Shell Game: Randomized representative based election to defend against 51% attacks in crowd sensing frameworks

Ajay Reddy Palleri Kesavan

A thesis

submitted in partial fulfillment of the

requirements for the degree of

Master of Science in Computer Science and Software Engineering

University of Washington

2019

Committee:

Brent Lagesse

David Socha

Yang Peng

Program Authorized to Offer Degree:

Computing and Software Systems

University of Washington

Abstract

Shell Game: Randomized representative based election to defend against 51% attacks in crowd sensing frameworks

Ajay Reddy Palleri Kesavan

Chair of the Supervisory Committee:
Associate Professor Brent Lagesse
Computing and Software Systems

Smart devices and wearable have become an epicenter of human lives and have increasingly become more complex and powerful to make people's life easier. Smart devices like smart phones and wearable like a smart watch today are equipped to provide pervasive connectivity, quality communication and a glut of other services made possible by an array of high-grade sensors like ambient light sensor, proximity sensor, barometer and gyroscope to name a few. This unique coupling between sensor technology and human interaction has a potential to offer a multitude of opportunities and applications in mobile crowd sensing paradigm, such as real-time road traffic monitoring, noise pollution, health monitoring etc. In this paradigm, people

become the centerpiece of the sensing process where users can gather data whenever and wherever,

using the mobile sensor devices and they own the process of data retrieval and maintaining of the cleanliness of the data. But humans may be unreliable and malevolent and can affect the cleanliness of the data being collected for their own benefit, which is why mechanisms for detecting and deterring malevolent activities in mobile crowd sensing become imperative than ever. This paper presents a unique and efficient fabric for impeding activities like 51% attack, maintaining the integrity of the data and reduce monetary loss for the data aggregator during such attacks. This has been achieved by implementing a moving target defense in a Randomized representative based election with a proof of stake payment mechanism. To test this method, we simulate an attack by an adversary who gives malicious data and assess their total gain and the percentage of adversary presence needed to obtain a profit.

Contents

List of Figures

1 Introduction

In recent years, there is a proliferation of smartphone usage. These cellphones have a plethora of sensors embedded in them to sense the external world around it. This has fostered the usage of smartphones to act as moving sensors, that can be used to study the external world. This usage of moving sensors on smartphones is called mobile crowd sensing [20]. Crowd sensing has typically been used for obtaining data about the physical world or to study the behavior of people they are augmented to. A crowd sensing framework constitutes of many smartphone users opting-in to provide data for a task called crowd workers [7]. We can find usage of crowdsensing tasks in applications such as T-share [16] and SignalGuru [13].

During the process of data aggregation from smartphone sensors, the crowd worker incurs a cost in the form of smartphone's resources like battery power, communication and computation. To compensate for the cost incurred by the crowd worker they must be incentivized to motivate them to continue participation. If the incentives are lower, then the crowd worker might not be motivated enough to participate in the sensing task. These compensations are in the form of rewards to the crowd worker which can either be monetary [10] and non-monetary [25] incentives. Non-monetary incentives can be either entertainment or service provided by the aggregator, but one major drawback of non-monetary incentives is that they are application specific and is not portable to another task.

On the other hand, monetary compensation can be used as an incentive to crowdsensing. If there is a monetary incentive involved in a task, an adversary can participate and obtain the incentives without performing the

task [26]. Such payments to adversaries are categorized as false payments.

This has given rise to research on defenses against false payments to discourage attackers from exploiting the crowd sensing frameworks [10] [25]. A framework proposed by Kantarci et al. uses Social network-aided collaborative trust scores to model a social network, where each node is interconnected through their common tasks performed. It is used to reduce manipulation probability of a malicious node [12]. Framework proposed by Pouryazdan et al. uses Anchor-Assisted and Vote-Based Trustworthiness as the reputation algorithm, where an anchor node is used to verify ground truth and trusted completely [19]. One of the major draw backs of the previously mentioned systems are that they need external data to evaluate the reputation of a node which may not be available in every crowd sensing task. An alternative to reputation systems to discourage malicious nodes is to use aggregation, where all the nodes out of consensus with the aggregation result are not paid any incentive. This model is vulnerable to a 51% attack [14] [24] where majority of the participants in a crowd sensing task are malicious so they are paid incentives. In 51% attack, the number of malicious nodes needed for a successful attack can be estimated by observing the incentives that are obtained. We propose an improvement to the model where we use Randomized Representative Based Election and Proof of Stake based payments to increase the number of malicious nodes needed to obtain an incentive. We then proceed to simulate our architecture to quantify the total gain of the adversaries and non-adversaries. The primary users of our research are the users of crowd sensing tasks to collect data.

2 Background

Crowdsensing is a technique which refers to the usage of sensors on smartphones to obtain information about the physical world [15]. In simpler terms crowd sensing could be considered analogous to using sensors which are mobile to gather information about the world.

There are two logical layers of crowdsensing framework; data collection service and payment system. Data collection service is responsible for receiving data from smartphones and storing them for further processing. In our framework to compute the incentives of a crowd worker, we use the payment system.

In this section, we provide a brief overview of the components and current research around crowdsensing. We highlight the research work that led to the conception of this framework.

2.1 Crowd worker

Crowd worker or benign crowd worker in our framework is a smartphone user who works by collecting data for the framework. The motivation of the crowd worker participating in the system is to earn incentives from the system. His/Her abilities are limited to collecting and providing observed sensor data.

2.2 Adversary

Adversary or attacker in our framework is a smartphone user like a crowd worker. The motivation of the adversary participating in the system is to

earn incentives from the system without performing the task. The adversary can use strategies like colluding with other adversaries to obtain incentives.

2.3 Crowd Data Collector

For the crowd workers to log their collected data, he/she must buy tokens from the store which in turn is used as stake to attach with the data. Crowd Data Collector takes the data and stake from the Crowd Worker and stores them into a Data storage system(Database) for the Payment service to consume as shown in Figure 3. A Crowd Data Collector is a service that has access to data storage and acts as an internet facing Application Program Interface(API) for crowdsensing entities(smartphones phones) to connect. This acts as a pipeline to receive data from the Crowd Workers(Mobile phones) and insert into the data storage.

2.4 Payment service

Payment service takes in the data provided by the crowd workers and executes randomized representative based election discussed in section 4. It also determines the payment per person based on the data provided and final data that is aggregated. Responsibilities of payment service are as follows;

1. Conducting voting based on the Randomized representative based election

2. Calculating payments after the payment is determined.

2.5 Incentives in Crowdsensing

For crowd workers to spend time, energy and data charges for a crowdsensing task, there must be an incentive given to compensate for the tasks and inspire them to participate in the tasks [18].

2.5.1 Entertainment as an incentive

Incentives of crowdsensing can vary from entertainment to service to monetary incentives. Entertainment as an incentive includes applications which use sensor data to augment interaction [1]. Games like Neat-o-Games [6], Ingress [2] and fitness tracking applications; which track GPS location to augment interactions with the application. Transforming a data collection task into a game makes entertainment the motivation to use the application. One of the drawbacks of such an incentive is that not every application can be gamified, and games may not be reused for all the sensors.

2.5.2 Service as an incentive

Service as an incentive is a model where the incentive that is provided in return for the work performed is a service.

Typically, aggregated information is given as incentive back to the crowd worker. SignalGuru [13], T-Share [16] are two examples of crowdsensing applications which aggregate information and give back information to the crowd worker. SignalGuru gives a platform for crowd workers to sense traffic signals based on speed to give an optimized path for reducing fuel consumption. T-Share is a Taxi sharing application which shares data on taxi users and their routes. It finds common paths for the passengers hence sharing

the taxi and decreasing the passenger's taxi fare. Service as an incentive shares the drawbacks of Entertainment as an incentive because not every sensor data would be valued by the crowd worker. The value of the services provided by the framework to the crowd worker may be subjective.

2.5.3 Monetary incentive

In this case, the crowd data collector who aggregates the data and uses it must pay a certain amount of money to compensate for the battery and communication charges incurred by the crowd worker. Monetary incentives have also been used to promote participation in the crowdsensing system [21]. the monetary incentive has an advantage over the other two incentives that it can be applied to a diverse set of sensing tasks [21] as money can be used as payment for performing the task.

2.6 Attacks inspired by incentives

The motivation for an adversary attacking the crowdsensing system can be for different reasons. An estate agent can profit by contributing forged data with low noise readings around his/her portfolio to drive up the prices [9]. One such motivation is to obtain incentives without performing the crowdsensing task [9]. Incentives obtained by adversaries are called False Payments [19]. The amounts of false payments determine the efficiency of the defense. To decrease the number of false payments, a crowdsensing network can adopt defense strategies. There are two defense strategies adopted to protect against such attacks namely Reputation based methods [19] and Majority based methods [21]. Reputation based methods have a drawback

that they need external data or relations between crowd workers to determine the reputation of a crowd worker. Reputation systems necessitate the maintenance of the external data for an extended period to build a graph or verification. Majority based methods need the aggregation of the data to determine the consensus of the crowdsensing system. Every crowd worker in consensus is paid and every crowd worker out of consensus is punished. A vulnerability of Majority based system is 51% attacks. A crowdsensing network is under 51% attack when an adversary or collusion of adversaries own 51% of the network and influence the consensus of the network to obtain incentives. In a 51% attack, the adversary obtains all the incentives and punishes the benign crowd workers.

For example in an application like SignalGuru[13], which determines the optimal path of a car based on crowd-sourced traffic signal information. An attacker can manipulate the input by providing fake data about signals. When the attacker simulates 51% of the devices connected to the network, he/she can determine the result of the aggregation of the data. The aggregation of data can be used to manipulate the directions used by other users of the application. The motivation behind such an attack can be for a monetary benefit like driving traffic to a road where an attacker has a vested interest in business or to reduce the traffic in the attacker's route for traffic free driving.

When the incentives are monetary in nature, The incentives themselves can act as motivators for attacks. An example of a crowd sensing task for a monetary incentive is obtaining money for mapping Wi-Fi signals. An attacker can perform a 51% attack on the task by simulating devices in a location and injecting false data. This process can be repeated by increasing

the number of simulated devices until the result of aggregation is taken over by the attacker.

This pattern of attack is possible because the effort required to get 51% ownership of the crowdsensing system is known and the adversary can get information on the state of the consensus using the incentives that he/she is receiving. For example, an adversary who owns less than 51% of the system is out of consensus and is not paid any incentive. But with the increase in adversary presence, there will be a tipping point after which the adversary starts receiving the incentives while giving malicious data. This state indicates that the adversary owns a majority in the system. The exposure of this boundary to the adversary in the crowdsensing system is a challenge that is being tackled in this thesis.

2.7 Collusion attacks

Collusion attacks are said to be performed when adversaries obtain more than one entry into the crowdsensing network and work towards a common goal of attacking the system [14]. When the motivation is monetary in nature, the collusion attack's goal is to maximize the monetary gain as an adversary. One such attack is 51% attack [14].

2.8 Attacker model

In our framework address an adversary who is capable of performing a collusion attack. The collusion attack is performed when an adversary communicates the false data to be sent to Crowd Data Collector to other adversaries and they coordinate to send the false data. Figure 1 shows a general image

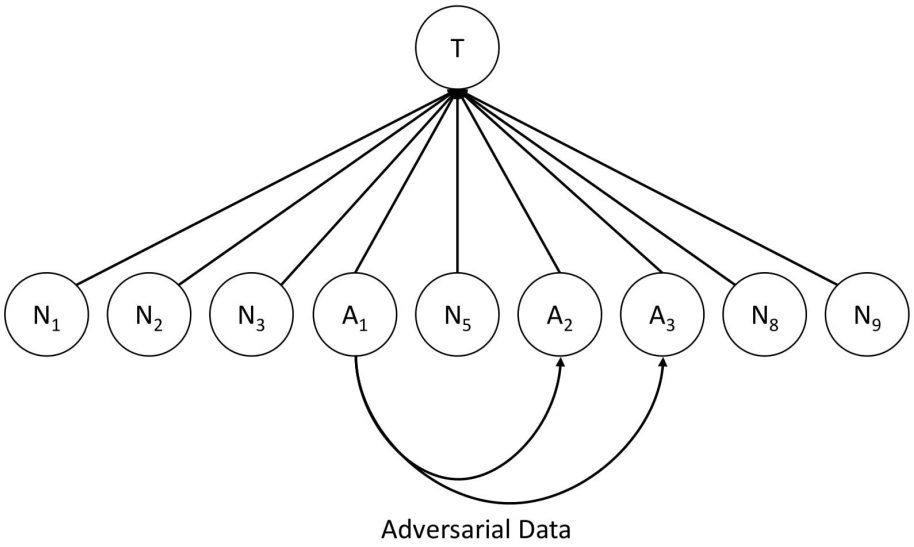

Figure 1: Attacker model where different adversaries collude in an attack.

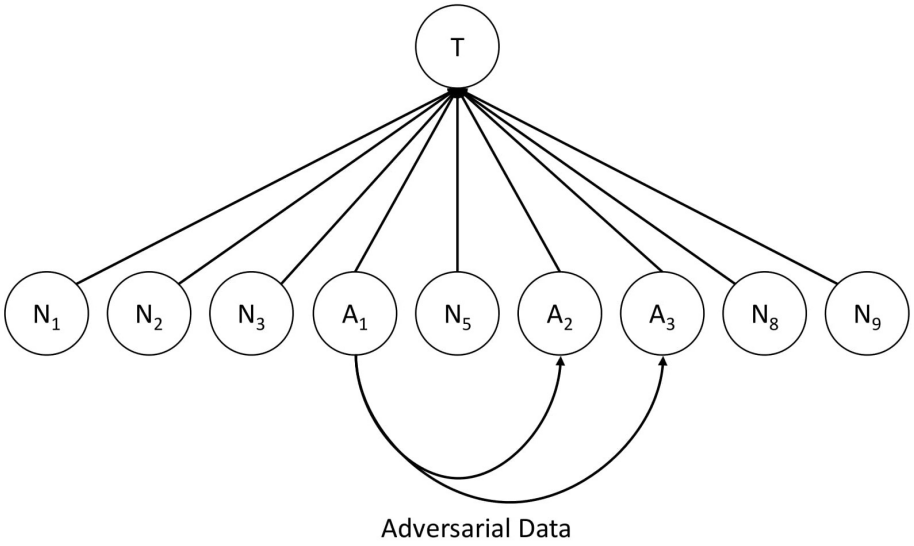

Figure 2: Attacker model where the adversary is only a controller in an attack.

of an attack. Let Node T be the target and AT1 and AT2 be attackers. AT1 and AT2 belong to the crowdsensing network N1,...,Ni. Attacker could be a node participating in the network and colluding with other nodes to exchange information about the false data to be provided.

Attacker could also be a controller AD controlling a set of simulated nodes S1,...Sn where $n <= i$ as shown in Figure 2. We assume the cost of simulation to be zero because we are imposing an additional cost of participation which

is detailed in section 3.2 which is greater than simulation cost.

In Figure 1 the attacker cannot include more adversaries to the network but can collude with them. The cost of the attacker colluding is negligible compared to participation cost as it includes only the communication cost. In Figure 2 the attacker's cost of colluding is negligible as the attacker can increase the number of adversaries. We assume the cost of simulating a new node Sn+1 where $n + 1 <= i$ in the attack is negligible because it involves only creating a new virtual machine and installing the application.

We also limit the capabilities of the adversary to increasing his/her presence in the system. Such a limit of capabilities is needed to restrict the number of variables under study. This limit is also realistic because there is a single point of interaction between the nodes and the crowdsensing framework which is use to insert data.

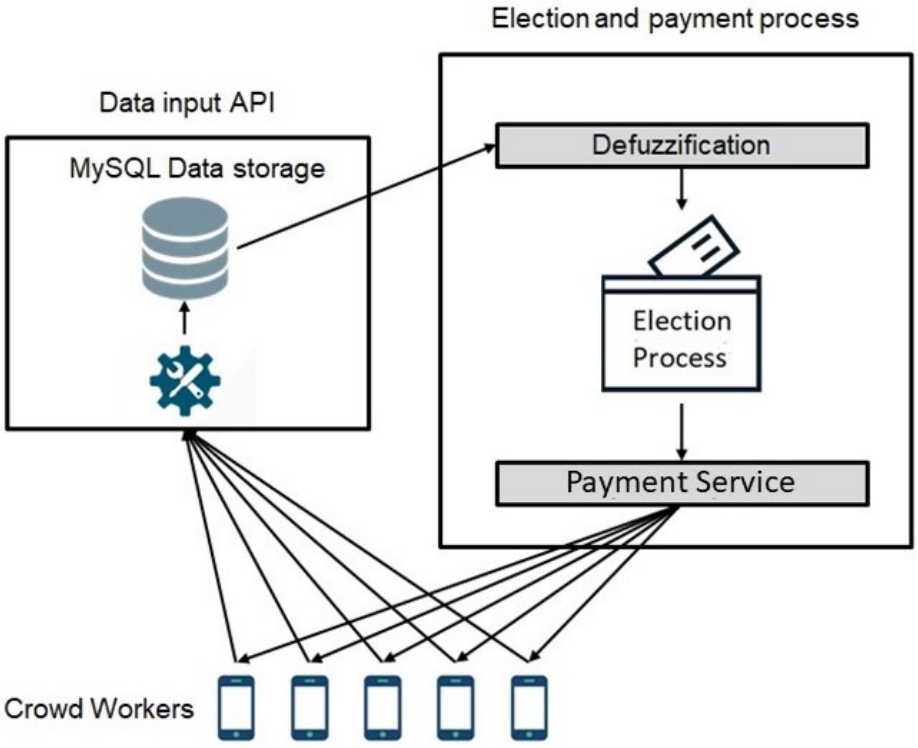

Figure 3: Representative based Moving Target Defense service architecture.

3 Design

In this section, we discuss our proposed architecture, its components and how they interact with each other.

3.1 Moving Target Defense

Moving Target Defense [3] (MTD) is the concept of introducing a degree of uncertainty in the system by moving the target and obscuring the target for the attacker to exploit the target.

MTD has traditionally been used in network security to protect from unauthorized access [5] [17] [11]. It has also been used to protect critical network infrastructure from being compromised. For example, Kai Wang et al.[23] have used moving target defense to perform network address shuffling to increase uncertainty for the attacker. They do so by increasing the scanning space of the attacker through a dynamic domain name method [23]. Green et al.[8] has categorized five properties of a Moving Target Defence namely; Vastness, Uniqueness, Unpredictability, Revocability, and periodicity.

MTD is useful when a system is susceptible to influence by external entities, it focuses on tolerating the influence of an adversary and maintaining the function of the system. Crowdsensing has a similar use-case since it is exposed to any crowd worker without authorization.

3.2 Proof of Stake

Proof of Stake is a payment mechanism that is used in many crypto-currencies as an alternative to Proof of Work [4]. In Proof of Stake, a node uploads the data for verification to a blockchain by attaching crypto-currency tokens as an investment(Stake) to participate in the system. When the data is validated by other participants, the crypto-currency tokens are returned to the node along with an incentive in the form of additional tokens. If the data is invalid, the node not only loses the incentive but also the investment imparting loss of tokens on the node providing invalid data. A similar payment mechanism is adopted in our framework to distribute the incentives and prevent users from providing malicious data. We chose Proof of Stake as the payment method to

penalize the crowd workers providing inconsistent data with a loss of stake. The loss of stake discourages the participation of the crowd worker.

3.2.1 De-fuzzification

Defuzzification is a method used to transform fuzzy input into discrete output classes. We use defuzzification to dampen the variation in sensors and influenced by the physical world. The choice of defuzzification layer can expose other vulnerabilities to influence the class of the data, so It is important to select appropriate defuzzification methods depending on the type of data [22]. This depends on the data that is requested as there are also discrete values that can be requested like the number of cell towers or wi-fi access points at a location where the defuzzification layer is not necessary.

4 Methods

4.1 Randomized representative based election

One of the core components of our election process is having a moving target for the adversary to attack as it increases the uncertainty with which the adversary can get incentives. In this system, we allocate K representatives where K $<$ number of crowd worker for a given crowd task. Every crowd worker needs to be assigned a representative before the election process and this is done in a random and uniform manner as shown in Figure 4. This randomized uniform assignment makes sure that every representative has an equal number of crowd workers. This is done to prevent the attacker from strategizing on the selection of a representative to influence the outcome.

After all the crowd workers are assigned to a representative, an election process is conducted at each representative level where all the data collected is used as votes. These votes are bucketed to find the biggest cluster of data that agree with each other. The bucket that exceeds the second highest bucket's count and wins the election at the representative level with a simple majority. All the crowd workers providing the data point in the winning bucket are given incentives with the return of stake. This election process is performed for every representative.

Introducing random assignment of representatives is to generate a mapping of representative to crowd worker unknown to the adversary. This results in the adversary not being able to obtain a majority with certainty under any representative. All other crowd workers under a representative who did not conform with the majority not only lose the stake but also do not receive any incentive from the system as shown in algorithm 1. When an adversary increases the number of crowd workers performing a crowd sensing task by colluding together or simulating devices, it gives rise to interesting results which are discussed in the results section.

4.2 Implementation

We implemented the API for Crowd Data Collector using PHP programming language version 7.1.15 with MySQL as the choice of data storage. We chose PHP over other programming languages like Django and ASP.NET for the following reasons. The first reason being familiarity with the language and availability of libraries to connect to MySQL. Secondly, our choice of data

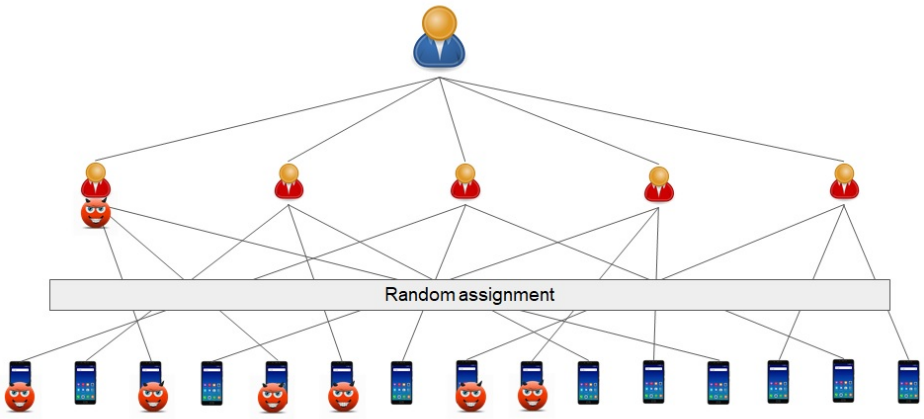

Figure 4: Moving Target Defense Consensus model.

Data: Data from crowd workers sensing device and Stake

Result: Total amount of monetary incentive returned

if Data conforms with the majority in the representative **then**

return Stake + incentive

else

return 0

end if

Algorithm 1: Proof of Stake Algorithm from Server side.

Data: Batch of Data from crowd workers sensing Device

Result: Boolean result on winning of election at the representative level

1: Generate list of K representatives

2: Populate the sensor data and the Crowd Worker's Identifier under K representatives choosing them randomly.

3: Conduct an election on the sensor data for each representative to determine the simple majority.

4: Class of the sensor data is determined with defuzzification.

5: The class with the highest count is declared as the winner under each representative.

6: **if** sensor data belongs to the winning class **then**

7: return true

8: **else**

9: return false

10: **end if**

Algorithm 2: Moving Target Defense Voting Algorithm.

base is MySQL, we can accomplish the data collection task by exposing a RESTful API. RESTful API is useful as it takes advantage of HTTP and additional software is not necessary when creating it. Having Restful APIs gives us the flexibility of using different programming languages and frameworks at the client side as the protocol of communication is HTTP. We can simulate many phones sending data using a load test. We use Apache server to host the service which servers the API. For Consensus and Payment Service we use python programming language to calculate representative election results. Python has an active developer community that can help with implementing our prototypes. Python also has libraries that can reduce the work necessary to accomplish the requirement. For the simulation of malicious and benign requests, we chose Visual Studio's load test module because it has an easy to use graphical user interface. The adversarial and benign actions were defined in the form of unit tests. These unit tests were run with a load test configuration tool which gives us the flexibility of defining the total load and percentage of load that is to be issued.

4.3 Simulation

This simulation is our attempt to replicate a real-life scenario with attention to aspects of the system that must be under study. Our motivation for conducting this simulation is to observe the total payout gained by the adversary as a function of the percentage of adversaries in the system. Since there is a random assignment in the election process, we repeat the simulation 100,000 times to get an average income by the adversary at each percentage of ownership of the crowd sensing task. 100,000 repetitions of the simulations were

chosen because repetitions below 100,000 give a wide range of variance in the payout and give non-reproducible results. We note the percentage of adversary presence in the Crowd Sensing network at which the income for the adversary goes from negative to positive.

In our simulation, we populate a total of 10,000 benign entries which have sensor data in consensus with each other which we take as the start state of the system. After each step, we increase the percentage of adversary data and conduct the election process. At end of the election process, we calculate the total amount gained or lost by the adversary. The same process is repeated until the adversary reaches 99% ownership of the system. We could only reach 99% ownership as to reach 100% ownership would imply deleting benign crowd workers data. The adversaries' profit is calculated by the formula

$$AdversaryProfit = TotalNumberofStakesReturnedinElectionProcess$$

$$+ (TotalNumberofIncentivesGainedinElectionProcess * Incentive)$$

$$- TotalInvestmentinTheSystembyTheAdversary \quad (1)$$

For example, if the incentive is 0.1 token total number of incentives gained is 5 tokens which are the total number votes in the adversary majority representatives and total investment in the system by the adversary is 50 tokens,

the total profit earned by the adversary is 5 + 5 * 0.1 - 50 = -44.5.

The simulation has a randomized election process involved. To normalize the probability of the results, repeat the simulation 100,000 times and aggregate them. We plot a graph with average profit gained by the adversary as Y-axis and percentage of the crowdsensing network owned by the adversary as the X-axis.

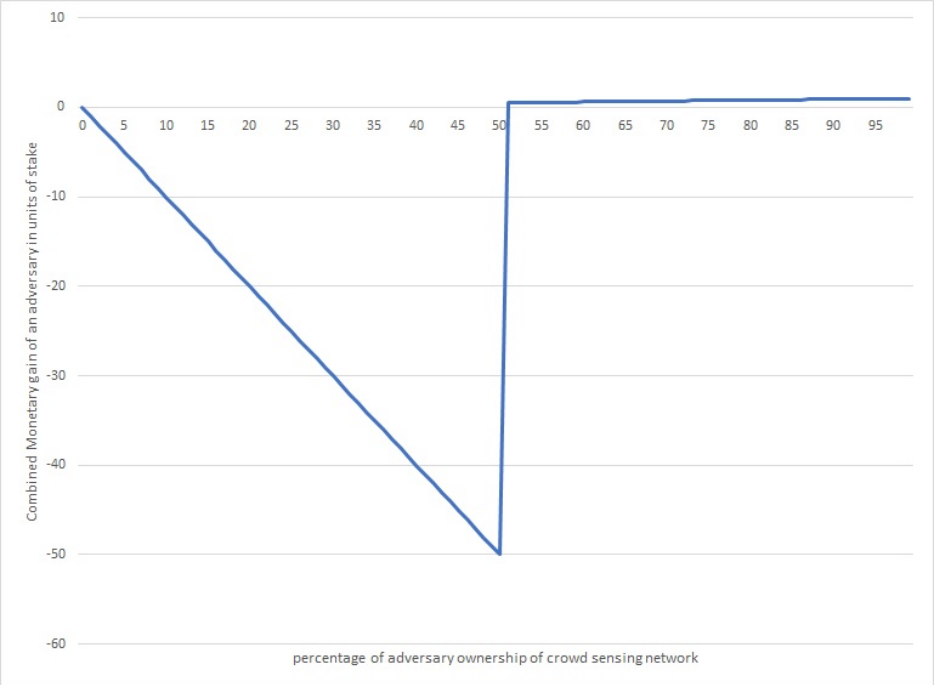

Figure 5: Average adversary's monetary gains observed in a simple election framework with incentive= 1/100 of the stake.

5 Results of the Simulation

In this section, we discuss the results generated by the simulation. Our aim in this section is to showcase the framework and its impact on the adversary's monetary gain.

5.1 Impact of incentives

One of the key components that contribute to adversary's monetary gain is the incentives provided for the task performed. A majority in an election is determined by the percentage of crowd workers in consensus with the data

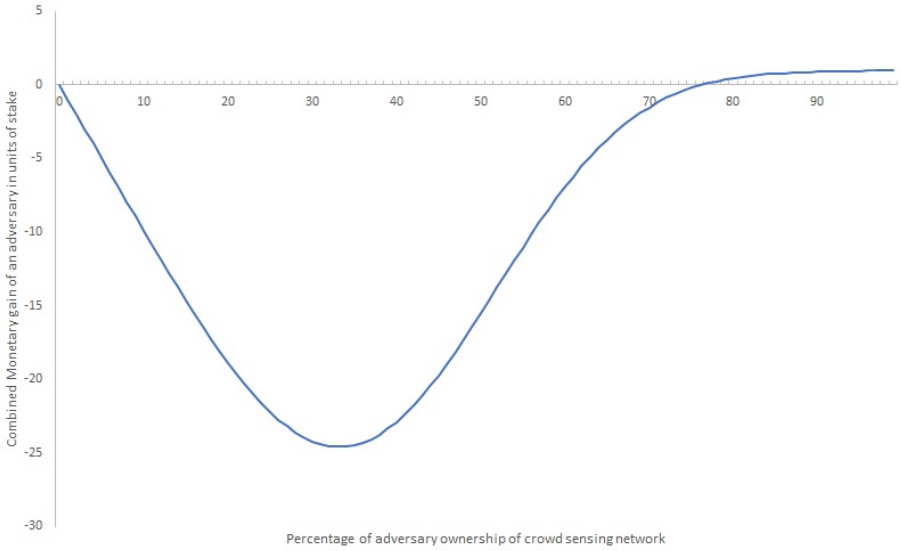

Figure 6: Average adversary's monetary gain observed with a change in adversary ownership (over 100000 tries) with incentive= 1/100 of the stake.

provided. So, we decided to evaluate adversary's monetary gains as a function of the percentage of adversary's presence in the crowd sensing network. This also gives us the flexibility to extend the results of the framework to any number of participants in the system.

We examine the impact of the increase in the percentage of adversary ownership and combined incentive received by the adversary. To examine the effect of this framework, we compare the incentives owned by a benign crowd worker and combined monetary gains of the adversary.

Figure 5 illustrates the monetary gains for an adversary in proof of stake framework. The group with a simple majority under a representative gets the incentive along with the stake and the group that is out of consensus loses the stake. The X-axis is the percentage of adversary ownership in the

crowd sensing network and the Y-axis describes the average monetary gains in units of stake. In Figure 5, we can observe that combined monetary gain of the adversary is negative till 50% and positive after 50%. We observe that the adversary is profitable at 51% ownership of the crowd sensing network because adversary owns the majority and aggregation would detect benign users as out of consensus.

This is also known as the 51% attack, where the adversary can change the consensus by winning a simple majority in the election process thereby receiving the incentive for malicious data that is given.

Figure 6 illustrates the average monetary gains when the ownership of adversary in the crowdsensing network is changed from 0 percent to 99 percent in a Representative based MTD. Our goal was to demonstrate that the break-even point of an adversary in a Representative based MTD is higher than 51%. Figure 6 shows that we observe positive values of incentives only past 77%. This shows that the election process ensures that the total ownership needed to attain the break-even point with 1/100 of the stake as the incentive is higher than 50%. The Representative based MTD is effective in increasing the break-even point.

5.2 Effects of representative based Moving Target Defense on benign crowd workers

Incentives to the benign crowd worker is an important component in inspiring them to participate in the crowd sensing network. We evaluate the change in a benign users incentives with the increase in adversarys presence in the crowd sensing network as illustrated by Figure 7. In Figure 7 the X-

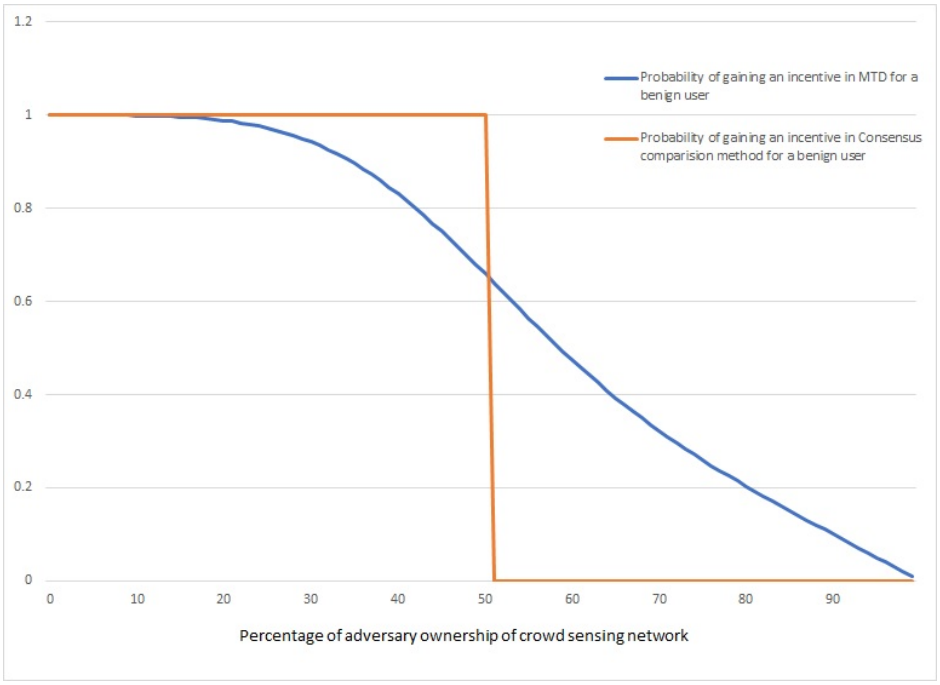

Figure 7: Average benign crowd worker's monetary gain observed with an increase in adversary ownership (over 100000 tries) with incentive= 1/100 of the stake.

axis represents the total percentage of network influenced by the adversary and the Y-axis represents the probability of obtaining an incentive by the benign crowd worker over 100,000 tries. We observe that with an increase in adversary presence, there is a decrease in benign users incentives. The probability of a benign user's incentives approaches 0.5 when the adversary has 58% ownership of the system. In a simple majority based election, the incentive is unchanged until the adversary takes more than 50% ownership of the system, after which the benign user loses all elections and consequently the incentives as well.

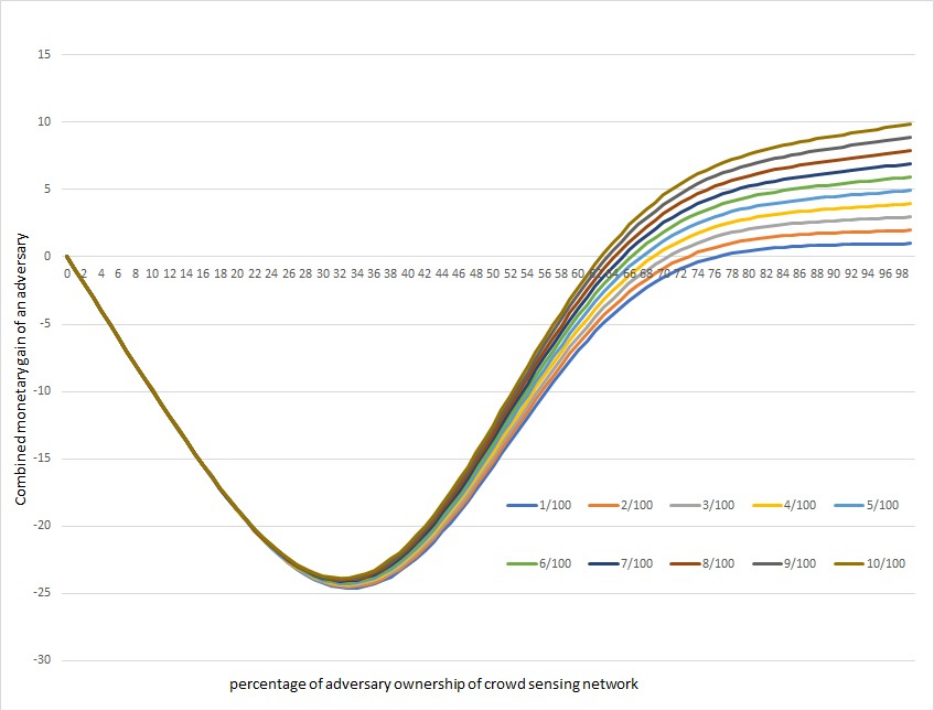

Figure 8: Average adversary's monetary gain observed with an increase in adversary ownership with each line depicting different fractions of buy in.

Hence, we can observe that the break-even point of the benign crowd worker is increased from 50% to 58% proving that the adaptation of the framework is not detrimental to the benign crowd workers.

5.3 Effects of incentive on break-even point

We evaluate the change in break-even point for the adversary with the change in the incentive, as the total income of an adversary is a function of the incentive. This enables us to establish a relation between the percentage of ownership of the adversaries and the incentive. For example, an adversary

buys in X dollars and incentive I, which is a fraction of the stake. We evaluate the break-even point for I = 1/100 of stake till 10/100 of the stake, with increments of 1/100 of the stake and 100 representatives to observe changes in the break-even point.

Figure 8 illustrates the relationship between the break-even point and percentage of adversary ownership of the crowdsensing network. Y-axis represents the combined monetary gain of an adversary and X-axis represents the percentage of adversary ownership. We observe that the incentive and break-even point have an inverse relation, as increasing the amount of incentive decreases the amount of ownership necessary by the adversary to break-even. We increase the incentive until the break-even point is equal to 50% of adversary presence. This gives us the maximum incentive that can be provided while outperforming the simple majority based system. As demonstrated in Figure 9 we observe that when the incentive is 50/100 of the stake the break-even point is 50%. Hence, to outperform simple majority based system the incentive must be less than 50/100 of the stake.

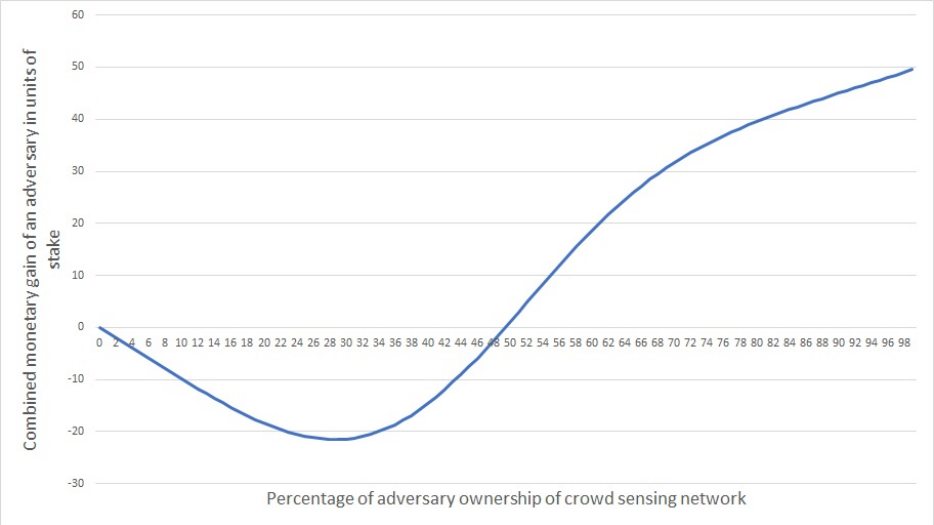

Figure 9: the maximum amount of incentive given to obtain 50% break-even point with incentive= 50/100 of the stake.

6 Discussion

6.1 Application of randomized representative based election in a crowd sensing framework

This framework is useful in participatory sensing with monetary payments. A typical example of a crowd sensing task is, mapping the open Wi-Fi connections in a location. In this task, a user is needed to go to a specified location to scan the total number of open Wi-Fi connections. In a traditional crowd sensing framework number of Wi-Fi connections are collected and the data is uploaded in return of an incentive. The incentive provided could encourage the adversary to poison the data and obtain incentives without traveling to the location or collecting Wi-Fi connection data. To avoid

the adversary from exploiting the system we propose to use proof of stake coupled with the randomized representative based election. When proof of stake is the payment mechanism a participant in the crowd sensing network has to couple Wi-Fi connection data with a stake. When the randomized representative based election is conducted and an adversary loses the election he/she also loses the stake. As represented in section 5 the implementation of the randomized representative election in this crowd sensing task can increase the effort needed for an attack to be profitable with high probability. This makes it unprofitable for the adversary to poison the data by increasing the number of adversaries giving a fake list of Wi-Fi signals and exploiting the framework for incentives. This not only decreases the number of false payments but also decreases the amount of fake data in the system. It is also hard for the adversary to strategize on influencing the election process as the randomized representative based election creates a moving target for the adversary as explained in section 6.2

6.2 Moving Target Defense

As identified by [8], our architecture provides moving target defense with the following characteristics:

- Randomized target: With the implementation of Randomized representative based election, the adversary cannot optimize on the number of nodes needed to gain a profit in the crowd sensing network. The crowd workers are randomly assigned to representatives, this randomizes the target as the target is to obtain a majority under a representative. Since representative chosen for a node is unpredictable it becomes a moving

target and ensures that the target is moved in a random manner. This property holds good for benign entities as well, but in our crowd sensing framework, benign entities are not strategizing on winning the election process.

- Periodicity: Since the randomization is conducted in every election, The target is moved regularly to ensure any data collected by the adversary about the previous election process is unusable for the current election.

- Vastness: The number of possible configurations a randomized representative based election is conducted. It is the same as placing N nodes under R positions of representatives, where n is the number of nodes and R is the number of representatives participating in the election. This can be calculated by

$$number of configurations = N * R \tag{2}$$

  using the formula 2 we can determine that the number of configurations increases by the O(R) while the number of nodes increases by the O(c). Hence the total number of configurations is a vast space to explore.

- Strategy of the adversary: The variable under control of an adversary is the number of adversaries colluding in the crowd sensing system. For adversaries to increase their total income, they are forced to increase the total number of adversaries in the system. According to Figure 6 and Figure 8 we can observe that the total monetary gain for the adversary collusion is non-negative only when the adversary has 77% ownership of

the crowd sensing network. This is an increase in the total percentage of adversaries necessary from traditional Majority based systems as shown in Figure 5.

6.3 Proof of Stake

We use proof of stake as a payment method because it is difficult to calibrate the cost incurred by the adversary, as there are a variety of means an adversary can perform an attack. For example, a simulated device would have a different cost than that of an actual device. Proof of stake is a viable choice as the user needs to pay the stakes to earn incentives.

6.4 Limitations

In this section, we discuss the limitations of the framework.

6.4.1 Distribution of incentives

Our proposed architecture of using Proof of stake-based payment mechanism to distribute incentives which needs an initial stake from the crowd worker to obtain incentives and payment is made based on the Randomized representative based election at the end of the task. This needs a large amount of investment from the user to gain profits from a task that spans over a long period of time. We consider this an acceptable trade-off as a single larger task can also be divided into multiple smaller tasks with reduced time frame where payments are made at the end of each subtask.

6.4.2 Simulations are only an imitation of real life

Our simulations are simplified imitations of the adversary and crowd worker behaviors. The aim of the simulation is to control the unpredictability of real-life situations and act as a proof of concept. For example, we have simplified the possibility of an adversary taking over the crowd sensing task in the absence or meager presence of benign crowd workers. Such simplification might not be reflective of real-life situations considering benign crowd workers are also involved in the payment process. If a benign crowd worker is a rational player in the game, the crowd data might be influenced by the behavior of the crowd worker and their preference of tasks.

6.4.3 De-fuzzification vulnerabilities

Since our framework needs discrete classes of data for voting, it is a good solution for sensor values like the number of active Wi-Fi connections. When sensors give values associated with physical world it might vary from device to device. To standardize the values, we use a defuzzification layer. A defuzzification layer can misclassify the values when adversary's data is close to the benign data. This misclassification can be utilized to shift the centroid of the defuzzification layer thereby poisoning the data.

7 Future work

An avenue of future work is deploying the Randomized representative based election on a crowd sensing task and observe the actions taken by a benign user in the presence of an adversary. In this thesis, we have focused on the

adversary's perspective of the framework and attacks that they can perform. On the contrary, we can also study the effect of benign users payment in the system when their presence increases in the system from 0% to 99% and how much time the system takes to bounce back to the benign state. Other incentive-based mechanisms can be implemented alongside our framework and it can be compared for benign crowd worker retention. To validate the correctness of data provided by a crowd worker, excess data is required. For example to obtain the location data in addition to GPS coordinates if the crowd worker is asked to take a picture of a landmark nearby as validation. It would be interesting to observe the effects of validated data coupled with our framework as a second layer of difficulty for the adversary to overcome. Another avenue of evaluation would be determining the minimum amount of incentive to be provided to keep the benign crowd workers invested in the system as the incentive and adversaries breakeven point have an inverse relationship. We can study the influence of various variables that were fixed in this framework like stake needed to complete a task, which is defined by the task. In our framework, an adversary or benign crowd worker provide the same amount of stake every time in the system. If the stakes where variable and randomly assigned every time it would make the estimation of total profit for an adversary difficult. It would be interesting to observe the adversary's behavior in such a framework. In the future, We propose to evaluate other clustering algorithms to use as a classifier instead of a defuzzification layer and study the impact on false payments.

8 Conclusion

In this Thesis, we proposed a payment system which conducts Randomized representative based election and receives data with proof of stake. Our architecture provides a moving target by randomizing the representative that a crowd worker can vote for. By dynamically changing the representative every time, we reduce the knowledge an adversary has over a target. This prevents him/her from strategizing. We have also shown that the break-even point can be increased in our framework by decreasing the amount of incentives. Our work is primarily aimed at enabling crowd sensing frameworks to prevent adversaries and decrease false payments. Organizations running large scale crowd sensing tasks are the primary beneficiaries of our architecture. They are also a limitation of our Randomized representative based election method, Crowdsensing networks with a smaller number of crowd workers are still vulnerable to adversarys presence with a minimal number of nodes while having an upper limit to how much payment can be made. Another limit of our work is that we rely on simulated data to prove the results. Deploying a crowd sensing task and observing the behavior of benign crowd worker and adversary could account for the uncertainties of the external world. Thus, while further research is required to build on the accuracy of the results, our simulations indicate that our proposed Randomized representative based election with proof of stake-based payments method offers a significant improvement in increasing the effort needed for an adversary to exploit the crowd sensing system.

References

[1] Catch Pokmon in the Real World with Pokmon GO!: http://www.pokemongo.com/.

[2] Ingress: https://www.ingress.com/.

[3] CSD-MTD, June 2013.

[4] Iddo Bentov, Charles Lee, Alex Mizrahi, and Meni Rosenfeld. Proof of Activity: Extending Bitcoin's Proof of Work via Proof of Stake [Extended Abstract]Y. *SIGMETRICS Perform. Eval. Rev.*, 42(3):34–37, December 2014.

[5] M. Dunlop, S. Groat, W. Urbanski, R. Marchany, and J. Tront. MT6d: A Moving Target IPv6 Defense. In *2011 - MILCOM 2011 Military Communications Conference*, pages 1321–1326, November 2011.

[6] Yuichi Fujiki, Konstantinos Kazakos, Colin Puri, Pradeep Buddharaju, Ioannis Pavlidis, and James Levine. NEAT-o-Games: blending physical activity and fun in the daily routine. *Computers in Entertainment*, 6(2):1, July 2008.

[7] R. K. Ganti, F. Ye, and H. Lei. Mobile crowdsensing: current state and future challenges. *IEEE Communications Magazine*, 49(11):32–39, November 2011.

[8] Marc Green, Douglas C. MacFarland, Doran R. Smestad, and Craig A. Shue. Characterizing Network-Based Moving Target Defenses. In *Pro-*

ceedings of the Second ACM Workshop on Moving Target Defense, MTD '15, pages 31–35, New York, NY, USA, 2015. ACM.

[9] Daojing He, Sammy Chan, and Mohsen Guizani. User privacy and data trustworthiness in mobile crowd sensing. *IEEE Wireless Communications*, 22(1):28–34, February 2015.

[10] L. G. Jaimes, I. J. Vergara-Laurens, and A. Raij. A Survey of Incentive Techniques for Mobile Crowd Sensing. *IEEE Internet of Things Journal*, 2(5):370–380, October 2015.

[11] P. Kampanakis, H. Perros, and T. Beyene. SDN-based solutions for Moving Target Defense network protection. In *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*, pages 1–6, June 2014.

[12] B. Kantarci, P. M. Glasser, and L. Foschini. Crowdsensing with Social Network-Aided Collaborative Trust Scores. In *2015 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, December 2015.

[13] Emmanouil Koukoumidis, Li-Shiuan Peh, and Margaret Rose Martonosi. SignalGuru: Leveraging Mobile Phones for Collaborative Traffic Signal Schedule Advisory. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, MobiSys '11, pages 127–140, New York, NY, USA, 2011. ACM.

[14] Brian Neil Levine, Clay Shields, and N Boris Margolin. A Survey of Solutions to the Sybil Attack. page 11.

[15] H. Ma, D. Zhao, and P. Yuan. Opportunities in mobile crowd sensing. *IEEE Communications Magazine*, 52(8):29–35, August 2014.

[16] S. Ma, Y. Zheng, and O. Wolfson. T-share: A large-scale dynamic taxi ridesharing service. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pages 410–421, April 2013.

[17] Douglas C. MacFarland and Craig A. Shue. The SDN Shuffle: Creating a Moving-Target Defense Using Host-based Software-Defined Networking. In *Proceedings of the Second ACM Workshop on Moving Target Defense*, MTD '15, pages 37–41, New York, NY, USA, 2015. ACM.

[18] Dan Peng, Fan Wu, and Guihai Chen. Pay As How Well You Do: A Quality Based Incentive Mechanism for Crowdsensing. In *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '15, pages 177–186, New York, NY, USA, 2015. ACM.

[19] M. Pouryazdan, B. Kantarci, T. Soyata, L. Foschini, and H. Song. Quantifying User Reputation Scores, Data Trustworthiness, and User Incentives in Mobile Crowd-Sensing. *IEEE Access*, 5:1382–1397, 2017.

[20] Moo-Ryong Ra, Bin Liu, Tom F. La Porta, and Ramesh Govindan. Medusa: a programming framework for crowd-sensing applications. page 337. ACM Press, 2012.

[21] Sasank Reddy, Deborah Estrin, Mark Hansen, and Mani Srivastava. Examining micro-payments for participatory sensing data collections. page 33. ACM Press, 2010.

[22] T. A. Runkler. Selection of appropriate defuzzification methods using application specific properties. *IEEE Transactions on Fuzzy Systems*, 5(1):72–79, February 1997.

[23] Kai Wang, Xi Chen, and Yuefei Zhu. Random domain name and address mutation (RDAM) for thwarting reconnaissance attacks. *PLOS ONE*, 12(5):e0177111, May 2017.

[24] J. Xue, Z. Yang, X. Yang, X. Wang, L. Chen, and Y. Dai. VoteTrust: Leveraging friend invitation graph to defend against social network Sybils. In *2013 Proceedings IEEE INFOCOM*, pages 2400–2408, April 2013.

[25] X. Zhang, Z. Yang, W. Sun, Y. Liu, S. Tang, K. Xing, and X. Mao. Incentives for Mobile Crowd Sensing: A Survey. *IEEE Communications Surveys Tutorials*, 18(1):54–67, 2016.

[26] Xinglin Zhang, Zheng Yang, Wei Sun, Yunhao Liu, Shaohua Tang, Kai Xing, and Xufei Mao. Incentives for Mobile Crowd Sensing: A Survey. *IEEE Communications Surveys & Tutorials*, 18(1):54–67, 2016.