Secure and Efficient UAV Tracking in Space-Air-Ground Integrated Network

Jiachun Li, *Student Member, IEEE*, Weijiong Zhang, Yan Meng, *Member, IEEE*, Shaofeng Li, Lichuan Ma, *Member, IEEE*, Zhen Liu, *Member, IEEE*, and Haojin Zhu, *Fellow, IEEE*

***Abstract*—With the development of 5G and other communication techniques, the space-air-ground integrated network (SAGIN) is regarded as a promising solution to provide wide-range, cost-effective, and real-time wireless access. Unmanned Aerial Vehicle (UAV) Tracking plays a crucial role in guaranteeing the performance of SAGIN. However, UAV tracking still faces a series of challenges in the real-world deployment, especially in the case of the presence of malicious attackers, who may launch a series of attacks (e.g., message spoofing attack, routing misbehavior attack) to disrupt the systems. In this study, to enhance the security and efficiency of SAGIN when conducting object tracking, we propose a novel and secure object tracking system named SecTracker to overcome the emerging security problems in the SAGIN. Focusing on the unmanned aerial vehicle (UAV) tracking, we design the Local Voting based Detection Module to defend the message spoofing attack and implement the Routing Evidence based Detection Module to defend the routing misbehavior attack. Besides, efficiency enhancement mechanisms (i.e., probabilistic detection algorithm and game theory based algorithm) are proposed in this paper to improve the efficiency of SecTracker. Considering the above-mentioned attacks, the overall tracking accuracy is improved from 88.16% in existing schemes to 96.64% in SecTracker, which demonstrates the effectiveness of the proposed system.**

***Index Terms*—Space-air-ground integrated network, message spoofing, misbehavior attack, UAV tracking.**

Manuscript received 29 May 2022; revised 1 December 2022 and 8 February 2023; accepted 27 February 2023. Date of publication 8 March 2023; date of current version 15 August 2023. This work was supported in part by the National Key R&D Program of China under Grant 2022YFB3103500, in part by the National Natural Science Foundation of China under Grants 62132013, 61972453, and 62072305, in part by the Shanghai Aerospace Science and Technology Innovation Foundation under Grant SAST2019-008, in part by the Key Research and Development Programs of Shaanxi under Grant 2021ZDLGY06-03, and in part by the China Postdoctoral Science Foundation under Grant 2022M721736. The review of this article was coordinated by Dr. Antonella Molinaro. *(Corresponding author: Haojin Zhu.)*

Jiachun Li, Yan Meng, Zhen Liu, and Haojin Zhu are with the Department of Computer Science and Engineering, School of Electronics, Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: jiachunli@sjtu.edu.cn; yan_meng@sjtu.edu.cn; liuzhen@sjtu.edu.cn; zhu-hj@cs.sjtu.edu.cn).

Weijiong Zhang is with the Shanghai Electro-Mechanical Engineering Institute, Shanghai 201109, China (e-mail: billcat2021@sina.com).

Shaofeng Li is with the Frontier Research Center, Peng Cheng Laboratory, Shenzhen, Guangdong 518055, China (e-mail: lishf@pcl.ac.cn).

Lichuan Ma is with the State Key Laboratory of Integrated Services Networks and Shaanxi Key Laboratory of Blockchain and Secure Computing, Xidian University, Xi'an, Shaanxi 710071, China (e-mail: lcma@xidian.edu.cn).

Digital Object Identifier 10.1109/TVT.2023.3253894

I. Introduction

With the wide deployment of 5G and other communication techniques, the space-air-ground integrated network (SAGIN) is expected to play an increasingly important role due to its unique advantages including wide-ranging, high throughput, and ease of deployment [1]. Compared with the traditional method which is based on terrestrial communication, the implementation of SAGIN can ensure the quality of services (QoS) when facing extreme environments (e.g., earthquakes, and fire disasters), which is expected to outperform the convention networking architectures by allowing the QoS guaranteed communications and facilitating human rescue and relief [2].

To enable QoS guaranteed UAV based SAGIN, one of the critical challenges is how to achieve the accurate and secure tracking of unmanned aerial vehicles (UAVs). The existing works (e.g., the CCOT, the BACF [3], [4]) mainly focused on improving the accuracy of object tracking, while paying less attention to the security issues of SAGIN. As a result, the performance of the existing routing and tracking schemes will drop sharply under the attacks. Therefore, it is highly desirable to introduce an effective security scheme to overcome the existing threats (e.g., the message spoofing attack, the routing misbehavior attack) in SAGIN.

To achieve secure, efficient, and accurate UAV tracking, we need to address the following research challenges. i) **Malicious behavior issues:** The malicious attacks performed by the tracker nodes can be categorized into the following two kinds, the *message spoofing attack*, and the *routing misbehavior attack*. More specifically, the malicious terrestrial tracker nodes (e.g., the vehicle sensor nodes) may report fake tracking results (i.e., UAVs location information) or perform malicious behaviors (e.g., the message flooding attack, the black hole attacks), which cause the UAV tracking system crash [5]. Therefore, how to ensure data authenticity and routing security remains the first research challenge. ii) **Efficiency issue:** There are numerous sensors deployed in SAGIN to achieve the different functionality (e.g., UAV tracking, GPS locating), which are communicating with each other over the wireless communications. Due to the limitations of the energy capability of UAV tracking nodes, it is highly desirable to take the energy issues into consideration when the security protocols are designed. iii) **Security Aware Tracking Design:** When performing UAV tracking [6], the existing object tracking schemes [3], [4] fail to take the UAV characteristics (e.g., the fast mobility, the small size) into

considerations, which will decrease the QoS between the terrestrial, the aerial, and the satellite layers. How to achieve security-aware UAV tracking represents the third challenge.

To address the above-mentioned challenges, in this study, we propose a novel and secure framework named SECTRACKER. SECTRACKER addresses the security, efficiency, and accuracy issues existing in the UAV tracking task via the following aspects. *First,* we design a *Local Voting based Detection Module* to identify the trackers with message spoofing attacks. To detect malicious behaviors (e.g., the message flooding attack, the black hole attack), we design a novel *Routing Evidence based Detection Module* as the countermeasure. *Then,* to improve the efficiency of the proposed detection schemes, we design a probabilistic detection algorithm, followed by the detailed game theory based analysis. *Finally,* as the traditional mechanisms of object tracking fail to meet the requirements of accurate, secure, and real-time tracking [3], [7], we design a novel UAV Tracking Module fitting with the SAGIN structure to deal with the tracking accuracy problem. Note that, this study focuses on the security issues when tracking UAVs. Accurate tracking is leveraged as an example to verify the usability of SECTRACKER.

More specifically, for the security and efficiency aspects, SECTRACKER mainly considers the above-mentioned two attacks (i.e., message spoofing and routing misbehavior) in the terrestrial layer. For the message spoofing attack which drops out the tracking result or sends fake UAV locations to spoof the UVA tracking system, we propose a novel Local Voting based Data Voting and Aggregation algorithm to detect the message spoofing behavior of the tracker nodes (e.g., the vehicle sensors, the mobile base stations) in the range of terrestrial stations timely to ensure the security of the components in SAGIN. Considering the routing misbehavior attack, we introduce the Routing Evidence based Detection Algorithm to cross-check the interaction history and message sharing via tracker nodes to ensure security in SAGIN. To achieve an efficient management, we set up a managing center station (MCS) to manage the whole UAV tracking system. By leveraging the probabilistic detection algorithm and game theory based analysis, the overhead of detection can be optimized, which can achieve the trade off between security and utility.

We implement the SECTRACKER on a new dataset named SF-UAV for the real-world space-air-ground scenarios. Compared with the state-of-the-art results in UAV tracking, for instance, the CCOT, the BACF [3], [4], *etc*, SECTRACKER is superior in the aspects of the tracking accuracy and the tracking speed. SECTRACKER can achieve the accuracy of 96.64% with a tracking speed of 61.41 frames per second (fps) when all proposed mechanisms (i.e., Local Voting based Detection Module and Routing Evidence based Detection Module) are employed. The precision of SECTRACKER in the experimental evaluation in terms of temporal and spatial robustness is also above 95%, which is superior to existing results. The contributions of this work are summarized as follows:

- We propose a novel and effective UAV tracking framework, which is aiming to solve the existing security, efficiency, and accuracy issues in UAV tracking. SECTRACKER can achieve high-accuracy and low-time-latency tracking results in SAGIN.
- To overcome the security challenges, we implement the Local Voting based Detection Module and the Routing Evidence based Detection Module to exclude the malicious nodes. We further propose a novel probabilistic detection algorithm and game theory based analysis to reduce the overhead of energy consumption, which can enhance the efficiency of the proposed SECTRACKER.
- To demonstrate the effectiveness of SECTRACKER, in this study, we collect a real-world UAV tracking dataset and will open-source it to the public for further research.

The remainder of this paper is organized as follows. In Section II, we introduce the preliminaries of this work. The overview of SECTRACKER is introduced in Section III. In Section IV, we elaborate the detailed design of SECTRACKER which is followed by evaluation, discussion, and conclusion in Section V, VI, VII respectively.

II. PRELIMINARIES AND RELATED WORK

In this section, we introduce the preliminaries and related works about SECTRACKER.

A. Sagin

As illustrated in Fig. 1, SAGIN consists of the space networks (e.g., low earth orbit (LEO) satellites), the aerial networks (e.g., unmanned aerial vehicles (UAVs)), and the terrestrial networks (e.g., vehicle nodes, mobile base stations). The wide deployment of Wi-Fi, Long-Term Evolution (LTE), and 5G [8] contributes to the development of the Internet of Things and the Internet of Vehicles [9], [10], which also provides great convenience to deploy SAGIN.

The superiority of SAGIN (e.g., cost-efficient, wide-range, high-throughout) receives the attention of both the industry and academia. For instance, The United States Department of Defense (DoD) and the National Aeronautics and Space Administration (NASA) propose the Global Information Grid (GIG) and the transformational satellite (TSAT) system to leverage the advantages of SAGIN [11], [12]. Researchers [13], [14] also put efforts to optimize the SAGIN structure or develop an effective platform to ensure the quality of service (QoS) requirements by software-defined networking (SDN), network function virtualization (NFV) [15], and artificial intelligence techniques. Furthermore, the important work [16] proposes the future of SAGIN with the development of 6 G, and poses serious security threats in SAGIN.

Till now, existing research in SAGIN mainly focuses on the optimization and implementation of the network, especially the deep space and satellites [17]. UAV tracking task in the aerial layer is not only the backbone but also the bottleneck to support and develop the communication between the aerial, the terrestrial, and the satellite layers, which is the main focus of this study.

B. UAV Tracking

As shown in Fig. 1, since UAV plays as the backbone of the communication, automatic UAV tracking is more than critical

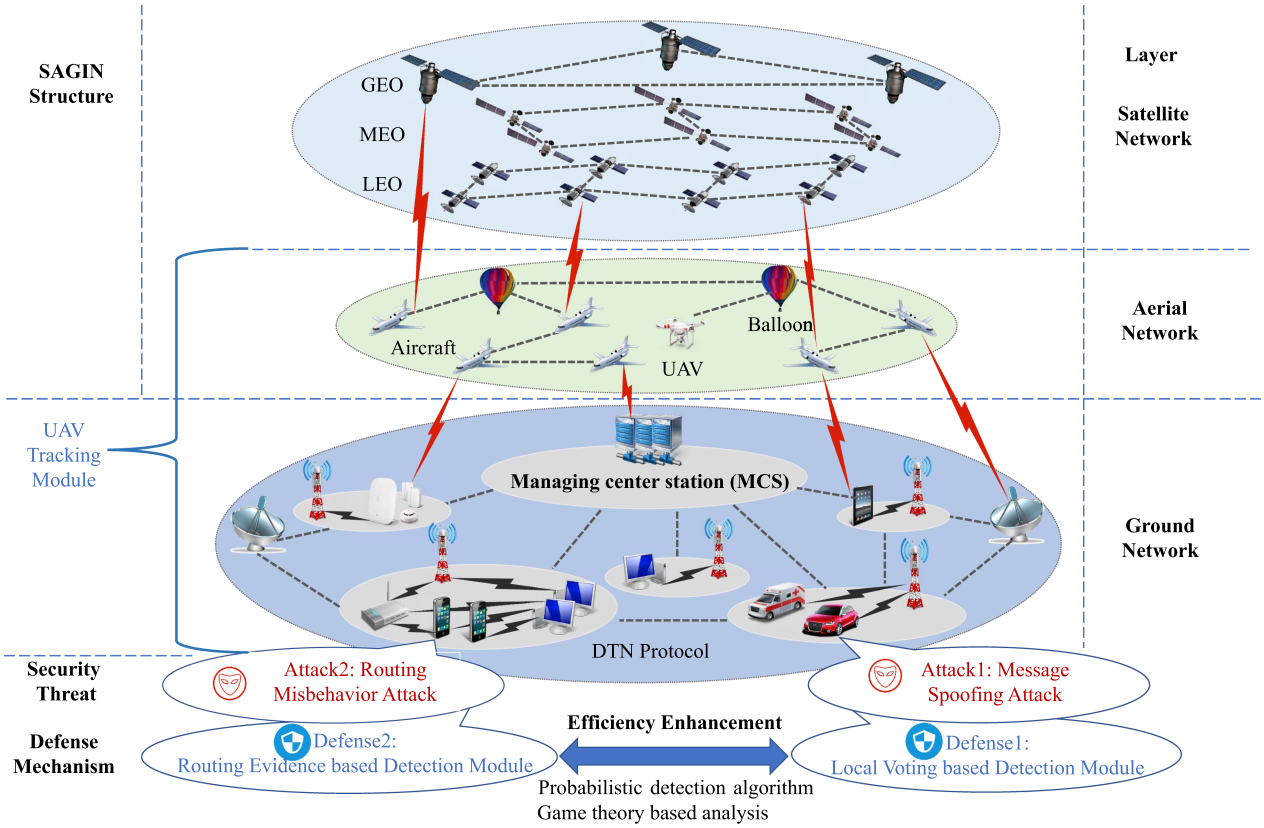

Fig. 1. The overall framework of SECTRACKER.

for the wide deployment of SAGIN. The tracking of UAVs is based on the collaboration of the vehicle nodes in the terrestrial layer, the MCS, and the tracked UAV. The nodes in the terrestrial layer send commands to the UAV, which are monitored and processed by the MCS, and the UAV executes the actions according to the commands from the MCS. At the same time, the nodes capture the videos of the UAV and implement the existing tracker to track the UAV. With the assistance of nodes in different layers, the location of the UAV can be precisely determined. After that, the tracked information is shared inside the SAGIN.

The existing works in the field of UAV tracking can be categorized into two kinds: tracking mechanism based on correlation filtering [3], [4], or based on deep learning [18]. To track the UAV effectively and automatically, a background-aware correlation filter for movement tracking (BACF) [4] is proposed, which can adjust the model according to the different background environments. Besides, the continuous convolution operator (CCOT) [3] is designed to achieve the fine-grained UAV tracking by feature integration. Furthermore, an efficient convolution operator (ECO) [19] leverages the advantages of CCOT, which improves the accuracy and decreases the time-latency simultaneously by optimizing the CCOT. However, existing works receive less attention on the aspects of security, which motivate us to propose SECTRACKER. Different from existing tracking schemes, SECTRACKER considers much more about the security issue, such as detecting the malicious nodes, setting up the secure routing strategy.

C. Security Threats and Efficiency Limitation in SAGIN

The security threats in UAV tracking are rarely studied by the researchers, which may cause serious consequences with the deployment of SAGIN [20]. There are mainly two kinds of attacks in SAGIN. The first is the message spoofing attack [21], and the other is the routing misbehavior attack [21]. In this work, we aim to defend against these two kinds of attacks when securely tracking UAVs.

Message spoofing attack: In the terrestrial layer, the malicious tracker nodes controlled by the attacker may generate fake results (i.e., UAV locations) to spoof the UAV tracking system. Besides, the aforementioned malicious attack can seriously threaten the QoS of the communication between the UAV and terrestrial nodes in SAGIN, which may cause the loss of useful information, and even the crash of SAGIN. As a result, it is particularly crucial to detect the malicious nodes before conducting the tracking process.

Routing misbehavior attack: In the terrestrial layer, the malicious tracker nodes may perform the attacking behaviors to threaten the security of SAGIN, which includes the message flooding attack, and the black hole attack [16]. The mentioned attack will also seriously decrease the QoS of the trackers, more specifically, SAGIN will crash due to the effect of the attack. Different from existing works [22], [23], [24], our work SECTRACKER considers more kinds of malicious attacks in SAGIN.

Efficiency Issue: For a given terrestrial station, multiple nodes are managed in the range of the terrestrial station. The overhead

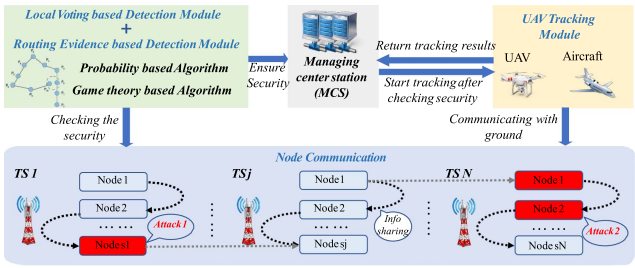

Fig. 2. System module of SECTRACKER.

of detecting the involved nodes and verifying all the messages may be huge in real-world deployment. [25] proposes a novel UAV edge computing IoT networks (UECIN) framework to balance the load of UAV in SAGIN, which poses an important direction for energy optimization. [26] proposes a new style of ultra-dense edge computing (UDEC) to decrease the overall efficiency, which is an impressive work in energy issues. To reduce the overhead of verification and detection schemes, we introduce the probabilistic detection algorithm to enhance the efficiency of SECTRACKER. Besides, inspired by [27], we conduct the game theory based analysis to discuss the optimal strategies in the different cases.

In this study, focusing on the above-mentioned security threats, SECTRACKER implements the defense mechanism to achieve both excluding malicious nodes and improving the efficiency of the proposed mechanisms.

III. PROBLEM FORMULATION

A. Threat Model

Attack types in this study: As illustrated in Fig. 1, we mainly consider two types of attacks, *message spoofing attack* and *routing misbehavior attack* as mentioned in Section II-C. When conducting the UAV tracking in SAGIN, *message spoofing* represents the malicious nodes spreading fake UAV locations or even abandoning the inferred trace. At the same time, *routing misbehavior* represents the message flooding, the black hole attack, and the grey hole attack, which may threaten the whole UAV tracking system. Thus the QoS of SAGIN will be seriously impacted.

Attacker's capability: For the aspect of the attacker's capability, in this study, we assume that the attacker can perform two above-mentioned attacks via hijacking the tracker nodes in the terrestrial layer of SAGIN. More specifically, as illustrated in Fig. 2, the malicious tracker nodes (e.g., vehicles sensor nodes, mobile base stations) can seriously damage the quality of services in SAGIN via performing the man-in-the-middle attack (MITM) [28], the DDoS attack, and so on. As a result, it is of great significance to propose the SECTRACKER to ensure the security of the UAV tracking process. In this study, we do not assume the attacker has the ability to get control of the majority of the nodes in SAGIN or intrude on the management center station (MCS), since they are quite strong assumptions in the real-world scenario.

B. Design Goal

Compared with existing trackers (e.g., CCOT [3], BACF [4], AutoTrack [29]) without the security mechanisms as mentioned in Section II, SECTRACKER is designed to achieve the following goals.

Secure tracking and routing: To ensure the robustness of the UAV tracking system in SAGIN, eliminating the damages caused by the malicious attacks plays a key role. Therefore, our proposed SECTRACKER should detect the tracker nodes with the aforementioned attack (i.e., the message spoofing attack, the routing misbehavior attack) and exclude them from further tracking tasks.

Efficiency improvement: In the large-range SAGIN, the managing and detection of the whole nodes require lots of calculation resources [30], [31]. Therefore, the proposed SECTRACKER should be efficient during the whole procedure of UAV tracking.

Accurate tracking: When there exist malicious attacks as mentioned in Section II, SECTRACKER should achieve the goal of tracking UAVs with a high accuracy that is superior to the existing system. Besides, the tracking algorithm should be lightweight to be deployed in SAGIN and realize real-time tracking with a low time latency.

IV. SYSTEM DESIGN

A. System Overview

As illustrated in Fig. 2, to achieve the design goals of secure, efficient, and accurate UAV tracking, SECTRACKER consists of three components: Local Voting based Detection Module, Routing Evidence based Detection Module, and UAV Tracking Module. Firstly, to ensure the secure UAV tracking configuration, the Local Voting based Detection Module detects the tracker nodes conducting message spoofing attacks and excludes them. Secondly, to handle the various misbehavior attacks existing in the network layer, for instance, the message flooding attack, the black hole attack, and the grey hole attack, we design the Routing Evidence based Detection Module, which utilizes the interaction information between tracker nodes as the evidence. By comparing the interactive routing evidence, the contact and forwarding process can be inferred. To check the malicious nodes in this network, we define the routing checking rules for various attacks. Then, based on the aforementioned rules, the routing misbehavior can be discovered after checking the collected evidence. Thirdly, to reduce the checking and verification overhead, the probabilistic detection algorithm and game theory based analysis are proposed to evaluate the performance of SECTRACKER. Finally, armed with the two aforementioned modules, UAV Tracking Module can track the geographical location of the target UAV securely and accurately.

Note that, the important symbols are shown in Table I.

B. Local Voting Based Detection Module

In this subsection, SECTRACKER deploys the Local Voting based Detection Module to detect those tracker nodes which perform message spoofing attacks in the terrestrial layers. We

TABLE I
SUMMARY OF IMPORTANT SYMBOLS

Symbol	Definition
TS	Terrestrial station
s_j	The nodes in the j-th TS
P_i	The estimated UAV locations
P_v	The set of all voting nodes
P_{aggr}	The aggregated location
$K_{i,j}$	The generated key between two given nodes
$Q_{nexthop}$	The set of next hop for a given node
$E_{i\Rightarrow j}^{forward}$	The forwarding evidence between node i and node j
SIG	A signature generated for message sharing
PF_{Node}^{i}	The payoff when the node choose to abandon the message
$Cost_{Reward}$	The received reward to encourage the node transmitting
T_{trail}	The detection time period
$D(B_i)$	the number of detected benign nodes for TS_i

elaborate details on each step of our proposed modules that are implemented to enhance the security of SECTRACKER.

1) Message Sharing by Tracker Nodes: Given a $L \times W$ region in the terrestrial space where L is the length and W is the width, we divide it into N parts, each part is in the control of a specific terrestrial station TS_i. Note that, for N parts (e.g., $N = 9$ in our experiments), we denote N terrestrial stations as $TS = (TS_1, TS_2, \ldots, TS_N)$, and the j-th part contains s_j tracker nodes.

For a given part with M tracker nodes, before conducting UAV tracking, M nodes need to recognize and localize the target UAV independently. Different from UAV tracking, the initial localization of UAV can be determined by nodes via utilizing the blob detection techniques [32] (2D position) and 3D reconstruction [33], [34] (3D position). SECTRACKER starts to detect the nodes with the message spoofing behaviors in this phase. Note that our proposed Local Voting based Detection Module is decentralized, thus ensuring securing message sharing by M tracker nodes plays a critical role.

For the node n_i which aims to share the estimated UAV 3D localization (x_i, y_i, z_i) to the node n_j, it generates a tuple $m_{i\Rightarrow j} = ((x_i, y_i, z_i), n_i, n_j, t_s)$, where t_s is the timestamp of the message sharing. Let *GroupGen* be a polynomial time algorithm, we input security parameter 1^k and output a cyclic group $\mathbb{G}$, while p is the prime order and g is the generator. Note that, $\mathbb{G}$ is required to satisfy the Decisional Diffie-Hellman Assumption (DDH) [35], which is to ensure the generated key is computationally indistinguishable with a uniformly chosen group element for any probabilistic polynomial time attacker. The i-th node n_i and the j-th node n_j pick a, b randomly from Z_p and exchange g^a, g^b, and finally g^{ab} mod p is set as the generated key $K_{i,j}$. Then, n_i generates a signature $SIG_{i\Rightarrow j} = Sig(K_{i,j}, H(m_{i\Rightarrow j}))$ by the signature algorithm $Sig(\cdot)$, where $H(\cdot)$ is a Hash function [36]. Finally, the message broadcasted by n_i to nearby wireless tracker nodes can be represented as

$$M_{i\Rightarrow j} = (m_{i\Rightarrow j}, SIG_{i\Rightarrow j}). \tag{1}$$

$M_{i\Rightarrow j}$ may experience multiple-hop and arrive at the receiver as $M'_{i\Rightarrow j} = (m_{i\Rightarrow j}, ' SIG_{i\Rightarrow j})$. Since the signature is immutable, $SIG_{i\Rightarrow j}$ cannot be forged by any potential attacker.

For the receiver n_j, when it receives $M'_{i\Rightarrow j}$, it leverages the pre-shared $K_{i,j}$ to calculate a new signature $SIG_{j\Rightarrow i} = Sig(K_{i,j}, H(m'_{i\Rightarrow j}))$. If $SIG_{j\Rightarrow i} = SIG_{i\Rightarrow j}$, it means the location information never undergoes any tamper. For all M nodes, SECTRACKER requires each node to share the message with other nodes following the above steps. Ultimately, M nodes share the UAV location messages securely, then SECTRACKER detects the nodes with message spoofing behaviors.

2) Local Voting Based Message Spoofing Detection: For a given terrestrial part with M nodes, SECTRACKER checks each node for whether it has message spoofing behavior. For the target tracker node n_{target}, we refer other $M-1$ nodes as voting nodes $N_v = \{n_1, n_2, \ldots, n_{M-1}\}$. The basic detection insight is that the nodes that conduct message spoofing will share the spoofed UAV location to mislead the final UAV tracking results. The spoofed location will inevitably be inconsistent with the location results from other nodes (i.e., voting nodes) only if the benign nodes occupy the majority, which can be held on in most cases. The detailed voting steps are listed below:

Location aggregation from voting nodes: For the i-th node n_i in the voting nodes N_v, note that $n_{target} \notin N_v$, it has the estimated UAV location $P_i = (x_i, y_i, z_i)$. SECTRACKER aggregates estimated locations from all voting nodes $P_v = \{P_1, P_2, \ldots, P_{M-1}\}$ as

$$P_{aggr} = \left(\frac{\sum_{i=1}^{M-1} x_i}{M-1}, \frac{\sum_{i=1}^{M-1} y_i}{M-1}, \frac{\sum_{i=1}^{M-1} z_i}{M-1} \right). \tag{2}$$

Distance calculation: For each node in the voting nodes, SECTRACKER calculates the distance between its estimated location and the aggregated location P_{aggr}. For instance, the distance derived from the node n_i can be represented as

$$d_i = |P_i - P_{aggr}|. \tag{3}$$

Similarly, for the detected node v_t, SECTRACKER calculates its distance d_t.

Voting based detection: For $M-1$ voting nodes, we assign its voting values as $V = \{v_1, v_2, \ldots, v_{M-1}\}$, where v_i is the vote value of the node n_i. For the target detected node n_{target}, node n_i decides its vote value according to the following principles:

- *Case 1:* when $d_t \leq d_i$, it means the estimated location of n_{target} is closer to the aggregated location P_{aggr} than that of n_i. In this case, n_i sets v_i as 1.
- *Case 2:* when $d_t \geq d_i$ and $d_t - d_i \leq \epsilon$, where ϵ is a predefined detection threshold, it means P_t (i.e., P_t represents the location of the tested node, which may be a benign node or a malicious node) is far away from P_{aggr} than that of n_i but the deviation is acceptable. Note that, ϵ is set empirically and SECTRACKER chooses $\epsilon = 2 \times \sigma$ in the experiments, where σ is the standard variance of the voting nodes' distances $d_v = \{d_1, d_2, \ldots, d_{M-1}\}$. In this case, n_i still sets v_i as 1.
- *Case 3:* when $d_t \geq d_i$ and $d_t - d_i \geq \epsilon$, it means P_t occurs an obvious deviation compared with n_i. In this case, n_i does not trust n_{target} and will set v_i as 0.

After getting vote values from all voting nodes, we calculate the final vote value $Score$ as below

$$Score = \frac{\sum_{i=1}^{M-1} v_i}{M-1}. \tag{4}$$

If $Score \geq 0.5$, it means the majority of the voting nodes trust the detected tracker node n_{target}. Otherwise, it means n_{target} is likely a node conducting the message spoofing attack, and SECTRACKER excludes it from further tracking procedures.

During a UAV tracking task, SECTRACKER conducts several iterations of Local Voting based Detection Modules in order to exclude as many malicious nodes as possible. In the next subsection, we will discuss how to reduce the energy consumption generated in this module.

C. Routing Evidence Based Detection Module

In this subsection, we first introduce the packet transmission process in SAGIN, which is a kind of delay tolerant network (DTN) [17]. After that, we propose the Routing Evidence based Detection Module to defend against the routing misbehavior attack (e.g., the message flooding attack, the black hole attack, the grey hole attack).

1) Packet Transmission in DTN: SAGIN is a combination of sensor networks with scheduled intermittent connectivity [17], [37], which is regarded as a typical type of DTN. Because of the unique characteristics, the packet transmission follows the DTN protocols. More specifically, the nodes will send the message packet until the path for forwarding appears, for instance, the next hop node is activated or moving into the ranges of the given nodes, which is formulated as "opportunistic" schemes. However, the malicious behavior may be performed by the attacker. The message packets can be abandoned or partly abandoned by the malicious nodes, which is notated as the black and grey hole attack [16]. Besides, the malicious nodes can execute the message flooding attack to cause the DoS attack, which will seriously affect the QoS of SAGIN.

2) Routing Evidence Generation: As the SAGIN remains the characteristics of intermittent connectivity existing in DTN, the discovery of malicious behavior faces the challenges of lacking witness. Thus, inspired by [27], [36], we propose the routing evidence to record the behaviors in SAGIN for checking. For the k-th given terrestrial station, the routing evidence should be uploaded to the TS_k by the tracker nodes. The interaction evidence is categorized into two parts: the forwarding evidence and the contact evidence.

Forwarding Evidence: For the i-th node n_i and the j-th node n_j managed by TS_k, the i-th node n_i will determine whether the j-th node n_j is suitable for the packet transmission in terms of the routing protocol. More specifically, firstly, we define two transmission condition bits C_{flag1} and C_{flag2} with the initial value 0. Then, if the distance $Dis_{i\Rightarrow j}$ between n_i and n_j satisfies $Dis_{i\Rightarrow j} \leq Dis_{th}$, where Dis_{th} is the pre-defined threshold, we set the C_{flag1} as 1. Simultaneously, if the next hop n_j satisfies the DTN routing protocols (e.g., the Direct protocol, the Randomized Routing protocol, the Epidemic protocol, the Spray & Wait protocol) [38], we set the second condition symbol bit C_{flag2} as 1. Finally, we calculate the combined condition symbol bit $C_{flag} = C_{flag1} \cap C_{flag2}$. If C_{flag} is 1, it means the next hop node n_j is suitable for transmission, thus the message packet will be sent. At the same time, the forwarding evidence will be generated after the forwarding. The $SIG_{i\Rightarrow j}$ defined in Section IV-B2 is involved to ensure the security of the message packet by the signature immutability. The evidence $E_{i\Rightarrow j}^{forward}$ is defined as following

$$E_{i\Rightarrow j}^{forward} = (m, n_j, t_s, t_l, C_{flag}, SIG_{i\Rightarrow j}), \tag{5}$$

where t_s is the timestamp of the message packet sharing, t_l is the Time-to-live (TTL) value before the message m got discarded.

Contact Evidence: In this study, different from the forwarding evidence, the contact evidence includes the contact history between the two nodes, even the forwarding condition does not satisfies ($C_{flag} = C_{flag1} \cap C_{flag2} = 0$). The contact evidence between the i-th node n_i and the j-th node n_j is defined as

$$\begin{aligned} &E_{i\Rightarrow j}^{contact} \cup E_{i\Leftarrow j}^{contact} \\ &= (m, n_i, n_j, t_s, t_l, C_{flag}, SIG_{i\Rightarrow j}, SIG_{i\Leftarrow j}), \end{aligned} \tag{6}$$

where the notation is the same as the forwarding evidence.

For a given terrestrial station TS_k with s_k tracker nodes, the evidence during the packet transmission is collected as

$$E_{aggr}(i) = \left(E_{i\Rightarrow j}^{forward}, E_{i\Rightarrow j}^{contact} \cup E_{i\Leftarrow j}^{contact}\right), \tag{7}$$

$$E_{aggr}^{TS_k} = E_{aggr}(1) \cup E_{aggr}(2) \ldots \cup E_{aggr}(s_k - 1) \cup E_{aggr}(s_k), \tag{8}$$

where $E_{aggr}(i)$ represents the aggregated evidence for the i-th node in TS_k, and $E_{aggr}^{TS_k}$ is the aggregated evidence for all the collected evidence in TS_k.

3) Detection Schemes: After collecting all the evidence, we design the detection schemes to identify the misbehavior. According to the aggregated collected evidence $E_{aggr}^{TS_k}$ in TS_k, we construct the set of next hop as $Q_{nexthop}(i)$ for the i-th node. Besides, we also construct the set of contacting nodes as $Q_{contact}(i)$ for the i-th node. Note that, $\Gamma 1_m$ represents the set of message packets that are needed to be sent. $\Gamma 2_m$ represents the set of forwarded messages according to the evidence.

For the black hole attack or the grey hole attack, in other words, the node refuses to forward the message packets or abandon the message packets selfishly even under the condition that forwarding is satisfied. The checking rules are

$$\exists m \in \Gamma 1_m, m \in \Gamma 2_m, Q_{nexthop}(i) \not\subseteq Q_{contact}(i). \tag{9}$$

$$\exists m \in \Gamma 1_m, m \notin \Gamma 2_m, Q_{contact}(i)! = 0 \cup Q_{nexthop}(i) == 0. \tag{10}$$

The mentioned situation represents when the forwarding opportunity appears, the i-th node chooses not to forward the message m or send it to other unauthorized nodes which breaks the routing rules. These nodes satisfied with the two mentioned situations will be regarded as malicious nodes.

For the message flooding attack, we design another detection scheme, the k-th terrestrial station TS_k collects the timestamp from forwarding evidence between the node n_i and node n_j. For a given message m, the t_m represents the packet transmitting time from n_i to n_j, which is calculated as $t_m = t_{s1} - t_{s2}$. t_{s1} and t_{s2} come from different evidence, which are the timestamp for message m during the transmission. The message packet size is defined as $|Msg_m|$ for message m. Note that, the set of

message packets is defined as Set_M. The traffic speed of node n_i at t_{s1} is calculated as

$$TF_i(t_{s1}) = \sum_{m \in Set_M} \frac{|Msg_m|}{t_m}. \tag{11}$$

For a given TS_k, we calculate the traffic speed at t_{s1} as

$$Mean(TF(t_{s1})) = \frac{\sum_{i=0}^{s_k} TF_i(t_{s1})}{s_k}. \tag{12}$$

If the traffic speed of node n_i is larger than a pre-defined threshold (i.e., in experiments, we set the threshold as 3) more than the mean value of traffic in SAGIN in most time of the transmission, we will regard it as the malicious node which performs the message flooding attack. Besides, according to the checking rules in the aforementioned attack, SECTRACKER can also assist to detect the message eavesdropping attack and the denial of service (DoS) attack.

D. Efficiency Enhancement Mechanisms

After excluding the nodes with the misbehavior, however, the UAV tracking still suffers from the problem of efficiency. SECTRACKER utilizes a probabilistic detection algorithm and game theory based analysis to balance the efficiency and security, which are used in the detection and UAV tacking.

1) Probabilistic Detection Algorithm: To save the detection and verification overhead during the detection as mentioned in Section IV-B and Section IV-C, SECTRACKER deploys a probabilistic detection algorithm, which is widely used in malicious node checking [27]. The basic insight is that: in a given terrestrial part, during the detection process, it is not necessary to check each node in each iteration. Otherwise, to save the overhead, SECTRACKER can pay more attention to those low reputation nodes and less attention to the high ones. Based on this, this algorithm will increase the checking probability of the suspicious nodes and decrease the checking probability of the possibly benign nodes in each checking iteration. Inspired by [27], [36], we design the probabilistic detection algorithm to check the malicious nodes and behaviors.

We implement the probabilistic detection algorithm as an enhancement of the two detection modules to detect nodes with message spoofing and routing misbehavior in the j-th terrestrial station TS_j. For the s_j sensor nodes in the divided part controlled by the j-th terrestrial station TS_j, TS_j collects the message and routing evidence of each sensor node for detection. We initially set the probability of checking each node as $Prob$. For benign nodes, in its first several checking iterations, if the node behaviors are all recognized as benign by SECTRACKER, the reputation of these nodes will be regarded as "benign," the node will receive a reward $Cost_{Reward}$ from the TS_j, and the MCS will decrease the corresponding checking probability (i.e., $Prob$) of these nodes. Otherwise, For malicious nodes, when they are checked and detected as malicious, their reputation will decrease, and the TS_j will punish $Cost_{Punish}$ the node. A low reputation will attract the MCS's more attention and result in a high checking probability. Note that, the transmitting overhead for each node is defined as $Cost_{Tran}$, and the checking overhead for TS_j is $Cost_{Check}$. Note that, if the message is transmitted correctly, the corresponding TS will receive an information gain as $Gain$. The current reputation for each node is represented as $Current^i_{Node}$, while the current reputation for the corresponding TS is notated as $Current_{TS}$.

Thus, in each iteration, in the s_j tracker sensor nodes, the number of nodes that need to be checked can be calculated as

$$Num_{CN}(j) = \lfloor s_j \times Prob \rfloor, \tag{13}$$

$$Num_{CN} = \sum_{j=1}^{N} Num_{CN}(j), \tag{14}$$

where Num_{CN} is the number of nodes that should be checked in SAGIN (which is chosen randomly), and N is the number of terrestrial stations TS. The checking criteria are elaborated in Section IV-B and Section IV-C. With the assistance of the probabilistic detection algorithm, the overhead during detecting message spoofing nodes can be significantly reduced.

E. Game Theory Based Analysis

After defining the reputation based strategy, inspired by [27], [36], [39], we conduct the game theory analysis to discuss the tradeoff between detection and overhead.

Theorem 1: As the $Cost_{Tran}$, $Cost_{Reward}$ and $Cost_{Punish}$ are defined in Section IV-D1, if the checking probability $Prob$ is set as

$$Prob = \frac{Cost_{Tran} + \zeta}{Cost_{Reward} + Cost_{Punish}}, \tag{15}$$

the i-th node must choose to forward for the consideration of payoff, and TS_k will save the checking cost compared with checking all the nodes.

Proof: For each node, it has two choices when a message is needed to be forwarded, which are "forward" or "abandon". For a given terrestrial station TS, it has two behaviors too in a detection trail, which are "checking" or "not checking". Thus, the detection trail can be modeled as a static game, which can be categorized into 4 cases.

- *Case 1:* If the i-th node chooses to forward honestly, and the event is detected by the TS, the reputation for node i is updated as $Current^i_{Node} + Cost_{Reward} - Cost_{Tran}$. The reputation for TS is $Current_{TS} + Gain - Cost_{Reward} - Cost_{Check}$.
- *Case 2:* If the i-th node chooses to forward honestly, but the event is not detected by the TS, the reputation for node i is $Current^i_{Node} + Cost_{Reward} - Cost_{Tran}$, the reputation for TS is $Current_{TS} + Gain - Cost_{Reward}$.
- *Case 3:* If the i-th node chooses to abandon the message selfishly, and the event is detected by the TS, the reputation for node i is $Current^i_{Node} - Cost_{Punish}$, the reputation for TS is $Current_{TS} + Cost_{Punish} - Cost_{Check}$.
- *Case 4:* If the i-th node chooses to abandon the message selfishly, but the event is not detected by the TS, the reputation for node i is $Current^i_{Node} + Cost_{Reward}$, the reputation for TS is $Current_{TS} - Cost_{Reward}$.

In equation (15), the checking probability $Prob$ is less than 1. It means the number of checking nodes equals to $\sum_{j=1}^{N} s_j \times Prob$, which is less than the number of all nodes $\sum_{j=1}^{N} s_j$. For the i-th node, the payoff when it chooses "abandon" or "forward" can be shown as below.

Condition 1: If the node chooses to abandon, the payoff is

$$\begin{aligned} PF_{Node}^{i} = & -Prob \times Cost_{Punish} \\ & + (1 - Prob) \times Cost_{Reward} \\ = & Cost_{Reward} - Cost_{Tran} - \zeta. \end{aligned} \tag{16}$$

Condition 2: Otherwise, if the node chooses to forward honestly, the payoff is

$$\begin{aligned} PF_{Node}^{i} = & Prob \times (Cost_{Reward} - Cost_{Tran}) \\ & + (1 - Prob) \times (Cost_{Reward} - Cost_{Tran}) \\ = & Cost_{Reward} - Cost_{Tran}, \end{aligned} \tag{17}$$

which is more than $Cost_{Reward} - Cost_{Tran} - \zeta$. As a result, the node i will choose to forward to maximize the payoff PF_{Node}^{i}. As for the corresponding TS, it doesn't need to check all the nodes, the detection overhead and verification overhead will decrease. The static game realizes the Nash Equilibrium [39]. ■

Theorem 2: The transmission in DTN is modeled as a stochastic process, which satisfied the exponential distribution with parameter δ_{ij} [40]. For s_k nodes in TS_k, $Prob$ represents the detection probability, $Mean(\delta)$ represents the mean value of δ_{ij}, in the detection time period T_{trail}, the cost for transmission and verification when making a single contact is notated as $Overhead_{Tran}$ and $Overhead_{Sig}$ separately, while the cost for each trail $Overhead_{T+S}$ is calculated as the sum of $Overhead_{Tran}$ and $Overhead_{Sig}$. The summarized cost in all DTN is calculated as

$$\frac{1}{2} Prob \sum_{k=1}^{N} |s_k|^2 Mean(\delta) T_{trail} (Overhead_{T+S}). \tag{18}$$

Proof: Considering a stochastic process satisfying the exponential distribution with parameter δ_{ij}, δ_{ij} represents the contact rate between n_i and n_j, the interval is calculated as $\frac{1}{\delta_{ij}}$. For s_k nodes in TS_k, the summarized contact number in TS_k is calculated as

$$\begin{aligned} |Num_{contact}^{TS_k}| &= \frac{1}{2} \sum_{i} \sum_{j \neq i} T_{trail} / \frac{1}{\delta_{ij}} \\ &\approx \frac{1}{2} Mean(\delta) T_{trail} \sum_{k=1}^{N} |s_k|^2. \end{aligned} \tag{19}$$

As a result, when the detection probability is set as $Prob$, the overall cost is derived as

$$\begin{aligned} OverallCost &= Prob \sum_{k=1}^{N} |Num_{contact}^{TS_k}| \\ &= \frac{1}{2} Prob \sum_{k=1}^{N} |s_k|^2 Mean(\delta) T_{trail} Overhead_{T+S}. \end{aligned} \tag{20}$$

As shown in Theorem 2, the overhead of detection is linear with the $Prob$. Thus, an appropriate $Prob$ selection can decrease the detection overhead and save the resources in SAGIN. ■

F. UAV Tracking Module

In this subsection, after implementing all the security mechanisms, SECTRACKER finishes the final UAV tracking task by employing the novel proposed algorithms in UAV Tracking Module. To handle the limitations of the traditional tracking mechanisms (e.g., CCOT, BACF) when facing the tasks of UAV tracking in SAGIN, SECTRACKER optimizes the ECO algorithm by jointing the location decision and object tracking to adaptively fit the framework of SAGIN. Finally, we achieve accurate and low-time-latency tracking for UAVs.

Considering the characteristic of UAV tracking in SAGIN, the UAV Tracking Module is comprised of two steps, the location decision and the object tracking respectively. The location decision is designed not only to decide the initial location of the UAVs for the convenience of tracking the UAVs, but also to avoid the moving object missing during the tracking process. With the assistance of this component, if the object in the tracking figure gets lost, the location decision can be reused to relocate the location of the UAV. We implement the blob detection algorithm in the Open Source Computer Vision Library (OpenCV) libraries [32] to realize the aforementioned function.

In the object tracking step, after receiving the location from the location decision step, for the same image patch, each training sample set from the testing figure of UAV contains j channels, which are represented as $x_1, x_2, \ldots, x_j$. An interpolation operator J_d defined as below is adopted to achieve the goal of UAV tracking.

$$J_d \left\{x^d\right\}(t) = \sum_{n=0}^{N_d - 1} x^d[n] b_d \left(t - \frac{T}{N_d} n\right), \tag{21}$$

where the b_d is the interpolation kernel ($T > 0$). For the shifted function b_d, $x^d[n]$ represents the weight parameters and N_d is denoted as the resolution for each feature layer in the framework.

To overcome the disadvantage of over-fitting and complexity, we adopt a more efficient convolution operator (ECO) [19] to track the UAV in this study. In the implementation of ECO, compared with the set of constitutional filters $f = (f^1, f^2, \ldots, f^D) \in L^2(D)^T$, the number of filters C needed in this approach is less than the number of filters D, defined in the CCOT. The combination of the linear filters in ECO frameworks is calculated as $\sum_{c=1}^{C} \lambda_{d,c} f^c$, where the coefficients of linear combinations are denoted as $\lambda_{d,c}$, a matrix ϕ^{T} is constructed by the coefficients of linear combinations $\lambda_{d,c}$ with the dimension $D \times C$. The purpose of factorized convolution operator $S_{Pf}\{x\}$ is to reduce the parameter numbers, which is calculated as

$$S_{Pf}\{x\} = \sum_{c,d} \lambda_{d,c} f^c \times J_d \left\{x^d\right\} = f \times \phi^{\mathrm{T}} J\{x\}. \tag{22}$$

In the training process, to reduce the scale of the training samples, we employ the Gaussian Mixture Model (GMM) to evaluate the training loss E. Compared with the training loss

defined in CCOT, the training loss E is optimized as follows

$$E(f) = \mathbb{E}\left\{\|S_{Pf}\{x\} - y\|_{L^2}^2\right\} + \sum_{d=1}^{D} \left\|wf^d\right\|_{L^2}^2, \quad (23)$$

where y is the output of the ECO framework.

Besides, if the target (i.e., UAV) gets lost when conducting the tracking process, the location decision component will be reused to locate the UAV. In addition, with the assistance of blob detection, the terrestrial nodes can continuously track UAVs with the proposed SECTRACKER. For the j-th terrestrial part with the terrestrial station TS_j, after all the tracker nodes finish UAV tracking, the cluster head nodes will aggregate the location messages from each node and send them to the corresponding TS_j. Then, for all TS, the UAV location messages will be sent to MCS for executing the later commands and maintaining the communication between satellite and ground networks. Finally, with the deployment of the UAV Tracking Module, SECTRACKER can achieve the design goal of high-efficiency and low-time latency simultaneously.

V. EVALUATION

In this section, we first evaluate the overall performance of SECTRACKER, including the visualization of tracking results. Then we explore various factors that may affect the performance of SECTRACKER. Finally, we compare our work with existing priors.

A. Experimental Settings

1) Experimental Configurations: In the process of UAV tracking, we first implement the Local Voting based Detection Module and the Routing Evidence based Detection Module to exclude the impact of the malicious tracker nodes, the number of the terrestrial station N is set to be 9. The summarized number of sensor nodes $\sum_{j=1}^{N} s_j$ is set to 50, 100, and 200 randomly generated in the region of $L \times W$, which is set as 300×300 km^2. In this study, 10% of the nodes are assumed to be malicious nodes. The opportunistic networking simulator [41] is adopted to simulate the SAGIN. With the setting of the parameters, the experiments are conducted in 100 rounds.

After the checking process, the tracking parameters are introduced as follows. The training learning rate is set to be 0.009, and the maximum number of stored training samples is limited to 50 for the training speed consideration. To implement the factorized convolution formulation, we adopt the Principal Component Analysis (PCA) for the initialization of the projection matrix, the regularization parameter of the projection matrix is set to be 5×10^{-8} empirically. All the evaluation experiments of SECTRACKER are conducted on a desktop with 64-bit Ubuntu 18.04 OS, Intel Core i7 CPU, and 64 GB RAM.

2) Dataset: To demonstrate the effectiveness of SECTRACKER in the real-world scenario in the field of UAV tracking, we establish a dataset, dubbed as SF-UAV, which is expected to be open-source for the community. To verify the effectiveness and robustness of SECTRACKER, the dataset is comprised of a tracking video captured by a 4 K camera with a speed of 29 frames per second. The time range of the pre-processed dataset is 21 seconds. The summarized frames are calculated as $21 \times 29 = 609$. Besides, compared with the existing dataset UAV123 [42], our dataset is superior in terms of the sharpness and usability in the quality of videos. Since all the experiments are conducted by UAVs controlled by the experts, this study is exempt from the IRB approval of the institutions.

3) Evaluation Metrics: In this study, we define $Overall_{Accu}$ to evaluate the overall accuracy of the existing mechanism implemented in the UAV tracking process.

$$Overall_B = \frac{\sum_{i=1}^{N} D(B_i) \times TrackPrecision_i}{\sum_{i=1}^{N}(D(B_i) + UD(M_i))}, \quad (24)$$

$$Overall_M = \frac{\sum_{i=1}^{N} UD(M_i) \times (1 - TrackPrecision_i)}{\sum_{i=1}^{N}(D(B_i) + UD(M_i))}, \quad (25)$$

where $D(B_i)$ represents the number of detected benign sensor nodes for the i-th terrestrial station TS, they are detected by the tracking mechanisms and will honestly transmit and forward the packets in the corresponding TS, where $UD(M_i)$ represents the number of malicious sensor nodes that are not detected by the tracking mechanisms, it is the complement of the $D(B_i)$. Note that, the $Track_{Precision}$ is referring to the real-time tracking accuracy of the UAV, which represents the proportion of the detected objects area to the square frame in the videos, which is shown in Fig. 4. In the case that the malicious nodes transmit irrelevant or wrong information of the tracking results, $1 - Track_{Precision}$ (the remaining part of the square frame threshold) is selected for the malicious sensor nodes. In fact, the tracking precision rate for the malicious nodes will be even lower, which will further demonstrate the effectiveness of SECTRACKER in terms of $Overall_{Accu}$.

$$Overall_{Accu} = Overall_B + Overall_M, \quad (26)$$

where $Overall_B$ refers to the accuracy of the detected benign nodes, $Overall_M$ is for the accuracy of the undetected malicious nodes. $Overall_{Accu}$, representing the overall accuracy of the UAV tracking process, is the combination of the results of $Overall_B$ and $Overall_M$.

B. Overall Secure Tracking Performance

1) Overall Performance of SECTRACKER: When considering the message spoofing attack and the routing misbehavior attack, the tracking accuracy of SECTRACKER can achieve the $Overall_{Accu}$ of 96.64% with 61.41 fps. The time overhead with all the proposed mechanisms (i.e., the Local Voting based Detection Module, the Routing Evidence based Detection Module) only increases 1.30 seconds delay for each tracker node, which is acceptable for UAV tracking in real-world scenarios compared with the tracking time overhead (at least 8.58 seconds listed in Table III). The existing mechanism can achieve the best $Overall_{Accu}$ by the means of ECO [19], which is calculated as 88.16%. Therefore, the $Overall_{Accu}$ of SECTRACKER can achieve a significant performance improvement with the implementation of the security mechanisms.

Fig. 3. Background subtraction result from SECTRACKER.

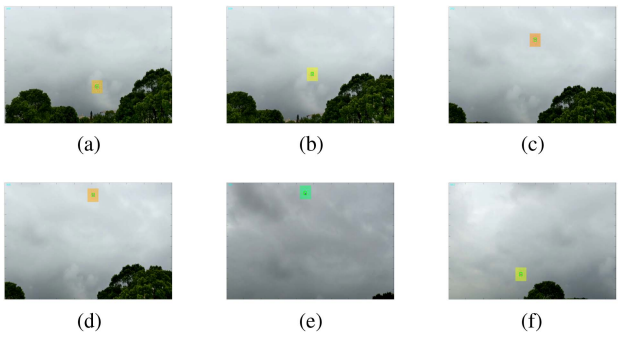

Fig. 4. Tracking Results from SECTRACKER in Time Order (UAV in the Bound). (a) Tracking Results 1. (b) Tracking Results 2. (c) Tracking Results 3. (d) Tracking Results 4. (e) Tracking Results 5. (f) Tracking Results 6.

C. Visualization of the Tracking Results

With the assistance of the blob detection, the location of the tracking target (i.e., UAVs) can be located by pixels, which can be utilized to estimate the size and the initial location of the UAV. After this procedure, the optimized ECO framework in SECTRACKER can be implemented to track the UAV continuously. The results are shown in Fig. 3.

As illustrated in Fig. 4, SECTRACKER can realize the function of locating the initial location for the UAV and conducting all procedures of the real-time tracking under the security protection, which indicates that our SECTRACKER achieves the design goals of effectiveness and efficiency.

D. Impact of Various Factors on SECTRACKER

In this subsection, we evaluate the impact of various factors (e.g., sensor node numbers, $Prob$ choices, sensor node mobility) on SECTRACKER. We define the detection rate [43] to evaluate the performance of the malicious node detection. The detection rate is calculated as

$$Detection = \frac{\sum D(M_{node})}{\sum M_{node}}. \tag{27}$$

where $D(M_{node})$ represents the number of the malicious nodes which are detected. Note that, the detection rate is for the proportion of detected malicious nodes among all malicious nodes.

1) Impact of Tracker Node Number: To evaluate the impact of the node scale on SECTRACKER, we set the node number as 50, 100, and 200 when conducting simulations. We implement the Local Voting based Detection Module and the Routing Evidence based Detection Module to detect the malicious nodes. As shown in Fig. 5(a), the detection rate for malicious nodes increases when the node number increase. The reason is that when the node number increases, the contact between nodes will be more

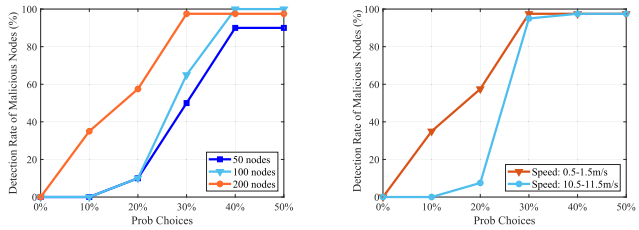

Fig. 5. Evaluation of various factors on SECTRACKER. (a) Detection rate of the malicious nodes when changing $Prob$ and node number. (b) Detection rate of the malicious nodes when changing node moving speed.

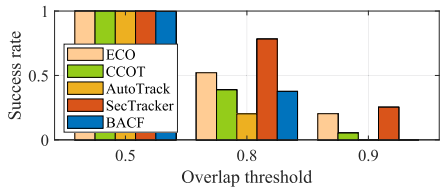

Fig. 6. Precision plot of OPE when changing location error threshold.

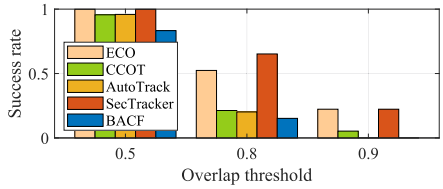

Fig. 7. Precision plot of TRE when changing location error threshold.

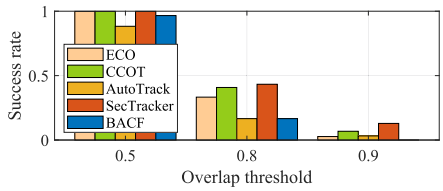

Fig. 8. Precision plot of SRE when changing location error threshold.

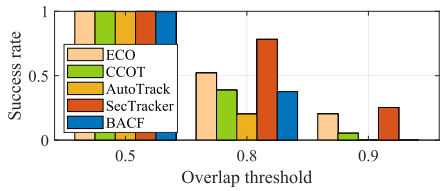

Fig. 9. Success rate plot of OPE when changing overlap threshold.

frequent [44]. At the same time, there will be more useful routing evidence to collect in the management center station (MCS), which can be used to detect the malicious nodes. Thus, the detection rate for the malicious nodes will increase too.

2) Impact of Probability Selection: To evaluate the impact of probability selection when conducting detection, we set the $Prob$ as 10%, 20%, 30%, 40%, and 50% and evaluate the performance. As depicted in Fig. 5(a), the detection rate increases with the increase of checking $Prob$. As a result, to achieve the design goal of low-cost and security, we should balance the trade-off

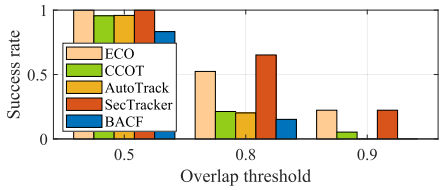

Fig. 10. Success rate plot of TRE when changing overlap threshold.

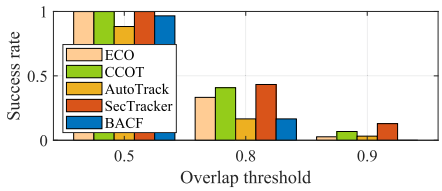

Fig. 11. Success rate plot of SRE when changing overlap threshold.

TABLE II
COMPARISON BETWEEN OUR SYSTEM AND EXISTING SECURE ROUTING MECHANISMS

Work	Message Spoofing Attack	Flooding Attack	Black/Grey Hole Attack	Efficiency Issue	Probability Checking
Our work	✓	✓	✓	✓	✓
[22]	✓	✗	✗	✗	✗
[23]	✗	✓	✗	✗	✗
[24]	✗	✗	✓	✗	✗

between the energy consumption and the performance of SECTRACKER, which is adjusted by the parameter $Prob$. Besides, when the $Prob$ reaches 40%, 90% of the malicious nodes can be identified. Thus, the results demonstrate the usability of the proposed SECTRACKER in the security protection aspects.

3) Impact of Moving Speed: We also change the moving speed of the nodes, which is designed to simulate different kinds of sensors. As shown in Fig. 5(b), when the moving speed increases, the detection rate decreases. It is because when the moving speed of the node increase, the continuous connection between the nodes is even worse. Simultaneously, the collected useful routing evidence will decrease. Thus, more malicious nodes will be ignored when detecting based on the routing evidence, and the detection rate will decrease too. Otherwise, when the $Prob$ reaches 30%, above 90% of the malicious nodes can be detected. The results show the efficiency of SECTRACKER when the moving speed changes.

E. Comparison With State-of-The-Art Works

In this subsection, we first make a detailed comparison in the aspects of the secure routing system. We compare our work with the state-of-the-art work in attacking types, efficiency issue, and the checking method.

According to the result in Table II, it is observed that our work can solve various attacks in SAGIN, for instance, the message spoofing attack, the flooding attack, the black/grey hole attack and so on. Besides, SECTRACKER also proposes the probability based checking schemes and the game theory based algorithm to optimize the efficiency issue. Specifically, existing works mainly consider one type of attack method. The state-of-the-art works do not concentrate on the efficiency issue. In summary, our scheme is superior to existing schemes in secure routing.

Then, to show the superiority of SECTRACKER, we make a comparison between SECTRACKER and existing UAV tracking systems (CCOT [3], BACF [4], AutoTrack [29] ,[1] ECO [19]) in three different aspects: the precision rate, the success rate, and the overall tracking speed. Note that, the other existing tracking systems implement the traditional direct transmitting protocol. SECTRACKER deploys the Local Voting based Detection Module and the Routing Evidence based Detection Module to ensure the security and efficiency of the tracking system.

To compare the performance of SECTRACKER with that of existing mechanisms, three measurements are proposed by [45]. For a given testing video, one-pass evaluation (OPE) is to evaluate the performance only once after initialization from the ground truth location. Temporal robustness evaluation (TRE) and spatial robustness evaluation (SRE) are proposed to evaluate the robustness of SECTRACKER, which differs in the initial setting. TRE perturbs the initialization temporally in the staring frames of the testing figure. SRE perturbs the initialization temporally in the starting boundary boxes in the testing figure. The evaluation results of the success rate and the precision scores are shown in the Figs. 6–11.

Table III shows that for the tracking process without security protection mechanisms, the precision rate of SECTRACKER is still above 96%. SECTRACKER can achieve a success rate of 85% in the tracking process at the speed of 61.41 frames per second (fps). The time overhead of SECTRACKER increases because of the deployment and execution of security mechanisms. However, the overhead is reasonable and acceptable in the scenario of tracking UAVs in SAGIN. For instance, in the OPE evaluation, the time overhead of SECTRACKER is only 25.99% above the best occasions in the existing systems. Therefore, SECTRACKER ensures security protection and energy saving in SAGIN with an acceptable time overhead.

Table IV demonstrates the robustness of SECTRACKER in the aspects of temporal and spatial. Note that, because of all the security and efficient mechanisms implemented, the $Overall_{Accu}$ of SECTRACKER is much superior to the state-of-the-art UAV tracking systems listed in both Table III and Table IV, which can contribute to the development of the tracking system's performance enhancement. Finally, SECTRACKER is superior to existing works in both the performance of the secure routing and the tracking issue.

VI. DISCUSSION

In this section, we conduct the security analysis and discuss some limitations of SECTRACKER.

A. Security Analysis

In this subsection, we give a security analysis for the proposed mechanisms. Firstly, the message of UAV location from the malicious nodes differs from the benign nodes. Because the

[1] AutoTrack is designed for both UAV self-localization and UAV tracking.

TABLE III
COMPARISON WITH STATE-OF-THE-ART SYSTEMS IN THE ASPECTS OF ON SF-UAV FOR OPE EVALUATION

Mechanism	Tracking precision rate	Success rate	Tracking speed	Tracking time overhead	Detection overhead per node	Signature overhead per node	Total overhead per node	Overall accuracy
CCOT	97.30%	75.50%	0.85 fps	658.89 s	0 s	0 s	658.89 s	87.84%
BACF	96.80%	73.70%	68.28 fps	8.58 s	0 s	0 s	8.58 s	87.44%
AutoTrack	97.20%	76.60%	45.50 fps	12.92 s	0 s	0 s	12.92 s	87.76%
ECO	97.70%	79.90%	3.31 fps	178.61 s	0 s	0 s	178.61 s	88.16%
SECTRACKER	96.90%	84.90%	61.41 fps	9.51 s	1.10 s	0.2 s	10.81 s	96.64%

TABLE IV
COMPARISON WITH STATE-OF-THE-ART SYSTEMS ON SF-UAV FOR TRE AND SRE EVALUATION

Mechanism	TRE tracking precision rate	TRE success rate	TRE tracking speed	SRE precision rate	SRE success rate	SRE tracking speed	TRE overall accuracy	SRE overall accuracy
CCOT	96.20%	69.10%	0.91 fps	96.10%	72.20%	0.92 fps	86.96%	86.88%
BACF	95.40%	65.00%	68.99 fps	95.70%	66.40%	65.23 fps	86.32%	86.56%
AutoTrack	95.60%	70.70%	51.80 fps	95.30%	64.50%	46.96 fps	86.47%	86.23%
ECO	97.30%	78.60%	3.31 fps	96.50%	70.90%	3.33 fps	87.83%	87.20%
SECTRACKER	97.10%	81.40%	62.45 fps	96.10%	75.90%	53.39 fps	96.84%	95.84%

majority of nodes are benign, local voting can ensure the accurate discovery of the malicious nodes. Secondly, according to the DDH assumption, $K_{i,j}$ is computationally indistinguishable with a uniformly chosen group element for any probabilistic polynomial time attacker. When the message $M_{i\Rightarrow j}$ is transmitting in SAGIN, only the two pairs of nodes n_i and n_j keep the generated key $K_{i,j}$. If the attacker aims to forge the message $M_{i\Rightarrow j}$, even the tuple $m_{i\Rightarrow j}$ is leaked, it is of no possibility to generate the correct $Sig(K_{i,j}, H(m_{i\Rightarrow j}))$ without knowing the shared key $K_{i,j}$ and the spoofing communication will be terminated. From this sense, the message sharing process is secure under the protection of the signature algorithms. Lastly, the terrestrial station and MCS are believed to be benign which is described in our threat model. So both the message and the evidence are secure after uploading to the terrestrial station and MCS. Besides, both the message sharing and the evidence uploading are under the protection of signature.

B. Discussions

Dataset: Because of the limitations of the hardware configurations in the experiments, the dataset we constructed still needs to be extended in the range of spatial and temporal. Besides, the performance of SECTRACKER on the other existing dataset is still under exploration. However, we still believe the performance of SECTRACKER is effective in other potential datasets as long as they have similar experimental configurations to us.

Weather: In this study, because of the limitation of the equipment and the regulation of UAVs, it is hard to capture the videos of UAVs in extreme weather. The performance of SECTRACKER under such environments (e.g., the rainy day, the windy day) in the different time periods of the daytime (e.g., sunrise, sunset) is left for further work.

Other possible attack types: In this study, we only consider two main security issues named the message spoofing attack and the routing misbehavior attack. However, other security threats still need to be explored in further research. For instance, the GPS jamming and spoofing, the obfuscation of the sensors [46], and other communication security issues [5].

Other issues: In this study, we mainly focus on the issue of single target tracking. Since multiple UAVs should be tracked simultaneously, it should be explored for the multiple targets tracking to meet the requirements of military and civilian purposes. Besides, the performance of SECTRACKER when an emergency occurs (e.g., the forced landing, escaping from the range of terrestrial stations) should be considered.

VII. CONCLUSION

In this study, we propose SECTRACKER, which is a secure and efficient UAV tracking system in the space-air-ground integrated network (SAGIN). To achieve the design goal of security, power efficiency, and accuracy, we implement the Local Voting based Detection Module and Routing Evidence based Detection Module as countermeasures to defend against the message spoofing attack and the routing misbehavior attack. We further implement a novel UAV Tracking Module to improve the tracking performance. Experimental results show that when there exist the aforementioned attacks, SECTRACKER realizes the accuracy of 96.64% in the speed of 61.41 fps with all the proposed optimized mechanisms implemented, which is superior to the existing works in UAV tracking.

REFERENCES

[1] J. Yu, X. Liu, Y. Gao, and X. Shen, "3D channel tracking for UAV-satellite communications in space-air-ground integrated networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 12, pp. 2810–2823, Dec. 2020.
[2] Y. Kawamoto, H. Nishiyama, N. Kato, and N. Kadowaki, "A traffic distribution technique to minimize packet delivery delay in multilayered satellite networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 7, pp. 3315–3324, Sep. 2013.
[3] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 472–488.
[4] H. Kiani Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1135–1143.
[5] A. Fotouhi et al., "Survey on UAV cellular communications: Practical aspects, standardization advancements, regulation, and security challenges," *IEEE Commun. Surv. Tut.*, vol. 21, no. 4, pp. 3417–3442, Fourthquarter 2019. .

[6] S. Zhang, "Object tracking in unmanned aerial vehicle (UAV) videos using a combined approach," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2005, pp. ii/681–ii/684.
[7] S. Gladh, M. Danelljan, F. S. Khan, and M. Felsberg, "Deep motion features for visual tracking," in *Proc. IEEE 23rd Int. Conf. Pattern Recognit.*, 2016, pp. 1243–1248.
[8] T. Ma et al., "UAV-LEO integrated backbone: A ubiquitous data collection approach for B5G internet of remote things networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 11, pp. 3491–3505, Nov. 2021.
[9] Y. Fan, Y. Jiang, H. Zhu, and X. Shen, "An efficient privacy-preserving scheme against traffic analysis attacks in network coding," in *Proc. IEEE INFOCOM Conf.*, 2009, pp. 2213–2221.
[10] H. Zhu, R. Lu, X. Shen, and X. Lin, "Security in service-oriented vehicular networks," *IEEE Wireless Commun.*, vol. 16, no. 4, pp. 16–22, Aug. 2009.
[11] V. P. Hubenko, R. A. Raines, R. F. Mills, R. O. Baldwin, B. E. Mullins, and M. R. Grimaila, "Improving the global information grid's performance through satellite communications layer enhancements," *IEEE Commun. Mag.*, vol. 44, no. 11, pp. 66–72, Nov. 2006.
[12] M. Hamdi, N. Boudriga, and M. S. Obaidat, "Bandwidth-effective design of a satellite-based hybrid wireless sensor network for mobile target detection and tracking," *IEEE Syst. J.*, vol. 2, no. 1, pp. 74–82, Mar. 2008.
[13] N. Kato et al., "Optimizing space-air-ground integrated networks by artificial intelligence," *IEEE Wireless Commun.*, vol. 26, no. 4, pp. 140–147, Aug. 2019.
[14] N. Cheng et al., "A comprehensive simulation platform for space-air-ground integrated network," *IEEE Wireless Commun.*, vol. 27, no. 1, pp. 178–185, Feb. 2020.
[15] J. Li, W. Shi, H. Wu, S. Zhang, and X. Shen, "Cost-aware dynamic SFC mapping and scheduling in SDN/NFV-enabled space–air–ground-integrated networks for internet of vehicles," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 5824–5838, Apr. 2022.
[16] H. Guo, J. Li, J. Liu, N. Tian, and N. Kato, "A survey on space-air-ground-sea integrated network security in 6G," *IEEE Commun. Surv. Tut.*, vol. 24, no. 1, pp. 53–87, Firstquarter 2022.
[17] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-air-ground integrated network: A survey," *IEEE Commun. Surv. Tut.*, vol. 20, no. 4, pp. 2714–2741, Fourthquarter 2018.
[18] G. Ciaparrone, F. Luque Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera, "Deep learning in video multi-object tracking: A survey," *Neurocomputing*, vol. 381, pp. 61–88, 2020.
[19] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6638–6646.
[20] X. Lu, L. Xiao, C. Dai, and H. Dai, "UAV-aided cellular communications with deep reinforcement learning against jamming," *IEEE Wireless Commun.*, vol. 27, no. 4, pp. 48–53, Aug. 2020.
[21] K. Jansen, M. Schäfer, D. Moser, V. Lenders, C. Pöpper, and J. Schmitt, "Crowd-GPS-sec: Leveraging crowdsourcing to detect and localize GPS spoofing attacks," in *Proc. IEEE Symp. Secur. Privacy*, 2018, pp. 1018–1031.
[22] Y. Xiang et al., "Congestion attack detection in intelligent traffic signal system: Combining empirical and analytical methods," *Secur. Commun. Netw.*, vol. 2021, pp. 1–17, 2021.
[23] Z. Li, B. Yang, X. Zhang, and C. Guo, "DDoS defense method in software-defined space-air-ground network from dynamic Bayesian game perspective," *Secur. Commun. Netw.*, vol. 2022, pp. 1–13, 2022.
[24] Y. Ouyang, W. Liu, Q. Yang, X. Mao, and F. Li, "Trust based task offloading scheme in UAV-enhanced edge computing network," *Peer-to-Peer Netw. Appl.*, vol. 14, no. 5, pp. 3268–3290, 2021.
[25] H. Guo, X. Zhou, Y. Wang, and J. Liu, "Achieve load balancing in multi-UAV edge computing IoT networks: A dynamic entry and exit mechanism," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 18725–18736, Oct. 2022.
[26] H. Guo, W. Huang, J. Liu, and Y. Wang, "Inter-server collaborative federated learning for ultra-dense edge computing," *IEEE Trans. Wireless Commun.*, vol. 21, no. 7, pp. 5191–5203, Jul. 2022.
[27] H. Zhu, S. Du, Z. Gao, M. Dong, and Z. Cao, "A probabilistic misbehavior detection scheme toward efficient trust establishment in delay-tolerant networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 1, pp. 22–32, Jan. 2014.
[28] J. Wang and J. Liu, "Location hijacking attack in software-defined space–air–ground-integrated vehicular network," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 5971–5981, Apr. 2022.
[29] Y. Li, C. Fu, F. Ding, Z. Huang, and G. Lu, "AutoTrack: Towards high-performance visual tracking for UAV with automatic spatio-temporal regularization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11923–11932.
[30] P. Zhang, C. Wang, N. Kumar, and L. Liu, "Space-air-ground integrated multi-domain network resource orchestration based on virtual network architecture: A DRL method," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2798–2808, Mar. 2022.
[31] F. Tang, H. Hofner, N. Kato, K. Kaneko, Y. Yamashita, and M. Hangai, "A deep reinforcement learning-based dynamic traffic offloading in space-air-ground integrated networks (SAGIN)," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 276–289, Jan. 2022.
[32] "The blob detection algorithm in opencv libraries," 2023. [Online]. Available: https://opencv.org/
[33] R. Chen, S. Han, J. Xu, and H. Su, "Point-based multi-view stereo network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 1538–1547.
[34] H. Fan, H. Su, and L. Guibas, "A point set generation network for 3D object reconstruction from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2463–2471.
[35] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.
[36] H. Zhu, X. Lin, R. Lu, Y. Fan, and X. Shen, "SMART: A secure multilayer credit-based incentive scheme for delay-tolerant networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 8, pp. 4628–4639, Oct. 2009.
[37] H. Zhu, X. Lin, R. Lu, X. Shen, D. Xing, and Z. Cao, "An opportunistic batch bundle authentication scheme for energy constrained DTNs," in *Proc. IEEE INFOCOM Conf.*, 2010, pp. 1–9.
[38] S. M. Tornell, C. T. Calafate, J.-C. Cano, and P. Manzoni, "DTN protocols for vehicular networks: An application oriented overview," *IEEE Commun. Surv. Tut.*, vol. 17, no. 2, pp. 868–887, Secondquarter 2015.
[39] M. Felegyhazi, J.-P. Hubaux, and L. Buttyan, "Nash equilibria of packet forwarding strategies in wireless ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 5, pp. 463–476, May 2006.
[40] W. Gao and G. Cao, "User-centric data dissemination in disruption tolerant networks," in *Proc. IEEE INFOCOM Conf.*, 2011, pp. 3119–3127.
[41] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation," in *Proc. 2nd Int. Conf. Simul. Tools Techn.*, 2009, pp. 1–10.
[42] M. N. Mueller Smith and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 445–461.
[43] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 886–893.
[44] N. C. Fernandes, M. D. D. Moreira, and O. C. M. B. Duarte, "An efficient and robust addressing protocol for node autoconfiguration in ad hoc networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 3, pp. 845–856, Jun. 2013.
[45] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.
[46] Y. Zhi, Z. Fu, X. Sun, and J. Yu, "Security and privacy issues of UAV: A survey," *Mobile Netw. Appl.*, vol. 25, no. 1, pp. 95–101, 2020.

Jiachun Li (Student Member, IEEE) received the B.S. degree in communication engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2020. He is currently working toward the Ph.D. degree with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. His research interests include smart home security and smart healthcare security.

Weijiong Zhang received the Ph.D. degree from the School of Electronic, Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2010. He is currently a Research Fellow with Shanghai Electro-Mechanical Engineering Institute, Shanghai. His research interests include wireless Ad-Hoc network and networks security management.

Yan Meng (Member, IEEE) received the B.S. degree in electronic and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2016, and the Ph.D. degree in computer science and technology from Shanghai Jiao Tong University, Shanghai, China, in 2021. He is currently a Research Assistant Professor with Shanghai Jiao Tong University. He authored or coauthored more than 20 papers, including IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING,IEEE TRANSACTIONS ON MOBILE COMPUTING,IEEE WIRELESS COMMUNICATIONS, ACM CCS, and USENIX Security. His research interests include wireless network security and IoT security. He was the recipient of the 2022 ACM SIGSAC China Doctroal Dissertation Award.

Shaofeng Li is currently a Postdoctoral Researcher with Frontier Research Center, Peng Cheng Laboratory, Shenzhen, China. His research interests include machine learning and security, specifically exploring the robustness of machine learning models against various adversarial attacks. He was the recipient of the ACM CCS Best Paper Award Runner-Up for his work.

Lichuan Ma (Member, IEEE) received the B.S. degrees in information security from the School of Mathematics, Shandong University, Jinan, China, in 2012, and the Ph.D. degree in information security from Xidian University, Xi'an, China, in 2018. He is currently with the School of Cyber Engineering, Xidian University, and a Member of the State Key Laboratory of Integrated Services Networks. His research interests include trust management and privacy-preserving techniques for intelligent systems.

Zhen Liu (Member, IEEE) received the Ph.D. degrees in computer science from the City University of Hong Kong, Hong Kong, and Shanghai Jiao Tong University, Shanghai, China, in 2013. He is currently an Associate Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His research interests include applied cryptography, studying provable security and designing cryptographic primitives, such as encryption and signature schemes, for the research problems motivated by practical applications.

Haojin Zhu (Fellow, IEEE) received the B.Sc. degree in computer science from Wuhan University, Wuhan, China, in 2002, and the M.Sc. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2005, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2009. He is currently a Professor with the Department of Computer Science, Shanghai Jiao Tong University. He authored or coauthored more than 70 international journal papers, including JSAC, TDSC, TPDS, TMC, TIFS, and 90 international conference papers, including IEEE SECURITY PRIVACY, ACM CCS, USENIX Security, NDSS, ACM MOBICOM. His research interests include IoT security and privacy enhancing technologies. He was the recipient of a number of awards including: IEEE VTS Distinguished Lecturer in 2022 ACM CCS Best Paper Runner-Ups Award in 2021, IEEE TCSC Award for Excellence in Scalable Computing, Middle Career Researcher, 2020, IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award in 2014, Top 100 Most Cited Chinese Papers Published in International Journals in 2014, Distinguished Member of the IEEE INFOCOM Technical Program Committee in 2015, 2020, best paper awards of IEEE ICC in 2007 and Chinacom in 2008, WASA Best Paper Runner-up Award in 2017. He is serving the Editorial Board for IEEE TRANSACTION ON WIRELESS COMMUNICATIONSand program committees for top conferences, such as USENIX Security, ACM CCS, NDSS, and IEEE INFOCOM.