Dynamic Tessellation of Geographical Regions to Ensure K-anonymity

Hamilton Turner, Danny Guymon, Brian Dougherty, Jules White
Dept. of Electrical and Computer Engineering
Virginia Tech
Email:{hamiltont, dguymon, brianpd, julesw}@vt.edu

***Abstract*—Smartphone-powered data collection systems are rapidly becoming an effective method of gathering field data. One major challenge of using smartphones to collect data is the ability to link smartphone metadata, such as location at a specific time, back to the user–thereby violating the privacy of that individual. A promising approach to helping ensure user privacy is through geographical k-anonymity, which attempts to ensure that every gathered data reading is geographically indistinguishable from k other readings. The approach helps prevent precise localization of the user or reverse engineering of reported data by leveraging the user's known location. This paper presents a dynamic tesselation algorithm for k-anonymity that provides better privacy preservation and data reporting precision that previous static algorithms for k-anonymity. The paper presents empirical results from a real world data set that demonstrate the improvements in privacy provided by the algorithm.**

***Index Terms*—smartphone, privacy, data collection**

I. Introduction

Emerging Trends and Challenges. The number of deployed smartphone devices has grown rapidly in recent years to over 81 millions units in Q3 2010 [1], [2]. Significant interest in using smartphones as distributed data collection systems has emerged because smartphones are widely deployed and have a number of computing capabilities that make them desirable for data collection systems, including: a plethora of sensors, plentiful processing and storage, a regular connection to the Internet, rapid application development through high-level programming languages, and frequent recharging by end-users. During the recent tragedy of the Gulf Oil Spill [3], multiple smartphone developers created applications that allowed citizens to help collect and report field data, such as images and text of oil-impacted animals or environments, to researchers and scientists. The success of these 'citizen scientist' applications has raised interest in smartphone-powered data collection, in which data measurements are collected by a number of smartphones dispersed over an environment of interest.

Smartphone data collection systems are currently used for a variety of data collection applications, such as health monitoring [4], CO_2 emission tracking [5], traffic accident detection [6], [7], traffic flow measurement [8], and cardiac patient monitoring [9]. Additionally, multiple middleware layers are being created to enable rapid creation and deployment of smartphone data collection applications [10]–[13]. Due to the increasing use of smartphones for data collection, multiple research efforts are attempting to quantify various aspects of using human-carried smartphones as sensors [10], [13], [14].

Open Problem $\Rightarrow$ Location Data from Smartphone-powered Data Collection Systems can be Used to Invade the Personal Privacy of Users. One major challenge of using smartphones for data collection is the ability to leverage a user's known location to determine what data was submitted by a user. For example, in a remote health monitoring system, if each health report includes the location of the user, then an attacker could follow the user and utilize the user's location to determine which health report was his or hers. Conversely, if specific information about the user is known, such as hair color, eye color, and weight, the attacker could potentially use this information to filter the data reports and determine the user's exact latitude and longitude.

One promising approach of protecting user's location data and helping to prevent location data from being used to find private user data is geographical k-anonymity [12]. Geographical k-anonymity involves making an informed guess regarding the temporal and spatial distribution of incoming data, and using this assumption to logically break a geographical area into a number of regions [12] that each contain a minimal number of data readings, as shown in Figure 1. By sharing this tessellation map with end-user smartphone devices, they can report regional id's instead of specific latitude/longitude locations. If the real-world incoming data distribution matches the assumption used to generate the regional tiles, then the incoming data will have the property of being k-anonymous, where at least k data readings are indistinguishable from one another [12], [15]. This ambiguity helps to protect an attacker from determining a user's exact latitude/longitude or discovering their private data using their latitude/longitude.

A good k-anonymity tessellation map is critical for preserving user privacy in a smartphone-powered data collection system. A key challenge with current tessellation approaches for k-anonymity is that they use a static, one-time tessellation based on predictions about the data that will enter the system. If the expected prediction used to generate the tessellation differs too much from the actual incoming data, the algorithms experience quality of service failures where either privacy is not preserved, or data precision is reduced needlessly. Even with an accurate prediction, unpredictable events such as disasters can radically alter the distribution of incoming data,

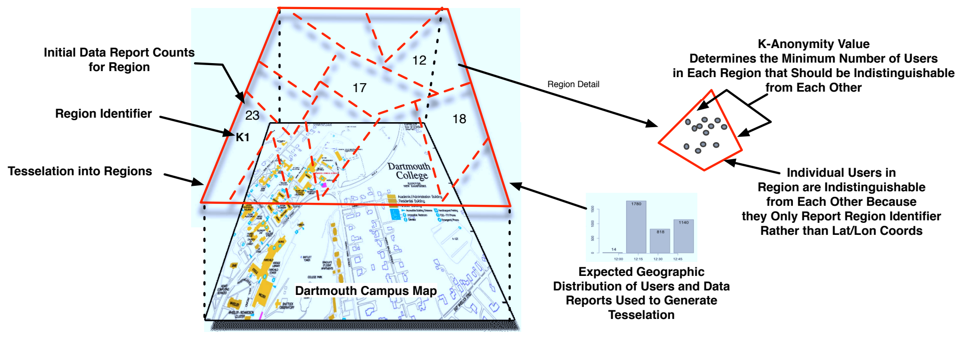
Initial Data Report Counts for Region
Region Identifier
Tesselation into Regions
23
17
12
18
K1
Dartmouth College
Dartmouth Campus Map
Region Detail
K-Anonymity Value Determines the Minimum Number of Users in Each Region that Should be Indistinguishable from Each Other
Individual Users in Region are Indistinguishable from Each Other Because they Only Report Region Identifier Rather than Lat/Lon Coords
Expected Geographic Distribution of Users and Data Reports Used to Generate Tesselation

Fig. 1. Tessellation of the Dartmouth Campus

causing a previously derived tessellation to perform poorly. Moreover, common situations, such as weekend vs weekdays or peak vs off-peak hours, can have vastly different spatial and temporal distributions of incoming data, which will result in poor performance.

Solution Approach ⇒ Constructing Location Regions Dynamically. To ensure user privacy in smartphone data collection systems, we present a dynamic tessellation algorithm, called Anonoly e.g. *ANON*ymous p*OLY*gons, that addressed the potential failures present in current static tessellation approaches. Anonoly uses samples of incoming real-world data report location and times to update the algorithm's prediction of future incoming data spatial and temporal distribution. As this distribution assumption is updated, a new tessellation map that more closely models the real world is generated. Anonoly's approach to retessellation is that regions which received too few data readings to safely enforce privacy increase in size by merging with nearby regions, and regions which received more than the minimum number of readings needed to ensure privacy split into multiple sub-regions.

This dynamic tessellation approach can adapt to changes in the real-world incoming data distribution, and therefore ensure that both privacy and data precision are consistently balanced. This paper provides the following contributions to the study of privacy within a smartphone-powered data collection system:

- We describe a new algorithm, called Anonoly, for dynamically tessellating a geographical environment to acheive k-anonymity
- We provide empirical results that illustrate the limitations of static tessellation approaches for k-anonymity
- We present empirical results showing that Anonoly provides better privacy preservation and data precision than static tesselation approaches
- We demonstrate that Anonoly can operate under a wide range of scenarios by running experiments which show the algorithm maintaining a desired privacy level for a two month dataset

The remainder of this paper is organized as follows: Section II provides a motivating example to be used in outlining the challenges of preserving locational privacy; Section III discusses the specific challenges faced when attempting to generate k-anonymous location regions; Section IV formally introduces our algorithm for dynamic tessellation; Section V presents empirical results from analyzing the proposed algorithm; and Section VII presents concluding remarks and lessons learned.

II. A Smartphone-Powered Data Collection System

In order to motivate the challenges associated with maintaining user privacy within a smartphone-powered data collection system, we present an example smartphone data collection system intended to collect field data about the status of the cellular network from end-user smartphones and dynamically generate a signal strength map. This type of data collection system could be used to monitor the status of the cellular network, build cellular network coverage maps, ensure that cellular signal propagation models are corroborated by real-world data, or ensure that all cellular chipsets are receiving similar signal strength. In this data collection system, user-generated data reports include information about cellular signal strength, the hardware/software configuration of the smartphone, the location at which the reading was captured, and the timestamp when the reading was taken.

Assume an attacker is interested in tracking the user's location and knows the type of phone that the user is carrying. If the attacker can leverage the reported cell signal characteristics to identify the user's phone, then it will be possible to filter the data reports and identify reports coming from the user. If location is reported as a precise latitude/longitude, then the attacker will be able to know the user's exact location.

K-anonymity in a smartphone data collection system gives a high level of confidence that there will consistently be at least k data reports that are indistinguishable. This ambiguity makes it difficult for an attacker to re-associate a user's latitude and longitude with a data report for a given user. The value of k can be adjusted based on the the required data precision for the application, sensitivity of the data, and density of users to provide a specific level of anonymity.

The anonymity is provided by overlaying a tessellation map on top of a geographical region (see figure 1), and allowing smartphones to report a region identifier instead of a precise latitude/longitude. The region tessellation is generated using an assumption of when and where data readings will enter the system, and if the assumption is close to the real-world data distribution then each region will receive at least k data readings. The goal is to prevent an outsider from determining what data a user reported by knowing the location of the user or determining the user's precise location with information specific to that user. If the attacker knows the user's location, there will be at least k data reports with the same region identifier as the user. Moreover, if the attacker can find a specific data report for a user, they can only map their location to a specific region, the size of which is a function of the value of k.

III. Challenges of k-anonymous tessellation

Current approaches to smartphone data collection frequently report the sensed data, time, and a precise location where the data was collected. Most methods of anonymizing or increasing the privacy of data involve reducing the fidelity of that data, which consequently tends to reduce the usefulness of the data. For example, tessellating a region into polygons of size $10m^2$ provides more accurate data about measurements associated with the data report but provides less privacy than tessellating into regions of $100m^2$. This section presents the challenges associated with attempting to enforce k-anonymity across a geographic region.

1) Challenge 1: Unknown Data Reading Distribution Makes Region Tessellation Difficult: A key challenge of tessellating a geographic region to produce k-anonymity is that it requires knowledge of the temporal and spatial distribution of the future readings from the area. When the assumed data distribution is incorrect, static tessellation can have two types of quality of service failures. First, if the assumption over-estimates the number of incoming data readings, and generates a tessellation map based on that assumption, then the regions will not receive the minimum number of readings required to ensure user locational privacy. Second, if the assumption under-estimates the number of incoming data readings, then it will generate very large regions in order to ensure the minimum required number of readings per region is met. This will lead to each region receiving far more data readings than minimally required, but the locational accuracy of the incoming data will by much worse than it could have been if the assumption was correct and the regions were smaller. We term this 'privacy-induced imprecision,' whereby the incoming data is reduced in fidelity in order to protect privacy, but the reduction is over-aggressive and data fidelity (e.g. locational accuracy) is needlessly sacrificed. Static tessellation algorithms can experience either of the failures if the original assumption about the incoming data distribution was incorrect

In a smartphone data collection system designed to collect information about cellular signal strength, the challenge of interest may be generating cellular network coverage maps for remote regions e.g. on hiking trails, above lakes, etc. In these types of situations, there may be little to zero initial information about the number of smartphone users that carry their devices into these areas. Additionally, user smartphone usage habits will likely change in these areas versus more rural locations, and therefore there is little to no information on how many data readings will be captured. These unknowns make it difficult to generate an initial assumption about the incoming data spatial and temporal distribution.

2) Challenge 2: Volatile Data Reading Distribution makes Balancing Privacy and Data Precision Hard: As discussed in Section III-1, a correct assumption regarding the spatial and temporal distribution of incoming data readings can be used to generate regions that enforce k-anonymity. However, there is no guarantee that the time and location distribution of incoming data readings will remain static. The amount of incoming data per region can fluctuate incredibly rapidly, making it very difficult for the assumption about incoming data to correctly match the real incoming data distribution. This volatility makes it very difficult to generate a distribution assumption that can constantly match the real-world distribution. In situations where there are fewer data readings entering the system than expected (a.k.a the assumption over-estimated), the data reports will not be grouped with at least k other reports, thereby risking the privacy of users. Conversely, if the assumption under-estimates the number of incoming data readings, then the tessellation map will have overly large regions (to ensure that each region receives at least k data reports). In this situation, however, the regions could be smaller without causing any privacy violations, and therefore the locational imprecision is unnecessary. Balancing the orthogonal desires of privacy and data precision is a challenging topic.

In a data collection system to collect information on a cellular network, for example, an assumption might be made that the geographical area of interest will receive at least 200 data readings every hour, and that assumption used to generate a tessellation map. However, if there are only 50 data readings being received in one hour, then there will be multiple region identifiers that are used fewer than the desired k number of times, and the data reports entered into those regions will not be as private as desired. If the system instead receives 10,000 data readings in one hour, then it is likely that each region identifier will be used far more than the minimum k number of times. In this case, it would be possible to use smaller regions, thereby adding better locational granularity to the dataset, without violating the desired user privacy margin.

IV. Anonoly - The Anonymous Polygon Region Tessellation Algorithm

This section describes the *ANON*ymous p*OLY*gons, or Anonoly, algorithm in detail. It starts with a high-level outline of the Anonoly algorithm, then introduces a formal model of the algorithm, proceeds to discuss the execution of the algorithm, and lastly concludes with discussion of the major tradeoffs and parameters of Anonoly.

A. Overview of Anonoly

Anonoly is an algorithm for dynamically tessellating a geographic region in order to maintain k-anonymity, where k-anonymity is a property that has been shown to statistically protect privacy by making it difficult to associate specific individuals with specific data items [15]. By ensuring k-anonymity, Anonoly provides a guaranteed level of privacy (e.g. k data reports are indistinguishable) which has a number of practical benefits, including removing a potential deterrent for smartphone users interested in participating in data collection, reducing the severity of system data leaks, and increasing the potential for commercial datasets to be released for research purposes.

Figure 2 shows how the anonoloy algorithm generates and dynamically modifies a tessellation map of a geographic region in order to protect user privacy. A tessellation map is a set of non-overlapping polygons with points and edges defined via latitude longitude locations, where the union of the polygons spans an entire geographical region of interest. The goal of producing the tessellation map, sharing that map with end-user smartphones, and allowing the smartphones to enter regional identifiers, is to ensure that user privacy is statically protected by the k-anonymity property of the polygons.

Walking counter-clockwise around Figure 2, the following steps are taken:

1) The first tessellation map is shared with end-user smartphones. This can either be generated using some *a priori* information about the spatial and temporal distribution of incoming data readings, or can simply be a tessellation containing a single polygon covering the entire region
2) Smartphones input data reports, using regional id's instead of precise latitude longitude locations
3) The number of reports that used each regional id is summed, and used to determine which regions are causing violations.
4) The reporting information is used to improve upon the current tessellation map, by increasing the size of regions that received too few data reports and decreasing the size of regions that received too many data reports. This improved tessellation map is shared with end- user smartphones
5) Smartphones input data reports, using regional id's instead of precise latitude longitude locations

If the distribution of incoming data is non-volatile, then the tessellation map will converge upon a map that causes no privacy or imprecision violations. If the distribution of incoming data changes frequently, then the tessellation map will change rapidly to ensure that the privacy of users is maintained without significant loss in data precision.

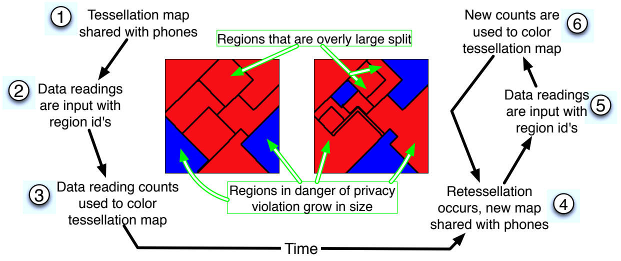
1 Tessellation map shared with phones
Regions that are overly large split
New counts are used to color tessellation map 6
2 Data readings are input with region id's
Data readings are input with region id's 5
3 Data reading counts used to color tessellation map
Regions in danger of privacy violation grow in size
Retessellation occurs, new map shared with phones 4
Time

Fig. 2. Evolution Anonoly's Generated Tessellation Map

B. *Formal Model of Anonoly*

A formal model of Anonoly is defined for ease of discussion. The formal model can be described as a 9-tuple:

$$DAR = < R_s, T_c, K, C_r, R_{nk}, A, U, M, S_{fty}, S_{plit}, S >$$

where:

- R_s is a set containing all of the location regions that are currently defined. Each item in R_s is a closed polygon that defines the outline of one location region
- T_c is the timeslice size, or time between each regeneration of the tessellation
- K is the desired k-anonymity value. For example, a K of 10 would imply that each region would receive 10 data readings per T_c. If a region receives fewer than 10 readings in T_c, then that region currently invalidates the k-anonymity of the system, and should be increased in size. If a region receives more than K readings, then that region can potentially be reduced in size without invalidating k-anonymity, which would improve the locational accuracy of the incoming data
- C_r is a set of counts that specify the number of data readings that have been input for each region since the last recalculation. $C_{r,i}$ indicates the number of data readings input from the ith region in R_s. After each recalculation, all of the counts in C_r are reset to zero, and C_r is resized to ensure that $|R_s| = |C_r|$. Individual data reading counts from C_r are referred to as C
- R_{nk} is a function that accepts two regions from R_s and returns an ordering for the two regions that defines which region is farther from optimal. This allows system administrators to implement any definition of 'optimal' they desire in their system. Multiple possibilities for R_{nk} are discussed in IV-D
- A is the total area for the environment of interest, in $distanceunits^2$
- U is a function that determines how much a region should grow in response to not meeting the desired K value. It accepts a region R, and the associated data reading count C for that region, and returns the amount that the region should grow (in area squared) in response to a non-optimal C value
- M is a function that can resize two regions by removing space from one region and allocating it to another. M is given a region that needs to be resized (the consumer), a value for the amount of change in area desired (retrieved from U), and a region that touches the region we are attempting to change (the resource). M will make a best-effort to resize the two given regions so that the consumer region is given up to the desired amount of area from the resource region. M will return either a single region that is the full merge of the two individual regions, or two resized regions
- S_{fty} is a scaling factor that is multiplied by the desired K value before determining if a region should be split in response to having an overly large K value. This should never be below two, as splitting a region that does not have enough incoming data readings to support 2 * the desired K value is likely to result in a privacy violation for one of the descendant regions
- S_{plit} is a function that can determine the number of sub-

regions an overly large region should be split into to meet k-anonymity during the next cycle.

- S is a function that can resize a region by splitting it into multiple sub-regions. The current implementation splits into similarly sized regions. S is given the region to be to be resized (the consumer region) and a value for the desired number of partitions. It will return the generated set of regions

C. *Anonoly Execution*

When the Anonoly algorithm is initially started, R_s is initialized with a single region that encloses A completely.

```
1) For every time range $T_c$:
    a) If there is only one region in $R_s$, and if the sum of $C_r$ is less than $K$, then wait one more $T_c$
    b) Order all regions in $R_s$ using $R_{nk}$
    c) Mark all regions as unused
    d) While there are unused regions in $R_s$:
        i) Get the next unused region($R$) from $R_s$
        ii) Get the number of data readings($C$) that were input into $R$ during this cycle $T_c$
        iii) If $C$ is equal to $K$, then mark this region as used and continue
        iv) If $C$ is less than $K$, then:
            A) Retrieve the amount of desired area change for $R$ from $U$
            B) Find all the neighboring regions of $R$ that have a $C$ value is larger than $K$, and order them according to $R_{nk}$
            C) While $R$ has yet to be changed by the desired area amount (or there are no unused neighbors), pass $R$, the updated desired area change, and the next-unused neighbor to $M$, marking each the resulting regions as used
        v) If $C$ is greater than $K$ * $S_{fty}$, then:
            A) Retrieve the number of desired partitions from $S_{plit}$
            B) Pass $R$ and the number of desired partitions to $S$
            C) Mark all of the returned regions as used and add them to $R_s$
```

If the desired K value is impossible to obtain (e.g. a k-value of 100 is desired, but each time span T_c only results in 10 data readings), then this initial region will never increase in size.

D. *Ranking Regions By Their Distance From 'Optimal'*

The R_{nk} function can be implemented in multiple ways, depending upon the needs of the system. In general, this ranking algorithm is used to order regions by optimality, where different system policies determine what is more or less optimal. For example, our implementation of R_{nk} considers privacy violations as being worse (e.g. less optimal) than having far too many data readings for one region. Therefore our implementation, in attempting to range regions with a desired k of 10, would consider a region with a k-value of 8 to be less optimal than a region with a k-value of 2000, even though the latter could clearly be split into multiple regions and increase location accuracy. During Anonoly execution, R_{nk} is used to determine which regions get precedence during the region resizing, based upon an implicit assumption that allow the most non-optimal regions first priority in retessellation will result in faster system convergence to a more optimal state. Other approaches for the R_{nk} function could treat granularity as more important, or could treat distance from K as the determining factor (thereby sacrificing some locational privacy for improved granularity). Therefore, R_{nk} is an effective method of allowing a system policy configuration (e.g. privacy vs granularity).

E. *Speed of Convergence*

The U function of the DAR algorithm, which specifies how much a region should grow in response to having a smaller-than-desired k-value, can be used to adjust the desired aggressiveness in correcting privacy violations. An overly-aggressive response to a privacy violation will cause a data imprecision, as the region size will be drastically increased in order to ensure privacy. Conversely, an under-aggressive response will not increase the region size enough, and therefore the region will likely experience another privacy violation during the next cycle. For example, if a region receives five data readings but the system is using 10-anonymity, then U could either return that the region should double in size, or U could report that the region should quadruple in size if the more aggressive correction of privacy invasions is desired. The same aggressiveness arguments can be applied to the S_{plit} function for determining how many sub-regions an overly-large region is split into.

F. *Merging and Splitting Locational Regions*

The M and S functions, respectively, are used to merge and split regions. Currently there are no parameters that direct *how* these functions operate internally, and our initial implementation simply enforces that polygons remain closed and contiguous (e.g. there cannot be a gap in between two closed sections of a polygon). Our algorithms do allow for concave polygons, which can create some technical difficulties when attempting polygon transform operations such as merging two polygons. Implementations that only permit convex polygons may be less computationally intensive. Additionally, all of the tessellation maps will need to be shipped to smartphones at some point, and therefore an additional property of interest in a tessellation map would be a small filesize. This concern could be addressed in the M and S functions by attempting to use long straight lines for polygon edges whenever possible to reduce the amount of data needed to represent the tessellation map.

V. Empirical Evaluation

This section compares the Anonoly algorithm to prior static tessellation approaches using a real-world dataset obtained from the CRAWDAD.org repository [16]. We ran one experiment to directly compare Anonoly to static tessellation over the course of a two month timespan by recording the achieved k-anonymity for both algorithms, and comparing that achieved k-anonymity over the two month timespan to the desired k-anonymity. Additionally, we ran a second experiment that compares the ability of Anonoly and static tessellation to balance privacy versus data precision when the incoming data distribution was undergoing changes.

In order to bootstrap our experiments, we used the CRAWDAD.org dataset to simulate data reports entering a smartphone data collection system. Each incoming data report must contain a timestamp and a location(starting as a latitude/longitude on the smartphone device, but converted into a regional identifier before it reaches Anonoly) in order to be used as input to the Anonoly algorithm. The original dataset is a log of wireless access point associations on the Dartmouth campus, and therefore we used the time of each access point association, and the latitude/longitude location of the access point, as the required incoming data. Prior published tessellation algorithms require manual human intervention to create a tessellation map, and we therefore implemented a static-tessellation algorithm by running Anonoly for a small time on the dataset and then storing the generated tessellation for use as a static map.

A. *Experiment Setup*

These experiments were conducted on a 2.66 GHz Intel Core i7 MacBook Pro with 4Gb 1067 MHz DDR3 RAM running Mac OS X 10.6.7 and Java SE Runtime 1.6.0_24.

B. *Experiment 1: Comparing Anonoly to Static Tessellation on Real-world Data*

Static tessellation algorithms make a static prediction about the distribution of the incoming data, and therefore we predicted that this class of algorithms would perform poorly in situations where the incoming distribution of data changes significantly during data collection. In this experiment, we generated a tessellation map and then statically utilized that single tessellation map for the entire duration of data collection. On the same data, we also utilized the Anonoly algorithm to dynamically re-tessellate, allowing comparison of the Anonoly algorithm to a static tessellation algorithm.

Hypothesis: The Anonoly algorithm will avoid or mitigate the quality of service failures which cause static tessellation algorithms to be ineffective. We predicted two ways in which static tessellation algorithms could underperform. First, by over-estimating the number of data readings that enter the data collection system, static tessellation algorithms could result in a privacy violation occurring when the minimum number of readings required to ensure k-anonymity are not received. Second, static tessellation algorithms could underestimate the number of incoming data readings, which would result in the tessellation map being much coarser than necessary to ensure privacy. This 'privacy-induced imprecision' would reduce the utility of the entered data. For example, a 16x16 grid tessellation would provide much finer data for a data collection system than a 4x4 grid tessellation. In this experiment, we hoped to see evidence of these quality of service issues in a static tessellation, and evidence that Anonoly either a) completely avoided privacy violations and privacy-induced imprecisions, or b) Anonoly experienced these quality of service failures less frequently or to a lesser magnitude. If our hypothesis regarding static tessellation quality of service failures is correct, and we find evidence that Anonoly does not experience the same failures, then we will have shown that Anonoly can outperform static tessellation at preserving user privacy while increasing data precision.

Experiment 1 Results. Figure 3 shows the k-anonymity which was achieved by one a static tessellation algorithm and the Anonoly algorithm over the course of two months. Each datapoint is the median of the k-anonymity values achieved for all regions during that timeslice. The optimal values for k-anonymity are located between the blue privacy violation region, and the red privacy-induced imprecision region. By allowing the k-anonymity value to drop below the set k-anonymity algorithm parameter of fifteen into the blue privacy violation region, the privacy of users in the smartphone data collection system is no longer statistically protected. K-values above the safety margin of forty-five (e.g. in the red zone) indicate that the tessellation could be composed of smaller regions without causing a privacy violation, and therefore the precision of the locational data could be improved with no adverse affects.

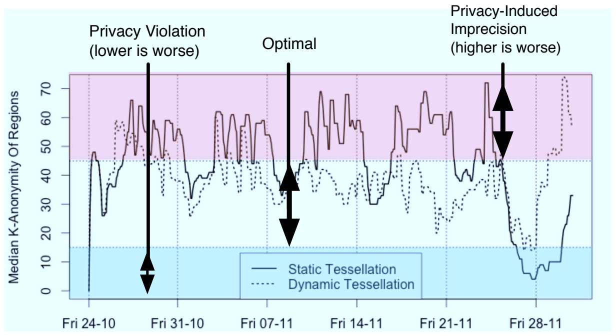

Fig. 3. Anonoly vs Static Tessellation Over Two Months

In order to better understand the results shown in Figure 3, we analyze the data on a week-by-week basis. The first week, shown in Figure 4, shows Anonoly initially mirroring the static tessellation algorithm results, but the algorithms begin to differ as Anonoly updates its assumptions about the incoming data distribution. For this experiment, the Anonoly algorithm was tuned to very aggressively react to privacy violations, operating under a policy that considered data precision only after ensuring privacy was maintained. Therefore, near 10-26 datapoint the Anonoly algorithm reacts to a slight drop in the median k-value and corrects too aggressively, reaching into the non-desirable privacy-induced imprecision region. However, the height of Anonoly's over-correction is still significantly lower than the height of the static algorithm's imprecision on 10-27. Therefore, while the policy chosen for Anonoly favors adding imprecision over potentially violating privacy, Anonoly still performs with significantly more precision over the week-long timespan.

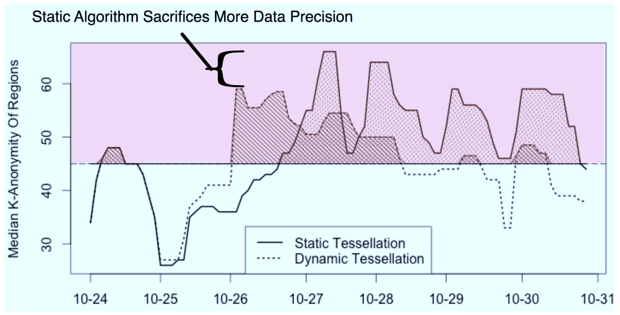

Fig. 4. Anonoly vs Static Tessellation; Oct 24-Oct 31 Timespan

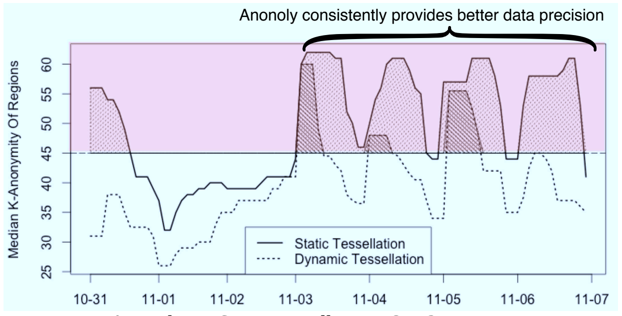

Fig. 5. Anonoly vs Static Tessellation; Oct 31-Nov 7 Timespan

On the third week of data the difference between the static algorithm and Anonoly become more significant. There are multiple k-value peaks in the static algorithm where the regions in the tessellation are larger than is required to ensure user privacy. However, the Anonoly algorithm is able to effectively avoid sacrificing data precision needlessly, and generates a finer tessellation map as the number of incoming data readings increases, thereby gaining data precision while safely maintaining the required level of data privacy.

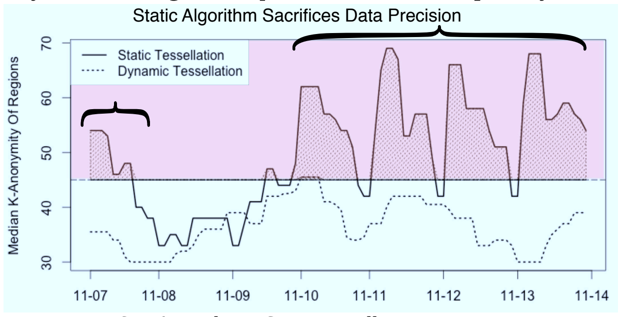

Fig. 6. Anonoly vs Static Tessellation; Nov 7-Nov14

With the fourth week of data (shown in 7) the Anonoly algorithm is capable of entirely avoiding the privacy-induced imprecision, except for a small portion of Nov 17.

On the fifth week of the dataset, first privacy violations occur. In the hours immediately preceding Nov 24th, Anonoly reacts to a large increase in the number of incoming data readings (apparent by the rise of k-values in the static algorithm) and over-aggressively splits regions, causing a privacy violation for a short period of time. The Anonoly algorithm rapidly corrects its mistake, and does so without causing the privacy-induced imprecision error that we see in the static algorithm. Moreover, from Nov 26-Nov 28, there is a

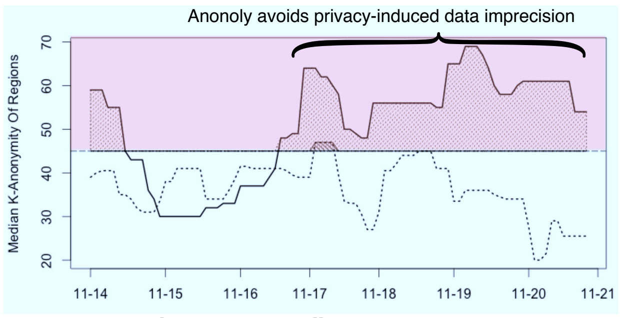

Fig. 7. Anonoly vs Static Tessellation; Nov 14-Nov 21 Timespan

significant drop in the number of incoming data readings, and the static algorithm has an extended period of violating user privacy. Anonoly, however, manages to react appropriately and maintain k-values in the optimal region for this two day period.

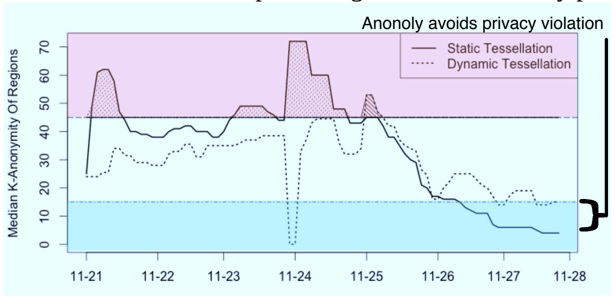

Fig. 8. Anonoly vs Static Tessellation; Nov 21-Nov 28 Timespan

Figure 9 shows a quantitative comparison of quality of service failures. For imprecision quality of service failures, this value was created by summing any k-values over the imprecision cutoff of forty five. For privacy, this value was created by summing any negative distance from fifteen for all data readings taken in that week. The figure shows that Anonoly was substantially more effective at mitigating both privacy violations and imprecision violations.

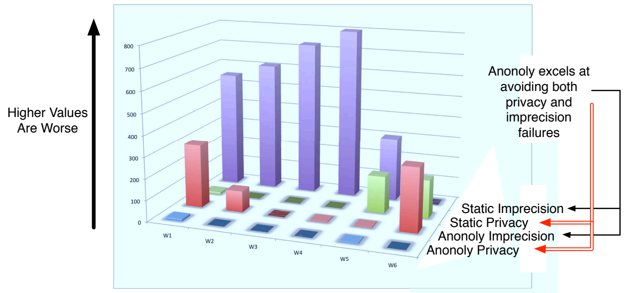

Fig. 9. Anonoly vs Static Tessellation Quality of Service; 6 Week Timespan

C. Experiment 2: Evaluating Anonoly's Ability to Maintain Desired K-anonymity Value

The primary goal of Anonoly is to construct regions that will have the desired k-anonymity property, in order to ensure end-user privacy. In this experiment, we tested Anonoly's ability to maintain the desired level of user privacy e.g. k-anonymity, across a range of algorithm configurations and incoming data distributions. We executed the static tessellation, and the Anonoly algorithm, on a two month section of the CRAWDAD.org dataset, setting multiple desired k-anonymity values and using multiple timeslice sizes. Our aim in this experiment was to extend V-B, which showed us that Anonoly could avoid privacy-induced imprecision and privacy violations, to also show that Anonoly can avoid these quality of service failures while also maintaining the desired k-anonymity value.

Hypothesis: The Anonoly algorithm will have k-values closer to the desired k-anonymity than static tessellation's generated k-values. We hypothesized the Anonoly algorithm would enforce k-anonymity as well or better than static tessellation methods. Due to Section V-B showing that Anonoly is able to successfully avoid or mitigate the two quality of service failures present in static tessellation algorithms–privacy violations and privacy-induced imprecisions–we predicted that Anonoly would also be able to more effectively maintain a desired k-anonymity value, even with a wide range of algorithm parameters and different incoming data distributions. Moreover, we predicted that Anonoly could achieve this result without the strict requirement for *a priori* information necessary to seed a static tessellation algorithm, making the Anonoly algorithm more useful than static tessellation algorithms for many real-world data collection systems where this *a priori* information is unavailable.

Experiment 2 Results. Figure 10 shows k-anonymity values generated by the Anonoly algorithm and a static tessellation. Multiple desired k-values are considered on the x-axis, and the actual achieved values are plotted on the y-scale. Each point represents the median of the k-values, error bars represent the 5-95th quantiles. Each algorithm (static and Anonoly) was evaluated with three different timeslices for each desired k-value, and all achieved k-values were merged before generating the 5th, 50th, and 95th quantiles.

Figure 10 shows that Anonoly's 5th quantile is consistently at or above the privacy violation, while the static tessellation consistently falls below the privacy violation margin. Moreover, the static tessellation's 95th quantile is consistently above the Anonoly's algorithms' 95th, except for x=75 where the two are equal. Therefore, Anonoly manages to perform better than static tessellation in all scenarios.

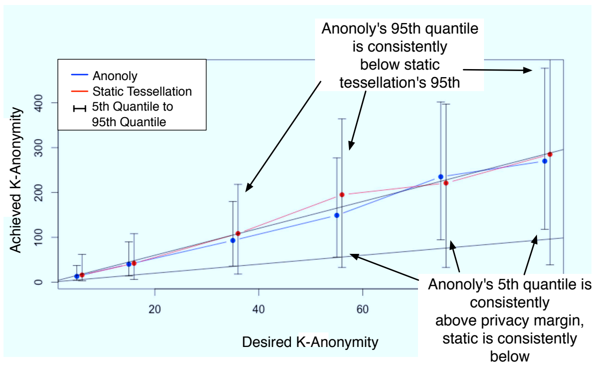

Fig. 10. Achieved vs Desired K-anonymity; Two Month Timespan

To highlight the difference between Anonoly and a static tessellation, we created a histogram of the data used for the x=55 portion of Figure 10, with the results shown in figure 11. The upper distribution is from Anonoly, while the lower is from the static tessellation. The desired region for k-values to fall is between the two vertical lines, with more left values indicating privacy violations, and more right values indicating privacy-induced imprecision values. Figure 11 shows Anonoly is doing a substantially better at maintaining k-values within the desired region. Anonoly is configured to rank privacy violations as more serious than privacy-induced imprecision violations, and therefore almost all of the quality of service violations are imprecision violations, caused by Anonoly ensuring that privacy violations do not occur. Moreover, Anonoly manages to have fewer overall privacy-induced imprecision violations, and the ones that it does have are contained closer to the desired region.

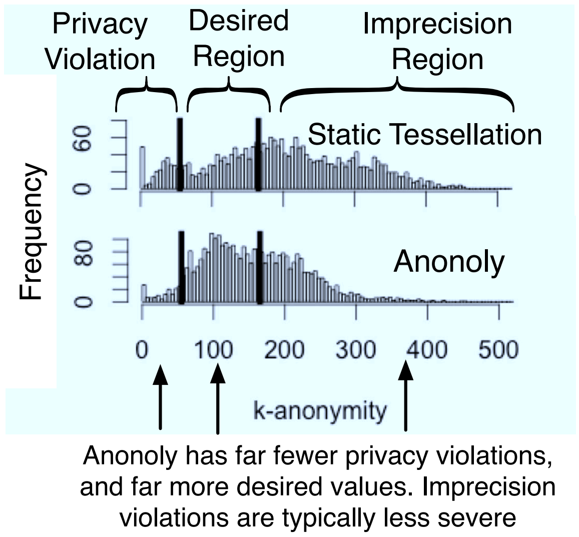

Fig. 11. Distribution of Anonoly vs Static Tessellation K-values; Two Month Timespan

VI. Related Work

This section compares Anonoly to prior research, outlining a taxonomy of related works that includes spatial/temporal blurring, k-anonymity, and pre-production region generation.

Spatial or Temporal Blurring reduces the precision of data readings to help ensure the privacy of the user's location [17]. For example, users may choose to only report their location to an accuracy of within 1 mile. The primary limitation with using blurring independently is that it does not guarantee privacy. Entering data reports with very imprecise locations may seem secure, but there is no guarantee that multiple data reports were entered for each imprecise location region. Therefore, an attacker that knows a user's rough location at a time of data reporting may be able to reassociate that user and their data report with minimal difficulty because there are very few reports from that location at that time. Moreover, user-set privacy preferences can actually leak user identity. Any non-standard locational imprecision settings, such as highly-privacy very imprecise location reporting, can actually be used by an attacker to identify data reports coming from the non-standard user (e.g. there may only be one user that has extremely inaccurate location reporting, making it easy to identify that user's data report). Anonoly, however, uses both a standard imprecision metric and ensures that there are multiple reports for every location region.

K-anonymity is a method that groups and manipulates data so that 'k' data items are indistinguishable from one another, which solves the issues of simply applying spatial or temporal blurring. This method is typically applied only after the data has been received, so the system can ensure that there are at least k-1 data readings in the same locational region [17]–[19]. This requires smartphone users to share their private data with nontrusted sources. Anonoly in contrast operates by sharing a tessellation map with end-user devices, therefore enforcing k-anonymity without requiring private data to be shared with a nontrusted source.

Region Generation is the tessellation of a large area into multiple regions, where each region has an id that is valid to submit to the smartphone data collection system as a localization method [12]. Predictions about the incoming data report locations and times are used to generate the regions, and if these predictions match the real-incoming data then all incoming data reports will be k-anonymous. The tessellation map is shared with smartphones, thereby overcoming the requirement for private data to be shared with nontrusted parties. A key issue is that it is not always possible to predict the location and time distribution of data readings in advance. Moreover, the distribution will likely change over time, rendering the original assumption incorrect and the assumption ineffective. As we have described in Section V, Anonoly operates by updating its assumption over time, thereby reflecting a much closer approximation to the real-world data.

VII. Concluding Remarks & Lessons Learned

An emerging privacy challenge in smartphone data collection systems is that data reports can be reassociated with the specific users that submitted those reports, allowing multiple types of private user data to be exposed. This possibility of reassociation has a number of potential ramifications, including deterring smartphone users from participating in data collection systems, increasing the severity of malicious or accidental system data leaks, and reducing the potential for commercial datasets to be released for research purposes.

Existing research on preventing reassociation of location data with users leverages a static tessellation approach based on the expected spatial temporal distribution of future data readings. These research approaches are frequently so aggressive in protecting user privacy that the incoming data is relatively useless to a data collection system (e.g. all of the interesting details or features of the dataset have been removed). Moreover, current methods require *a priori* information specifying the temporal and spatial distribution of incoming data to seed the algorithms, which is impractical for many smartphone data collection systems where this information is nonexistent or difficult to predict.

This paper presented, Anonoloy, an algorithm for dynamically tessellating a geographical region in order to ensure user privacy without severe loss of data precision. Empirical results that we obtained from experiments on real-world datasets show that Anonoly can successfully protect user privacy better and provider higher data precision than prior static approaches. Furthermore, Anonoly can be seeded with *a priori* information about the incoming data, or it can operate with no *a priori* information, allowing it to be used in situations where the expected spatial and temporal distribution of data reports is not known.

The Anonoly algorithm implementation, and the experiments and data described in this paper, are available in opensource form from https://github.com/crabpot8/Anonoly.

References

[1] (2010, Nov.) Gartner says worldwide mobile phone sales grew 35 percent in third quarter 2010; smartphone sales increased 96 percent. [Online]. Available: http://www.gartner.com/it/page.jsp?id=1466313

[2] (2010, Oct.) Gartner says worldwide pc shipments grew 7.6 percent in third quarter of 2010. [Online]. Available: http://www.gartner.com/it/page.jsp?id=1451742

[3] W. Park. (2010, June) Gulf oil spill apps let you track and report on bp deepwater horizon disaster. [Online]. Available: http://www.intomobile.com/2010/06/01/gulf-oil-spill-apps-let-you-track-and-report-on-bp-deepwater-horizon-disaster/

[4] T. Saponas, J. Lester, J. Froehlich, J. Fogarty, and J. Landay, "iLearn on the iPhone: Real-Time Human Activity Classification on Commodity Mobile Phones," *University of Washington CSE Tech Report UW-CSE-08-04-02*, 2008.

[5] J. Froehlich, T. Dillahunt, P. Klasnja, J. Mankoff, S. Consolvo, B. Harrison, and J. Landay, "UbiGreen: investigating a mobile tool for tracking and supporting green transportation habits," in *Proceedings of the 27th international conference on Human factors in computing systems*. ACM, 2009, pp. 1043–1052.

[6] C. Thompson, J. White, B. Dougherty, and D. C. Schmidt, "Optimizing Mobile Application Performance with Model-Driven Engineering," in *Proceedings of the 7th IFIP Workshop on Software Technologies for Future Embedded and Ubiquitous Systems*, 2009.

[7] W. D. Jones, "Forecasting Traffic Flow," *IEEE Spectrum*, 2001.

[8] G. Rose, "Mobile phones as traffic probes: practices, prospects and issues," *Transport Reviews*, vol. 26, no. 3, pp. 275–291, 2006.

[9] P. Leijdekkers and V. Gay, "Personal heart monitoring and rehabilitation system using smart phones," in *Proceedings of the International Conference on Mobile Business*. Citeseer, 2006, p. 29.

[10] T. Das, P. Mohan, V. N. Padmanabhan, R. Ramjee, and A. Sharma, "Prism: platform for remote sensing using smartphones," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, ser. MobiSys '10. New York, NY, USA: ACM, 2010, pp. 63–76. [Online]. Available: http://doi.acm.org/10.1145/1814433.1814442

[11] C. Cornelius, A. Kapadia, and D. Kotz, "AnonySense: Privacy-aware people-centric sensing," *Proceeding of the 6th* ..., 2008.

[12] A. Kapadia, N. Triandopoulos, and C. Cornelius, "AnonySense: Opportunistic and privacy-preserving context collection," *Pervasive* ..., 2008.

[13] H. Lu, N. Lane, S. Eisenman, and A. Campbell, "Bubble-sensing: A new paradigm for binding a sensing task to the physical world using mobile phones," in *Workshop on Mobile Devices and Urban Sensing, IPSN*. Citeseer, 2008.

[14] A. Kapadia and D. Kotz, "Opportunistic sensing: security challenges for the new paradigm," ... *Systems and Networks* ..., 2009.

[15] L. Sweeney *et al.*, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty Fuzziness and Knowledge Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[16] D. Kotz, T. Henderson, I. Abyzov, and J. Yeo, "CRAWDAD trace set dartmouth/campus/movement (v. 2005-03-08)," Downloaded from http://crawdad.cs.dartmouth.edu/dartmouth/campus/movement, Mar. 2005.

[17] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati, "k-anonymity," 2007.

[18] L. Sweeney, "K-anonymity: A model for protecting privacy," in *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 10, no. 5, 2002, pp. 557–570.

[19] K. Emam and F. K. Dankar, "Protecting privacy using k-anonymity," vol. 15, no. 5, 2008.