Drone Helps Privacy: Sky Caching Assisted k-Anonymity in Spatial Querying

Hema Naga Sai Sudha Jagarlapudi[†] Sunho Lim[†*] Jinseok Chae[§*] Gyu Sang Choi[‡] Cong Pu[¶]

[†]T^{2}WISTOR: TTU Wireless Mobile Networking Laboratory, Dept. of Computer Science,
Texas Tech University, Lubbock, TX 79409, {hema-naga-sai-sudha.jagarlapudi, sunho.lim}@ttu.edu

[§]Dept. of Computer Science and Engineering, Incheon National University, Incheon 22012, Korea, jschae@inu.ac.kr

[‡]Dept. of Information and Communication Engineering, Yeungnam University,
Gyeongsan 38541, Korea, castchoi@ynu.ac.kr

[¶]Dept. of Computer Sciences and Electrical Engineering, Marshall University, Huntington, WV 25755, puc@marshall.edu

***Abstract*—As wireless and mobile devices are prevalent and resource-rich, users can frequently query point-of-interests (POIs) and enjoy diverse location-based services (LBS) in infrastructure-based wireless networks. However, users often trade their location privacy with services by sharing their location information with an LBS server, without knowing that the server can fully be trusted. In this paper, we propose a sky caching-aided spatial querying scheme with the support of a flying drone. The basic idea is that both user and its drone collaboratively generate a set of dummy locations to hide the current location of the user from the server. We also propose cache admission control techniques to efficiently cache the queried results to minimize the number of duplicated cached copies. The location privacy is analyzed and measured in terms of the location obfuscation and the size of the convex hull area that consists of dummy locations resided within the cloaking areas of user and drone. We conduct extensive simulation experiments using the OMNeT++ for performance evaluation and comparison. The simulation results show that the proposed approach can reduce the number of queries sent to the server, extend the cloaking area, and improve the location privacy of users significantly.**

***Index Terms*—Cache, Drone, k-Anonymity, Location Privacy, Spatial Query**

I. Introduction

As smartphones are becoming increasingly popular and resource-rich, users[1] run mobile applications (Apps) and enjoy diverse location-based services (LBS), applications, and systems, such as geo-social networking, vehicle navigation, or point-of-interest (POI) queries. For example, smarphone users in the world are about 3.8 billion in 2021 and it is more than 48% of the world's population. In particular, smarphone users in the U.S. are about 280 million in 2020 and it is expected to grow more than 311 million by 2025 [1]. With jumping on the bandwagon, location-based (or geo-targeted) mobile marketings are rapidly growing. For example, e-flyers are broadcasted wirelessly to the users who are closely located to their POIs. Unlike traditional advertisements through the mass media, the markets spent about $17 billion for location-based mobile advertisements and their expenditure will increase more than $38 billion by 2022 [2]. Over 50% of U.S. brands already utilize location data for their business purposes and upgrade or newly deploy LBS applications and systems for marketing aggressively.

When a user requests an LBS, it generates and sends a location-based query (later in short, query) piggybacked with its current location to an LBS server[2]. In this query, it is implicitly assumed that the user has agreed to share the current location with the server without knowing that the server could fully be trusted. In fact, more than 65% of users reveal or share their location with a weather App [3], trading their location privacy with the service. Since the server can collect and store incoming queries in local storage, it might easily infer a certain level of confidential information by analyzing and tracing the locations obtained from users, such as home and working locations, mobility patterns, life preferences, or even health conditions. The server could also release or sell the confidential information extracted from the queries to the third parties for mobile marketing without the consensus of users. An adversary could even compromise the server and misuse the stored location information. In addition, recent smartphone-based social networks and applications often share users' sensitive location information and expedite the privacy breaches [4]. In light of these, spatial cloaking techniques have been widely used by considering location perturbation, obfuscation, anonymity, and their variants [5]–[11].

In this paper, we identify a set of implicit assumptions and potential issues observed in prior LBS literature. (i) **Frequent Queries:** When a user sends a query to a server, it often generates a set of dummy locations resided within a cloaking area to hide its current location from the server. Upon receiving the queried results from the server, the user answers the query and caches the queried results in local storage. Due to blind caching or the lack of cache management mechanism for the queried results, the user tends to send more queries to the server. This can increase the chance of the user revealing its current location to the server. The server may infer the user location by analyzing multiple incoming queries, e.g., inference attack. (ii) **Obsolete User Query Probability:** A set of dummy locations can be selected based on the previous query history. A queried area is virtually divided

[*] Corresponding authors

[1] In this paper, we use the term user and smartphone user interchangeably to refer to a mobile device or a person who carries it.

[2] From a user's point of view, an access point (AP) or base station (BS) is transparent to an LBS server. Thus, we use the terms AP (or BS) and LBS server (later in short, server) interchangeably in this paper.

into a set of rectangle cells and each cell has a user query probability [8]. A set of cells having the same or similar query probabilities is chosen as dummy locations to achieve the maximum entropy. Since the query probabilities are time-varying, however, it is hard (if it is not impossible) for each user to keep the most updated query probabilities in the area. Prior approaches of the user query probability sharing can be deployed. For example, the server can periodically disseminate the user query probabilities, or local access points (APs) can collect and broadcast them. However, the approaches require a non-negligible computing and communication overhead. A local AP can even easily be compromised by an adversary. (iii) **Attack-Prone Peer-to-Peer Queries:** To achieve k-anonymity, a query issuing node often requests a set of locations from its adjacent nodes to collect k - 1 dummy locations to hide its real location. In this approach, there is an implicit assumption that each node will collaboratively forward and share the query result and location information, resulting in another location privacy issue. Since the query is disseminated to the network in an ad hoc manner [12], [13], public key infrastructure (PKI) based on cryptography tools can be deployed to avoid an eavesdropping attack from an adversary. However, the adversary can also conduct a denial-of-service (DoS) attack to interfere with any ongoing communication, such as a selective forwarding attack [14].

In light of these, we propose a novel sky caching-aided spatial querying scheme in the LBS, in which a low-altitude unmanned aerial vehicle (UAV) [15], commercially well known as a drone, is deployed. Drones are increasingly popular to the public because of their versatility, easy installation and interface, and relatively low operating cost. Thus, we envision that each user will equip with a light-weighted, micro-sized, and user-friendly drone embedded into a smartphone in near future. The basic idea is that a user and its personal drone will conduct querying and caching operations collaboratively to protect the location privacy from the server. To the best of our knowledge, no prior research effort has been devoted to deploying drones to protect the location privacy of the user in spatial querying research. Our contributions are summarized in three-fold:

- We newly introduce a use case of the drone by deploying it as an object that can assist in preserving the user's location privacy. The research paradigm of drones has been changed from privacy-defending from a malicious drone, often witnessed in traditional drone-based literature, to privacy-assisting with the help of a drone.
- We propose a sky caching-aided spatial querying scheme and its corresponding operations with the support of a flying drone. Both user and drone collaboratively generate a set of dummy locations for querying operations to hide the current location of the user from the server.
- We also propose three cache admission control (CAC) techniques to efficiently cache the queried results in the local storage of user and drone: Switched, Filtered, and Forward. This approach reduces the duplication of cached copies between the user and drone to increase the accessibility of queried results received from the server.
- The drone-assisted location privacy is analyzed in terms of the location obfuscation and the size of the cloaking area. The size of the extended cloaking area is measured using a convex hull that consists of dummy locations chosen by the user and drone.

We develop a customized discrete-event driven simulator using the OMNeT++ [16]. We extensively conduct simulation-based studies to observe the impact of query interval and anonymity on the caching and location privacy performance. We also measure the performance in terms of the cache hit that consists of local and remote cache hits, the number of cache replacements, the number of queries sent to the server, and the size of the convex hull area. The simulation results show that higher cache hits and the extended cloaking area can reduce the chance of user revealing its location privacy to the server.

The rest of this paper is organized as follows. The prior work is briefly reviewed and analyzed in Section II. We propose and analyze drone-assisted spatial querying and caching strategies and their corresponding operations in Section IV. The proposed scheme is extensively evaluated, analyzed, and discussed in Sections V and VI. The concluding remarks are presented in Section VII.

II. Related Work

A significant research effort has been devoted to developing location privacy-preserving strategies. Location perturbation has been widely used with techniques, including virtual dummies [5], [6], [9], [10], [17], spatial cloaking [7], [18]–[21], differential privacy [11], [22], caching [12], [13], [23]–[27], information theories [28], [29], and their variants, to protect the location privacy of users from an untrusted server or an adversary. We classify and analyze prior location privacy-preserving approaches and their querying operations.

Privacy-Preserving Location Perturbation: The basic idea is to add a random or strategic noise into a location query on purpose to obfuscate the query result. For example, a set of locations including a query issuing user and its neighbors is sent to the server as a part of query parameters to hide the exact location of the user from the server [5], [6]. In location obfuscation [7], [9], [18], a set of dummy or fixed locations is required to form a query. To achieve k-anonymity, a queried area is often geographically enlarged to encompass at least k - 1 dummy or real users' locations. However, k-anonymity may need an extra computation to create or locate k - 1 additional locations and reduce the accuracy of query results replied from the server. In [18], a set of dummy locations is generated without deploying a trusted third-party anonymizer, and its corresponding mobility is also updated based on the density of queried areas to avoid an adversary from tracking the real location. To reduce the communication cost, the amount of unnecessary queried results received from the server is limited by using keywords that do not relate to the queried location data. To remove any trusted entity, however, a function generator located between users and the server periodically distributes transformation parameters to them [21]. These parameters are used to convert real locations to dummy locations and vice versa. In [19], each user specifies a privacy profile containing k-anonymity requirements and the minimum size of the cloaking area. The size of the cloaking area can be variable depending on the distribution of users, indexed by a hierarchy of grid structure. A virtual circle or square-shaped cloaking region is often created to include the locations of user and k - 1 dummies, in which the distribution of dummy locations can be controlled [7]. In [10], a set of dummy locations is generated and filtered out by considering a spatiotemporal correlation in terms of reachability, direction similarity, and in-degree/out-degree to prevent the server from identifying or inferring the dummies. A differential privacy scheme is proposed by perturbing a user location in an LBS network that is virtually divided into a set of hexagon cells [11]. The user's real location is hidden by generating a noise, which becomes a dummy location. Rather than piggybacking the dummy location into the query directly, a cell containing the real and dummy locations and its central location are sent to the server.

Privacy-Preserving Caching: Users cache the LBS data obtained from prior queries in local storage to answer potential future queries without contacting the server. This caching technique can reduce the query latency, the number of queries sent to the server, and the probability of the exact location of the user exposed to the server. In the MobiCrowd [30], each user requests its adjacent users of interested LBS data in a peer-to-peer (P2P) manner and sends a query to the server, only if the requested LBS data are not locally available. The Caché is proposed to improve the location privacy of users by periodically pre-fetching LBS data of large geographic areas, where users would most likely visit [24]. Users can either locally access cached LBS data with a long time-to-live period according to the preferred POIs or only share a general geographic region with the server.

The MobiCache combines caching and k-anonymity techniques to improve the location privacy by increasing the cache hit of the stored LBS data [12]. Here, a network area is virtually divided into a set of cells, and k - 1 dummy locations are randomly selected from the cells that have never been queried. In [13], the cache hit can be improved using a caching-aware dummy selection algorithm by carefully selecting k - 1 dummy locations that either have similar query probabilities or can contribute more to caching performance. All queried results sent from the server are cached for potential future queries. The proposed algorithm is further enhanced by relaxing its implicit assumption of query probabilities that are stable for an entire day [27]. All the cells of query probability are observed over different time slots and modeled by different normal distributions. The selected dummy locations have similar mean and standard deviation values of distribution to that of the query originating location. Unlike pull-based querying from users to the server, cache proxies located in strategic places push the most popular location-related content to visiting users [25]. Here, a cache proxy is located between users and the server stores on-flying queried results to save the query latency and network traffic outgoing to the server. If a requested query is not found in the pushed content, cache miss, the user generates and sends a query containing k - 1 dummy locations to the server.

The Cachecloak plays a role as a trusted third-party anonymizer located between the server and users and replies cached LBS data to the users [23]. When a user generates a query, the Cachecloak sends a predicted user path that is extended until the path is intersected with other paths to avoid a series of user locations from being tracked by the server. In [26], a trusted anonymizer also cloaks the current locations of users and caches frequent queried results to directly answer future queries. Since the trusted anonymizer can be compromised by an adversary, however, each user caches extra queried results and shares them with others through a P2P network, where queried results are extensively searched before the queries are submitted to the server [31].

Privacy-Preserving Drone: Drones are equipped with devices, such as a camera, sensor, or radar and often deployed in a mission-oriented operation (e.g., aerial surveillance or reconnaissance) to track, monitor, or sense important objects and restricted or cyber-critical areas to collect privacy- and security-sensitive information. Single or multiple drones can also play an important role as a cellular-connecting node not only to seamlessly relay ongoing communications for improving the connectivity, but to cache popular data items in a local storage for enhancing accessibility and availability [32], [33]. An object tracking drone records and extracts only meaningful and intended information from the surrounding objects and scenes using recent vision and deep learning techniques to avoid any unnecessary privacy issue [34]. Proposed approaches in [35]–[37] protect users and restricted or cyber-critical areas from an unwanted intrusion and attack by, for example, regulating the drone's altitude and adjusting its attached equipment capability. In [38], the fifth generation (5G) enabled drones 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.

In summary, as far as the authors' best knowledge, the proposed approach that integrates drone-aided aggregate caching and k-anonymity techniques to protect the location privacy of users has not been investigated in the realm of LBS. We also shift a research paradigm of drones from protecting users and restricted areas from a privacy-invading drone into helping in hiding sensitive location information of users from the server and adversary in this paper.

III. System and Threat Models

First, we present a system model of user and drone in general. Users carry a wireless and mobile device (e.g., smartphone) equipped with an on-board global positioning system (GPS) and freely move in the network under a mobility model, such as a random waypoint. Each user is aware of the current

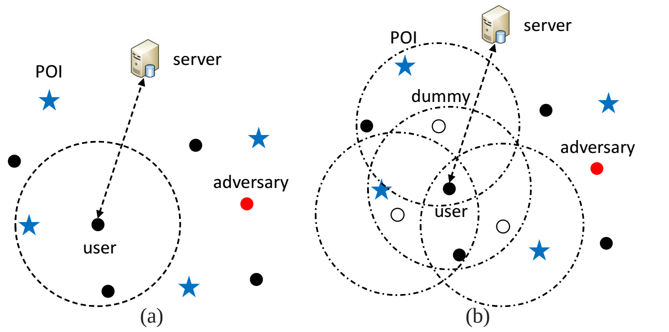


Fig. 1. (a) A user generates a query for a POI and sends the query to a server. Here, a user, adversary, and POI are marked as a black circle, red circle, and star, respectively. A dashed line of circle shows a communication range of user; and (b) A set of dummy locations is generated within a cloaking area that is the same as the communication range of user. Here, the cloaking area is marked by a single-dot dashed line.

location and generates a spatial query for a POI that is closely located to its current location. When the user receives a query result from a server, it stores the query result in local storage. To partially assist the query operation, a programmable and micro-sized drone is able to take off and land back after completing a mission and fly toward a specific location in an autonomous mode [39]. The drone is basically equipped with a wireless communication device and GPS. Depending on a query mission, the drone can also attach any device and sensor, such as a camera, lidar, or radar. However, the drone may have limited computing power, memory storage, and battery power. The drone can also be affected by weather (e.g., wind) and may not correctly navigate to the location requested by the user.

Second, we present a threat model of server and adversary and its potential misbehavior or attack. A server connected with the infrastructure is located between users and their launching drones. The server relays ongoing communications and answers the queries generated by users and drones. When a user generates a query, an anonymizer can be deployed to generate and collect a set of dummy locations to hide the user's real location from the server by using a peer-to-peer communication with other users. Here, the anonymizer is located between the user and server and is often assumed to know and store the users' location information. In this paper, we do not deploy the anonymizer in LBS because this third-party anonymizer could be compromised by an adversary. The server could also be compromised by the adversary and behave maliciously. Since the server receives and stores all queries and their corresponding location information from users and drones, it might conduct an *inference attack* by extracting side information and deduce the users' location information. In addition, the adversary could intercept any on-flying packet in the network. A set of network components is depicted in Subfig. 1(a).

IV. Privacy-Preserving Spatial Querying and Sky Caching

In this section, we first overview a basic k-anonymity based query operation. Then we present drone-based query and caching operations and analyze the location privacy of users.

A. Basic Query Operation with k-Anonymity

When a user generates a query for a POI, it checks its local cache if the queried result is available to answer the query. We consider a static POI that is stationary or does not change its location in a short period, such as a gas station, restaurant, museum, hospital, etc. [40]. If the queried result is not found in the cache, the user sends the query to a server. The query contains the current location of the user (e.g., (x, y)), k - 1 dummy locations, a queried POI (POI_{qry}), and a radius of query area (r), where k is meant to k-anonymity. To prevent the server from inferring the exact location of the user, the user randomly generates k - 1 dummy locations and mixes them with its current location. As depicted in Subfig. 1(b), a set of dummy locations is generated within a virtual circle, cloaking area. In this paper, the size of the cloaking area is the same as the user's communication range. We limit the size of query area in searching the queried POI in the network. This is because the queried POI located far away from the user may not be useful. We also set the size of the query area to the same size as the user's communication range. As the cloaking area increases, dummy locations can be distributed widely in the network. The server becomes harder to infer the exact location of the user. Since the size of the cloaking area is set to the same size as the query area, however, the number of queried POIs replied from the server to the user can be increased. This can increase network traffic and some of queried POIs received may not be fully utilized. Thus, both cloaking and query areas are not increased blindly.

When the server receives a query from a user (u_i), $p_{qry,i}$ $[(x, y)_k$, POI_{qry}, $r]$, it searches the queried POI that is found within the query area for k locations repeatedly. The server collects the locations of queried POI, piggybacks them to a query reply, $p_{rpy,i}$ $[(x, y)_{POI_{qry},k}]$, and sends it back to the user. Upon receiving the query reply from the server, the user extracts one of the queried results to answer the query and caches all the queried results in local storage.

Although multiple locations of queried POI can be found, the server might conduct a *bait attack* by replying a single location to lure the user to the queried POI location and attempt to reveal the current location of user. The server can also attempt to reveal the user path by replying a set of single POI locations consecutively. This is because the server could be compromised and show such misbehavior, or contract with a third party to advertise a specific POI location for a business purpose. In this paper, unless otherwise specified, the user discards the query reply piggybacked with a single POI location to defend the bait attack. The user only accepts the query reply piggybacked with multiple locations of queried POI and chooses the one of POI locations randomly.

B. Drone-Assisted Query and Cache Operations

In this paper, we deploy a drone to assist a query operation as well as protect the location privacy of the user from a server. The basic idea is that the drone helps the user in generating

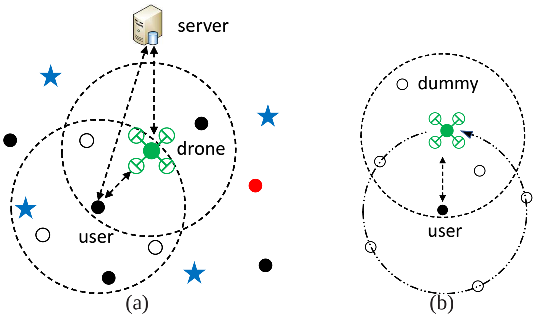


Fig. 2. (a) A drone communicates with both user and server. A communication range of drone is marked as a dashed line; and (b) A drone follows a circular mobility model and generates a set of dummy locations. Here, the drone's circular path is marked as a double-dots dashed line.

a set of dummy locations to confuse the server in identifying and tracking the exact location of the user. Also both user and drone judiciously cache the received queried results to improve the cache hit, reducing the number of queries sent to the server. Here, we consider two types of cache hit, a local cache hit and a remote cache hit. A local cache hit occurs when the queried data item is available in the user's local cache. A remote cache hit implies that the queried data item is available in the drone's local cache.

First, a drone can communicate with both user and server as depicted in Subfig. 2(a). The drone and user can communicate directly if they are located within their communication ranges. When the drone is out of the communication range of the user, it can still communicate with the user through the server indirectly. The communication range of the drone may vary depending on the capacity of the wireless device attached. In this paper, we assume that both communication ranges of user and drone are the same. As shown in Subfig. 2(b), the drone moves under a circular mobility model. As the user moves, the drone flies around the user by keeping a distance between them. The drone can play a role as a dummy user and hide the real user's location from the server. Other mobility models can also be applied to the drone, such as a linear, zig-zag, or random.

Second, when a user (u_i) generates a query that cannot be answered by its local cache, it can either send the query to the server or forward the query to a drone (d_i). If the user randomly decides to forward the query to the drone, it generates k - 1 dummy locations and piggybacks them to the query including its current location. Upon receiving the query, the drone searches its local cache and replies a query result to the user directly, if cached. If not, the drone also generates k' dummy locations within its cloaking area as shown in Subfig. 2(b). The drone-generated dummy locations can be part of or added to the k - 1 dummy locations that the user has originally generated. Then there are several variants in selecting the number of dummy locations. For example, the drone can randomly choose k dummy locations including the current location of the user out of k + k'. To clearly see the impact of the number of dummy locations, we use the combined $k + k'$ dummy locations for the query. The drone piggybacks the combined dummy locations to the query and sends it to the server, $p_{qry,i}$ $[(x,y)_{k+k'}$, POI_{qry}, $r]$. When the server receives the query, it collects the locations of queried POI that are resided within the query area for each $k + k'$ location, piggybacks them to a query reply, $p_{rpy,i}$ $[(x,y)_{POI_{qry},k+k'}]$, and sends it back to the query sender. In fact, the server does not know whether the query is originally generated and sent from the user or drone. Thus, the drone can play a role as another user and hide the real user's location from the server.

When the drone is far away from the user, both cloaking areas of the drone and user can be separated without overlap. Then the server may easily distinguish dummy locations (e.g., k – 1 or k' - 1) where they have been generated from which cloaking area. To avoid generating dummy locations blindly, if the distance between user and drone is temporarily greater than the radius of communication range, both user and drone should not generate dummy locations but wait until they are located to communicate directly.

Third, when the drone receives the query reply, it checks the number of replied POI locations. If a single POI location is replied to, the drone drops it immediately to defend against a luring attack from the server. If not, the drone triggers a cache admission control (CAC) to decide whether it caches the queried POI locations in its local storage and/or forward them to the user.

C. Drone-Assisted Cache Admission Control

We also propose three drone-assisted CAC schemes to efficiently cache the queried results in the user and/or drone: (i) *Switched*, (ii) *Filtered*, and (iii) *Forward*. The primary goal of CAC is to decide whether to cache the queried results or not in order to improve the data accessibility and availability that can be measured by cache hit. In this paper, we consider the concept of aggregate caching in which both caches of user and drone virtually form an aggregate cache [41]. The decision of whether to cache the queried results depends on the user itself as well as the drone. Since the drone is often limited by resource (e.g., smaller cache size) compared to that of the user, it caches only a part of the queried results in local storage. Thus, the basic idea is to cache as many queried results as possible but to avoid storing too many duplicated cache copies in both user and drone to improve the cache hit. A rationale behind these CAC techniques is to reduce the number of queries sent to the server that ultimately decreasing the probability of user location being exposed to the server and enhancing the location privacy of the user.

First, we describe three CAC schemes and their basic caching operations. (i) In the Switched scheme, both user and drone cache the queried results alternately. If the drone has cached the queried results first, then it forwards the next queried results to the user and vice versa. In this scheme, both user and drone try to keep the recent queried results in their caches. (ii) The Filtered scheme allows both user and drone to selectively cache the queried results that are not currently stored in the cache. The rest of queried results that are not cached in the drone is forwarded to the user and vice versa. In this scheme, the user or drone updates its cache first with the recent queried results. The queried results forwarded to either the drone or user may partially be duplicated with the existing cached data items. (iii) In the Forward scheme, the drone forwards the queried results to the user. Then the user caches the queried results forwarded and triggers a cache replacement policy when the cache becomes full. When the existing cached data items are evicted, the user sends them to the drone for caching. When the drone's cache becomes full, it also triggers a cache replacement policy and removes any evicted data item permanently. Here, the least recently used (LRU) is deployed for the cache replacement policy.

Second, we analyze and compare three CAC schemes. All three CAC schemes follow the aggregate caching and try to distribute the queried results to the user and drone to increase the cache hit. Since the user always generates a query, caching recently received the queried results in the user aggressively rather than the drone is important to reduce the query latency. However, we put more weight on the data accessibility than on the query latency. Thus, the drone caches the queried results just like the user. This is because the number of queries sent to the server in case of cache miss directly affects the location privacy of the user. Also all three CAC schemes are sensitive to the level of anonymity (k) because of the limited cache size. As k increases, the size of queried results increases, and more cached data can be evicted for cache replacement. Both Filtered and Forwarded schemes may send more cached data evicted from the user to the drone, or vice versa. The Switched scheme does not forward any evicted cached data to the other part. The overall proposed drone-assisted spatial querying and caching operations are summarized by an event-driven pseudo code in Fig. 3.

```
Notations:
• $u_i$, $d_i$: A query issuing user and its drone, respectively.
• $c_{u,i}$, $c_{d,i}$: A local cache of $u_i$ and $d_i$, respectively.
• $p_{qry,i}$ [$(x,y)_{k+k'}$, $POI_{qry}$, $r$]: A query packet is generated by $u_i$, containing $k$ + $k'$ dummy locations including $u_i$'s current location, a queried POI ($POI_{qry}$), and a query area ($r$).
• $p_{rpy,i}$ [$(x,y)_{POI_{qry},k+k'}$]: A query reply packet containing the query results of $POI_{qry}$.
◇ When $u_i$ generates a query that can be answered by a query result ($r_{qry}$),
    if $r_{qry} \in c_{u,i}$
        Use $r_{qry}$ to answer the query and exit;
    else
        Generate $k$ - 1 dummy locations;
        Randomly send the query either to the server or forward it to the drone, $p_{qry,i}$ [$(x,y)_k$, $POI_{qry}$, $r$];
◇ When $d_i$ receives a query that can be answered by a query result ($r_{qry}$) from $u_i$,
    if $r_{qry} \in c_{d,i}$
        Send $r_{qry}$ back to $u_i$ and exit;
    else
        Generate $k'$ additional dummy locations;
        Send the query to the server, $p_{qry,i}$[$(x,y)_{k+k'}$, $POI_{qry}$, $r$];
◇ When the server receives a query either from $u_i$ or $d_i$,
    Search $POI_{qry}$ and send a query reply back either to the query sender (i.e., $u_i$ or $d_i$), $p_{rpy,i}$ [$(x,y)_{POI_{qry},k+k'}$];
◇ When $u_i$ receives a query reply from the server,
    Conduct a cache admission control (CAC); /* Switched, Filtered, or Forward */
    Cache the queried results in $c_{u,i}$;
    if $c_{u,i}$ is full
        Evict a victim data item from $c_{u,i}$; /* LRU */
        if CAC == Forward
            Send the victim data item(s) to $d_i$, if there is any;
◇ When $d_i$ receives a query reply from the server,
    Conduct a CAC; /* Switched, Filtered, or Forward */
    Cache the queried results in $c_{d,i}$;
    if $c_{d,i}$ is full
        Evict a victim data item from $c_{d,i}$; /* LRU */
```

Fig. 3. A pseudo-code of our proposed drone-assisted spatial querying operations.

D. Analysis of Drone-Assisted Location Privacy

We analyze location privacy with the help of drones in two aspects: location obfuscation and cloaking area. We also measure the location privacy in terms of cache hit and size of convex hull area.

Location Obfuscation: We count a *local* cache miss when a user randomly forwards a query either to the drone or server but the queried data item is not found in the user's cache. The drone may also send the forwarded query from the user to the server if the queried result is not available either, called a *remote* cache miss. The number of queries sent to the server is proportional to the cache misses. In this paper, we deploy three CAC techniques to increase both local and remote cache hits by reducing the duplicated cached copies stored in both user and drone. Higher cache hits not only reduce the number of queries sent to the server but decrease the chance of the user revealing the current location from the server.

Cloaking Area: When a user forwards a query to a drone, it piggybacks k dummy locations including its current location to the query. k - 1 dummy locations are randomly generated within the user's cloaking area. When the drone receives the query, it also generates k' dummy locations resided within its cloaking area. Since the drone moves around the user by following a circular mobility, the cloaking areas of both user and drone are partially overlapped (see Subfig. 2(b)). In this paper, we deploy a convex hull algorithm to approximate the size of combined cloaking areas. A convex hull is the smallest convex polygon containing a given set of points in a two-dimensional area. We use the Graham-Scan based method [42] to construct a convex hull. Given randomly generated $k+k'$ dummy locations, 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. Then the initially chosen location is connected with a location with the minimum angle. We incrementally keep connecting the locations only if they are located to the counterclockwise of the line connecting the previous two locations. This convex hull becomes an area where $k+k'$ dummy locations reside. The bigger the size of the convex hull area is, the harder the server guesses the exact location of the user. Note that the size of the convex hull area is directly proportional to the cloaking area where the dummy locations are distributed. If the dummy locations are widely spread out in the area, the size of the convex hull area increases.

In summary, the proposed drone-assisted spatial querying can improve location obfuscation and extend the cloaking area. A location obfuscation can be enhanced because both user and drone can generate a query and a set of dummy locations

TABLE I
SIMULATION PARAMETERS

Parameter	Values
Network size	2,000 × 2,000 m^2
Number of users	5
Number of POIs	100
User communication range	100, 150 m
User velocity	1 m/sec
User mobility type	Random Waypoint
User cache size	40 slots
Drone velocity	6 m/sec
Drone cache size	20 slots
Data transmission rate	2 Mbps
Anonymity (k)	6 to 22
Cloaking area radius	100, 150 m
Query interval	5 to 500 sec

collaboratively. The server can be confused in identifying and tracing the exact locations of the user. A cloaking area of the user can be extended by including a cloaking area of the drone. A set of dummy locations can be distributed widely in the network. This would take more time for the server to infer the exact location of the user.

V. PERFORMANCE EVALUATION

A. Simulation Testbed

We developed a customized discrete-event driven simulator using the OMNeT++ [16]. A rectangle network is deployed, where a set of users is uniformly distributed. Each user is equipped with an onboard GPS receiver and multiple network interface cards and communicates with a server and drone. A user communicates with its drone wirelessly using a wireless local area networking, e.g., IEEE 802.11p. The radio transmission range is assumed to be 100 or 150 m. The two-ray ground propagation channel is assumed with a data rate of 2 Mbps. We deploy a simple carrier sense multiple access with collision avoidance (CSMA/CA) based medium access for the link layer. For the sake of simplicity, we assume that both user and drone have the same communication range and use a symmetric link. Both user and drone can also communicate with the server directly through an infrastructure network, e.g., a cellular network. A set of POIs is uniformly distributed in the network, where the total number of POIs is set to 100 with a single type. The random waypoint mobility model [43] is deployed to simulate user mobility (e.g., 1 m/sec) without a pause time. Under the mobility model, the user travels toward a randomly selected destination in the network. After the user arrives at the destination, it travels toward another randomly selected destination. The drone follows a circular mobility model with a given speed (see Fig. 2). A set of dummy locations (k) is generated either by user or drone. Both user and drone have their local storage to cache the queried results. The cache size in terms of the number of data items is set to c_{size} and c'_{size} for user and drone, respectively. A set of major simulation parameters and their values are summarized in Table I.

B. Simulation Results

We measure the performance in terms of the cache hit, number of cache replacements, number of queries sent to the server, and size of convex hull area. We also compare the performance of proposed CAC techniques assisted with a drone (*WD*), denoted as *Switched*, *Filtered*, and *Forward*, respectively. The performance without a drone case, denoted as *W/O Drone*, is used as a performance lower bound. The W/O drone case represents a traditional k-anonymity based approach in this paper. Note that the cache hit has been widely used as a performance matrix to measure data accessibility and availability in diverse research areas. In this paper, the primary interest in measuring the cache hit is not only to measure the accessibility and availability of queried results but to see how efficiently the proposed drone-assisted querying and caching operations and CAC schemes work. However, the number of queries sent to the server implies how many queries would be exposed to the server or attacked by potential adversaries. Although we did not apply any potential attack to on-flying queries and queried results sent to/from the server, which is out of our scope, and the number of queries sent to the server indicates the possibility of a location privacy attack. In [12], [13], [30], the cache hit can also be used as one of the major performance matrices to evaluate location privacy.

Cache Hit Ratio: We first measure cache hit by changing query intervals in Fig. 4. Here, the cloaking radius is set to 150 m. As query interval increases, overall cache hits decrease in entire CAC schemes. The performance of caching without a drone shows the lowest cache hit. Since both caches of user and drone can virtually form an aggregate cache, the caching of the user alone without the help of the drone shows the lowest performance. The Switched scheme shows the highest cache hit compared to other CAC schemes. Both user and drone take a turn in caching the queried results received from the server. The cache hit increases because the most recently queried results can be available in either user or drone. The Filtered scheme is to reduce the number of duplicated cached copies in the user and drone. After the drone selectively caches the queried results, it forwards the leftover cached copies to the user and vice versa. When the user forwards the leftover cached copies to the drone, however, the drone may lose the recent queried results for eviction. Thus, the Filtered scheme shows the lowest cache hit among the CAC schemes. In the Forward scheme, the drone stores the queried results evicted from the user's cache. Since the evicted data items are the least recently accessed, the user would less likely access them in a near future. The cache hit shows the performance between that of Switched and Filtered schemes. To clearly see the performance difference, unless otherwise specified, we compare the performance of Switched and Forward schemes in the rest of this paper.

Second, we measure the cache hit by changing the cloaking radii in the Forward and Switched schemes in Fig. 5. As shown in Subfig. 5(a), when the drone is deployed in querying operation, the cache hit increases because the cache of the drone plays a role as an extended cache for the user. The drone also contributes to the cache hit by generating additional dummy locations that ultimately more queried results can be cached and used to answer future queries of the user. In

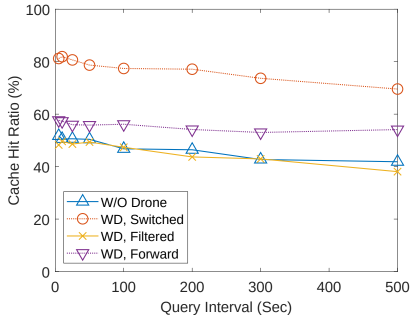
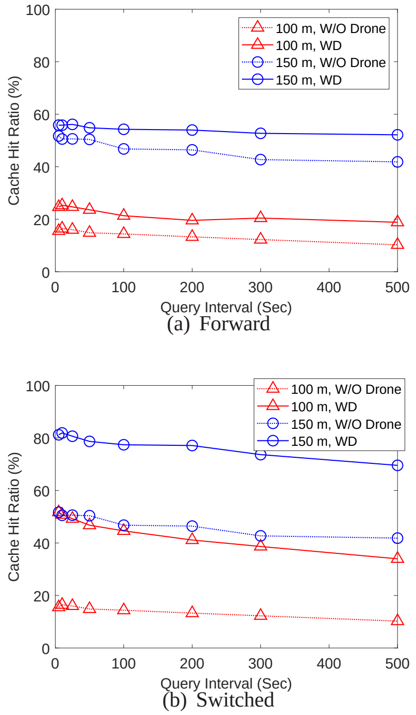


Fig. 4. Cache hit ratio against query intervals for comparing the proposed cache admission control schemes with or without the assistance of a drone. Here, the cloaking radius is set to 150 m.



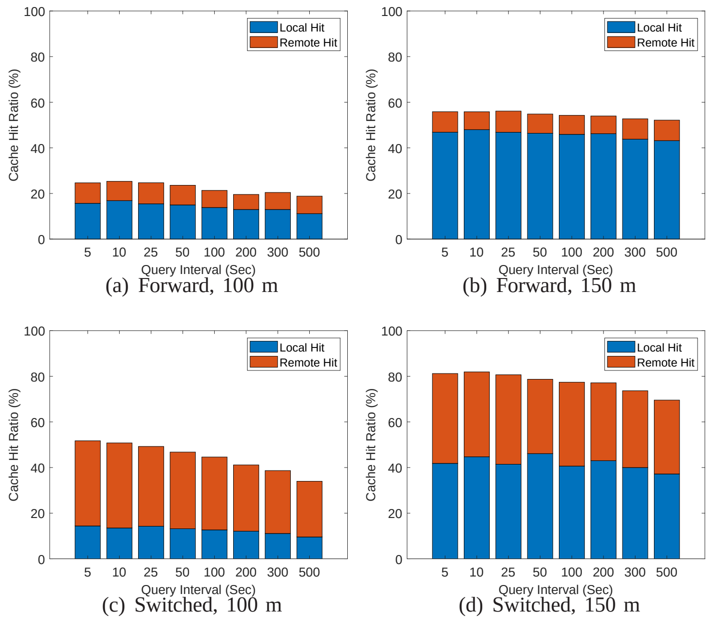
(a) Forward

(b) Switched

Fig. 5. Cache hit ratios against query intervals with different cloaking radii (100 m or 150 m) with (WD) or without (W/O Drone) the assistance of a drone.

addition, a wider cloaking area helps the cache hit. Even without the assistance of a drone, the cache hits increases significantly when the size of the cloaking area becomes wider (i.e., 150 m). This is because more queried POIs can be found and cached to the user. In Subfig. 5(b), the cache hit shows a similar performance pattern depending on the drone assistance and the size of the cloaking radius. However, the Switched scheme shows a higher cache hit than that of the Forward scheme. The Switched scheme improves the cache hit significantly even without the assistance of a drone. Since the queried results are cached in both user and drone alternately, the Switched scheme can achieve the high cache hit because most recently queried results can be available in either user or drone.

Third, we analyze the cache hit in terms of local and remote cache hits in the Forward and Switched schemes in Fig. 6.



(a) Forward, 100 m

(b) Forward, 150 m

(c) Switched, 100 m

(d) Switched, 150 m

Fig. 6. Local and remote cache hits in the Forward and Switched schemes with different cloaking radii.

As shown in Subfigs. 6(a) and (b), the Forward scheme has a higher local cache hit than the remote cache hit. The gap between local and remote cache hits is clear and larger when the cloaking area is wider (i.e., 150 m). This is because the drone forwards the queried results to the user but caches the evicted queried results forwarded from the user. Thus, the drone caches least recently used queried results and the remote cache hit decreases. In Subfigs. 6(c) and (d), the Switched scheme shows more remote cache hits than that of the Forward scheme. When the cloaking area is narrow (i.e., 100 m), the remote cache hit is higher than the local cache hit. Since the drone has additional dummy locations for querying and finds more queried POIs located around the user, the user can use the queried results cached in the drone to answer its queries. As the cloaking area is wider (i.e., 150 m), the local cache hit increases. This implies that both local and remote caches are well utilized.

Fourth, we measure the total number of cache replacements counted in both user and drone in Fig. 7. The percentage is calculated as $\frac{n_{rpl}(c_u)+n_{rpl}(c_d)}{n_{qry}\cdot k}\cdot 100$, where n_{rpl} and n_{qry} are the number of cache replacements in user or drone and the number of queries, respectively. Here, k is set to 9. Since both caches of user and drone are less frequently accessed as the query interval increases, the cache hit ratio decreases and the number of cache replacements increases. The Switched scheme shows less number of cache replacements than that of the Forward scheme for entire query intervals. This is because both user and drone frequently forward the queried results and evicted data items to each other in the Forward scheme, resulting in more number of cache replacements.

Number of Queries Sent to the Server: In Fig. 8, we measure the number of queries sent to the server in the Forward and Switched schemes. When a generated query cannot be

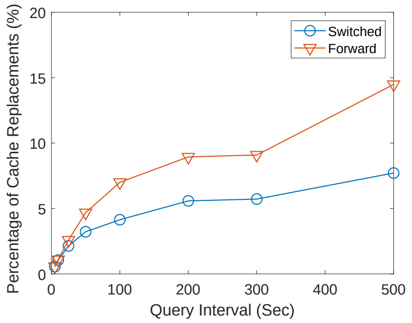
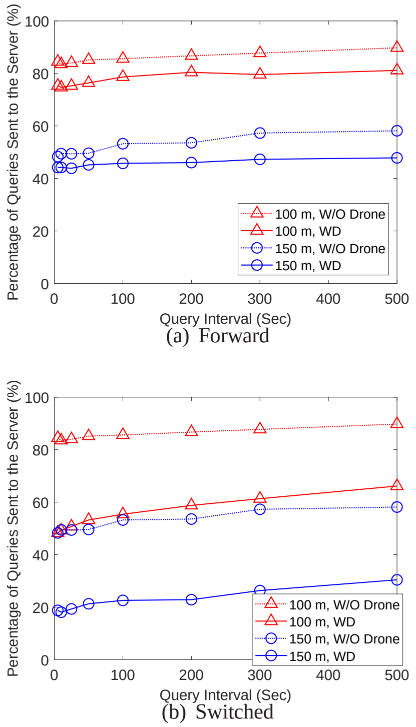


Fig. 7. A percentage of cache replacements. Here, the cloaking radius is set to 100 m.



(a) Forward

(b) Switched

Fig. 8. A percentage of queries sent to the server with different cloaking radii (100 m or 150 m) with (WD) or without (W/O Drone) the assistance of a drone.

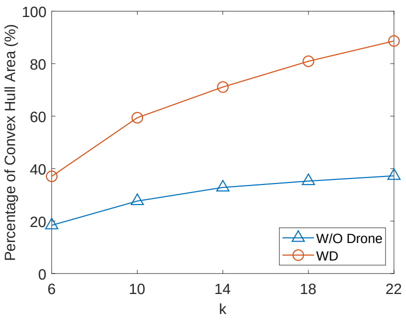


Fig. 9. A percentage of convex hull in the cloaking area(s). Here, the cloaking radius is set to 150 m.

answered by cached results in both user and drone, either user or drone sends it to the server with dummy locations. The more queries are sent to the server, the higher the current location of the user or drone can be exposed to the server. Thus, it is critical for the user and drone not to send as many queries as possible to the server for protecting their location privacy. In Subfig. 8(a), the Forward scheme shows high percentages of queries sent to the server. The percentage decreases when the assistance of a drone is available and the wider cloaking area is deployed. Since the drone caches only evicted queried results from the user, however, a non-negligible percentage of queries is still sent to the server. In Subfig. 8(b), the Switched scheme shows lower percentages than that of the Forward scheme. With the assistance of a drone and the deployment of a wider cloaking area, the percentage decreases significantly. Note that either user or drone sends a query to the server only when the queried result is not available in its cache, which is in fact *cache miss*. In this paper, the percentage of queries sent to the server becomes cache miss and communication cost. A lower cache miss, better location privacy and lower communication cost.

Size of Convex Hull Area: Finally, we measure the percentage of convex hull area in the cloaking area. Although the user randomly generates a set of dummy locations within a given cloaking area, the distribution of dummy locations is critical to obfuscate the server. With the help of a drone, the user is able to obtain extra dummy locations located outside of the cloaking area and ultimately extend the cloaking area. The user and drone randomly generate k and k' dummy locations within their cloaking areas, respectively. Here, k' = k and total k + k' dummy locations can be generated. In this paper, we analyze the cloaking area by building a convex hull and measuring the size of the convex hull area. We use the Graham-Scan based method [42] to construct a convex hull. The convex hull is the smallest convex polygon containing a given set of dummy locations in the network. This convex hull area becomes an obfuscated area that confuses the server in predicting the current location of the user or drone.

Fig. 9 shows the percentage of convex hull area out of the cloaking area. The size of the convex hull area is directly proportional to the anonymity, k and k'. As the anonymity increases, more dummy locations are widely spread out in the cloaking area, and thus, the size of the convex hull area increases. With the assistance of a drone, the percentage increases significantly.

VI. Discussion

A. Drone Feasibility and Capability

In this paper, we simplify the drone as a low-altitude UAV with resource constraints. However, we envision that each user will be equipped with a personal, light-weighted, and user-friendly drone just like a smartphone in a near future. There have been significant research efforts on a portable drone that can conduct an operation or mission under no or minimized control of the user in civil and military environments [44]. A micro- or nano-drone could be integrated with a mobile device (e.g., smartphone) so that users can use it quickly at any time/place. For example, U.S. Army recently purchases a bulk of pocket-sized drones, equipped with GPS-guided autopilot, video recording, and a high-definition camera, to support each soldier's surveillance and reconnaissance capabilities [45]. This nano-drone has been extended to support disaster response as well. On the commercial side, a personal drone carried by the user is integrated with the user's smartphone, called *PhoneDrone*, in which the smartphone runs a path planning software and controls the drone [46]. An autonomous drone activated by a pre-program or a remote user's smartphone conducts surveillance cycles inside the home [47]. Also a tiny quad-propeller drone equipped with cameras and sensors is designed to embed it into a smartphone for portability and usability [48]. Thus, we believe that a drone-assisted approach will be increasingly popular and deployed in diverse research areas.

B. Drone Mobility

We deploy simple and popular mobility models, random waypoint and circular mobilities, for the user and drone, respectively (see Subfig. 2(b)). As the user moves, the cloaking area of the drone also moves. In this paper, the drone generates the dummy locations as soon as it receives the query. If a real-time constraint for query response is not required, the drone can judiciously delay in generating the dummy locations. For example, the drone randomly generates a dummy location every second based on the current location until all k' dummy locations are generated. Since the drone location is time-varying, this delaying operation can exert the same effect in increasing the size of the cloaking area and distributing the dummy locations in a wider cloaking area. Thus, the server would be harder to guess the current location of the user.

A mobility-aware approach or fine-grained mobility management can also be applied to the proposed approach. When the user or drone sends a query attached with its current location and a set of dummy locations to the server, it caches the queried results received from the server. Since the query is answered only by a part of queried results, most queried results are redundant but they can be used to answer future queries. With the help of mobility prediction mechanism [49], the dummy locations resided along the way where the user moves can be chosen proactively. The proposed CAC schemes can also be extended to judiciously cache the queried results that are located along the predicted path. Thus, more queried results cached are used to answer future queries and fewer queries are sent to the server. This would reduce the probability of user location being exposed to the server.

C. Potential Factor and Use Case

We further exploit the extension in our study and investigate practical use cases that will lead to interesting research directions to pursue. Our approach can be deployed in drone-based target tracking and detection operations [50]. In this mission-oriented use case, a drone tracks and detects a static or mobile target without exposing its current location and future trajectory and path to an adversary. When a target is mobile, the drone frequently sends queries to the server or user to update the whereabouts of the target. Thus, the drone generates and piggybacks k - 1 dummy locations to the queries to prevent the server or adversary from finding or guessing its location. Here, the dummy locations are randomly selected within the single cloaking area positioned along the trajectory of the drone. Our collaborative k-anonymity technique can be deployed by considering the dummy locations generated by both user and drone (i.e., $k + k'$) and randomly choosing k - 1 out of them. All or a subset of k -1 dummy locations can be cached for later use.

A rapidly emerging drone-enabled small-cell network [51], [52] is also considered, where a set of drones covers a small-sized area and plays a role as a small-cell or mobile base station. The drones cache popular content to support the capacity of wireless backhauls and reply back to users directly to reduce the query latency. One of the major issues in small-cell networks is how to efficiently divide and cache the popular content at the small-cell base station and drones in a distributed manner. In light of these, our proposed CAC schemes and aggregating caching can be applied by deciding whether to cache the queried results or not at the base station and drones. The drones try not to cache duplicated cached copies to improve the data accessibility and availability.

VII. Concluding Remarks

In this paper, we investigated spatial querying with the help of a drone to protect the location privacy of users from the server. Three implicit assumptions and major issues have been identified often witnessed and overlooked in prior LBS literature. We proposed a drone-assisted querying and its corresponding operations for both user and drone to collaboratively hide the current location of the user from the server by extending the cloaking areas and increasing the number of dummy locations. Three CAC techniques are also proposed to efficiently store the queried results in the storage of user and/or drone by minimizing the duplicated cached copies. We analyzed the drone-assisted location privacy in terms of the location obfuscation and the size of the cloaking area. We conducted extensive simulation-based experiments and performance comparisons by changing key parameters. The simulation results show that the proposed approach can increase cache hit and improve location privacy significantly by reducing the number of queries sent to the server.

Unlike prior prejudice on the drone, such as incurring privacy attacks, the proposed approach showed a potential possibility for the drone to assist privacy-preserving spatial querying. Our research will pave a new research direction in protecting the location privacy of users using a drone in the LBS.

Acknowledgment

The authors would like to thank the anonymous reviewers whose insightful comments and suggestions helped us to improve the paper. This research was supported in part by International Cooperative Research Grant in 2018 from Incheon National University (Incheon, Korea) and Brain Pool program funded by the Ministry of Science and ICT through the National Research Foundation of Korea (NRF–2021H1D3A2A01080645).

References

[1] *Number of smartphone users in the United States, Mar 2021*, https://www.statista.com/.

[2] *Local Mobile Ad Spending, Feb 2018*, http://www.biakelsey.com/.

[3] *Location Based Services : Expected Trends and Technological Advancements, June 2017*, http://geoawesomeness.com/.

[4] H. Li, H. Zhu, S. Du, X. Liang, and X. Shen, "Privacy Leakage of Location Sharing in Mobile Social Networks: Attacks and Defense," *IEEE Trans. on Dependable and Secure Computing*, vol. 15, no. 4, 2018.

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

[6] 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.

[7] 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.

[8] 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.

[9] T. Hara, A. Suzuki, M. Iwata, Y. Arase, and X. Xie, "Dummy-Based User Location Anonymization Under Real-World Constraints," *IEEE Access*, vol. 4, 2016.

[10] H. Liu, X. Li, H. Li, J. Ma, and X. Ma, "Spatiotemporal Correlation-aware Dummy-based Privacy Protection Scheme for Location-based Services," in *Proc. IEEE INFOCOM*, 2017.

[11] J. Hua, W. Tong, F. Xu, and S. Zhong, "A Geo-Indistinguishable Location Perturbation Mechanism for Location-Based Services Supporting Frequent Queries," *IEEE Trans. on Information Forensics and Security*, vol. 13, no. 5, 2018.

[12] X. Zhu, H. Chi, B. Niu, W. Zhang, Z. Li, and H. Li, "MobiCache: When k-anonymity Meets Cache," in *Proc. IEEE GLOBECOM*, 2013.

[13] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Enhancing Privacy through Caching in Location-Based Services," in *Proc. IEEE INFOCOM*, 2015.

[14] C. Pu, S. Lim, B. Jung, and J. Chae, "EYES: Mitigating Forwarding Misbehavior in Energy Harvesting Motivated Networks," *Computer Communications*, vol. 124, no. 2018, 2018.

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

[16] *OMNeT++ Documentation*, https://omnetpp.org/documentation/.

[17] K. Fawaz and K. Shin, "Location Privacy Protection for Smartphone Users," in *Proc. ACM Computer Communications Security*, 2014.

[18] H. Kido, Y. Yanagisawa, and T. Satoh, "An Anonymous Communication Technique using Dummies for Location-based Services," in *Proc. Int'l Conf. on Pervasive Services*, 2005.

[19] M. F. Mokbel, C. Y. Chow, and W. G. Aref, "The New Casper: Query Processing for Location Services without Compromising Privacy," in *Proc. VLDB*, 2006.

[20] 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, 2014.

[21] T. Peng, Q. Liu, and G. Wang, "Enhanced Location Privacy Preserving Scheme in Location-Based Services," *IEEE Systems Journal*, vol. 11, no. 1, 2017.

[22] X. He, G. Cormode, A. Machanavajjhala, C. M. Procopiuc, and D. Srivastava, "DPT: Differentially Private Trajectory Synthesis Using Hierarchical Reference Systems," in *Proc. VLDB Endowment*, 2015.

[23] J. Meyerowitz and R. R. Choudhury, "Hiding Stars with Fireworks: Location Privacy through Camouflage," in *proc. ACM MOBICOM*, 2009.

[24] S. Amini, J. Lindqvist, J. Hong, J. Lin, E. Toch, and N. Sadeh, "Caché: Caching Location-Enhanced Content to Improve User Privacy," CMU CyLab Technical Report CMU-CyLab-10-019, Tech. Rep., 2010.

[25] M. Chen, W. Li, X. Chen, Z. Li, S. Lu, and D. Chen, "LPPS: A Distributed Cache Pushing Based K-Anonymity Location Privacy Preserving Scheme," *Mobile Information Systems*, vol. 2016, 2016.

[26] S. Zhang, K. Choo, Q. Liu, and G. Wang, "Enhancing Privacy through Uniform Grid and Caching in Location-based Services," *Future Generation Computer Systems*, vol. 86, 2018.

[27] X. Mu, H. Shen, and Z. Lu, "A Temporal Caching-aware Dummy Selection Location Algorithm," in *Proc. Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, 2019.

[28] D. Yang, D. Zhang, B. Qu, and P. Cudr-Mauroux, "PrivCheck: Privacy-Preserving Check-in Data Publishing for Personalized Location based Services," in *Proc. ACM Pervasive and Ubiquitous Computing*, 2016.

[29] D. Yang, B. Qu, and P. Cudr-Mauroux, "Privacy-Preserving Social Media Data Publishing for Personalized Ranking-Based Recommendation," *IEEE Trans. on Knowledge and Data Engineering*, vol. 31, no. 3, 2019.

[30] R. Shokri, G. Theodorakopoulos, P. Papadimitratos, E. Kazemi, and J. Hubaux, "Hiding in the Mobile Crowd: Location Privacy through Collaboration," *IEEE Trans. on Dependable and Secure Computing*, vol. 11, no. 3, 2014.

[31] K. Jung and S. Park, "Collaborative Caching Techniques for Privacy-Preserving Location-based Services in Peer-to-Peer Environments," in *Proc. IEEE BIGDATA*, 2017.

[32] 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 WCNC*, 2019.

[33] 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.

[34] 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.

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

[36] 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.

[37] Z. Li, C. Gao, Q. Yue, and X. Fu, "Toward Drone Privacy via Regulating Altitude and Payload," in *Proc. ICCNC*, 2019.

[38] 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.

[39] *Crazyflie 2.0*, https://www.bitcraze.io/crazyflie-2/.

[40] B. Jung, S. Lim, J. Chae, and C. Pu, "VRSense: Validity Region Sensitive Query Processing Strategies for Static and Mobile Point-of-Interests in MANETs," *Computer Communications*, vol. 116, no. 2018, 2018.

[41] S. Lim, W. Lee, G. Cao, and C. R. Das, "A Novel Caching Scheme for Improving Internet-based Mobile Ad Hoc Networks Performance," *Ad Hoc Networks Journal*, vol. 4, no. 2, 2006.

[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] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *Mobile Computing*, Kluwer 1996.

[44] B. Araki, J. Strang, S. Pohorecky, C. Qiu, T. Naegeli, and D. Rus, "Multi-robot Path Planning for a Swarm of Robots that Can Both Fly and Drive," in *Proc. IEEE Robotics and Automation*, 2017.

[45] *Nano Drones: The Tiny Personal Drones the Military is Buying in Bulk, June 2020*, https://dronelife.com/2020/06/18/nano-drones-the-tiny-drones-the-military-is-buying/.

[46] *PhoneDrone: Let your smartphone be your personal drone, June 2015*, https://bgr.com/tech/this-crazy-smartphone-comes-with-a-built-in-camera-drone/.

[47] R. Saracco, *Put a Drone into Your Home*, IEEE Future Directions, Oct 2020.

[48] *This Crazy Smartphone Comes with a Built-in Camera Drone, July 2021*, https://bgr.com/tech/this-crazy-smartphone-comes-with-a-built-in-camera-drone/.

[49] D. Stynes, K. Brown, and C. Sreenan, "A Probabilistic Approach to User Mobility Prediction for Wireless Services," in *Proc. IWCMC*, 2016.

[50] S. Chinthi-Reddy, S. Lim, G. Choi, J. Chae, and C. Pu, "DarkSky: Privacy-Preserving Target Tracking Strategies Using A Flying Drone," *Vehicular Communications*, vol. 35, 2022.

[51] Y. Zeng, R. Zhang, and T. Lim, "Wireless Communications with Unmanned Aerial Vehicles: Opportunities and Challenges," *IEEE Communications Magazine*, vol. 54, no. 5, 2016.

[52] N. Zhao, F. R. Yu, L. Fan, Y. Chen, J. Tang, A. Nallanathan, and V. C. Leung, "Caching Unmanned Aerial Vehicle-Enabled Small-Cell Networks: Employing Energy-Efficient Methods That Store and Retrieve Popular Content," *IEEE Vehicular Technology Magazine*, vol. 14, no. 1, 2019.