Maintaining the Balance between Privacy and Data Integrity in Crowd Sensing Systems

Name

Dept and University

EMail

Date

ABSTRACT

As tapping into the sensory data and resources of smartphones becomes common place, it is necessary to ensure the privacy of the device user whilst maintaining the accuracy and integrity of the data collected. Crowd sensing systems often sacrifice privacy for data integrity or vice versa. It has also become important to limit the computational cost and burden on user devices, as increasingly more services desire to tap into the resource that these devices have become. The framework being proposed, called the balanced truth discovery (BTD) framework, attempts to meet all three of the aforementioned demands. The simple nature of the framework also leaves the possibility for easy modification (e.g. cryptography and weight assignment). Possible modifications in these areas will be discussed. BTD framework reduces user participation in truth discovery. This reduces computation cost for the user device, but also limits the interactions between the user device and the server, which is essential to data integrity. BTD framework also takes steps to blur the user device's original sensory data, by processing results in groups called zones. An enhanced method takes privacy preservation a step further, by protecting the user from an untrusted data-collecting party. Analysis of simulations running the BTD framework will provide evidence for the preservation of data integrity.

1. INTRODUCTION

Smartphones and other handheld devices are being used as a sensory-data resource at an increasing rate. This has revealed three overwhelming demands in the process in discovering truths: privacy for user, data integrity for data-collection party, and low computational costs as demanded by the natural growth of crowd sensing systems. Other truth discovery and crowd sensing systems [1-6] have a range of results on the spectrum from private to accurate, as well as a range of computational cost for user devices.

These frameworks often sacrifice one of the three demands to satisfy another, or simply neglect a demand. PPTD [1] focusses on preserving the privacy of a user participating in a crowd sensing system. However, in doing so, the PPTD framework obscures the original sensory data of the user devices and requires a greater sample size before the estimated truth becomes accurate. This is caused specifically when the estimated ground truth is initialized randomly. This is a direct sacrifice of data accuracy for privacy.

The other frameworks also place computational burden on user devices. PPTD, in particular, requires user devices participation in the weighted calculation and partial decryption of the estimate ground truth. A computation which is by itself small, but in great numbers a burden on the user device. Furthermore, the more user intervention the greater the chance for loss of data integrity, be it from faulty equipment or an active attack on integrity. The possibility of an active attack in the real world is very real. Whether the attacks come from a third party or the device user themselves, it must be addressed in order to preserve data integrity.

The balanced truth discovery framework being proposed takes the ideas put forth by PPTD [1] and other frameworks [2-6] and attempts to satisfy all three of the demands of crowd sensing systems. By combining some of the privacy preserving functions of PPTD and the idea of zones and blurring data at the individual level [2], privacy is preserved. Limiting the participation of user devices in the computation of the estimated truth, results in a reduction in both the cost and burden put onto user devices as well as the damage caused by an active attack on data integrity. This framework treats user devices as if they were a database resource of sensory data. It connects to the device, demands a low-cost result and gets out, leaving the user device for other services to use (in layman's terms).

The process begins by initializing the estimated ground truth with a random, within a context, value. The randomly-initialized ground truth is sent to *k* user devices; these *k* devices are called a zone. A device receives the value and combines it with its own sensory data. The device then sends its result to the server. The device may encrypt the data using any desired cryptography. This ends the user device's participation in the crowd sensing system. The server then receives the *k* results from *k* devices and finds the average. Using this average, the server then extracts the original randomly-initialized ground truth using an extraction method. The estimated ground truth that the server is left with is an exact match to an average of the original sensory data of the user devices.

The method of masking a user device's sensory data with a combined average, summing the results of a zone, then removing the so-called "mask" to be left with a true average will repeat with every zone being processed. The results from each zone is aggregated with the currently estimated ground truth. This method prevents individual data from being sent to the server and solely representing the currently estimated ground truth. This prevents personal information from being acquired through eavesdropping.

In the event that the data-collecting party is untrusted, the BTD framework provides a method with enhanced privacy. The enhanced method allows user devices to use a random weight when calculating the average of their own sensory data and the estimated ground truth sent by the server. This will prevent the server from being able to extract the user device's sensory data with 100% accuracy by using the same extraction method previously described. The range of this weight can be provided to the user by the server, to best fit the context of the ground truth, or by the user device for maximum protection. It is suggested that the user device selects its own range based on its evaluation of the data-collection party (i.e. use zero weight variance with trusted sites, +/- *v*% with unknown/untrusted sites).

The BTD framework also leaves the ability to add modifications. Cryptography is left completely up to the user of this framework. The ability to weigh the reliability of user devices is also in the hands of the user of this framework. Weighting algorithms and methods can be found in [1, 3, 7 and 8].

In this paper, we will first go in-depth with the related works that inspired the methods within this framework in Section 2. Section 3 will describe the methods within the framework: 3.1) The initialization of the ground truth and zone processing, 3.2) The extraction method, 3.3) The possible modifications as well as suggested approaches to cryptography and reliability weight. Section 4 will describe the enhanced method. Finally, we will conclude with Section 5, the analysis of the simulations running the BTD framework, as well as a conclusion in Section 6.

2. MOTIVATION FROM RELATED WORKS

The purpose of this paper is to correct and build upon the framework laid out by *Cloud-Enabled Privacy-Preserving Truth Discovery in Crowd Sensing Systems* [1]. To prevent user devices from sending their individual sensory data to the server, as well as preventing the estimated ground truth from reflecting an individual's sensory data, the ground truth is initialized with a random value. However, the authors of this paper [1] do not address the effect on data accuracy that this has. It is the extraction method, discussed in Section 3.2, which addresses and completely solves this problem in the BTD framework.

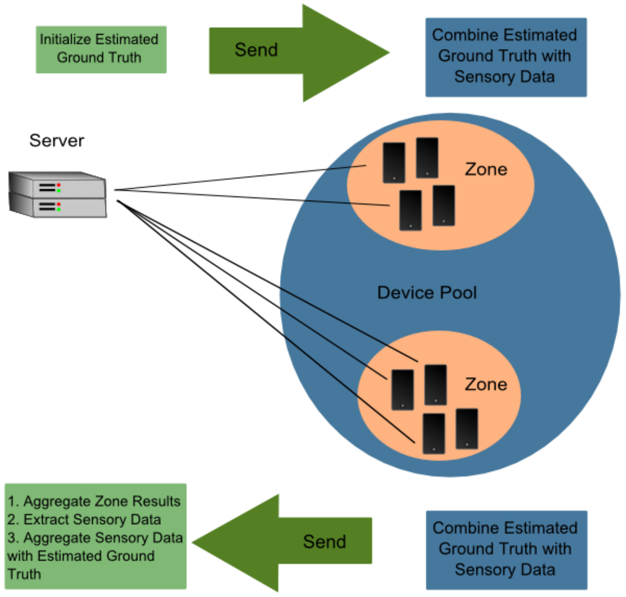
The PPTD framework [1], also requires users to participate in the truth discovery, not only by calculating and sending a result based on sensory data, but also by making a weighted calculation and decrypting a portion of the estimated ground truth for the server. Although this preserves privacy to a great degree, this presents two issues.

First, the low cost demand as presented by the increase in desire of crowd sensing services. The performance evaluation given [1], shows a runtime on a smartphone around 0.039s. As the demand for crowd sensing increases this number may prove to be too large. As an example, if ten services happened to utilize the same device, 0.4s of computation is humanly noticeable and may disrupt the user experience of the device.

Second, the integrity of data as demanded by the data-collection party. Although the weighted reliability attempts to prevent bad sensory data from affecting the estimated ground truth, the second chance to participate for user devices leaves a possibility for bad data or even an attack to affect the estimated ground truth. It is unclear the degree of damage resulting from a computational error from faulty equipment. The PPTD framework makes the assumption that no device user is actively trying to modify that data which it sends. For real world use, this has to be addressed. The user, instead of sending a decrypted portion of the result, could send back an extreme result attempting to alter the data. It is also unclear what kind of damage this may cause.

The BTD framework makes use of zone processing, which attempts to blur data at an individual level as inspired by *A Privacy-Preserving Location Monitoring System for Wireless Sensor Networks* [2]. This paper proposes a system which masks an individual's sensor information within a "cloaked area" [2]. The sensory data that results from this area is the aggregated sensory data of multiple devices. This idea is utilized in the BTD framework, by only processing sensory data in groups of devices called zones. The individual sensory data of a device is never handled outside the device itself.

3. THE BTD FRAMEWORK


Initialize Estimated Ground Truth
Send
Combine Estimated Ground Truth with Sensory Data
Server
Zone
Device Pool
Zone
1. Aggregate Zone Results
2. Extract Sensory Data
3. Aggregate Sensory Data with Estimated Ground Truth
Send
Combine Estimated Ground Truth with Sensory Data

The BTD framework is comprised of the following steps:

1. SERVER: Randomly initialize estimated ground truth
2. SERVER: Send estimated ground truth to *k* devices (a zone)
3. DEVICE(S): Using equation (1), calculate the average of the estimated ground truth and device's sensory data
4. DEVICE(S): Send result to server
5. SERVER: Calculate the average of all *k* results from a zone
6. SERVER: Extract sensory data by using equation (2) to remove the data sent in step 2.
7. SERVER:
    a. If this is first zone, set the estimated ground truth equal to the result of step 6.
    b. If this is not first zone, aggregate the result of step 6 with the current estimated ground truth using equation (3).

3.1 Initialization and Zone Processing

Section 3.1 will discuss steps 1-5.

As inspired by the PPTD framework [1], the BTD framework begins by initializing the estimated ground truth to a randomized value. The initialization happens within the server by the data-collection party. The randomized value will have little to no effect on the estimated ground truth when using the BTD framework. However, it is still suggested to use a random value that makes contextual sense, especially when using the enhanced method described in Section 4. This initialized value is used, so no individual sensory data is sent to the server.

The randomized ground truth is sent to *k* user devices, which the server wishes to utilize. These *k* devices represent a zone and will be processed together, so no individual sensory data is handled by the server. The number *k* represents the size of the zone. The value of *k* can be chosen by the data-collection party. If the aforementioned party utilizes only a small number of devices, it is suggested to use a small value for *k*. A small value for *k* ensures that the estimated ground truth is updated frequently. The value of *k* may also be determined within context. The way *k* devices are chosen for a particular zone is entirely up to the data-collection party to decide. It may make contextual sense to divide a crowd into *n* zones of size *k* based on location, device specifications or simply as they become available to the server. The degree of anonymity required varies from context to context.

Each of the *k* user devices calculate the summation of their own sensory and the estimated ground truth sent by the server. The summation is calculated as if there are only two parts, their own weighed at 50% and the servers, also weighed at 50%. The formula appears below:

$$x^k = \text{sensoryData} * 0.5 + x * 0.5 \tag{1}$$

Each of the *k* user devices sends the data back to the server once their individual calculations are complete. The server aggregates the results. The BTD framework simulated for analysis purposes simply finds the mean of the results without weighted reliability.

The utilization of zone processing obscures data at an individual level, but the bigger picture (the one of *k* devices) remains crystal clear. As an example, the location and change of location of a group of people in New York City can be monitored. If someone attempts to learn the individual location and habits of a person, the data becomes indecipherable within the group. However, useful information can found that can answer these questions: where do people go in New York City in the evening versus where they go at night, which burrows have the more active nightlife, what is the demographic or origin of people who visit *this* particular area at *this* particular time?

Algorithm 1: Zone Processing

```
        Input: k user devices: device[k]
        Output: Estimated ground truth: x

1 Randomly initialize the ground truth for each object;
2 repeat
3       for each user device k do
4       Send ground truth to user device;
5       Aggregate device results;
6    end
7       if this is first zone do
8       Extract randomized value;
9       x = remainder;
10   else do
11      Extract value;
12      Aggregate remainder and x;
13 until all zones have been processed;
14 return x;
```

Algorithm 2: User Calculation

```
        Input: Estimated ground truth: x
        Output: Result

1       Result = aggregate sensory data and x
        (e.g., Eqn. (1));
2 return Result;
```

3.2 Extraction Method

Section 3.2 will discuss steps 6-7.

Randomly initializing the ground truth (within a range based on context), presents a data accuracy issue if not properly addressed. The BTD framework addresses the issue using a method which extracts the estimated ground truth sent to *k* users within a zone from their aggregated sensory data. As a reminder, the user calculation aggregates the user device's sensory data with the estimated ground truth provided by the server as two equal parts. Therefore, the aggregated sensory data of the *k* user devices in a zone is one half user device sensory data and one half estimated ground truth. The following equation extracts the estimated ground truth and leaves the server with the aggregated sensory data:

$$s^z = x^z * 0.5 + [x^z - 2(x - x^z)] * 0.5 \tag{2}$$

Where s^z represents the aggregated sensory data of zone *z*, x^z represents the aggregated results supplied by user devices (i.e. s^z pre-extraction), and *x* is the currently estimated ground truth. This extraction method is not only used to extract the randomized value the estimated ground truth is initialized to, but also in conjunction with the following equation to update the ground truth:

$$x = s^z * k/c + x * (c-k)/c \tag{3}$$

Where *k* is the number of user devices in a zone and *c* is the number of user devices that have participated in the crowd sensing system including the *k* user devices of the zone currently being processed.

The extraction method, used in conjunction with the randomly initialized value and the idea of cloaking the sensory data of a user device, allows for the preservation of privacy without a sacrifice in data integrity. The results of this method are theoretically exact copies of the aggregation of the sensory data of all *c* devices.

3.3 Possible Modifications

3.3.1 Cryptography

The BTD framework does not implement a specific cryptography method. However, it is still suggested to use a cryptosystem with the BTD framework to correctly preserve privacy of all parties involved. A potential threat exists if a third party intercepts both the data sent from the server to the user device and the data sent from the user device to the server. If the attacker who intercepted the data understood the nature of the BTD framework, they could use the extraction method to obtain the individuals sensory data (Note: The enhanced method in Section 4 will protect against this). It is also important to note that the estimated ground truth for which the data-collection party is aggregating data, may not be information the party wishes to disclose to a third-party

In order to meet the demand for low-cost computation on the user device side, it is suggested that low-cost cryptosystem be used. Examples of possible cryptosystems AES, DES, RSA, Blowfish, etc. The cryptosystem used in PPTD [1] is complex and requires user device participation at a great degree; cryptosystems like these should be avoided if possible.

3.3.2 Weighted Reliability

The BTD framework does not implement a method to weigh the reliability of user devices. Each device has an equal say in the aggregation of the ground truth. Weighted reliability theoretically protects the integrity of the data. It is suggested that a weighted reliability be used within the BTD framework. (See [1, 3, 7, 8]).

PPTD [1] offers a weighted reliability method that can be altered slightly to work with BTD. The alteration is necessary as PPTD sends weight information to user device, whereas BTD requires such a method be calculated and used purely on the server side. The equation used for weighted reliability as given by PPTD [1] is as follows:

$$w_k = \log\left(\frac{\sum_{k'=1}^{K}\sum_{m=1}^{M} d(x_m^{k'}, x_m^*)}{\sum_{m=1}^{M} d(x_m^{k}, x_m^*)}\right)$$

$$d(x_m^k, x_m^*) = \frac{(x_m^k - x_m^*)^2}{std_m}$$

These equations, when used together, base the weight of a user device on the standard deviation and the normalized squared distance function. It is also important to note that the reliability of a device may differ from context to context [3]. For best possible data integrity, it is suggested that weighted reliability be calculated by the server for each user device for each context (i.e. if the server is finding two or more ground truths using different sensors).

4. Enhanced Method

When handling the privacy of device users, it is important to address all possible threats to privacy. This includes the possibility of untrustworthy data-collection parties. If a server claims to be running the BTD framework, but is actually phishing for personal data, the individual sensory data of a user device may be obtained by using the extraction method; this is assuming the server fails to aggregate the data among *k* user devices in a zone. To correctly preserve privacy, we must protect privacy against threats from all parties not just outside threats and eavesdropping.

The enhanced method addresses this issue by blurring the individual sensory data once more [2]. The method allows user devices to use a randomized weight (within a range in context) when combining its own sensory data to the estimated ground truth provided by the server. This prevents the server from extracting the exact individual sensory data. The extraction method is only exact when the result being sent from the user device to the server is 50% device sensory data and 50% estimated ground truth. This does have a negative effect on data integrity. However, as data is processed within a zone of *k* user devices all with their own small variance on weight the aggregated result is very accurate. The data integrity become imperfect, but within certain contexts the effect is minuscule.

When using the enhanced method, there are multiple ways to assign a proper weight variance. The variance on weight may be supplied by the server, in order to satisfy the requirements of the context, or may be held within the device itself. It is suggested to take a combined approach. Let the server define the context, but if the server declares a variance that is too small, risking the privacy of the device user, the user device can use a value declared within its own range of safe variance options.

Algorithm 3: Enhanced Method (User calculation

```
        Input: Estimated ground truth: x, Variance:  v
        Output: Result
```

```
if v is not a sufficient value do
        Replace v with desirable value;
end
weight = random between 50-v and 50+5;
Result = (sensory data * weight/100) + (x * 100-
        weight/100)
return Result;
```

5. Evaluation: Simulation and Analysis

5.1 Java Simulation Code

For the purpose of experimenting and testing, a simple simulation running the BTD framework was made using Java programming language. The simulation makes use of three classes: 1) the simulation class, 2) server and 3) device. The server class defines a server object, which has the necessary methods to mimic the behavior of a server running BTD as described throughout this paper. The device class in a similar manner, mimics the behavior expected of a user device. The simulation class provides the server with the necessary devices to calculate a ground truth. The simulation program, as a whole, is equipped to run a crowd sensing simulation using BTD framework both with and without the enhanced method.

Device Class

Each device object initializes a sensory data variable. It is random within a range in context. For this particular simulation: 75-125.

```
public class Device {

   private int sensorData;
```

The class is equipped with two methods: addData(double x) and addDataEnhanced(double x, double v). These are simulations of Algorithm 2 and 3, respectively.

```
public double addData(double estimation){
       //Calculate average
       double truth=estimation*(50/100) +
sensorData*(50/100);

       return truth;
   }

   public double addDataEnhanced(double estimation,
double v){
       //Generate varied percentage
       Random rand = new
Random(System.currentTimeMillis());
       double vP = (double) (rand.nextInt((int)(v*200)) +
            50 * 100 -v*100)/100.00;

       //Calculate average
       double truth=estimation*((100-vP)/100) +
sensorData*(vP/100);

       return truth;
}
```

Server Class

The server object has a variable k representing zone size, variable uCount representing the number of user devices that have participated in the crowd sensing system, and a variable truth representing the estimated ground truth. When a server object is initialized, the truth variable is initialized using a random value within the range of plausible values in the context. For this simulation: 50-150.

```
public class Server {
   private double truth;
   private int k;
   private double p;
   private int uCount;
   private double std;

   public Server(int min, int max, int zone){
       Random rand = new
Random(System.currentTimeMillis());
       truth = (double)rand.nextInt(max-min)+min;
       k=zone;
       uCount=0;
       p=100.00/uCount;
   }
```

The class is equipped with three methods: processZone(Device[] d), processZoneEnhanced(Device[] d, double v), and getTruth() to display the end result. The two processZone methods are simulations of Algorithm 1, where the first calls the addData method for each device, and the second calls the addDataEnhanced method.

```
public void processZone(Device[] d){
       //zEst holds estimations given by sensory data
       double zEst= 0;

       //get sensory data from devices
       for(int i=0; i<k;i++){
         zEst += d[i].addData(truth);
```

```
        }
        zEst /= k;

        //increment uCount
        uCount += k;

        //Extract sensory data and aggregate
        if(uCount>k){
           zEst = (zEst * 0.5) + ((zEst-(truth-zEst)*2) *
             0.5);
           truth = (zEst* ((double)k/(double)uCount)) +
             (truth * (((double)uCount-(double) k) /
             (double) uCount));
        } else {
           truth = (zEst * 0.5) + ((zEst-(truth-zEst)*2) *
             0.5);
        }
    }

    public void processZoneEnhanced(Device[] d, double
v){
        //zEst holds estimations given by sensory data
        double zEst= 0;

        //get sensory data from devices
        for(int i=0; i<k;i++){
          zEst += d[i].addDataEnhanced(truth, v);
        }
        zEst /= k;

        //increment uCount
        uCount += k;


        //Exctract sensory data and aggregate
        if(uCount>k){
           zEst = (zEst * 0.5) + ((zEst-(truth-zEst)*2) *
             0.5);
           truth = (zEst* ((double)k / (double) uCount))  +
(truth * (((double) uCount-(double)k) /         (double)
uCount));
        } else {
           truth = (zEst * 0.5) + ((zEst-(truth-zEst)*2) *
             0.5);
        }
    }
```

Simulation Class

The simulation class defines the context for the truth discovery. The range of plausible values is between 50 and 100. Variables such as zone size and weight variance are manually changed within this class to meet the testing needs.

5.2 Analysis of Simulations

The purpose of these simulations was to measure the accuracy of the data. Computation cost on user devices, relative to PPTD [1], is assumed to be less.

5.2.1 Basic Simulation (Non-enhanced)

For this simulation, the estimated ground truth as calculated by the server object was compared against the average sensory data of all devices. The simulation uses 360 devices, 6 to a zone for 60 zones. The absolute value of the difference between the estimated ground truth from the server and the average sensory data from the devices is calculated for 400,000 identical trials.

The average difference calculated by the simulation is <5.0e-14. This figure is virtually zero. It arises only when extremely small decimal numbers, believed to be outside the range of accuracy of the CPU, are used in calculations. Altering the size of zones has no effect on this result.

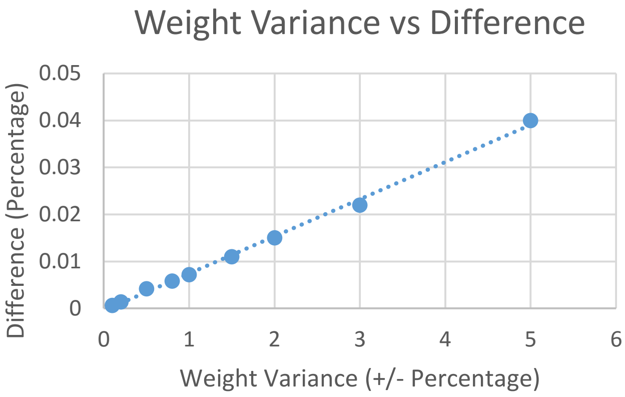
5.2.2 Enhanced Method Simulation

The majority of analysis is made for the enhanced method and its effects on data accuracy and integrity.

Simulation 1)

Device Pool: 360 devices
Zone Size: 6 devices
Zone Count: 60 zones

The value of weight variance is measured against its effect on data accuracy. As with all the simulations in Section 5, the simulation outputs a result that represents the difference between the estimated ground truth as calculated by the server vs the average sensory data from devices.



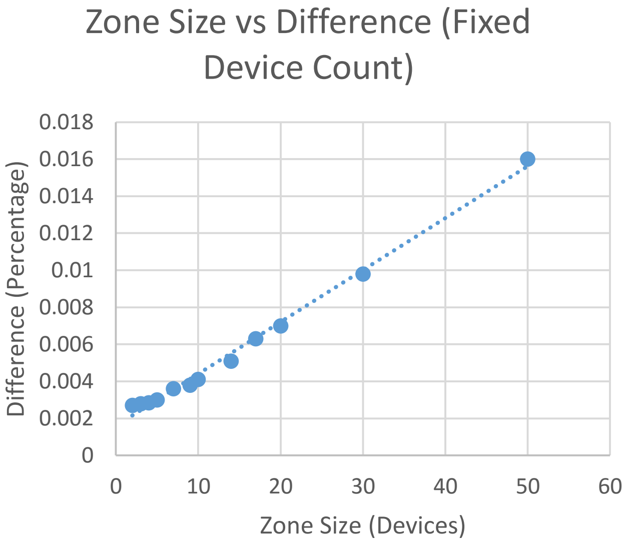
Graph (a): Percentage weight variance vs Difference

The results of simulation 1 show an increase of percentage difference (decrease in accuracy) as weight variance increases. This increase appears to be a linear fashion. User devices using a +/- 5% variance in weight have an average effect of 0.04% on the accuracy of data. In many contexts, this effect is negligible, however certain contexts may require a lower variance.

Simulation 2)

Device Pool: 500 devices
Zone Size: Variable
Zone Count: 500 devices / Zone Size
Variance: +/- 0.5%

The purpose of Simulation 2 is to calculate the effect that zone size has on data accuracy when using the enhanced method. Zone sizes 2-50 are used in the simulation and the device pool is a fixed size.



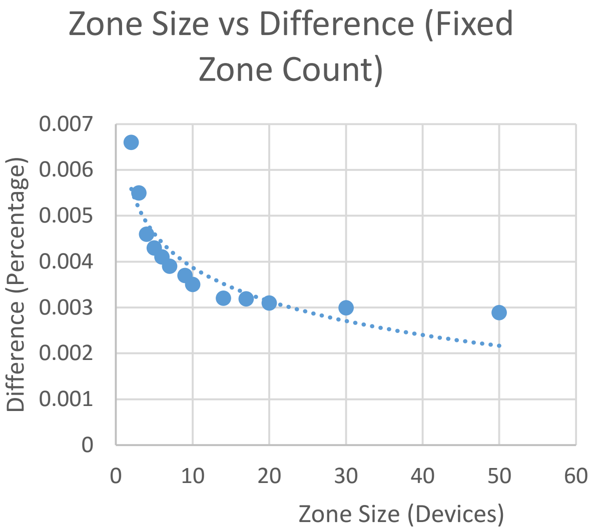
Graph (b): Zone Size vs Difference with fixed device pool

The results of simulation 2 show that an increase in zone size causes an increase in difference (decrease in accuracy) while maintaining a fixed device count. The magnitude of the effects are small (around +/- 0.01% accuracy with a zone size of 30), however certain contexts may require greater accuracy or low device pools.

Simulation 3)

Device Pool: Zone Size * Zone Count
Zone Size: Variable
Zone Count: 60
Variance: +/- 0.5%

The purpose of simulation 3 is similar to simulation 2. This time, zone count is fixed and device pool is dependent on zone size. The effects on accuracy is measured.

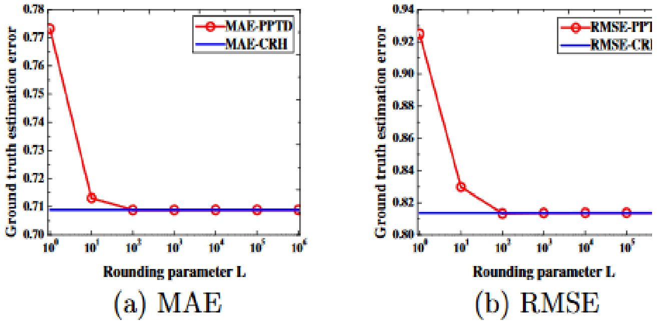


Graph (c): Zone size vs Difference with fixed zone count

The results of simulation 3 provides a near opposite result as simulation 2. Although it can be said that the decrease in difference (increase in accuracy) is a result of increasing the device pool, it is also important to note that this provides a way to mitigate the effects that increasing zone size has on accuracy. If the device pool is large enough, the server can utilize a larger zone size to better protect the privacy of device users, without damaging the integrity and accuracy of the data.

5.3 BTD AND PPTD COMPARISON

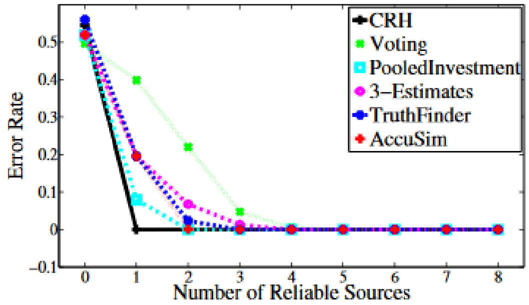
The experiments used to test PPTD [1] and the simulations of BTD, which were previously discussed, are not the same. The results of the two cannot be compared directly. However, inferences can made from the results of the PPTD experiments, the BTD simulations and also the analysis of CRH [9], a crowd sensing framework used within PPTD. Section 5.3 will reference the graphs below.



(a) MAE

(b) RMSE

Graph (d): Ground Truth Estimation Error using PPTD [1].



Graph (e): Error Rate using CRH [9] (Compared with other truth discovery frameworks).

Graph (d) shows a minimum error rate of 0.70-0.71. Regardless of the rounding parameter L [1], when the MAE, or mean of absolute error, method that is used in PPTD, is combined with CRH (Graph (d) blue line) the error is 0.70-0.71. Graph (e) shows, as long as there is at least one reliable source, that CRH (black line) has an error rate of 0. This shows that PPTD, without any of the privacy preserving additives produces an error of approx. 0.71.

Looking back at Section 5.2, our simulations show BTD to have an error of 0.05 during its “worst case”, a high weight variance +/- 5%, scenario while using the enhanced method. However, using a lower weight variance can yield a much lower error of around 0.001.

As stated previously, we cannot directly compare the results of the BTD simulations and the PPTD experiments. The significance of the errors produced by PPTD still gives evidence to the improvement in data accuracy that BTD provides. It is also important to note that BTD also takes steps to ensure data integrity by protecting against users with intent to alter the estimated ground truth.

6. CONCLUSIONS

The balanced truth discovery framework improves upon previous frameworks by satisfying the three demands of crowd sensing systems: preservation of privacy for the device user, data integrity for the data-collection party, and low-cost computation on the user device end. The possibility of modification allows for the framework to be improved and molded to serve a particular context. Simulations running the BTD framework show that accurate data within a few thousandths of a percentage can be achieved whilst preserving the privacy of the device user.

7. REFERENCES

[1] C. Miao, W. Jiang, L. Su, S. Gao, Z. Q. H. Xiao, J. Gao, and K. Ren. Cloud-Enabled Privacy-Preserving Truth Discovery in Crowd Sensing Systems. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (Sensys'15)* 183-196, 2015.

[2] C.-Y. Chow, M. F. Mokbel, and T. He. A privacy-preserving location monitoring system for wireless sensor networks. *IEEE Transactions on Mobile Computing*, (1):94–107, 2010.

[3] F. Ma, Y. Li, Q. Li, M. Qiu, J. Gao, S. Zhi, L. Su, B. Zhao, H. Ji, and J. Han. Faitcrowd: Fine grained truth discovery for crowdsourced data aggregation. *In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(SIGKDD'15)*, 2015.

[4] O. Goldreich. Secure multi-party computation. Manuscript. Preliminary version, 1998

[5] Y. Chon, N. D. Lane, F. Li, H. Cha, and F. Zhao. Automatically characterizing places with opportunistic crowdsensing using smartphones. *In Proceedings of the 2012 ACM Conference on Ubiquitous Computing (UbiComp'12)*, 2012.

[6] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher. Greengps: a participatory sensing fuel-efficient maps application. *In Proceedings of the 8th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys'10)*, 2010.

[7] Q. Li, Y. Li, J. Gao, L. Su, B. Zhao, M. Demirbas, W. Fan, and J. Han. A confidence-aware approach for truth discovery on long-tail data. *Proceedings of the VLDB Endowment*, 8(4):425–436, 2014.

[8] S. Wang, D. Wang, L. Su, L. Kaplan, and T. F. Abdelzaher. Towards cyber-physical systems in social spaces: The data reliability challenge. *In Proceedings of the 35th IEEE International Conference on Real-Time Systems Symposium (RTSS'14)*, 2014.

[9] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. *In Proceedings of the 2014 ACM SIGMOD international conference on Management of data (SIGMOD'14)*, 2014.

[10] O. Goldreich. Secure multi-party computation. Manuscript. Preliminary version, 1998

[11] Y. Chon, N. D. Lane, F. Li, H. Cha, and F. Zhao. Automatically characterizing places with opportunistic crowdsensing using smartphones. *In Proceedings of the 2012 ACM Conference on Ubiquitous Computing (UbiComp'12)*, 2012.

[12] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher. Greengps: a participatory sensing fuel-efficient maps application. *In Proceedings of the 8th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys'10)*, 2010.

[13] Q. Li, Y. Li, J. Gao, L. Su, B. Zhao, M. Demirbas, W. Fan, and J. Han. A confidence-aware approach for truth discovery on long-tail data. *Proceedings of the VLDB Endowment*, 8(4):425–436, 2014.

[14] S. Wang, D. Wang, L. Su, L. Kaplan, and T. F. Abdelzaher. Towards cyber-physical systems in social spaces: The data reliability challenge. *In Proceedings of the 35th IEEE International Conference on Real-Time Systems Symposium (RTSS'14)*, 2014.

[15] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. *In Proceedings of the 2014 ACM SIGMOD international conference on Management of data (SIGMOD'14)*, 2014.