AEALV: Accurate and Efficient Aircraft Location Verification for ADS-B

Haomiao Yang, *Member, IEEE*, Qixian Zhou, Dongxiao Liu, *Member, IEEE*,
Hongwei Li, *Senior Member, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—The automatic dependent surveillance-broadcast (ADS-B) collects and analyzes massive cognitive aircraft location data, and is essential for safe air traffic controls in the modern aviation industry. In this paper, we present an accurate and efficient aircraft location verification scheme (AEALV) that preserves aircraft location privacy by utilizing grid-based *k*-nearest neighbor (*k*NN) algorithms. Specifically, we introduce a new approach to efficiently find the *k* nearest grid squares in the ciphertext domain by leveraging vector homomorphic encryption. Further, we present a quick identification technique for aircraft legitimacy by validating claimed locations instead of estimating real locations of the aircraft. This validation only involves calculating encrypted Euclidean distances in a small training circle, thereby significantly saving the verification time of claimed locations. We conduct extensive experiments and evaluate AEALV using real-world data. The results show that AEALV achieves accurate and efficient verification for aircraft locations while maintaining the confidentiality of both aircraft locations and grid data.**

***Index Terms*—Cognitive network, aviation big data, aircraft location verification, *k*NN, privacy preservation.**

I. Introduction

With the advance of Industry 4.0 technologies, including big data, cognitive computing, etc., the aviation industry has evolved into Aviation 4.0. In Aviation 4.0, the air traffic surveillance system is improved to help people make decisions automatically and intelligently, thereby enhancing flight safety [1]. Positioning aircraft is a crucial feature to guarantee flight safety by avoiding aircraft collision and missing. Especially, the automatic dependent surveillance-broadcast (ADS-B) technology requires aircraft to regularly broadcast their geographic locations acquired from the global navigation satellite system, notably increasing the positioning capability of the aircraft [2], [3]. To take the full advantage of Industry 4.0, massive ADS-B ground stations that form a large-scale sensor network, generate extremely large amounts of aviation data. One example is *OpenSky*,[1] a participatory sensor network that has collected over 20 trillion ADS-B messages from more than 2,000 sensors worldwide. Therefore, it is fairly essential to mine aviation big data to provide better intelligent decision-making. To this end, advanced machine learning algorithms can be utilized to automatically extract knowledge from aviation big data, greatly improving the cognitive ability of such sensor networks [4]–[7].

ADS-B is originally presented without sufficient security consideration [8]. Particularly, due to the lack of message authentication mechanisms, ADS-B is vulnerable to location-spoofing attacks [9]. To verify claimed locations of aircrafts, cryptographic message authentication mechanisms have been proposed [10], [11]. Nonetheless, the cryptographic solutions may face cross-border regulatory and technical complexities [8]. Although there are other non-cryptographic approaches to ensure location authenticity in conventional wireless broadcast networks [12]–[14], ADS-B surveillance networks have distinguishing characteristics and safety requirements [2]. Specifically, deep learning models can be used to predict aircraft locations [15], [16]. Since deep learning models are often regarded as unexplainable black boxes, the unexplainability of prediction models may be inappropriate for critical tasks with high safety requirements, such as air traffic controls. Therefore, technical advances are required for aircraft location verification (ALV) in ADS-B surveillance networks.

Using the technique of time difference of arrival (TDOA), a grid-based *k*NN algorithm has been recently designed for ALV [17]. To pinpoint aircraft locations, it splits the predefined airspace into a large number of small grid squares, and the square size is required small enough (e.g., $75m \times 75m$) to achieve high accuracy. For a typical monitoring airspace of $33000km^2$, there are about six million squares requiring the calculations of Euclidean distance. Owing to such enormous computational burdens, the computational efficiency of ALV cannot be guaranteed, thereby posing severe risks to flight safety.

Manuscript received December 12, 2020; revised March 10, 2021; accepted April 7, 2021. Date of publication April 13, 2021; date of current version December 9, 2021. This work is supported by the National Natural Science Foundation of China (62072081) and the National Key Research & Development Program of China (2017YFB0802000, 2017YFB0802003, 2017YFB0802300). The associate editor coordinating the review of this article and approving it for publication was S. Mao. *(Corresponding author: Dongxiao Liu.)*

Haomiao Yang is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: haomyang@uestc.edu.cn).

Qixian Zhou and Hongwei Li are with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: qxzhou1010@qq.com; hongweili@uestc.edu.cn).

Dongxiao Liu and Xuemin Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: dongxiao.liu@uwaterloo.ca; sshen@uwaterloo.ca).

Digital Object Identifier 10.1109/TCCN.2021.3072853

[1] https://opensky-network.org

To validate aircraft position claims efficiently and accurately, the *k*NN computations, as well as the large-scale position data, can be entrusted to a powerful third party, such as public or private clouds [18]–[20]. Unfortunately, public clouds are usually not fully trustworthy and thus raise location privacy issues for general aviation, since aircraft locations in general aviation may be leveraged to infer future business and geopolitical dealings [18], [21]. Compared with public clouds, private aviation clouds offer increased security through dedicated intranets [19]. Typically, an airport collaborative decision-making system is required to create a private cloud platform to exchange internal information among airports, air traffic controllers (ATC), airlines, etc [22]. Although it can eliminate the privacy concerns of public clouds, data sensitivity concerns may arise due to information sharing within the private cloud. The sensitive business intelligence of an airline may fall into the hands of other competitive airlines in the same private cloud. More noticeably, it is of paramount importance to protect the aeronautical information at ATC (e.g., the positions of the presidential plane) from unauthorized access of participants even in the same private cloud. Therefore, ATC and private aviation clouds may not belong to the same trust domain. In this case, if outsourcing aircraft position data to private aviation clouds, ATC may lose physical control of these sensitive data, thereby increasing the risk of leaking aircraft position privacy. For the privacy protection of aircraft positions, it is a natural thought to encrypt positions before outsourcing them [23]. Nevertheless, traditional ciphers (e.g., AES) fail in verifying encrypted positions due to no capability of manipulating ciphertexts. While homomorphic encryption (HE) supports computation over ciphertexts, common HEs, such as Paillier HE [24] or fully HE [25], suffer from significant computational efficiency issues. Recently, an efficient vector homomorphic encryption (VHE) scheme has been proposed for encrypting multi-dimensional data in a batching manner [26], providing a choice for ALV on massive encrypted data. However, it is not trivial for tailoring the design of ALV for the usage of VHE. It is challenging to measure encrypted Euclidean distances efficiently between the target aircraft and all squares in a grid plane without decryption [27]. Additionally, previous methods ignore the influence of aircraft altitude changes in the grid design [17], [28], which reduces the accuracy of aircraft location verification.

In this paper, we propose an Accurate and Efficient Aircraft Location Verification scheme (AEALV), while preserving aircraft location privacy. Concretely, our contributions are threefold:

– We propose an AEALV scheme by exploiting the grid-based *k*NN algorithm over encrypted aviation big data. Particularly, we develop a new approach that efficiently measures similarity between different ciphertexts, thereby being capable of finding the *k* nearest encrypted grid squares. Furthermore, we design a grid plane especially considering the influence of aircraft altitude change for aircraft location verification.

– We further develop a quick identification technique for aircraft legitimacy by validating claimed aircraft locations but not estimating real locations. As it is unnecessary to traverse the big grid plane, the calculation only involves evaluating encrypted Euclidean distances in a small training circle, thereby significantly saving the verification time.

– Security analysis proves that AEALV protects the privacy of both aircraft locations and grid data. Also, accuracy analysis shows that AEALV provides almost the same accuracy as the plaintext version over *OpenSky*. Besides, performance evaluation exhibits that AEALV achieves high efficiency. Even for six million grid squares, the time of evaluating encrypted distances is less than 200 ms.

TABLE I
NOTATIONS

Notation	Meaning
ATC	Air Traffic Controller
ADS-B	Automatic Dependent Surveillance-Broadcast
kNN	k-Nearest Neighbor
TDOA	Time Difference of Arrival
VHE	Vector Homomorphic Encryption
CPS	Central Processing Station
ALV	Aircraft Location Verification
v_c	Light speed
λ, l and w	Parameters of security, binary representation and scaling, respectively.
χ	Error distribution χ on $\mathbb{Z}_q$
p, q	Modulus of plaintext and ciphertext spaces, respectively
m, n	Dimensions of plaintext and ciphertext vectors, respectively
$\lvert \cdot \rvert$	Maximum entry of a vector
$\lVert \cdot \rVert$	Length of a vector
$\lceil \cdot \rfloor$	Nearest integer of each entry of a vector
$(\cdot)^*$	Binary representation of a vector
$\boldsymbol{e}$	Error vector with $\lvert \boldsymbol{e} \rvert < w/2$
$\boldsymbol{x}$, $\boldsymbol{c}$	Plaintext and ciphertext of a message, respectively
$\boldsymbol{S}$, $\boldsymbol{M}$ and $\boldsymbol{H}$	Matrices of Secret key, key-transforming and comparison, respectively

The rest of this paper is organized as follows. In Section II, we state preliminaries. In Section III, we give models and design goals. In Section IV, we elaborate on our solution. We then analyze security analysis and evaluate performance in Sections V and VI, respectively. In Section VII, we survey related works. In Section VIII, we conclude our work.

II. Preliminaries

We give preliminaries including *k*NN and VHE. Table I illustrates the notations used in this paper.

A. kNN

A severe risk of ADS-B is that wrong location reports, due to avionics malfunction or spoofing, may disrupt ATC operations [8]. To solve this problem, a grid-based *k*NN method is proposed for aircraft location verification [17]. This approach leverages a grid to train a *k*NN regression model by constructing a rectangular grid plane and then dividing it into a large number of squares, as shown in Fig. 1. Specifically, we assume that the grid plane covers *t* fixed ADS-B sensors. For each grid square, then there is a TDOA vector (also called a fingerprint) of dimension *t*. Particularly, an aircraft sends an ADS-B message at the light speed v_c, which will be received by *t* ADS-B

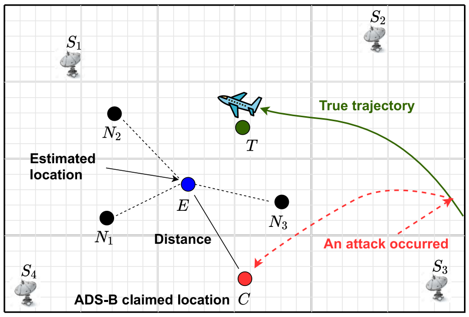


Fig. 1. Estimating the aircraft locations with 3NN. Using the TDOA fingerprints from four sensors S_1, S_2, S_3 and S_4, the three nearest neighbors N_1, N_2 and N_3 are averaged to acquire the estimated location E [28].

sensors. Time values of arrival at sensors are gathered, and *t* TDOA values are further calculated. Such *t* TDOA values constitute the finger vector of dimension *t*. Suppose that there is an aircraft in the grid plane broadcasting an ADS-B message, which includes its claimed location. We then can learn the current fingerprint of the aircraft. We further evaluate fingerprint similarity between the aircraft and each square in the grid plane by measuring their Euclidean distances. We finally find top-*k* squares with the smallest distances and then calculate their average. The average is the estimated location of the aircraft. If the gap between the estimated and claimed locations is greater than a pre-specified threshold, the claimed location may potentially be fake.

B. VHE

To protect aircraft location privacy, VHE is introduced to encrypt fingerprint vectors efficiently [26]. Although Bogos *et al.* [29] have demonstrated security issues of the original VHE, an improved VHE has been proposed to fix such security flaws, which is proved to be semantically secure [30]. The improved VHE includes five probabilistic-polynomial-time algorithms as (*VHE.KG*, *VHE.Enc*, *VHE.Dec*, *VHE.Add*, *VHE.KS*).

VHE.KG: Taking the security parameter λ as the input, choose randomly $w, p, q, m, n, l \in \mathbb{Z}$, and a noise distribution χ on $\mathbb{Z}_q$, in which $q \gg p$, $n > m$, $l = \lceil \log_2 (q-1) \rceil$ and $q > w(p-1)$. Generate the secret $\boldsymbol{S} = [\boldsymbol{I}, \boldsymbol{T}]$, in which $\boldsymbol{I}$ and $\boldsymbol{T}$ are the $m \times m$ identity and $m \times (n-m)$ random matrices, respectively. Finally, keep $\boldsymbol{S}$ privately and publish $Param = (w, p, q, m, n, l, \chi)$ publicly.

VHE.Enc: Taking the message $\boldsymbol{x} \in \mathbb{Z}_p^m$ and the secret key $\boldsymbol{S} \in \mathbb{Z}^{m \times n}$ as the inputs, generate the corresponding encrypted vector $\boldsymbol{c} \in \mathbb{Z}_q^n$ satisfying $\boldsymbol{Sc} = w\boldsymbol{x} + \boldsymbol{e}$, in which $\boldsymbol{e}$ is the error vector.

VHE.Dec: Taking the ciphertext $\boldsymbol{c} \in \mathbb{Z}_q^n$ and the secret key $\boldsymbol{S} \in \mathbb{Z}^{m \times n}$ as the inputs, recover the corresponding message $\boldsymbol{x} \in \mathbb{Z}_p^m$ satisfying $\boldsymbol{x} = \lceil \frac{\boldsymbol{Sc}}{w} \rfloor_q$.

VHE.Add: For $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ and the corresponding ciphertexts $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ under the same $\boldsymbol{S}$, we have $\boldsymbol{S}(\boldsymbol{c}_1 + \boldsymbol{c}_2) = w(\boldsymbol{x}_1 + \boldsymbol{x}_2) + (\boldsymbol{e}_1 + \boldsymbol{e}_2)$.

VHE.KS: Through the key-switching matrix $\boldsymbol{M}$, convert the old pair $(\boldsymbol{S}_{old}, \boldsymbol{c}_{old})$ to the new pair $(\boldsymbol{S}_{new}, \boldsymbol{c}_{new})$ with the same $\boldsymbol{x}$. Then, we have $\boldsymbol{c}_{new} = \boldsymbol{M}(\boldsymbol{c}_{old})^*$ and $\boldsymbol{S}_{new}\boldsymbol{c}_{new} = \boldsymbol{S}_{old}\boldsymbol{c}_{old} = w\boldsymbol{x} + \boldsymbol{e}$ [26]. Specifically, let $\boldsymbol{S}_{old} = \boldsymbol{I}$, $\boldsymbol{S}_{new} = \boldsymbol{S}$ and $\boldsymbol{e} = \boldsymbol{0}$. Then, we have

$$\boldsymbol{c} = \boldsymbol{M}(w\boldsymbol{x})^*, \ \ w\boldsymbol{x} = \boldsymbol{I}^*(w\boldsymbol{x})^*. \tag{1}$$

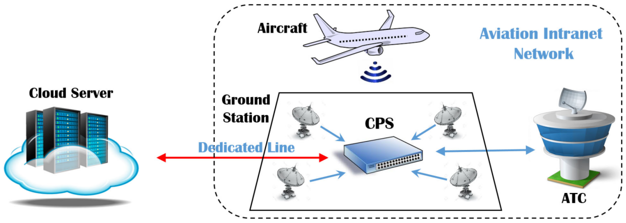


Fig. 2. System model.

III. Models and Design Goals

We give system model, threat model and design goals.

A. System Model

Fig. 2 shows the system model comprising entities below.

- *Aircraft:* An aircraft with ADS-B transmitter sends its location claims to multiple ground stations.
- *Ground Station:* A ground station with ADS-B sensor receives a message containing claimed location, recording the arrival time of the message at the station. Then, the claimed location and arrival time are both sent to the central processing station.
- *Central Processing Station (CPS):* For different arrival times of the same ADS-B message to multiple ADS-B stations, CPS firstly calculates TDOAs. CPS then encrypts DOAs of training grid squares and aircraft, and uploads the encrypted TDOAs to the cloud. CPS further receives the encrypted estimated position from the cloud, then decrypting and sending it to ATC.
- *Air Traffic Controller (ATC):* Upon receiving claimed and estimated locations from CPS, ATC judges if the location claim is legitimate. If yes, ATC displays it on monitoring screens. Besides, ATC initializes the whole system of ALV.
- *Cloud Server:* The cloud server performs computationally intensive tasks of the grid-based *k*NN over massive encrypted data from CPS, and then returns the results to CPS.

B. Threat Model

Due to the lack of message authentication mechanisms in ADS-B, the location data in broadcast ADS-B messages may be forged or modified. Therefore, claimed aircraft locations need to be validated for security. We perform such verification using TDOA, where multiple ADS-B ground stations measure the message arrival time when they receive the same ADS-B message from the aircraft. The ground stations, then, transmit these data to CPS, which can calculate the aircrafts real location using TDOA. Further, ADS-B ground sensors, CPS and ATC are connected by the high-speed aviation intranet. Thus, we can consider that they all belong to the same trust domain. In this way, communications among them are secure and efficient, and data can be transmitted in clear-text without encryption. However, the cloud does not belong to the same trust domain and is usually assumed to be semi-honest [18]. The cloud is interested to pry aircraft location privacy for commercial purposes and other uses. Especially, if these sensitive data are directly uploaded to the cloud, in the verification phase, the cloud can know claimed and estimated positions of aircraft. Even in the training phase, the cloud can also learn fingerprints and locations of grid squares in the monitoring area, which are private for the aviation network. Besides, it is worth noting that the interface between the cloud server and the aviation network is via CPS, and communications between them can be through the high-speed dedicated line, as illustrated in Fig. 2, thereby guaranteeing the authenticity, integrity and efficiency of transmission.

In summary, for the threat model, there exist location privacy issues in the cloud. It is assumed that the cloud is semi-honest and has incentives to snoop location privacy for commercial interests or other purposes. On the one hand, the cloud faithfully follows the designated algorithm to perform calculations for aircraft location verification. It cannot tamper with the calculation results. On the other hand, the cloud is curious and wants to know the aircraft location from outsourced data. To this end, the cloud can launch privacy leakage attacks.

C. Design Goals

The overall goal is to achieve accurate and efficient aircraft location verification with privacy preservation in air surveillance networks, and thus the following design goals need to be achieved.

– *Accuracy:* The high accuracy for aircraft location verification in the ciphertext domain should be guaranteed.
– *Security:* The confidentialities on two classes of data should be protected: 1) Aircraft position data, including claimed locations and TDOA fingerprints; 2) Grid data, containing coordinates and fingerprints of grid squares. Also, the secure distance metric should be achieved.
– *Performance:* The verification time is bounded to ensure that ATC could efficiently monitor aircraft.

IV. Proposed Scheme

Firstly, we design a grid plane considering influence of aircraft altitude change. We then evaluate fingerprint similarity over encrypted vectors. On this basis, we propose an aircraft location validation scheme with privacy protection. Finally, we develop a quick identification technique for aircraft legitimacy.

A. Grid Design Considering Altitude Change

For the grid-based *k*NN, the conventional method only sets grid planes at the typical cruising altitude of 11000*m*. However, with the increase of high-altitude jet traffic, the aircraft capacity in the airspace of 11000*m* altitude is becoming crowded. Therefore, the cruising altitude nowadays hovers between 9000*m* and 13000*m* the altitude band preferred by airlines for fuel economy. Besides, to enhance flight safety, the surveillance range includes not only en-route cruising, but also the airport, runaway, and terminal. Consequently, the altitude change influence on ALV needs further study.

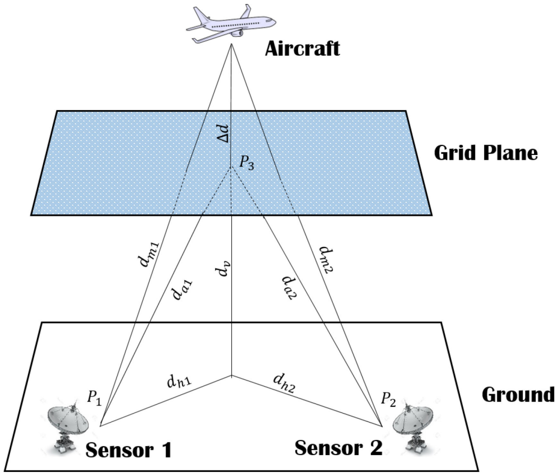


Fig. 3. Influence of altitude.

As shown in Fig. 3, Sensor 1 and 2 are located at points P_1 and P_2, respectively, on the ground, and the aircraft is located at point P_3 on the grid plane. Note that sensors may be located at different heights from the ground. However, relative to the cruise altitude of the aircraft, the heights of the sensors may be negligible. Aircraft fly at cruise altitudes that are typically 11,000 meters or higher, while sensors are typically deployed at altitudes between 0 and 500 meters.

The horizontal and vertical distances between the sensor and aircraft are d_h and d_v, respectively, and thus the actual distance is d_a. Then, there is a differentiation term Δd added to the vertical distance representing the aircraft altitude change. Hence, the measured distance is d_m. $|\Delta d|$ is much less than $2d_v$ because $|\Delta d|$ is usually no more than 2000*m* while $2d_v$ is larger than 18,000*m*. Consequently, we have

$$d_a = \sqrt{d_h^2 + d_v^2},\ d_m = \sqrt{d_h^2 + (d_v + \Delta d)^2}. \tag{2}$$

Since TDOA is used to estimate the aircraft location, it is necessary to find the distance difference from the aircraft to two different sensors, say Sensor 1 and 2. Note that the horizontal distances d_{h1} and d_{h2} are different from the aircraft to such two sensors, but the vertical distances are the same ($d_v = d_{v1} = d_{v2}$) because they are referred to the same grid plane. Therefore, we have

$$d_{a1} = \sqrt{d_{h1}^2 + d_v^2},\ d_{m1} = \sqrt{d_{h1}^2 + (d_v + \Delta d)^2}, \tag{3}$$

$$d_{a2} = \sqrt{d_{h2}^2 + d_v^2},\ d_{m2} = \sqrt{d_{h2}^2 + (d_v + \Delta d)^2}. \tag{4}$$

Let $a = d_{h1}^2 + d_v^2, b = d_{h2}^2 + d_v^2$. Then, the original distance difference of two sensors to the aircraft is

$$d_1 = \sqrt{a} - \sqrt{b}, \tag{5}$$

and considering the altitude change, the difference of measured distances is

$$d_2 = \sqrt{a + e} - \sqrt{b + e}, \tag{6}$$

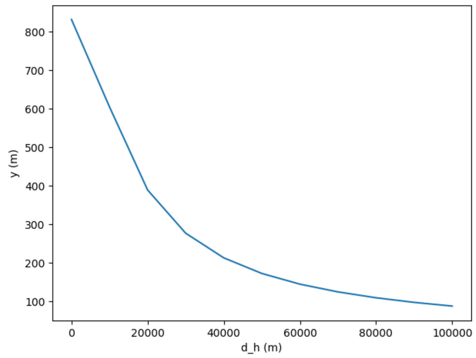


Fig. 4. The relationship between y and d_h.

where $e = \Delta d(\Delta d + 2d_v) \approx 2d_v \Delta d$ since $|\Delta d|$ is much less than $2d_v$.

To measure the changed TDOA value ($\Delta t = (d_1 - d_2)/v_c$, in which v_c is the light speed), the value of $d = (d_1 - d_2)$ is also needed to be calculated as

$$d = d_1 - d_2 = \left(\sqrt{a} - \sqrt{a+e}\right) - \left(\sqrt{b} - \sqrt{b+e}\right). \quad (7)$$

To acquire d_{max}, the maximum of the error d caused by altitude change, let $max(d_1)$ and $min(d_2)$ be the maximum of d_1 and the minimum of d_2, respectively. As both of them can be simultaneously obtained, we have

$$d_{max} = max(d_1) - min(d_2). \quad (8)$$

Obviously, d_i is a function of the horizontal distance d_{hi} between Sensor *i* and the aircraft. Hence, we have

$$d_i = \sqrt{x_i} - \sqrt{x_i + e}, \quad (9)$$

where $x_i = d_{hi}^2 + d_v^2, i = 1, 2$. Without loss of generality, the function relationship is expressed as

$$y = \sqrt{x} - \sqrt{x+e}, \quad (10)$$

where $x = d_h^2 + d_v^2$. We illustrate the function in Fig. 4, in which $d_v = 11000m$, $|\Delta d| = 2000m$ and $0 < d_h < 100{,}000m$ regarding the detection range of typical ADS-B sensors is 100*km*. Note that *y* varies from the maximum of 950.1*m* to the minimum of 99.4*m*. Consequently, we have $max(d_1) = 950.1m$ and $min(d_2) = 99.4m$. Further, we have $d_{max} = max(d_1) - min(d_2) = 850.7m$, that is to say, d_{max} can be achieved if one sensor is almost beneath the aircraft and the other is far away from the aircraft. Further, concerning the light speed v_c, the corresponding time change is 2.84*us*, which has little effect on TDOA calculations.

Finally, we run experiments to demonstrate the influence of altitude change on the accuracy of aircraft location estimation. The experiment setup is referred to as Section VI-A. As shown in Table II, the errors between real positions and estimated positions are the positive correlation with $|\Delta d|$. And for 600*m* grid, if $|\Delta d| = 1500m$, the mean error is no more than 450*m* less than the grid size; thus the estimated position is unchanged according to the grid-based *k*NN. Therefore, we neglect the influence of altitude change within 1500*m* when designing a grid plane. Otherwise, we suggest that the grid plane may be divided into multiple layers, and inside each layer, the altitude change should be less than 1500*m*.

TABLE II
INFLUENCE OF ALTITUDE CHANGE (600m SQUARE)

Altitude change [m]	**Horizontal Error** [m]	
	mean	**median**
500	237.25	174.96
1000	341.61	251.32
1500	449.62	341.3
2000	609.93	456.68

B. Similarity Measurement Over Ciphertexts

To achieve the non-interactive and efficient similarity measurement over ciphertexts that is essential for AEALV, one possibility is to invoke existing methods, such as secure multiparty computation (SMC) [27] or fully HE [25]. However, these approaches have respective shortcomings. SMC requires multiple rounds of interactions, and fully HE is costly to evaluate the comparison circuits.

With VHE, we present a novel technique for secure similarity measurement, which will play an essential role in the grid-based *k*NN. Assume that there are three vectors $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ that are encrypted into $\boldsymbol{c}_1$, $\boldsymbol{c}_2$ and $\boldsymbol{c}_3$ by VHE, respectively. The challenge is how to conduct a similarity metric among these three encrypted vectors. Therefore it is necessary to know which vector between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ is closer to $\boldsymbol{x}_3$ according to the Euclidean distance in the encrypted domain. To meet such challenge, we design a comparison matrix $\boldsymbol{H}$ as follows. First, by solving equation $\boldsymbol{AM} = \boldsymbol{I}^*$, the matrix $\boldsymbol{A}$ can be obtained, in which $\boldsymbol{M}$ is a key-switching matrix corresponding to the secret key matrix $\boldsymbol{S}$ and $\boldsymbol{I}^*$ is a binary representation of the identity matrix $\boldsymbol{I}$. Then, $\boldsymbol{H}$ is calculated as $\boldsymbol{H} = \boldsymbol{A}^T\boldsymbol{A}$. Note that only given $\boldsymbol{H}$, it is impossible to retrieve $\boldsymbol{A}$ from $\boldsymbol{H}$ and further acquire $\boldsymbol{M}$. The detailed security analysis will be given in Section V-B later. As a result, $\boldsymbol{x}$ is not uniquely decided even if knowing $\boldsymbol{c}$ and $\boldsymbol{H}$. Furthermore, we have the following two prepositions.

Proposition 1: Given the comparison matrix $\boldsymbol{H}$, a plaintext vector $\boldsymbol{x}$ and the corresponding ciphertext vector $\boldsymbol{c}$, we have

$$\boldsymbol{c}^T\boldsymbol{H}\boldsymbol{c} = w^2\|\boldsymbol{x}\|^2. \quad (11)$$

Proof: According to Eq. (1), we have $\boldsymbol{c} = \boldsymbol{M}(w\boldsymbol{x})^*$ and $w\boldsymbol{x} = \boldsymbol{I}^*(w\boldsymbol{x})^*$. Therefore, we have

$$\begin{aligned}\boldsymbol{c}^T\boldsymbol{H}\boldsymbol{c} &= (\boldsymbol{M}(w\boldsymbol{x})*)^T\left(\boldsymbol{A}^T\boldsymbol{A}\right)(\boldsymbol{M}(w\boldsymbol{x})*)\\ &= \left((w\boldsymbol{x})^*\right)^T(\boldsymbol{AM})^T(\boldsymbol{AM})(w\boldsymbol{x})^*\\ &= \left((w\boldsymbol{x})^*\right)^T(\boldsymbol{I}^*)^T(\boldsymbol{I}^*)(w\boldsymbol{x})^*\\ &= \left(\boldsymbol{I}^*(w\boldsymbol{x})^*\right)^T\left(\boldsymbol{I}^*(w\boldsymbol{x})^*\right)\\ &= (w\boldsymbol{x})^T(w\boldsymbol{x})\\ &= w^2\|\boldsymbol{x}\|^2 \qquad \blacksquare\end{aligned}$$

Proposition 1 implies the relation among $\boldsymbol{c}$, $\boldsymbol{H}$ and $\|\boldsymbol{x}\|^2 (= \boldsymbol{x}^T\boldsymbol{x})$. It means that by computing only over ciphertexts, the length of $\|\boldsymbol{x}\|$ can be achieved. Note that what we know is only the length of $\boldsymbol{x}$, but the content of $\boldsymbol{x}$ is still unknown.

Proposition 2: Given three plaintexts $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$, the corresponding ciphertexts $\boldsymbol{c}_1$, $\boldsymbol{c}_2$ and $\boldsymbol{c}_3$, and the comparison matrix $\boldsymbol{H}$, the following condition holds: if $(\boldsymbol{c}_1-\boldsymbol{c}_3)^T\boldsymbol{H}(\boldsymbol{c}_1-\boldsymbol{c}_3) \leq (\boldsymbol{c}_2-\boldsymbol{c}_3)^T\boldsymbol{H}(\boldsymbol{c}_2-\boldsymbol{c}_3)$, then

$$\|\boldsymbol{x}_1-\boldsymbol{x}_3\| \leq \|\boldsymbol{x}_2-\boldsymbol{x}_3\|.$$

The correctness of Proposition 2 can be directly achieved by applying Proposition 1 and $VHE.Add$. It is worth noting that Proposition 2 has an important meaning, that is, using only ciphertexts, we can complete vector similarity metrics. Especially, it is easy to see that such similarity measurement over ciphertexts does not require interactions, thereby saving communication overhead.

C. Secure and Efficient Aircraft Location Verification

AEALV contains the following three phases: *Initialization*, *Offline Training* and *Online Verification*.

1) Initialization: In this phase, ATC sets up the ALV scheme and publishes the system parameters, including the following steps.

- *Step 1:* With respect to requirements of air traffic surveillance, ATC first specifies the grid parameter *GridParam* containing the longitude, latitude, altitude, the size of grid plane and the size of each grid square.
- *Step 2:* ATC generates the secret keys $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, and the VHE parameter $VHEParam = (l, m, n, p, q, w, \chi)$ by invoking $VHE.KG(\lambda)$.
- *Step 3:* ATC designates the *k*NN algorithm parameter *kNNParam* including the *k* value and threshold d_{th}.
- *Step 4:* ATC publishes in public the system parameter (*GridParam*, *VHEParam*, *kNNParam*), and transmits $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ to CPS for privately keeping.

2) Offline Training: In this phase, CPS performs the offline training for the whole grid plane of surveillance, including the steps below.

- *Step 1:* CPS calculates the fingerprint for each square in the grid plane $\mathcal{R}$ by leveraging the collected values of TDOA from ADS-B sensors. Assuming that $\mathcal{R}$ contains *m* squares and there are *n* sensors, there is a data set $\boldsymbol{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i), i = 1, \ldots, m\}$, in which $\boldsymbol{x}_i \in \mathbb{Z}^n$ is the fingerprint vector of the *i*-th square with respect to *n* sensors, and $\boldsymbol{y_i} \in \mathbb{Z}^2$ is the position of two-dimensional coordinates of the *i*-th square in the plane.
- *Step 2:* CPS encrypts $\boldsymbol{D}$ into $\boldsymbol{D}_e$ with the secrets $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ as $\boldsymbol{D}_e = \{(\boldsymbol{c}_i, \boldsymbol{l}_i), i = 1, \ldots, m\}$, where $\boldsymbol{c}_i = VHE.Enc(\boldsymbol{x}_i, \boldsymbol{S}_1)$ and $\boldsymbol{l}_i = VHE.Enc(\boldsymbol{y}_i, \boldsymbol{S}_2)$.
- *Step 3:* CPS firstly extracts the key-switching matrix $\boldsymbol{M}$ corresponding to the secret matrix $\boldsymbol{S}_1$ from $\boldsymbol{S}_1\boldsymbol{M} = \boldsymbol{I}^* + \boldsymbol{E}$, in which $\boldsymbol{E}$ is the small error matrix. CPS, then, calculates the comparison matrix $\boldsymbol{H}_1$ based on $\boldsymbol{M}$ as described in Section IV-B.
- *Step 4:* CPS uploads $\boldsymbol{D}_e$ and $\boldsymbol{H}_1$ to the cloud. Note that the $\{(\boldsymbol{c}_i, \boldsymbol{l}_i), i = 1, \ldots, m\}$ should be transmitted in a disorder of the index *i*; otherwise, it would cause serious security concerns. The reason is briefly given as follows: If the cloud server knows the published grid parameter, then the index *i* can be reverted to the coordinate of square by taking the inverse procedure of transmission.

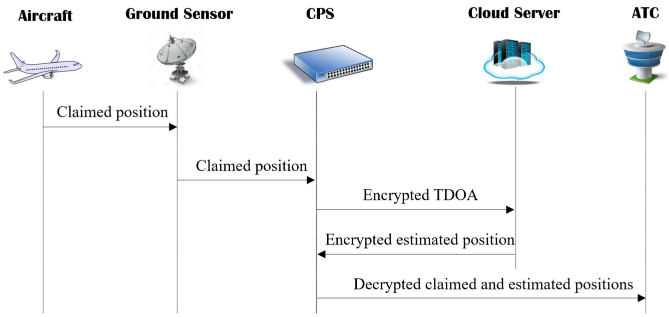


Fig. 5. Online verification process.

Algorithm 1 Aircraft Location Estimation

```
Input: $\boldsymbol{D}_e$, $\boldsymbol{H}_1$ and $\boldsymbol{c}$
Output: $\boldsymbol{l}'$
for $i = 1$ to $m$ do
    Calculate $t_i \leftarrow (\boldsymbol{c}-\boldsymbol{c}_i)^T\boldsymbol{H}_1(\boldsymbol{c}-\boldsymbol{c}_i)$
end for
Find $k$ nearest squares $\{t'_i, i = 1, \ldots, k\}$ by running kNN in $\{t_i, i = 1, \ldots, m\}$
Find the corresponding $\{\boldsymbol{l}'_1, \ldots, \boldsymbol{l}'_k\}$ in $\boldsymbol{D}_e$
Calculate $\boldsymbol{l}' \leftarrow \frac{1}{k}(\boldsymbol{l}'_1 + \cdots + \boldsymbol{l}'_k)$
return Encrypted estimated location $\boldsymbol{l}'$
```

3) Online Verification: In this phase, the online process of aircraft location verification is described in the following steps. The process is also illustrated in Fig. 5.

- *Step 1:* Suppose that there is an aircraft in the surveillance area broadcasting ADS-B messages. CPS firstly calculates the fingerprint $\boldsymbol{x}$ through collecting messages from sensors and encrypts $\boldsymbol{x}$ into $\boldsymbol{c}$ with $\boldsymbol{S}_1$ by invoking $\boldsymbol{c} \leftarrow VHE.Enc(\boldsymbol{x}, \boldsymbol{S}_1)$. Then, the encrypted fingerprint $\boldsymbol{c}$ is uploaded onto the cloud.
- *Step 2:* The cloud evaluates the estimated aircraft location in the ciphertext domain. Specifically, by running the grid-based *k*NN algorithm, the cloud server finds the *k* nearest squares $\boldsymbol{l}'_1, \ldots, \boldsymbol{l}'_k$, and calculates the average value as $\boldsymbol{l}' = \frac{1}{k}(\boldsymbol{l}'_1 + \cdots + \boldsymbol{l}'_k)$. Note that $\boldsymbol{l}'$ is the encrypted estimated location. The cloud server then returns $\boldsymbol{l}'$ to CPS. The process of *Step 2* can be given in Algorithm 1, the correctness of which can be assured as
$$(\boldsymbol{c}-\boldsymbol{c}_i)^T\boldsymbol{H}_1(\boldsymbol{c}-\boldsymbol{c}_i) = w^2\|\boldsymbol{x}-\boldsymbol{x}_i\|^2.$$
- *Step 3:* CPS first acquires the estimated location $\boldsymbol{y}'$ by invoking $\boldsymbol{y}' \leftarrow VHE.Dec(\boldsymbol{l}', \boldsymbol{S}_2)$, then extracts the claimed location $\boldsymbol{y}$ from the received ADS-B message *message*, and finally transmits *message*, $\boldsymbol{y}$ and $\boldsymbol{y}'$ to ATC for aircraft location verification.
- *Step 4:* ATC checks if $\|\boldsymbol{y}-\boldsymbol{y}'\| < d_{th}$, and if yes, the claimed location is testified valid and displayed on the ATC monitoring system; otherwise, the received *message* is discarded.

D. Quick Identification of Aircraft Legitimacy

In the above scheme, we leverage the grid-based *k*NN algorithm to judge the legitimacy of aircraft claimed position. Unfortunately, the computational costs rapidly rise with the

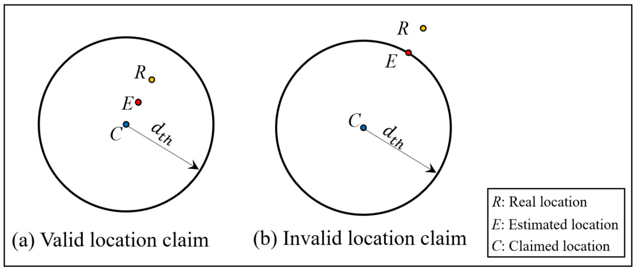


Fig. 6. Valid and invalid location claims.

number of squares. For the grid plane of two degrees longitude by two degrees latitude, to improve the accuracy of air location estimation, the square size needs to be set small enough, e.g., $75 \times 75m^2$. In this case, there are about six million squares in the plane, and the aircrafts fingerprint needs to be compared with the fingerprints of such massive squares. In the following, we propose a quick aircraft location verification technique, which only involves a small number of squares taking part in the calculation. Recall the scheme in Section IV-C. The estimated location of aircraft is firstly calculated based on the grid-based *k*NN by measuring encrypted Euclidean distances of fingerprints between the airplane and all squares in the plane. Subsequently, it is compared with the claimed position. Finally, if the gap between them is larger than the designated threshold d_{th}, then the position claim testifies invalid.

It is easy to see that the most time-consuming operation in the above scheme is to evaluate the estimated location by traversing the whole plane for all squares. Therefore, if only validating the location claim but not estimating the real position of aircraft, the amount of calculation would be reduced. Based on such thoughts, a quick verification technique is presented as follows. The basic idea is to draw a circle with the claimed location as the center and $R_1 = d_{th}$ as the radius. If the position claim is real, it is inside this circle; otherwise, the estimated position is near the boundary of this circle. Note that the spoofed airplane should have been outside the circle, but the training algorithm is limited to run inside this circle. Hence, its estimated position can only reach the boundary of this circle, as shown in Fig. 6.

The grid-based *k*NN takes the average of top *k* squares to determine the estimated position, and the *k* squares are adjacent. Accordingly, it is possible that although the estimated position is inside this circle, one or more of *k* squares may locate outside it. Thus, to remain the accuracy, the training circle should be bigger to cover those squares. For $k = 5$ of the typical setting, if the estimated position exactly locates at the boundary of the circle, the maximum distance occurs when such five adjacent squares make a straight line, and the midpoint is just the estimated position, which is also at the boundary of the circle, as shown in Fig. 7. Therefore, a bigger circle is drawn with $R_2 = d_{th} + 2s$ as the radius, where s is the square side length. As a consequence, if the estimated position R is inside the smaller circle ($R < R_1$), the location claim is valid; if $\boldsymbol{R}$ is between the smaller and larger circles ($R_1 < R < R_2$), the location claim is invalid.

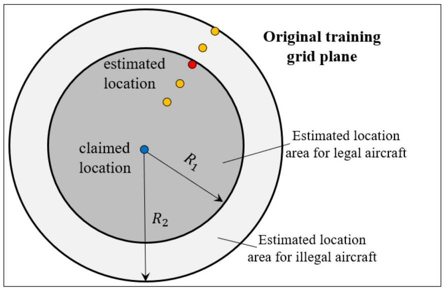

Original training grid plane
estimated location
claimed location
R_1
R_2
Estimated location area for legal aircraft
Estimated location area for illegal aircraft

Fig. 7. Different areas in the training grid plane.

In the typical setting, for a grid plane, the area is $3.3 \times 10^{10} m^2$. However, for the training circle, if $d_{th} = 500m$ and $s = 75m$, the area is only $1.33 \times 10^6 m^2$, which is much less than that in original plane. Consequently, it is much quicker in the training circle than in the whole grid plane to identify the legitimacy of the aircraft. As a result, if not evaluating real aircraft location, squares outside the training circle is redundant for distinction because distances between these squares and the claimed position are more than the threshold. Only when ATC needs to find the real position of the ghost aircraft, these squares would take part in calculation.

V. Security Analysis

In general, communications among entities of the aviation intranet, including ground stations, CPS and ATC, are considered to be secure. Communications between CPS and outside clouds are protected by the high-speed dedicated line, and authenticity and integrity are preserved from other attackers. We further assume that the cloud is semi-honest and has incentives to snoop position privacy for commercial interests or other purposes. Therefore, security should be protected from the cloud for outsourced *k*NN computing tasks. In this section, we analyze security from two aspects: (1) confidentialities of location claims and fingerprints of aircraft, and coordinates and fingerprints of grid squares, (2) secure distance metrics in outsourced *k*NN. First of all, we give the definition of the learning with errors (LWE) problem [31] for the following security proof.

Definition 1 (The Learning With Error (LWE) Problem): Given polynomially many samples of $(\boldsymbol{a}_i, b_i) \in \mathbb{Z}_q^m \times \mathbb{Z}_q$ satisfy

$$b_i = \boldsymbol{v}^T \boldsymbol{a}_i + \varepsilon_i, \tag{12}$$

in which $\varepsilon_i \in \mathbb{Z}_q$ is an error term generated from a certain probability distribution χ, there does not exist any solvers to obtain $\boldsymbol{v} \in \mathbb{Z}_q^m$ without negligible probability.

A. Confidentiality

As aforementioned, we build AEALV confidentiality over VHE. That is, aircraft position data (location claims and fingerprints) and grid data (coordinates and fingerprints of squares), are all encrypted using VHE with the secret keys $\boldsymbol{S}_1$ or $\boldsymbol{S}_2$. The VHE security is further achieved because of the hardness of the LWE problem as shown in the following theorem.

Theorem 1: VHE given in Section II-B is semantically secure assuming the hardness of LWE.

Proof: The security proof of VHE is presented in our previous work [30]. ■

Consequently, such confidentialities are assured if $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ are kept privately by CPS. The cloud cannot extract any useful information from ciphertexts, preserving data confidentiality. Additionally, in the phase of *offline training*, encrypted coordinates and fingerprints of grid squares, $\{(\boldsymbol{c}_i, \boldsymbol{l}_i), i = 1, \ldots, m\}$, have already been disordered for transmission. In this case, even if the cloud learns published grid parameters, it cannot yet find top *k* coordinates of squares by taking the inverse procedure of transmission.

B. Secure Distance Metrics

In AEALV, CPS respectively encrypts $\boldsymbol{x}$ and $\boldsymbol{x}_i$ into $\boldsymbol{c}$ and $\boldsymbol{c}_i$, where $\boldsymbol{x}$ and $\boldsymbol{x}_i$ are the plaintext fingerprint vectors of the claimed position and grid squares i $(i = 1, \ldots, m)$, respectively. CPS also calculates the comparison matrix $\boldsymbol{H}$ from $\boldsymbol{SM} = \boldsymbol{I}^* + \boldsymbol{E}$, where $\boldsymbol{S}$, $\boldsymbol{M}$, $\boldsymbol{I}$, and $\boldsymbol{E}$ are the matrices of the secret key, key-transforming, identity, and error, respectively. CPS further uploads $\boldsymbol{c}$, $\boldsymbol{c}_i$, and $\boldsymbol{H}$ onto the cloud for outsourcing calculations of distance metrics between $\boldsymbol{c}$ and $\boldsymbol{c}_i$. The cloud then calculates the distances $\|\boldsymbol{x} - \boldsymbol{x}_i\|^2 \leftarrow \frac{1}{w^2}(\boldsymbol{c} - \boldsymbol{c}_i)^T \boldsymbol{H} (\boldsymbol{c} - \boldsymbol{c}_i)$, where w is the scaling parameter. To guarantee security, the distance metrics should leak no information about the fingerprint values, $\boldsymbol{x}$ and $\boldsymbol{x}_i$, and the secret matrix $\boldsymbol{S}$. Due to Theorem 1, the cloud cannot extract fingerprints directly from ciphertexts. Nonetheless, if given access to $\boldsymbol{H}$, the cloud is equipped with stronger capabilities. Even in this case, the distance metrics are still secure. Below, we demonstrate security of fingerprints and the secret key in Theorem 2 and Theorem 3, respectively.

Theorem 2: Given access to $\boldsymbol{c}$, $\boldsymbol{c}_i$, and $\boldsymbol{H}$, it is infeasible to recover $\boldsymbol{x}$ and $\boldsymbol{x}_i$.

Proof: Without loss of generality, we merely analyze the infeasibility of recovering $\boldsymbol{x}$ and the case of $\boldsymbol{x}_i$ is similar. The comparison matrix $\boldsymbol{H}$ leaks on information on the key-switching matrix $\boldsymbol{M}$; otherwise, given $\boldsymbol{M}$, we can recover the message $\boldsymbol{x}$ from $\boldsymbol{c}$ according to $\boldsymbol{c} = \boldsymbol{M}(w\boldsymbol{x})^*$. This reason is as follows. Firstly, we have

$$\begin{cases} \boldsymbol{AM} = \boldsymbol{I}^* \\ \boldsymbol{H} = \boldsymbol{A}^T \boldsymbol{A} \end{cases}. \tag{13}$$

At first glance, it is probable to recover $\boldsymbol{M}$ according to $\boldsymbol{I}^* = \boldsymbol{AM}$ by first extracting $\boldsymbol{A}$ from $\boldsymbol{H} = \boldsymbol{A}^T\boldsymbol{A}$. Nevertheless, the following analysis shows that only given access to $\boldsymbol{H}$, there are infinitely many $\boldsymbol{A}$ satisfying $\boldsymbol{H} = \boldsymbol{A}^T\boldsymbol{A}$, so does $\boldsymbol{M}$. There are infinitely many orthogonal bases $Q \in \mathbb{R}^n$ satisfying $\boldsymbol{I} = \boldsymbol{Q}^T\boldsymbol{Q}$. Consequently, we have

$$\begin{aligned} \boldsymbol{A}^T\boldsymbol{A} &= \boldsymbol{A}^T\boldsymbol{I}\boldsymbol{A} \\ &= (\boldsymbol{QA})^T(\boldsymbol{QA}). \\ &= \boldsymbol{A}^T\boldsymbol{Q}^T\boldsymbol{QA}. \end{aligned}$$

■

Theorem 3: Given access to $\boldsymbol{H}$, it is infeasible to extract $\boldsymbol{S}$ from $\boldsymbol{SM} = \boldsymbol{I}^* + \boldsymbol{E}$ assuming the hardness of LWE.

Proof: Assuming that $\boldsymbol{M}$ can be inferred from $\boldsymbol{H}$, then the problem of extracting $\boldsymbol{S}$ can be reduced to the hardness of LWE. We first assume that there is a solver $\mathcal{A}$ for the equation $\boldsymbol{SM} = \boldsymbol{I}^* + \boldsymbol{E}$. We then take each element in $\boldsymbol{S}$ and $\boldsymbol{M}$ from $\mathbb{Z}_q$ with $q >> max\{|\boldsymbol{S}|, |\boldsymbol{M}||\boldsymbol{E}|\}$. Further, we treat $\boldsymbol{S} \in \mathbb{Z}^{m \times n}$ as *m* *n*-dimensional row vectors as shown in Eq. (14). Note that $\boldsymbol{s}_j^T (j = 1, 2, \ldots, m)$ is an *n*-dimensional row vector.

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{s}_1^T \\ \boldsymbol{s}_2^T \\ \cdots \\ \boldsymbol{s}_m^T \end{bmatrix}_{m \times n} \tag{14}$$

Similarly, we treat $\boldsymbol{M} \in \mathbb{Z}^{n \times ml}$ as *ml* *n*-dimensional column vectors as shown in Eq. (15), where *l* is a binary representation parameter [26]. Note that $\boldsymbol{m}_j (j = 1, 2, \ldots, ml)$ is an *n*-dimensional column vector.

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{m}_1 & \boldsymbol{m}_2 & \cdots & \boldsymbol{m}_{ml} \end{bmatrix}_{n \times ml} \tag{15}$$

We further rewrite $\boldsymbol{I}^* \in \mathbb{Z}^{m \times ml}$ and $\boldsymbol{E} \in \mathbb{Z}^{m \times ml}$ as

$$\boldsymbol{I}^* = \begin{bmatrix} I_{1,1} & I_{1,2} & \cdots & I_{1,ml} \\ \cdots & \cdots & \cdots & \cdots \\ I_{m,1} & I_{m,2} & \cdots & I_{m,ml} \end{bmatrix}_{m \times ml} \tag{16}$$

$$\boldsymbol{E} = \begin{bmatrix} E_{1,1} & E_{1,2} & \cdots & E_{1,ml} \\ \cdots & \cdots & \cdots & \cdots \\ E_{m,1} & E_{m,2} & \cdots & E_{m,ml} \end{bmatrix}_{m \times ml} \tag{17}$$

According to $\boldsymbol{SM} = \boldsymbol{I}^* + \boldsymbol{E}$, we have equations as

$$\begin{cases} \boldsymbol{s}_1^T \boldsymbol{m}_1 = I_{1,1} + E_{1,1} \\ \boldsymbol{s}_1^T \boldsymbol{m}_2 = I_{1,2} + E_{1,2} \\ \cdots \\ \boldsymbol{s}_i^T \boldsymbol{m}_j = I_{i,j} + E_{i,j} \\ \cdots \\ \boldsymbol{s}_m^T \boldsymbol{m}_{ml} = I_{m,ml} + E_{m,ml} \end{cases} \tag{18}$$

There are $m \times ml$ samples of $(\boldsymbol{m}_j, I_{i,j})$ in

$$I_{i,j} = (\boldsymbol{s}_i)^T \boldsymbol{m}_j - E_{i,j}. \tag{19}$$

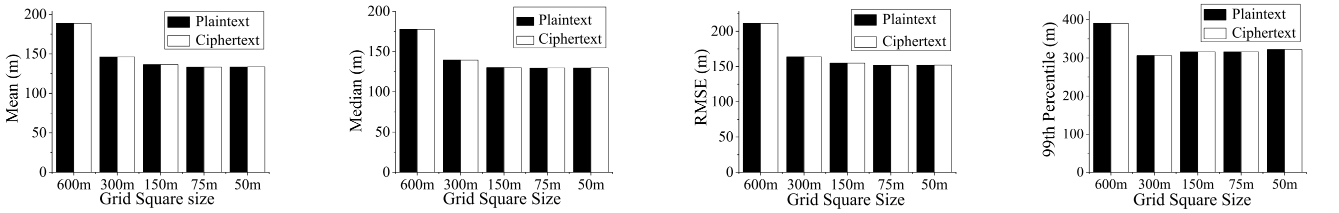
We can use the solver $\mathcal{A}$ to handle Eq. (19) for $\boldsymbol{s}_i^T$, which is equivalent to solving LWE $b_i = \boldsymbol{v}^T \boldsymbol{a}_i + \varepsilon_i$ for $\boldsymbol{v}^T$. Therefore, if the LWE problem is hard, it is also hard for solving $\boldsymbol{S}'\boldsymbol{M} = \boldsymbol{I}^* + \boldsymbol{E}$. As assumed at the beginning of the proof, $\boldsymbol{M}$ can be acquired given access to $\boldsymbol{H}$. Actually, from Theorem 2, we can obtain an infinite number of $\boldsymbol{M}$ from $\boldsymbol{H}$. Consequently, given access to $\boldsymbol{H}$, it is also infeasible to extract $\boldsymbol{S}$ from $\boldsymbol{SM} = \boldsymbol{I}^* + \boldsymbol{E}$. ■

VI. Performance Evaluation

We evaluate AEALV from accuracy, efficiency, and communication overhead.

A. Experiment Environment

To perform simulations, we use a graphic workstation from *Alibaba Cloud* with NVIDIA V100 GPU and 32 GB RAM running CUDA 10.1. The graphic workstation plays the role of the cloud server. The main functionality of CPS, when using TDOA, is to evaluate positions by utilizing



(a) Mean of horizontal error (b) Median of horizontal error (c) RMSE of horizontal error (d) 99th percentile of horizontal error

Fig. 8. Accuracy of estimating legitimate aircraft (k = 5, 3 sensors).

time-stamped ADS-B messages received at multiple ADS-B ground stations [8], [32]. In this case, CPS can be implemented based on an embedding device with an ARM processor. We use the embedding device with Kirin@1600MHz ARM processor and 2G RAM running Android 4.2.2, acting as CPS.

We set the system parameters in the experiments from the following considerations. First, for *GridParam*, the surveillance area is a grid plane with a longitude of 2 degrees, a latitude of 2 degrees, and an altitude of 11,000*m* covering the aircraft en-route flight phase at a cruising altitude. Moreover, the size of grid squares varies from 600*m* down to 50*m* to test accuracy and efficiency of AEALV. Second, for *VHEParam*, we take the security parameter as $\lambda = 128$ to ensure practical security. We also specify $w < 2^{30}$, $l = 50$ and $eBound = 200$, to guarantee the operation correctness in the ciphertext domain. Third, it is noteworthy that Strohmeier *et al.* [17] have experimented verifying that the optimal choice of neighbors $k = 5$ and the threshold $d_{th} = 500$ *m*. We accordingly specify *kNNParam*. Furthermore, we utilize the *OpenSky* flight data including over 300,000 ADS-B messages, each of which is received at least by three sensors. And our source code for experiments can be seen at [33].

B. Accuracy

We first use collected flight data to evaluate the accuracy of the location-estimating approach for legitimate aircraft. We then inject fake positions to test our quick identification technique for fake aircraft.

1) Accuracy of Estimating Legitimate Aircraft: For a legitimate aircraft, claimed positions are directly acquired from the global navigation satellite system and thus can be considered to be real. We evaluate the accuracy of location estimation by using real *Opensky* data. In particular, we implement the evaluation on the same datasets both for plaintexts and ciphertexts. By comparing such two implementations, we also check the correctness of AEALV. The evaluating results are illustrated in Fig. 8. Obviously, reducing the square size will have a positive effect, and the smaller the square, the more accurate the estimate. A reduction in the square size from 600*m* to 50*m* decreases mean errors by up to 29.22%. On the other hand, due to the noise of real data, the 99th percentile metric is more outlier-sensitive than the median error, mean error, and root mean squared error (RMSE). In specific, for a 600*m* grid square, the horizontal error between the real position and the estimated position is 390*m*. Consequently, we set the threshold $d_{th} = 500m$ in *kNNParam* for the subsequent location validation. More importantly, AEALV achieves almost the same accuracy compared with the baseline plaintext scheme. Nevertheless, AEALV has the substantial merit of privacy-protection, which is of high importance to secure aircraft position estimation in the cloud environment, but the plaintext scheme cannot yet provide this protection.

TABLE III
ACCURACY OF DETECTING FAKE AIRCRAFT (k = 5, 3 SENSORS)

Threshold $[m]$	**500**		**600**		**700**		**800**	
	Plx	**Cxt**	**Plx**	**Cxt**	**Plx**	**Cxt**	**Plx**	**Cxt**
Accuracy	1	1	1	1	0.93	0.93	0.5	0.5

Recently, Zhao *et al.* proposed a hybrid time-difference-of-arrival/angle-of-arrival (TDOA/AOA) positioning technology using extended Kalman filters for ADS-B positioning [34]. It can improve positioning accuracy and enhance surveillance robustness. Nevertheless, AEALV achieves an estimation error of RMSE 210*m* for the grid square of 600*m*, while it is 650*m* in [34]. More importantly, AEALV can preserve location privacy but the work in [34] cannot. Therefore, AEALV not only has a better estimation accuracy but also can protect the privacy of aircraft location, compared with the work in [34].

2) Accuracy of Detecting Fake Aircraft: First of all, we simulate one attacker sending fake locations. We first extract a real track composed of 500 positions from the *OpenSky* data and then resemble them as a fake track. In specific, we parse the *OpenSky* data according to the format of ADS-B messages, modify the corresponding field containing positional information, and regenerate the parity-check value [10]. Note that these resembled data are in consistence with the ADS-B message format, and thus capable of being parsed by ADS-B transponders, seeming that they are from a real aircraft. Further, these false messages follow the specific location pattern, where the real and fake tracks have the same starting position, but then horizontally diverge at a random angle between 5○ and 30○.

Subsequently, using the quick verification technique as described in Section IV-D, we check whether the fake locations can be verified or not. Concretely, we first calculate the attackers TDOA values, estimate the real aircraft locations and analyze the detection rate. Table III demonstrates the test results of the proposed location verification approach. For detecting fake aircraft, the proposed ciphertext (cxt) scheme

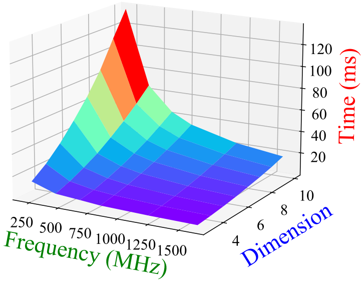


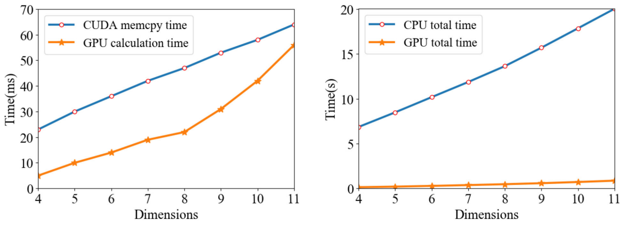
Fig. 9. CPS encrypting time (ms).

achieves the same classification accuracy, as the plaintext (plx) scheme. Note that $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$, in which *TP*, *TN*, *FP* and *FN* are *True Positive*, *True Negative*, *False Positive* and *False Negative*, respectively [35]. If the threshold is too large, the increase in *FN* leads to a decrease in *Accuracy*; if the threshold is too small, an increase in *FP* also leads to a decrease in *Accuracy*. The *Accuracy* drops down to 50% for the threshold of 800*m*. Accordingly, the threshold of 500*m* is a good choice to ensure the *Accuracy*.

C. Time Cost

The phases of *Initialization* and *Offline Training* are executed before starting validation, and thus they do not influence AEALV efficiency even if they may cost much time. Therefore, the efficiency capability is dependent on the time of the *Online Verification* phase. Especially, it mainly involves the time of CPS encrypting the claimed position and fingerprint of aircraft, the time of the cloud running the grid-based *kNN* algorithm, and the round-trip time of transmission between CPS and the cloud. As the transmission is via a dedicated line, the time may be negligible. In the following, the efficiency evaluation focuses on *CPS Encrypting Time* and *Cloud Running Time*. We finally evaluate the total time of the online verification process to analyze ALV performance in realistic ADS-B environments, in which multiple aircraft appear in the surveillance area at the same time.

CPS Encrypting Time: As aforementioned, when an aircraft flies into the designated surveillance area, CPS encrypts the fingerprint $\boldsymbol{x} \in \mathbb{Z}^n$ by invoking $\boldsymbol{c} \leftarrow VHE.Enc(\boldsymbol{x}, \boldsymbol{S}_1)$. Hence, the VHE encryption time of n-dimensional vector needs to be measured. In the surveillance grid plane, the number of sensors is usually no more than ten. On the other hand, if at most two sensors receive the same ADS-B message, then the TDOA-based method is inapplicable. Therefore, we set $3 \leq n \leq 10$. At the same time, considering that CPS is usually a resource-limited device, e.g., AX680 series provided by THALES, we evaluate such encryption time based on the simulation by an embedding device with the Kirin 910 ARM processor running the Android 4.2.2 system. We first explore Java 1.7.0 and NDK R9 to rebuild the VHE encryption codes and download them to the embedded device. Then, we obtained root privileges for the android system, so we can adjust the processor frequency by running APPs such as SetCPU. As shown in Fig. 9, we set the frequency to different values from 1596 MHz to 208 MHz. With the frequency decreases or the dimension increases, the encryption becomes less efficient. Nonetheless, even with the lowest frequency of 208 MHz, it takes only 136.09 ms to encrypt a 10-dimensional vector. Thus, the VHE encryption is much efficient suitable for the resource-constrained CPS.



(a) Memcpy and calculation (b) GPU and CPU

Fig. 10. Comparison on cloud running time.

Cloud Server Running Time: The main time of consumption on the cloud side is that of calculating $(\boldsymbol{c}-\boldsymbol{c}_i)^T\boldsymbol{H}_1(\boldsymbol{c}-\boldsymbol{c}_i)$ for massive ciphertexts, e.g., the number of ciphertexts is up to six million considering a 75*m* square. To guarantee efficiency of such calculations, we use a server from *Alibaba Cloud* with NVIDIA TESLA V100 GPU and 32 GB RAM, running CUDA 10.1. The V100 has 5120 CUDA cores and can provide sufficient computing cycles to large-scale parallel applications. On the other hand, the calculation of $(\boldsymbol{c}-\boldsymbol{c}_i)^T\boldsymbol{H}_1(\boldsymbol{c}-\boldsymbol{c}_i)$ with millions of items are independent of each other, and thus naturally suitable for parallelization. Consequently, it is an ideal application for GPU acceleration. In the following, we evaluate the cloud running time with six million data items.

In the calculations of $(\boldsymbol{c}-\boldsymbol{c}_i)^T\boldsymbol{H}_1(\boldsymbol{c}-\boldsymbol{c}_i)$, compared with vector-matrix multiplications, the time of doing vector subtraction is negligible. Therefore, we reformulate it as $\boldsymbol{v}_i^T\boldsymbol{M}\boldsymbol{v}_i$, where $\boldsymbol{v}_i, i = 1, \ldots, n$ is an m-dimensional vector and $\boldsymbol{M}$ is an $m \times m$ matrix. To leverage CUDA to implement such calculations in parallel, the contiguous memory space is firstly allocated for n data items of m dimensions, and then these data are copied from the memory to the GPU global memory. GPU then executes the task of $\boldsymbol{v}_i^T\boldsymbol{M}\boldsymbol{v}_i$ in the single instruction multiple threads (SIMT) mode. In particular, the matrix $\boldsymbol{M} \in \mathbb{R}^{m\times m}$ can be treated as $\boldsymbol{M} = [\boldsymbol{M}_1, \ldots, \boldsymbol{M}_m]$, where $\boldsymbol{M}_j$ is the j^{th} column of $\boldsymbol{M}$. Thus, the vector-matrix multiplication of $\boldsymbol{v}_i^T\boldsymbol{M}$ is split into m vector-vector multiplications as $\boldsymbol{v}_i^T\boldsymbol{M}_j, j = 1, \ldots, m$, thereby achieving higher parallelization. Finally, the results are copied back from the GPU global memory to the memory.

Fig. 10(a) illustrates a time comparison between CUDA memory copying and GPU calculating. It is easy to see that both of them rise with dimensions. However, the former linearly increases, and the latter does quadratically. Besides, for vectors with low dimensions, the former is usually longer than the latter. Especially, the gap becomes more significant as the memory copying occurs many times in one calculation. Hence, to speed up memory copying, we use the page-locked technique enabling faster copying. Also, note that in the dimension of 8, the time of GPU calculation increases a little slower. This reason is due to the limitation of the data organization structure in GPU, if more than eight dimensions, the number of warps in a block may exceed the upper limit of the simultaneous

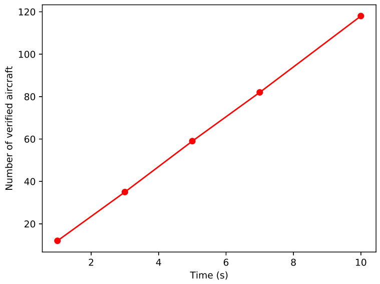


Fig. 11. Verification capacity of ALV.

operation of SM (streaming multiprocessor), resulting in a quicker increase in the calculating time.

Finally, we give a comparison of total time (including copy and calculation) between CPU and GPU. As shown in Fig. 10(b), for the same task, the calculating based on GPU is much faster than that on CPU. Especially, even for the slowest with 11-dimensional data items, the total time of GPU is 683 ms (617 ms for CUDA memory copy and 56ms for GPU calculation), which is far fewer than that of CPU (more than 20s), only about $\frac{1}{30}$. Consequently, AEALV is efficient for aircraft location verification with the high-performance GPU.

Total Verification Time: The online verification process mainly includes CPS encrypting and cloud server running as shown in Fig. 5. Regarding real-world cases, five ADS-B receivers are sufficient for aircraft location verification, so fingerprint vectors are 5-dimensional. As illustrated in Fig. 9, with the lowest CPS frequency of 208 MHz, it only costs 34 ms for encrypting a 5-dimensional vector. After encryption, the 5-dimensional fingerprint vector becomes a 6-dimensional ciphertext vector, which is then sent to the cloud server. For a 6-dimensional vector, the time of cloud running the kNN algorithm is 50 ms as shown in Fig. 10(a). Consequently, the total verification time is less than 85 ms. In a realistic scenario, there may be multiple aircraft simultaneously occurring in the surveillance airspace. Hence, we need to evaluate the number of aircraft validated at the same time. Fig. 11 points such the number with a given time. In 3 seconds, AEALV can complete the online verification of about 35 aircraft. Thus, it has a good performance even in busy airports and peak hours.

D. Communication Cost

We only analyze communication cost in the phase of *Online Verification* since it may affect validating efficiency. Such communications include two parts. Firstly, CPS uploads the encrypted fingerprint of aircraft to the cloud; secondly, the cloud returns the encrypted estimated location to CPS. As the total amount of data transmission is less than 1 KB, the communication cost is ignorable concerning high-speed dedicated lines.

VII. Related Work

Many works have studied the secure location validation problem in conventional wireless broadcast networks [7], [12]–[14], [36]–[38]. However, there exist distinct features such as *vast distance*, *outdoor line-of-sight* and *few multi-path effect* in aircraft surveillance networks. Although the TDOA system has limited performance in the indoor environment (due to multi-path effect [2]), they perform well in the long-distance, outdoor and line-of-sight environment suitable for air traffic control. Therefore, some aircraft location validation approaches based on TDOA have been proposed [8], [17]. Specifically, the grid-based *k*NN scheme suggests effective localization even with low-cost hardware (e.g., SBS-3) and arbitrary receiver placement, which is beneficial to extensive deployment [17]. Unfortunately, to achieve high accuracy, the number of grid squares are required to be exceptionally large. Consequently, the efficiency of localization cannot be guaranteed owing to huge computation overhead, which would bring severe risks to flight safety. Although the computationally intensive ALV can be outsourced to the powerful cloud, the cloud is generally not considered to be fully trusted. Hence, aircraft location privacy has become an urgent issue. Researchers have proposed many privacy protection methods for aircraft locations [10], [21], [39]–[41]. Some of them require the use of pseudonyms to break the link between the location and the aircraft identifier [10], [39]. Sampigethaya and Poovendran [39] recommended using a pseudonym in ADS-B broadcasts to provide anonymity of location data, because the pseudonym can prevent access to the aircraft's ICAO address, which may be regarded as the real identity of the aircraft for unique identification. Such pseudonym methods do not seem to disclose the aircraft's real identity. However, there may be a temporal and spatial correlation between the locations of the aircraft. In this case, pseudonyms are not capable of preventing location tracking, and thus cannot ensure aircraft location privacy. Random silent period is another method to enhance aircraft location privacy. Sampigethaya *et al.* [21] exploited this method to alleviate the spatial and temporal correlation, thereby mitigating location tracking. Unfortunately, the introduction of random silent periods prolongs the broadcast cycle of ADS-B, that is, reduces the timely availability of aircraft traffic signals, which may degrade airspace security. Recently, Smith *et al.* [41] explored the concept of blocked aircraft to protect aircraft location privacy by obscuring aircraft movements and communications. However, a moderate attacker with only one set of sensors can still learn a lot about aircraft locations. Additionally, Andrés *et al.* [40] defined the differentially private geo-indistinguishability for location data and presented a perturbation technique of adding random noise to locations. Nevertheless, the location verification upon disturbed data may not be accurate enough. Consequently, it is imperative to enable privacy-preserving aircraft location verification in the cloud while maintaining accuracy.

VIII. Conclusion

In this paper, we have proposed an AEALV scheme for ADS-B by using grid-based *k*NN that exploits the cloud server to handle massive training squares and improve the accuracy of aircraft location verification. Security analysis has proven that AEALV protects the privacy of both aircraft location and grid data. We have also presented a quick identification technique for aircraft legitimacy that saves the verification time significantly. Performance evaluation has exhibited that AEALV achieves high efficiency on both embedding ARM processors and cloud servers. Even for six million squares, the time of evaluating encrypted distances, including GPU calculating time and CUDA memory copying time, is no more than 200 ms. As a consequence, AEALV ensures accurate and efficient aircraft location verification in a privacy-protecting fashion. In the future, we will explore more ADS-B security problems, such as secure broadcast authentication and physical layer intrusion detection.

References

[1] R. A. Valdés, V. F. G. Comendador, A. R. Sanz, and J. P. Castán, *Aviation 4.0: More Safety Through Automation and Digitization*. London, U.K.: IntechOpen, 2018.

[2] M. Strohmeier, M. Schafer, V. Lenders, and I. Martinovic, "Realities and challenges of nextgen air traffic management: the case of ADS-B," *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 111–118, May 2014.

[3] J. Sun, F. Liu, Y. Zhou, G. Gui, T. Ohtsuki, S. Guo, and F. Adachi, "Surveillance plane aided air-ground integrated vehicular networks: Architectures, applications, and potential," *IEEE Wireless Commun.*, vol. 27, no. 6, pp. 122–128, Dec. 2020.

[4] G. Gui, F. Liu, J. Sun, J. Yang, Z. Zhou, and D. Zhao, "Flight delay prediction based on aviation big data and machine learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 140–150, Jan. 2020.

[5] G. Gui, Z. Zhou, J. Wang, F. Liu, and J. Sun, "Machine learning aided air traffic flow analysis based on aviation big data," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 4817–4826, May 2020.

[6] J. Tan, Y.-C. Liang, N. C. Luong, and D. Niyato, "Toward smart security enhancement of federated learning networks," *IEEE Netw.*, vol. 34, no. 1, pp. 340–347, Jan./Feb. 2021, doi: 10.1109/MNET.011.2000379.

[7] G. Yang, Y.-C. Liang, R. Zhang, and Y. Pei, "Modulation in the air: Backscatter communication over ambient OFDM carrier," *IEEE Trans. Commun.*, vol. 66, no. 3, pp. 1219–1233, Mar. 2018.

[8] M. Strohmeier, V. Lenders, and I. Martinovic, "On the security of the automatic dependent surveillance-broadcast protocol," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 1066–1087, 2nd Quart., 2015.

[9] D. Moser, P. Leu, V. Lenders, A. Ranganathan, F. Ricciato, and S. Capkun, "Investigation of multi-device location spoofing attacks on air traffic control and possible countermeasures," in *Proc. ACM MobiCom*, 2016, pp. 375–386.

[10] H. Yang, Q. Zhou, M. Yao, R. Lu, H. Li, and X. Zhang, "A practical and compatible cryptographic solution to ADS-B security," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3322–3334, Apr. 2019.

[11] S. Sciancalepore and R. Di Pietro, "SOS: Standard-compliant and packet loss tolerant security framework for ADS-B communications," *IEEE Trans. Depend. Secure Comput.*, early access, Aug. 14, 2020, doi: 10.1109/TDSC.2019.2934446.

[12] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-based fingerprinting for indoor localization: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 763–776, Jan. 2017.

[13] S. B. Cruz, T. E. Abrudan, Z. Xiao, N. Trigoni, and J. Barros, "Neighbor-aided localization in vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 10, pp. 2693–2702, Oct. 2017.

[14] J. Luo, Z. Zhang, C. Wang, C. Liu, and D. Xiao, "Indoor multifloor localization method based on WiFi fingerprints and LDA," *IEEE Trans. Ind. Informat.*, vol. 15, no. 9, pp. 5225–5234, Sep. 2019.

[15] D. Adesina, O. Adagunodo, X. Dong, and L. Qian, "Aircraft location prediction using deep learning," in *Proceedings of IEEE MILCOM*, 2019, pp. 127–132.

[16] Y. Zhao, Y. Yin, and G. Gui, "Lightweight deep learning based intelligent edge surveillance techniques," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 4, pp. 1146–1154, Dec. 2020, doi: 10.1109/TCCN.2020.2999479.

[17] M. Strohmeier, I. Martinovic, and V. Lenders, "A k-NN-based localization approach for crowdsourced air traffic communication networks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 3, pp. 1519–1529, Jun. 2018.

[18] P. Vagdevi and H. Guruprasad, "A study on cloud computing in aviation and aerospace," *Int. J. Comput. Sci. Eng. Technol.*, vol. 6, no. 3, pp. 94–98, 2015.

[19] T. Larsen, "Cross-platform aviation analytics using big-data methods," in *Proc. IEEE ICNS*, 2013, pp. 1–9.

[20] H. Ren, H. Li, D. Liu, G. Xu, N. Cheng, and X. Shen, "Privacy-preserving efficient verifiable deep packet inspection for cloud-assisted middlebox," *IEEE Trans. Cloud Comput.*, early access, Apr. 29, 2020, doi: 10.1109/TCC.2020.2991167.

[21] K. Sampigethaya, R. Poovendran, and C. S. Taylor, "Privacy of general aviation aircraft in the nextGen," in *Proc. IEEE DASC*, 2012, p. 7.

[22] E. G. Modrego, M.-G. Iagaru, M. Dalichampt, and R. Lane, "Airport CDM network impact assessment," in *Proc. ATM*, 2009, pp. 459–468.

[23] M. Grissa, A. A. Yavuz, and B. Hamdaoui, "Location privacy in cognitive radios with multi-server private information retrieval," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 4, pp. 949–962, Dec. 2019.

[24] P. Paillier and D. Pointcheval, "Efficient public-key cryptosystems provably secure against active adversaries," in *Proc. ASIACRYPT*, 1999, pp. 165–179.

[25] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. ACM STOC*, 2009, pp. 169–178.

[26] H. Zhou and G. Wornell, "Efficient homomorphic encryption on integer vectors and its applications," in *Proc. IEEE ITA*, 2014, pp. 1–9.

[27] D. Demmler, T. Schneider, and M. Zohner, "ABY–A framework for efficient mixed-protocol secure two-party computation," in *Proc. NDSS*, 2015, pp. 1–16.

[28] M. Strohmeier, V. Lenders, and I. Martinovic, "Lightweight location verification in air traffic surveillance networks," in *Proc. 1st ACM Workshop Cyber-Phys. Syst. Security*, 2015, pp. 49–60.

[29] S. Bogos, J. Gaspoz, and S. Vaudenay, "Cryptanalysis of a homomorphic encryption scheme," *Cryptography Commun.*, vol. 10, no. 1, pp. 27–39, 2018.

[30] H. Yang, S. Liang, J. Ni, H. Li, and X. Shen, "Secure and efficient k NN classification for industrial Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 10945–10954, Nov. 2020. doi: 10.1109/JIOT.2020.2992349.

[31] Z. Brakerski, C. Gentry, and S. Halevi, "Packed ciphertexts in LWE-based homomorphic encryption," in *Publ. Key Cryptography (PKC)*, pp. 1–13, Jan. 2013.

[32] M. Strohmeier, M. Schäfer, M. Smith, V. Lenders, and I. Martinovic, "Assessing the impact of aviation security on cyber power," in *Proc. 8th IEEE Int. Conf. Cyber Conflict*, 2016, pp. 223–241.

[33] Q. Zhou. *Secure Aircraft Location Verification*. Accessed: Oct. 2, 2019. [Online]. Available: https://github.com/qxzhou1010/SecureAircraftLocationVerification

[34] D. Zhao, J. Sun, and G. Gui, "En-route multilateration system based on ADS-B and TDOA/AOA for flight surveillance systems," in *Proc. IEEE VTC*, 2020, pp. 1–6.

[35] C. Sammut and G. I. Webb, *Encyclopedia of Machine Learning*. Boston, MA, USA: Springer, 2011.

[36] Y.-C. Liang, Q. Zhang, E. G. Larsson, and G. Y. Li, "Symbiotic radio: Cognitive backscattering communications for future wireless networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 4, pp. 1242–1255, Dec. 2020, doi: 10.1109/TSMC.2020.3002566.

[37] G. Yang, Q. Zhang, and Y.-C. Liang, "Cooperative ambient backscatter communications for green internet-of-things," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1116–1130, Apr. 2018.

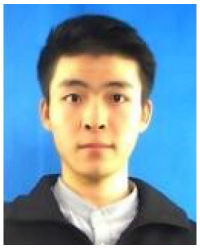
[38] D. Liu, J. Ni, C. Huang, X. Lin, and X. S. Shen, "Secure and efficient distributed network provenance for IoT: A blockchain-based approach," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7564–7574, Aug. 2020.

[39] K. Sampigethaya and R. Poovendran, "Enhancing location privacy of future aircraft wireless communications," in *Proc. ATIO*, 2010, pp. 1–13.

[40] M. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proc. ACM CCS*, 2013, pp. 901–914.

[41] M. Smith, D. Moser, M. Strohmeier, V. Lenders, and I. Martinovic, "Undermining privacy in the aircraft communications addressing and reporting system (ACARS)," *Privacy Enhancing Technol.*, vol. 2018, no. 3, pp. 105–122, 2018.

Haomiao Yang (Member, IEEE) received the M.S. and Ph.D. degrees in computer applied technology from the University of Electronic Science and Technology of China, China, in 2004 and 2008, respectively, where he is currently an Associate Professor with the School of Computer Science and Engineering. He worked as a Postdoctoral Fellow with the Kyungil University from June 2012 to June 2013. He also worked as a Visiting Scholar with the University of Waterloo from December 2018 to December 2019. His research interests include cryptography, cloud security, and cyber security for aviation communication.



Qixian Zhou received the B.S. degree in biomedical engineering from Southwest Medical University, Luzhou, China. He is currently pursuing the M.S. degree in computer science with the University of Electronic Science and Technology of China. His current research interests include data security and artificial intelligence security.

Dongxiao Liu (Member, IEEE) received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Canada, in 2020. He is a Postdoctoral Research Fellow with the Department of Electrical and Computer Engineering, University of Waterloo. His research interests include mobile computing and blockchain.

Hongwei Li (Senior Member, IEEE) received the Ph.D. degree from the University of Electronic Science and Technology of China, China, in 2008, where he is currently the Department Head and a Professor with the Department of Information Security, School of Computer Science and Engineering. He worked as a Postdoctoral Fellow with the University of Waterloo from October 2011 to October 2012. He has published more than 80 technical papers. His research interests include network security and applied cryptography. He serves as an Associate Editor of the IEEE INTERNET OF THINGS JOURNAL and *Peer-to-Peer Networking and Applications*, the Guest Editor of the IEEE NETWORK and the *IEEE Internet of Things Journal*. He won the Best Paper Award from the IEEE MASS 2018 and the IEEE HELTHCOM 2015. He is the Distinguished Lecturer of IEEE Vehicular Technology Society.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990.

He is currently an University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research focuses on resource management in interconnected wireless/wired networks, wireless network security, social networks, smart grid, and vehicular ad hoc and sensor networks. He received the R.A. Fessenden Award in 2019 from IEEE, Canada, the James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, the Joseph LoCicero Award in 2015, and the Education Award in 2017 from the IEEE Communications Society. He has also received the Excellent Graduate Supervision Award in 2006, and the Outstanding Performance Award five times from the University of Waterloo and the Premier's Research Excellence Award in 2003 from the Province of Ontario, Canada. He served as the Technical Program Committee Chair/Co-Chair for the IEEE Globecom'16, the IEEE Infocom'14, the IEEE VTC'10 Fall, the IEEE Globecom'07, the Symposia Chair for the IEEE ICC'10, the Tutorial Chair for the IEEE VTC'11 Spring, the Chair for the IEEE Communications Society Technical Committee on Wireless Communications, and P2P Communications and Networking. He is the President of the IEEE Communications Society. He is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society.