SecureFind: Secure and Privacy-Preserving Object Finding via Mobile Crowdsourcing

Jingchao Sun, *Student Member, IEEE,* Rui Zhang, *Member, IEEE,* Xiaocong Jin, *Student Member, IEEE,* and Yanchao Zhang, *Senior Member, IEEE*

Abstract—The plummeting cost of Bluetooth tags and the ubiquity of mobile devices are revolutionizing the traditional lost-and-found service. This paper presents SecureFind, a secure and privacy-preserving object-finding system via mobile crowdsourcing. In SecureFind, a unique Bluetooth tag is attached to every valuable object, and the owner of a lost object submits an object-finding request to many mobile users via the SecureFind service provider. Each mobile user involved searches his vicinity for the lost object on behalf of the object owner who can infer the location of his lost object based on the responses from mobile users. SecureFind is designed to ensure strong object security such that only the object owner can discover the location of his lost object as well as offering location privacy to mobile users involved. The high efficacy and efficiency of SecureFind are confirmed by extensive simulations.

Index Terms—Crowdsourcing, security, privacy, Bluetooth tag, RFID.

1 Introduction

The loss and recovery of physical *object*s is a significant issue around the world. Here an object can refer to anything valuable such as personal assets, children, elderly with dementia, and pets. For example, about 800,000 US children are reported lost each year [1], 113 cell phones are lost/stolen every minute in the US [2], and 19,000 items are lost every year by New York subway and bus riders [2]. The predominant method for recovering lost objects is through a lost-and-found place, where lost objects are turned in and returned to their owners with proper identification. Many (if not most) lost objects, however, may not be found or turned in, and the object owner may not know which of the possibly many lost-and-found places he should resort to. The recovery rate for lost objects is thus very low. For instance, University of California Police reported only 19.3% of lost items recovered [2]. In addition, the recovery latency of this traditional method may be too long to be useful. As an example, by the time a lost object is found and turned in to an airport office, the object owner may have departed to a different city or country.

The plummeting cost and ultra-low energy consumption of Bluetooth tags make them very promising to revolutionize the lost-and-found service. In contrast to RFID tags, Bluetooth tags can directly communicate with any mobile device with a Bluetooth tag or interface within a long communication range up to 160 ft. Besides, Bluetooth tags can be used continuously for one year without changing the battery [3], [4] by adopting the Bluetooth Low Energy (Bluetooth LE) technique, and they only cost several dollars which are often negligible in comparison with the value of lost objects. In the lost-and-found context, a cheap and miniature Bluetooth tag can be attached to every valuable object and contain its owner's identification information. Once finding his object missing, the owner can use his mobile device to search for the corresponding tag. If the tag gets queried, it can report its location or sound an alert to be located. There are growing commercial Bluetooth-based products for locating personal assets, such as Tile [3], BlueBee [5], and StickNFind [4]. These attractive products, however, often require that a lost object be sufficiently close to the searching device. For example, BlueBee tags [5] and StickNFind tags [4] support up to 160 ft and 100 ft, respectively. This inherent range limitation makes it infeasible to recover the lost objects far away from their owners.

A promising solution to overcoming the above range limitation is via mobile crowdsourcing, which refers to the practice of obtaining needed services or data by soliciting contributions from many mobile users. The emergence of mobile crowdsourcing is driven by the skyrocketing growth of mobile devices. For example, the number of mobile-connected devices would exceed the world population in 2013 and hit 10 billion in 2016 [6]. Ubiquitous mobile devices can jointly sense and interact with the physical world at an unprecedented scale, thus enabling many otherwise infeasible applications [7], [8], [9]. One can imagine a service provider offering the object-finding service. An object owner submits an object-finding request as a tag query to the service provider, which in turn forwards the query to selected mobile users referred to as *mobile detectors* hereafter. Every detector then locally broadcasts the query. The tag on the lost object responds to any such query, and the corresponding detector finally sends the tag response and his own location via the service provider to the object owner. Every mobile detector can be rewarded at a fixed rate or in commensurate with the object value. Although the object owner may have to pay for the service, he can recover his valuable object with overwhelming probability.

Crowdsourcing the lost-and-found service faces some great challenges. First, the object in search may be of high value so that the mobile detector discovering it may want to keep it instead of reporting its whereabout to the service provider. Thus we need to alleviate the security concerns of the owners about their lost objects. Second, mobile users may be unwilling to disclose their locations which may indicate too much personal information. Therefore, we must protect the location privacy of mobile users

- *J. Sun, X. Jin and Y. Zhang are with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ, 85287. E-mail: {jcsun,xiaocong.jin,yczhang}@asu.edu.*
- *R. Zhang is with the Department of Electrical Engineering, University of Hawaii, Honolulu, HI, 96822. E-mail: ruizhang@hawaii.edu.*

Manuscript received March 25, 2015; revised August 31, 2015.

to stimulate their participation in the lost-and-found system. Last, both Bluetooth tags and mobile devices are resource-constrained, so the object-finding process should be very efficient in computation and communication, especially for energy-constrained mobile detectors [10]. Although some companies such as Tile [3] and BlueBee [5] are offering the crowdsourced lost-and-found service, they ensure neither object security nor location privacy of involved mobile detectors.

This paper presents SecureFind, a crowdsourced object-finding system that offers strong object security to the object owner and also location privacy to mobile detectors. The essential idea in SecureFind is to let some mobile detectors generate dummy tag responses which are indistinguishable from the real tag response in the eye of the service provider and other mobile detectors. Only the object owner can identify the real tag response, so strong object security can be ensured. In addition, the location of each mobile detector discovering the lost object is kept from the service provider and only disclosed to the object owner under a dynamic pseudonym. So the location privacy of mobile detectors can be well guaranteed.

Our contributions are mainly threefold. First, we are the first to formulate secure and privacy-preserving object finding via mobile crowdsourcing to the best of our knowledge. Second, we propose two solutions to this problem. The basic scheme provides strong object security at the cost of low efficiency. In contrast, the advanced scheme seeks to achieve a middle ground among object security, location privacy, and energy efficiency. Finally, we thoroughly evaluate the performance of our schemes by theoretical analysis and extensive simulations.

2 Related Work

Several schemes have been proposed for tracking and locating lost objects. AutoWitness [11] is a personal asset tracking system that uses an embedded tag with inertial sensor to estimate asset's position change and proactively transmit trajectory data to an external server via cellular link to facilitate asset retrieval. In contrast, SecureFind depends on low-cost Bluetooth tags without any inertial sensor or cellular communication capabilities, thus more suitable for wide adoption. Moreover, Sherlock [12] is a system designed to localize objects with embedded RFID tags in some closed space, which cannot be applied to find lost object in outdoor and is thus orthogonal to SecureFind.

Recent years have witnessed significant research on missing-tag detection [13], [14], [15], [16], [17], [18], [19], [20], [21] and identification [22], [23], [24] in RFID systems. This line of work aims to quickly detect whether or which tags are missing in a large RFID system, while SecureFind targets a totally different problem. In particular, a lost tag in SecureFind is a tag lost by its owner but still in the SecureFind service provider's service region, and SecureFind aims to determine which mobile detector has the lost tag in his coverage in order to locate and retrieve the lost object without revealing such information to either the mobile detector or service provider. In contrast, a missing tag in [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24] means a tag taken away from the monitored area, and the goal there is to determine if any tag is missing. Therefore, existing missing-tag detection schemes are inapplicable to our problem.

Also related is the line of work on privacy-preserving tag identification and authentication in RFID systems, e.g., [25], [26], [27], [28], [29], [30]. These schemes allow efficient identification and authentication of an RFID tag without disclosing any information that can be used to uniquely identify the tag. All the RFID tags belong to the same administrator, and there is no attempt to hide the locations of the RFID tags from the administrator. In contrast, each Bluetooth tag in SecureFind belongs to the corresponding object owner, and its location should be protected from the service provider as well. Therefore, SecureFind differs significantly from these schemes in its aim and scope.

Protecting location privacy in crowdsourcing system is also loosely related to our work. In [31], the authors proposed a novel privacy-preserving framework for spatial crowdsourcing, which allows the service provider to assign spatiotemporal tasks to crowdsourcing workers without sacrificing their location privacy. In addition, Pournajaf *et al.* [32] studied the privacy-preserving spatial task assignment in which crowdsourcing workers obfuscate their locations using spatial cloaking technique. Although both [31] and [32] considered location privacy of crowdsourcing workers, their problems are completely different from ours, and their solutions are not directly applicable.

3 Preliminaries

3.1 System Model

We assume a SecureFind service provider offering the object-finding service via mobile crowdsourcing. The service provider fulfils every object-finding request through a number of mobile users referred to as *mobile detector*s hereafter. Every detector has a mobile device such as a smartphone or tablet to communicate with the service provider and also nearby Bluetooth tags. Almost all mobile devices are having the Bluetooth functionality, and it has been shown in [33] that Bluetooth devices can communicate with each other without explicitly establishing a connection. In addition, nearby mobile detectors can communicate via WiFi-direct, Frequency Hopping, or other available Device-to-Device (D2D) technologies which are widely used in many other applications [33], [34], [35], [36].

An object owner refers to a person who lost a valuable object. We assume that the lost object is attached with a Bluetooth tag hard to remove without breaking the object and use "lost tag" and "lost object" interchangeably henceforth. A Bluetooth tag is a small piece of device with an on-board battery, which can perform simple computation and communicate with nearby mobile devices via Bluetooth. Several off-the-shelf Bluetooth tags are currently commercially available for personal asset tracking, such as Tile [3], StickNFind [4], and BlueBee [5] tags. The cost of a Bluetooth tag is currently around a few dollars [3] and is plummeting due to rapid technological advance and growing market demand. It is thus reasonable to assume that every high-value object will be attached with a Bluetooth tag to enable object finding in the near future. Moreover, we assume that every tag i has a unique ID ID_i known only to its owner.

The object-finding service in SecureFind works as follows. Assume that the object owner knows that his lost object is likely in a possibly large *target area*, e.g., lower Manhattan. He submits to the service provider an object-finding request containing some information about the lost tag and also the target area. The service provider then forwards the object-finding request to all mobile detectors in the target area, each of which in turn locally broadcasts the request. The lost tag responds to any object-finding request intended for it. Every detector hearing a tag response forwards it and his own location via the server to the object owner. Based

on the tag responses, the object owner can derive an approximate location (area) of his lost object, e.g., by multilateral triangulation. Finally, the object owner can go to the derived location and send a tag query in person, in which case the lost tag can respond with its GPS location like a SticknFind tag [4] or sound an alert like a Tile [3] or BlueBee [5] tag. During this process, the object owner may initiate multiple requests to keep track of the dynamic locations of his lost tag (object) which may be carried and in motion. All the system operations are automatically executed without user involvement through an SecureFind app installed on each mobile device.

Sound incentives must be provided to all the involved parties to materialize SecureFind. The service provider can either charge the object owner at a rate commensurate with the object value, and it may also provide free services and profit by web advertisement when its service goes very popular. Every mobile detector can be rewarded either at a fixed rate or in accordance with the object value. Such rewarding mechanisms as perks or badges have been proved to be very successful in soliciting mobile users for crowdsourcing applications like Foursquare. The object owner may need to pay for the service, but he will be able to quickly recover his lost object of high value. Here we assume the existence of such incentive mechanisms and refer readers to existing rich literature such as [37], [38] for incentive design for mobile crowdsourcing.

3.2 Adversary Model

We assume that the service provider is honest-but-curious (HBC) [39], which is a widely adopted assumption for rational service providers. In particular, the service provider is trusted to faithfully follow the protocol execution, but it may have interest in the location of the lost object and also the locations of mobile detectors. In addition, the service provider does not collude with any object owner or mobile detector.

Mobile detectors are curious and also location-sensitive. By curious, we mean that mobile detectors try to locate the lost object and take it away prior to the object owner's arrival. To do so, mobile detectors may attempt to infer whether the lost object is in their vicinity from the information they receive during protocol execution. By location-sensitive, we mean that mobile detectors do not want any party (including the server) to know their accurate locations or equivalently linking their accurate locations to their real IDs.

How to deal with other possible attacks on SecureFind is beyond the scope of this paper. For example, an attacker may jam all radio transmissions, replay intercepted messages, and/or inject bogus messages. Such denial-of-service attacks can target any wireless/mobile system like SecureFind and can be mitigated by existing anti-jamming communication techniques and message authentication.

3.3 Design Objectives

We have the following major design objectives.

- *Correctness*: The object owner should be able to obtain an approximate location of the lost object as long as it is within the transmission range of at least one mobile detector.
- *Object security*: The location of the lost object should be known to the object owner only. Strong object security means that the reported data from detectors that have the lost object in their coverage and those not are indistinguishable, such that no mobile detector can infer whether the lost object is within its coverage.
- *Location privacy*: The mapping between the real ID and location of every mobile detector should be kept from any other party.
- *Efficiency*: The object-finding process should incur low communication and computation overhead.

Note that we do not intend to guarantee the recovery of the lost object, as it depends on whether the lost object is covered by at least one mobile detectors and further the density of mobile detectors in the target area. When the lost object is outside of mobile detectors' coverage, neither SecureFind nor any of the existing systems [3], [4], [5] would be able to recover the lost object.

3.4 Framed Slotted ALOHA Protocol

Our schemes depend on Framed Slotted ALOHA, which is a popular anti-collision MAC protocol adopted by many RFID systems [17], [18], [22], [40], [41]. Since Bluetooth tag is much more powerful than RFID tag, it is reasonable to assume that Bluetooth tag can support Framed Slotted ALOHA with minimal modification. In SecureFind, Framed Slotted ALOHA is executed between one mobile detector and a number of nearby Bluetooth tags and works as follows. First, the mobile detector broadcasts a request with two parameters $\langle r, f \rangle$, where r is a random number, and f is the number of time slots in one frame where the f slots are numbered from 0 to $f-1$. Upon receiving the request $\langle r, f \rangle$, each tag i responds in slot $h(ID_i||r) \mod f$, where ID_i denotes the unique ID of tag i, and $h(\cdot)$ denotes a publicly known hash function. Each of the f time slots can then be an *empty* slot without any tag response, a *singleton* slot with a single tag response, or a *collision* slot with more than one tag responses.

4 A Basic Scheme

In this section, we present a basic scheme for secure and privacy-preserving object finding. The essential idea is to let some mobile detectors in the target area act as *dummy tags* to send dummy tag responses for concealing the real tag response. Since the mobile detectors near the lost object cannot differentiate between real and dummy tag responses, the security of the lost object can be well protected. The major design challenge here is how to let the object owner discover the mobile detectors close to his lost object without drawing the attention of these mobile detectors or the service provider.

We propose an iterative multi-round protocol as a solution. In each round, each mobile detector executes the Framed Slotted ALOHA protocol in Section 3.4 and forwards the execution result to the object owner via the service provider. The object owner then excludes some mobile detectors who are unlikely near his lost object according to their execution results. The protocol completes when no more mobile detectors can be excluded. Finally, the object owner retrieves the locations of the remaining mobile detectors from the server provider using some specific cryptographic technique and then infer the location of his lost object. Our scheme ensures that neither the service provider nor the remaining mobile detectors can learn the location of the lost object.

4.1 Scheme Description

The service provider divides its service region into multiple physical zones, and every mobile detector reports the index of the zone in which it resides when it decides to participate in object finding and whenever it moves into a new zone. The choice of zone size represents the tradeoff between the overhead and location privacy of mobile detectors. On the one hand, a large zone size can alleviate the mobile detectors' concerns about their location privacy to stimulate their participation, but some mobile detectors outside of the target area will participate in object finding and thus incur higher communication and computation overhead. On the other hand, a small zone size enables more accurate selection of mobile detectors but allows the service provider to infer mobile detectors' locations and thus jeopardize their location privacy. To strike a good balance, we suggest to divide the service area based on cellular tower's coverage, which does not reveal any additional information beyond what cellular service providers already know about mobile detectors' locations.

To initiate lost-object finding, the object owner submits an object-finding request $\langle H(\widehat{ID}||r), r, \mathsf{PK}\rangle$ and the target area to the service provider, where $\widehat{ID}$ denotes the ID of the lost tag, r is a random seed, $H(\cdot)$ denotes a publicly known cryptographic hash function, and PK is the object owner's public key. We can also replace PK with a public-key certificate to prevent the service provider from changing PK to its own choice.

Upon receiving the request, the service provider finds the set of candidate zones that enclose the target area and forwards the request to all the mobile detectors in the candidate zones. Each mobile detector can determine whether to participate in the object-finding task according to the sensitivity of his spatiotemporal presence. For example, if a mobile user is present near hospital during working hours, he can choose not to participate in the object-finding task even if the location alone is not sensitive. Each participating mobile detector then locally broadcasts a tag query $\langle H(\widehat{ID}||r), r\rangle$. Here we assume a suitable MAC protocol to resolve potential collisions among mobile detectors; e.g., each mobile detector can wait for some random time before sending the tag query. Every tag seeing such a tag query can check whether it is the intended tag by comparing the hash over its ID and r with the received one, and only the lost tag gets prepared to respond. In addition, each mobile detector returns his location encrypted with PK to the service provider so that the service provider cannot figure out his accurate location. The service provider temporarily buffers these encrypted locations.

The object owner then initiates a polling phase consisting of multiple rounds. Consider round $x \geq 1$ as an example. The object owner sends a polling request $\langle r_x, f\rangle$ via the service provider to each mobile detector, where f denotes the frame length as a fixed system parameter, and r_x is a fresh random seed. Every detector i then locally broadcasts $\langle r_x, f\rangle$. Every other detector hearing the polling request from detector i chooses himself as a dummy tag with probability q, which is a tunable system parameter given by the service provider. Each dummy tag j also generates a random pseudonym ID_j. Let $\mathcal{T}_{x,i}$ denote a set of tags comprising all the dummy tags near detector i and also the lost tag if it hears the polling request from detector i as well. Let $h_1(\cdot), \ldots, h_k(\cdot)$ be k publicly known hash functions, where k is a system parameter. Every tag $j \in \mathcal{T}_{x,i}$ computes k slots to reply, where the αth slot is computed as $\mathsf{s}^{\alpha}_{j,x} = h_\alpha(ID_j||r_x) \mod f$ for all $\alpha \in [1, k]$. During the execution of Framed Slotted ALOHA, every tag j sends a one-bit short response in each of its k computed slots. In the end of round x, detector i obtains a bit vector $\mathsf{V}_{i,x} = \langle v_{i,x}[0], \ldots, v_{i,x}[f-1]\rangle$, where $v_{i,x}[y] = 0$ if slot y is an empty slot and $v_{i,x}[y] = 1$ otherwise. Note that here we do not differentiate between singleton and collision slots, which would require each tag to reply a long multi-bit response and thus incur higher communication overhead. Then detector i sends its bit vector $\mathsf{V}_{i,x}$ to the object owner via the server.

Assuming that there are totally C mobile detectors in the target area, the object owner receives C bit vectors $\{\mathsf{V}_{i,x}\}_{i=1}^{C}$ in round x. He then checks if any mobile detector can be excluded, which is certainly not in the transmission range of his lost tag. To do so, the object owner maintains a candidate detector set. Let $\mathcal{C}_x$ be the candidate detector set at the beginning of round x, where $\mathcal{C}_1 = \{1, \ldots, C\}$. For each detector $i \in \mathcal{C}_x$, the object owners checks if at least one of the bit positions (or slots) $\{h_\alpha(\widehat{ID}||r_x) \mod f\}_{\alpha=1}^{k}$ in $\mathsf{V}_{i,x}$ is zero (or empty), where $\widehat{ID}$ is the ID of his lost tag. If so, the lost tag is certainly not around detector i, and no dummy tag replied in that slot either. So detector i can be safely removed from $\mathcal{C}_x$. The object owner terminates the polling phase if the number of candidate detectors drops to one or remains unchanged after $\tau \geq 2$ polling rounds, where τ is a system parameter. The latter case occurs when the lost tag lies in the coverage of multiple detectors. Also note that the candidate detector set remains confidential to the object owner, and all the C mobile detectors need to broadcast the polling request and process the responses in each round of the polling phase even if some of them may have been confidentially excluded by the object owner.

Once the polling phase is over, the object owner retrieves the encrypted locations of the remaining candidate detectors from the service provider. Finally, he can derive an approximate range for his lost object based on the decrypted detector locations. We can see that the service provider will know which mobile detectors are not excluded. Since the service provider knows the physical zone each mobile detector resides (instead of his real location), it can deduce that the lost object is in one of the physical zones of the remaining detectors. There are two ways to alleviate this security concern. First, the object owner can request the encrypted locations of $c \geq 1$ detectors that include both the remaining detectors and some excluded detectors to confuse the service provider. Second, the object owner can execute an efficient Private-Information-Retrieval protocol [42] to retrieve the encrypted locations of the remaining candidate detectors without revealing whose locations are retrieved.

4.2 Performance Analysis

Now we analyze the performance of the basic scheme.

Correctness. The basic scheme can guarantee that the object owner obtains an approximate location for his lost object as long as it is within the transmission range of at least one mobile detector. Assume that there are totally N mobile users in a region of area S. Also suppose that the number of mobile detectors in any subregion of area s, denoted by $X(s)$, follows a homogeneous spatial Poisson process with intensity N/S: $\mathsf{Pr}(X(s) = k) = \frac{(Ns/S)^k e^{-Ns/S}}{k!}$. Let R denote the transmission range of the lost tag and also mobile detectors. It is easy to see that the basic scheme is correct with probability $1 - \mathsf{Pr}(X(\pi R^2) = 0) = 1 - e^{-\pi N R^2/S}$.

In addition, the basic scheme may incur false positives, which occur when the lost object is not close to any mobile detector (i.e.,

the given target area is wrong), but some dummy tags happen to respond just like the lost tag in each round of the polling phase. The object owner thus will be misled to wrong locations. We can estimate the false-positive probability as follows. Consider any of the C detectors in the target area, say detector i, which has on average $c = \lfloor \pi N R^2/S \rfloor$ other mobile detectors in his transmission range and does not have the lost tag $\widehat{ID}$ there. Since each mobile detector acts as a dummy tag with probability q, there are totally cq dummy tags in detector i's coverage. Recall that the lost tag needs to respond in slots $\{\mathsf{s}_{j,x}^{\alpha} = h_\alpha(ID_j||r_x) \mod f\}_{\alpha=1}^{k}$ in round x if hearing a polling request. Assume that the output of every hash function is uniformly distributed in $[0, f-1]$. Then the average number of distinct slots the lost tag needs to respond is given by

$$\mu = \sum_{l=1}^{k} l \times \frac{\binom{f}{l}}{f^k} \ . \tag{1}$$

As said, each dummy tag also responds in up to k slots uniformly distributed in $[1, f]$. The probability that no dummy tag responds in a particular slot of the lost tag is given by $(1-1/f)^{kqc}$. For detector i to stay in the object owner's candidate detector set in round x, at least one dummy tag needs to respond in each of the μ distinct slots, which occurs with probability $p_{\text{one}} = \left(1-(1-1/f)^{kqc}\right)^{\mu}$. Assume that the polling phase terminates in t rounds. For the false positive to occur, at least one detector needs to survive all the t rounds, which occurs with probability $1-(1-p_{\text{one}}^t)^C$.

Object Security. The basic scheme offers strong object security. In particular, the information the service provider can obtain during object finding includes the initial object-finding request $\langle H(\widehat{ID}||r), r, \mathsf{PK}\rangle$, the polling results in each round, and from which candidate detectors the object owner requested the location. Since the service provider knows neither ID of the lost tag nor the random pseudonym of each dummy tag, he cannot directly infer which detectors have the lost tag in their coverage from the polling results besides knowing that one of the detectors for which the object owner requested the locations does.

Can the service provider do better? To make quantitative analysis possible, we assume that the average number of tags in each detector's communication range are the same, e.g., cq. Under this assumption, the detector with the lost tag in its coverage may observe slightly more non-empty slots than those without during the polling phase. In particular, each detector covering the lost tag, called a real detector hereafter, observes a non-empty slot in each slot with probability $p_1 = 1-(1-1/f)^{(cq+1)k}$, whereas each detector not covering the lost tag, called a fake detector hereafter, does so with probability $p_1' = 1-(1-1/f)^{cqk}$. Although this is only a rough estimate because the number of dummy tags around each mobile detector are most likely different, the service provider may still try to gain some information from the polling results by ranking all the detectors according to the numbers of bit ones in their reported vectors. More specifically, the higher the rank of a detector (i.e., the more bit ones in reported vectors), the more likely the detector is a real one, and vice versa.

Now we analyze the probability distribution of the real detector's rank. Consider a real detector i and a fake detector j in round x as an example. Denote by b_i and b_j the numbers of bit-one positions in their reported vectors $\mathsf{V}_{i,x}$ and $\mathsf{V}_{j,x}$, respectively. Let $u = \min(f, (cq+1)k)$ and $u' = \min(f, cqk)$. The probability that detector i has more bit-one positions than detector j is given by

$$\begin{aligned} p_m &= \mathsf{Pr}(b_i \geq b_j) \\ &= \sum_{z=0}^{u'} \mathsf{Pr}(b_i \geq z) \cdot \mathsf{Pr}(b_j = z) \\ &= \sum_{z=1}^{u'} \sum_{z'=z}^{u} \mathsf{Pr}(b_i = z') \cdot \mathsf{Pr}(b_j = z) \\ &= \sum_{z=1}^{u'} \sum_{z'=z}^{u} \binom{u}{z'} p_1^{z'} (1-p_1)^{u-z'} \binom{u'}{z} {p'}_1^{z} (1-{p'}_1)^{u'-z} \ . \end{aligned} \tag{2}$$

For simplicity, assume that there is only one real detector. The p.d.f. of real detector's rank is then given by

$$\mathsf{Pr}(\text{rank} = r) = \binom{C-1}{r-1} p_m^{r-1} (1-p_m)^{C-r} \ . \tag{3}$$

We can see from Eqs. (2) and (3) that if the number of dummy tags (i.e., cq) is large, p_1 is very close to p_1'. This means that the real detector will be ranked in the middle of all the detectors with high probability, and the object security can thus be guaranteed.

In addition, neither true or fake mobile detectors can distinguish the responses from the lost tag and from dummy tags and thus cannot determine whether the lost tag is in its vicinity.

Location Privacy. The basic scheme offers location privacy to mobile detectors. Specifically, each mobile detector can report a physical zone encompassing his location instead of his real location to the service provider to participate in SecureFind. Therefore, the service provider cannot get the accurate location of any detector. Even if the location of every responding detector is disclosed to the object owner, we can hide the real ID of the detector from the object owner by letting the service provider replace the real ID with a dynamic pseudonym. Since the object owner does not collude with the service provider per our adversary model, the location privacy of every mobile detector is well preserved.

Efficiency. To analyze the communication overhead of the basic scheme, we first derive the expected number t of polling rounds. For any mobile detector not covering the lost tag, the object owner excludes it from the candidate detector set with probability

$$p_e = 1 - p_{\text{one}} = 1 - \left(1-(1-1/f)^{kqc}\right)^{\mu} \ ,$$

where μ is given in Eq. (1). So the object owner can exclude p_{e} fraction of the remaining candidate detectors after each polling round. Assume that the number of candidate detectors drops to one after t rounds. Then we have $Cp_{\text{e}}^t = 1$ and thus

$$t = \lfloor \log_{p_{\text{e}}} \frac{1}{C} \rfloor \ . \tag{4}$$

Each mobile detector sends its encrypted location to the service provider at the beginning, and he also broadcasts a polling request and sends a f-bit vector to the service provider in each polling round. In addition, since each tag needs to reply k one-bit responses in each round, the total communication overhead incurred by tag responses is about $cktC$ bits. Moreover, the object owner sends one object-finding request and t polling messages. Finally, the object owner retrieves λ encrypted detector locations from the service provider.

As for the computation overhead, each tag (dummy or lost)

needs k efficient hash operations in each polling round, leading to $cktC$ hash operations in total. Moreover, each mobile detector performs one public-key encryption, and the object owner needs to carry out one public-key decryption for each non-excluded mobile detector. The most expensive public-key encryptions and decryptions can be done very efficiently on current mobile devices. For example, for the standard Elliptic Curve Integrated Encryption Scheme (ECIES), one point multiplication and two point multiplications are needed for one decryption and one encryption, respectively, and a point multiplication takes less than 7.3 ms on an Android Galaxy Nexus smartphone [43].

5 An Advanced Scheme: Selected Polling

The basic scheme provides strong object security. However, in each polling round, each mobile detector needs to send an f-bit vector to the service provider which incurs large communication overhead and low efficiency. In this section, we present an advanced scheme to strike a middle ground between object security and system efficiency.

5.1 Basic Idea

The advanced scheme stems from an observation about the basic scheme. Specifically, the response from every detector in each polling round is an f-bit vector. The object owner excludes some candidate detectors in each round x by checking the bit values at k positions $\{\mathsf{s}_{j,x}^{\alpha} = h_\alpha(ID_j||r_x) \mod f\}_{\alpha=1}^{k}$, which we refer to as *real* positions. There are at most k real positions because some modular hash values may be the same. Accordingly, we refer to the rest no less than $f-k$ bit positions as *dummy* positions. The dummy positions can effectively hide the real positions so that the detector with the lost object in its coverage cannot tell. The efficiency can be improved if fewer dummy positions are used in each polling round, and the accompanying cost is that real positions will have a higher chance of exposure.

The advanced scheme implements the above thinking by letting the object owner selectively poll fewer than f bit positions in each round, among which the fraction of real positions is adjusted based on the results in previous polling rounds. Intuitively, the more real positions polled in each round, the fewer polling rounds needed to locate the lost tag, the lower the communication and computation overhead, the higher chance of exposing the lost tag, and vice versa. The challenge is how to characterize the exposure of the lost tag and then properly adjust the fraction of real positions.

What is the impact of polling fewer dummy positions on object security? Consider an arbitrary mobile detector, say i. If detector i has the lost tag in his coverage, he is more likely to observe more non-empty slots than other detectors not covering the lost tag. More specifically, assume that the object owner queries ω out of f bit positions, which consists of $\gamma \geq 1$ real positions and $\omega - \gamma$ dummy positions. Recall that each detector on average has $c = \lfloor \pi R^2 N/S \rfloor$ other detectors in his coverage, each acting as a dummy tag with probability q. If detector i covers the lost tag, the probability that a randomly queried bit position having a one (or equivalently the corresponding slot is busy) can be estimated as

$$\begin{aligned} p_1 &= (1-(1-1/f)^{cqk})\frac{\omega-\gamma}{\omega} + \frac{\gamma}{\omega} \\ &= 1-(1-1/f)^{cqk} + (1-1/f)^{cqk}\frac{\gamma}{\omega}\,. \end{aligned} \tag{5}$$

If the lost tag is outside detector i's coverage, the above probability is $p_1' = 1-(1-1/f)^{cqk}$. It is easy to see that $p_1' < p_1$ for $\gamma \geq 1$. As we normally have $\gamma/\omega > k/f$, the gap between p_1 and p_1' becomes more noticeable in the advanced scheme, leading to lower object security. In addition, the larger γ, the more quickly the object owner ruling out the candidate detectors not covering the lost object, the fewer polling rounds needed, the larger the probability gap, the lower object security, and verse versa.

To strike a balance between object security and system efficiency, we let the object owner maximize the number of real positions in each polling round as long as the polling result (i.e., the ω-bit vector) observed by the detector covering the lost object is *statistically indistinguishable* from the one observed by a detector not covering the lost tag. More specifically, let the null hypothesis be that the ω-bit vector obtained by a detector is generated from the binomial distribution $B(\omega, p_1')$, i.e., the theoretical distribution. We can then test the hypothesis using Pearson's chi-squared test [44] with the test statistics given by

$$\chi^2 = \frac{(p_{\text{ob}}-p_1')^2}{p_1'} + \frac{((1-p_{\text{ob}})-(1-p_1'))^2}{(1-p_1')}\,, \tag{6}$$

where p_{ob} is the observed frequency of bit ones, and $p_1' = 1-(1-1/f)^{cqk}$ is the theoretical frequency. Finally, we can compute a p-value from χ^2 using the chi-squared distribution for one degree of freedom, which gives us the probability of observing such difference if the ω-bit vector is generated from $B(\omega, p_1')$.

5.2 Scheme Description

The pre-polling phase of the advanced scheme is exactly the same as that of the basic scheme, so we do not repeat it here for lack of space.

As in the basic scheme, the polling phase in the advanced scheme also consists of multiple rounds. Consider round $x \geq 1$ as an example. The object owner sends a polling request $\langle r_x, f, \mathsf{d}_{x,0}, \ldots, \mathsf{d}_{x,\omega-1}\rangle$ via the service provider to each mobile detector, where f denotes the frame length as a fixed system parameter, r_x is a fresh random seed, and $0 \leq \mathsf{d}_{x,0} < \mathsf{d}_{x,1} < \cdots < \mathsf{d}_{x,\omega-1} \leq f-1$ are the ω bit positions that the object owner intends to poll in round x. These ω bit positions include γ_x real and $\omega - \gamma_x$ dummy positions, and how to choose them will be discussed shortly. Every detector i then locally broadcasts $\langle r_x, f, \mathsf{d}_{x,0}, \ldots, \mathsf{d}_{x,\omega-1}\rangle$. Every other detector hearing the polling request from detector i chooses himself as a dummy tag with probability q which is a system parameter. Let $\mathcal{T}_{x,i}$ denote the set of tags comprising all the dummy tags near detector i and also the lost tag if it is covered by detector i. The Framed Slotted ALOHA protocol is still used to collect tag responses. Every tag $j \in \mathcal{T}_{x,i}$ computes k candidate slots to reply, where the αth slot is computed as $\mathsf{s}_{j,x}^{\alpha} = h_\alpha(ID_j||r_x) \mod f$. Then for each $\mathsf{d}_{x,y}, y \in [0, \omega-1]$, tag j checks if $\mathsf{d}_{x,y} = \mathsf{s}_{j,x}^{\alpha}$ for some α. If so, tag j knows that it should reply a one-bit response in slot y and keeps silent otherwise. In the end of round x, detector i obtains a ω-bit vector $\mathsf{W}_{i,x} = \langle \mathsf{w}_{i,x}[0], \ldots, \mathsf{w}_{i,x}[\omega-1]\rangle$, where $\mathsf{w}_{i,x}[y] = 0$ if slot y is an empty slot and $\mathsf{w}_{i,x}[y] = 1$ otherwise. Then detector i sends $\mathsf{W}_{i,x}$ to the object owner via the service provider.

Given totally C mobile detectors in the target area, the object owner receives $\{\mathsf{W}_{i,x}\}_{i=1}^{C}$ in round x. As in the basic scheme, he maintains a set of candidate detectors which initially contain all the C detectors. After receiving $\{\mathsf{W}_{i,x}\}_{i=1}^{C}$, the object owner

eliminates all the detectors from the candidate set $\mathcal{C}_x$ with each having at least one zero at the γ_x real positions in his polling result. The polling phase stops when the number of candidate detectors drops to one or remains unchanged after $\tau \geq 2$ rounds, where τ is a system parameter.

After the polling phase, the object owner retrieves the encrypted locations of $\lambda \geq 1$ detectors that include both the remaining detectors and some excluded detectors from the service provider. Finally, he can derive an approximate range for his lost object based on the decrypted detector locations as in the basic scheme.

5.3 Choosing Polling Positions

Now we discuss how to choose the ω_x polling positions $\{\mathsf{d}_{x,j}\}_{j=0}^{\omega-1}$ in each round x.

The first step is to determine γ_x, the number of real positions in round x. We propose to derive γ_x based on the C polling results received in all previous rounds such that the expected polling results in round x are statistically indistinguishable from the results generated from the theoretical binomial distribution $B(\omega, p_1')$. In particular, recall that $\mathcal{C}_x$ denote the set of remaining candidate detectors at the beginning of round x. Let $b_{i,x-1}$ be the number of bit ones in $\mathsf{W}_{i,x-1}$ for all $i \in \mathcal{C}_x$, where we set $b_{i,0} = \lceil (1-(1-1/f)^{cqk})\omega \rceil$. As discussed, the probability of any bit position in $\mathsf{W}_{i,x}$ being one for any detector $i \in \mathcal{C}_x$ not covering the lost object can be derived as $p_{i,1} = 1-(1-1/f)^{cqk}$. Then the object owner tries to find $\gamma_{x,i}$ for each detector $i \in \mathcal{C}_x$, the largest number of real positions can be polled in round x, if detector i covers the lost tag. To do so, the object owner initially set $\gamma_{x,i} = 0$. According to Eq. (5), the probability of any bit position in $\mathsf{W}_{i,x}$ being one if detector i covers the lost tag is

$$\hat{p}_{i,1} = (1-(1-1/f)^{cqk}) \cdot \frac{\omega - \gamma_{x,i}}{\omega} + \frac{\gamma_{x,i}}{\omega} .$$

He then computes the expected fraction of bit ones in $\mathsf{W}_{i,x-1}||\mathsf{W}_{i,x}$ as $p_{\text{ob}} = \frac{\hat{p}_{i,1}\omega + b_{i,x-1}}{2\omega}$, the corresponding test statistics χ^2, and finally the p-value (denoted by $p_{\text{val},i}$). If $p_{\text{val},i} > p_{\text{thre}}$, where p_{thre} is the threshold chosen by the object owner, he increases $\gamma_{x,i}$ by one and repeats the above process until find the largest possible $\gamma_{x,i} \leq k$. Finally, he chooses γ_x as the minimum among $\{\gamma_i | i \in \mathcal{C}_x\}$. After determining γ_x, the object owner then constructs $\mathsf{q}_{x,0}, \ldots, \mathsf{q}_{x,\omega-1}$ by randomly choosing γ_x real positions from $\{\mathsf{s}_{j,x}^{\alpha}\}_{\alpha=1}^{k}$ and $\omega - \gamma$ dummy positions. The above process is summarized in Algorithm 1.

5.4 Performance Analysis

The advanced scheme is correct with the same overwhelming probability and offers the same level of location privacy to mobile detectors as the basic scheme.

Object Security. Similar to that in the basic scheme, the service provider may rank the detectors based on the number of bit ones in their reported vectors. Since we normally have $\gamma/\omega > k/f$, the gap between p_1 and p_1' is more noticeable in the advanced scheme than that in the basic scheme. We thus expect that the advanced scheme offers lower object security than the basic scheme does. Since the number of real positions queried in each polling round is jointly determined by the previous polling results and p_{thre}, we have not been able to derive the rank distribution of the real detector. Instead, we evaluate the object security of the advanced scheme in Section 6.

Algorithm 1: Computing γ_x for round x

input : Bit vectors $\{b_{i,x-1} | i \in \mathcal{C}_x\}$, frame length f, p-value threshold p_{thre}
output: γ_x: the number of real positions in round x

```
γ_x ⟵ min(k, ω);
foreach i ∈ C_x do
    γ_{x,i} ⟵ 0, p_{val,i} ⟵ 1;
    p_{i,1} ⟵ 1 − (1 − 1/f)^{cqk};
    while p_{val,i} > p_thre do
        p̂_{i,1} ⟵ (1 − (1 − 1/f)^{cqk}) · (ω − γ_i)/ω + γ_i/ω;
        p_ob ⟵ (p̂_{i,1} ω + b_{i,x−1}) / 2ω;
        χ² = (p_ob − p_{i,1})² / p_{i,1} + ((1 − p_ob) − (1 − p_{i,1}))² / (1 − p_{i,1});
        Update p_{val,i} according to χ² based on chi-square distribution;
        if p_{val,i} > p_thre then
            γ_{x,i} ⟵ γ_{x,i} + 1;
        else
            γ_{x,i} ⟵ γ_{x,i} − 1;
    if γ_{x,i} < γ_x then
        γ_x ⟵ γ_{x,i};
return γ_x;
```

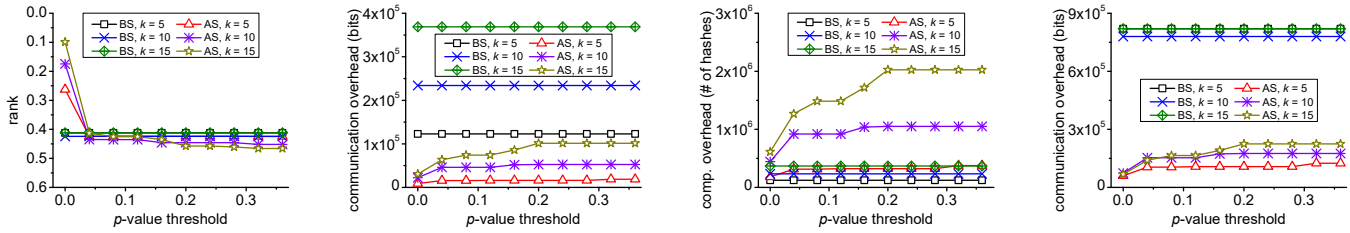
Efficiency. The communication overhead of the advanced scheme depends on the number of polling rounds. Each mobile detector sends its encrypted location to the service provider at the beginning, and he also broadcasts a polling request and sends a ω-bit vector to the service provider in each polling round. In addition, each tag needs to reply $k\omega/f$ one-bit responses on average in each round, so the total communication overhead incurred by tag response is about $ck\omega tC/f$ bits. Moreover, the object owner sends one object-finding request and t polling messages. Finally, the object owner retrieves λ encrypted detector locations from the service provider.

As for the computation overhead, each tag (dummy or lost) needs k efficient hash operations in each polling round, leading to $cktC$ hash operations in total. Because the number of polled real positions in the advanced scheme is smaller than that in the basic scheme, the number of polling rounds is also larger in the advanced scheme, resulting in more hash operations and thus larger tag computation overhead. Moreover, each mobile detector performs one public-key encryption, and the object owner needs to carry out one public-key decryption for each non-excluded mobile detector. As said, such public-key encryptions and decryptions can be efficiently done on modern mobile devices.

Again, since the number of polling rounds is jointly determined by the previous polling results and p_{thre}, we have not been able to derive a closed-form result for the communication and computation overhead of the advanced scheme, which is evaluated via simulations in Section 6.

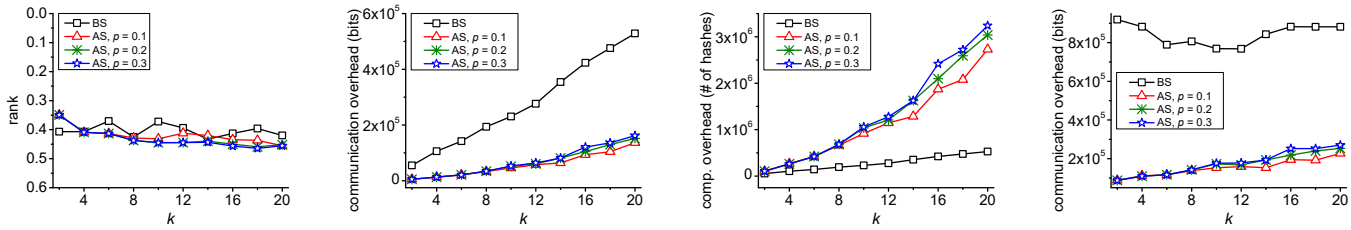
6 Performance Evaluation

In this section, we evaluate the proposed schemes via extensive simulations.



(a) Normalized rank (b) Tag-comm. overhead (c) Tag-comp. overhead (d) Detector-comm. overhead

Fig. 1. Impact of p_{thre}, where BS and AS stand for the basic and advanced schemes, respectively.



(a) Normalized rank (b) Tag-comm. overhead (c) Tag-comp. overhead (d) Detector-comm. overhead

Fig. 2. Impact of k.

TABLE 1
Default Simulation Settings

Para.	Value	Meaning
C	10000	The number of mobile detectors
q	0.9	The probability of acting as dummy tag
f	300	The frame length in Frame Slotted ALOHA
k	10	The number of hash functions
ω	15	The number of polled positions

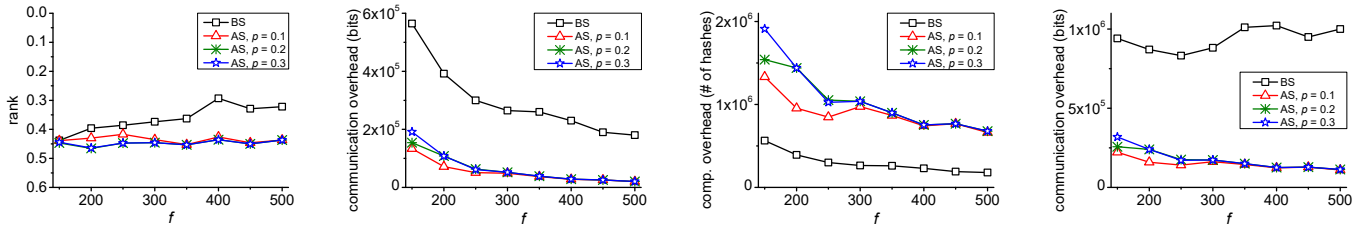
6.1 Simulation Setting

We consider a square region with a side length of $4,000$m, in which 10,000 mobile detectors are distributed uniformly, or 625 mobile detectors per square kilometer. Such density is approximately one sixth of the population density of the downtown area of Austin, TX [45] or one tenth of that of Portland, OR [46]. We assume that each mobile detector acts as a dummy tag with probability $q = 0.9$, which is a tunable system parameter. We set the transmission ranges of both mobile detectors and the lost tag 100m, which is the lower bound of the transmission range of Bluetooth Low Energy technique [47]. In addition, we assume that the number of hash functions is 10, and the frame length in Frame Slotted ALOHA is 300. The two parameters can be adjusted to ensure that the ratio between the number of bit-one positions and the frame length is not too close to zero or one. The number of polled positions ω is set to be 15. Larger ω incurs higher communication overhead but less rounds to find the object. Other simulation parameters are summarized in Table 1 unless stated otherwise.

Since both the basic and the advanced schemes can offer mobile detectors' location privacy and also ensure that the lost object is recoverable almost for sure in all our simulations, our subsequent evaluation focuses on object security, communication overhead, and computation overhead. We assume that the following strategy is adopted by the service provider. On receiving the polling results from all the detectors, the service provider runs the Pearson's chi-squared test as the owner does in the advanced scheme and computes a p-value for each detector. The service provider then ranks all the detectors based on their p-values. The lower the p-value of a detector, the more likely that the lost tag is in his coverage. We then use the relative rank of the detector covering the lost tag to measure the security of the lost object. If the lost tag is covered by multiple detectors, we use the highest rank available. Note that this strategy is a generalization of ranking collectors according to the numbers of bit-one positions discussed in Section 4.2, as it additionally considers the possible different numbers of dummy tags around each collector.

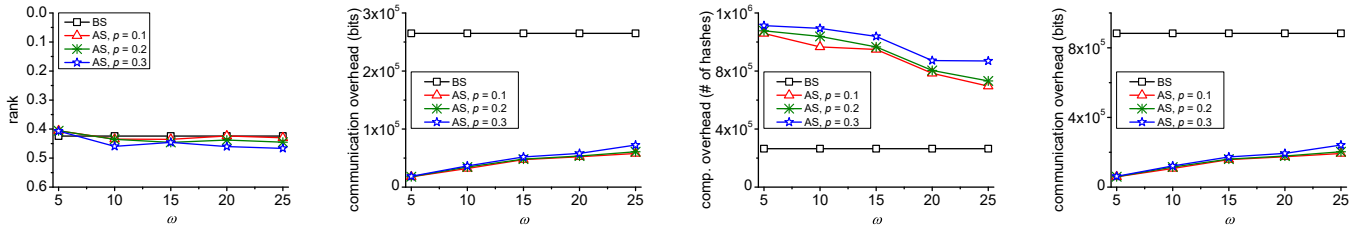
6.2 Simulation Results

Impact of p_{thre}. Figs. 1(a) to 1(d) show the object security in terms of the real detector's normalized rank, tag-communication overhead, tag-computation overhead in the number of hash computations performed, and detector-communication overhead of the basic and advanced schemes, respectively. Since the basic scheme is not affected by p_{thre} (the p-value threshold), its performance is plotted for reference only. We can see from Fig. 1(a) that as p_{thre} increases from 0 to 0.3, the real detector's normalized rank under the advanced scheme increases from around 0.1 to 0.4. This is anticipated, as the higher p_{thre}, the fewer real positions polled in each polling round, the smaller the gap between p_1 and p'_1, the lower the rank of the real detector, the higher object security, and vice versa. In addition, we can see from Figs. 1(b) to 1(d) that the tag-communication overhead, tag-computation overhead, and detector-communication overhead of the advanced scheme all increase as p_{thre} increases. The reason is that higher p_{thre} leads to fewer real positions polled in each round and thus more polling rounds needed to locate the lost object. Moreover, the advanced scheme incurs higher tag-computation overhead than the basic scheme, as the advanced scheme requires more polling rounds than the basic scheme and thus each every tag to perform more hash computations. Finally, Figs. 1(b) and 1(d) show that the advanced scheme incurs lower tag- and detector-communication overhead than the basic scheme. This is of no surprise because much fewer bits are transmitted from each detector to the service provider in each round under the advanced scheme.



(a) Normalized rank (b) Tag-comm. overhead (c) Tag-comp. overhead (d) Detector-comm. overhead

Fig. 3. Impact of f.



(a) Normalized rank (b) Tag-comm. overhead (c) Tag-comp. overhead (d) Detector-comm. overhead

Fig. 4. Impact of ω.

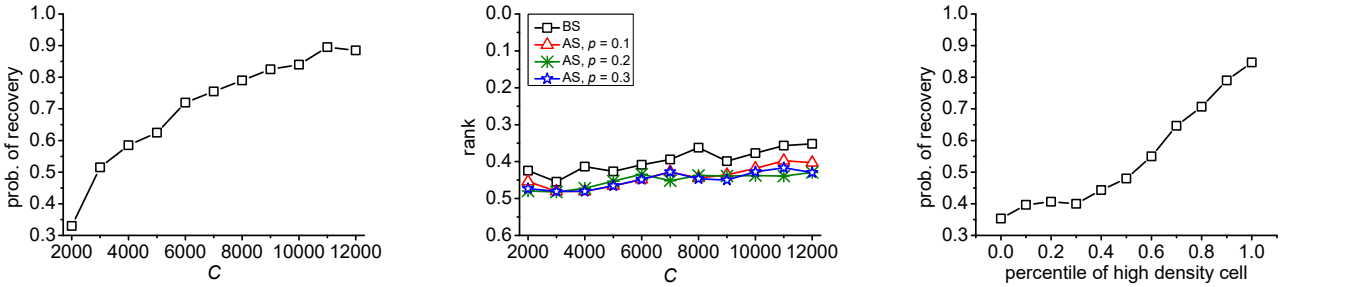
Impact of k. Figs. 2(a) to 2(d) compare the basic and advanced schemes when k (the number of hash functions) varies from 2 to 20. We can see from Fig. 2(a) that the real collector's normalized rank fluctuates as k increases under both schemes. The reason is that the increase in k leads to higher p_1 for the real detector as well as higher p_1' for fake collectors, which nevertheless has little impact on the gap between p_1 and p_1'. In addition, Figs. 2(b) shows that the tag-communication overhead of both schemes increases with k. The reason is that the larger k is, the more slots every tag needs to respond in each polling round, which leads to higher tag-communication overhead. In addition, the advanced scheme incurs much lower communication overhead than the basic scheme, which is expected. Moreover, we can see from Fig. 2(c) that the tag-computation overhead of both schemes increases as k increases and that the advanced scheme incurs higher computation overhead. The reason is that the larger k is, the more hash computations each tag needs to perform in each polling round. In addition, since we generally have $\gamma < k$ in the advanced scheme, it requires more rounds to locate the lost tag, while every tag needs to perform k hash computations in each round.

Impacts of f. Figs. 3(a) to 3(d) show the object security in terms of the real detector's normalized rank, tag-communication overhead, tag-computation overhead in the number of hash computations performed, and detector-communication overhead of the basic and advanced schemes, respectively. Similar to k, f has very limited impact on the normalized rank of the real detector. In addition, we can see from Fig. 3(b) and Fig. 3(c) that the tag-communication and tag-computation overhead of both schemes decrease as f increases. The reason is that the larger f, the fewer polling rounds needed to locate the lost tag, the lower tag-communication and tag-computation overhead for both schemes, and vice versa. In addition, the advanced scheme incurs lower tag-communication overhead but higher tag-computation overhead. Moreover, we can see from Fig. 3(d) that the detector-communication overhead of the advanced scheme decreases as f increases. The reason is that in each polling rounds, each detector needs to transmit a ω-bit vector which is not affected by f. Fewer polling rounds thus lead to lower detector-communication overhead. In contrast, the detector-communication overhead of the basic scheme remains stable as f increases. The reason is that the detector-communication overhead of the basic scheme is the product of the number of polling rounds and the frame length. Since the increase in f leads to the decrease in the number of polling rounds, the detector-communication overhead of the basic scheme is relatively stable.

Impacts of ω. Figs. 4(a) to 4(d) show the impact of ω on the advanced scheme, where the performance of the basic scheme is plotted for reference only. We can see from Fig. 4(a) that ω has very limited impact on the object security. In addition, we can see from Figs. 4(b) to 4(d) that the tag-communication and detector communication overhead both increase and the tag-computation overhead decreases as ω increases.

Impact of mobile detector density. As we mentioned in Section 3.3, SecureFind can find the lost object only if it is within the transmission range of at least one mobile detector, which is affected by the density of mobile detectors. Fig. 5(a) shows the impact of C on the probability that the lost object is within the transmission range of at least one mobile detector, i.e., the probability that the lost object can be recovered. As we can see, the probability of the lost object being found increases as the number of mobile detectors increases, which is expected. In particular, as the number of mobile detectors increases from 2000 to 12000, i.e., the mobile detector density increases from 125 to 750 per square kilometer, the probability of the lost object being found increases from 35% to 90%. We would like to stress that the density of mobile detectors affects only the probability of the lost object being found but not the correctness of SecureFind.

We also evaluated the impact of non-uniform distribution of mobile detectors. In particular, we divided the whole region into 100 equal-size square cells. The mobile detector density in each cell is either 20 per cell or 100 per cell, which correspond to low



(a) Probability of recovery under uniform distribution

(b) Normalized rank

(c) Probability of recovery under non-uniform distribution

Fig. 5. Impacts of C and non-uniform detector distribution.

and high density, respectively. Fig. 5(c) shows the probability of the lost object being recovered with the ratio of high density cells from 0 to 1. We can see that the probability of the lost object being recovered increases from 35% to 90% as the ratio of high density cells increases, which is expected.

Fig. 5(b) shows the impact of C on the object security in terms of the real detector's normalized rank in the basic and advanced schemes, respectively, given that the lost object is within the transmission range of at least one mobile detector. We can see that the rank is relatively insensitive to the change in C or mobile detector density, and both the basic and advanced schemes can offer high object security.

Energy consumption. We measured the latency and the energy consumption of hash operation and Bluetooth transmission (i.e., the two major operations in SecureFind) on two Nexus 7 tablets with Android 4.3. Our experiments show that $7,500,000$ hash operations take 75s and consume 61.25 J of energy on Nexus 7 tablet. This indicates that one hash operation takes 0.01 ms and consumes 8.17×10^{-6} J on average. We measured that the transmission rate of Bluetooth Low Energy is approximately $109 \sim 113$ KB/s, which consumes energy at a rate of $202 \sim 223$ mW. This means that transmitting one bit consumes approximately $2.24 \sim 2.49 \times 10^{-7}$ J of energy.

Based on our measurement results, we further estimate the energy consumption of mobile detector and dummy tag during one object-finding operation. We assume the parameter settings in Table 1 where $f = 300$, $k = 10$, $\omega = 15$, and $p = 0.3$. Consider the simulation results shown in Fig. 1 as an example. It takes 2.56 rounds on average to find the lost object by adopting the basic scheme. In the basic scheme, each mobile detector needs to transmit $2.56 \times 300 = 768$ bits to the service provider, incurring 7.68×10^{-3} J of energy[1]. In addition, each dummy tag needs to perform on average $2.56 \times 10 = 25.6$ hash operations and transmit $2.56 \times 10 = 25.6$ bits, which incur 2.1×10^{-4} J and $5.7 \sim 6.4 \times 10^{-6}$ J of energy, respectively. For the advanced scheme, it takes 10 rounds on average to find the lost object, during which each mobile detector needs to transmit $10 \times 15 = 150$ bits to the service provider and incur 1.5×10^{-3} J of energy consumption. Moreover, each dummy tag needs to perform on average $10 \times 10 = 100$ hash operations and transmit $10 \times 0.5 = 5$ bits, which incur 8.17×10^{-4} J and $1.12 \sim 1.25 \times 10^{-6}$ J of energy consumption, respectively. In general, a typical smartphone's battery stores approximately $15,000 \sim 20,000$ J of energy [49]. Therefore, the operations of SecureFind have negligible impact on a mobile device's battery life.

1. According to [48], the energy consumption of LTE upload link is 1×10^{-5}J/bit.

6.3 Discussion

Our above evaluations have shown that both the basic and the advanced schemes can enable object finding while ensuring the security of the lost object and also the location privacy of the mobile users participating in object finding. Now we discuss some additional factors that may impact SecureFind's performance.

Impact of insufficient dummy tags. SecureFind relies on mobile detectors serving as dummy tags to offer object security. If there are insufficient detectors around the lost object to serve as dummy tags, the mobile detector that receives response from the lost object may be able to infer that the lost object is nearby and the object security cannot be guaranteed. However, this is only possible if the malicious mobile detector can distinguish whether the response he receives is indeed from the lost object or dummy tag. Even if there is no dummy tag near the lost object, as long as there are normal people around, a malicious mobile detector would be unable to to tell whether the response is from the lost object, as it is extremely difficult to tell whether any particular person nearby is serving as mobile detector and dummy tag.

In the most extreme case when there is no people around, the malicious detector can determine that the lost object is nearby. We note that in such cases neither SecureFind nor any existing Bluetooth-tag-based scheme such as Tile [3], BlueBee [5], and StickNFind [4] is capable of recovering the lost object, as there lacks honest mobile detector (including the object owner himself) that has the lost object in the transmission range. Since the owner has already lost the object, it makes no difference between the object being recovered by some malicious mobile detector or unknown person. Therefore, SecureFind can help the object owner recover the lost object if the mobile detector density is not extremely low and does not cause any extra damage to the object owner otherwise.

Impact of detector mobility. During the object finding process, some mobile detectors and dummy tags may move into or out of the transmission range of the lost object due to mobile detector's mobility, which may affect the object-finding result in different ways. First, some dummy tags may move into or out of the transmission ranges of the mobile detectors that collect polling result. For any mobile detector that collects polling result, the increase (or decrease) in the number of surround dummy tags will result in the increase (or decrease) in the number of bit-one positions in bit vector at each round, which makes it less

(or more) likely for the object owner to filter out fake detectors at the end of object-finding process and thus more (or fewer) false positives. Second, an initially real detector may move out of the transmission range of the lost object before the end of the object-finding process, making the object owner unable to find the lost object via this particular detector. Third, an initially fake detector may move into the transmission range of the lost object before the end of the object-finding process. If the detector is not ruled out by the polling results before the movement, this fake detector become a real detector and would help the owner find the lost object.

We expect that the above events happen rarely in practice in due to the low latency of the polling phase in both the basic and the advanced schemes. In particular, each slot takes 321 μs in Slotted ALOHA according to [21]. Under the parameter settings in Table 1, each polling round needs 96.3 ms and 4.8 ms for the basic and advanced schemes, respectively. Take the simulation results shown in Fig. 1 as an example, it takes about 250 ms and 48 ms to finish all polling rounds for the basic and advanced schemes, respectively. Since our simulation results show that a single object-finding process takes less than one second in most cases, we expect detector mobility has very limited impact on SecureFind's performance.

7 Conclusion

This paper presented the design, analysis, and evaluation of SecureFind, the first secure and privacy-preserving crowdsourced object-finding system. In particular, we first introduced a basic scheme which provides strong object security at the cost of system efficiency, and then presented an advanced scheme to strike a good balance between object security and system efficiency. Detailed simulations confirmed that SecureFind can enable very fast and efficient object finding while ensuring the security of the lost object and also the location privacy of the mobile users participating in object finding.

There are still many open challenges to tackle. For example, in our current design, all the mobile detectors in the target area specified by the object owner need to participate in object finding. Since some of them may have overlapping coverage, there may be significant room for reducing the communication and computation overhead. One possible solution is to let the service provider select the minimum number of mobile detectors that can jointly cover the target area. This solution, however, requires the service provider to know more accurate locations of mobile detectors. Such tradeoff between system efficiency and location privacy deserves careful investigation. In addition, our current design assumes that mobile detectors are honest-but-curious. There may be dishonest mobile detectors who report fake search results to earn reward without actually performing the object search. How to catch and then punish such dishonest mobile detectors is nontrivial and may conflict with the location-privacy requirement of mobile detectors. We hope that this paper can stimulate further interest in crowdsourced object finding and other exciting mobile crowdsourcing applications.

Acknowledgments

This work was supported in part by the US National Science Foundation under grants CNS-1320906, CNS-1421999 and CNS-1514381. The authors would like to thank anonymous reviewers for their constructive and helpful advice.

References

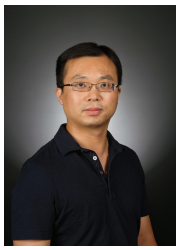
[1] http://www.cnn.com/2013/10/22/us/lost-children-fast-facts/.
[2] http://www.micro-trax.com/statistics/.
[3] http://www.thetileapp.com/.
[4] https://www.sticknfind.com/.
[5] http://www.indiegogo.com/projects/bluebee-a-lost-and-found-in-your-pocket.
[6] "Cisco visual networking index global mobile data traffic forecast update 2012-2017." [Online]. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html
[7] R. Zhang, J. Zhang, Y. Zhang, and C. Zhang, "Secure crowdsourcing-based cooperative spectrum sensing," in *INFOCOM'13*, Turin, Italy, Apr. 2013.
[8] D.-H. Shin, S. He, and J. Zhang, "Joint sensing task and subband allocation for large-scale spectrum profiling," in *INFOCOM'15*, Hongkong, Apr. 2015.
[9] R. Zhang, J. Sun, Y. Zhang, and C. Zhang, "Secure spatial top-k query processing via untrusted location-based service providers," *Dependable and Secure Computing, IEEE Transactions on*, vol. 12, no. 1, pp. 111–124, Jan. 2015.
[10] S. He, J. Chen, Y. Sun, D. Yau, and N. K. Yip, "On optimal information capture by energy-constrained mobile sensors," *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 5, pp. 2472–2484, Jun. 2010.
[11] S. Guha, K. Plarre, D. Lissner, S. Mitra, B. Krishna, P. Dutta, and S. Kumar, "Autowitness: Locating and tracking stolen property while tolerating GPS and radio outages," in *SenSys'10*, Zurich, Switzerland, Nov. 2010.
[12] A. Nemmaluri, M. Corner, and P. Shenoy, "Sherlock: automatically locating objects for humans," in *MobiSys'08*, Breckenridge, CL, Jun. 2008, pp. 187–198.
[13] D. Hush and C. Wood, "Analysis of tree algorithm for RFID arbitration," in *IEEE ISIT'98*, Cambridge, MA, Aug. 1998, pp. 107–107.
[14] C. Law, K. Lee, and K. Siu, "Efficient memoryless protocol for tag identification," in *ACM DIAL-M'00*, Boston, MA, Aug. 2000, pp. 75–84.
[15] F. Zhou, C. Chen, D. Jin, C. Huang, and H. Min, "Evaluating and optimizing power consumption of anti-collision protocols for applications in RFID systems," in *ACM ISLPED'04*, Newport, CA, Aug. 2004, pp. 357–362.
[16] J. Myung and W. Lee, "Adaptive splitting protocols for RFID tag collision arbitration," in *ACM Mobihoc'06*, Florence, Italy, May 2006, pp. 202–213.
[17] C. Tan, B. Sheng, and Q. Li, "How to monitor for missing RFID tags," in *ICDCS'08*, Beijing, China, Jun. 2008, pp. 295–302.
[18] ——, "Efficient techniques for monitoring missing RFID tags," *IEEE Transactions on Wireless Communication*, 2010.
[19] W. Luo, S. Chen, T. Li, and S. Chen, "Efficient missing tag detection in RFID systems," in *INFOCOM'11*, Shanghai, China, Apr. 2011.
[20] W. Luo, S. Chen, Y. Qiao, and T. Li, "Missing-tag detection and energy-time tradeoff in large-scale RFID systems with unreliable channels," *IEEE/ACM Trans. Netw.*, vol. 22, no. 4, pp. 1079–1091, Aug. 2014.
[21] W. Luo, S. Chen, T. Li, and Y. Qiao, "Probabilistic missing-tag detection and energy-time tradeoff in large-scale RFID systems," in *MobiHoc'12*, Hilton Head, USA, May 2012.
[22] T. Li, S. Chen, and Y. Ling, "Identifying the missing tags in a large RFID system," in *ACM Mobihoc'10*, Chicago, IL, Sep. 2010, pp. 1–10.
[23] R. Zhang, Y. Liu, Y. Zhang, and J. Sun, "Fast identification of the missing tags in a large RFID system," in *IEEE SECON'11*, Salt Lake City, Utah, June 2011.
[24] Y. Zheng and M. Li, "P-MTI: physical-layer missing tag identification via compressive sensing," in *INFOCOM'13*, Turin, Italy, Apr. 2013.
[25] L. Yang, J. Han, Y. Qi, and Y. Liu, "Identification-free batch authentication for RFID tags," in *ICNP'10*, Kyoto, Japan, Oct. 2010.
[26] T. Li, W. Luo, Z. Mo, and S. Chen, "Privacy-preserving RFID authentication based on cryptographical encoding," in *INFOCOM'12*, Orlando, USA, Mar. 2012.
[27] L. Lu, J. Han, R. Xiao, and Y. Liu, "Action: Breaking the privacy barrier for RFID systems," in *INFOCOM'09*, Rio de Janeiro, Brazil, Apr. 2009.
[28] Q. Yao, Y. Qi, J. Han, J. Zhao, X. Li, and Y. Liu, "Randomizing RFID private authentication," in *PerCom'09*, Galveston, TX, Mar. 2009.
[29] B. Alomair, A. Clark, J. Cuellar, and R. Poovendran, "Scalable RFID systems: a privacy-preserving protocol with constant-time identification," in *DSN'10*, Chicago, IL, Jun. 2010.
[30] H. Tan, S. Jha, D. Ostry, J. Zic, and V. Sivaraman, "Secure multi-hop network programming with multiple one-way key chains," in *WiSec'08*, Mar. 2008, pp. 183–193.

[31] H. To, G. Ghinita, and C. Shahabi, "A framework for protecting worker location privacy in spatial crowdsourcing," in *VLDB'14*, Hangzhou, China, Jun. 2014.
[32] L. Pournajaf, L. Xiong, V. Sunderam, and S. Goryczka, "Spatial task assignment for crowd sensing with cloaked locations," in *MDM'14*, Brisbane, Australia, Jul. 2014.
[33] Z. Yang, B. Zhang, J. Dai, A. Champion, D. Xuan, and D. Li, "E-SmallTalker: A distributed mobile system for social networking in physical proximity," in *ICDCS'10*, Genoa, Italy, June 2010, pp. 468–477.
[34] J. Sun, R. Zhang, and Y. Zhang, "Privacy-preserving spatiotemporal matching," in *IEEE INFOCOM'13*, Turin, Italy, Apr. 2013.
[35] J. Sun, X. Chen, J. Zhang, Y. Zhang, and J. Zhang, "SYNERGY: A game-theoretical approach for cooperative key generation in wireless networks," in *IEEE INFOCOM'14*, Toronto, Canada, Apr. 2014.
[36] R. Zhang, J. Sun, Y. Zhang, and X. Huang, "Jamming-resilient secure neighbor discovery in mobile ad hoc networks," *Wireless Communications, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, Jun. 2015.
[37] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *MobiCom'12*, Istanbul, Turkey, Aug. 2012.
[38] D. Zhao, X.-Y. Li, and H. Ma, "How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint," in *INFOCOM'14*, Toronto, Canada, Apr. 2014.
[39] O. Goldreich, *The Foundations of Cryptography*. Cambridge University Press, 2004, vol. 2.
[40] V. Shah and V. Wong, "Cardinality estimation in RFID systems with multiple readers," *Wireless Communications, IEEE Transactions on*, vol. 10, no. 5, pp. 1458–1469, May 2011.
[41] L. Zhang, J. Zhang, and X. Tang, "Assigned tree slotted aloha RFID tag anti-collision protocols," *Wireless Communications, IEEE Transactions on*, vol. 12, no. 11, pp. 5493–5505, Nov. 2013.
[42] G. Asharov, Y. Lindell, T. Schneider, and M. Zohner, "More efficient oblivious transfer and extensions for faster secure computation," in *ACM CCS'13*, Berlin, Germany, Nov. 2013.
[43] X. Fan and G. Gong, "Securing nfc with elliptic curve cryptography – challenges and solutions," in *RFIDSec'13 Asia*, Guangzhou, China, Nov. 2013.
[44] H. Chernoff and E. L. Lehmann, "The use of maximum likelihood estimates in χ^2 tests for goodness of fit," *Ann. Math. Statist.*, vol. 25, no. 3, pp. 423–630, 1954.
[45] http://zipatlas.com/us/tx/austin/zip-code-comparison/population-density.htm.
[46] http://zipatlas.com/us/or/portland/zip-code-comparison/population-density.htm.
[47] https://en.wikipedia.org/wiki/Bluetooth_low_energy.
[48] J. Huang, F. Qian, A. Gerber, M. Mao, S. Sen, and O. Spatscheck, "A close examination of performance and power characteristics of 4G LTE networks," in *MobiSys'12*, Low Wood Bay, Lake District, UK, Jun. 2012.
[49] http://www.bbc.com/future/story/20130227-what-is-killing-smartphones.

Jingchao Sun received the B.E. in Electronics and Information Engineering and the M.E. in Communication and Information System from Huazhong University of Science and Technology, China, in 2008 and 2011, respectively. He is currently a Ph.D. student in School of Electrical, Computer, and Energy Engineering at Arizona State University. His primary research interests are network and distributed system security and privacy, wireless networking, and mobile computing.

Rui Zhang received the B.E. in Communication Engineering and the M.E. in Communication and Information System from Huazhong University of Science and Technology, China, in 2001 and 2005, respectively, and the PhD degree in electrical engineering from the Arizona State University, in 2013. He was a software engineer in UTStarcom Shenzhen R&D center from 2005 to 2007. He has been an assistant professor in the Department of Electrical Engineering at the University of Hawaii since July 2013. His primary research interests are network and distributed system security, wireless networking, and mobile computing. He is a member of IEEE.

Xiaocong Jin received the B.E. in Information Engineering from Shanghai Jiao Tong University, China, in 2009. He received the M.S. in Information, Production, and Systems Engineering from Waseda University, Japan, in 2010. He also received the M.S. in Signal and Information Processing from Shanghai Jiao Tong University, China, in 2012. Currently he is a Ph.D. student in School of Electrical, Computer, and Energy Engineering at Arizona State University. His primary research interests are network and distributed system security and privacy, wireless networking, and mobile computing.



Yanchao Zhang received the B.E. in Computer Science and Technology from Nanjing University of Posts and Telecommunications in 1999, the M.E. in Computer Science and Technology from Beijing University of Posts and Telecommunications in 2002, and the Ph.D. in Electrical and Computer Engineering from the University of Florida in 2006. He is an Associate Professor in School of Electrical, Computer and Energy Engineering at Arizona State University. His primary research interests are network and distributed system security, wireless networking, and mobile computing. He is an Editor of IEEE Transactions on Mobile Computing, IEEE Transactions on Vehicular Technology, and IEEE Wireless Communications. He was also a TPC Co-Chair of Communication and Information System Security Symposium, IEEE GLOBECOM 2010. He received the NSF CAREER Award in 2009 and is a senior member of IEEE.