UASTrustChain: A Decentralized Blockchain-based Trust Monitoring Framework for Autonomous Unmanned Aerial Systems

MAHSA KESHAVARZ1, MOHAMMED GHARIB1, FATEMEH AFGHAH 1, AND JONATHAN ASHDOWN 2

1School of Informatics, Computing, and Cyber Systems (SICCS), Northern Arizona University, Flagstaff, Arizona, 86011, USA (e-mail: {mk959, mohammed.algharib, fatemeh.afghah }@nau.edu)
2Air Force Research Laboratory, Rome, NY 13441 USA (e-mail: Jonathan.Ashdown@us.af.mil)

Corresponding author: Mahsa Keshavarz (e-mail: mk959@nau.edu).

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-20-1-0090 and the National Science Foundation under Grant Number 2034218. Distribution A: Approved for Public Release, distribution unlimited. Case Number 88ABW-2020-2588 on 5 Sep. 2020.

ABSTRACT Unmanned aerial systems (UASs) are prone to several cyber-physical attacks, which decrease the performance of the network and may cause damage to the unmanned aerial vehicles (UAVs) or their surrounding environment. In this paper, we propose UASTrustChain, a trust management framework based on Blockchain time-stamped series. We consider a system of models, consists of a number of autonomous UAVs, whose behaviors are regularly monitored by a set of distributed observers (DOs). Since most cyber attacks cause interruption in the operations of UAVs or deviation from their original path, the DOs keep track of the UAVs’ behavior in terms of their trajectory, as well as the number of their successful tasks. The DOs calculate a relative trust score for each UAV and keep these scores in a transparent, reliable, secure and open ledger. This framework can detect UAVs’ abnormal behavior in a real-time manner further to detect the compromised distributed observers, if any. The proposed framework could also distinguish abnormal activities due to real attacks from those caused by harsh environmental conditions. We evaluate the proposed framework for its functionality and accuracy by performing extensive simulation experiments. Our simulation results show that the proposed trust model can detect compromised distributed observers and fades their effect on the UAVs trust scores. Results further show the ability of the system in detecting malicious UAVs, which can be under various cyber-physical attacks.

INDEX TERMS Blockchain, Cyber-physical attacks, Decentralized trust monitoring, distributed systems, UAV systems

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) are being more widely used in several civilian and military operations including remote sensing, surveillance, package delivery, disaster relief and medical services [1]–[7]. Due to their unique features such as high-mobility, ease of deployment, and their ability to hover, UAVs can provide services including urgent Internet and communication services in time-critical missions or natural disasters [8], [9]. In one hand, the UAVs are vulnerable to several attacks, including cyberattacks such as false data injection [10], physical attacks such as targeting the UAVs using firearms [11], and cyber-physical attacks such as GPS spoofing [12]. On the other hand, in multi-agent systems like UAS, the agents need to rely on one another’s services and resources to perform their tasks.
Any malicious behaviour accordingly may lead to serious system breakdown, in long term [13]. Therefore, developing robust trust monitoring mechanisms to identify potential attacks on these systems is crucial.

There are several existing trust management mechanisms as further discussed in Section II to detect malicious and under-attacked UAVs. Most of the proposed trust management approaches look at the history of the UAVs’ trust to evaluate if they have deviated from normal behavior long enough to impact their reputation [14], [15]. Reputation-based trust monitoring methods can be implemented in both centralized or distributed manners. While the centralized approaches suffer from the single point of failure problem, the distributed methods are often vulnerable to agents’ false reports, and require a large memory to store the reputation scores. Another group of trust monitoring approaches can detect an attack in real-time based on observing the statistical measures [16]. To the best of our knowledge, however, these methods cannot distinguish between the malicious behavior or the potential change of behavior due to harsh environmental conditions.

In our previous work [17], we proposed a real-time trust management mechanism, in which the trust score of each UAV relative to its neighbors was calculated at a centralized controller unit. The proposed model in [17] considered a scenario, where only one UAV in each cluster can be compromised at a given time. Such assumption limits the application of this model in large scale UAV networks, where multiple UAVs can be under an attack. Moreover, monitoring the agents’ trust by a central unit could put the system at risk of ‘single point of failure’.

To extend the developed algorithm in [17], we propose UASTrustChain, a novel decentralized trust monitoring framework to detect cyber-physical attacks on one or multiple UAVs in a real time manner. In this model, the trust score of each UAV relative to other agents is regularly calculated by several distributed observers. We proposed an aggregation mechanism for the DOs to combine their observations. After the aggregation, we use the power of Blockchain in efficiently and transparently tamper-proofing the observed trust scores in decentralized manner, to store the trust scores in an open immutable ledger. The trust scores of each UAV are chained in a time-stamped series and linked to one another by cryptographic hashes, to form a Blockchain for each UAV. In this way, the trust scores of each UAV are accessible by all other UAVs, traceable, and immutable. The idea of relatively evaluating the UAV agents’ behavior adds the power of distinguishing between the honest behavioral change due to the environmental conditions from that kind of malicious behavioral change.

In the system of study, each UAV is assigned with its specific tasks, where these tasks pre-define the UAV path. The UAVs are assumed to be autonomous and they have to follow their pre-defined paths. However, each UAV may deviates from its pre-defined path or does not perform all of its tasks because of harsh environmental conditions or being under an attack. GPS spoofing attack, man-in-the-middle attack, and forwarding attack are some instances that lead to malicious behaviour.

The DOs observe two main factors related to behavior of each UAV: (i) how successful a UAV was in performing its assigned tasks, and (ii) the deviation of the UAV from its pre-defined path. By considering these two factors, we can detect the most common attacks on UAV agents. The UAV under GPS spoofing attack is more likely to deviate from the pre-defined path [18]. Also, the UAV under the man-in-the-middle attack or forwarding attack often fails in performing its assigned tasks [19]. This model is robust against compromising a significant fraction of the UAVs.

For each UAV, the DOs form a Blockchain to store its trust scores. We use the Blockchain technology to guarantee the transparency, integrity, and accessibility of the trust scores calculated by the DOs. The proposed trust management framework forms an initial block for each UAV with the initial trust score of that agent, and a new block will be linked for any updates in the trust score of the UAVs. The trust score of each UAV agent will be evaluated compared to its previous trust scores as well as other agents’ trust scores to identify if a UAV has been attacked. Since all the neighbor UAVs experience similar environmental conditions, if a UAV presents a significant trust score variation, the UAV is identified as a potentially under-attack agent.

Furthermore, we account for the possibility of cyber-physical attacks targeting the DOs and implement a mechanism to identify such compromised DOs. The attacker can target some of DOs and force them to report an inaccurate observation or erroneous information. In this case, the false reports can affect the total trust score of under-observation UAVs. To face this issue, we consider a trustworthiness score for each DO as well as a penalty for the DOs, which act maliciously. The penalty affects the trustworthiness of the DO and consequently, its weight in aggregating the observations. If a DO keeps acting maliciously, the mechanism automatically discards its reports. The DO needs to represent good performance to be able to regain its trustworthiness.

We show the accuracy of the proposed trust management framework by performing an exhaustive simulation. The results show this model can detect the attacks in real-time. It could also detect the compromised DO and remove the effect of its reports from the aggregated trust scores. The proposed method can also differentiate the malicious behavior from the undesired behavior due to harsh environmental conditions.

In summary, the contributions of UASTrustChain, the proposed trust management framework for the UAS, are
followed:

- It offers a novel tamper-proof framework for trust management and detection of malicious agents based on Blockchain for multi-agent systems operating in dynamic environments.
- It replaces the need for a single point of failure with distributed observers.
- It can differentiate between the abnormal behaviors due to the attacks and the potential unusual actions due to harsh environmental conditions.
- It works accurately even in the situation that one or more DOs are compromised.
- It keeps the trust scores in an open tamper-proof, time-stamped ledger. Even a compromised DO cannot tamper the trust scores of the UAVs.
- We exhaustively evaluate the proposed mechanism and show its accuracy in calculating the UAV trust scores and detecting the malicious UAV agents as well as compromised DOs.

The rest of this paper is organized as follows. Section II reviews the related works. We define the network and adversary model carefully and represent the model assumptions in Section III. The proposed model is described in Section IV, followed by the simulation setting, results, and analysis in Section V. Finally, we conclude the paper in Section VI and represent our future directions.

II. RELATED WORKS

There is a rich state-of-the-art for trust management in multi-agent systems [13], where it could be categorized into three general approaches, fully centralized, fully distributed, and partially distributed algorithms.

Our recently proposed work [17] is an instance of a fully centralized approach. In this work, there is a central unit that observes the behavior of UAV agents in an unmanned aerial system and calculates a trust score for each agent. The UAVs form different clusters where the central unit can detect the malicious agents when at most, one compromised agent in each cluster exists. Fan et al. [20] proposed another fully centralized trust management system to choose the most trusted paths for data transfer in SDN network. Authors considered the network as a multi-agent system, and looked to the switches and routers as the agents. The authors assumed that their centralized unit has full control over the agents’ route selections. The fully centralized approaches, in most cases, require highly complicated resources for the central unit, at the same time, suffer from a single point of failure problem. The central unit needs to be fully available, tamper-proofed, and physically secure. It further requires to have an accurate observation of all the system agents. Gathering all of the mentioned characteristics altogether significantly increase the cost of implementing such approaches.

Several fully distributed algorithms have been proposed for the systems that all agents have the same level of resources, and the system has no infrastructure or central controller. Mobile ad hoc networks (MANET), implemented in areas that the infrastructure’s existence is not feasible, is an instance of such systems [21]. As an instance, Wei et al. [22] proposed a trust management system for MANETs that considers both the direct observation and the reports from neighboring nodes to calculate a trust score for each neighbor. Cai et al. [23] proposed an evolutionary self-cooperative trust management scheme, another instance of a fully distributed trust management system. In this system, the mobile agents exchange their observations from other agents’ behavior and analyze the received observations based on their cognitive judgment. Eventually, each agent uses its cognition to calculate the trust scores of other agents. Shabut et al. [24] proposed another instance of fully distributed trust management systems based on recommendations of other agents supported by a defense scheme to omit the impact of dishonest recommendations.

Generally, in all fully distributed trust management systems, each agent calculates a trust score for other agents based on its observations or based on other agents’ reports in addition to its own. In such systems, each agent’s trust score might not be unique, and the system suffers from low precision and, consequently, a high false-negative and false-positive rate in detecting misbehave agents. Moreover, the distributed trust monitoring mechanisms impose a high computation load on the agents. Such systems are vulnerable to cooperative attacks, where a set of malicious agents cooperatively make negative observations for the other agents and make positive reports for themselves.

The partially distributed approaches are proposed for the systems that a partial infrastructure is possible. Vehicular ad hoc networks (VANET) where there are some roadside units (RSUs) and the vehicles can be an instance. However, the partially distributed approaches require a robust consensus algorithm for their trust score calculations, which leads the designing of such approaches to be more complicated. Such systems can rely on the agents’ reports or just on the observation of their observers.

Huang et al. [25] proposed a partially distributed trust management approach for VANET. In this approach, the agents generate the reputation rates for their neighbors and upload these rates into the nearby RSU. However, the aggregated reputation scores might not reflect the real reputation of the system agents and might not be consistent. Xiao et al. [26] recently proposed a trust management approach for VANET based on local and global trust values calculated by the vehicles and RSUs, respectively. The total trust score of each vehicle is then calculated from the aggregation
of both scores. This approach works based on the link trustworthiness and hence suffers from the low precision in highly dynamic systems, where the links are formed and expired fast.

Furthermore, several trust management algorithms in different applications use Blockchain technology to utilize its unique characteristics. Yang et al. [27] proposed a trust management algorithm for VANETs and kept the vehicle’s trust scores in a public ledger. This algorithm is fully distributed, where the trust scores are calculated based on the vehicles’ reports. Although the RSUs aggregate the vehicles’ reports, the RSUs act only as a computational unit and do not impact the aggregated value. Hence, the system suffers from the issues of fully distributed systems. Lu et al. [28] proposed BARS, a Blockchain-based algorithm for trust management in VANETs with centralized nature. The goal of this work is to break the linkability between real identities and public keys. However, the centralized nature of this algorithm keeps the concerns of fully centralized techniques for this method. Alexopoulos et al. [29] proposed a trust management algorithm for multi-agent systems based on Blockchain technology. This algorithm aims at studying the advantages of using a distributed ledger on a global scale as a part of a trust management system. However, the model is built based on several central entities to keep the system’s overall structure fits the global scale.

On the one hand, the trust management algorithms that depend on a central unit suffer from a single point of failure problem, and the fully distributed algorithms, in most cases, suffer from low precision and non-consistence trust values. On the other hand, and to the best of our knowledge, there is no general-purpose well-defined distributed trust management framework for UAV to detect the malicious agents and compromised decentralized units with feasible computation overhead. The fact that motivated us to propose UASTrustChain framework.

III. SYSTEM AND ADVERSARY MODEL

In this section, we describe the proposed system of system. It consists of several UAVs and multiple distributed observers, as depicted in Fig. (1). We consider a heterogeneous UAV agents, which can be any types of UAV. Each UAV is assigned with a set of particular tasks that determine the UAV’s waypoints. Each waypoint is a location that the UAV is supposed to meet in order to perform a certain task. Hence, the set of task assigned to each UAV agent forms the UAV flight pattern. The UAVs are expected to complete each task within a certain deadline, or the task will be considered as a failed task. Some examples of these tasks include package delivery, aerial photography, and geophysical survey. Each UAV agent is equipped with a global positioning system (GPS). The UAVs are the risk of being compromised by different attack mechanisms. Therefore, we aim at developing a distributed real-time trust monitoring mechanism to detect the compromised agents.

The task of trust score calculation is done by N distributed observers (DOs), working together to keep the UAV trust scores accurate and consistent. We assume that there is a low-bandwidth reliable communication channel between the DOs and the UAVs for command and control purposes. The DOs have a reliable communication channel between themselves, too.

In our model, the attacker can compromise any of the UAVs. It may also compromise any of the distributed observers. However, we assume that the attacker cannot compromise a large number of distributed observers altogether at the same time. If the attacker becomes able to compromise more than half of UAV agents or more than half of DOs, the entire system is considered as compromised.

We consider two factors to calculate the trust score: the task success score (TSS), and the deviation of the UAV’s flight trajectory from its expected path. The reason behind choosing these two factors is that most common attacks impact one or more of these behavioral factors. For instance, when a UAV is under the man-in-the-middle attack, it usually skips some assigned tasks. Consequently, it presents a low task success rate. Furthermore, when the UAV is under the GPS spoofing attack, the attacker wants to mislead the UAV into another location. Consequently, the UAV deviates from its pre-defined path.

In the proposed trust monitoring model, we consider several DOs instead of the central unit to solve the single point of failure problem, and have more observers to gain more accurate observations of the UAVs. We assume that the DOs might not have an accurate observation of the UAVs’ successful tasks. The distance between the DO and the task location is one of the parameters that may affect the observation. The closer DO may have a more precise observation about the task’s success. In this case, we propose a model to decide which DO has an accurate observation on the UAV, based on its distance from that agent. Fig. (1) shows a general view of the system model. Table (1) shows the notations used through the paper.

IV. PROPOSED ALGORITHM

Our trust monitoring system for UASs calculates a trust score for each UAV agent to detect any abnormal behaviors at given times. This system forms a consistent Blockchain for every single agent, generated by a set of DOs. The trust score is calculated based on the task success score of the UAV and its deviation from the pre-defined path. The proposed model can differentiate between the abnormal behaviors due to the attacks and the potential unusual actions due to harsh environmen-
Observe Agents’ Behavior

Distributed Servers

1) Calculate TSS
2) Calculate Path Deviation
Add the Trust Score to the Blockchain

Blockchain

Block 1
...
Block i

Fig. 1: System Model

Table 1: The table of notations.

| Symbol | Description |
|--------|-------------|
| n      | Number of UAV agents |
| N      | Number of DOs |
| u_j    | j\textsuperscript{th} UAV agent |
| D_i    | i\textsuperscript{th} Distributed Observer |
| \left(b^t_j, d^t_j, u^t_j\right) | Probability of trust, distrust, and uncertainty of UAV u_j at time t. |
| \left(s^t_j, f^t_j, x^t_j\right) | Number of successful, failed, and uncertain tasks of UAV u_j at time t. |
| p'(T_j) | Predicted Task Success Score of UAV u_j |
| p(T_j) | Corrected Task Success Score of UAV u_j |
| w_{(i,j)} | Accuracy Weight for D_i Observation on u_j |
| d_{(i,j)} | Distance Between the u_j and D_i. |
| P^{(t)}_{(j,i)} | Precision of D_i’s report on u_j’s TSS at time t |
| T^{(t)}_{(i)} | Trustworthiness Score of D_i at time t |
| R^{(t)}_{(i)} | The reward added to D_i’s trustworthiness at time t |
| \varphi | The penalty of DO’s trustworthiness |
| \left(x'_j, y'_j, z'_j\right) | Pre-defined Location of the UAV u_j |
| \left(x_j, y_j, z_j\right) | Actual Location of the UAV u_j |
| \alpha | Time Slot |
| \delta_j | Deviation score of u_j |
| T_j | Total Trust Score of UAV u_j |
| w_T | Weight of Task |
| w_b | Weight of Deviation Trust |

TSS

Since physical conditions such as harsh weather affect all UAV agents, the relative trust variation could help to identify if an agent is under an attack or it is facing harsh environmental conditions.

To improve the validity and reliability of measured trust scores at the DOs, we define a trustworthiness level for each DO to account for the familiarity of the DO with each UAV at a given time and the chance of the DO being under an attack. Thus, our model can deal with the cases where the DOs do not have a perfect observation of a UAV’s operation as well as the compromised DOs to keep the trust score of the UAVs accurate and protected.

In this section, we first describe how to calculate the task success score of each UAV agent. Then, we discuss how to calculate the deviation from the pre-defined path and bring it into consideration. We further describe how to calculate the trustworthiness of the DOs and determine its impact in the UAVs’ trust score. Finally, we discuss the total trust score calculation and forming the corresponding Blockchain.

A. TASK SUCCESS SCORE

One of the main factors to evaluate the trustworthiness of an agent is the task success rate. During a flight, each UAV is assigned with several tasks where it is expected to perform each task in a certain period of time. Several
factors including harsh environment conditions or cyber-physical attacks may disturb the normal operation of the agent, but not all in the same way. When the UAV operates under a severe weather condition, it may not be able to complete all of its tasks. However, when the agents are under specific attacks such as the man-in-the-middle attack, or the forwarding attack, they deliver a less number of successful tasks in comparison with the sever weather condition [19].

To evaluate the behavior of UAVs in terms of performing their assigned tasks, we consider the history of completed tasks for the UAVs during each mission. The trust factor is defined in such a way to differentiate whether the underlying reason for not completing the assigned tasks is related to an attack or because the UAV is experiencing any environmental issues. Therefore, the trust factor considers the relative performance of the UAVs rather than their individual performance. In this case, if the UAVs are operating under adverse weather conditions, they all will underperform and deliver a low task completion rate. However, if one of the UAVs is under an attack, this UAV will experience a low task completion rate and will show a significant difference between the last and current trust scores.

The task completion rate of each UAV at a given time can be estimated by all DOs based on their direct or indirect observation of the under-observation UAV. For instance, if the UAV’s mission was to survey a specific area, the DO can determine if the task has been completed based on whether obtaining the image of the entire region or not. Indeed, the DO cannot always have such a direct observation for all the assigned tasks. For instance, if the UAV’s task was to drop a fireball to ignite the fire in a particular area (a mechanism used to initiate controlled fires to prevent wildfires [30]), then the DO might not be able to accurately determine if the ball was dropped. However, it can estimate if the UAV performed the task by observing its impact on the field, i.e. fire in the target area, with some level of uncertainty.

To account for such potential uncertainty in the assessment of the DO in terms of the number of successful or failed tasks for each UAV, we propose a trust factor based on a subjective logic framework (SLF) [31]. Let assume that \( s^j_u \) is the number of successful tasks, \( f^j_u \) is the number of unsuccessful tasks, and \( x^j_u \) is the number of tasks for UAV \( u_j \) at time \( t \) that the DO cannot certainly declare as successful or unsuccessful. Based on this theory, the trust is composed of a vector \( \{ b^j_u, d^j_u, \hat{u}^j_u \} \). The parameters \( b^j_u, d^j_u, \hat{u}^j_u \) respectively represent the probability of trust, the probability of distrust, and the chance of uncertainty of UAV \( u_j \) at time \( t \), where they satisfy \( b^j_u + d^j_u + \hat{u}^j_u = 1 \). The aforementioned parameters are calculated as presented in (1).

\[
\begin{align*}
\hat{b}^j_u &= s^j_u + s^j_u, \\
\hat{d}^j_u &= s^j_u + f^j_u + x^j_u, \\
\hat{u}^j_u &= s^j_u + f^j_u + x^j_u
\end{align*}
\]

Hereby, the observation of \( D_i \) from the task success of UAV agent \( u_j \) is represented as a probability score in the range of \([0 \ 1]\) using (2). For the rest of the paper, we drop the superscripts \( t \) to simplify the notations.

\[
p(T_j|D_i) = \frac{2\hat{b}^j_u + \hat{u}^j_u}{2}
\]

As we mentioned before, every DO may have different observations of a specific UAV. To keep the task success score and consequently, the total trust score of UAVs consistent, we use the predictor-corrector method. In this method, we first predict the TSS of each UAV agent and then enhance the score by considering the trustworthiness of DOs. The predicted TSS for the UAV \( u_j \) is referred to as \( p'(T_j) \) and it could be calculated by (3) in which \( w_{i,j} \) is the weight for the accuracy of \( D_i \)’s observation from UAV \( u_j \).

\[
p'(T_j) = \sum_{i=1}^{N} p(T_j|D_i) \times w_{i,j}
\]

Since the distance of DOs from the UAVs can affect the accuracy of their report on the number of successful tasks, we consider the distance in the accuracy weight as it is shown in (4), where \( d_{i,j} \) is the distance between UAV \( u_j \) and \( D_i \). We discuss how to calculate the distance between the DOs and the UAVs in Section (IV-B). Every DO sends its observed score to all other DOs, and then the predicted total task success score is calculated by every DO. As it is obvious, in these calculations, the trustworthiness of the DOs has not been taken into consideration. We discuss how to consider the trustworthiness of the DOs in Section (IV-C).

\[
w_{i,j} = \frac{1/d_{i,j}}{1/\sum_{i=1}^{N} d_{i,j}}
\]

### B. PATH DEVIATION

Each UAV is assigned with several tasks during its mission before the mission has started. A particular path including waypoints, with each waypoint corresponds to a specific task, is expected for each UAV. The UAVs should follow the expected path and its waypoints one by one. However, due to the different reasons, the actual trajectory of the UAVs may differ from the expected paths. One reason can be extreme weather conditions such as strong wind. However, such deviation could be due to a cyber-physical attack such as a GPS spoofing.

To calculate the deviation of the UAV from its predefined path, and to find out the main reason behind the deviation, the DOs observe the current location of the UAVs based on (5 and 6). The DOs can find the UAV’s
position by measuring the radial distance of the received signal from the UAV. Hereby, the UAV sends a single packet to all DOs simultaneously, including its current time. Using (5), each DO can calculate its distance with the UAV, based on the time of receiving the packet. In this equation, \( t_i \) is the time of \( D_i \) when receiving the packet, \( t_j \) is the time of sending the packet based on \( u_j \)’s clock, and \( \beta \) is a variable that represents the synchronization factor between the precise DO clock and the UAV agent clock. This factor is essential because the UAV’s clock might not be precisely synchronized. In this equation, \( c \) is a constant equal to the light speed. On the other hand, the DOs can form (6) and let it equal to (5). The tuple \((x_i, y_i, z_i)\) in (6) represents the exact location of \( D_i \). Thus, any four DOs can make a system of four linear equations with four variables, i.e. \( x, y, z, \beta \), to find the exact UAV position and the deviation of its clock from the synchronized DO clocks. Thus, given the travel time of the signals and the exact location of DOs, the UAV position can be determined in three dimensions along the x-axis, along the y-axis, and altitude. It is worth mentioning that the Doppler effect may cause some imperfection in our calculations. However, we believe that the Doppler effect is negligible given the common UAV’s speed as well as the relatively small packet size compared to the bandwidth.

\[
d_{(i,j)} = (t_i - \beta - t_j) \times c \tag{5}
\]

\[
d_{(i,j)} = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} \tag{6}
\]

The UAV agent, as we mentioned earlier, sent its position packet to all DOs, but only four honest DOs are required to extract its precise position. To prevent the effect of possible compromised DO, we propose for the DOs to form the system of linear equations and calculate the UAV position in all possible combinations of four DO groups. Then the majority will represent the precise UAV position. DOs use the calculated position and (7) to calculate the deviation from the pre-defined path at each time slot. The tuple \((x_j^{(t)}, y_j^{(t)}, z_j^{(t)})\) is the pre-defined location of the UAV and \((x_j^{(t)}, y_j^{(t)}, z_j^{(t)})\) is the actual location of \( u_j \) at time \( t \). In this equation, the deviation is calculated for the time slot range \([t - \alpha, t]\), where \( \alpha \) is an integer value that represents the number of time steps.

\[
\delta_j = \frac{1}{\alpha} \sum_{\tau = t - \alpha}^t \sqrt{(x_j^{(\tau)} - x_j^{(t)})^2 + (y_j^{(\tau)} - y_j^{(t)})^2 + (z_j^{(\tau)} - z_j^{(t)})^2} \tag{7}
\]

Since the deviation could be in any positive range, we use (8) to normalize it. Equation (8) calculates the deviation score in the range of \([0, 1]\), such that the lower score represents a larger deviation from the pre-defined path.

\[
\delta_j = \frac{\delta_{\text{max}} - \delta_j}{\delta_{\text{max}}} \tag{8}
\]

If the deviation from the expected path is due to facing an object or temporal harsh weather conditions, the UAV is expected to return to the original path within a short time. However, if the UAV has been attacked, this deviation will be observed for an extended time. In this case, the value of the deviation trust score is increased, which impacts the total trust score. Since the trust scores are compared with one another in each time slot, if the UAVs face extreme weather conditions, all UAVs have a considerable long-term deviation from their original paths, and this condition can be differentiated from potential attacks.

**C. TRUSTWORTHINESS OF DOs**

Similar to the UAVs, the DOs are also at risk of various attacks; therefore, we define a trustworthiness score for each DO to keep track of their performance. We assign to each \( D_i \), an equal trustworthiness score \( T_i^{(t_0)} = \frac{1}{N} \)

where \( \sum_{i=1}^{N} T_i^{(t_0)} = 1 \), at the initialization phase of the network. After receiving the reports from all the DOs, each DO calculates the \( p(T_j) \) for each UAV according to (3). The difference between this aggregated \( p(T_j) \) and the report of each DO, i.e. \( p(T_j|D_i) \), is considered as the precision of that DO at time \( t \) concerning the corresponding UAV. We refer to it as \( P_{(i,j)}^{(t)} \). Equation (9) shows how we can calculate this value.

\[
P_{(i,j)}^{(t)} = |p(T_j) - p(T_j|D_i)| \tag{9}
\]

The DO with the lowest precision is subject to a penalty. A low precision value can indicate that the DO is either under an attack, or it did not have a good view to observe the UAV’s behavior; thereby, in both cases it cannot be a reliable source to report the UAV’s condition. Hence, we fade its impact on the value of the total trust score. While the lower precision decreases the trustworthiness value and consequently the DO’s impact on the total trust score, whenever the DO reports precise values it regains its trustworthiness. Considering \( D_i \) as the DO, which has to pay the penalty, the value of penalty referred to as \( \varphi_j \), and it can be calculated according to (10).

\[
\varphi_j = \begin{cases} 
T_i^{(t)} & \text{if } T_i^{(t)} > P_{(i,j)}^{(t)} \\
0 & \text{otherwise}
\end{cases} \tag{10}
\]

According to the penalty calculated in (10), we need
to update the trustworthiness of all DOs. First, we subtract the value of penalty, i.e. \( \psi_j \) from the current trustworthiness score as it shown in (11), where \( T_i^{(t)} \) is the current trustworthiness of \( D_i \), i.e. at time \( (t) \) and \( T_i^{(t+1)} \) is its updated value.

\[
T_i^{(t+1)} = T_i^{(t)} - \psi_j \tag{11}
\]

We further need to update the trustworthiness score of other DOs to keep the equation \( \sum_{i=1}^{n} T_i^{(t+1)} = 1 \) always true. As a reward, we add \( \psi_j \) to other trusted DOs based on their precision. For the reward to be fair, we use (12 and 13) to calculate the fraction of the reward, which each of the remaining DOs has to receive, and how to calculate the updated trustworthiness score of the rest of DOs, respectively. In these equations, \( k = 1, 2, \ldots, N \) and \( k \neq i \).

\[
R_{k}^{(t)} = \frac{1/P(t)}{1/N \sum_{m=1, m \neq i}^{N} P_{(m,j)}^{(t)} - \psi_j} \tag{12}
\]

\[
T_k^{(t+1)} = T_k^{(t)} + R_{k}^{(t)} \tag{13}
\]

As we mentioned before, based on DOs’ distances from the UAV, they might have different observations from the behavior of a UAV. It means that the closer DO to the UAV has a more accurate observation. That is why we calculate the weight for each DO based on (4). However, the DOs are not fully trusted, and the closest DO is prone to be under an attack. To make a fair consideration, we take also the trustworthiness score of each DO into account to evaluate the reliability of its observation. We redesign (3), for calculating the task success score of UAV \( u_j \), to (14) to be a Bayesian formula. In this equation, \( p(t)(D_i) \) is the weight of \( D_i \) observation in time \( t \) based on its distance as well as its trustworthiness score. This value could be calculated according to (15).

\[
p(t)(D_i) = \sum_{i=1}^{n} p(T_j | D_i) \times p(t)(D_i) \tag{14}
\]

\[
p(t)(D_i) = \frac{(w_{(i,j)} + T_i^{(t)})}{2} \tag{15}
\]

D. Blockchain-Based Trust Monitoring

Based on the observation that the DOs have on the behavior of UAVs, they aggregate their calculated task success score and deviation score for each agent, to generate a total trust score.

We then utilize the power of Blockchain to maintain a consistent trust score among all members. The Blockchain is first proposed in [32] and then used by Satoshi Nakamoto to abstract the core techniques of the well-known digital currency Bitcoin [33]. The distributed nature of Blockchain in forming a transparent immutable accessible open ledger differentiates it from other techniques.

At the starting time of the system, the DOs generate a block for each UAV to keep its trust score. Obviously, the first block contains the same value for all UAVs. Then, they keep the updated total trust scores in time-stamped blocks, linked to the former ones, by a mathematical hash function. Hence, there will be an immutable Blockchain for each UAV. The total trust score for UAV agent \( u_j \) can be calculated by (16), where \( w_\gamma \) and \( w_\delta \) are weights, \( w_\gamma + w_\delta = 1 \), and \( w_\gamma, w_\delta \in [0 \ 1] \).

\[
T_j = w_\gamma \times p(T) + w_\delta \times \delta \tag{16}
\]

The total trust score for each UAV is in the range of \([0 \ 1] \). The network administrator defines the weights \( w_\gamma \) and \( w_\delta \) based on the environmental conditions and the network setting. For example, if the system operates in bad weather conditions or an area with many obstacles, the deviation from the pre-defined paths increases. Thereby, the value of \( w_\delta \) has to be much lower than \( w_\gamma \).

In our proposed trust monitoring model, the Blockchain consists of a list of timestamped blocks, where each block stores the aggregated trust score of each UAV at a specific time. Each block includes the header and the body, as it is shown in Fig. (2). Each block’s header includes the timestamp, nonce, which is the answer to a mathematical problem, the block ID, and UAV ID. In the body, we have the hash value of the previous block, the current hash value, and the total trust score of agent \( u_j \). Accordingly, any agent can request the last block of the Blockchain corresponding to any other agent to see whether it is a trusted agent or not.

After adding the total trust score of each UAV into the Blockchain, we calculate the difference between the last trust score of each UAV and the current trust score, to recognize the potentially compromised agents. Since all the UAVs experience similar environmental conditions, the difference between the previous trust score and the existing trust score of all UAVs should be in the same range regardless of the weather is windy or normal. However, if the scale of the difference for one of the UAVs is different from the other UAVs, it means that the UAV is potentially under attack. Algorithm (1) represents the pseudocode of our trust management algorithm.

V. Simulation Results and Analysis

To show the precision of the proposed algorithm in detecting the malicious UAV agents as well as malicious DOs, we designed several simulation scenarios. In this section, we first discuss the simulation setting and then represent the results and analyze the system.
A schematic view of the Blockchain.

**Algorithm 1:** Distributed trust management pseudocode for UAS

A: for UAV $u_j$ and $j = 1, \ldots, n$ do

1) for DO $D_i$ and $i = 1, \ldots, N$ do
   
   $p'(T_j) = \sum_{i=1}^N p(T_j|D_i) \times w_{(i,j)}$
   
   where $w_{(i,j)} = 1/\sum_{i=1}^N d_{(i,j)}$
   
   end

2) for DO $D_i$ and $i = 1, \ldots, N$ do
   
   $P_{(i,j)} = |p'(T_j) - p(T_j|D_i)|$
   
   end

3) Find max($P_{(i,j)}$).

4) Update the trustworthiness score of each DO based on (11 and 13).

5) Calculate $p(T_j) = \sum_{i=1}^n p(T_j|D_i) \times p(D_i)$.

6) Calculate $\delta_j$.

7) Calculate the total trust score based on (16).

8) Create a new block and add the total trust score to the block.

end

B: Compare the difference between the UAV’s last trust score and the current trust score.

C: Detect the malicious UAV with the out of range difference.

**A. SIMULATION SETTING**

We have simulated a UAS with 3 flying UAV agents over a $3.5 \times 3.5$ km area. We have distributed the UAVs uniformly at random in the field before the mission. Each agent moves according to its pre-defined path. The pre-defined path is designed such that it starts from the initial UAV position and then follows a random waypoint (RWP) mobility model [34]. Each waypoint represents a location of a certain task. Once the pre-defined path is planned, it will be saved for that specific agent. For the real trajectory of the agent, we consider random obstacles and random bad weather conditions. The bad weather condition impacts all the agents in the same way. However, the random obstacle positions are distributed randomly in the area and considered as fixed-position obstacles. Each UAV is expected to perform each of its assigned tasks in a specific location and hence with a specific deadline. The deadline for each task is calculated as the time that the UAV is expected to be at that particular position plus a short variance time. The extra time is considered for any unexpected deviations from the pre-defined path. We perform a sensitivity analysis to see how much a compromised DO can affect the total trust score of a UAV agent. We further show the effect of the distance between the UAV and the DO on the calculated trust score.

For the last situation, we consider both the attack and a bad weather condition to evaluate the performance of the proposed mechanism. We show how our algorithm is able to differentiate between the UAVs’ abnormal behavior when they are under an attack and when they are operating under harsh environmental conditions. It is worthy to mention that, since the block generation in Blockchain is performed by the consensus among all DOs, the generated blocks are always consistent among all DOs. Hence, the correctness of the generated block is based on the correctness of the computed trust value.
which is evaluated in the second scenario. For this reason we did not consider the evaluation of the Blockchain in the simulation section.

**B. EXPERIMENTAL RESULTS**

To show how precise *UASTrustChain* is in recognizing malicious DOs, Fig. (3) shows the task success score when one of the DOs generates erroneous reports. The x-axis of this figure shows the difference between the reported TSS with the real one; the y-axis shows the distance of the malicious DO from the UAV, and the z-axis shows the aggregated task success score. The UAV, in this figure, successfully performs all of its assigned tasks. Hence, any TSS values of less than one represents the negative effect of the malicious DO.

Fig. (3a) shows the raw values of (3), i.e. the predictive equation, which takes into the account all of the DO reported trust scores without considering their trustworthiness. It shows the impact of distance on task success score. The less distance between the malicious DO and the UAV results in more negative effect. The worst case for the aggregated task success score is 0.6524 and it happens when the UAV agent has the closest distance to the malicious DO and the DO reports all the agents’ tasks as failed.

Fig. (3b) shows the same value, but in this case, the DOs ignore the false report by considering a threshold. For any DOs that its report differs from the aggregated value more than the threshold 1, we consider its report as a malicious one and discard it. As it is clear from the figure, the malicious DO negatively affects the task success rate up to the point that the difference between its report and the real value is 0.4. After that, the report of the malicious DO is ignored, which leads to the TSS returning to its true value. By considering the threshold, in the worst case, the malicious DO can lower the UAV’s TSS into 0.8610. Although this scenario improves the task success score, it results in increasing the chance of ignoring the report of the accurate DOs. Therefore, we define the trustworthiness metric for the DOs (as defined in Section IV-C).

Fig. (3c) shows the corrected TSS value by considering the trustworthiness of the DOs. In this figure, we can see the impact of trustworthiness factor on the TSS even though the malicious DO is the closest one to the UAV. In the worst case, the task success score is reported as 0.7585. In this figure, we first consider the malicious DO as the most faraway DO from the UAV agent; hence, its trustworthiness score is only slightly decreased. If in our simulations, we first consider it as the closest one, its trustworthiness score turns into zero at the first step and never rises.

Next, we investigate the time required for the trustworthiness of a compromised DO to fall into zero. Furthermore, we investigate the recovery time required for the trust score of the compromised DO, after it recovers from the attack. We observed that each compromised DO needs to report at most three false task success scores to experience zero trustworthiness. We also consider the case when two DOs are compromised. The interesting results show that the value of the trustworthiness falls down to zero for both compromised DOs in no more than three steps. The trustworthiness of the DO with worse report falls down in one step and the other one will lose its trustworthiness in a next couple of steps. However, each DO, to regain its trustworthiness, needs more time than the time required for its trustworthiness to drop to zero. This specific property of *UASTrustChain* prevents the compromised DOs from acting probabilistically malicious to keep their trustworthiness value healthy. Fig. (4) shows the recovery time for a compromised DO with zero trustworthiness. As this figure shows, the distance from the UAV agent plays a certain role in the recovery time.

Fig. (5) aims at representing the UAV’s movement and their deviation from the pre-defined paths. In this scenario, each UAV has to meet 20 waypoints in its trajectory. Fig. (5a) shows the pre-defined paths for all the three UAVs versus their real trajectory, in normal conditions. As it is expected, in normal conditions, the deviation from the pre-defined path is negligible. Fig. (5b) shows the deviation in the case that UAV u1 is under GPS spoofing attack. Hence, this agents reads faulty GPS information and moves far away from its pre-defined path. We add the strong windy condition as possible bad weather. Fig. (5c) shows the results when UAVs experience bad weather conditions, and UAV u1 is under GPS spoofing attack, at the same time. This figure shows that the bad weather condition may deviate the UAV agents from their pre-defined paths; however, the effect of GPS spoofing attack causes much more deviation.

In Fig. (6), we show the effect of path deviation on the total trust score. While Fig. (6a) shows the average results for the agent’s total trust score in different situations, Fig. (6b) shows the results for a relative comparison between the agents under bad weather condition in the accordance with Fig. (5c). In this figure, UAV agent u1 is under GPS spoofing attack. It is worthy to mention that when the UAV agent fly far away from its pre-defined path, it cannot perform its task successfully in the corresponding waypoints. Hence, the total trust score will be affected by both the deviation from the pre-defined path and the failure in performing the tasks in a certain time.

Now, we aim at evaluating the proposed trust management system in detecting the malicious UAVs. Hence, we show in Fig. (7) the variation of UAVs’ total trust
scores in their Blockchains over time. Fig. (7a) shows the results for three UAVs in a normal condition. As this figure shows, the variation of trust score is negligible in normal conditions. Fig. (7b) shows the same results.
where UAV $u_1$ is under man-in-the-middle and forwarding attacks. Both of these attacks affect the task success score of the UAV. Lower task success score leads to a significant decrease in total trust score. In this scenario, the attacker stops attacking the UAV agent at time 97 Sec. The obvious variation in total trust score around the time 100 Sec, which the attacker stops attacking the agent, shows the power of the proposed algorithm.

Fig. (7c) shows the variation in total trust score for all three agents when an attacker performs a GPS spoofing attack on UAV $u_1$. This attack leads the agent to mis-navigate and hence to deviate from its predefined path. Again, the clear variation in total trust score in comparison with other UAVs and around the time 100 Sec, shows how accurately our model can detect the attacks on the UAVs. Fig. (7d) shows the same results for a bad weather situation where there is a strong wind. Since the bad weather is a situation that affects the UAVs altogether, we do not see any obvious trust score deviation by comparing the UAVs’ trust score together. It means that all UAVs are experiencing the same situation. It is worth mentioning that we show the variation of the total trust score in comparison with the last score, not the trust score itself. Hence, Figs. (7a and 7d) are quite similar. Fig. (7e) aims at representing the ability of our model to differentiate between the UAVs’ abnormal behavior when they are under an attack or when they are operating under harsh environmental conditions. Hereby, this figure shows the total trust score variation in time for a network under bad weather conditions, where UAV $u_1$ is under GPS spoofing attack. Again, the trust management mechanism clearly detects the attack, which is until the time around 100 Sec and recognizes it from the miss-behavior, which is due to the bad environmental conditions.

VI. CONCLUSION

We proposed UASTrustChain, a general decentralized real-time trust monitoring framework to score the trustworthiness of UAVs and detect the malicious ones. In this method, we exclude the central decision maker and replaced it by several DOs, to remove the single point of failure. The distributed DOs observe the UAVs’ behavior in terms of the flight trajectory and successfully performed assigned tasks. The framework differentiates between the abnormal behaviors due to the cyber-physical attacks and those due to harsh environmental conditions. It further considers a trustworthiness score for each DO, to prevent the potentially compromised DO from making a significant variation in the decision-making process. The trust scores of the UAVs are tamper-proof by the Blockchain to guarantee the integrity and the transparency of them. The proposed trust monitoring framework estimates the performance of the UAVs based on the observation of the distributed observers, rather than relying on the self-report of the UAVs. The uncertainty of the DOs observation is also taken into account. The performance of UASTrustChain is evaluated with extensive simulations which shows the impact of different attacks on the agents’ trust scores. The results further show the precision of the proposed algorithm in detecting the compromised DO. As the future direction, the real implementation of the proposed trust management framework is suggested.

REFERENCES

[1] A. Rahmati, X. He, I. Guvenc, and H. Dai, “Dynamic mobility-aware interference avoidance for aerial base stations in cognitive radio networks,” in IEEE INFOCOM 2019-IEEE Conference on Computer Communications. IEEE, 2019, pp. 595–603.
[2] J. C. Hodgson, S. M. Baylis, R. Mott, A. Herrod, and R. H. Clarke, “Precision wildlife monitoring using unmanned aerial vehicles,” Scientific reports, vol. 6, p. 22574, 2016.
[3] A. Eldosouky, W. Saad, and N. Mandayam, “Resilient critical
FIGURE 5: UAVs’ real trajectory against pre-defined path: (a) normal condition; (b) u₁ is under GPS spoofing attack; (c) bad weather condition and u₁ is under GPS spoofing attack.
FIGURE 6: The impact of deviation form the pre-defined path on the total trust score: (a) an average values for different UAV agents; (b) a relative comparison for three agents in bad weather condition where $u_1$ is under GPS spoofing attack.

IEEE Pervasive Computing, vol. 16, no. 1, pp. 24–32, 2017.

[10] U. Challita, A. Ferdowsi, M. Chen, and W. Saad, “Machine learning for wireless connectivity and security of cellular-connected uavs,” IEEE Wireless Communications, vol. 26, no. 1, pp. 28–35, 2019.

[11] A. Eldosouky, A. Ferdowsi, and W. Saad, “Drones in distress: A game-theoretic countermeasure for protecting uavs against gps spoofing,” arXiv preprint arXiv:1904.11568, 2019.

[12] R. Altawy and A. M. Youssef, “Security, privacy, and safety aspects of civilian drones: A survey,” ACM Transactions on Cyber-Physical Systems, vol. 1, no. 2, pp. 1–25, 2016.

[13] H. Yu, Z. Shen, C. Leung, C. Miao, and V. R. Lesser, “A survey of multi-agent trust management systems,” IEEE Access, vol. 1, pp. 35–50, 2013.

[14] F. Afghah, M. Zaeiri-Amirani, A. Razi, J. Chakareski, and E. Bentley, “A coalition formation approach to coordinated task allocation in heterogeneous uav networks,” in 2018 Annual American Control Conference (ACC), June 2018, pp. 5968–5975.

[15] A. Afghah, A. Shamsoshoara, L. Njilla, and C. Kambou, “A reputation-based stackelberg game model to enhance secrecy rate in spectrum leasing to selfish iot devices,” in IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2018, pp. 312–317.

[16] J. McNeely, M. Hatfield, A. Hasan, and N. Jahan, “Detection of uav hijacking and malfunctions via variations in flight data statistics,” in 2016 IEEE International Carnahan Conference on Security Technology (ICCST). IEEE, 2016, pp. 1–8.

[17] M. Keshavarz, A. Shamsoshoara, F. Afghah, and J. Ashdown, “A real-time framework for trust monitoring in a network of unmanned aerial vehicles,” in International Conference on Computer Communications, INFOCOM’20. IEEE, 2020, p. to appear.

[18] J. S. Warner and R. G. Johnston, “Gps spoofing countermeasures,” Homeland Security Journal, vol. 25, no. 2, pp. 19–27, 2003.

[19] N. M. Roddaiy, R. d. O. Schmidt, and A. Pras, “Exploring security vulnerabilities of unmanned aerial vehicles,” in NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium. IEEE, 2016, pp. 993–994.

[20] W. Fan and F. Yang, “Centralized trust-based in-band control for adn control channel,” IEEE Access, vol. 8, pp. 4289–4300, 2020.

[21] Z. Movahedi, Z. Hosseini, F. Bayan, and G. Pujolle, “Trust-distortion resistant trust management frameworks on mobile ad hoc networks: A survey,” IEEE Communications Surveys Tutorials, vol. 18, no. 2, pp. 1287–1309, 2016.

[22] Z. Wei, H. Tang, F. R. Yu, M. Wang, and F. Mason, “Security enhancements for mobile ad hoc networks with trust management using uncertain reasoning,” IEEE Transactions on Vehicular Technology, vol. 63, no. 9, pp. 4647–4658, Nov 2014.

[23] R. J. Cai, X. J. Li, and P. H. J. Chong, “An evolutionary self-cooperative trust scheme against routing disruptions in manets,” IEEE Transactions on Mobile Computing, vol. 18, no. 1, pp. 42–55, 2019.

[24] A. M. Shabut, K. P. Dahal, S. K. Bista, and I. U. Awan, “Recommendation based trust model with an effective defence scheme for manets,” IEEE Transactions on Mobile Computing, vol. 14, no. 10, pp. 2101–2115, 2015.

[25] X. Huang, R. Yu, J. Kang, and Y. Zhang, “Distributed reputation management for secure and efficient vehicular edge computing and networks,” IEEE Access, vol. 5, pp. 25408–25420, 2017.

[26] Y. Xiao and Y. Liu, “Bayestrust and vehiclerank: Constructing an implicit web of trust in vanet,” IEEE Transactions on Vehicular Technology, vol. 68, no. 3, pp. 2850–2864, 2019.

[27] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, “Blockchain-based decentralized trust management in vehicular networks,” IEEE Internet of Things Journal, vol. 6, no. 2, pp. 1545–1560, 2019.

[28] Z. Lu, Q. Wang, G. Qu, and Z. Liu, “Bars: A blockchain-based anonymous reputation system for trust management in vanetns,” in 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE
FIGURE 7: UAVs’ behavior under different conditions: (a) normal condition; (b) man-in-the-middle and forwarding attacks; (c) GPS spoofing attack; (d) bad weather condition (e) both bad weather and GPS spoofing.
[32] S. Haber and W. S. Stornetta, “How to time-stamp a digital document,” Journal of Cryptology, vol. 3, pp. 99–111, 1991.
[33] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” White paper, 2009.
[34] D. B. Johnson and D. A. Maltz, “Dynamic source routing in ad hoc wireless networks,” in Mobile Computing. Kluwer Academic Publishers, 1996, pp. 153–181.

MAHSA KESHAVARZ received her M.S. degree in Computer Science from the North Carolina Agricultural and Technical State University, Greensboro, NC, USA, in 2018. She is currently a PhD student in the School of Informatics, Computing and Cyber Systems at Northern Arizona University (NAU), Flagstaff, AZ, USA. Her research interests include cybersecurity, Internet of Things (IoT), multi-agent systems, machine learning, smart homes, and privacy violation.

MOHAMMED GHARIB is currently a post-doctoral research fellow at the School of Informatics, Computing, and Cyber Systems, Northern Arizona University, Flagstaff, AZ, USA. He was formerly a postdoctoral research fellow at the Department of Computer Science, Johns Hopkins University, Baltimore, MD, US, and School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran. He was also a senior lecturer at the Department of Computer Engineering, Sharif University of Technology, Tehran, Iran, between 2010 to 2018. Mohammed received a B.S. degree from the Department of Computer Engineering, Baghdad University of Technology, Baghdad, Iraq, in 2007, and both M.S. and Ph.D. degrees from the Department of Computer Engineering, Sharif University of Technology, Tehran, Iran in 2009 and 2015, respectively. During the year 2014, Mohammed had six months visiting research program at University of California Irvine. His main research interest is the performance evaluation of computer networks including wireless networks, cloud networks, data center networking, and their security aspects.

FATEMEH AFGHAH is an Associate Professor with the School of Informatics, Computing and Cyber Systems, Northern Arizona University (NAU), Flagstaff, AZ, USA, where she is the Director of Wireless Networking and Information Processing (WINIP) Laboratory. Prior to joining NAU, she was an Assistant Professor with the Electrical and Computer Engineering Department, North Carolina A&T State University, Greensboro, NC, USA, from 2013 to 2015. She is the recipient of several awards including the National Science Foundation (NSF) CAREER award in 2020, the Air Force Office of Scientific Research Young Investigator Award in 2019, and the NSF CRII Award in 2017. Her research interests include wireless communication networks, decision making in multi-agent systems, radio spectrum management, hardware-based security, and artificial intelligence in healthcare.

JONATHAN ASHDOWN was born in Niskayuna, NY, USA, in 1984. He received the B.S., M.S., and Ph.D. degrees from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 2006, 2008, and 2012, respectively, all in electrical engineering. His Ph.D. dissertation was on a high-rate ultrasonic through-wall communication system using MIMO-OFDM in conjunction with interference mitigation techniques. In 2012, he was a recipient of the Best Unclassified Paper Award at the IEEE Military Communications Conference. From 2012 to 2015, he worked as a Civilian Research Scientist with the Department of Defense (DoD), SPAWAR Systems Center Atlantic, Charleston, SC, USA, where he was involved in several basic and applied research projects for the U.S. Navy, mainly in the area of software defined radio. In 2015, he transferred within DoD. He is currently an Electronics Engineer with the Air Force Research Laboratory, Rome, NY, USA, where he is involved in the research and development of advanced emerging technologies for the U.S. Air Force.