AnonySense: Opportunistic and Privacy-Preserving Context Collection

Apu Kapadia[1], Nikos Triandopoulos[2], Cory Cornelius[1], Daniel Peebles[1], and David Kotz[1]

[1] Institute for Security Technology Studies, Dartmouth College, Hanover, NH 03755, USA
[2] Department of Computer Science, University of Aarhus, 8200 Aarhus N, Denmark

Abstract. Opportunistic sensing allows applications to "task" mobile devices to measure context in a target region. For example, one could leverage sensor-equipped vehicles to measure traffic or pollution levels on a particular street, or users' mobile phones to locate (Bluetooth-enabled) objects in their neighborhood. In most proposed applications, context reports include the time and location of the event, putting the privacy of users at increased risk—even if a report has been anonymized, the accompanying time and location can reveal sufficient information to deanonymize the user whose device sent the report.

We propose AnonySense, a general-purpose architecture for leveraging users' mobile devices for measuring context, while maintaining the privacy of the users. AnonySense features multiple layers of privacy protection—a framework for nodes to receive tasks anonymously, a novel blurring mechanism based on tessellation and clustering to protect users' privacy against the system while reporting context, and k-anonymous report aggregation to improve the users' privacy against applications receiving the context. We outline the architecture and security properties of AnonySense, and focus on evaluating our tessellation and clustering algorithm against real mobility traces.

1 Introduction

Opportunistic sensing has been gaining popularity, with several systems and applications being proposed to leverage users' mobile devices to measure environmental context. In these systems, applications can *task* mobile nodes (such as a user's sensor-equipped mobile phone or vehicle) in a target region to report context information from their vicinity. With opportunistic sensing, applications need not rely on a static sensor deployment, and can instead glean context from any region that mobile nodes visit. Applications of opportunistic sensing include collecting traffic reports or pollution readings from a particular street [19], locating Bluetooth-enabled objects with the help of users' mobile devices [11], and even inferring coffee-shop space availability [35]. Examples of opportunistic-sensing systems include *CarTel* [19], *Mobiscopes* [1], *Urbanet* [32], *Urban Sensing* [6], *SenseWeb* [33] and our own *MetroSense* [5] at Dartmouth College. These systems predominantly rely on mobile nodes whose carriers are humans (or their personal vehicles) in an urban environment, thus putting the privacy of users at risk. For example, location and time, which are often included in context reports, reveal movement patterns of the reporting users.

THIS COPY IS THE AUTHORS' PRE-PUBLICATION VERSION; it may differ slightly from the official published version.
Proceedings of the International Conference on Pervasive Computing (Pervasive), May 2008.
doi:10.1007/978-3-540-79576-6_17.

In many opportunistic-sensing applications, knowing the identities of the devices is unnecessary. Indeed, to protect users' privacy, Tang et al. [35] take the approach of suppressing the node's identity from reports, and Calandriello et al. [3] support pseudonymous or anonymous reports. Unfortunately, knowing users' movement patterns is often enough to deanonymize their reports [26]. Furthermore, a report taken inside Alice's office, for example, allows one to infer that Alice was likely in her office, even if her name was suppressed from the report. We assume that *both* the system and applications may attempt to deanonymize users using information contained in reports, and have developed a system called AnonySense to protect users against these privacy threats. Specifically, no entity should be able to link a report to a particular user.

We leverage k-anonymity. To improve users' privacy, *k-anonymity* [34] requires that at least k reports are combined together before being revealed, adding enough "confusion" in the data to make it difficult to pinpoint exact times and locations (and sensor data) for the individuals reporting the data. Even if an adversary knows that a user Alice has reported data, the attacker is unable to distinguish Alice's report from $k-1$ other reports. Existing server-based techniques apply location and time blurring (known as *spatial* and *temporal cloaking*) for combining reports to provide k-anonymity [17, 15, 12, 23, 29]. The main challenge with this approach, however, is that users must reveal their private information to a trusted server, resulting in a single point of failure with complete knowledge of users' sensitive information. Approaches that aim to improve the privacy against the system suggest the use of a peer-to-peer (p2p) mechanism for users to aggregate data locally before presenting the data to the server [8, 13]. Unfortunately, the p2p approach requires $k-1$ other users to be online (i.e., present) in the user's vicinity. Ideally users should achieve k-anonymity even if the other $k-1$ users visit the same area at a later point within a particular time window (in this regard, a server-based approach can combine reports from users who may have visited a particular location at different times). Another problem is that the $k-1$ users may be located beyond the range of p2p communication (e.g., if Bluetooth is used). Mokbel and Chow outline several challenges [28] in this space and suggest a multi-hop approach to reach such nodes [8]. Their approach, however, assumes that nodes trust other nodes with their location information. Ideally, the p2p protocol should require the nodes to anonymously exchange and combine reports while preserving privacy. Secure multi-party computation [14] provides such solutions, but they are computationally expensive and require the k parties to be online simultaneously, neither of which may be acceptable for mobile, pervasive devices. Last, but not least, the k reports may come from within Alice's office (e.g., during a meeting), making it obvious that Alice is at her office, exposing her location privacy. Spatiotemporal cloaking must therefore take into account that some spatial regions can leak information about users even if k-anonymity is obtained.

Local location blurring to improve k-anonymity. Location blurring by users' devices [20, 10] can indeed improve the privacy of users. In this approach, the granularity of the user's reported location is altered *before* sending the report to the system, thereby adding uncertainty to the user's location and hence protecting their identity. The level of blurring, however, may not be sufficient to prevent deanonymization by the system. For example, blurring Alice's location to a region larger than her office might be insuf-

ficient at 2am, when she is the only person usually present in that region at that time. While existing solutions allow users to specify the amount of location blurring [30, 18], we believe this approach is unrealistic for opportunistic sensing, which is a secondary application with no direct benefit to the user. Therefore, *an automatic mechanism is needed to blur the user's location without their intervention*.

AnonySense—anonymous tasking and reporting. We present AnonySense, a general-purpose framework for anonymous tasking that is designed to provide users with privacy from the ground up. Using AnonySense, applications can deliver tasks to anonymous nodes, and eventually collect reports from anonymous nodes. Furthermore, nodes accepting tasks cannot be linked with nodes submitting reports. Since cryptographic anonymity cannot prevent inference-based attacks on privacy, we provide a multi-layered approach to improve the user's privacy. We present a novel solution based on *tessellation*, which partitions the geographical area into *tiles* large enough to preserve the users' privacy. Users report locations at the granularity of these tiles. At a high level, a tile represents a region (centered on a public space) that k users normally visit during a typical interval (e.g., 5 minutes), thereby ensuring that users cannot be identified within a set of k users (with a high probability) if their report is indexed by tile and time interval. Such an approach provides users with what we call "statistical k-anonymity" *before* any report aggregation is performed, and therefore protects users' privacy against the system itself at a lower (first) layer. Our approach, therefore, provides users with location privacy without requiring any user intervention, and is efficient because the amount of blurring is determined locally without the need to communicate with other peers. In this paper, we focus on blurring the location and time of a device's reports, and assume that environmental sensing itself does not leak (too much) information about the reporting user to the system. To further protect the users' privacy against applications, however, reports are aggregated at a higher (second) layer to ensure that several, namely ℓ, reports are combined before sending context information to applications, thereby implementing ℓ-anonymity at the sensed-context level and providing better privacy against applications. AnonySense thus ensures privacy throughout the tasking lifecycle without requiring any user intervention.

Our contributions.

- We present AnonySense, a *general-purpose* framework for anonymous opportunistic tasking, that allows any authorized application to leverage sensors on mobile devices while preserving the privacy of users.
- We develop a new *automatic local blurring* technique where users report locations based on a tessellation large enough to provide statistical k-anonymity against the system. Additional report aggregation is performed to provide ℓ-anonymity against applications.
- We evaluate our tessellation technique based on real mobility traces representing 6,553 active users over 77 days and show that a reasonable tradeoff can be achieved between users' privacy and spatiotemporal granularity.

Paper outline. We present AnonySense's architecture and formalize the notion of privacy in Section 2, then describe our tessellation-based technique in Section 3. We

present our trust assumptions and protocol for anonymous tasking in Section 4, followed by an evaluation of the privacy provided by tessellation in Section 5. We discuss several related issues Section 6, and conclude in Section 7. Focused on anonymity, this paper omits some design and implementation details of our system due to lack of space.

2 Architecture

In what follows, we describe the architecture of our context tasking and reporting system, as well as the threat model and desired security properties.

2.1 System overview

AnonySense is a general-purpose system that allows applications to collect and process (e.g., view, store, monitor and fuse) large volumes of sensed data from urban areas, inspired by the MetroSense [5] vision of opportunistic sensing. Applications can specify tasks using an expressive language, annotated with appropriate spatial and temporal semantics. Applications can use collected data to learn, infer or analyze contextual information related to everyday human-behavior patterns, natural phenomena and sporadic social or environmental events. Following the new sensor-networking paradigm that leverages humans or mobile objects in the sensing infrastructure, AnonySense implements context collection though *opportunistic sensing*, where mobile (mainly) sensing devices voluntarily participate in the system's data-sensing capabilities. Overall, AnonySense offers the following core functionality: applications submit specifications called *tasks* for collecting context through sensing, which are answered by the system through a *privacy-preserving* opportunistic context-collection technique that employs a mobile set of heterogeneous sensing entities.[3]

We identify three parties in our model: the *application*, the *system* and the *users*. An application connects to the system through an interface to request collection of context data from the users' devices. Given a context request, the system employs a *static networking infrastructure* that it owns and controls and a *mobile sensing infrastructure* that the users implement collectively. The static networking infrastructure consists of servers that are responsible for anonymously contacting mobile sensor devices to request sensor data, anonymously collecting sensor data back from the devices, and aggregating the data into context information. The mobile sensing infrastructure consists of individual users carrying sensor devices that dynamically participate in the data collection process. AnonySense's architecture is presented in Figure 1.

The primary goal for AnonySense is to protect users' anonymity with respect to their sensing activity; privacy is desirable not only in its own right, but also as a key factor in encouraging voluntary participation [22]. Indeed, any opportunistic human-centered sensing system should be protecting privacy of its members to support its own existence and ensure proper context collection. AnonySense maintains anonymity at two levels: protecting the user's anonymity against the system (through anonymous

[3] This is a necessity rather an assumption: indeed, participants in an opportunistic context collection system have different sensing profiles as they carry devices with different sensing capabilities and they are not like-minded with respect to participation patterns and privacy concerns.

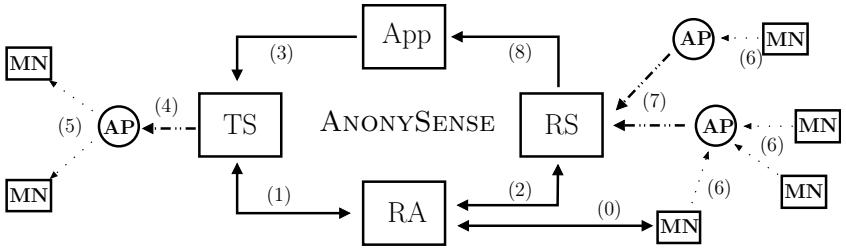

Fig. 1. The AnonySense architecture. A collection of sensor-equipped mobile nodes (MNs), owned by participating *users*, register (0) in the system through the registration authority (RA); RA also certifies the authenticity of the *system*'s components: (1) the task server (TS) and (2) report server (RS). *Applications* (App) submit (3) tasks to the TS; the MNs occasionally download (4,5) new tasks from the TS using the Internet and any available wireless access point (AP). The task specifies when the MN should sense information, and under what conditions to submit reports. MNs report (6) sensed data via APs and their reports eventually arrive (7) at the RS. At its convenience, the App fetches (8) the data from the RS.

protocols and a tessellation-based technique for k-anonymous location blurring) and protecting the user's anonymity against the applications (through ℓ-anonymous report aggregation). Before describing these properties, we define the system's components.

The mobile sensing infrastructure consists of individual *mobile nodes* (MNs), which are devices with sensing, computation, memory, and wireless communication capabilities. These mobile nodes may be *motes* or, we anticipate, cellphone-class devices. Mobile nodes are carried by people, or are attached to moving objects, such as vehicles.[4] The *carrier* of a node is the person carrying the node, or the owner of the vehicle.

Applications specify the desired context as *tasks*, which specify when and what sensor readings to collect, and when to report these readings back to the system. After accepting a task, an MN produces a series of *reports*, each of which is a tuple of the form (taskID, location, time, data...). We defined a simple and expressive Lisp-like language called *AnonyTL* for applications to specify what reports to produce, over what region of space-time, and under what conditions (either periodic or value threshold).

The static networking infrastructure consists of several components, each of which may be distributed or replicated:

- A *Task Server (TS)* that accepts tasks from Apps, and distributes tasks to MNs.
- A *Report Server (RS)* that receives and aggregates sensor reports.
- A *Registration Authority (RA)* that registers participating users and their mobile nodes, installing the AnonyTL interpreter and the cryptographic certificates necessary to later validate the authenticity of the MN to the system, and certificates for the MN to validate the authenticity of the TS and RS.

[4] For simplicity we explicitly assume only mobile nodes. In general, our system can also include static nodes, attached to stationary objects (e.g., buildings), controlled by the system or administered by individual members of the system.

– A set of *access points (APs)* that are wired to the Internet. As described in Section 4, we rely on APs to collect anonymous statistics about user associations for the generation of tessellation maps.

2.2 Threat model and desired properties

Users participate in the system on a voluntary basis by enabling their mobile nodes to respond to tasks distributed by the system and contribute reports. This opportunistic sensing functionality, however, is implemented by nodes reporting sensed data while their carriers are performing everyday activities, and thus tied to many aspects of users' professional, personal and social lives. Consequently, data reports designed for context collection carry explicit or implicit sensitive information about the carriers of the nodes. This information is primarily related to a spatiotemporal component describing a user's activity, since any report annotated with a (time, location) pair clearly reveals the exact time the carrier was at a certain location, and secondarily to a data-related component of the report, i.e., the actual sensed data. On the surface, these reports are anonymous, that is, they are not tagged by the node's identifier. De-identified reports, however, may not provide sufficient anonymity. By associating pairs of time and location with *observed* or *predictable* activity patterns of individuals, sensitive information about carriers can be revealed or inferred. Using recent terminology [24], the *digital footprints left by users implementing opportunistic sensing are easily obtainable* and are a rich source of information for inference about users' location and activity. And even if such inferred information cannot possibly form an undeniable attestation about a user's behavior,[5] it can easily provide statistical inferences about a user's behavior that correspond to reasonably high confidence levels or that can hold with almost certainty.

This threat is directly addressed by AnonySense. The adversary in our anonymity model is an entity that, we must assume, has unlimited access to the users' activities and social, professional or personal every-day life patterns over an unlimited time window extended both in the past and in the future *but not in the present*. That is, the adversary is knowledgeable but not omniscient. In particular, we instantiate this entity to two concrete cases: (1) the system itself, that is, the AnonySense software and hardware and their owners, and (2) the applications and their users. We assume that these two adversaries can observe, collect and learn any information related to the activities of any number of individual users of the system, and the adversary's goal is to deanonymize reports obtained through AnonySense for the time period when a particular user is not under direct observation by the adversary.

In this setting, our goal is to apply privacy-protection techniques in a trade-off between anonymity and context accuracy. We apply the principles of k-anonymity [34] to design mechanisms that allow opportunistic sensing in our model, such that the following two properties are satisfied:

Privacy against the system. To protect users against the system, we provide statistical k-anonymity to any participating user. Thus, even though the system may observe

[5] Indeed, since reports carry no identification labels (at least from a computational point of view through the use of "unbreakable" cryptography), a user can always deny that a certain report came from his or her device (e.g., if brought to court or been accused for a certain behavior).

all reports, it is statistically difficult to link any report to a specific user within a set of k users based on the location or timestamp within the report. Furthermore, nodes also remain (statistically) k-anonymous during the process of receiving tasks and sending reports. Here, k is a known, system parameter controlling the users' anonymity against the system.

Privacy against applications. To protect users against the applications, we provide ℓ-anonymity to any participating user. That is, at least ℓ reports are combined together before the aggregate is revealed to the application. Here, ℓ is an anonymity parameter enforced by the system. We impose no assumption on the relation of k and ℓ. In practice k might be equal to ℓ.

Our techniques ensure that when a node is tasked, neither the TS nor any AP will learn the identity of the node or its carrier. Similarly, when the node later sends a report, neither the RS nor AP will be able to identify the node or its carrier. We also ensure that the system cannot link the tasking event to the reporting events, or link the multiple reports that a single node may submit.

In the next section, we present mechanisms that achieve the above goals.

3 Tessellation

To protect the user's privacy against the system, an MN needs to blur the location in its reports. We aim for a statistical form of k-anonymity, reporting a region rather than a point. The challenge is to divide a geographical area into an appropriate set of regions, such that each region is large enough to provide k-anonymity (usually), but small enough to retain a reasonable level of location accuracy. Our method "tessellates" the area into tiles, according to historical patterns of user locations.

Tessellation map generation. We assume that AnonySense has access to anonymous user-presence data for the geographical area covered by the organization's pervasive environment. In our evaluation we show how such datasets are easily obtained from Wi-Fi AP association counts from within the organization. Given recent historical data, AnonySense generates a tessellation of the area such that each tile represents a region that k users typically visit within a time interval t (e.g., 5 minutes). In reports sent to the system, MNs include the tile ID (obtained from the tessellation map) and time interval ID (time is partitioned into periods of length t). Users are, therefore, able to perform local spatiotemporal cloaking.

We begin the tessellation by constructing a Voronoi diagram at the granularity of the dataset. For example, if the data represents "location" symbolically by AP name, then we first map the AP locations as points on a plane and then construct a Voronoi diagram. For each resulting region (or polygon) we calculate the number of association counts across all time periods, and compute the threshold number of association counts that represent the p-th percentile of all the counts. For example, with $p = 95\%$, a polygon may have threshold value 12, indicating that 95% of the time periods have at least 12 associated users. In effect, we expect high values of p to be better predictors of k-anonymity, as is verified in Section 5. Next, we cluster the resulting polygons into *tiles*

such that the sum of the threshold values for each polygon exceeds k. We use this value of k as the predicted statistical k-anonymity provided by each tile.

In Section 5 we describe our approach as applied to a specific dataset, and study how much a user's actual k-anonymity deviates from the predicted k-anonymity.

4 Protocol

Before we describe our privacy-preserving protocol for anonymous tasking and reporting, we detail the trust assumptions between the various entities in AnonySense.

4.1 Trust assumptions.

Mobile nodes. We assume that all MNs communicate with the TS and RS using Wi-Fi APs. MNs do not trust the APs to maintain their location privacy. For now we assume that APs are owned by a single organization, and may collude with the TS and RS. MNs trust the RA to certify the identities of TS and RS. The MN, therefore, can establish secure connections (e.g., SSL) to the correct TS and RS. Likewise, the RA certifies each MN, which can then prove to the TS and RS that it is a valid node in the system. As we explain below, MNs can prove their validity anonymously.

Applications. Like MNs, applications also trust the RA to certify the TS and RS. Apps must trust the TS and RS to deploy tasks and collect reports as demanded and, additionally, to collect reports only from valid MNs. For now Apps are not certified. In the future we may either require Apps to authenticate, or require the *querier* (user of the App) to authenticate.

Task Server and Report Server. The TS and RS trust the RA to certify valid MNs in the system. The RA is responsible for issuing calibration certificates to MNs, attesting to the fact that the MNs' sensors are properly calibrated, as well as authentication certificates. For simplicity, we assume an authentication scheme such as Direct Anonymous Attestation (DAA) [2], but it may also be possible to use other proposed anonymous authentication schemes [3]. If MNs include trusted hardware such as the Trusted Platform Module (TPM) [36],[6] DAA can also allow the TS to verify the integrity of the MN. For example, the TPM can assert that the MN has not been tampered with, and is running unmodified and authorized software. We leave such TPM-based "remote attestation" to future work.

4.2 Tasking protocol

We first consider the protocol for anonymously assigning tasks to MNs.

[6] TPMs are now installed on most new laptops, and we expect mobile devices to include TPMs in the near future as well [27].

Task Generation. The App generates a task using the tasking language and sends the task to the TS using a server-authenticated SSL channel. This way, the TS accepts tasks from applications and can avoid tampering by third parties. As part of the task, the application specifies an expiration date, after which the task is deleted by the TS and MNs. The TS generates a unique task ID for the task, and sends the application an acknowledgment that contains this task ID along with a TS-signed certificate for the task ID. The application later uses this certificate to access reports for the task.

Tasking language. The AnonyTL tasking language allows an application to specify a task's behavior by providing a set of acceptance conditions, report statements, and termination conditions. The acceptance conditions are evaluated by the MN after retrieving tasks from the TS; for example, these conditions indicate that the MN must have certain sensors. Report statements periodically check a set of report conditions against polled sensor values, and if the conditions are met, report application-specified fields to the RS. These periodic evaluations continue until a termination condition is satisfied, at which point the task is removed from the MN's task pool. We note that tasks are *not executable code*; tasks specify sensor readings desired at a particular granularity, and under what conditions an MN should report data. As a simple example, the following task collects temperature measurements from sensing devices in the Sudikoff building every one minute, and reports temperature values that do not belong to ComfLevel, a predefined range of comfortable temperatures, after being annotated with the corresponding time and location of the reading.

```
(Task 20534)(Expires 1896000453)
(Accept (In location 'sudikoff'))
(Report (temp senseTime location)
        (Every 1 Min) (Not (In temp ComfLevel)))
```

Tasking Nodes. When MNs have Internet access, they periodically poll the TS for tasks over a server-authenticated SSL channel. For each connection, the MN uses DAA to prove to the TS that it is a valid MN in the system, without revealing its identity. The TS delivers all outstanding tasks to the MN. (In future work, MNs may download a random subset of tasks, or the MN may also reveal certain attributes thereby reducing the number of tasks downloaded at the expense of some privacy.)

Since the MNs do not trust APs with their location privacy, an MN contacts the TS only when it is associated with a popular AP, that is, the AP's polygon alone meets the k-anonymity test.

The MN ignores any tasks it has considered in an earlier download, and considers the acceptance conditions of new tasks. During DAA, an MN proves to the TS that it is a valid node, and if TPM-enabled, can prove that it is operating in a secure configuration. The task is deactivated on the TS when it reaches its expiration date.

Reporting. The MN processes the task using the AnonyTL interpreter, reading sensors when required and generating reports as necessary. The MN stores reports in an outgoing queue; when the network is available, and there are queued reports, the MN contacts the RS over a server-authenticated and encrypted channel. As with the tasking protocol, MNs submit reports only when connected to popular (k-anonymous) APs. The MN

uses DAA to prove its validity to the RS without exposing its identity. To prevent the RS from linking multiple reports within or across tasks, the MN must make a separate connection to the RS for each report.[7] Most importantly, the time and location values in the report are specified using the granularity from the tessellation map. Location is reported at the granularity of tiles, and time is reported at the granularity of time periods used for generating those tiles. As a result, users are given statistical k-anonymity based on the historical implication that the number of users visiting this tile in this interval is likely to be greater than k.

Data Fusion. The RS aggregates reports from a task before delivering the aggregated results to the application. Reports are combined using standard k-anonymity techniques with parameter ℓ, according to which individual fields of the reports are either generalized (i.e., values become less specific) or suppressed (i.e., values are not released). This ℓ-anonymity provides the second layer of privacy protection to the mobile user. The specific aggregation method depends on many factors such as the type of data sensed (such as a picture, an audio file, or temperature reading) and the needs of the App. A detailed discussion of aggregation methods is beyond the scope of this paper.

Report Collection. The App polls the RS for available context using a server-authenticated and encrypted channel. The application presents the TS-issued certificate with the task ID, proving that it is authorized to access the reports for that task. Encryption prevents eavesdroppers from learning potentially sensitive context data.

MAC Address Recycling. Using DAA for anonymous authentication is useless if an MN can be tracked using its static MAC address, because MNs assume the APs may collude with other components of the system. We assume the MN changes its MAC and IP addresses using one of the standard mechanisms [16, 21] so that an MN's report and task actions may not be linked, but leave its implementation to future work. Addresses should be recycled for *each* report for maximum privacy, thereby ensuring reports are unlinkable. Recently, Pang et al. have shown how users' privacy can be reduced through 802.11 fingerprinting [31], so it may be necessary to seek other methods beyond MAC-address rotation to maintain privacy against especially snoopy APs.

5 Evaluation

Due to the ready availability of wireless traces from CRAWDAD.org, we did not perform a live simulation of AnonySense on real devices. Instead, we used historical movement traces to run simulations for different system parameters.

We generated a tessellation of the Dartmouth College campus based on the data set publicly available from CRAWDAD [25]. This data set represents the locations of (anonymized) wireless-network users: each entry represents a user's device associating, reassociating, or disassociating from an AP on campus. Before constructing the Voronoi diagram, as described in Section 3, we flattened the AP locations to two dimensions by ignoring the "floor number" provided for each AP. (We leave three-dimensional

[7] For performance reasons we plan to explore batch reporting, at some trade-off to privacy.

tessellation for future work.) This planarization step causes some locations to have tight clusters of APs that result when a tall building has an AP at the same location on every floor; thus, we group APs that are within a certain Euclidean distance of each other into a single AP. Given this set of points on the plane, we generated a Voronoi diagram to produce a polygon for each AP, and applied our clustering algorithm to generate a tessellation of the geographical region. We generated tessellation maps for 6-hour time periods in the day (due to varying mobility patterns throughout the day), and focused only on the time period 12noon–6pm in our experiments.

Figure 2 shows a tessellation for $k = 10$, with a time granularity of 10 minutes. The points within the tiles indicate the locations of APs that were clustered together for that tile. There are 90 tiles in Figure 2, the smallest and largest being 82 m^2 and 2,694,063 m^2, with a median area of 1629 m^2. We note that the tiles near the edges of the Dartmouth campus tend to have a large area because we do not crop the tiles to the general campus area.

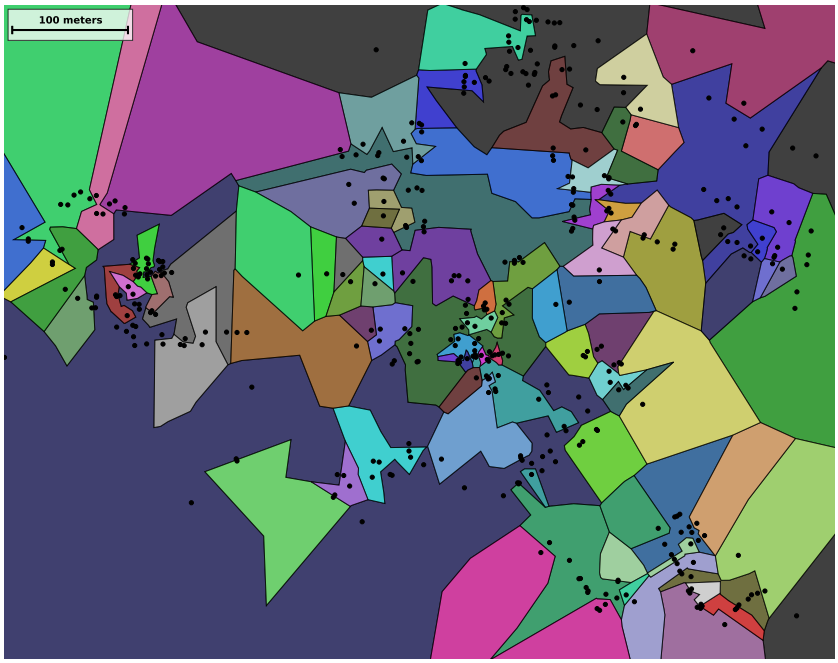

Fig. 2. We generated a Voronoi diagram using the set of APs (represented as black dots). We then combined neighboring polygons to form tiles, the colored tiles in the figure, such that $k \geq 10$ for each tile between the hours of 12pm–6pm from 9/24/2003–10/31/2003. In general, the area of a tile is roughly inversely proportional to the number of AP associations in that area.

Although the tessellation map was generated using historic AP visitation data, we expect a user's k-anonymity to be similar to the historically observed values. To eval-

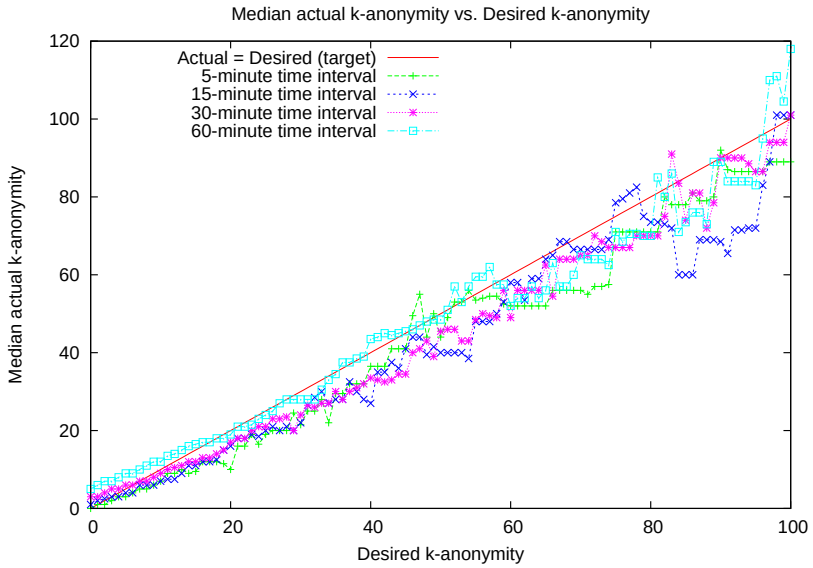

Fig. 3. This graph plots the median of the actual k-anonymity attained across all tiles against the desired k-anonymity. We can see that the expected minimum k-anonymity closely matches the desired k-anonymity regardless of the time interval used.

uate this claim, we measured the deviation of a user's anonymity from the expected k-anonymity using the same 2003 Dartmouth dataset. We generated the tessellation map for data from the 24 September–31 October range, first half of their Fall term, and then evaluated AnonySense with the visits from the 1 November–10 December range. This was done by calculating the effective anonymity for each tile using the same counting heuristic used in its generation, at the 95th percentile (i.e., 95% of the time, what value was the tile's effective k-anonymity greater than?) Figure 3 illustrates the performance of our technique for various time blurring intervals. The objective is to achieve a simulated k-anonymity equal to the desired k-anonymity parameter (95% of the time) used to generate the tiling, and the figure demonstrates that this level of k-anonymity is closely matched with the expected value. It also shows that the accuracy of our tiling method is mostly independent of the time interval used to generate the tiling. This is important, as it means that it is possible to select a time interval without sacrificing anonymity (although, of course, spatial accuracy will be sacrificed, as we show in Figure 4.)

We then evaluated the trade-off between temporal and spatial accuracy, for different levels of k-anonymity. Figure 4 shows this trade-off, with a roughly inverse relationship between the two. This result, along with the fact that the quality (with respect to k-anonymity) of our tessellations is independent of the time interval chosen, means that

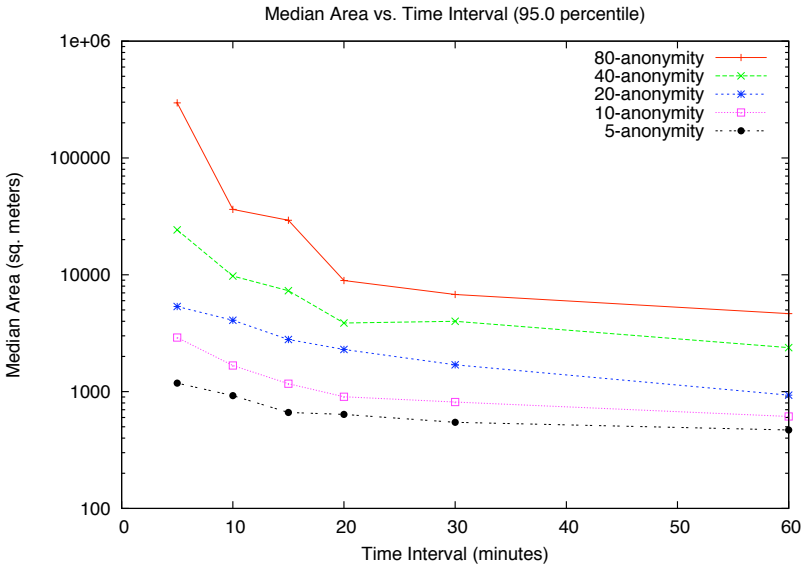

Fig. 4. This graph plots the median spatial granularity against the chosen time granularity for various values of desired k-anonymity. We see that for a given k, time granularity can be traded off for spatial granularity. Increasing the amount of k anonymity requires larger tiles, and thus reduces the spatial granularity. Note the log scale.

statistical k-anonymity can be maintained regardless of an application's desired spatial accuracy, at the cost of temporal accuracy.

Finally, we evaluated the distribution of actual k-anonymity for specific instances of our tessellation. Figure 5 shows one such distribution for $k = 10$ and $t = 10$. The histogram shows that our algorithm not only gives anonymity to a majority of users, but actually gives most users significantly greater anonymity than the tessellation parameters require.

We note that the clustering algorithms used to generate the tessellation are not necessarily ideal. As can be seen in Figure 2, some of the tiles have elongated and irregular shapes, which would not necessarily translate to "useful" locality readings from a human point of view. Although the shapes of the tiles do not affect AnonySense's privacy properties (assuming the basic tile generation rules are met), more sophisticated clustering algorithms that favor compact tiles might give better results, while still maintaining similar values for k.

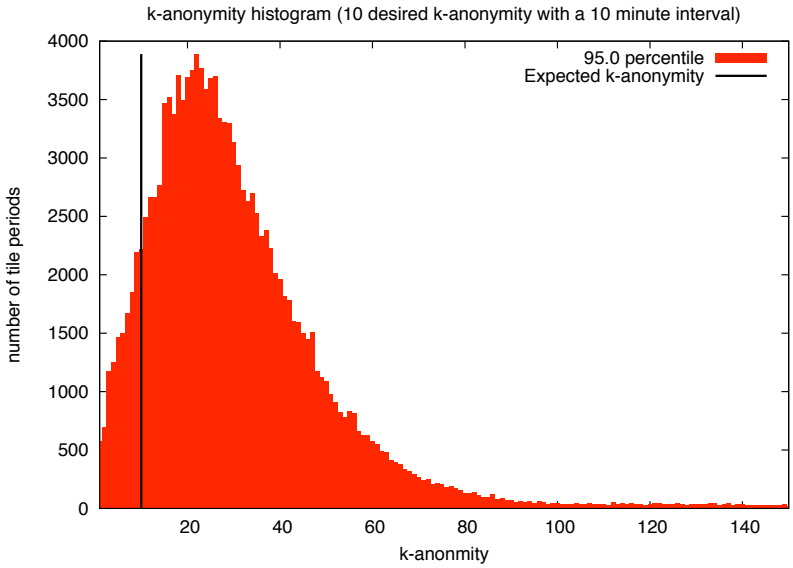

Fig. 5. This histogram shows the distribution of actual k-anonymity achieved during each tile-time-period combination for desired $k \geq 10$ and $t = 10$ minutes.

6 Discussion

There are many issues raised by this approach, and several challenges remain.

Real-time information. Due to the opportunistic nature of its sensing model, AnonySense is inherently not a real-time context collection system. Not only is there an inevitable system delay arising from intermittent MN connectivity and task acceptance conditions, but AnonySense intentionally maintains the time-location trade-off for a given level of k-anonymity. Thus, an application needing timely information would likely sacrifice significant location accuracy at all but the busiest of sensing locations.

Varying the degree of k-anonymity. One natural suggestion would be to allow users to specify their desired degree of k-anonymity, and pick the right tile to match the specified k. We caution against this approach, because the user's preferred degree of anonymity can itself leak information about the user. Consider the example where a known paranoid user prefers k-anonymity with $k = 200$. Receiving a report with a large blurred range can be linked to that user. Users' privacy is therefore maximized when all users use the same value of k.

Privacy. AnonySense takes a conservative approach at providing users with privacy, by blurring all dimensions of a report. Specifically, ℓ reports are combined so that time, location, and sensor readings are aggregated according to standard techniques used in the literature. We plan to explore other forms of privacy, where the fidelity of some dimensions of reports are preserved. For example, location blurring may be sufficient to provide users with privacy even if their identities are known. Indeed, much of the work on location privacy related to sharing location with contacts assumes that the identity of the person is known, and the location is blurred enough to provide users with privacy. AnonySense, however, aims to provide stronger privacy. For example, Alice may select a campus-level location granularity, but still leaks the exact time she arrives on campus.

Security metrics. Quantifying the privacy of users is a hard problem. We assume a strong attack model, where adversaries can build histories of users, and then try to deanonymize reports obtained from the tasking service. We further assume that the system itself may misuse the information about the time and location of sensor reports, by blurring the location and time, but we trust the system to properly anonymize (aggregate) the sensor data. Blurring the sensor data *at the nodes* may be possible, using a similar tessellation map built from historical sensor data. We do not evaluate such an approach because of lack of data. Furthermore, we believe that the blurring of environmental sensor data (such as room temperature or pollution levels) will provide diminishing returns to a user's privacy in addition to what is already provided by time and location blurring.

Collecting user-presence data. For simplicity we assume that users trust an entity called the *Map Server (MS)* to collect AP-association traces and generate the tessellation map accurately. There are several avenues for future work for MNs to maintain anonymous associations with the APs, but allow the APs to accurately count MNs, and allow the MNs to verify the accuracy of the resulting map. We hint at two such approaches based on standard cryptographic techniques. (1) Users authenticate with APs using a variant of anonymous authentication that allows only one authentication per time period [4]. Users obtain a token from the APs certifying that some anonymous user successfully associated with that AP. The user posts this token onto a public bulletin board. These AP-association "logs" can be then used for generating tessellation maps. Using the publicly posted information, users can verify that their associations have been used in generating the tessellation map. Note that the bulletin board is *not* trusted storage. If the bulletin board cheats by throwing away some data, it can be observed by users with high probability. Such techniques are common to e-voting protocols [7]. (2) Alternatively, users can maintain a vector of location visits during a day (e.g., using GPS traces or by counting their AP associations) and upload their vectors to the bulletin board at the end of the day. Such techniques are standard to the "homomorphic encryption" family of e-voting protocols [9].

Location data, and tessellation. We describe our tessellation approach using data about the location history of wireless-network users, which is provided as a sequence of associations with Wi-Fi APs. The movements are discrete, hopping from one location to another, and the locations are discrete points on the plane. In other settings, such as

locations obtained from GPS, the locations are continuous (any coordinate on the earth) and the movements may be less discrete (a path connecting waypoints). We believe that our tessellation approach can be adapted to other location models, such as heat maps showing continuous mobility distribution, although the details remain future work.

7 Conclusions

We present AnonySense, a comprehensive system aimed at preserving the privacy of users in opportunistic-sensing environments. AnonySense uses a protocol for anonymous tasking and reporting, and performs report aggregation to provide k-anonymity against applications. We show how users can proactively improve their privacy against the system by using a novel technique based on *tessellation*. Using our technique, users' devices can automatically blur time and location information in reports to provide users with statistical k-anonymity. Automatic blurring is achieved by reporting locations at the granularity of *tiles*, where the tiles are generated based on historical data. We evaluated our approach using real AP-association traces from the CRAWDAD dataset representing 6,553 active users over 77 days and show that a reasonable tradeoff can be achieved between users' privacy and spatiotemporal granularity.

Acknowledgments

This research program is a part of the Institute for Security Technology Studies, supported by Grants 2005-DD-BX-1091 awarded by the Bureau of Justice Assistance, 60NANB6D6130 awarded by the U.S. Department of Commerce, and by the Institute for Information Infrastructure Protection (I3P) under an award from the Science and Technology Directorate at the U.S. Department of Homeland Security. The second author was additionally supported by the Center for Algorithmic Game Theory at the University of Aarhus under an award from the Carlsberg Foundation. Computations were performed on cluster machines supported under NSF grant EIA-98-02068. The views or opinions in this paper do not necessarily reflect the views of the sponsors. We thank the MetroSense team at Dartmouth College and Vijay Bhuse for their helpful comments.

References

1. T. Abdelzaher, Y. Anokwa, P. Boda, J. Burke, D. Estrin, L. Guibas, A. Kansal, S. Madden, and J. Reich. Mobiscopes for human spaces. *IEEE Pervasive Computing*, 6(2):20–29, 2007.
2. E. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. In *CCS '04: Proceedings of the 11th ACM Conference on Computer and Communications Security*, pages 132–145. ACM Press, 2004.
3. G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy. Efficient and robust pseudonymous authentication in VANET. In *VANET '07: Proceedings of the Fourth ACM International Workshop on Vehicular Ad Hoc Networks*, pages 19–28. ACM Press, 2007.

4. J. Camenisch, S. Hohenberger, M. Kohlweiss, A. Lysyanskaya, and M. Meyerovich. How to win the clonewars: efficient periodic n-times anonymous authentication. In *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS 2006)*, pages 201–210. ACM Press, 2006.
5. A. Campbell, S. Eisenman, N. Lane, E. Miluzzo, and R. Peterson. People-centric urban sensing. In *The Second Annual International Wireless Internet Conference (WICON)*, pages 2–5. IEEE Computer Society Press, August 2006.
6. CENS Urban Sensing project, 2007. http://research.cens.ucla.edu/projects/2006/Systems/Urban_Sensing/.
7. D. Chaum, P. Y. A. Ryan, and S. A. Schneider. A practical voter-verifiable election scheme. In *ESORICS*, pages 118–139, 2005.
8. C.-Y. Chow, M. F. Mokbel, and X. Liu. A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In *GIS '06: Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems*, pages 171–178. ACM Press, 2006.
9. R. Cramer, R. Gennaro, and B. Schoenmakers. A secure and optimally efficient multi-authority election scheme. *Advances in Cryptology—EUROCRYPT '97: International Conference on the Theory and Application of Cryptographic Techniques, Lecture Notes in Computer Science*, 1233:103–118, 1997.
10. M. Duckham and L. Kulik. A Formal Model of Obfuscation and Negotiation for Location Privacy. *Pervasive Computing: Third International Conference (Pervasive 2005), Munich, Germany*, May 2005.
11. C. Frank, P. Bolliger, C. Roduner, and W. Kellerer. Objects calling home: Locating objects using mobile phones. In *Proceedings of the 5th International Conference on Pervasive Computing (Pervasive 2007)*, May 2007.
12. B. Gedik and L. Liu. Location privacy in mobile systems: A personalized anonymization model. In *ICDCS '05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, pages 620–629. IEEE Computer Society, 2005.
13. G. Ghinita, P. Kalnis, and S. Skiadopoulos. Prive: anonymous location-based queries in distributed mobile systems. In *WWW '07: Proceedings of the 16th International Conference on World Wide Web*, pages 371–380. ACM Press, 2007.
14. O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *ACM Symposium on Theory of Computing*, pages 218–229, 1987.
15. M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *MobiSys '03: Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, pages 31–42. ACM Press, 2003.
16. M. Gruteser and D. Grunwald. Enhancing location privacy in wireless LAN through disposable interface identifiers: a quantitative analysis. *Mobile Networks and Applications*, 10(3):315–325, 2005.
17. B. Hoh and M. Gruteser. Protecting location privacy through path confusion. In *SECURECOMM '05: Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, pages 194–205. IEEE Computer Society, 2005.
18. J. I. Hong and J. A. Landay. An architecture for privacy-sensitive ubiquitous computing. In *Proceedings of MobiSys 2004*, pages 177–189, Boston, MA, USA, June 2004.
19. B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. K. Miu, E. Shih, H. Balakrishnan, and S. Madden. CarTel: A Distributed Mobile Sensor Computing System. In *4th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, November 2006.

20. G. Iachello, I. Smith, S. Consolvo, M. Chen, and G. D. Abowd. Developing privacy guidelines for social location disclosure applications and services. In *Proceedings of the 2005 Symposium on Usable Privacy and Security*, July 2005.
21. T. Jiang, H. J. Wang, and Y.-C. Hu. Preserving location privacy in wireless LANs. In *MobiSys '07: Proceedings of the 5th International Conference on Mobile Systems, Applications and Services*, pages 246–257. ACM Press, 2007.
22. P. Johnson, A. Kapadia, D. Kotz, and N. Triandopoulos. People-Centric Urban Sensing: Security Challenges for the New Paradigm. Technical Report TR2007-586, Dartmouth College, Computer Science, Hanover, NH, February 2007.
23. P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preserving anonymity in location based services. Technical Report TRB/06, National University of Singapore, Department of Computer Science, 2006.
24. A. Kapadia, T. Henderson, J. J. Fielding, and D. Kotz. Virtual walls: Protecting digital privacy in pervasive environments. In *Proceedings of the Fifth International Conference on Pervasive Computing (Pervasive)*, volume 4480 of *LNCS*, pages 162–179. Springer-Verlag, May 2007.
25. D. Kotz, T. Henderson, and I. Abyzov. CRAWDAD trace dartmouth/campus/movement/aplocations (v. 2004-11-09). Downloaded from http://crawdad.cs.dartmouth.edu/dartmouth/campus/movement/aplocations, Nov. 2004.
26. J. Krumm. Inference attacks on location tracks. In *Proceedings of the Fifth International Conference on Pervasive Computing (Pervasive)*, volume 4480 of *LNCS*, pages 127–143. Springer-Verlag, May 2007.
27. Mobile Phone Work Group, Trusted Computing Group. https://www.trustedcomputinggroup.org/groups/mobile.
28. M. F. Mokbel and C.-Y. Chow. Challenges in preserving location privacy in peer-to-peer environments. *Seventh International Conference on Web-Age Information Management Workshops*, page 1, 2006.
29. M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The new Casper: query processing for location services without compromising privacy. In *VLDB '06: Proceedings of the 32nd International Conference on Very Large Data Bases*, pages 763–774. VLDB Endowment, 2006.
30. G. Myles, A. Friday, and N. Davies. Preserving privacy in environments with location-based applications. *IEEE Pervasive Computing*, 2(1):56–64, Jan.-Mar. 2003.
31. J. Pang, B. Greenstein, R. Gummadi, S. Seshan, and D. Wetherall. 802.11 user fingerprinting. In *MobiCom '07: Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*, pages 99–110. ACM Press, 2007.
32. O. Riva and C. Borcea. The Urbanet revolution: Sensor power to the people! *IEEE Pervasive Computing*, 6(2):41–49, 2007.
33. Microsoft Research SenseWeb project, 2007. http://research.microsoft.com/nec/senseweb/.
34. L. Sweeney. *k*-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems*, 2002.
35. K. P. Tang, J. Fogarty, P. Keyani, and J. I. Hong. Putting people in their place: An anonymous and privacy-sensitive approach to collecting sensed data in location-based applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 93–102, 2006.
36. Trusted Computing Group (TCG), May 2005. https://www.trustedcomputinggroup.org/home.