DarkSky: Privacy-Preserving Target Tracking Strategies Using a Flying Drone

Samhith Reddy Chinthi-Reddy[†] Sunho Lim[†*] Gyu Sang Choi[‡] Jinseok Chae[§*] Cong Pu[¶]
[†]T^{2}WISTOR: TTU Wireless Mobile Networking Laboratory, Dept. of Computer Science,
Texas Tech University, Lubbock, TX 79409, {samhith-reddy.chinthi-reddy, sunho.lim}@ttu.edu
[‡]Dept. of Information and Communication Engineering, Yeungnam University,
Gyeongsan 38541, Korea, castchoi@ynu.ac.kr
[§]Dept. of Computer Science and Engineering, Incheon National University, Incheon 22012, Korea, jschae@inu.ac.kr
[¶]Dept. of Computer Sciences and Electrical Engineering, Marshall University, Huntington, WV 25755, puc@marshall.edu

***Abstract*—Commercially well-known drones, unmanned aerial vehicles (UAVs), are increasingly popular with the public and have been widely deployed in diverse applications. However, a drone equipped with tracking, monitoring, or sensing device(s) can illegally collect privacy- and security-sensitive information and intrude restricted areas. Thus, recent literature focuses on the protection of users and restricted areas from an unwanted privacy attack and intrusion caused by the drone. Unlike prior research, however, we fundamentally shift the privacy paradigm from protecting users and restricted areas from a malicious drone into protecting and hiding the sensitive information of a drone from an adversary. In light of these, we propose three privacy-preserving target tracking strategies based on the shortest path, random locations, and dummy locations. The basic idea is to obfuscate the current location of the drone and randomize the trajectory to prevent the adversary from locating and tracking the drone. We also analyze drone privacy in terms of location and trajectory and measure them through entropy-based anonymity, the size of convex-hull, and the number of paths. We conduct extensive simulation experiments using the OMNeT++ for performance evaluation and comparison with stationary and moving target tracking scenarios under three mobility models. The simulation results indicate that the proposed strategies can be a viable approach to track the target while preserving a certain level of location and trajectory privacy.**

***Index Terms*—Drone, Location privacy, Target tracking, Unmanned Aerial Vehicle (UAV)**

I. Introduction

Commercially well-known drones, unmanned aerial vehicles (UAVs), are increasingly popular to the public because of their versatility, easy installation and interface, and relatively low operating cost. Drones are being evolved to be smaller, lightweight, low-altitude navigated, and user-friendly because of recent technological advances in embedded computing and high-speed wireless networking. Diverse drone-based applications have been developed in military and civilian environments. For example, reconnaissance and surveillance, post-disaster search and rescue, extending ad hoc networks, an aerial inspection of industrial infrastructures, freight and package delivery services, deterring birds from entering the airspace around airports, and so on [1]. The federal aviation administration (FAA) has more than 0.8 million commercial and recreational drones registered with the organizations in 2020 [2]. The business market for commercial drones is expected to grow by $13 billion by 2025 [3]. As drones rapidly become ubiquitous, however, it is not surprising to frequently witness them flying over private properties, restricted areas, or cyber-critical infrastructures. Drones equipped with a camera, sensor, or radar can easily track, monitor, sense objects and areas, and illegally collect privacy- and security-sensitive information. In light of these, recent research efforts [4], [5], [6], [7], [8] have been devoted to the protection and prevention of users from an unwanted privacy attack and the expulsion of drones from entering the restricted area.

Unlike prior literature, we raise a novel issue on *privacy* in conducting drone-based target tracking operations in this paper. Our research goal is for a drone to successfully complete a given target tracking mission without exposing its location, trajectory, and path information to an adversary. For example, if the drone follows a simple path (e.g., shortest path) towards a target, the adversary could collude with the target, locate the drone, predict a future trajectory, and identify a path, and take a counteraction in advance. Since the location and trajectory information of the drone is critical in target tracking operations, the drone should keep this information secret and should not share them with anyone except the trusted one. To the best of our knowledge, our proposed approach is the first in exploring the privacy of drone itself in the realm of UAV research. Our contributions are summarized in three-fold:

- We first fundamentally shift a privacy paradigm of drone from protecting users and restricted areas from a malicious drone into protecting and hiding sensitive information of the drone from an adversary.
- Three privacy-preserving target tracking strategies are proposed based on the shortest path, random location, and dummy location. The basic idea is to obfuscate the current location of the drone and randomize the path towards the target to prevent the adversary and target from locating and tracking the drone. We also enhance the strategies by varying the method in selecting a set of dummy locations.
- The privacy of drone is analyzed in terms of two aspects:

[*] Co-corresponding authors

location and trajectory. We deploy an entropy-based metric to quantify the degree of anonymity for the location privacy of drone. We also consider a convex hull area and the number of paths to measure the trajectory privacy of drone.

In this paper, we develop a customized discrete-event driven simulator using the OMNeT++ [9]. We extensively conduct the performance evaluation and comparison of three target tracking strategies in terms of entropy-based anonymity, the size of convex hull area, the number of paths, drone traces, detection delay, and the average number of packets exchanged with the server. We consider both stationary and mobile targets with three mobility scenarios: linear, circular, and random waypoint. The simulation results show that the proposed target tracking strategies can preserve the location and trajectory privacy of drone and be a viable approach.

The rest of this paper is organized as follows. Prior literature is reviewed in Section II. We investigate and analyze privacy-preserving target tracking strategies in Section III. The proposed strategies are extensively evaluated and discussed in Sections IV and V, respectively. The concluding remarks are presented in Section VI.

II. Related Work

Privacy-preserving research has been extensively studied in various wireless and/or mobile networks. For example, in mobile ad hoc networks (MANETs), due to their inherent resource constraints and lack of centralized access control, each node periodically broadcasts a beacon message at least containing its identification and location to advertise its presence for peer-to-peer communication [10]. An adversary can capture on-flying messages and track and learn the traces of mobile nodes' location. To avoid the location privacy revealed, several privacy-preserving routing algorithms are proposed to protect both sender and receiver from leaking their location privacy [11], [12]. Privacy-enhancing techniques including anonymity sets, multiple routes, hiding routing control messages, etc. are embedded into routing algorithms so that the adversary can be confused to locate the exact source and destination. In mobile social networks (MSNs), users can communicate and share private information and data with others flexibly through one-to-one, one-to-many, and many-to-many ways. To prevent the adversary from stealing private social data, diverse privacy-preserving strategies are investigated primarily targeting location-based services, social routing, social profile, and social morality, social health, and service-oriented sociality [13]. In this section, we analyze major privacy-preserving research activities of users and drones and focus on their location privacy.

Drone Path Planning: Diverse path planning strategies have been widely investigated. The primary goal of path planning is to find an optimal path or good quality of route that satisfies desired performance objectives, such as maximum traversability and safety, minimum cost, or shortest navigating time and route. A simple shortest path and its numerous variants have been proposed to drive the minimum target cost, using *A* search, heuristic approach, or Dijkstra's algorithm [14], [15]. Path planning can be classified based on how environmental information is utilized to compute an optimal path: (i) Global path planning and (ii) Local path planning [16]. Global path planning uses a global geographical map to find an optimal path. Both heuristic searching methods and intelligent algorithms are deployed [17], [18], [19]. When a sudden obstacle appears or an unexpected event occurs, the local path planning constantly collects the sensed information from the surrounding environment and adjusts the current optimal path using an evolutionary algorithm or machine learning techniques [20], [21].

Drone-based Target Tracking: A drone or a group of drones equipped with a camera or sensor monitors the environment, communicates with a base or other drones, and detects an object or event that occurred on the ground. Most approaches [22], [23], [24] focus on how to efficiently extract meaningful information from the surrounding scenes and improve the quality of object detection using computer vision algorithms or deep learning techniques. Since a recent vision camera is light-weight, high-resolution, and cost-efficient, processing and analyzing a series of incoming high-resolution images from a flying drone in a real-time manner become critical under potential obstacles, such as multiple objects or environmental uncertainties.

Privacy-Attacking and -Preserving Drone: Each drone equipped with a camera, sensor, or radar is capable of conducting a privacy attack by tracking, monitoring, and sensing an object or event to illegally collect privacy- and security-sensitive information in personal properties, restricted areas, or cyber-critical infrastructures. Recent approaches [5], [6], [7], [8] focus on the protection of users and restricted areas from unwanted intrusion and attack caused by a drone. In [8], both drone's altitude and its attached equipment capability are regulated and limited to reduce a privacy attack. In [25], the fifth generation (5G) enabled drones to deploy a blockchain-based consensus technique to protect data and trajectory privacy from a malicious drone. The malicious drone may conduct misbehavior by disrupting a drone network and intercepting data transmissions between legitimate drones. Recent privacy-preserving approaches [26], [27] are also not to intrude on any privacy-sensitive area. In [26], multiple drones cover an area, where the area is virtually divided into a set of square-shaped public and private sub-areas. Each drone generates a graph for navigation not to intrude private sub-areas by considering both plan and elevation views and flies through the center of squares in a straight-line based. In [27], a path planning algorithm is proposed for the drone not only to cover a given target area without violating privacy-sensitive areas but to minimize total navigation time. Note that the drone is still considered as an object that could intrude and attack users and security-sensitive areas.

Preserving User's Location Privacy: When a user requests a location-based service (LBS) in an infrastructure-based network, it sends a spatial query piggybacked with its current location to an LBS server. In this querying, it is implicitly assumed that the user has agreed to share its location with the server without knowing whether the server could fully be trusted or not. To protect the location privacy of users from an untrusted server, diverse techniques including location perturbation, obfuscation, anonymity, and their variants have been deployed [28], [29]. The basic idea is that a set of real or dummy locations including a query issuing user and its adjacent neighbors is sent to the server as a part of query parameters to hide the exact location of the user from the server [30], [31]. To achieve a k-anonymity, a queried location is often geographically enlarged into a cloaking region to encompass at least k - 1 adjacent neighbors or dummy locations. However, the k-anonymity may require extra computation and reduce the accuracy of query results replied from the server [32], [33]. A virtual circle or square shape of cloaking region can be created to include the locations of user and k - 1 dummies [32].

User Location Tracking: Whenever users access an infrastructure-based network, they randomly generate a valid, unlinkable, and short-term lived medium access control (MAC) address[1] to avoid their locations from being tracked by an adversary [34]. To mitigate locations being tracked, users update their ids in a specific period (e.g., when changing a velocity or direction) or location (e.g., when arriving at an intersection or pre-determined area) to reduce the adversary's ability to continuously infer and track their locations [35]. Users can even keep quiet for a random period before updating their ids to confuse the adversary [36]. In [37], users can opportunistically exchange their ids with the adjacent neighbors to increase the k-anonymity.

In summary, extensive research efforts have been devoted to drones and their diverse applications over the past decades, but drones have not been considered as an object whose location privacy should be protected. To the best of our knowledge, this is the first attempt to explore the location privacy of the drone for target tracking in the realm of UAV research. This research is also to shift a privacy paradigm of the drone from protecting users and security-sensitive areas/facilities from an intruding drone into protecting the location privacy of drone from a stationary or mobile adversary.

III. Privacy-Preserving Target Tracking Strategies

We shift the paradigm of privacy on drones from protecting the privacy of users from an intruding drone to protecting the location privacy of drones from an adversary. The primary goal of our approach is for a drone to complete a mission by tracking and detecting a target without exposing its current location and future trajectory to an adversary. In the following, we present system and adversarial models, propose three target tracking strategies with stationary and moving target scenarios, and analyze the privacy of drones in terms of location and trajectory.

A. System and Adversarial Models

A user[2] controls a drone[3] via a wireless local area network (e.g., IEEE 802.11) in a base, where the drone is launched and landed back after completing a mission. The user also communicates with a server through the fourth-generation (4G) wide area wireless networks (e.g., LTE). Thus, the user can communicate with the server and drone simultaneously. The drone is equipped with multiple network interface cards (e.g., IEEE 802.11 and LTE) and an onboard geographical position system (GPS). The drone can also be attached with any device for detection depending on a given mission, such as a camera, sensor, or radar. Here, each device has a different detection range. Due to the inherent resource constraints, however, the drone has limited computing power, memory storage, and battery power. The drone periodically or on-demand basis reports the current location/status to the user and follows the command issued by the user. The drone can be affected by weather (e.g., wind) and may not correctly navigate to the location requested by the user. The server is accessible by both user and drone directly through the 4G wireless networks.

A target can be a static or mobile object (or point-of-interest). A static object is stationary and never changes or does not change its location in a short period. A mobile object refers to a moving object and its location is time-varying, such as a pedestrian, vehicle, or drone. In this paper, the target is considered to be detected if it is located within a detection range of a drone. As shown in Subfig. 1(a), a user communicates with a drone directly if they are located within their communication ranges. If the drone is far away from the user and becomes out of sight, both user and drone can still communicate together through a server indirectly. The server is located between the user and drone, relays their communication, and answers a query generated by the user or drone. Due to the recent development of tracking technologies, e.g., a satellite video technique [38], the server is able to locate the target and provide both user and drone with an approximated location of the target. Here, the technologies often suffer from their inherent constraints, such as inaccurate satellite video cameras, video filtering algorithms, or GPS error bounds.

An adversary is to attack our target tracking operation by interfering with network protocols or intercepting on-fly communications. Unlike a drone, the adversary is assumed to stay in an active mode for an extended period without resource constraints. The adversary is able to compromise a server to behave maliciously. Then the malicious server

[1] Each user can be uniquely identified by a 48-bit long MAC address that is inherently burned into its carrying device under the IEEE 802 standard. Here, the MAC address is used as a user identifier (*id*) as well.

[2] In this paper, we use a *user* to refer to a person who carries and controls a wireless/mobile device or a drone wirelessly.

[3] In general, both terms, *UAV* and *drone*, are used interchangeably. For clarity, we use the term, drone, for general users to easily install, deploy, and control it in their applications in this paper. Unlike UAV, drones are often characterized by small-size, lightweight, low-altitude navigation, and user-friendliness.

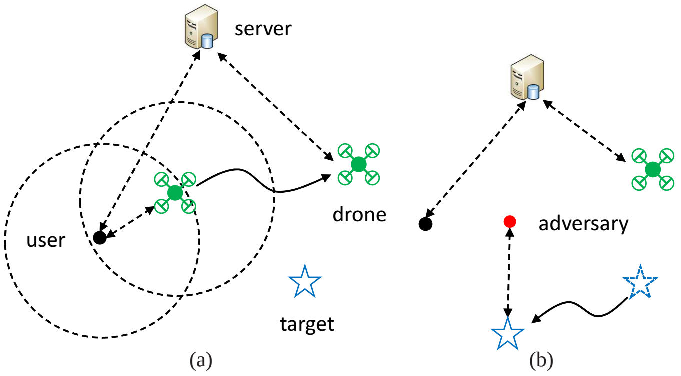


Fig. 1. (a) A user marked as a black circle communicates with a drone directly or indirectly through a server. The communication ranges of the user and drone are marked as a dotted line circle. A solid line indicates a mobility; and (b) An adversary and a target are marked as a red circle and a star, respectively. The adversary can overhear ongoing communications and collude with the target.

may selectively or strategically forward the current location of the drone to the adversary. Both target and adversary could collude together to avoid being tracked by the drone. In Subfig. 1(b), the adversary might eavesdrop on an on-flying packet and inject false information or modify the packet header information to mislead both the user and drone. However, if the user authenticates an incoming packet with a lightweight digital signature [39], it can verify and detect any modification. In this paper, we consider a target tracking operation and its potential adversarial scenarios that cannot be detected by digital signature and cryptographic techniques. We do not consider cryptographic primitives.

B. Target Tracking Strategies

In this paper, we use the term *path* as a spatial construct for the drone to reach from the user to the target. Since the drone flies through a set of designated locations as an intermediate destination to detect the target, we use the term *trajectory* to refer to a movement from an intermediate location to the next along the path. Here, a path consists of several trajectories.

Shortest Path-based Tracking: To locate and track the target efficiently, it is essential for the user, server, and drone to communicate and collaborate seamlessly. With the current location of the target provided by the server, the drone can observe and track the target to quickly respond and take an action for the event that occurred in the ground under the guidance of user.

We propose a simple tracking approach based on the shortest path between drone and target. When a user launches a drone to track a target, it sends a query to a server for the whereabouts of the target. The server replies an approximated location of the target (x_t, y_t) to the user using tracking technologies. Upon receiving the reply, the user uploads the location to the drone and launches it. The drone can calculate a straight line with a slope, $y = \frac{y_t - y_d}{x_t - x_d}x$, where ($x_d$, y_d) is the current location of the drone. The drone flies towards the target by following the line. Since the line is considered as the shortest path between the drone and target, the drone can detect the target quickly. If the target is stationary or does not move in a short period, however, the path can easily be predictable by an adversary. Since the adversary could collude with the server, the target may take an action in advance to avoid being tracked. Here, we define a detection delay as an elapsed time measured from when a drone is launched to when a target is detected.

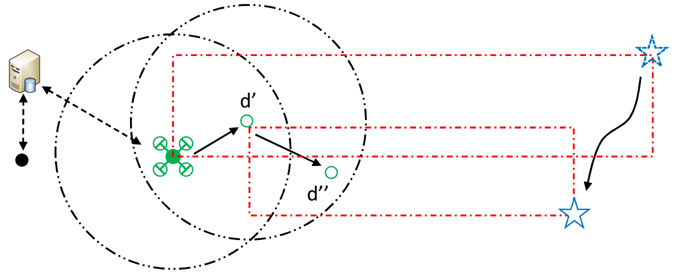


Fig. 2. A random location-based target tracking approach, where the rectangles are marked as a dashed-dot line and are virtually built with a diagonal starting from the current location of the drone to the target location. A randomly generated location, marked as a green circle, is used as an intermediate destination (e.g., d' and d''). When a drone arrives at an intermediate destination, it builds a new rectangle and generates a random location as the target moves.

While the drone is flying towards the target, it periodically sends additional queries to the server for any update of the target location. Upon receiving the queries, the server replies the updated target locations to both user and drone. Since the server might be compromised by an adversary and show misbehavior by replying a false target location, the user should be updated to monitor the responses of the server and track the target. The user should also monitor the current location of the drone to check the process of target tracking. Thus, the drone includes its current location in the query and sends it to the server. To avoid the drone from being exposed its exact location to the server and adversary, unless otherwise specified, we deploy a location obfuscation scheme [32], [33] in this paper. To achieve k-anonymity, the drone sends the query additionally piggybacked with k - 1 dummy locations.

Random Location-based Tracking: The shortest path based tracking approach can provide the lowest detection delay but the future trajectory of the drone can be predictable by an adversary. This is because the drone follows a simple and linear path to the target. To avoid the trajectory of the drone being predicted, we propose a random location-based tracking approach. The basic idea is to randomize the trajectory to confuse the adversary while reducing the detection delay of the target. The drone virtually builds a rectangle with a diagonal starting from its current location to the target. Then the drone randomly generates a location as an intermediate destination. The random location should be positioned within the overlapped areas of the rectangle and detection range to enforce the drone flying towards the target. This approach guides the drone not to deviate too much from the path that may increase the detection delay of the target.

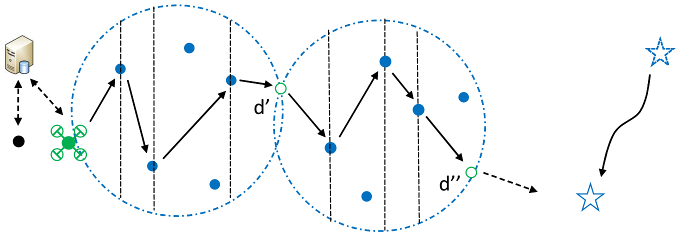


Fig. 3. A dummy location-based target tracking approach, where intermediate destinations (e.g., d' and d'') are marked as a green circle. A set of dummy locations marked as a blue solid circle is generated and distributed within a cloaking area, which is marked as a blue dashed-dot line.

Notations:

- u, dr: A query issuing user and its launching drone, respectively.
- s, tg: A server and a target, respectively.
- $p_{qry,u|dr}$ $[(x,y)_k,\ tg]$: A query packet generated by u (or dr), in which the current location of u (or dr), k - 1 dummy locations, and tg are piggybacked.
- $p_{rpy,s}$ $[(x,y)_{tg}]$: A query reply packet sent by the server, in which the current location of the target ($(x,y)_{tg}$) is piggybacked.

```
◇ When a user launches a drone at the base,
  Send a query, $p_{qry,u}[tg]$, to a server;
  /* Wait until receiving a reply, $p_{rpy,s}[(x,y)_{tg}]$ */
  Generate an intermediate destination, $(x,y)_d$;
  Upload $(x,y)_{tg}$ and $(x,y)_d$ to the drone and launch it;
◇ When the drone flies towards an intermediate destination, e.g., $(x,y)_{d'}$,
  Generate $k$ dummy locations located within the cloaking area;
  repeat
      Follow $k'$ dummy locations; /* where, $k' \leq k$ */
  until the drone arrives $(x,y)_{d'}$;
◇ When the drone arrives at an intermediate destination, e.g., $(x,y)_{d'}$,
  Generate $k$ dummy locations located within the cloaking area;
  Send a query, $p_{qry,dr}[(x,y)_k, tg]$, to the server;
  /* Wait until receiving a reply, $p_{rpy,s}[(x,y)_{tg}]$ */
  Fly towards the next intermediate destination, $(x,y)_{d''}$;
```

Fig. 4. A pseudo code of dummy location-based tracking operations.

In Fig. 2, when the drone receives the location of the target from the server, it builds a virtual rectangle, generates a random location as an intermediate destination (e.g., d'), and flies towards the destination. Upon arriving at the destination, the drone sends a query to the server for any update of the target location. As the target moves, the drone builds a new virtual rectangle based on the current location of the target. Then the drone generates a random location as another intermediate destination (e.g., d'') and flies towards the destination. The drone repeats this procedure until it detects the target. If the y-axes of drone and target are the same, the virtual rectangle becomes a vertical line. The drone randomly chooses the location positioned along the line overlapped with the detection range of drone.

Dummy Location-based Tracking: Although the random location-based tracking approach produces a limited randomized path, if the size of detection range is small, the drone needs to send queries to the server frequently for any update of target location whenever it arrives at intermediate destinations. Both the number of queries sent to the server and the total length of path depend on the size of the drone's detection radius. If the detection radius increases, fewer intermediate destinations are created but can reduce the path randomization. This is because the distance between intermediate destinations increases and intermediate trajectories become simple and linear.

To balance the number of queries and path randomization, we propose a dummy location-based approach. The basic idea is to send a fewer number of queries to the server but to randomize the path more to the target. This approach can reduce the probability of a drone and its future trajectory being detected by an adversary. A user initially generates an intermediate destination along the path (e.g., d') to the target and uploads it to a drone before launching as shown in Fig. 3. The intermediate destination is positioned at the boundary of detection range to reduce the detection delay. Then the drone generates a set of dummy locations (e.g., k = 5) that are randomly distributed within its cloaking area. In this paper, we set the size of the cloaking area to a circle with a diameter that is the distance between the current location of the drone and the next intermediate destination. The drone flies towards the next intermediate destination through the dummy locations. When the drone arrives at the destination, it contacts the server for any update of the target location, generates another k dummy locations, and flies towards the next generated intermediate destination (e.g., d'') through the dummy locations. The drone repeats this procedure until it detects the target.

We enhance the proposed approach by varying the method in selecting dummy locations. In this paper, k is a system parameter and directly impacts the balance between the path randomization and detection delay. For example, if k increases, the path is more randomized but the detection delay increases simultaneously. To maintain a certain level of path randomization but reduce the detection delay, a subset of k dummy locations (e.g., k' = 3) is randomly selected, where $k' \leq k$. This method can reduce the detection delay and confuse an adversary in predicting the trajectory of the drone as well. To further reduce the detection delay, the next dummy location selected from the current dummy location should be located closer to the target as shown in Fig. 3. This can prevent the drone from flying back to the target and increasing the detection delay. A set of major target tracking operations is summarized in Fig. 4.

C. Analysis of Drone Privacy

Location Privacy: When a drone arrives at an intermediate destination, it sends a query piggybacked with its current location to the server. Upon receiving the query reply from the server, the drone updates the target location and reports its current location to the user. Since the query might be intercepted or overheard either by the server or adversary, the current and future locations of the drone can be exposed and predicted. In light of this, we use dummy locations to protect the location privacy of drones and achieve k-anonymity. To quantify the location privacy of drones, we deploy an entropy-based metric [40] in which entropy has been widely used to measure the degree of anonymity in diverse research areas.

First, we follow the procedure of cell-based entropy [41]. We virtually divide a network area into a set of cells with equal size, $n \times n$ cells. Each cell (c) has a probability of being queried based on the history of queries. A query probability in a cell c_i (p_{c_i}) can be expressed as $\frac{n_{c_i}}{\sum_{j=1}^{n^2} n_{c_j}}$, where n_{c_i} is the number of queries generated in c_i. Here, $\sum_{i=1}^{n^2} p_{c_i} = 1$. When the drone arrives at the first intermediate destination at time t, it sends a query to the server with k locations (i.e., cells) that include the real location of the drone and k - 1 dummy locations, $L_t = \{l_1, l_2, \ldots, l_k\}$. A probability that the i^{th} location is the real location of the drone can be expressed as, $p_t = \frac{p_{c_i}}{\sum_{j=1}^{k} p_{c_j}}$.

Second, we also follow the procedure of transition-based entropy [41]. We consider the number of intermediate destinations (n_d), where the drone sends a new query containing dummy locations to the server. Before detecting the target, the drone sends multiple queries to the server. Suppose the drone arrives at the second intermediate destination at time t + Δ and sends a query to the server with another set of k locations, including the real location of drone and k - 1 dummy locations, $L_{t+\Delta}$ = $\{l'_1, l'_2, \ldots, l'_k\}$. Given the probability that i^{th} location is the real location of the drone in L_t, a probability that one of k locations in $L_{t+\Delta}$is the real location of the drone, ($p_{t+\Delta}$), can be expressed as,

$$p_{t+\Delta} = \sum_{x=1}^{k} Pr(l_x \rightarrow l'_y | l_x = l_i) \cdot Pr(l_x = l_i). \qquad (1)$$

Here, $l_x \in L_t$ and $l'_y \in L_{t+\Delta}$. In Eq. 1, $Pr(l_x \rightarrow l'_y | l_x = l_i)$ can be calculated as, $\frac{n_{x\rightarrow y}}{\sum_{y=1}^{k} n_{x\rightarrow y}}$, where $n_{x\rightarrow y}$ is the number of movements from l_x to l'_y at $t + \Delta$. $Pr(l_x = l_i)$ is p_t and can be calculated as, $\frac{p_{c_i}}{\sum_{i=1}^{k} p_{c_i}}$. Based on [40], the entropy (E) can finally be expressed as,

$$E = -\sum_{y=1}^{k} p_{t+\Delta} \cdot log_2(p_{t+\Delta}). \qquad (2)$$

Note that we consider this entropy as uncertainty in determining the real location of drones from potential candidates including dummy locations.

Trajectory Privacy: When a drone flies towards the target, its trajectory can be exposed and predicted by an adversary. To protect the trajectory privacy, we randomize the trajectory using dummy locations until the drone detects the target. In this paper, we consider a convex hull and the number of paths to quantify the trajectory privacy of drones.

First, we deploy a convex hull algorithm to approximate an area where the drone flies. A convex hull is defined as the smallest convex polygon containing a given set of points in a two-dimensional area. To construct a convex hull, we use the Graham-Scan based method [42] and adapt it into our context, consisting of four major operations. (i) We randomly generate k dummy locations within a cloaking area (see Fig. 3). (ii) We initially choose a location with the minimum x- and y- coordinates and sort the rest of the locations based on the angle to the location in counterclockwise order. (iii) Then the initially chosen location is connected with a location with the minimum angle. (iv) We incrementally keep connecting the locations only if they are located to the counterclockwise of the line connecting the previous two locations. Finally, this convex hull becomes an area where the drone flies and confuses the adversary in predicting the trajectory. The bigger the size of the convex hull is, the harder the adversary predicts the trajectory. Note that the size of the convex hull is directly proportional to the cloaking area, where dummy locations are distributed. If dummy locations are widely spread out in the area, the size of the convex hull increases but the detection delay also increases.

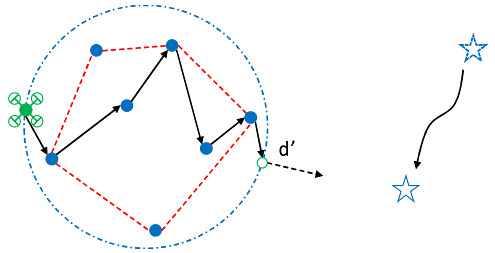


Fig. 5. A convex hull depicted as a polygon marked by a red-dashed line is created using dummy locations. Here, k and n_r are set to 7 and 5, respectively. A set of dummy locations is marked as a blue solid circle.

Suppose there is the number of intermediate destinations (n_d) between the base and target. The drone constructs a convex hull (c_i) when it flies towards the i^{th} intermediate destination. The total area of the convex hull is the summation of the convex hull generated at each intermediate destination and can be expressed as, n_s = $\sum_{i=1}^{n_d} c_i$. However, the size of the cloaking area depends on the number of intermediate destinations. As the number of intermediate destinations increases implying that the distance between intermediate destinations decreases, the size of the cloaking area decreases. This can impact the size of the convex hull. Thus, we measure the average percentage of areas covered by convex hull out of entire cloaking areas,

$$n_s(\%) = \frac{\sum_{i=1}^{n_d} \frac{c_i}{\pi\gamma_i^2}}{n_d + 1} \times 100, \qquad (3)$$

where γ_i is the radius of the cloaking area located between $(i-1)^{th}$ and i^{th} intermediate destinations. The 0^{th} intermediate destination is the base. In Fig. 5, we show a convex hull consisting of dummy locations. As depicted, the convex hull is a polygon connecting the smallest number of dummy locations, i.e., connecting five dummy locations out of seven. In this paper, we measure the percentage of convex hull area out of the cloaking area for performance evaluation (See Fig. 11).

Second, we count the number of paths that the drone can potentially use until detecting the target. When the drone flies towards an intermediate destination, it generates dummy locations, produces a set of paths by connecting the locations, and randomly selects one of the paths for traverse (see Fig. 3). The adversary may predict the location of intermediate destination but it would be hard to predict the trajectory traversed by the drone. Note that since the drone uses dummy locations to protect its location privacy at each intermediate destination, the adversary would also be hard to predict the location of intermediate destination. For the purpose of comparison, the adversary may easily predict a trajectory produced by the proposed shortest path-based tracking approach because the drone uses a single path until detecting the target for the entire flight.

Suppose there is a set of intermediate destinations (n_d) between the base and target. The drone generates k_i dummy locations when it flies towards the i^{th} intermediate destination. If each path includes all dummy locations without any duplicated location, the total number of paths is the concatenation of all the trajectories generated at each intermediate destination and can be expressed as,

$$n_p = \prod_{i=1}^{n_d+1} k_i!. \tag{4}$$

In addition, we can even produce more paths by changing the number of dummy locations involved. For example, if each path can include at least one dummy location, the total number of paths can be calculated as,

$$n'_p = \prod_{i=1}^{n_d+1} (\sum_{n_r=1}^{k} \frac{k!}{(k-n_r)!}). \tag{5}$$

IV. Performance Evaluation

We evaluate and compare the performance of proposed target tracking strategies with both static and mobile target scenarios by changing key simulation parameters.

A. Simulation Testbed

We develop a customized discrete-event driven simulator using the OMNeT++ [9] and conduct extensive experiments on a Windows 10 machine with a 1.6 GHz Intel i5 processor and 8 GB RAM. A rectangular network (e.g., 1,000 × 1,000 m^2) is deployed, where three main objects including user, drone, and target are located. A user equipped with an onboard GPS receiver and multiple network interface cards (e.g., IEEE 802.11 and LTE) communicates with a server and a drone simultaneously. We set the detection range of the drone to 150 m but it may vary depending on the attached devices and sensors, such as camera, lidar, or radar. The two-ray ground propagation channel is assumed with a data rate of 2 Mbps. We deploy a simple CSMA/CA-based medium access for the link layer. The user communicates with the drone directly if they are located within their communication ranges. When the drone flies out of the user's communication range, the user can still communicate with it indirectly through the server. For the sake of simplicity, we assume that both user and drone have the same communication range (e.g., 150 m) and use a symmetric link. Both user and drone communicate with the server directly through the 4G wireless networks.

The user stays in a base where the drone is launched. The drone flies through a set of destinations assigned by the user and detects a target. A stationary target is randomly located in the network. A mobile target moves by following a mobility model: linear, circular, and random waypoint. Initially, the user and target are located at the leftmost bottom and rightmost up in the network respectively. To clearly observe the trace of drone for target tracking, the velocities of drone and mobile target are set to relatively low, 1 m/sec. The drone generates the number of intermediate destinations (n_d, 3 to 10) along the path. The drone also generates a set of dummy locations (n_r, 3 to 12) within the cloaking area. A value of anonymity (k) ranges from 3 to 15.

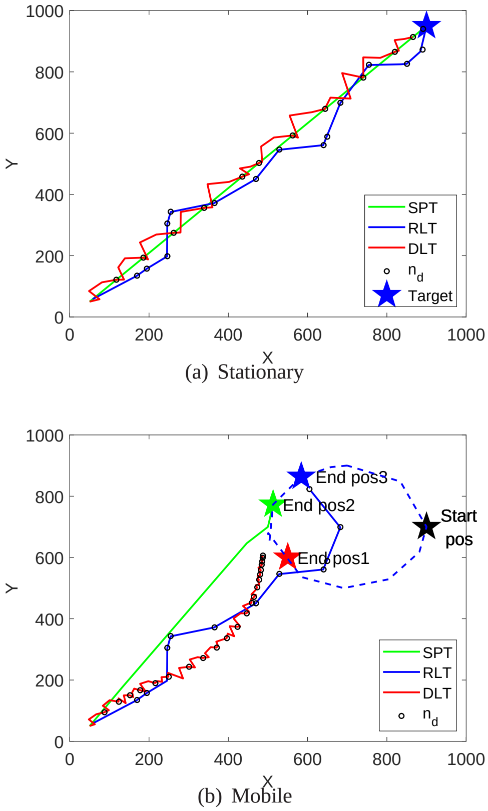
B. Simulation Results

We measure the performance of the proposed target tracking strategies in terms of entropy-based anonymity, the size of the convex hull, the number of paths, drone traces, detection delay, and the number of packets exchanged with the server. We compare the performance of our strategies with stationary and mobile target scenarios. Here, the proposed shortest path-, random location-, and dummy location-based target tracking strategies are denoted as *SPT*, *RLT*, and *DLT* respectively. The SPT can be used as a performance lower bound.

Snapshot of Drone Trace: We snapshot a set of drone traces to observe the changes of trajectories towards a stationary or mobile target in Fig. 6. Given a stationary target, the SPT builds a path from the base to the target as shown in Subfig. 6(a). The drone flies along the path without visiting any intermediate destination. Since the path is a straight line to the target, an adversary can easily predict the future trajectory of the drone. In the RLT, however, the drone flies towards randomly generated locations along the path to the target. Each random location is generated within the overlapped regions of the virtually built rectangle and the detection range of the drone. The path is a little randomized but the trajectory between random locations is still straight. In the DLT, the drone not only visits a set of intermediate destinations but also flies through a set of dummy locations between the intermediate destinations, where n_d and n_r are 10 and 3, respectively. The path can be randomized by changing the number of dummy locations.

In Subfig. 6(b), a mobile target follows a circular mobility model. The SPT still builds a simple path based on the last updated location of the target. Both RLT and DLT show that the paths are adaptively changed while the target is moving. This is because whenever the drone arrives at the intermediate destination, it contacts the server, updates the current location of the target, and rebuilds a trajectory towards the target.

Dummy Location Vs. Intermediate Destination: In Fig. 7, we investigate the impact of the number of dummy locations (n_r) and intermediate destinations (n_d) on the path towards the target and draw the traces of the drone. In the DLT, we can observe a set of randomized paths by changing n_r and n_d from 5 to 12 and from 5 to 10, respectively. In Subfig. 7(a), the drone should go through randomly generated dummy locations



(a) Stationary

(b) Mobile

Fig. 6. Snapshots of drone traces for stationary and mobile targets. A mobile target follows circular mobility and moves from a starting position to ending positions.

(i.e., n_r = 5) between two adjacent intermediate destinations (i.e., n_d = 5). The drone flies in a zig-zag manner for the entire path compared to a straight line under the SPT (see Subfig. 6(a)). As n_r increases, more zigzag patterns are observed in Subfigs. 7(b) and (c). As n_d increases, the distance between intermediate locations decreases. This will decrease the size of the cloaking area located between intermediate destinations. In Subfigs. 7(d) to (f), as n_r increases, more dummy locations are generated within a smaller cloaking area. Thus, the drone flies through the trajectories showing more short and narrow-width zig-zag patterns.

The DLT can enhance the randomization of the path by flexibly selecting the number of dummy locations located between intermediate destinations. In Fig. 8, we show a snapshot of the randomized path including intermediate destinations and dummy locations. In each trajectory, the drone can choose a different number of dummy locations. For example, the drone selects one to four dummy locations out of five in each trajectory and confuses the adversary.

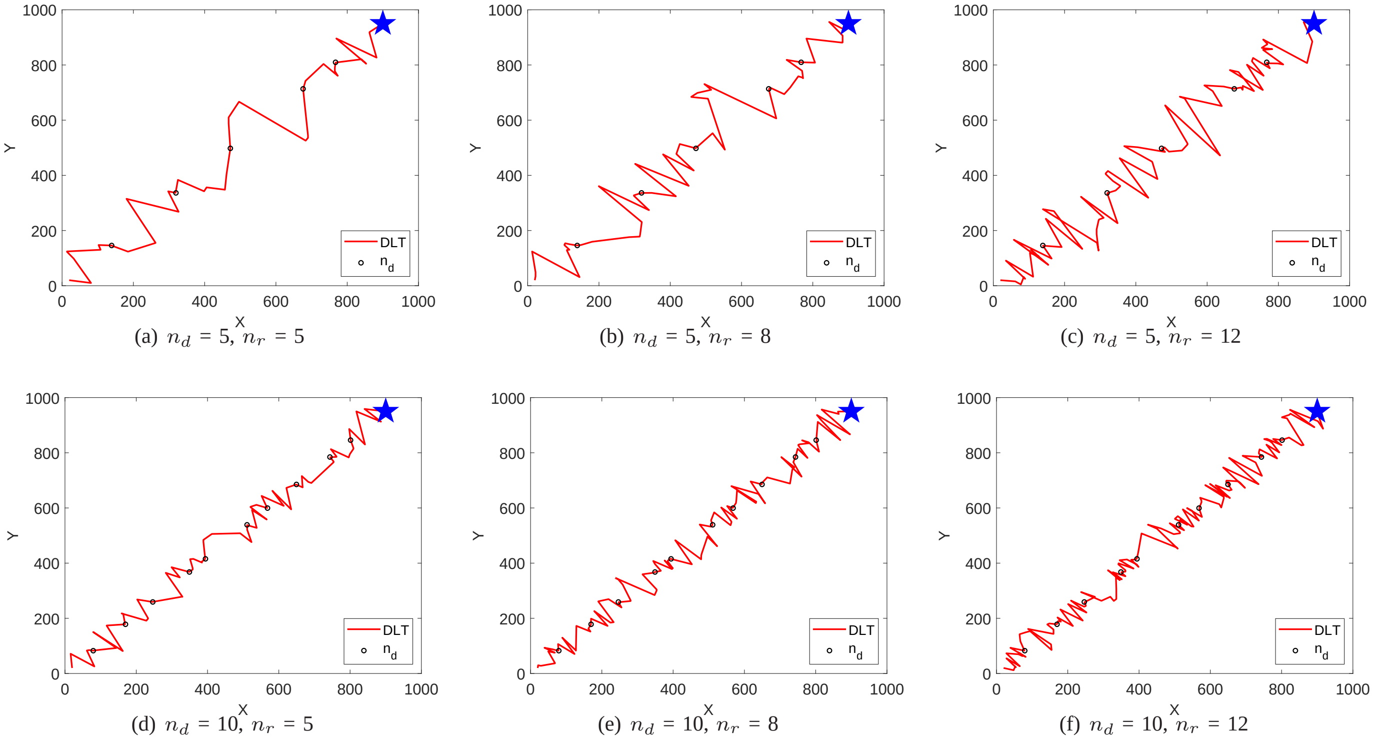
Drone Trace of Mobile Target: As depicted in Fig. 9, we trace the path of the drone using the DLT in the presence of a mobile target to see whether the path can adaptively be changed. Here, the target moves under one of three mobility models: linear, circular, and random waypoint. The number of intermediate destinations and dummy locations is set to three and five respectively for entire traces. Under random waypoint mobility, the target moves towards a randomly selected destination in the network. After the target arrives at the destination, it stays for a pause time (e.g., rest period). When the pause time expires, the target moves towards another randomly selected destination. The target follows the procedure repetitively. In this paper, the pause time is set to zero implying that the target always moves in the network.

The drone follows adaptive trajectories and flies towards the moving target as shown in Subfigs. 9(a) to (c). Here, n_d and n_r are initially set to three and five, respectively. As the drone flies and becomes closer to the target, the distance between intermediate destinations decreases. As the target shows a dynamic movement by following either circular or random waypoint mobility, the drone follows adaptive trajectories and detects the moving target. In Subfigs. 9(d) to (f), we show another set of traces, where n_d and n_r are set to 10 and five, respectively. With higher n_d, the traces have more short and narrow-width zig-zag patterns compared to the traces shown in Subfigs. 9(a) to (c). Since the distance between intermediate destinations decreases, the same dummy locations are located in a smaller cloaking area. Even though each trajectory is randomized, it is similar to a straight line.

Location and Trajectory Privacy: In Fig. 10, we measure the entropy to quantify the location privacy of drones for static and mobile targets by changing k. When the drone arrives at an intermediate destination, it generates a set of dummy locations to hide the current location from the server and adversary. As k increases, the entropy increases gradually in both static and mobile target scenarios as shown in Subfigs. 10(a) and (b), respectively. We compare the measured entropy with the optimal entropy that can be achieved when each query probability in all k locations is the same. In Subfig. 10(a), as n_d increases, the entropy also increases because more intermediate destinations can positively affect the randomness of the path. In Subfig. 10(b), when the target moves, the entropy closer to the optimal is observed compared to that of the stationary target regardless of the mobility models of the target.

We also measure the percentage of the convex hull out of the cloaking area located between intermediate destinations for stationary and mobile targets by changing n_d and n_r as shown in Fig. 11. As n_r increases, the dummy locations tend to spread widely within the cloaking area. Thus, the percentage of the convex hull increases for entire mobility models. As n_d increases, however, the percentage of the convex hull does not increase significantly in Subfigs. 11(a) and (b). This is because as n_d increases, the distance between intermediate destinations reduces and the size of the cloaking area reduces as well. Under the reduced size of the cloaking area, the dummy locations are limited to be distributed within the cloaking area. More importantly, as the convex hull size increases, it becomes difficult for the adversary to predict the trajectory of the drone.

In addition, we measure the total number of paths that the drone can select to detect the stationary and mobile targets in Fig. 12. Both n_p and n'_p heavily depend on n_d and k (see



(a) n_d = 5, n_r = 5

(b) n_d = 5, n_r = 8

(c) n_d = 5, n_r = 12

(d) n_d = 10, n_r = 5

(e) n_d = 10, n_r = 8

(f) n_d = 10, n_r = 12

Fig. 7. Impact of the number of dummy locations and intermediate destinations on the path towards the target.

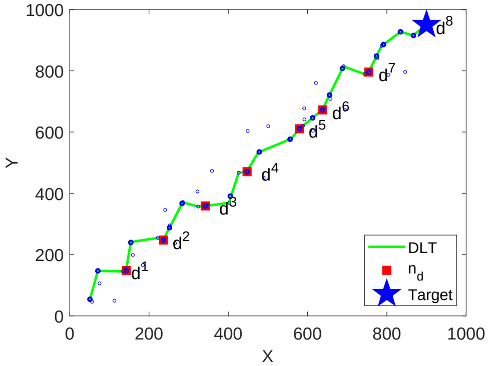
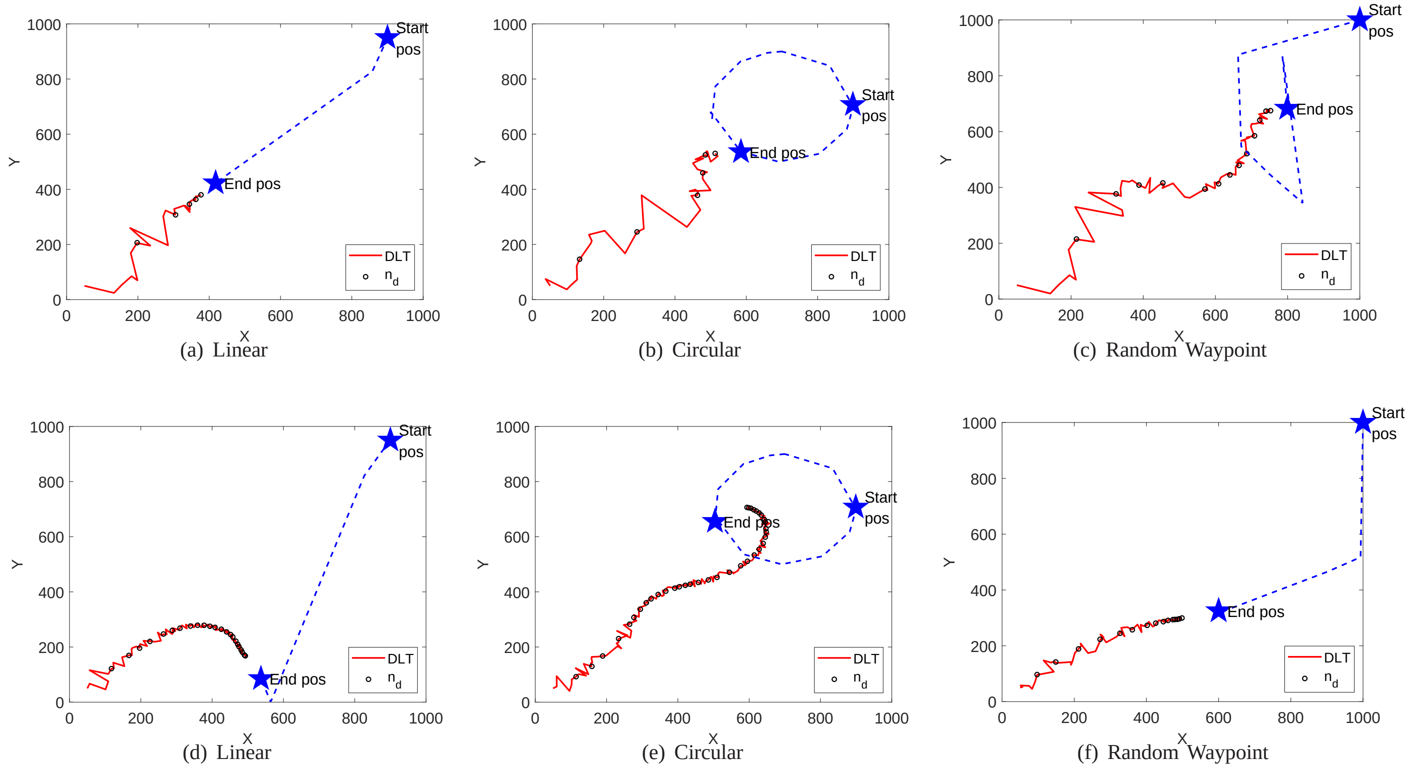


Fig. 8. A snapshot of the randomized path, where the empty blue circle and red rectangle are a dummy location and an intermediate destination, respectively. A solid blue circle represents the point where the drone flies towards the target. Here, n_d = 8, n_r = 5.

Eqs. 4 and 5). Since the path consists of a set of trajectories located between intermediate destinations, the total number of paths is proportional to n_d. Since k directly affects the number of trajectories, n_p increases as k increases. The difference between n_p and n'_p is how flexibly the drone selects the dummy locations contributing to the trajectory. n'_p counts the paths that include at least one dummy location but n_p should include all the dummy locations. Thus, n'_p is significantly higher than n_p. In Fig. 12, we set n_d to two because of the complexity in calculating the number of paths. Although n_d is small, the total number of paths is quite high. As the target moves, the path is supposed to be changed with additional intermediate destinations (see Fig. 9). All the n_p are similar for entire mobility models. However, n'_p under the linear and circular mobility models is significantly higher than that of random waypoint mobility. This is because more intermediate destinations and their corresponding trajectories are generated.

Detection Delay: In Fig. 13, we measure the detection delay of three target tracking strategies. The detection delay is a latency that the drone takes from launching at the base to locating the target within the detection range. The detection delay of the mobile target may vary depending on the mobility model. Thus, we experiment with a stationary target for performance comparison. In Subfig. 13(a), the detection delay of SPT shows the lowest detection delay in the strategies because the path is a straight line from the base to the target. The delay of SPT is used as the performance bottom line in this paper. The delay of RLT slightly increases as the drone has limited randomization of the path by visiting randomly generated intermediate destinations.

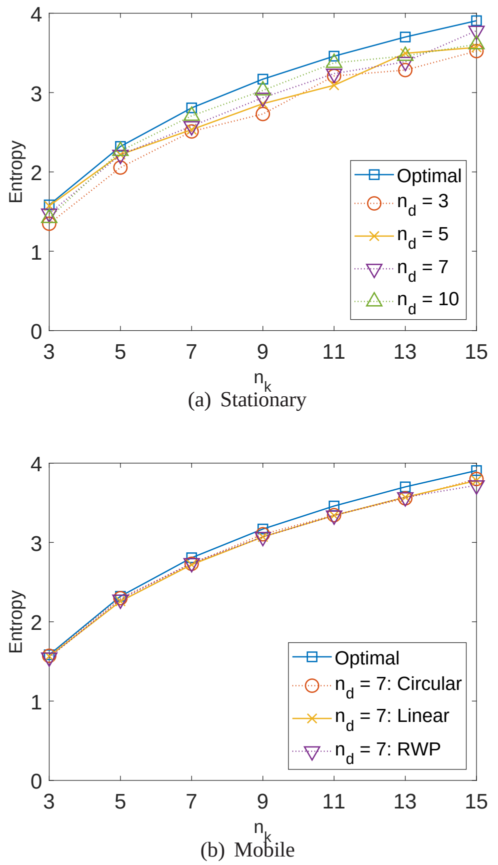
Three variants of DLT are provided by selecting a different number of dummy locations involved in the trajectory in Subfig. 13(b). Given n_r, the drone can choose different trajectories between intermediate destinations by visiting single (DLT-Single), selective (DLT-Selective), or all (DLT-All) dummy locations. Since the path randomization is proportional to n_r, the detection delay increases as n_r increases. Here, n_r ranges from 3 to 15 and n_d is set to 5. As n_r increases, the drone moves more in a zigzag manner towards the target and thus, the detection delay of all three variants increases. The DLT-All shows the highest detection delay in the variants because the drone follows all the dummy locations in each trajectory.



(a) Linear (b) Circular (c) Random Waypoint

(d) Linear (e) Circular (f) Random Waypoint

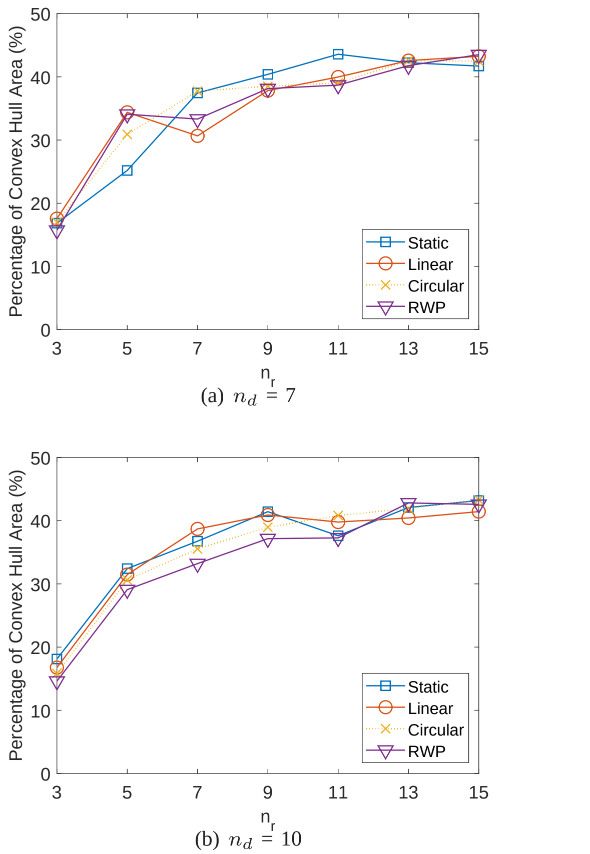
Fig. 9. Drone traces for mobile target with three mobility models: linear, circular, and random waypoint. Here, Subfigs. (a) to (c) deploy n_d = 3 and n_r = 5. Subfigs. (d) to (f) deploy n_d = 10 and n_r = 5.



(a) Stationary

(b) Mobile

Fig. 10. Entropy against k for stationary and mobile targets, respectively. In Subfig. 10(a), n_d is set from 3 to 10. In Subfig. 10(b), n_d is set to 7.



(a) n_d = 7

(b) n_d = 10

Fig. 11. The average percentage of convex hull area out of the cloaking area against n_r for stationary and mobile targets, respectively.

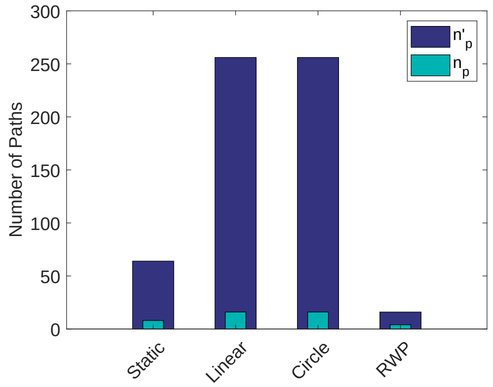
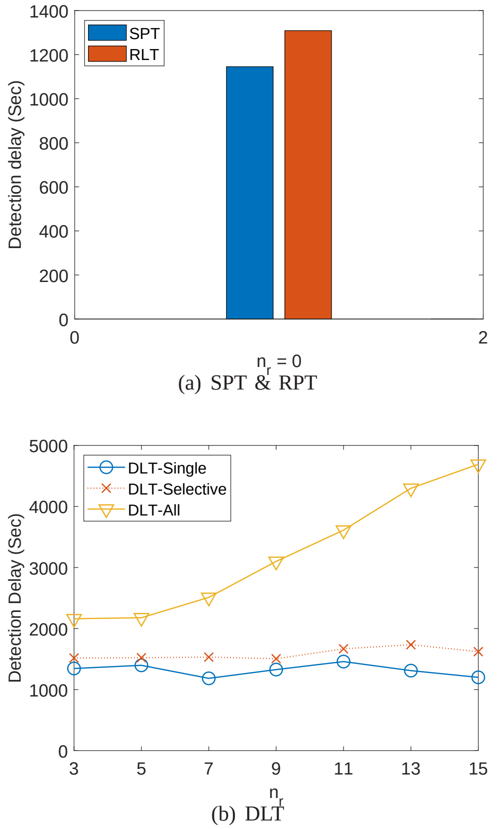


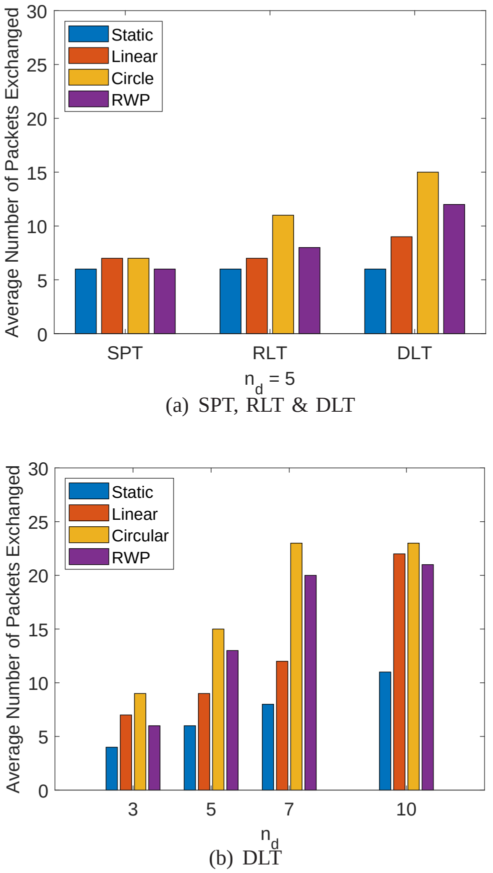
Fig. 12. The total number of paths for both stationary and mobile targets, respectively. Here, n_d is set to 2.



(a) SPT & RPT

(b) DLT

Fig. 13. Detection delay against n_r ranging from 3 to 15. Here, n_d is set to 5.



(a) SPT, RLT & DLT

(b) DLT

Fig. 14. The average number of packets exchanged between the drone and server.

Here, the DTL-All can be considered as the performance upper bound. Since the DLT-Selective involves a subset of dummy locations in the trajectory, its detection delay is slightly higher than that of the DLT-Single. However, the DTL-Selective increases the path randomization.

Number of Packets Exchanged: In Fig. 14, we count the average number of packets exchanged between the drone and server. In Subfig. 14(a), we compare the performance of three target tracking strategies with both stationary and mobile target scenarios. For comparison purposes, we set n_d to 5 for the DLT. The RLT shows the lowest average number of packets because the drone follows the shortest path to the target. Since the DLT frequently sends queries to the server whenever it arrives at intermediate destinations, the DLT shows a higher average number of packets than that of both SPT and RLT. When the target follows circular mobility, both RLT and DLT show a higher average number of packets than that of the SPT because more updates of the target location are needed.

In Subfig. 14(b), for a stationary target, the drone contacts the server at every pre-determined intermediate destination. This is because the drone does not know whether the target is stationary or mobile. The average number of packets increases as n_d increases. In the case of the mobile target, the current location of the target is time-varying. Since the mobility of the target increases n_d, the drone frequently contacts the server for whereabout the target. Thus, the average number of packets increases significantly.

V. Discussion

To see the full potential of the proposed tracking strategies, we raise several research issues for discussion and future work. In particular, we introduce mission-oriented use cases in which our research can be an important role in both civilian and military environments.

A. Constraints of Target Tracking

Due to the inherent resource constraints of drone, we need to consider potential constraints that could impact the design and performance of the proposed target tracking strategies. In this paper, we deploy both stationary and mobile target scenarios in a limited size network. If a target should be located in a wide area network, however, energy-conserving based tracking is essential because the drone is powered by a limited battery. Since energy consumption is proportional to flying distance, the drone may not be feasible to track such a long-distance target for an extended period. The path should be built not only to quickly locate the target but also to efficiently confuse an adversary. The shortest path can reduce the energy consumption of drone, but the future trajectory of the drone can easily be predictable by the adversary. Thus, energy-conserving and privacy-preserving approaches are conflicting requirements, and optimization of both is admittedly extremely complex.

We also implicitly assume that a path between user and target is not affected by any static obstacles, such as buildings, mountains, or trees. A weather condition (e.g., rain or wind) can be a part of dynamic obstacles. However, we need to relax this assumption by considering the limited number of intermediate destinations or even the unavailability of some intermediate destinations along the path during flight. Due to the obstacles, a planned intermediate destination may not be involved in the path. This can directly impact the trajectory and its randomization, leading to the change of a set of dummy locations. To balance the energy consumption and level of privacy, the number of dummy locations can be adjusted depending on the distance between intermediate destinations.

In addition, the drone conducts the proposed tracking strategies based on the updated location of the target (e.g., mobile target) that is replied to by the server. If the drone cannot be updated because of link disconnection or packet loss, it may fly toward the next intermediate destination with the staled location of the target, resulting in a longer detection delay. In this paper, we deploy a single drone that solely relies on communication with the server. Reliable link connectivity between the server and drone may directly affect the performance of target tracking. Multiple drones can further be considered to collaboratively track the target.

B. Map-based Target Tracking and Path Planning

The proposed drone-assisted target tracking strategies can play a critical role in implementing mission-oriented operations in both civilian and military environments, such as surveillance, reconnaissance, or post-disaster relief. We envision that the advance in technology will enable users to have a personal drone, just like a smartphone. A light-weighted and micro-sized drone will be integrated with a wireless and mobile device for portability. For example, a soldier will carry a drone equipped with appropriate sensors and devices and conduct a reconnaissance operation by launching the drone. The drone flies towards a set of strategic locations to scan a target or unknown area. To hide the current location of the drone and soldier from an enemy, the drone randomizes trajectories while flying from one location to another.

In the reconnaissance operation, the soldier may not be familiar with the area to scan. It would take a time to identify a set of strategic locations even before launching a drone. In light of this, we will develop a path planning algorithm to produce a set of locations by considering geographical features that can be extracted from a map. Depending on a given mission, the soldier will use mission-related constraints and conditions to further filter the locations to scan. The extracted locations will be connected to build a path, in which the drone will randomize trajectories while flying from one location to another. A rationale behind this approach is that, for example, Google Maps embeds a variety of information, such as flooding, wildfire, road accident, or geographical data, and shows it on the map directly in a real-time fashion. The embedded information can be retrieved by using the Google Maps application programming interface (API). We expect that the public map will embed more information to enhance the public interest and safety. Compared to the public map, a military map embeds more detail and accurate geographical and strategic information. The map-based path planning will be best suited to strategically scan an area when it is not clearly identified during the night.

In addition, the map-based path planning can be extended to a civilian application, disaster relief operation, if the drone does not randomize the trajectories. For example, a rescue team will extract a set of target locations (e.g., residential area) to find any survivors in an emergency site impacted by disasters, such as an earthquake or flooding. This is because survivors would likely be found more in the residential or commercial areas than in rural areas. The drone is able to strategically rescue survivors by following the locations.

C. Drone-Assisted Smart Communities

In this paper, a drone communicates with the server to update the whereabouts of a static or mobile target. We need to relax this server-based approach and extend the limited connectivity of drones. There has been a research effort for flexible connectivities and accessibilities in drone-based operations [43], [44]. A single or multiple drones are deployed in a micro cell-sized network to seamlessly relay ongoing user communications to enhance the connectivity. Drones are also deployed in an isolated area or emergency site, where the existing infrastructure network is not available or has been collapsed, and play an important role as a mobile base station to extend ad hoc networks. Frequently accessed or popular data are stored in drones to enhance data accessibility and availability [45], [46].

We will develop a collaborative target tracking technique to efficiently share the location and status of the target for detection via infrastructure-based or infrastructure-less networks. In this research, a vehicle is considered as a mobile target under a hit-and-run scenario. Due to high-speed mobility, the vehicle resides in a coverage area for a short period. For the sake of simplicity, a single drone powered by a battery is deployed in a network, where roadside units (RSUs) and access points (APs) are available for communications. Since wireless communication could be responsible for more than half of total energy consumption, we will investigate how to minimize the energy consumption of the drone but to maximize interactions among network components to reduce the detection delay of the target. A set of control packets and its corresponding operations among network components will also be investigated. In addition, we will build a small-scale testbed with a micro drone (e.g., Crazyflie [47]) for proof-of-concept. We will conduct the experiments in both indoor and outdoor environments. Here, Crazyflie is a programmable micro drone and has been widely used in research, education, and industry for its complete control and full flexibility while deployed in diverse applications, such as autonomous flight, pathfinding, environment scanning, etc.

VI. Concluding Remarks

Unlike prior research focusing on the protection of users and restricted areas from an unwanted privacy attack and intrusion, we shifted the privacy paradigm to protecting the drone's location privacy. Three drone-based target tracking strategies were proposed to detect a static or mobile target while protecting the location privacy of the drone via obfuscating the current location of the drone and randomizing the trajectory. We analyzed and measured drone privacy in terms of entropy-based anonymity, the size of the convex hull, and the number of location paths. We evaluated the proposed strategies through extensive simulation experiments. The results show that the proposed strategies can track and detect the target by following randomized trajectories. We envision that the proposed strategies can be applied to diverse drone-based applications in realizing a smart and connected community.

Acknowledgment

This research was supported in part by the International Cooperative Research Grant in 2020 from Incheon National University (Incheon, Korea) and the Brain Pool program funded by the Ministry of Science and ICT through the National Research Foundation of Korea (2021H1D3A2A01080645).

References

[1] P. Fahlstrom and T. Gleason, *Introduction to UAV systems*. John Wiley & Sons, 2012.

[2] *Unmanned Aircraft Systems (UAS) by the Numbers, Federal Aviation Administration (FAA), U.S. Dept. of Transportation*, https://www.faa.gov/uas/resources/by_the_numbers/ (last accessed at Dec 2021).

[3] *Commercial Unmanned Aerial Vehicles (UAVs) – Statistics & Facts*, https://www.statista.com/topics/3601/commercial-uavs/ (last accessed at Dec 2021).

[4] A. A. Paranjape, S. Chung, K. Kim, and D. H. Shim, "Robotic Herding of a Flock of Birds Using an Unmanned Aerial Vehicle," *IEEE Trans. on Robotics*, vol. 34, no. 4, pp. 901–915, 2018.

[5] P. Blank, S. Kirrane, and S. Spiekermann, "Privacy-Aware Restricted Areas for Unmanned Aerial Systems," *IEEE Security & Privacy*, vol. 16, no. 2, pp. 70–79, 2018.

[6] S. Winkler, S. Zeadally, and K. Evans, "Privacy and Civilian Drone Use: The Need for Further Regulation," *IEEE Security & Privacy*, vol. 16, no. 5, pp. 72–80, 2018.

[7] A. Fitwi, Y. Chen, and S. Zhu, "No Peeking through My Windows: Conserving Privacy in Personal Drones," in *Proc. IEEE Int'l Smart Cities Conference*, 2019.

[8] Z. Li, C. Gao, Q. Yue, and X. Fu, "Toward Drone Privacy via Regulating Altitude and Payload," in *Proc. Int'l Conf. on Computing, Networking and Communications*, 2019.

[9] *OMNeT++ Documentation and Tutorials*, http://www.omnetpp.org/documentation/.

[10] J. Freudiger, M. H. Manshaei, J. L. Boudec, and J. Hubaux, "On the Age of Pseudonyms in Mobile Ad Hoc Networks," in *Proc. IEEE INFOCOM*, 2010.

[11] S. Taheri, S. Hartung, and D. Hogrefe, "Achieving Receiver Location Privacy in Mobile Ad Hoc Networks," in *Proc. IEEE Social Computing*, 2010.

[12] K. E. Defrawy and G. Tsudik, "Privacy-Preserving Location-Based On-Demand Routing in MANETs," *IEEE Journal of Selected Areas in Communications*, vol. 29, no. 10, 2011.

[13] L. M. M. A. Ferrag and A. Ahmim, "Privacy-Preserving Schemes for Ad Hoc Social Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, 2017.

[14] P. Bhattacharya and M. L. Gavrilova, "Roadmap-Based Path Planning – Using the Voronoi Diagram for a Clearance-Based Shortest Path," *IEEE Robotics & Automation Magazine*, vol. 15, no. 2, pp. 58–66, 2008.

[15] H. O. Arranz, D. R. Llanos, and A. G. Escribano, *The Shortest-path Problem: Analysis and Comparison of Methods*. Synthesis Lectures on Theoretical Computer Science, San Rafael, California, 2015.

[16] Y. Lu, Z. Xue, G. Xia, and L. Zhang, "A Survey on Vision-based UAV Navigation," *Geo-spatial Information Science*, vol. 21, no. 1, pp. 21–32, 2018.

[17] A. Yershova, L. Jaillet, T. Simeon, and S. M. LaValle, "Dynamic-Domain RRTs: Efficient Exploration by Controlling the Sampling Domain," in *Proc. IEEE Robotics and Automation*, 2005.

[18] F. Andert and F. Adolf, "Online World Modeling and Path Planning for an Unmanned Helicopter," *Autonomous Robots*, vol. 27, no. 3, pp. 147–164, 2009.

[19] Q. Zhang, J. Ma, and Q. Liu, "Path Planning based Quadtree Representation for Mobile Robot using Hybrid-simulated Annealing and Ant Colony Optimization Algorithm," in *Proc. IEEE Intelligent Control and Automation*, 2012.

[20] Z. Fu, J. Yu, G. Xie, Y. Chen, and Y. Mao, "A Heuristic Evolutionary Algorithm of UAV Path Planning," *Wireless Communications and Mobile Computing*, 2018.

[21] H. Bayerlein, M. Theile, M. Caccamo, and D. Gesbert, "Multi-UAV Path Planning for Wireless Data Harvesting With Deep Reinforcement Learning," *IEEE Open Journal of the Communications Society*, vol. 2, 2021.

[22] D. Zorbas, T. Razafindralambo, D. P. P. Luigic, and F. Guerriero, "Energy Efficient Mobile Target Tracking using Flying Drones," in *Proc. Ambient Systems, Networks and Technologies (ANT)*, 2013.

[23] H. Zhang, G. Wang, Z. Lei, and J. Hwang, "Eye in the Sky: Drone-Based Object Tracking and 3D Localization," in *Proc. ACM Multimedia*, 2019.

[24] A. Hamdi, F. Salim, and D. Kim, "DroTrack: High-speed Drone-based Object Tracking Under Uncertainty," in *Proc. IEEE Fuzzy Systems (FUZZ)*, 2020.

[25] Y. Wu, H. Dai, H. Wang, and K. R. Choo, "Blockchain-Based Privacy Preservation for 5G-Enabled Drone Communications," *IEEE Network*, vol. 35, no. 1, 2021.

[26] H. Kim, J. Ben-Othman, and L. Mokdad, "UDiPP: A Framework for Differential Privacy Preserving Movements of Unmanned Aerial Vehicles in Smart Cities," *IEEE Trans. on Vehicular Technology*, vol. 68, no. 4, 2019.

[27] Y. Pan, S. Li, J. L. Chang, Y. Yan, S. Xu, Y. An, and T. Zhu, "An Unmanned Aerial Vehicle Navigation Mechanism with Preserving Privacy," in *Proc. IEEE ICC*, 2019.

[28] K. G. Shin, X. Ju, Z. Chen, and X. Hu, "Privacy Protection for Users of Location-Based Services," *IEEE Wireless Communications*, vol. 9, no. 1, pp. 30–39, 2012.

[29] M. Wernke, P. Skvortsov, F. Durr, and K. Rothermel, "A Classification of Location Privacy Attacks and Approaches," *Personal and Ubiquitous Computing*, vol. 18, no. 1, pp. 163–175, 2014.

[30] R. Shokri, P. Papadimitratos, and G. Theodorakopoulos, "Collaborative Location Privacy," in *Proc. IEEE MASS*, 2011.

[31] B. Niu, X. Zhu, W. Li, and H. Li, “EPcloak: An Efficient and Privacy-Preserving Spatial Cloaking Scheme for LBSs,” in *Proc. IEEE MASS*, 2014.

[32] H. Lu, C. S. Jensen, and M. L. Yiu, “PAD: Privacy-Area Aware, Dummy-Based Location Privacy in Mobile Services,” in *Proc. ACM Int’l Workshop on Data Engineering for Wireless and Mobile Access*, 2008.

[33] T. Hara, A. Suzuki, M. Iwata, Y. Arase, and X. Xie, “Dummy-Based User Location Anonymization Under Real-World Constraints,” *IEEE Access*, vol. 4, pp. 673 – 687, 2016.

[34] M. Gruteser and D. Grunwald, “Enhancing Location Privacy in Wireless LAN Through Disposable Interface Identifiers: A Quantitative Analysis,” in *Proc. WMASH*, 2003.

[35] L. Huang, K. Matusuura, H. Yamane, and K. Sezaki, “Towards Modeling Wireless Location Privacy,” in *Proc. Privacy Enhancing Technologies*, 2005.

[36] L. Huang, H. Yamane, K. Matusuura, and K. Sezaki, “Silent Cascade: Enhancing Location Privacy Without Communication QoS Degradation,” in *Proc. Security in Pervasive Computing*, 2006.

[37] M. Li, K. Sampigethaya, L. Huang, and R. Poovendran, “Swing & Swap: User-Centric Approaches Towards Maximizing Location Privacy,” in *Proc. Workshop on Privacy in the Electronic Society*, 2006.

[38] Y. Wang, T. Wang, G. Zhang, Q. Cheng, and J. Wu, “Small Target Tracking in Satellite Videos Using Background Compensation,” *IEEE Trans. on Geoscience and Remote Sensing*, vol. 68, no. 10, pp. 7010–7021, 2020.

[39] W. Stallings, *Cryptography and Network Security - Principles and Practices*. Prentice Hall, Inc., 2013, 6th Edition.

[40] A. Serjantov and G. Danezis, “Towards an Information Theoretic Metric for Anonymity,” in *Proc. Privacy Enhancing Technologies*.

[41] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, “Achieving k-Anonymity in Privacy-aware Location-based Services,” in *Proc. IEEE INFOCOM*, 2014.

[42] R. Graham, “An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set,” *Information Processing Letters*, vol. 4, no. 1, pp. 132–133, 1972.

[43] L. Bekmezci, O. K. Sahingoz, and S. Temel, “Flying Ad-Hoc Networks (FANETs): A survey,” *Ad Hoc Networks*, vol. 11, pp. 1254–1270, 2013.

[44] M. Deruyck, J. Wyckmans, L. Martens, and W. Joseph, “Emergency ad-hoc networks by using drone mounted base stations for a disaster scenario,” in *Proc. Workshop on Emergency Networks for Public Protection and Disaster Relief*, 2016.

[45] R. Amer, W. Saad, H. ElSawy, M. M. Butt, and N. Marchetti, “Caching to the Sky: Performance Analysis of Cache-Assisted CoMP for Cellular-Connected UAVs,” in *Proc. IEEE Wireless Communications and Networking Conf.*, 2019.

[46] M. Wei, Y. Chen, and M. Ding, “On the Performance of UAV-Aided Content Caching in Small-Cell Networks with Joint Transmission,” *Electronics*, vol. 10, no. 1040, 2021.

[47] *Crazyflie 2.1*, https://www.bitcraze.io/products/crazyflie-2-1/ (last accessed at Dec 2021).