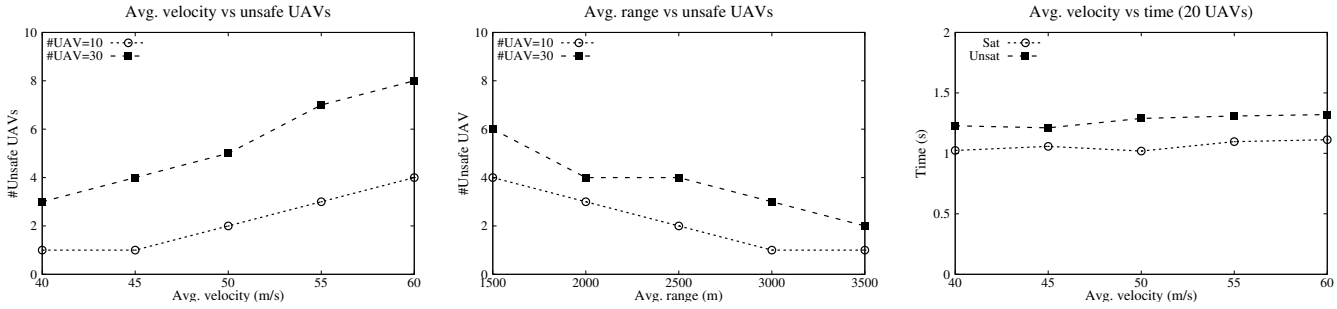
A. Methodology

To evaluate our framework, we ran experiments on different synthetic network topologies of $10-50$ UAVs. The speeds and directions, as well as ranges of the UAVs are varied. We also changed the number of unavailable or failed UAVs and the required percentage of safety of the available ones. The developed tool was run on a machine running Windows 10 OS with an Intel Core i7 Processor and 16 GB memory.

B. Relationships between UAV Parameters and Resiliency

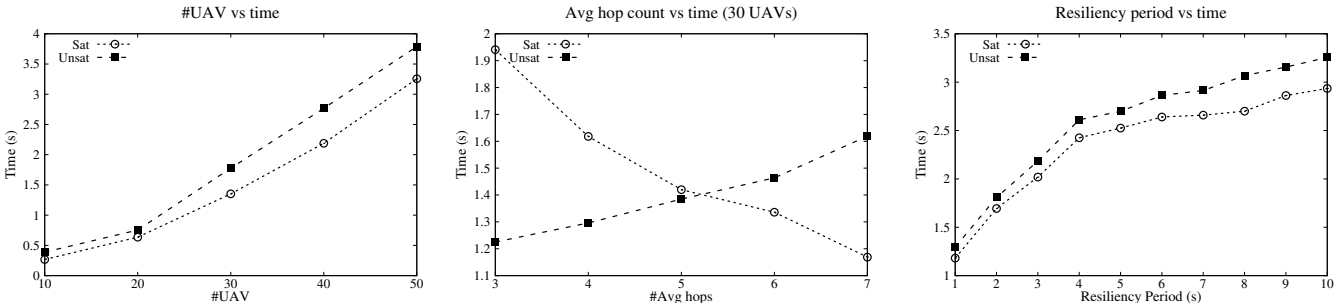
In this analysis, we ran a number of experiments on similar randomly generated network topologies of UAVs. We find the maximum resiliency of UAV networks by varying the network size and the required percentage of safe UAVs. According to our model, a network can be unsafe even if all the UAVs in it are available/alive. In this case, one or more UAVs have any of the fuel issue, position issue, or reachability issue. These issues can be avoided by positioning all the UAVs in such a way that there exists multiple paths and neighbors ensuring adequate isolation. However, in such a 'safe' network, if one or more UAVs become unavailable for some reason, they can create reachability problems for others, leading to an unsafe network. We want to analyze such networks and figure out how many UAVs are needed to be unavailable in order to create problems for others (make them unsafe).

Maximum Resiliency: We present the number of tolerable failures (maximum resiliency) for different sizes of networks in Fig.5(a). The number of tolerable failures is the maximum number of unavailable UAVs (k) that produces an UNSAT result for a given network. Any number of unavailable UAVs that is greater than the value of k will produce a SAT result. We show two cases: one for the expected safety of available UAVs of 100%, and another for 80%. We can observe that the



(a) (b) (c)

Fig. 6. (a) Number of unsafe UAVs found in SAT results w.r.t. average velocity, (b) Number of unsafe UAVs found in SAT results w.r.t. average range, (c) Verification time w.r.t. average velocity of the UAVs (for 20 UAVs).



(a) (b) (c)

Fig. 7. Verification time w.r.t. (a) number of UAVs for SAT and UNSAT results, (b) avg. hop count for UAVs to reach navigator, (c) resiliency period.

maximum resiliency increases almost linearly with the increase of network size. This is because when there are more UAVs in a network, there are usually more alternative paths for the UAVs to communicate with the navigator. That is why the network becomes more resilient to failure of some UAVs. In other words, more UAVs need to fail for a larger network in order to make it unsafe.

We vary the required percentage of safety and observe the maximum resiliency in Fig. 5(b). As the requirement increases for a particular network, the resiliency decreases. Again, a larger network is more resilient than a smaller network, if other factors remain constant. For example, if the requirement is to have safety for 100% of the available UAVs, then a 30-UAV network has a resiliency of 3-UAV failure, while a 10-UAV network has a 1-UAV resiliency.

Safety of Available UAVs: We present the number of unsafe UAVs by varying the average velocity and range in Fig. 6(a) and Fig.6(b), respectively. In both cases, the value of k is kept constant and the framework generates SAT results, which denotes the unsafety of the network. As the average velocity increases, we can observe that the number of unsafe UAVs increases for a particular network. This is because the number of pairs not in isolation increases as the velocity of the UAVs increases. We show this for two cases: for a total number of UAVs of 10 and 30. For greater number of total UAVs, there are more vulnerable UAVs. On the other hand, as the average range of the UAVs increases, it makes more UAVs resilient.

C. Verification Time Analysis

We evaluate the scalability of the tool by analyzing the time required to verify the resiliency of the swarm network by varying different parameters. The verification time includes the model generation time and the constraint verification time. We do not present any comparison of performance with other works, there is no other work that deals with the same problem of UAV network resiliency verification using SMT.

Impact of Number of UAVs: The model synthesis time with respect to varying number of UAVs is shown in Fig. 7(a). The figure shows the time required for both SAT and UNSAT results for different number of UAVs. We chose a resiliency time period of 1 s in this case. That is, the verification is performed for 1 timestep only. In this experiment, for a particular network size, we obtain SAT and UNSAT results by changing the value of k, *i.e.*, the number of unavailable UAVs. The time for an UNSAT result is measured for the maximum possible value of k which yields an UNSAT result. The time for a SAT result is for the minimum possible value of k. We observe that the required time increases with the increment of the number of UAVs. Verification of more constraints is required as the model size increases, and more time is required to reach a solution. The increase of verification time lies between linear and quadratic orders. It can also be observed that the time required for SAT and UNSAT results differs for a certain number of UAVs. To reach an UNSAT solution, all possible solutions are checked by the model checker, whereas to reach a SAT result, only one solution is adequate.

Fig. 7(b) presents the time required for different numbers of average hops for UAVs to reach the navigator. The experiment is run on a network of 30 UAVs. We change the initial positions of the UAVs in such a way that they can form a network according to their mutual range, where the minimum number of hops to reach the navigator varies. Thus, we change the average number of hops and measure the time for resiliency verification. We show two scenarios: one is for SAT and the other is for UNSAT result. We observe that, for SAT result, the verification time decreases as the average number of hops increases. This happens because, as the number of hops increases, it becomes easier for the verifier to find a scenario where a UAV can become unsafe because of its increased distance from the navigator. In this case, the unsafe space increases which makes it easier to find a SAT result. On the other hand, for UNSAT result, the model has more clauses to check as the number of hops increases. Eventually, to find an UNSAT result, it has to search the whole problem space and takes more time.

Impact of Resiliency Specification and UAV Properties: Fig. 7(c) shows the verification time analysis for different required resiliency time periods. We may recall that the resiliency period is the required time period throughout which the resiliency of the network of UAVs needs to be verified. It is obvious that the more the time period is, the more clauses need to be checked. As a result, the verification time increases with the increment of resiliency period. The increment of verification time is almost linear. Our model can provide solutions for longer periods without requiring an exponential amount of time.

Fig 5(c) presents the time required for verification with respect to the number of unavailable UAVs (k). As the value of k increases, it becomes easier for the system to find SAT results. As a result, the time decreases with the increment of k for a particular network. We demonstrate the fact for two different networks.

We also observe the model synthesis time by varying the average resultant velocity of the UAVs. Fig. 6(c) shows the results of this analysis. We observe that change in the velocity does not impact the verification time very much. The time remains almost the same as we increase the average velocity of 20 UAVs considered in this case. In any cases observed, the verification time is small enough for the navigator to issue commands to any unsafe UAV for returning to safety.

VIII. Conclusion and Future Works

UAV swarms are recent trends in research and industry. We present a tool that automatically verifies the resiliency of such a network. It formally models different UAV parameters such as positions, ranges, encryption methods, etc., as well as network requirements and resource constraints. Then it solves the model using an efficient solver. The output can determine whether a network is safe from cyberthreats or not, even if some of the UAVs are unavailable. If not, it can specify the UAVs that are vulnerable, which can be instructed to change their positions to make the network safe. It can also specify which UAVs make the network unsafe by being unavailable. The evaluation results show that the tool is scalable enough to use in practical situations to send commands for relocating UAVs. In future, we plan to incorporate our framework with UAVs positioned in the three-dimensional coordinate systems. We also plan to perform trajectory prediction while calculating the isolation between any two UAVs and satisfying secured communication between them. Energy management based on different types of fuel cells is another research area that can be associated with our proposed tool.

References

[1] W. L. Teacy, J. Nie, S. McClean, G. Parr, S. Hailes, S. Julier, N. Trigoni, and S. Cameron, "Collaborative sensing by unmanned aerial vehicles," 2009.

[2] "Fleet of cargo drones tested in the amazon," https://werobotics.org/blog/2017/10/18/fleet-of-cargo-drones-tested-in-amazon-rainforest/.

[3] "Pentagon issues call for drones that hunt like a pack of wolves," https://defensesystems.com/articles/2015/01/22/darpa-drones-pack-of-wolves-autonomy.aspx.

[4] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in uav communication networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1123–1152, 2016.

[5] L. de Moura and N. Bjørner, "Satisfiability modulo theories: An appetizer," in *Brazilian Symposium on Formal Methods*, 2009.

[6] S. Mahmoud and N. Mohamed, "Broker architecture for collaborative uavs cloud computing," in *Collaboration Technologies and Systems (CTS), 2015 International Conference on*. IEEE, 2015, pp. 212–219.

[7] E. Rivera, R. Baykov, and G. Gu, "A study on unmanned vehicles and cyber security," *Texas, USA*, 2014.

[8] J. Tisdale, Z. Kim, and J. K. Hedrick, "Autonomous uav path planning and estimation," *IEEE Robotics & Automation Magazine*, vol. 16, no. 2, 2009.

[9] R. W. Beard, D. B. Kingston, M. Quigley, D. Snyder, R. Christiansen, W. Johnson, T. W. McLain, and M. A. Goodrich, "Autonomous vehicle technologies for small fixed-wing uavs." *JACIC*, vol. 2, no. 1, pp. 92–108, 2005.

[10] N. Ceccarelli, J. J. Enright, E. Frazzoli, S. J. Rasmussen, and C. J. Schumacher, "Micro uav path planning for reconnaissance in wind," in *American Control Conference, 2007. ACC'07*. IEEE, 2007, pp. 5310–5315.

[11] S. A. Bortoff, "Path planning for uavs," in *American Control Conference, 2000. Proceedings of the 2000*, vol. 1, no. 6. IEEE, 2000, pp. 364–368.

[12] J. Kvarnström and P. Doherty, "Automated planning for collaborative uav systems," in *Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on*. IEEE, 2010, pp. 1078–1085.

[13] B. Bethke, J. P. How, and J. Vian, "Group health management of uav teams with applications to persistent surveillance," in *American Control Conference, 2008*. IEEE, 2008, pp. 3145–3150.

[14] E. Kuiper and S. Nadjm-Tehrani, "Mobility models for uav group reconnaissance applications," in *Wireless and Mobile Communications, 2006. ICWMC'06. International Conference on*. IEEE, 2006, pp. 33–33.

[15] S. Mahmoud and N. Mohamed, "Collaborative uavs cloud," in *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*. IEEE, 2014, pp. 365–373.

[16] A. Renault, "A model for assessing uav system architectures," *Procedia Computer Science*, vol. 61, pp. 160–167, 2015.

[17] N. P. Borges, C. G. Ghedini, and C. H. Ribeiro, "A local communication model for improving stealth in collaborative uav networks." *Anais do Computer on the Beach*, pp. 249–258, 2017.

[18] A. Tsourdos, "A formal model approach for the analysis and validation of the cooperative path planning of a uav team," 2005.

[19] G. Sirigineedi, A. Tsourdos, B. A. White, and R. Zbikowski, "Modelling and verification of multiple uav mission using smv," *arXiv preprint arXiv:1003.0381*, 2010.

[20] X. Zhu, Z. Liu, and J. Yang, "Model of collaborative uav swarm toward coordination and control mechanisms study," *Procedia Computer Science*, vol. 51, pp. 493–502, 2015.

[21] L. de Moura and N. Bjørner, "Z3: An efficient SMT solver," in *Conf. on Tools and Algo. for the Constr. and Analysis of Sys*, 2008.