Trust Quantification in a Collaborative Drone System with Intelligence-driven Edge Routing

Alicia Esquivel Morel*, Ekincan Ufuktepe*, Cameron Grant †, Samuel Elfrink‡,
Chengyi Qu*, Prasad Calyam* and Kannappan Palaniappan*
*Department of Electrical Engineering and Computer Science, University of Missouri - Columbia
†Department of Computer Science, University of Georgia - Athens
‡Department of Computer Science, Southeast Missouri State University - Cape Girardeau
Email: *{ace6qv, cqy78}@mail.missouri.edu, †cameron.grant@uga.edu, ‡selfrink3s@semo.edu, *{euh46, calyamp, pal}@missouri.edu

***Abstract*—Collaborative Drone systems (CDS) have the potential to benefit a variety of application areas such as agriculture, military operations, surveillance, and disaster response. At the same time, CDS can pose challenges due to their limited flight time impacted by battery capacities, and constrained edge computation capabilities on-board the drones. Furthermore, an understudied subject relates to when drones in a CDS trust each other to accomplish a task, resulting in new vulnerabilities that can be exploited via cyber attacks. In this paper, we propose a novel trust quantification methodology in a CDS with intelligence-driven edge routing, which can help detect malicious nodes in a CDS that compromise communication and disrupt the functionality of packet forwarding. Our approach for trust quantification is guided by a CDS vulnerability analysis that characterizes impact due to the presence of two malicious threat agents viz., flooder node and faker node. Detection of these threat agents in a CDS is aided by trust quantification in the form of trust scores obtained by using a Bayesian Network model that allows for decision-making on CDS nodes' trust levels. We validate our trust quantification methodology in ns-3 based simulation experiments and show how we can categorize nodes based on different thresholds of trust scores with varying sensitivities, which helps in the detection of CDS threat agents.**

***Index Terms*—Edge Routing Trust, Trust Quantification, Secure Network Protocol, Collaborative Drone Systems**

I. Introduction

Drone swarms or Collaborative Drones Systems (CDS) are being increasingly utilized for critical operations in applications such as precision agriculture, aerial imaging, disaster scenarios, and rescue response [1], [2]. With the continuous innovation in communication technologies, software, and hardware, these systems have grown in development, and as a consequence, they are also subject to security threats on confidentiality, integrity, authenticity, and availability [3].

Figure 1 shows a use-case scenario, in which a CDS is accomplishing a mission on a disaster scene. The collaboration in the drone swarm could involve information transmission over long distances as part of delivering application services

This material is based upon work supported by the National Science Foundation (NSF) under Award Numbers CNS-1950873 and CNS-1647182, Army Research Lab (ARL) under Award Number W911NF1820285, and the National Security Agency (NSA) under Award Numbers H98230-21-1-0260 and H98230-20-1-0297. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the NSF, ARL or NSA.

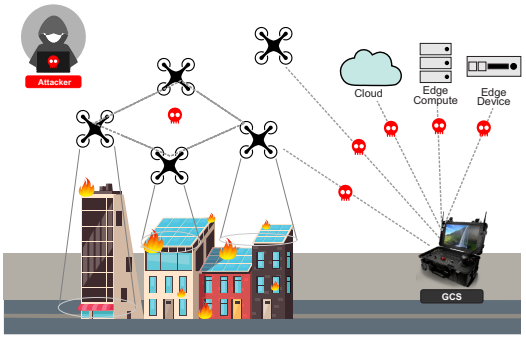

Figure 1. CDS and its integration with edge/cloud infrastructures accomplishing missions, which engenders cyber attacks that target drone's flight, communication, or reliability in data processing tasks.

in real-time, and providing situational awareness to critical decision-making operations. Correspondingly, a large-scale CDS should have secure and reliable coordination amongst all the components [4]. The collaboration of drones through a ground control station (GCS) in a Flying Ad-hoc NETwork (FANET), and their integration with edge/cloud infrastructures can engender cyber attacks on any component of the system. The attacks can jeopardize not only the flight (e.g., location and trajectory) and communication (e.g., packet transmission rates), but also the reliability (e.g., data accessibility) in data processing tasks. In addition, an issue that is understudied in prior works relates to securing drones in a CDS, where the drones trust each other to accomplish a task. Particularly, in a CDS where nodes have roles as hosts for data processing as well as routers for packet forwarding, it is common to have topologies that are stateless and lack a centralized administration [5].

Consequently, drone nodes might join and leave the CDS dynamically due to issues e.g., battery constraints, or communication range. Moreover, it is possible to have heterogeneous drones in a CDS that have different capabilities and are suitable for specific responsibilities for communication and data processing [6]. In all these cases, it is hard to maintain clear-cut network perimeters to enforce security in the protocols used to manage the CDS [7]. In this paper, we aim to address the above issues in securing a CDS

and propose a novel trust quantification methodology for a CDS where drones communicate and cooperate in a collaborative setting to satisfy application mission requirements (e.g., disaster response, ensuring border security). Detection of malicious nodes in a CDS is possible, through low trust scores, guaranteeing only trustworthy nodes in the FANET are giving privileges. Our strategy for trust quantification is guided by a CDS vulnerability analysis that characterizes impact due to the presence of two types of malicious threat agents viz., a flooder node and a faker node. The flooder and faker threat agents create attack vectors that target the dimensions of data throughput and energy consumption of CDS resources, respectively. Detection of malicious behavior of threat agents by trust quantification for calculation of trust scores is obtained by using a Bayesian Network model. In our evaluation experiments, we demonstrate the impact of these threat agents. Further, we validate our trust quantification methodology and show how we can categorize nodes based on different thresholds of trust scores with varying sensitivities, which helps in detection of CDS threat agents.

A summary of this paper's contributions is listed in the following:

- We propose an extendable model to quantify trust in the form of trust scores using a Bayesian Network, for intelligence-drive edge routing protocols in a CDS.
- We also demonstrate the impact of two prominent attack vectors in a CDS i.e., faker (relays misinformation of system state in terms of e.g., energy consumption), and flooder (overwhelms system resources and limits data throughput/system availability).
- Lastly, we show how the calculated trust scores can be used along with thresholds (i.e., security policies) to detect threat agents that impact CDS operations.

The remainder of this paper is organized as follows: in Section II, we discuss the related work. Section III details the vulnerability analysis of a CDS. In Section IV, we present our solution approach for trust quantification and detection of threat agents. Section V presents the performance evaluation of our solution. Lastly, Section VI concludes the paper.

II. Related Work

Trust plays an important role in providing security to edge networking systems, enabling the CDS to select trustworthy nodes for critical communication and data processing tasks [8]. The trust paradigm has been evolving, including in the area of strategies for zero trust in enterprises, and cloud computing environments with initiatives such as e.g., Google BeyondCorp [9], and the Gartner Continuous Adaptive Risk and Trust Assessment (CARTA) [10]. These strategies adopt security models that shift access controls from the perimeter to individual devices and users [11]. In addition, models in [12] aim to collect trustworthy data based on edge computing using the behavior of smart devices. However, prior works mainly focus on qualitative strategies, and there is a dearth of trust quantification methods for edge networking systems [13]. To the best of our knowledge, our work is the first to focus on trust quantification that can be used to secure a CDS with intelligence-driven edge routing. We remark that the quantification methods to determine trust are important and necessary for designing security policies that enable detection and mitigation of disruptive impacts due to threat agents.

For the purposes of this work, we use an exemplar intelligence-driven edge routing protocol viz., SPIDER [14], which is a state-of-the-art method that uses machine learning techniques on satellite images to provide physical obstacle awareness in geographic routing, while also maintaining energy awareness that boosts baseline performance. SPIDER outperforms popular routing protocols such as Geographical and Energy Aware Routing (GEAR), and Ad-Hoc On Demand Distance Vector (AODV) in terms of application throughput, and has comparable or improved energy-efficiency. By identifying security vulnerabilities in such an intelligence-driven edge routing scheme, we show how trust quantification can be used to generate trust score thresholds using statistical methods applied in works such as [15]. Our work is relevant to monitoring and controlling of security policies in any CDS that uses edge routing algorithms that are energy-aware and throughput-aware to satisfy application mission requirements in e.g., disaster response, smart farming, or border security.

III. CDS Vulnerability Analysis

In this section, we first present an overview of the intelligence driven edge routing in CDS. Following this, we present a threat model for the CDS edge routing.

A. Intelligence-driven Edge Routing in CDS

SPIDER is a energy-aware protocol that enables high data throughput routes in a CDS that can be used for critical use cases. The data flow in SPIDER can be observable when drone swarm nodes communicate with each other and the GCS in order to obtain system intelligence for performing efficient operations. Each drone swarm node uses a beacon to know information about the neighbor nodes (e.g., energy levels, data throughput levels). In addition, SPIDER packets are used to intelligently determine the routes for high throughput data transmission across the CDS.

B. Threat Model for CDS Edge Routing

We assume the threat agents compromise the drone and GCS devices in a CDS by using potential attack vectors outlined in [16]–[18]. Using the Microsoft STRIDE framework [19], we thoroughly model the threats in a CDS with intelligence-drive edge routing addressing the key assets, typical victims, and attack patterns. The threat agents we study target the CDS resources using attack vectors impacting the routing mechanism (i.e., in SPIDER or any other comparable method) along the dimensions of throughput and energy consumption data. We characterize these threat agents as causing "Flooder Node Presence" and "Faker Node Presence" attacks, respectively.

Flooder Node Presence Attack: Considering the Confidentiality, Integrity, and Availability (CIA) triad for vulnerabilities, the Flooder node primarily launches availability attacks

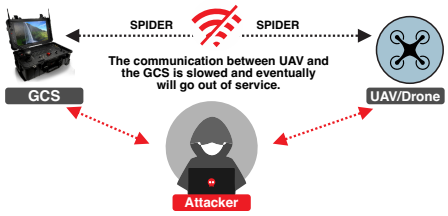

Figure 2. Flooder Node Presence attack overwhelms CDS network resources with illegitimate traffic. As a result, the network throughput decreases, and the drones/GCS communication is disrupted or taken out of service.

(as shown in Figure 2) on a CDS. The threat agent involves a compromised drone software component (e.g., a data export/processing module) or a hardware component (e.g., a sensor) that overwhelms network resources with illegitimate traffic e.g., sending UDP packets to other drones or the GCS. The target of the flood is unable to handle the incoming packets, and is made non-functional that impacts both communication and data processing tasks involving the drones and the GCS.

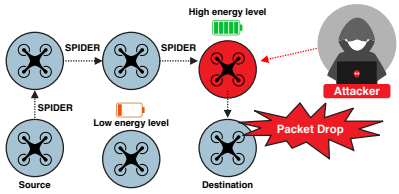

Figure 3. Faker Node Presence Attack introduces a malicious drone device in the CDS. The malicious device beacons other drones in the CDS, showing high levels of energy to get chosen as the preferred forwarding node, and consequently copies/alters/drop packets.

Faker Node Presence Attack: In contrast - while considering the CIA triad for vulnerabilities, the Faker node can cause a breach of confidentiality and integrity in a CDS (as shown in Figure 3) by using malicious beacons in drone information exchanges. For instance, the compromised Faker node can indicate high levels of energy to favor itself to be chosen for packet forwarding over other legitimate drones that are better suited. When packets arrive at the Faker node, the packets are maliciously copied, altered, or dropped to impact the routing protocol functioning as well as the overall CDS application mission.

IV. CDS Trust Quantification Methodology

In this section, we present our solution approach for trust quantification by first presenting how we formalize trust. Following this, we detail the algorithms to calculate trust scores as shown in Figure 5 using a Bayesian Network (BN) model and probabilistic value calculation. Lastly, we present a running example for showing the trust quantification steps.

A. Trust Formalization

To quantify trust in the CDS network, we build and use a BN model for decision-making on the nodes that could be trusted. BNs are directed acyclic graphs where nodes are encoded with

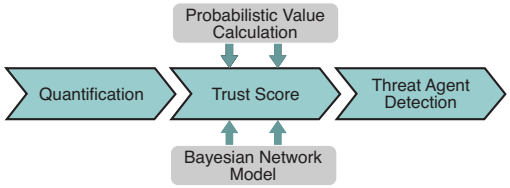

Figure 4. Trust Quantification that produces Trust Scores using Bayesian Network model and a probabilistic value calculation, which is used for threat agent detection i.e., detection of Flooder node or Faker node presence.

probabilistic information, and the edges represent the causality relationships between nodes. BN are also found useful in providing useful information under uncertainty and vague circumstances [20], [21]. We build our BN model based on the two major vulnerabilities in the CDS in terms of resource dimensions of energy consumption and data throughput, and use it to decide if a node in a CDS should be trusted or not. This type of approach is commonly used in the context of security to assess and measure the risks of a system and network [22]–[24]. We assume there is a *Trust Evaluation* node in a CDS (potentially a secured/hardened GCS), which will provide a probabilistic inference of a "Trust Score" based on the normal or attack activity detection on the network, as it is depicted in Figure 4.

*1) **Probabilistic Value Calculation for Threat Agent Presence**:* To quantify and calculate the probabilistic values of the Flooder Node Presence and the Faker Node Presence attacks that are being performed, we use the Soft-max function (4) on the number of sent packets, and the remaining energy levels separately for each node in the network, respectively. We calculate the mean of the number of sent packets of each node and the remaining energy levels using the Equation (1), where x_i could be the number of sent packets or the remaining energy level of the i^{th} node, n is the number of nodes in the network, and X is the mean of the number of sent packets or the remaining energy levels.

$$X = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{x_1 + x_2 + \cdots + x_n}{n} \tag{1}$$

Subsequently, we separately calculate the standard deviation as shown in Equation (2) based on the number of sent packets and the remaining energy levels in the network. The standard deviation will help us calculate the upper and lower boundaries of the number of sent packets and energy levels that should occur in the network, which is a common approach used in detecting any anomaly in a network path [15]. For the Flooder Node Presence attack, if there is an unusual frequency of packet transmission, the upper boundary will help determine any abnormality. Similarly, since the Faker Node Presence attack is based on imitated high energy levels in the network, we are only interested in the upper boundaries in the network. After the standard deviation and the mean values of the number of sent packets and the remaining energy levels are calculated, we calculate the upper boundary as shown in Equation (3).

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2} \tag{2}$$

$$U = \sigma + X \tag{3}$$

Once the upper boundary is calculated, we use a Euclidean distance calculation between the upper boundary of the number of sent packets and the number of packets that each node sent in the network. We also follow the same approach for the upper boundary of the energy level and the remaining energy levels of the nodes in the network. The Euclidean distance calculation is given as follows:

$$f(x, U) = \begin{cases} EuclideanDistance(x, U), \text{if } x > U \\ 0, \text{otherwise} \end{cases}$$

The higher the calculated distance is, the likelihood of the node performing an attack increases. However, the calculated distances are still not in probabilistic format. Therefore, we use the Soft-max (4) function to transform the calculated distances into probabilistic information, so that it could be used in calculating the Trust Score calculation. Figure 5 shows how the Bayesian Network model is leveraged for the Trust Score calculation; Faker and Flooder nodes presence attacks will influence the nodes' Trust Score values.

Soft-max is a function that is widely used in machine learning, for scaling data to probabilistic values between 0-1. The d_i is the distance calculated for the i^{th} node.

$$Softmax(d_i) = \frac{\exp(d_i)}{\sum_j \exp(d_j)} \tag{4}$$

It is important to highlight that an attack detection speed depends on the frequency of trust score calculations, and thus higher frequency can proactively detect attacks as they begin, versus lower frequency may lead to detection after the impact of an attack has occurred in the CDS.

*2) **Trust Score Calculation**:* To calculate the trust score, we use the joint probability distribution for our BN given in Equation (5), where x_1, x_2, x_3 are respectively the *Trust Score*, *Flooder Node Presence Attack*, *Faker Node Presence Attack* nodes. The X_i is the set of nodes related to x_i, therefore, $parents(X_i)$ is the set of given conditional probabilities with respect to x_i. For our case, we only have one node which is the parent node, which is the *Trust Evaluation* node.

$$P(x_1, x_2, x_3) = \prod_{i=1}^{3} P(x_i | parents(X_i)) \tag{5}$$

For each node in the BN we define two states: *True (T)* and *False (F)*. The *True* state for the two attack nodes represents the probability of the attack being present by a particular node in the network. On the other hand, the *True* state for the *Trust Score* node represents the probability of the node that should be trusted. The *False* state for every node represents the complement of their *True* state. In Equation (6), we use marginalization for probabilistic inference to calculate the probabilistic trust score to decide if the node in the network should be trusted. The j, and k are the set of two states (*True*, *False*) defined for each node.

$$P(x_1) = \sum_{j}\sum_{k}^{j,k \in \{T,F\}} P(x_1, x_2 = k, x_3 = j) \tag{6}$$

In Equation (7), we simplify the marginalization and calculate the probability of a node being trusted or not.

$$P(x_1) = \sum_{j}\sum_{k} P(x_1 | x_2 = k, x_3 = j) P(x_2 = k) P(x_3 = j) \tag{7}$$

Using our BN model for trust evaluation and quantification allows flexibility and versatility in terms of adopting additional attacks, which makes our trust score quantification extendable and generalizable. For instance, if any of the attack scenarios are eliminated, or if a new attack scenario needs to be added, this could be easily done by modifying the BN. Nevertheless, the confidence threshold for trusting or not trusting the node can also be modified as well, which might vary based on the attack types or the CDS environment. We remark that - our use of a Bayesian trust quantification in a CDS aims to delineate the uncertainty through probability. In addition, we leverage Bayesian inference calculated through inference models from past interactions between the nodes. We specifically utilize information (e.g., geographical position and energy levels) of the neighboring drones as well as their behavior patterns. This information is subsequently forwarded to other nodes or components of the CDS, which are involved in the computation of the trust score.

B. Trust Quantification Algorithm

In the case of a UDP Flooder Node Presence attack launched with the purpose of bringing down the CDS network, we propose a Detection and Probabilistic Quantification **Algorithm 1**. This algorithm's main function is to check for a maliciously high data rate of packets from drones in the network. We use standard deviation and mean to detect any anomaly that is occurring with the packet transmission frequency and quantify the probability of the drone performing a flooder attack. The addition of the mean and standard deviation will determine the upper boundary for typical or expected packet-sending frequency. Any node sending packets in a frequency above the upper boundary will be considered suspicious, and the higher distance to the upper boundary will be the indicator of the high likelihood of a Flooder attack. The distances will be processed with a Soft-max function to calculate their probabilistic values, and thus will be used in the BN to calculate the drone's trust score. A similar approach will be used for the Faker Node Presence attack, but energy level information is used instead of the packet transmission frequency information.

Ideally, none of the nodes should be trusted in the network, because trust can be considered another form of vulnerability and weakness. Unlike the frequentist approaches, BNs are capable of making inferences under uncertainty, which makes them a perfect candidate to use in our CDS context. Even though a node has never shown any malicious activity in its history, our BN model will still assign a small probability

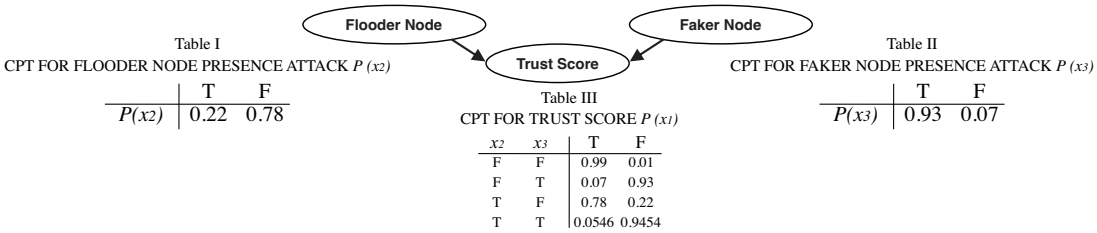

Table I
CPT FOR FLOODER NODE PRESENCE ATTACK $P(x_2)$

	T	F
$P(x_2)$	0.22	0.78

Table II
CPT FOR FAKER NODE PRESENCE ATTACK $P(x_3)$

	T	F
$P(x_3)$	0.93	0.07

Table III
CPT FOR TRUST SCORE $P(x_1)$

x_2	x_3	T	F
F	F	0.99	0.01
F	T	0.07	0.93
T	F	0.78	0.22
T	T	0.0546	0.9454

Figure 5. Trust Scores obtained using a Bayesian Network model and a probabilistic value calculation. Tables I and II represents the CPT for Flooder and Faker node presence attacks, whereas Table III the Trust Score respectively.

Algorithm 1: Detection and Probabilistic data collection of Flooder Node Presence Attack for Trust Scoring

Input: F // set of packet numbers sent of nodes
Output: P, Set of probability scores for Flooder Node Presence attack
$P \leftarrow \varnothing$ // Init. probability scores of each node
let m be a the mean of sent packets
/* Calculate mean of sent packets */
$m \leftarrow meanOfSentPackets(F)$
let s be the standard deviation of sent packets
/* Calculate std. dev. for sent packets */
$s \leftarrow stdDevSentPackets(m, F)$
let u be the upper number of sent packets
$u \leftarrow s + m$ // Calculate the upper boundary
let D be the set of distances
for *node i, number of sent packets* f_i *in* F **do**
  if $f_i > u$ **then**
    $d_i \leftarrow f_i - u$
  end
  else
    $d_i \leftarrow 0$
  end
  $D \leftarrow D \cup d_i$
end
for *node i, distance* d_i *in* D **do**
  $p_i \leftarrow softMax(d_i, D)$
  $P \leftarrow P \cup p_i$
end

to the nodes, giving room for something that is not taken into account that can make this node malicious. This small probability can refer to other malicious attacks that are not included in our BN model yet. On the other hand, based on the application or network protocol, the sensitivity of classifying nodes as malicious or non-malicious depends on the security policies. For instance, a strict security policy can define malicious nodes based on probabilities that are above 0.1. Or a more lenient security policy can classify malicious nodes that are above 0.5.

C. Detailed Trust Quantification example

For detailing our trust quantification steps, we leverage an example based on Conditional Probability Tables (CPT) for the nodes. Specifically, we focus on a scenario that takes into consideration both the attack vectors viz., Flooder Node presence, and Faker Node presence. We compute values based on variables that depend on the dynamicity of the nodes. In Tables I, II, III, we provide CPT for the nodes *Flooder Node Presence Attack*, *Faker Node Presence Attack*, and *Trust Score* respectively for a running example. We assume that the probabilities in Tables I, and II are calculated using the trust quantification algorithms. Calculating the probabilities for Table III is not an easy process, where every probability of condition and scenario must be calculated independently with different combinations. For these types of scenarios, Noisy-OR [25] distribution is a commonly used technique. Noisy-OR is a generalized model of the logical OR gate, which is capable of capturing non-determinism in a system with disjunctive interaction. Consequently, the probabilities in Table III are calculated using Noisy-OR distribution from Tables I, and II. Once we extend Equation (7), we replace the probabilities with the values we have from the CPT given in Tables I, II, III. After replacing the probabilities, the trust score is calculated as $P(x_1 = T) = 0.128$. The probabilistic score indicates that with a probability of 0.128, the node should be trusted. In other words, with the probability of 0.872 ($P(x_1 = F) = 1 - P(x_1 = T)$), the suspicious node should not be trusted.

V. Performance Evaluation

In this section, we first demonstrate the impact of the threat agents (i.e., flooder and faker nodes) in a CDS to analyze the vulnerabilities. Following this, we validate our trust quantification methodology in ns-3 simulator.

A. Simulation Setup

Our setup is based on parameters with realistic simulation settings relevant in actual application scenarios, as detailed in [26]. The CDS simulation scripts are based on [14], and we use the exemplar state-of-the-art SPIDER protocol to setup the network configurations with 17 regular and 2 malicious nodes (19 nodes in total) that have realistic mobility patterns, and for collecting measurements of packet throughput and node energy consumption.

B. Attack Impact Analysis:

Trust scores can be critical to deploy trust management mechanisms that protect the CDS components against attack vectors.

Flooder Node Presence attack: In order to demonstrate the Flooder Node presence vulnerability, a malicious node was introduced into the network. The malicious node was used as a source node to flood the GCS sink node with abnormally large UDP packets of size 60000 bytes at a data rate of 1000 Mbps. Figure 6 shows the network throughput results under normal (attack-free) conditions, and in presence of a Flooder node attack. In the normal condition, the number of packets

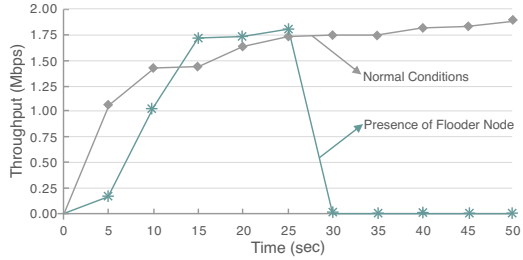

Figure 6. Impact on network Throughput in a CDS under normal (attack free) conditions, and in presence of a Flooder node attack.

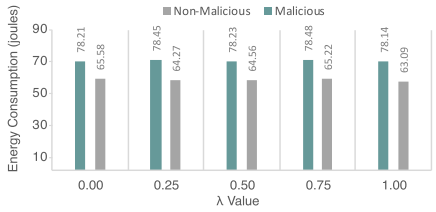

Figure 7. Impact on Energy consumption in a CDS under normal conditions (with non-malicious nodes behavior) and in presence of a Faker node attack, shown for different forwarding policies (λ values).

transmitted increases as expected over time. In contrast, the impact of one malicious node Flooder Node Presence on the network throughput is evident by the drastic drop in network throughput.

Faker Node Attack Impact: In order to demonstrate the Faker Node presence vulnerability, a malicious node was introduced into the network. The malicious node was used to misinform other drone nodes to choose itself versus other legitimate drones with higher energy levels to forward packets. Figure 6 shows the energy consumption results under normal (with only non-malicious node) conditions, and in presence of a Faker (with malicious node) attack. Comparing the energy consumption of non-malicious and malicious nodes at different forwarding policies (λ values) in SPIDER [27], we can see the high-energy malicious node consistently consumes more energy in the simulation. This indicates that the insecure SPIDER protocol gave preference to the Faker node by forwarding more packets through it than the other legitimate nodes.

C. Trust Quantification Analysis

In our experiment on 19 nodes, we performed both the Flooder Node and Faker Node Presence attacks. For each node, we calculate the probabilistic values for each attack, then we calculate the trust scores based on the attack probabilities. In Table I, we show the probabilities calculated for trust score calculation using the BN. In our case study, we configured $node_0$ to perform a Faker Node Presence attack and configured $node_7$ to perform a Flooder Node Presence attack. Setting of the security policy can impact how many false alarms can arise in the CDS. For instance, if we set a strict security policy in which we classify the trust score threshold to a probability of 0.98, we would be classifying $node_0$, $node_7$ and $node_{15}$

Table I
ATTACK PROBABILITY RESULTS TO DETECT FAKER AND FLOODER NODES BASED ON THEIR TRUST SCORE THRESHOLDS.

Node ID	Faker Node	Flooder Node	Trust Score
$node_0$	**0.6941**	0.0001	**0.5152**
$node_1$	0.017	0.0001	0.9899
$node_2$	0.017	0.0001	0.9899
$node_3$	0.017	0.0001	0.9899
$node_4$	0.017	0.0001	0.9899
$node_5$	0.017	0.0001	0.9899
$node_6$	0.017	0.0001	0.9899
$node_7$	0.017	**0.878**	**0.2278**
$node_8$	0.017	0.0001	0.9899
$node_9$	0.017	0.0022	0.9899
$node_{10}$	0.017	0.0001	0.9899
$node_{11}$	0.017	0.0001	0.9899
$node_{12}$	0.017	0.0001	0.9899
$node_{13}$	0.017	0.0001	0.9899
$node_{14}$	0.017	0.0001	0.9899
$node_{15}$	0.017	0.1188	0.9769
$node_{16}$	0.017	0.0001	0.9899
$node_{17}$	0.017	0.0001	0.9899
$node_{18}$	0.017	0.0001	0.9899

as malicious nodes. However, in our experiments, the only malicious nodes were $node_0$ and $node_7$. Unfortunately, due to the strict security policy, $node_{15}$ was classified as malicious (false positive), just because it has shown small tendencies of performing a flooder attack. In another scenario, where the security policy is more lenient and the trust score threshold is set to 0.5, we will only notice that only $node_7$ will be categorized as a malicious node, but not $node_0$ which was actually performing a faker attack. A lenient security policy has ended up with a false negative, which is undesirable for the application users. We have shown how security policies and the sensitivity of the trust score threshold can affect the decision on classifying nodes as malicious. The thresholds can be configured differently in other CDS applications, depending on how much of a strict and sensitive trust policy the network is configured.

VI. CONCLUSION

In this work, we demonstrated potential vulnerabilities in a CDS with intelligence-driven edge routing and showed how a trust quantification methodology can help to detect malicious nodes that compromise communication and disrupt the functionality of packet forward for data processing. The novelty of our approach is the methodology to calculate trust scores for each node in the network using a BN model that can be easily extended with additional attacks. Our experiments with threat agents involving Flooder and Faker node presence attacks showed that our trust score calculation approach was successful, and our approach is practical for CDS operators/users to manage security policies.

As a future work, a trust quantification calculation considering the behavior of nodes in computation tasks can be used to extend our current methodology focused on the behavior of nodes in edge routing can be considered. Additionally, our approach can be used in scalability studies involving different combinations of nodes and attack vectors in order to configure relevant CDS network settings.

REFERENCES

[1] M. Erdelj and E. Natalizio, "Uav-assisted disaster management: Applications and open issues," in *2016 international conference on computing, networking and communications (ICNC)*, pp. 1–5, IEEE, 2016.

[2] S. H. Alsamhi, O. Ma, M. S. Ansari, and F. A. Almalki, "Survey on collaborative smart drones and internet of things for improving smartness of smart cities," *Ieee Access*, vol. 7, pp. 128125–128152, 2019.

[3] S. Hussain, S. A. Chaudhry, O. A. Alomari, M. H. Alsharif, M. K. Khan, and N. Kumar, "Amassing the security: An ecc-based authentication scheme for internet of drones," *IEEE Systems Journal*, vol. 15, no. 3, pp. 4431–4438, 2021.

[4] S. Liao, J. Wu, J. Li, A. K. Bashir, and W. Yang, "Securing collaborative environment monitoring in smart cities using blockchain enabled software-defined internet of drones," *IEEE Internet of Things Magazine*, vol. 4, no. 1, pp. 12–18, 2021.

[5] S. Thapar and S. K. Sharma, "Direct trust-based detection algorithm for preventing jellyfish attack in manet," in *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 749–753, 2020.

[6] G. Eleftherakis, D. Pappas, T. Lagkas, K. Rousis, and O. Paunovski, "Architecting the iot paradigm: a middleware for autonomous distributed sensor networks," *International Journal of Distributed Sensor Networks*, vol. 11, no. 12, p. 139735, 2015.

[7] E. G. Mwangi, G. M. Muketha, and G. N. Kamau, "Trust-based security technique to curb cooperative black hole attacks in mobile ad hoc networks using otb-dsr protocol in ns-3," *Global Journal of Computer Science and Technology*, 2020.

[8] L. Wei, Y. Yang, J. Wu, C. Long, and B. Li, "Trust management for internet of things: A comprehensive study," *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7664–7679, 2022.

[9] R. Ward and B. Beyer, "Beyondcorp: A new approach to enterprise security," 2014.

[10] D. CeArley, B. Burke, S. Searle, and M. J. Walker, "Top 10 strategic technology trends for 2018," *The Top*, vol. 10, pp. 1–246, 2016.

[11] C. Buck, C. Olenberger, A. Schweizer, F. Völter, and T. Eymann, "Never trust, always verify: A multivocal literature review on current knowledge and research gaps of zero-trust," *Computers & Security*, vol. 110, p. 102436, 2021.

[12] T. Wang, L. Qiu, A. K. Sangaiah, A. Liu, M. Z. A. Bhuiyan, and Y. Ma, "Edge-computing-based trustworthy data collection model in the internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4218–4227, 2020.

[13] Y. Wang, "Trust quantification for networked cyber-physical systems," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2055–2070, 2018.

[14] H. Trinh, P. Calyam, D. Chemodanov, S. Yao, Q. Lei, F. Gao, and K. Palaniappan, "Energy-aware mobile edge computing and routing for low-latency visual data processing," *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2562–2577, 2018.

[15] P. Calyam, J. Pu, W. Mandrawa, and A. Krishnamurthy, "Ontimedetect: Dynamic network anomaly notification in perfsonar deployments," in *2010 IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pp. 328–337, IEEE, 2010.

[16] A. R. Eldosouky, A. Ferdowsi, and W. Saad, "Drones in distress: A game-theoretic countermeasure for protecting UAVs against gps spoofing," *IEEE Internet of Things Journal*, 2019.

[17] A. Abdallah, M. Z. Ali, J. Mišić, and V. B. Mišić, "Efficient security scheme for disaster surveillance UAV communication networks," *Inf.*, vol. 10, no. 2, p. 43, 2019.

[18] C. Guerber, N. Larrieu, and M. ROYER, "Software defined network based architecture to improve security in a swarm of drones," in *IEEE Int. Conf. on Unmanned Aircraft Syst.*, pp. 51–60, 2019.

[19] A. Shostack, *Threat Modeling: Designing for Security*. Wiley Publishing, 1st ed., 2014.

[20] J. Pearl, "Causal inference in statistics: An overview," *Statistics surveys*, vol. 3, pp. 96–146, 2009.

[21] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[22] S. Kondakci, "A causal model for information security risk assessment," in *2010 Sixth International Conference on Information Assurance and Security*, pp. 143–148, IEEE, 2010.

[23] S. Kondakci, "Network security risk assessment using bayesian belief networks," in *2010 IEEE Second International Conference on Social Computing*, pp. 952–960, IEEE, 2010.

[24] E. Ufuktepe and T. Tuglular, "Estimating software robustness in relation to input validation vulnerabilities using bayesian networks," *Software Quality Journal*, vol. 26, no. 2, pp. 455–489, 2018.

[25] J. Pearl, *Probabilistic reasoning in intelligent systems*, vol. 88. Elsevier, 2014.

[26] A. Nguyen, H. Nguyen, V. Tran, H. X. Pham, and J. Pestana, "A visual real-time fire detection using single shot multibox detector for uav-based fire surveillance," in *2020 IEEE Eighth International Conference on Communications and Electronics (ICCE)*, pp. 338–343, IEEE, 2021.

[27] H. Trinh, P. Calyam, D. Chemodanov, S. Yao, Q. Lei, F. Gao, and K. Palaniappan, "Energy-aware mobile edge computing and routing for low-latency visual data processing," *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2562–2577, 2018.