Received 6 January 2024; revised 1 February 2024; accepted 18 February 2024. Date of publication 21 February 2024; date of current version 19 March 2024.

Digital Object Identifier 10.1109/OJCOMS.2024.3368275

An Efficient and Dependable UAV-Assisted Code Dissemination in 5G-Enabled Industrial IoT

RAVI SHARMA (Member, IEEE), AND BALÁZS VILLÁNYI (Member, IEEE)

Department of Electronics Technology, Budapest University of Technology and Economics, 1111 Budapest, Hungary

CORRESPONDING AUTHOR: R. SHARMA (e-mail: ravi.sharma@edu.bme.hu)

This work was supported in part by the National Research Development and Innovation Office - Hungary (NKFIH) under Project K 145966.

ABSTRACT The Industrial Internet of Things (IIoT) has infiltrated our culture and is gaining traction in a variety of industrial applications. All of this is made possible by the use of 5G-connected massive Intelligent Sensing Devices (ISDs) and software-defined technology. It is sometimes desirable, and even required, to upgrade these ISDs without replacing hardware to make them smarter by adding new features and/or removing bugs through code dissemination. Using moving vehicles of the 5G-enabled Internet of Vehicles (IoV) infrastructure is one possible and efficient way to disseminate code. Specifically, safe code dissemination through a large number of vehicles in a 5G network has emerged as a critical issue. The motivation stems from the limitations of existing methods, often centralized and vulnerable to compromise, particularly in semi-connected networks. Therefore, this paper proposes an efficient and reliable Unmanned Aerial Vehicle (UAV)-assisted digital signature-based safe code dissemination framework for a 5G-enabled IIoT system. Our decentralized approach, which uses digital signatures and a Subjective Logic model, not only ensures code integrity but also identifies credible code mules, avoiding the pitfalls of traditional trust evaluation schemes. Furthermore, for UAV trajectory optimization, we redesign the trajectory with virtual waypoints to shorten the trajectory path for upgrading ISDs that were not upgraded due to the long-time trailing phenomenon. Our approach is useful in scenarios such as smart cities, where ISDs lack communication facilities. Through extensive experiments, our framework demonstrates superior efficiency and reliability compared to state-of-the-art methods.

INDEX TERMS Industrial Internet of Things, code dissemination, trust evaluation, 5G network, trajectory optimization.

I. INTRODUCTION

THE RAPID advancement of microprocessor technology has made Intelligent Sensing Devices (ISDs) more robust by increasing computing and storage capacity while decreasing in size [1]. These ISDs, with their intelligent sensing and communication capabilities, are the most important Industrial Internet of Things (IIoT) components [2]. By 2025, it is expected that approximately 22 billion ISDs generating up to 2.5 quintillion bytes of data per day will be connected to 5G technology [3]. They take a novel approach to solving complex sensing problems in areas such as smart cities, intelligent transportation, industrial monitoring, and defense [4]. Different sensor types, such as motion or daylight sensors for street lights; temperature, humidity, and wind sensors for weather prediction; and height sensors for smart garbage cans, are used in various scenarios to build a better ecosystem [4]. All of these capabilities and functionalities are combined to generate smart data, which is then used by an artificial intelligence (AI) algorithm to create an intelligent system [5].

Nowadays, the evolution of AI is changing by the day, and new AI algorithms are being introduced almost every now and then [5]. Therefore, it is sometimes desirable, and even required, to upgrade these ISDs without replacing hardware to make them smarter by adding new features and/or removing bugs through code dissemination. With the advent of software-defined technologies, these ISDs can now upgrade their firmware [6]. It enables deployed ISDs to have new functions and adapt to the new environment through software rather than hardware, avoiding the high costs associated with hardware updates [7]. However, disseminating the code to ISDs in a smart city is a difficult issue because many ISDs

are deployed without basic communication facilities, and their location changes frequently, making communication with the code center difficult [6]. All these applications can withstand delays and do not require real-time data reporting from the data center to ISDs; additionally, upgraded ISDs are backward compatible with non-upgraded ones [8]. During the code dissemination process, ISDs running old and new codes coexist, and all ISDs in the smart city do not need to be upgraded at the same time. However, the coexistence time may vary depending on the application requirements [9].

Many studies on code dissemination have been conducted, and the majority of them use direct communication links connecting ISDs to code centers, which is a relatively simple method [9], [10]. However, a significant proportion of ISDs in semi-connected networks that are unable to interact directly with the code center face a critical situation in IIoT applications [9], [10]. The most cost-effective and efficient way to disseminate code is through the 5G-enabled Internet of Vehicles (IoV) [6], [9]. Specifically, safe code dissemination through a large number of vehicles in a 5G network has emerged as a critical issue [6]. The security of code dissemination in semi-connected networks differs significantly from that of fully connected networks, where a simple encryption technique is sufficient to protect code dissemination [11], [12]. The use of vehicles as code mules for code dissemination in a semi-connected network can result in the attack models listed below.

- *Selective forwarding attack*: In this attack, malicious code mules attempt to abandon some or all of the code data while selectively discarding a portion of the valuable code, resulting in a failure to receive an integrated code on time [13].
- *Data pollution attack*: In this attack, malicious code mules can disrupt normal code dissemination by tampering, forging, and replaying code data [14].
- *Man-in-the-middle attack*: In this attack, malicious code mules act as intermediaries, tampering with and falsifying code to control ISDs in a completely transparent manner to both the code center and the ISDs [5].
- *On-off attack*: This attack has two stages: on and off; during the on stage, malicious code mules imitate normal behaviors to gain high trust levels, then switch to the off stage to launch malicious attacks [13].
- *Bad/good mouth attack*: In a bad mouth attack, malicious code mules slander credible code mules with low trust levels. During the good mouth attack, the malicious code mules work together to boost their trust levels far above the credible code mules [4].

These malicious attacks prevent ISDs from receiving the code they genuinely require in a timely manner by discarding or falsifying data and performing misleading trust evaluations. It will result in application failure or possibly system paralysis, which will have a significant impact on the IIoT system and necessitate secure code dissemination [15].

Building upon this context, our paper extends the exploration to the realm of ISDs in IIoT and their interaction with UAVs. There is no denying that UAVs have attained significant focus and research in recent years, and with 5G already being deployed, UAVs can now exploit the capabilities of the new networks [16]. The unique features of UAVs, such as movement flexibility and efficient code dissemination capabilities, become crucial in addressing the challenges posed by the upgrading ISDs that were not upgraded due to the long-time trailing phenomenon. By leveraging the capabilities of UAVs in efficiently disseminating code, we redesign the trajectory with virtual waypoints to shorten the path. This enhances the security and reliability of code dissemination in semi-connected networks, providing a holistic solution to the challenges posed by the dynamic and diverse nature of IIoT applications [17].

A. MOTIVATION AND CHALLENGES

Creating a trust network and disseminating code through trusted code mules is a well-studied method for ensuring code integrity, but it is a difficult task [18], [19]. Code integrity can be ensured, in particular, by disseminating code through credible code mules. Direct and indirect comprehensive evaluation methods, reporting evaluation rating, and third-party evaluation of object interactive behavior have all contributed to the development of trust [11], [20]. Many trust evaluation schemes have been proposed that use trust establishment based on object behavior, but these schemes have many flaws and are not suitable for semi-connected networks [6], [21], [22]. Firstly, with exponentially growing ISDs, on-time behavior monitoring is impractical and necessitates more resources and facilities. Secondly, semi-connected networks face a significant challenge when it comes to effectively conveying interactive behaviors, primarily due to their limited capacity for direct communication links with the code center. Because all of these trust evaluation methods are black boxes that establish trust based on object behavior rather than content integrity, the results of their evaluations cannot be verified [14], [23].

B. OUR CONTRIBUTIONS

In light of the shortcomings identified in previous studies, such as unreliability of code integrity, cost inefficiency, impractical and difficult trust calculation, inaccurate and unverifiable trust evaluation and so on, we propose an efficient and reliable UAV-assisted digital signature-based safe code dissemination framework in a 5G-enabled IIoT system. The following are the main contributions of our work:

- A UAV-assisted Digital signature-based safe Code Dissemination (UDCD) framework is designed, which uses the digital signature to verify the integrity of disseminated code in a decentralized manner appropriate for a semi-connected network.
- We propose a verifiable trust evaluation scheme based on digital signature verification to identify credible code mules using the Subjective Logic model.

- In addition, we propose a virtual waypoint-based UAV-trajectory optimization method for upgrading non-upgraded ISDs to reduce UAV-trajectory length.
- Finally, extensive experiments on a real-world dataset show that the proposed code dissemination scheme is more efficient and reliable than existing similar schemes.

C. ORGANIZATION

The remainder of this paper is structured as follows. Section II contains a list of related works. The system model and problem formulation are presented in Section III. Section IV describes the proposed UDCD trust evaluation method in detail, and Section IV-C describes the UAV-trajectory optimization technique. Section V provides and discusses the experimental setup and performance evaluation. Section VI summarizes and concludes the paper.

II. RELATED WORK

IoV infrastructure has proven to be the most efficient and cost-effective method of disseminating code via vehicles acting as code mules in a semi-connected 5G-enabled IIoT system over the last few decades [6], [9]. However, with the rapid growth of IIoT scale and large ISDs, management and control of these vehicles, as well as the integrity of the disseminated code, have become critical issues [6], [8], [9]. As previously discussed, malicious code mules can disseminate destructive code, causing significant harm to IIoT applications. Therefore, identifying and eliminating malicious code mules involved in code dissemination, as well as ensuring the integrity of the disseminated code, have become important research objectives.

In general, many studies have been conducted to increase the coverage ratio by using vehicles as code mules [24], [25], [26], [27]. Clustering and probabilistic broadcasting are used in [27] to create a reliable and stable communication scheme based on multi-vehicle communications. Although this scheme improves coverage ratios during vehicle-to-vehicle (V2V) communications, it is incapable of detecting malicious vehicles, which could lead to code insecurity. Zhou et al. [24] investigate the use of physical and social layer information in distributed networks to increase code dissemination. An iterative matching algorithm based on price rising is used to solve the joint peer discovery problem. Although it improved efficiency, it did not address the security issue, which resulted in malicious vehicles.

Reis et al. [25] present a scheme that uses parked vehicles to improve data dissemination coverage ratios through realistic modeling of mobility, parking, and communication. The findings indicated that parked vehicles could be used instead of fixed units, but they did not address data security. Inspired by parked vehicles as an alternative to fixed units, Li et al. [26] propose a Machine Learning-based Code Dissemination (MLCD) Scheme in 5G Networks that employs high-reliability vehicles as code disseminators. Although the results show that this scheme improves coverage ratio and safety degree, it does not guarantee code integrity and requires a significant amount of safety degree calculation to maintain a large historical GPS collection dataset.

In recent years, trust evaluation methods have grown in popularity as an effective means of identifying malicious code mules for reliable code dissemination [19], [20], [21], [22]. The earliest methods of evaluating trust were evaluation rating methods, but they are no longer considered reliable due to malicious ratings provided by untrustworthy users. Some researchers proposed comprehensive trust in the integration of direct trust through direct interaction to evaluate interactive objects; however, because direct interaction is not possible between some objects, trust is obtained through trust links, which is known as indirect trust [19], [20]. Although these methods are more advanced than evaluation rating methods [11], [20], which rely heavily on the reliability of trusted links and are not suitable for semi-connected networks. Even if a third-party UAV is used to observe and report on behavior in real-time, it is expensive and difficult to implement in a 5G-enabled IIoT system with large ISDs [11].

Despite their limitations, the above methods are unsuitable for a semi-connected network because they do not guarantee data integrity and do not support direct communication with code centers [9], [10]. Trust discovery has recently advanced at a rapid pace, prompting us to conduct this study. The methods used in [21], [22] rely on receiving reports to discover the trust, but verifying the authenticity of received reports is not guaranteed and is incorrect in the case of malicious reporters. The UAV-assisted Trustworthy Code Dissemination (UTCD) scheme [6] proposed a solution to the aforementioned problem by implementing a trustworthy scheme that uses UAV to collect code from specific ISDs and verify this code in a centralized manner to identify the credible and malicious code mules, thereby increasing UAV flight length. Later on, credible code mules join to assist with code collection for verification. Because the UTCD scheme does not perform a specific trust evaluation after identifying mobile vehicles as credible code mules, hijacked vehicles may compromise code integrity by modifying it, and hijacked devices may jeopardize security. Table 1 compares various types of code dissemination schemes in detail.

In summary, the majority of existing methods either do not verify the integrity of disseminated code or use a centralized verification system to discover trust. Because none of the methods took into account compromised vehicles at a later stage, their methods were either potentially compromised or required complex calculations to determine truth, which was both expensive and ineffective. As a result, we propose an efficient and dependable UDCD scheme for ensuring code integrity and valid trust discovery in a decentralized manner. Although some studies have suggested using the vehicle's joint UAV to disseminate code, determining the best and most optimized UAV flight trajectory is still an ongoing project [6], [11]. For which, we proposed a

TABLE 1. Comparison of different types of code dissemination schemes.

Scheme	Methodology	Trust evaluation	Compromised vehicle	Security risk	Structure
UTCD [6]	Direct and Indirect Trust	Yes	No	Moderate	Centralised
VCMCD [9]	Opportunistic communication	No	No	Severe	Centralised
D2D-V2V [24]	Iterative matching algorithm	No	No	Severe	Centralised
PC-RCU [25]	Realistic modelling	No	No	Severe	Centralised
MLCD [26]	Machine Learning	Yes	No	Moderate	Centralised
UDCD [ours]	Digital Signature-based Subjective Logic	Yes	Yes	Low	Decentralised

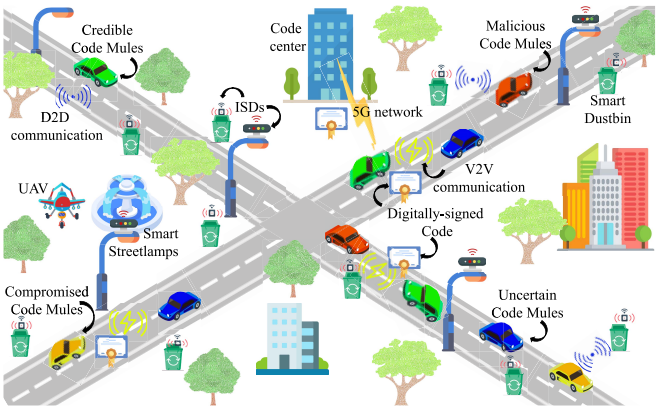

FIGURE 1. The UDCD framework.

virtual waypoint-based UAV-trajectory optimization method to upgrade non-upgraded ISDs and shorten the trajectory path.

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. SYSTEM MODEL

The 5G-enabled smart city model used in this paper is a modified version of [6], and it consists of three components, which are listed below (see Figure 1). Table 2 shows the list of notations.

1) CODE CENTER

The code center receives the upgrade code from the cloud data center using the 5G network and is in charge of distributing it to ISDs using code mules. The code center is responsible for disseminating the code to ISDs in the smart city in an efficient and secure manner. It also keeps track of all the upgraded and non-upgraded ISDs, as well as the various types of code mules in the network.

2) ISDS

ISDs are strategically placed throughout the city to provide smart monitoring and perception of various objects. These ISDs can adjust various control parameters by self-upgrading their firmware with the disseminated code to meet the needs of various applications. To achieve the functionality

TABLE 2. Notations.

Notations	Description
$\{\mathcal{I}_i\}_{i=1}^{n}$	Set of ISDs
$\{\mathcal{M}_i\}_{i=1}^{l}$	Set of code mules
$\widehat{\mathcal{I}}_i$	Upgraded ISDs
$\widehat{\mathcal{I}}_i^{\mathrm{T}}$	True upgraded ISDs
$\widehat{\mathcal{I}}_i^{\mathrm{F}}$	False upgraded ISDs
$d_{i,j}$	Distance between two consecutive waypoints w_i, w_j
$\mathcal{M}_i^{\mathrm{C}}$	Credible code mules
$\mathcal{M}_i^{\mathrm{M}}$	Malicious code mules
$\mathcal{M}_i^{\mathrm{U}}$	Uncertain code mules
$\Omega_{\mathcal{M}_i}$	Recommendation value of $\mathcal{M}_i$
δ	Recommendation reliability
σ	Digitally signed code data using SEMECS
β	Digital signature verification result, either pass or fail
$\Omega_{\mathcal{M}_i,j}$	Recommended value for code mule $\mathcal{M}_i$ given by the j-th recommender
$\Psi_{\mathcal{M}_i}$	Trust value of $\mathcal{M}_i$
$\|\cdot\|$	Euclidean norm
$\mathcal{Q}$	Ordered set of waypoints
$\widehat{\mathcal{Q}}$	Ordered set of virtual waypoints
$\|\cdot\|$	Cardinality of set
$\mathcal{C}_i$	A collection of clusters containing all possible UAV-trajectory locations.

described above, we assume that the proposed model has n ISDs denoted as $\mathcal{I} = \{\mathcal{I}_i\}_{i=1}^{n}$.

3) CODE MULES

Code mules are in charge of transmitting code from the code center to ISDs. Taking advantage of 5G-enabled IoV infrastructure, we use GPS and wireless communication-equipped vehicles as code mules with no modifications. We use l code mules, denoted as $\mathcal{M} = \{\mathcal{M}_i\}_{i=1}^{l}$, for opportunistic communication at no or minimal cost while

not interfering with their regular work. Although we use the "pay less and cover more" policy, we do not consider the cost of code dissemination in this paper for the sake of simplicity.

B. PROBLEM FORMULATION

Our primary objective is to disseminate code with high coverage in a short period while maintaining code integrity. Similar to [6], this study uses joint vehicles and UAV to disseminate code in semi-connected networks of 5G-enabled IIoT. To be more specific, the following performance indicators are being used:

1) COVERAGE RATIO (CR)

Due to uneven infrastructure in the smart city, downtown areas are more likely to receive code than outlying areas. Therefore, the combination of vehicle coverage and UAV trajectory results in a significantly improved coverage ratio.

$$\mathrm{CR} = \frac{\left|\bigcup_{i\in\mathcal{I}} \widehat{\mathcal{I}}_i\right|}{n} \tag{1}$$

where the upgraded ISDs are represented by $\widehat{\mathcal{I}}_i$.

2) DISSEMINATION RATE (DR)

The effectiveness of reliable code dissemination is highly dependent on the creation of true dissemination, which has a positive impact on the overall performance of the system integrity.

$$\mathrm{DR} = \frac{\left|\bigcup_{i\in\mathcal{I}} \widehat{\mathcal{I}}_i^{\mathrm{T}}\right|}{\left|\bigcup_{i\in\mathcal{I}} \widehat{\mathcal{I}}_i^{\mathrm{T}}\right| + \left|\bigcup_{i\in\mathcal{I}} \widehat{\mathcal{I}}_i^{\mathrm{F}}\right|} \tag{2}$$

where $\widehat{\mathcal{I}}_i^{\mathrm{T}}$ refers to ISDs that received legitimate code, and $\widehat{\mathcal{I}}_i^{\mathrm{F}}$ refers to ISDs that received malicious code.

3) UAV-TRAJECTORY LENGTH (UL)

Due to the trailing phenomenon, some ISDs are unable to obtain the code using only vehicles; thus, we are also using UAVs to cover the entire geographic area of a smart city. Because UAVs have limited energy, their flying distance is also limited, necessitating the use of an optimized UAV-trajectory length.

$$\mathrm{UL} = \sum_{i,j\subseteq \mathrm{route}} d_{i,j} \tag{3}$$

where $d_{i,j}$ denotes the distance between two consecutive ISDs' waypoints w_i and w_j.

4) RELIABILITY PERCENTAGE (RP)

Identifying and removing malicious code mules from the system is an effective method for improving overall system integrity.

$$\mathrm{RP} = \frac{\left|\bigcup_{i\in\mathcal{M}} \mathcal{M}_i^{\mathrm{C}}\right|}{\left|\bigcup_{i\in\mathcal{M}} \mathcal{M}_i^{\mathrm{C}}\right| + \left|\bigcup_{i\in\mathcal{M}} \mathcal{M}_i^{\mathrm{M}}\right|} \tag{4}$$

where $\mathcal{M}_i^{\mathrm{C}}$ denotes credible code mules, and $\mathcal{M}_i^{\mathrm{M}}$ denotes malicious code mules.

Taking the performance indicators mentioned above into account, the proposed UDCD scheme aims to achieve an improved coverage ratio, effective dissemination rate, optimized UAV-trajectory length, and a high reliability percentage.

IV. DESIGN OF UDCD SCHEME

We need a large number of ISDs to build a better smart city ecosystem, but these ISDs must be upgraded over time to meet new functionality. These ISDs are mostly located along roadsides and have limited communication capabilities due to their low cost. Managing and upgrading code to these ISDs becomes difficult in a semi-connected network. One possible cost-effective and efficient solution is to use a large number of vehicles (code mules) to disseminate code throughout the city [6], [9]. The integrity of this newly upgraded system, on the other hand, is solely dependent on the credibility of code and the trust of code mules. As a result, we have proposed the UDCD scheme to ensure that only trusted code mules participate in code dissemination. To address the issue that some ISDs are unable to obtain the code using only vehicles, we used UAVs to disseminate the code and proposed an optimized UAV flight trajectory algorithm.

A. OVERVIEW OF UDCD SCHEME

We assume that the code center has private/public keys (sk, PK), and each $\mathcal{M}_i$ is preconfigured with the code center's public key, which can later be exchanged with trusted entities. To better suit the IIoT environment, SEMECS[1] [28], an ultra-lightweight digital signature, is being used. SEMECS is suitable for embedded devices with limited resources because it achieves optimal signature and private key sizes for an Elliptic Curve (EC)-based signature without the need for any EC operations. Yavuz and Ozmen [28] fully implemented and tested SEMECS on an 8-bit AVR ATmega 2560 microprocessor, confirming up to 19× less battery consumption, 6× lower energy consumption, extremely fast signature generation (1.23 microseconds), and lightweight transmission, making it ideal for real-time implementation in industrial environments where all these factors are critical.

In our analysis, we assume that the upgrade code is small and can be transmitted all at once with a single transmission.

[1] https://github.com/ozgurozmen/SEMECS

In contrast, a large code size can be divided into fixed-size pages and transmitted using the Merkle hash tree [29]. As the entities in our communication system may join and leave over time, we use over-the-air periodic public key updates for the SEMECS digital signature. As soon as the code center announces the upgrade process, it distributes the digitally signed code (using SEMECS.Sig() function) to various credible code mules $\mathcal{M}_i$ using the 5G network.

$$\sigma \leftarrow \texttt{SEMECS.Sig}(sk,\text{CODE}) \tag{5}$$

For the first time, we assume that a few credible code mules already exist; this is not a necessary condition, but it is required to achieve good efficiency quickly. In addition to their routine information exchange, $\mathcal{M}_i$ exchanges this digitally signed code through V2V communication. $\mathcal{M}_i$ then validates this code using digital signature verification (using SEMECS.Ver() function) and sends a recommendation to the code center.

$$\beta \leftarrow \texttt{SEMECS.Ver}(PK, \sigma_i) \tag{6}$$

Later, the code center will use these recommendations to determine whether to remove as many malicious $\mathcal{M}_i$ from the network as possible to ensure the integrity of disseminated code. The UDCD scheme has four steps: firstly, code mules gather evidence by verifying the digital signature; secondly, trust is recommended for the code center; thirdly, trust is calculated based on recommendations; and finally, trust is updated to determine whether code mules are credible or not.

B. TRUST EVALUATION

1) EVIDENCE COLLECTION

The vehicle's prior behavior is the most important factor in determining trust in it. As vehicles join and leave the network, the trust factor becomes uncertain, and we use the Subjective Logic model [30] to deal with this uncertainty. The trust opinion in the Subjective Logic model can be expressed by the triplet vector T={C, M, U}, where C, M, and U represent the credibility, maliciousness, and uncertainty of code mules, respectively ($\texttt{C}, \texttt{M}, \texttt{U} \in [0, 1]; \texttt{C} + \texttt{M} + \texttt{U} = 1$). Uncertainty in the code mules are vehicles that are unsure of their trustworthiness. Following the trust model based on the Subjective Logic [30], [31], the recommendation value $\Omega_{\mathcal{M}_i}$ of the code mule $\mathcal{M}_i$ can be calculated using (7):

$$\Omega_{\mathcal{M}_i} = \texttt{C} + \gamma \cdot \texttt{U} \tag{7}$$

where γ is a constant that is by default set to 0.5 [32], indicating the effect of uncertainty on the trustworthiness of code mules, $\texttt{C} = \frac{p}{p+f+\omega}$, and $\texttt{U} = \frac{\omega}{p+f+\omega}$; p and f represent the number of pass and fail digital signature verification results. ω is a non-informative prior weight with a default value of 2 to ensure the uniform prior probability distribution function when $p = f = 0$, and $\gamma = 0.5$.

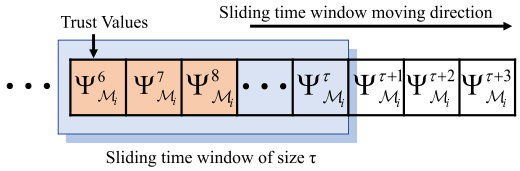

FIGURE 2. The sliding time window.

2) EVALUATION OF RECOMMENDATIONS

There is a possibility that some code mules are compromised and will try to influence the decision. As a result, after receiving multiple recommendations from various code mules, a code center must first determine the reliability of the recommendations. This can be accomplished by detecting outliers, such as comparing the consistency of multiple recommendations. However, comparing consistency is a time-consuming and inefficient task. Therefore, in this paper, we calculated the recommendation reliability δ among multiple recommendations by subtracting the recommended value from the mean value, as shown in (8):

$$\delta = 1 - \left|\Omega_{\mathcal{M}_i,j} - \overline{\Omega_{\mathcal{M}_i}}\right| \tag{8}$$

where $\overline{\Omega_{\mathcal{M}_i}}$ is the mean of the $\mathcal{M}_i$ recommendation values, and $\Omega_{\mathcal{M}_i,j}$ is the recommended value for code mule $\mathcal{M}_i$ given by the j-th recommender. The lower the difference is, the more reliable is the recommendation.

3) TRUST CALCULATION

Even a higher recommendation value becomes ineffective if $\delta < 0.5$ (below the threshold). For $\delta > 0.5$, the trust value $\Psi_{\mathcal{M}_i}$ of code mule $\mathcal{M}_i$ is calculated as given in (9):

$$\Psi_{\mathcal{M}_i} = \frac{\sum_{j=1}^{r} \delta \times \Omega_{\mathcal{M}_i,j}}{r} \tag{9}$$

where r is the number of recommenders.

4) TRUST UPDATE

The history records can be used as a reference for current trust evaluation as vehicles enter and exit the network over time. If the trust value is updated frequently, it consumes a lot of energy, and if it is too long, it cannot reflect current behavior efficiently. Hence, to address this issue, we use a sliding time window to store vehicle trust updates, as shown in Figure 2. The sliding time window with size τ stores a list of trust updates and advances in time from left to right. The historical record is cleared on a regular basis to save memory. The trust value is updated based on an event in the following cycle as given in (10):

$$\Psi^{\text{new}}_{\mathcal{M}_i} = \alpha \cdot \Psi^{\text{old}}_{\mathcal{M}_i} + (1 - \alpha) \cdot \Psi^{\text{new}}_{\mathcal{M}_i} \tag{10}$$

where $\Psi^{\text{new}}_{\mathcal{M}_i}$ and $\Psi^{\text{old}}_{\mathcal{M}_i}$ are the new and old trust values of code mule $\mathcal{M}_i$, and $\alpha = e^{\text{old}-\text{new}}$ is the aging factor in trust value attenuation, as the most recently computed historical trust value has greater importance than one computed a long time ago. To identify credible code mules, we compare the

Algorithm 1 Trust-Based Credible Code Mules Identification

Input: $\Psi_{\mathcal{M}_i}, \{\mathcal{M}_i\}, \{\mathcal{M}_i^{\mathrm{C}}\}$;
Output: $\{\mathcal{M}_i^{\mathrm{C}}\}$;

```
for each M_i do
    Calculate recommendation value Ω_{M_i} of the code mule M_i by eq. (7);
    for j = 1 to r do
        Calculate recommendation reliability δ using eq. (8);
        if δ > 0.5 then
            Calculate trust value Ψ_{M_i} using eq. (9);
        end if
    end for
    Update trust value using eq. (10);
end for
Calculate most credible code mules using eq. (11);
Add or remove M_i^{C*} from the set {M_i^C} to update it;
return {M_i^C};
```

trust values of the recommenders and calculate the optimal credible code mule values as given in (11):

$$\mathcal{M}_i^{\mathrm{C}*} = \arg\max_{i \in \mathcal{M}_i} (\Psi_{\mathcal{M}}) \tag{11}$$

Algorithm 1 shows the process of evaluating trust values and identifying credible code mules. Following the identification of credible code mules, it updates the set $\{\mathcal{M}_i^{\mathrm{C}}\}$, which is used as a code disseminator. Later on, a few code mules of different types will join the system and prove their credibility to join $\{\mathcal{M}_i^{\mathrm{C}}\}$.

C. UAV-TRAJECTORY OPTIMIZATION

UAV-trajectory optimization is another important aspect of achieving efficient code dissemination to ISDs that have not been upgraded due to the long-time trailing phenomenon. We assume the UAV travels in a straight line between two consecutive waypoints, and area a represents the projected trajectory of the UAV onto the ground. We also assume that the UAV will use appropriate combinations of code data size, flight speed, and flying altitude to reduce communication and computing latency [33], [34]. In the case of UDCD, the set of waypoints consists of a collection of non-upgraded ISDs. To address this issue, we have redesigned the trajectory with virtual waypoints to reduce UAV trajectory length. The UAV communication system typically operates on frequencies between 2.4 and 5.8 GHz and uses communication technologies such as WiMAX, GPRS, cellular, and so on to communicate with ISDs available in that region.

Let $i \in \mathcal{L}$ represent the coordinate of waypoint w_i of ISD $\mathcal{I}_i$ on the plane that needs to be upgraded, and each waypoint w_i could be the centroid of the area a covered by the UAV. For two locations $i, j \in \mathcal{L}$, let $d_{i,j} \propto \|w_i - w_j\|$ be a path covered by the UAV trajectory (i.e., the Euclidean distance of w_i, w_j), indicating the distance it takes for the UAV to move between two waypoints. The UAV creates a trajectory by visiting a subset of locations in a specific order. Consider an undirected complete graph $G = (\mathcal{L}, E, \boldsymbol{d})$, where $E = \{\{i, j\} : i, j \in \mathcal{L}, i \neq j\}$ is the set of links connecting the waypoints, and each link $\{i, j\}$ has a distance $d_{i,j}$ associated with it. A trajectory $\mathcal{Q}$ is a tour on the graph G, i.e., an ordered set of waypoints $\mathcal{Q} \triangleq (w_1, w_2, \ldots, w_m, w_1)$, in which the UAV visits each waypoint in the Hamiltonian cycle with the least weight in the specified order, except for w_1, which is the code station waypoint.

Flying over all the w_i for a given $a > 0$ is obviously unnecessary because the UAV may be connected to more than one w_i at the same time. Thus, the number of w_i visited by the UAV may be significantly less than m, especially when a is large and the w_i are densely distributed. Therefore, we create alternative waypoint designs based on the generation of virtual waypoints, which we refer to as $\tilde{w}_i \in \mathbb{R}^{2\times 1}, i = 1, \ldots, p$. Let $\mathcal{W} = \{w_i\}_{i=1}^{m}$ represent the set of original waypoints and $\widetilde{\mathcal{W}} = \{\tilde{w}_i\}_{i=1}^{p}$ represent the set of virtual waypoints.

Specifically, given w_i and the UAV coverage area a, the $\tilde{w}_i$ picking problem seeks to find the largest number of w_i such that each $\mathcal{W}$ is covered by at least one $\tilde{w}_i$. This is analogous to the standard base station placement of ensuring user coverage with a given coverage area a, which can be efficiently solved by the spiral base station placement algorithm like the one proposed in [35]. Let $\tilde{w}_i$ represent the minimum number of virtual waypoints obtained by using the spiral base station placement algorithm. To design an efficient route, the UAV must visit these $\widetilde{\mathcal{W}}$ sequentially by following the path generated by the Travelling Salesman Problem (TSP) algorithm over $\widetilde{\mathcal{W}}$, which is less than the original waypoints (i.e., $p < m$).

The $\mathcal{W}$ are essentially partitioned into p ordered clusters $\widetilde{\mathcal{W}}$ denoted as $\mathcal{C}_i$ using the spiral base station placement and TSP algorithm applied over $\mathcal{W}$, where all of the waypoints in $\mathcal{W}$ are covered by the $\widetilde{\mathcal{W}}$ when the spiral base station placement algorithm is used. We define the following set with w_i for the p^{th} ordered cluster using (12):

$$\mathcal{C}_i \triangleq \left\{ \|\tilde{w}_i - w_j\| \leq \frac{a}{2}; \forall \tilde{w}_i \in \widetilde{\mathcal{W}}, \forall w_j \in \mathcal{W} \right\} \tag{12}$$

In other words, $\mathcal{C}_i$ is the collection of all possible UAV-trajectory locations that ensure all non-upgraded ISDs waypoints in $\mathcal{W}$ are simultaneously connected to the UAV. It redesigns a new trajectory $\widehat{\mathcal{Q}}$, i.e., an ordered set of virtual waypoints $\widehat{\mathcal{Q}} \triangleq (\tilde{w}_1, \tilde{w}_2, \ldots, \tilde{w}_p, \tilde{w}_1)$. $\mathcal{C}_i$ is obviously non-empty because it contains w_i, and it is a convex set because it is an intersection of $|\widetilde{\mathcal{W}}|$ convex sets.

Without losing generality, consider $\widetilde{\mathcal{W}}$ to be the set of UAV-trajectory waypoints intersecting with the region $\mathcal{C}_i$. Because $\mathcal{C}_i$ is a convex set, all points of the trajectory path $\widehat{\mathcal{Q}}$ are also in $\mathcal{C}_i$, ensuring that all waypoints in $\mathcal{W}$ are in contact with the UAV. As a result, the waypoints in $\widehat{\mathcal{Q}}$ could be optimized by solving the following problem

Algorithm 2 UAV-Trajectory Optimization

Input: Waypoints $\mathcal{W} = \{w_i\}_{i=1}^{m}$;
Output: Optimised UAV-trajectory $\widehat{\mathcal{Q}}$;

1: Generate a sequence $w_i \in \mathcal{Q}$ using TSP;
2: **for** each $\mathcal{M}_i$ **do**
3: Pick $\tilde{w}_i$ using spiral base station placement algorithm [35];
4: $\widetilde{\mathcal{W}} \leftarrow \tilde{w}_i$;
5: Partitioned $\mathcal{W}$ into p ordered clusters $\widetilde{\mathcal{W}}$ in $\mathcal{C}_i$ using eq. (12);
6: **end for**
7: Generate a sequence $\tilde{w}_i \in \widehat{\mathcal{Q}}$ using TSP;
8: **while** until converge **do**
9: Converge eq. (13) using existing software like CVX [36];
10: **end while**
11: **return** $\widehat{\mathcal{Q}}$;

using (13):

$$\widehat{\mathcal{Q}} = \min_{\{\tilde{w}_i\}} \sum_{i=1}^{p} \|\tilde{w}_i - \tilde{w}_{i+1}\| + \|\tilde{w}_p - \tilde{w}_1\| \quad \text{s.t.} \quad \tilde{w}_i \in \mathcal{C}_i, \quad \forall i. \tag{13}$$

It is worth noting that the cost function $\widehat{\mathcal{Q}}$ has optimized waypoints $\widetilde{\mathcal{W}}$, which is a convex function with respect to $\widetilde{\mathcal{W}}$, and that all of $\widehat{\mathcal{Q}}$'s constraints are convex intersects every original waypoint. Thus, $\widehat{\mathcal{Q}}$ is a convex optimization problem that can be efficiently solved using standard convex optimization techniques or existing software like CVX [36]. Algorithm 2 details the process of UAV-trajectory optimization.

V. PERFORMANCE EVALUATION

In this section, we present the performance evaluation of our novel UAV-assisted digital signature-based Safe Code Dissemination (UDCD) scheme alongside similar existing schemes, such as UAV-assisted Trustworthy Code Dissemination (UTCD) [6] and Machine Learning-based Code Dissemination (MLCD) [26]. The UTCD scheme first deploys UAVs to collect sample code known as the *code waiting to be verified* (CWV) from specific selected ISDs. Subsequently, it selects credible code mules to assist in collecting more sample codes. These samples are then sent to the code center for verification, where they are compared to the original disseminated code to determine the credibility or maliciousness of the vehicle. Notably, the UTCD scheme employs UAVs to gather sample data for selecting credible code mules, considering them as the only trustworthy source to initiate the code dissemination process. However, the UTCD scheme does not conduct further evaluations of trust once a vehicle is determined to be credible or malicious.

On the other hand, the MLCD scheme utilizes a machine learning method to calculate a reliability degree based on a dataset of GPS locations of parked vehicles, thereby selecting credible code mules. MLCD deems vehicles with consistent parking patterns more credible than those with random parking patterns. To the best of our knowledge, Liang et al. [6]'s UTCD scheme and Li et al. [26]'s MLCD scheme have only performed studies similar to the one presented in this paper. However, these schemes may not perform optimally in scenarios where vehicles frequently enter and leave the network or in cases involving compromised vehicles, which motivated the proposal of the UDCD scheme.

The experiments were implemented on a desktop PC running Windows 10 Pro with an Intel i5 Skylake 2.6 GHz processor, 12 GB RAM, 475 GB SSD, 100/1000 M adaptive wireless network card, and Python version 3.10.

TABLE 3. Simulation parameters.

Parameter	Value
Number of ISDs	600
Total number of vehicles	300
Communication radius of ISDs	80 m
Communication radius of code mules	100 m
Hight of UAV	100 m
Flying speed of UAV	5 m/s
Code size	[0.1, 0.5] Mb
Maximum tolerant latency	[0.2, 1] s

A. EXPERIMENTAL SETUP

The real-world Rome taxi dataset[2] [37] is utilized to assess the effectiveness of the UDCD scheme in terms of improved coverage ratio, effective code dissemination rate, high reliability percentage, and optimized UAV-trajectory length. The dataset comprises GPS coordinates for approximately 300 taxicabs in Rome, Italy, collected between February 1 and February 8, 2014. Similar to [6], we adopt the same experimental scenario, selecting the city center as the code center, placing 600 coordinates randomly for the ISDs, and designating 300 vehicles as code mules in this dataset.

Specifically, we consider 210 credible code mules, 30 malicious code mules, and 60 uncertain code mules, each with a code dissemination probability of 100%, 0%, and 20% to 80%, respectively. Throughout the experiment, the trust value threshold for selecting a credible code mule in all three schemes is set at 0.5 on a scale of 0 to 1. Furthermore, we assume that out of 210 credible code mules, 6 are compromised at the start of each day, and by the end of each day, 50% of them are either credible or malicious, demonstrating the effect of compromised code mules. The objective of this study is to compare the operational performance of UDCD over compromised code mules with that of UTCD and MLCD schemes. Table 3 lists the other simulation parameters.

[2]https://crawdad.org/roma/taxi/20140717/

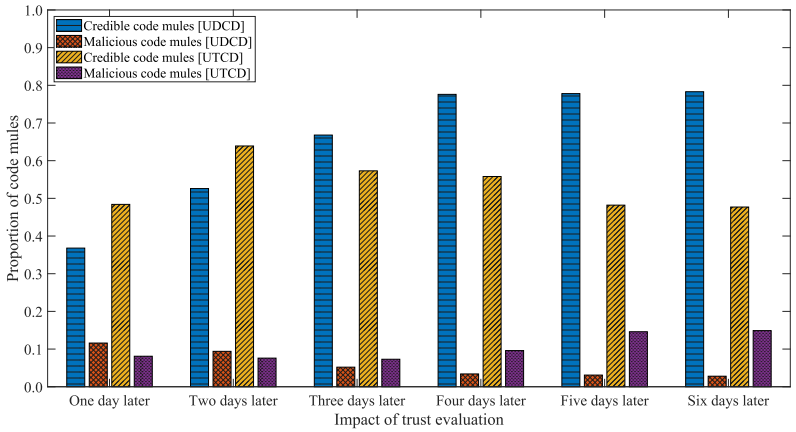

FIGURE 3. The distribution of $\mathcal{M}_i^{\mathrm{C}}$ and $\mathcal{M}_i^{\mathrm{M}}$ within seven days.

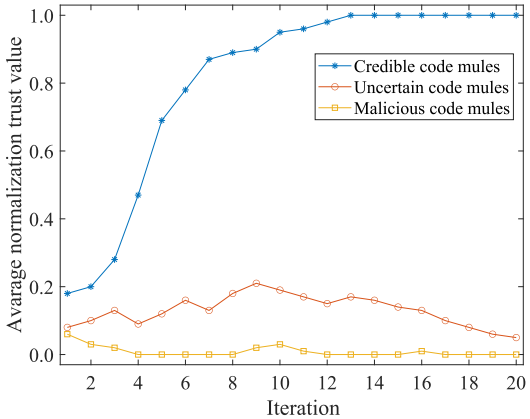

FIGURE 4. The average normalization trust value of each type of code mule.

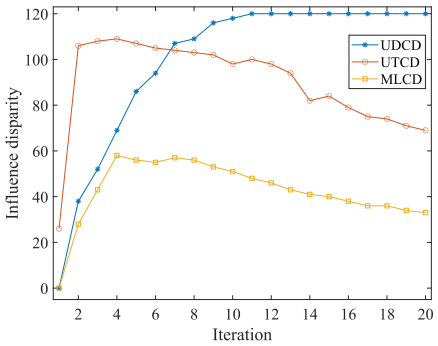

FIGURE 5. The influence disparity in each iteration.

B. HIGH RELIABILITY PERCENTAGE

Figure 3 illustrates the distribution of $\mathcal{M}_i^{\mathrm{C}}$ and $\mathcal{M}_i^{\mathrm{M}}$ over seven days using the UDCD and UTCD schemes, based on the impact of code mule trust value evaluation. The x-axis represents the daily proportion of $\mathcal{M}_i^{\mathrm{C}}$ and $\mathcal{M}_i^{\mathrm{M}}$ to the total available $\mathcal{M}_i$, which varies as discussed in Section V-A. According to the graph, after three days, UDCD can clearly distinguish between $\mathcal{M}_i^{\mathrm{C}}$ and $\mathcal{M}_i^{\mathrm{M}}$ and maintain a consistent proportion of $\mathcal{M}_i^{\mathrm{C}}$ and $\mathcal{M}_i^{\mathrm{M}}$ for the next four to seven days, with approximately 80% for $\mathcal{M}_i^{\mathrm{C}}$ and 3% for $\mathcal{M}_i^{\mathrm{M}}$. The UDCD scheme can effectively distinguish between different types of $\mathcal{M}_i$ because it has a reasonable trust value that is updated regularly, as shown in eq. (10) and eq. (11) in Section IV-B. However, it is impossible to distinguish between the types of $\mathcal{M}_i$ in the UTCD scheme because it contains a greater number of $\mathcal{M}_i^{\mathrm{M}}$ for code dissemination, lowering the overall proportion of $\mathcal{M}_i^{\mathrm{C}}$ and $\mathcal{M}_i^{\mathrm{M}}$.

We normalized the trust value of each iteration to more intuitively visualize the differences in the impact of trust value evaluation of the three types of code mules. This means that, for processing, the average trust value of a single code mule is mapped to a value between 0 and 1. Figure 4 illustrates the average normalized trust value of each type of code mule across different iterations. Once a stable cycle has been established, the value of $\mathcal{M}_i^{\mathrm{C}}$ approaches $\sim$1, while the value of $\mathcal{M}_i^{\mathrm{M}}$ approaches $\sim$0. Therefore, it is especially advantageous to select more $\mathcal{M}_i^{\mathrm{C}}$ for code dissemination and to eliminate $\mathcal{M}_i^{\mathrm{M}}$.

The difference between the average normalization trust value of $\mathcal{M}_i^{\mathrm{C}}$ and $\mathcal{M}_i^{\mathrm{M}}$ is referred to as influence disparity. Therefore, the greater the influence disparity, the better the UDCD scheme distinguishes between different types of code mules. Figure 5 illustrates the influence disparity in each iteration under the UDCD, UTCD, and MLCD schemes. The results show that the UDCD scheme maintains a particularly constant influence disparity after the tenth iteration because it regularly updates trust values even after differentiating the type of code mules.

The UTCD scheme exhibits a noticeable increase in influence disparity in the first two iterations due to the use of UAV to collect the verification code, but performance begins to degrade after two iterations. As both the UTCD and MLCD schemes are unable to distinguish between $\mathcal{M}_i^{\mathrm{C}}$ and $\mathcal{M}_i^{\mathrm{M}}$ with the growing number of compromised code mules, the code disseminator contains a large number of $\mathcal{M}_i^{\mathrm{M}}$. When malicious code mules are used as code disseminators to commit joint fraud, the authenticity of the disseminated code cannot be verified, making it difficult to distinguish between

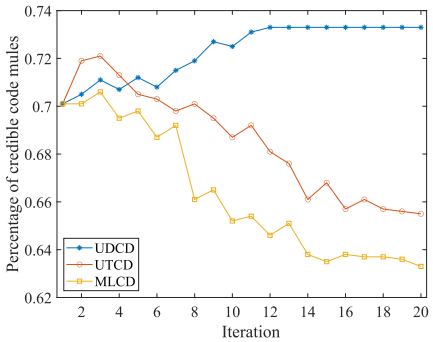

FIGURE 6. The percentage of $\mathcal{M}_i^{\mathrm{C}}$ in each iteration.

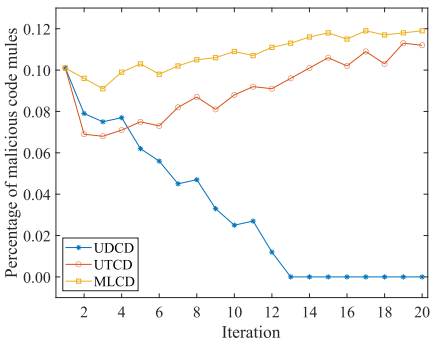

FIGURE 7. The percentage of $\mathcal{M}_i^{\mathrm{M}}$ in each iteration.

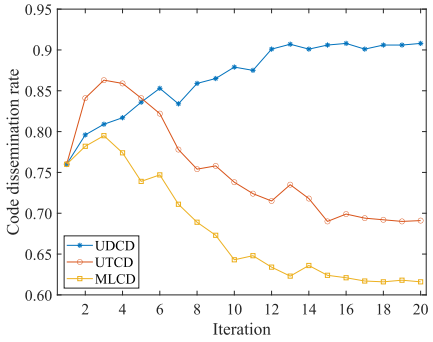

FIGURE 8. The integrated rate of code dissemination in different iterations.

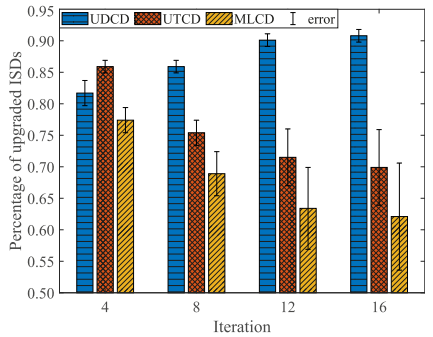

FIGURE 9. The percentage of upgraded ISDs and their error rate at various iterations.

the different types of code mules. It has been observed that the MLCD scheme has a high influence disparity because it has more $\mathcal{M}_i^{\mathrm{M}}$ as code disseminators at the start of the experiment. When compared to the UTCD and MLCD schemes, the UDCD scheme improves the influence disparity by 10.30% and 57.16%, respectively.

C. IMPROVED COVERAGE RATIO

The integrated rate of the improved coverage ratio is directly proportional to the number of $\mathcal{M}_i^{\mathrm{C}}$ and $\mathcal{M}_i^{\mathrm{M}}$ in the system. Figure 6 and Figure 7 illustrate the percentage of $\mathcal{M}_i^{\mathrm{C}}$ and $\mathcal{M}_i^{\mathrm{M}}$ in each iteration for the UDCD, UTCD, and MLCD schemes, respectively. When compared to the UTCD and MLCD schemes, the UDCD scheme detects 5.34% and 8.61% more $\mathcal{M}_i^{\mathrm{C}}$, respectively, and 6.28% and 8.19% more $\mathcal{M}_i^{\mathrm{M}}$.

UTCD and MLCD only consider selective forwarding and man-in-the-middle attacks, leaving them vulnerable to other types of attacks such as data pollution attacks, on-off attacks, and bad/good mouth attacks. Consequently, as the number of malicious code mules increases, so does the detection rate of these schemes. UDCD, on the other hand, is robust to all five types of malicious attacks.

D. EFFECTIVE DISSEMINATION RATE

The participation of $\mathcal{M}_i^{\mathrm{C}}$ in the code disseminator completely determines the effectiveness of the code dissemination rate. Figure 8 illustrates the integrated rate of code dissemination in different iterations under the UDCD, UTCD, and MLCD schemes. The graph demonstrates how the rate of code dissemination increases in the UDCD scheme as more $\mathcal{M}_i^{\mathrm{C}}$ join as code disseminators. In contrast, it is decreasing in the UTCD and MLCD schemes because UTCD and MLCD schemes are unable to make decisions on compromised code mules, which has a direct impact on the rate of code dissemination. In comparison to the UTCD and MLCD schemes, the UDCD scheme improves the code dissemination rate by 13.92% and 24.35%, respectively.

As previously stated, UTCD and MLCD schemes are unable to make decisions on compromised code mules and thus contain $\mathcal{M}_i^{\mathrm{M}}$ as code disseminators, resulting in more affected code dissemination. Figure 9 shows the percentage of upgraded ISDs and their error rate at various iterations. The affected code in the UTCD and MLCD schemes is 2.18% and 4.07% higher, respectively, than in the UDCD scheme.

E. COMPUTATIONAL OVERHEAD ANALYSIS

The computational overhead of the proposed UDCD scheme is significantly lower than that of the UTCD and MLCD. The increased computational overhead in the UTCD scheme is due to CWV verification for ISDs, whereas it nearly double in the MLCD scheme when compared to UDCD and UTCD due to the incorporation of the Machine Learning method. A detailed comparison of computational overhead is presented in Table 4.

TABLE 4. The comparison of computational overhead.

UDCD	UTCD	MLCD
9.2 s	13.8 s	23.1 s

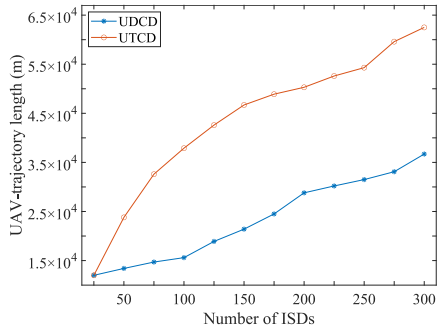

FIGURE 10. The UAV-trajectory length to update a given number of ISDs.

F. OPTIMISED UAV-TRAJECTORY LENGTH

To validate the performance of our virtual waypoints-based UAV flight trajectory optimization technique, we compare it to UTCD's Grouping-based UAV flight trajectory optimization technique [6]. Figure 10 displays the total length of the UAV trajectory required to update a given number of ISDs. According to the findings, UDCD's virtual waypoints-based UAV flight trajectory optimization technique covers a distance of 23.4 km, whereas UTCD's Grouping-based UAV flight trajectory optimization technique covers 43.6 km, nearly twice as far as the UDCD scheme, to upgrade 300 ISDs.

VI. CONCLUSION

The use of AI in IoT has increased the demand for code dissemination to upgrade ISDs using economical vehicles joint UAVs, making it an important research content in IIoT applications. However, secure code dissemination through a large number of vehicles in a semi-connected network of 5G-enabled IIoT systems is a challenging task. Therefore, we proposed a novel UAV-assisted Digital signature-based safe Code Dissemination (UDCD) framework for secure code dissemination. The main idea behind the UDCD scheme is to use V2V communication to verify the code with a digital signature and then send a recommendation to the code center. Later, the code center will use these recommendations in conjunction with the Subjective Logic model to determine whether to remove as many malicious code mules from the network as possible to ensure the integrity of the disseminated code. This is the first method that employs periodic trust updating to make appropriate decisions on compromised code mules at a later stage. Additionally, we proposed a virtual waypoint-based UAV-trajectory optimization method for upgrading non-upgraded ISDs to reduce UAV-trajectory length. The proposed UDCD framework, while robust, faces challenges in practical implementation related to regulatory constraints, integration with existing infrastructure, communication reliability, and adaptability to diverse environments. The results validated the UDCD scheme's performance on a real-world dataset compared to other existing state-of-the-art schemes. The UDCD framework, with its numerous application possibilities, is a powerful trust evaluation method that can be extended to all types of IoT devices in data collection. We intend to combine the traditional trust evaluation method with the UDCD method in the future to make it more generous and practical for semi-connected and distributed networks of 5G-enabled IIoT systems. Furthermore, our method will be tested in the development of a decentered octahedron growth/cellular automaton algorithm, which is used to predict microstructure formation during metal solidification, where conflicts can occur between the growing envelopes during the capture of neighboring cells.

ACKNOWLEDGMENT

Prof. Balázs Illés, Department of Electronics Technology, Budapest University of Technology and Economics, Budapest, Hungary, has made significant contributions to various aspects of the research, including experimental design, data collection, analysis, interpretation, and manuscript preparation.

REFERENCES

[1] M. Shen, A. Liu, G. Huang, N. N. Xiong, and H. Lu, "ATTDC: An active and traceable trust data collection scheme for industrial security in smart cities," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6437–6453, Apr. 2021.

[2] Y. Miao, X. Liu, K.-K. R. Choo, R. H. Deng, H. Wu, and H. Li, "Fair and dynamic data sharing framework in cloud-assisted Internet of Everything," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 7201–7212, Aug. 2019.

[3] S. Al-Sarawi, M. Anbar, R. Abdullah, and A. B. Al Hawari, "Internet of Things market analysis forecasts, 2020–2030," in *Proc. 4th World Conf. Smart Trends Syst., Security Sustain. (WorldS4)*, 2020, pp. 449–453.

[4] M. Huang, A. Liu, N. N. Xiong, and J. Wu, "A UAV-assisted ubiquitous trust communication system in 5G and beyond networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 11, pp. 3444–3458, Nov. 2021.

[5] F. Ahmad, F. Kurugollu, A. Adnane, R. Hussain, and F. Hussain, "MARINE: Man-in-the-middle attack resistant trust model in connected vehicles," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3310–3322, Apr. 2020.

[6] J. Liang, W. Liu, N. N. Xiong, A. Liu, and S. Zhang, "An intelligent and trust UAV-assisted code dissemination 5G system for Industrial Internet-of-Things," *IEEE Trans. Ind. Informat.*, vol. 18, no. 4, pp. 2877–2889, Apr. 2022.

[7] F. Restuccia, S. D'Oro, and T. Melodia, "Securing the Internet of Things in the age of machine learning and software-defined networking," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4829–4842, Dec. 2018.

[8] W. Xiao, W. Bao, X. Zhu, and L. Liu, "Cost-aware big data processing across geo-distributed datacenters," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 11, pp. 3114–3127, Nov. 2017.

[9] H. Teng et al., "A novel code data dissemination scheme for Internet of Things through mobile vehicle of smart cities," *Future Gener. Comput. Syst.*, vol. 94, pp. 351–367, May 2019.

[10] L. Hu, A. Liu, M. Xie, and T. Wang, "Uavs joint vehicles as data mules for fast codes dissemination for edge networking in smart city," *Peer-to-Peer Netw. Appl.*, vol. 12, no. 6, pp. 1550–1574, 2019.

[11] T. Wang, P. Wang, S. Cai, Y. Ma, A. Liu, and M. Xie, "A unified trustworthy environment establishment based on edge computing in industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 6083–6091, Sep. 2020.

[12] Y. Miao, J. Ma, X. Liu, J. Weng, H. Li, and H. Li, "Lightweight fine-grained search over encrypted data in fog computing," *IEEE Trans. Services Comput.*, vol. 12, no. 5, pp. 772–785, Sep./Oct. 2019.
[13] X. Liu, Y. Liu, A. Liu, and L. T. Yang, "Defending ON–OFF attacks using light probing messages in smart sensors for industrial communication systems," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 3801–3811, Sep. 2018.
[14] J. Wei, T. V. X. Phuong, and G. Yang, "An efficient privacy preserving message authentication scheme for Internet-of-Things," *IEEE Trans. Ind. Informat.*, vol. 17, no. 1, pp. 617–626, Jan. 2021.
[15] Y. Li, J. Ma, Y. Miao, L. Liu, X. Liu, and K.-K. R. Choo, "Secure and verifiable multikey image search in cloud-assisted edge computing," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5348–5359, Aug. 2021.
[16] T. Bouzid, N. Chaib, M. L. Bensaad, and O. S. Oubbati, "5G network slicing with unmanned aerial vehicles: Taxonomy, survey, and future directions," *Trans. Emerg. Telecommun. Technol.*, vol. 34, no. 3, 2023, Art. no. e4721.
[17] K. Messaoudi, O. S. Oubbati, A. Rachedi, A. Lakas, T. Bendouma, and N. Chaib, "A survey of UAV-based data collection: Challenges, solutions and future perspectives," *J. Netw. Comput. Appl.*, vol. 216, Jul. 2023, Art. no. 103670.
[18] V. Beretta, S. Harispe, S. Ranwez, and I. Mougenot, "Truth selection for truth discovery models exploiting ordering relationship among values," *Knowl.-Based Syst.*, vol. 159, pp. 298–308, Nov. 2018.
[19] Y. Liu, A. Liu, X. Liu, and M. Ma, "A trust-based active detection for cyber-physical security in industrial environments," *IEEE Trans. Ind. Informat.*, vol. 15, no. 12, pp. 6593–6603, Dec. 2019.
[20] A. Sharma, E. S. Pilli, A. P. Mazumdar, and P. Gera, "Towards trustworthy Internet of Things: A survey on trust management applications and schemes," *Comput. Commun.*, vol. 160, pp. 475–493, Jul. 2020.
[21] J. Tang, S. Fu, X. Liu, Y. Luo, and M. Xu, "Achieving privacy-preserving and lightweight truth discovery in mobile crowd-sensing," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 11, pp. 5140–5153, Nov. 2022.
[22] P. Sun et al., "Towards personalized privacy-preserving incentive for truth discovery in mobile crowdsensing systems," *IEEE Trans. Mobile Comput.*, vol. 21, no. 1, pp. 352–365, Jan. 2022.
[23] R. Sharma and B. Villányi, "Safe and secure oil and gas pipeline transportation system based on Industrial Internet of Things," *IEEE Sensors J.*, vol. 24, no. 5, pp. 6834–6845, Mar. 2024.
[24] Z. Zhou, C. Gao, C. Xu, Y. Zhang, S. Mumtaz, and J. Rodriguez, "Social big-data-based content dissemination in Internet of Vehicles," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 768–777, Feb. 2018.
[25] A. B. Reis, S. Sargento, and O. K. Tonguz, "Smarter cities with parked cars as roadside units," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 7, pp. 2338–2352, Jul. 2018.
[26] T. Li, M. Zhao, and K. K. L. Wong, "Machine learning based code dissemination by selection of reliability mobile vehicles in 5G networks," *Comput. Commun.*, vol. 152, pp. 109–118, Feb. 2020.
[27] L. Liu, C. Chen, T. Qiu, M. Zhang, S. Li, and B. Zhou, "A data dissemination scheme based on clustering and probabilistic broadcasting in VANETs," *Veh. Commun.*, vol. 13, pp. 78–88, Jul. 2018.
[28] A. A. Yavuz and M. O. Ozmen, "Ultra lightweight multiple-time digital signature for the Internet of Things devices," *IEEE Trans. Services Comput.*, vol. 15, no. 1, pp. 215–227, Jan./Feb. 2022.
[29] H. Li, R. Lu, L. Zhou, B. Yang, and X. Shen, "An efficient Merkle-tree-based authentication scheme for smart grid," *IEEE Syst. J.*, vol. 8, no. 2, pp. 655–663, Jun. 2014.
[30] X. Huang, R. Yu, J. Kang, Z. Xia, and Y. Zhang, "Software defined networking for energy harvesting Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1389–1399, Jun. 2018.
[31] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled Internet of Vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2906–2920, Mar. 2019.
[32] X. Huang, R. Yu, J. Kang, and Y. Zhang, "Distributed reputation management for secure and efficient vehicular edge computing and networks," *IEEE Access*, vol. 5, pp. 25408–25420, 2017.
[33] Y. Zhou et al., "Communication-and-computing latency minimization for UAV-enabled virtual reality delivery systems," *IEEE Trans. Commun.*, vol. 69, no. 3, pp. 1723–1735, Mar. 2021.
[34] L. Zhang and N. Ansari, "Latency-aware IoT service provisioning in UAV-aided mobile-edge computing networks," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10573–10580, Oct. 2020.
[35] H. Huang and A. V. Savkin, "Deployment of heterogeneous UAV base stations for optimal quality of coverage," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 16429–16437, Sep. 2022.
[36] M. Grant and S. Boyd, "CVX: MATLAB software for disciplined convex programming, version 2.1," 2014. [Online]. Available: https://cvxr.com/cvx/
[37] L. Bracciale, M. Bonola, P. Loreti, G. Bianchi, R. Amici, and A. Rabuffi, Jul. 2014, "CRAWDAD dataset roma/taxi." [Online]. Available: https://crawdad.org/roma/taxi/20140717

RAVI SHARMA (Member, IEEE) received the master's degree in computer science and engineering from the Indian Institute of Technology in Patna, Patna, India, in 2016. He is currently pursuing the Ph.D. degree in computer science and engineering with the Budapest University of Technology and Economics, Budapest, Hungary. His research interests include industry 4.0, Industrial Internet of Things, wireless sensor networks, security, and enterprise application integration.

BALÁZS VILLÁNYI (Member, IEEE) received the Ph.D. degree in computer science from the Budapest University of Technology and Economics, Budapest, Hungary, in 2016, where he is currently an Associate Professor and a Doctoral Advisor with the Faculty of Electrical Engineering and Informatics. His research interests include machine learning, schema matching algorithms, enterprise application integration, the Industrial IoT, and industry 4.0.