This4That: Tamper-proof Incentive Scheme for Community Sensing Systems

Diogo Miguel Pardal Calado
Instituto Superior Técnico
Universidade de Lisboa, Portugal

ABSTRACT

People are increasingly connected to the Internet through their Smartphones and each of these mobile devices has a wide range of sensors that can capture data about the world. The users themselves can be asked short questions about what they see. This *crowdsensing* has the potential to improve the daily lives of people by providing actual data about the environment and the use of services. However, there are significant obstacles to user participation like privacy concerns and resource consumption. But even if these concerns are addressed, as they have been in literature, there is still a need for *trusted incentives* to motivate the users to continue participating.

In this paper, we propose a tamper-proof incentive scheme for a mobile crowdsensing system that supports open sensing, with both automated and manual participation. We implemented a complete prototype of the system with both the server components and a mobile application. The proposed incentive scheme is called *This4That* because it implements a 'tit-for-tat' approach: positive user participation is rewarded with points that are stored in a shared record. This incentive ledger is decentralized and uses a Blockchain technology so that it can be trusted by everyone. The results show that the proposed scheme is practical and can be used to motivate increased participation in innovative crowdsourcing for positive societal impacts.

INTRODUCTION

Some years ago, it was not so easy to get your geographic position or counting the number of steps while jogging because equipments to make the necessary measurements were not available or were too costly. In the last decade, Smartphones have appeared and made all these tasks practical with just one electronic device. Also, the Smartphone is connected to the Internet and this allows sharing the captured data and making it even more useful. Furthermore, the Internet of Things (IoT) [1] trend further to connect the physical world to the digital world. This extended network of sensing devices can provide people with more awareness of the state of the world in a digital form and help them make better decisions in daily life.

Before data collection and sharing can be done by every user, there are *resource consumption* concerns, as the user may worry that too much battery power and network bandwidth may be consumed in the sensing activities. Also, there are *privacy* concerns [2, 3], as the user may refrain from using the system because sharing information from sensors can expose sensitive aspects about her personal life, like where she is and with whom.

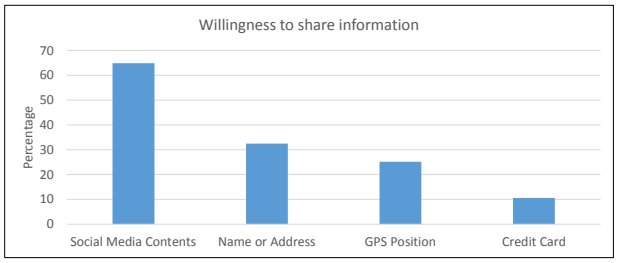

Figure 1. Survey answers regarding the willingness to share data.

End-user survey

To study the importance of both concerns we conducted an end-user survey with a universe of 150 persons, where 35.8% of the respondents had ages between 18 to 25 and 39.1% had ages between 41 to 65.

The first relevant result of the survey is that crowdsensing [4] is desired by end-users: 80% of the respondents said that they are available to participate, and 76% even answered that they would participate without rewards.

This presents a positive outlook as there are many end-users willing to help others, whenever it is possible.

Regarding *resource consumption*, the users stated that it would have to be in the same level as other popular mobile applications, like social networks.

Regarding *privacy*, 55% of the answers said that constant collection of data is a concern, and the majority of the respondents care about the information collected, like GPS positions, and where this information goes.

In Figure 1, we can see that people are not very willing to share sensitive data like, credit card, personal data or data from the Smartphone sensors. However, over 60% of people are willing to share social media content.

Overall, more than 90% of the respondents will only adhere to the system if their privacy is assured. These answers provide strong indication that the lack of privacy will have a huge impact in the user participation, so privacy mechanisms must always be present in this type of system.

The need for incentives

To help overcome both the resource consumption and privacy concerns, there is a need for an *incentive scheme* to motivate users to start using the system, and even more important, to continue using it.

Some of the common incentives for data sharing are [5]:

- Direct-exchange: do something and expect something in return;
- Monetary: money in different formats, like coupons, cash or other forms of electronic money;
- Reputation: earned by positive past behavior [6];
- Gamification: points, medals, trophies, or other game rewards [7].

The above list shows that there are alternatives to monetary incentives, and all of them have to be considered, depending on the targeted user communities [8].

In all cases, incentive records need to trusted by the participants, and the data needs to have good quality [9].

In this paper we propose This4That, an incentive scheme supported by a decentralized and tamper-proof ledger where all the incentive-related information is stored dependably using a Blockchain [10]. The incentives are applied to a crowdsensing system where user communities share data captured by their mobile devices.

Overview

The remainder of the paper is organized as follows. In Section we discuss existing research by presenting existing crowdsensing systems. Section describes the implemented crowdsensing system, and Section details the proposed incentive scheme. The evaluation in shown in Section . Finally, our conclusions about this work are stated in Section .

RELATED WORK

There are two approaches to reporting data from our Smartphones: *Opportunist Sensing* [11], where the user agrees that her Smartphone can be automatically activated on opportunities for data capture and data sharing; and, *Participatory Sensing* [12] where the user has a direct involvement in the data collection activity because the sensing required for the task needs a human observation from the real world. According to Guo [4], the *user participation* can be implicit or explicit by using the opportunistic or participatory sensing, respectively. We argue that both opportunity and user participation are important and should be addressed together.

A typical crowdsensing system has the following workflow, represented in Figure 2: one user creates a task and this task is spread to the other users in the crowd by the system. The users receive the task, execute it, and submit the results back to the system, and receive some kind of reward. The user community is central to the success of a crowdsensing system.

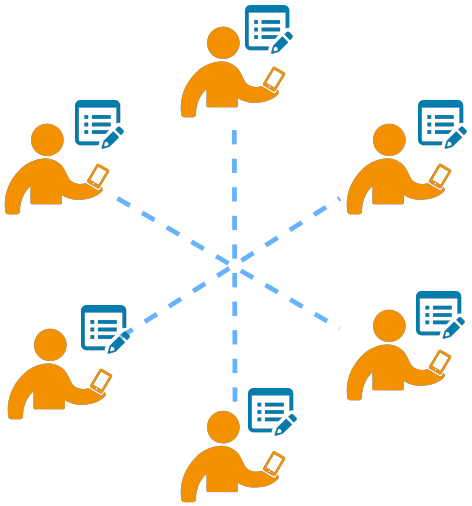

Figure 2. Crowdsensing community model. Each user receives useful information from the community and gives back to it by contributing with sensor readings and answers.

We define the *user* as a person that uses its Internet-connected Smartphone to capture and share information.

We define *community* as a group of people that have a shared goal and that join together to share information related with the goal. The information can be an answer to a question posed by another member of the community or data collected from the sensors of the Smartphone.

The vision for these communities is that they appear from the users' needs, are mostly self-organized and do not have "official" support from governmental or commercial entities. To make this possible, each user must contribute to the sustainability of the community by providing some resources, namely, computational resources that support the incentive ledger, as detailed in Section .

Next, we discuss the main features of two concrete systems that use crowdsensing: *Medusa*, that focuses on tasks and their incentives; and *AnonySense*, which focuses on the privacy of users.

Medusa

Medusa [13] is a programming framework for crowdsensing applications that provides an high-level abstraction to create tasks. Its goal is to simplify the creation and management of crowdsensing tasks by implementing a programming language aimed to the people which are not familiar with programming. The authors illustrated the system behavior using a video task, called *TakeVideo*, which consisted in making a video of a different part of the world. A user writes the task using the high-level language and submits the program to the system. The system creates the task and uses the Amazon Mechanical Turk (AMT)[1]to recruit people to perform any type of tasks and reward users with money when they

complete a certain task. After the users accept the task, they will execute a sensing task, which is this example consists in recording a video clip. Finally the video is sent to the system and AMT is used again to get another set of users to evaluate the best videos based on the requester requirements. When this step is concluded, the requester is notified.

The most important contributions from Medusa are: the work-flow to create and process the crowdsensing tasks; the use of AMT payments as incentive to motivate the crowd; and the data quality procedures, resorting again to AMT.

AnonySense

As we have seen in our own survey (Section), ensuring privacy in mobile crowdsensing tasks is crucial to motivate users to share information. AnonySense [14] is a framework that provides security and privacy in mobile crowdsensing tasks. Its main goal is to preserve user privacy in the entire process, such as in the task execution, distribution and in the report submission. The authors developed a task language, AnonyTL, to specify the behavior of the task, a set of acceptance conditions to execute the task, report statements and termination conditions. To provide *anonymity*, the authors used a Mix Network [15] where there is a chain of proxy servers between the mobile devices and the system that can shuffle the messages and make it very hard for the system to discover who is the sender.

To provide a form of *authentication*, the authors used a group signature [16]. These signatures can be made by a group of people and allow the system to authenticate a member of a group without knowing exactly who she is, just that she belongs to the group.

The most important contribution of AnonySense is the use of techniques to preserve the identity of the users when they receive the tasks and when they report the results.

CROWDSENSING PLATFORM

For implementing the incentive scheme we needed a crowdsensing platform that had efficient resource use and privacy protection. Since there was no system available for use with both characteristics, we did an implementation addressing both *task management*, based on Medusa, and *privacy protection*, based on AnonySense.

The goal of this implementation was to build a baseline system, so that we could then develop the incentive scheme on top of it.

Sensing task management

For the management of sensing tasks, we propose a system architecture with six main modules, represented in Figure 3. The *Task Creator*, the *Task Distributor*, and the *Repository* modules are inspired in Medusa to allow the creation of crowdsensing tasks. The *Incentive Engine* is needed to keep track of user actions and their respective rewards. The *Task Distributor* and the *Report Aggregator* modules are inspired in AnonySense and apply privacy techniques while distributing tasks and collecting reports from users. The *API* is an entry-point responsible for receiving the requests from the users and routing them to the destination.

The *Task Creator* is the node that receives the tasks specification and creates this entity in the system and applies rules, if necessary. It accepts two types of tasks:

- Sensing Task - specifies a sensor to be used in a given GPS position (opportunistic sensing);
- Interactive Task - specifies a question and a set of possible answers (participatory sensing).

The task record contains: a name, a topic that refers a set of tasks for a given subject, an expiration date, and, if it is a sensing task, the sensor to be used; or if it is an interactive task, the question and the set of possible answers.

The *Task Distributor* distributes the tasks to the registered users. The users can subscribe tasks by topic name which is specified along with the task specification and more tasks can be created in the same topic. This provides a way for the users to search tasks. The *Report Aggregator* module collects the reports and can apply to them. The *Repository* will store all the entities like the users, tasks and reports. Finally, the *Incentive Engine* keeps track of the users' contributions in an incentive ledger, described in detail in Section .

Privacy protection

Providing sensor data without any protection can expose sensitive personal information. For example, answering an interactive task about a physical space may indicate who is the user that is at the place. To provide a basic privacy protection in the platform, a *pseudonyms* mechanism [2] was added to protect anonymity by replacing the real names with different values.

Pseudonyms are not enough to avoid a correlation between the data and the user that reported this data [2] because the IP (Internet Protocol) address that came along with the request is not masked and can reveal the origin. In AnonySense the authors adopted a *Mix Network* to cover the origin IP address. In our implementation we opt for the same approach using an external service that provides a random proxy to every request, when needed. In this way the IP address that reaches the platform will always change, even if the user is the same.

Additionally, asymmetric keys will be used to do *Group Signatures*, just like in AnonySense. They provide a way to authenticate a user as part of a community without disclosing who the particular user is.

These functionalities - Mix Network and Group Signatures - are planned but not implemented in the current

[1]https://www.mturk.com/

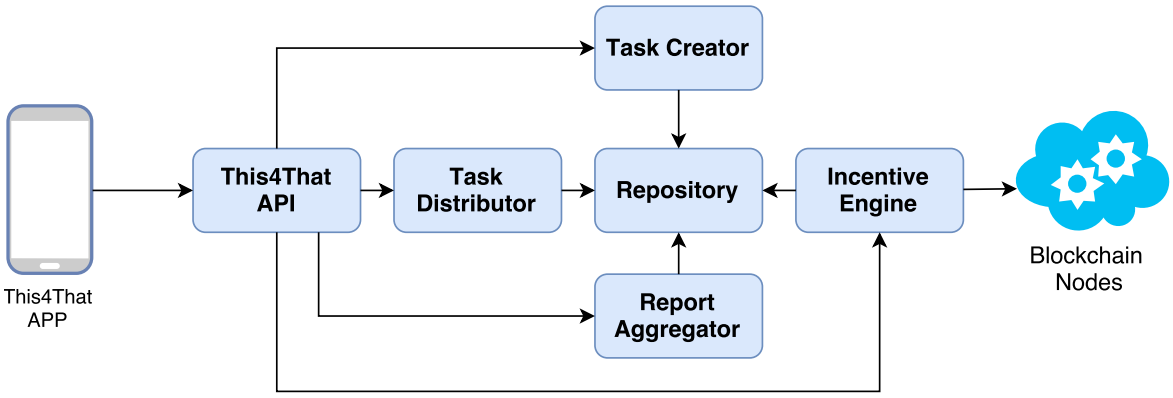

Figure 3. Crowdsensing architecture modules.

prototype. They are not essential for the testing and evaluation of the incentive scheme, the main contribution of this paper.

INCENTIVE SCHEME

The secure incentive scheme is intended to motivate users to keep contributing to the system with new data. If the users do not feel rewarded for their actions, they will eventually stop providing data [17].

As we want a system supported by the community, we want to empower the users to manage it themselves by enabling them to manage the incentive transactions. This is in contrast with Medusa [13] that uses Amazon Mechanical Turk (AMT) to reward the users with money, where all the power of decision to transfer the incentives relies on the provider (Amazon, in this case).

In our prototype we chose an incentive scheme based on *gamification*. The users receive *points* by answering task requests and spend points by creating tasks. This approach has been shown to motivate users in positive ways even in non-gaming environments [7]. Regardless of this option, our system was developed to allow the use of other types of incentives in future versions.

Ledger

The incentive ledger should be tamper-proof, meaning that there are integrity mechanisms in place to allow the records to be trusted by all users. As this is a community environment, we do not want a central authority or one person to be required to check if the incentives transactions are true or if one user is trying to cheat the system by changing the incentives transactions. If the Incentive Engine ordered a transaction to a certain user, every user on the network must agree with it in order to deliver the incentive. To do that, we used a Blockchain technology [10]. It is a public ledger distributed for all the participants where they can check the integrity of all the transactions made in the past. This4That was integrated with a variant called Multichain[2], which allows to build a private blockchain and allows to transfer not only money but also *assets*. The assets are something to store in the blockchain that has a name and quantity and they can be used to implement a gamification incentive scheme where the reward is not money, but points or trophies. This type of blockchain aims to ensure that everyone participates in the mining process and can contribute to the blockchain. The privacy is ensured by using pseudonyms like in the Bitcoin system, they used the public key cryptography where the public key is the user address and private key is used to sign contents. Besides that, to improve the level of privacy they developed a private blockchain, where blockchain content are only visible by the its users. In the Bitcoin system, as this is an open blockchain and there is no control on its users, the mining process is a race between all participants and the user with more power is the one that generates the next block.

The validation process of the transaction in Multichain is quietly different, they still use the concept of miners as Bitcoin, which are nodes in the network that provide their resources to validate the transactions and receive something in return for their work, but in Multichain is not the more powerful node who validates the block.

$$interval = miners \times mining\_diversity \quad (1)$$

In the Multichain, they applied the concept of *mining diversity*, which is a percentage of users to validate adjacent blocks without repetitions. For instance, using the equation 1 to determine the interval that an user has to wait to validate the next block. If the mining diversity is 0,5 and the number of miners are 10 the interval to be able to validate the next block is 5, so an user has to wait 5 blocks to generate another one. This enforces a rule to create blocks using a rotation between the users to avoid the monopoly of mining blocks and to improve the overall fairness of the blockchain.

When the users register themselves in the community they are asked to create a blockchain node in their computer and to create a connection to the community Multichain. At this step, an address to the user needs to

[2]https://www.multichain.com/

be provided. This step will authorize the node to participate in the Multichain network and contribute with user resources to sustain the incentive validation process. This process of validation has a computational cost depending on the security needs. The cost can be adjusted in order e.g. by requiring the hash result to have less or more zeros. The computational cost should be adequately matched to the value handled by the system. If necessary, a user can contribute more to support this process by adding more machines as miners and this contribution is reflected in the reward value.

Data Quality

The incentives scheme should include data quality procedures [9] to handle outlier values, so that they do not exceedingly bias the incentives. A modified Z-Score method is used to reward users according to the quality of their answers.

The quality of data is an important aspect of crowdsourcing solution because when there are incentives to be distributed, some users just want to execute the task to receive the incentive and do not care about the quality of the reported information. Submitting wrong information will introduce noise in the data samples and make it difficult to understand among the collective users what is the right information. To mitigate this problem, we used a statistical method to distinguish the *outliers* from the majority of the reports. In our example we do not want use the mean as parameter to calculate the Z-Score of each observation because the outliers will influence this value a lot, so we used the modified Z-Score which is more robust and uses the median as parameter to avoid the value oscillations that occur in the normal Z-Score. Iglewicz and Hoaglin [18] proposed the *Z-Score modified* to identify outliers. This method is different from the well-known method *Z-Score* which tells how many standard deviations an observation it is from the mean. This calculation will allow the system to infer how much an answer is away from the typical answer.

$$MAD = median(|X_i - median(X)|) \tag{2}$$

$$Z_i = \frac{0.6745(X_i - median(X))}{MAD} \tag{3}$$

Equation 2 refers to the Median Absolute Deviation (MAD) which calculates the central value for each observation deviation. Equation 3 calculates the modified Z-Score for each i-value, which will allow us to understand how much this value is deviated from the tendency.

$$k = \left| \frac{M_i}{Max(M_i)} \right| \tag{4}$$

$$FScore_i = TaskScore - k \times \frac{TaskScore}{2} \ , k \in [0, 1] \tag{5}$$

Equations 4 and 5 are designed specifically for the task reward calculation. Equation 4 is the impact that the Z-Score answer has compared with the other Z-Scores using the median of i responses (Mi) and the maximum of i responses (Max(Mi)), which is a value that goes from 0 to 1. Finally, Equation 5 calculates the final score to assign to each user which will take into account the weight of their response compared with the community answers. This method is expected to penalize the outlier users and to encourage them to share more accurate information in the future.

In Section we evaluate this method in a concrete scenario and compare it with other statistical methods.

Prototype

We deployed the server side which contains the modules presented in Section in a web-server that is part of the platform infra-structure. The web-server contains an REST API to be invoked from the client application in order to communicate and forward each request to the respective module. The client application was developed in Android, but since the back-end API is cross-platform, it allows integration with other programming languages and operating systems. Every user that joins or creates a community must have a Multichain node running in her computer and in order to get access to the community blockchain she must have its Multichain address to the Multichain master node. After these steps, she is able to create tasks and participate in other tasks. Figure 4 represents the interactions between the platform and the users, including: participate or create a community, create a task, report the results and get rewarded.

As described in Section , the incentive scheme is ready to accept different types of incentives, but as an example we could use the gamification where users receive points for completing tasks and spends points by creating tasks. Each user at registering phase will receive 1000 points, by creating a task will spend 100 points and by answering the community tasks will receive 50 points. In task creation the user creates a task with their specifications and uses her Multichain address to transfer the incentive to the system in order to pay for the task. The same process, but in reverse order, happens when an user is rewarded for her sharing, where the system transfers an incentive to the user wallet.

EVALUATION

In this Section we present the evaluation of our incentive scheme. First we present a scenario that can be improved by the use of crowdsourcing, once the appropriate incentives are put in place.

Hospital monitoring scenario

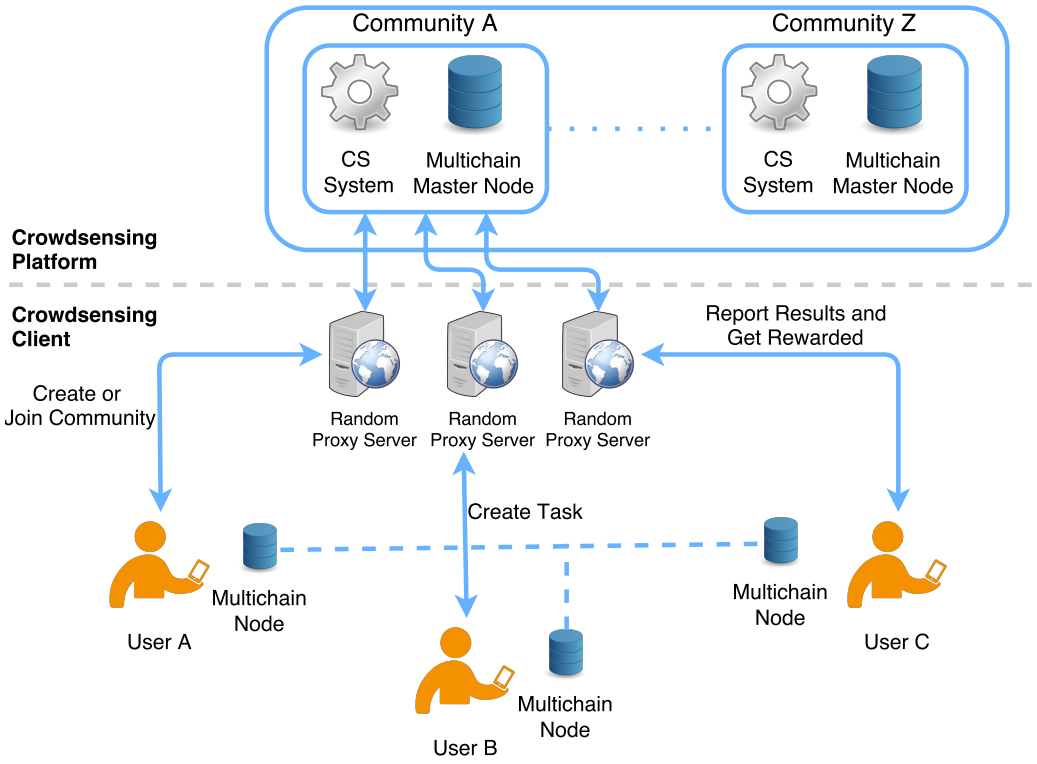

Figure 4. Crowdsensing platform and client interactions.

In the Winter time many people get sick with the flu and usually a trip to medical doctor is needed. The user lives in a city with three hospitals near her house at approximately the same distance and she wants to be seen by a doctor as fast as possible, but she has no way of knowing the wait time in each hospital before choosing one and going there. Of course, the hospitals could publish wait times on-line, but this information is sensitive because it the waiting lines are long it can damage the reputation of the institution.

One alternative solution is for a community of users to organize itself and share data about the hospital wait times. The user finds a hospital monitoring community that shares information about the hospitals around her home, creates the task to ask to the community: *How long is the line at the hospital?* Users present at the hospitals receive prompts for checking the state of the line, and answer back. After receiving the results from the community, the user can make a decision and avoid wasting precious time in a hospital that is completely full instead of going to a hospital with less affluence. Overall the general population benefits of a better use of the health services.

Later, when the user is at the hospital, it is her chance to give back to the community: she will be asked about the state of the waiting line. This information will contribute back to the community. This is a positive 'tit-for-tat' approach – that inspired the name of the system to be This4That – where sometimes you help, other times you receive help from others. For the system to work well, all the participations need to be dependably recorded and the people that contribute with good information should be rewarded.

In the crowdsourcing solution for this scenario we assume there are people interested to know how is state of the hospitals around their homes. A user starts by joining the community and create an interactive task to evaluate how is the urgency queue in a hospital nearby her home. She provides 4 options as possible answers to the task: "Empty"; "Few People"; "Lot of People"; "Overcrowded".

This hospital use case was the example used to obtain results with the prototype implementation.

Results

We started by evaluating the time necessary to execute the main activities like creating tasks and reporting results. Our focus was in the evaluation of the decentralized system where the incentives are stored in the blockchain. A centralized incentives database will be a baseline of comparison.

In the centralized version there is a node containing all the users wallets and the wallet address is the user identification number generated when he registered in the platform. In the decentralized version the user wallet identification is the address generated from Multichain.

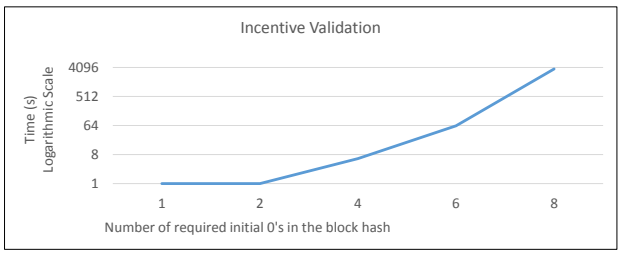

Figure 5. Validation time for a ledger block.

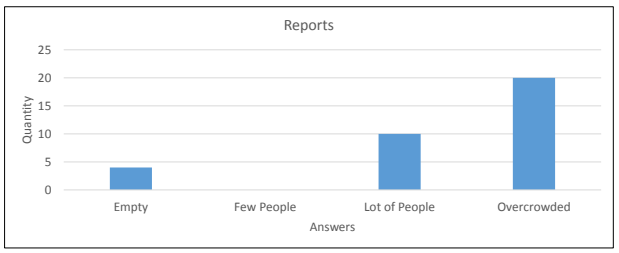

Figure 6. Sample of user answers.

Analyzing the time spent in each module, we can observe a difference of the execution time between the Centralized and Decentralized Incentive Engine. With our system configuration, the Centralized takes 30ms and the Decentralized 60ms. These times include the operations in the Transaction node for the centralized version and the Multichain Node for the decentralized version. This happens because in the centralized version it just checks if the user can make the transaction and records the incentive transaction with simple operations, but in the decentralized version we have to consider the time in the network to the packets can travels from the Incentive Engine to the Multichain node where the transaction will be stored and distributed to the other community nodes. We also have to consider the internal process of Multichain in sending an incentive (which is known by an asset in Multichain) amount from an address to another. When an asset is transferred to another user it generates an transaction and this transaction must be checked.

This process in Multichain, as stated in Section , is different from the *proof-of-work* [10] used in Bitcoin where all the miners try to find the next block by performing a heavy computational work. In Multichain mining diversity is used instead. This mining diversity avoids the monopoly in the validation process but if the same user with a lot of computational power has different Multichain accounts, the mining diversity is not enough, because the user can generates different blocks, change its content and still respect the ordering.

In blockchain the blocks identification is calculated by hashing the block content and a nonce and this will be incremented at the hashing result satisfies a required number of zeros. Observing Figure 5, the time to create a block is exponential and for block with 8 initial zeros it takes at least, in these evaluation conditions, 3680 seconds to find the block hash or generate a new block.

The Multichain provides a configuration file that can be changed in order to adjust the difficulty of creating a block, so depending on the incentive, the effort to protect the incentives must be adjusted.

A break even point must be achieved and the users must receive their incentives in an acceptable time because if the block that contains transactions takes too long to be generated, the users cannot receive the incentives and spend them until they are valid. On the other hand the difficulty level to mine a block is used to discourage malicious nodes that want to modify the chain in order to change previous transactions and to claim the rewards that were modified because if we change a block, all the next blocks must be generated again. So, as stated before, a break even point must be identified in order to balance the time and the security needed. Analyzing these results we can see that is possible a community to manage the incentives transactions between themselves.

Data quality

In order to evaluate the quality of data reported by the users we used the modified Z-Score. We compare this approach with other statistical methods to compare and demonstrate why the modified Z-Score is more appropriate to this scenario.

In Figure 6 are the trial answers reported by the users to the interactive task about the urgency queue in the hospital. 15 people reported “Empty”, 1 reported “Few People”, 10 reported a “Lot of People” and 20 reported that the hospital was “overcrowded”. Analyzing the data something is strange because there are 16 persons, in the same time window, saying the queue had few people or was empty.

We will start by assessing this data with the Interquartile Range (IQR) [19] method to detect outliers. This method proposes that an outlier must be out of the range:

$$\left[Q_1 - 1.5(Q_3 - Q_1), Q_3 + 1.5(Q_3 - Q_1)\right] \qquad (6)$$

The Q1 is the first quartile which represents the middle number between the median and the lowest number in our sample, the Q3 represents the middle number between the median and the highest number. Our answers, “empty”, “few people”, “lot of people” and “overcrowded” are mapped to answer 1, 2, 3 and 4 respectively. The interval to discover outliers based on method defined above is [1.5; 5.5]. This result shows this method is not the most appropriate to evaluate and discover the outliers in our system because the interval of acceptable answers is too large.

The next method uses the Z-Score which tells how many standard deviations an answer is from the mean. In Figure 7, the blue bars represents the Z-Score value for each answer and as we can see the answer “Lots of People” is

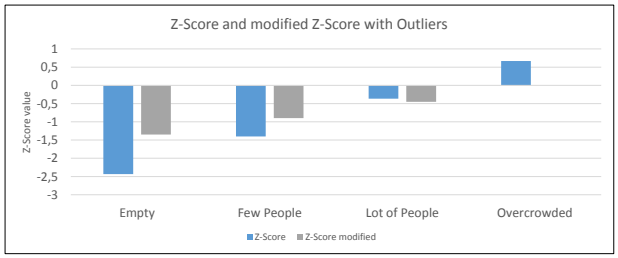

Figure 7. Comparison between Z-Score and modified Z-Score.

Figure 8. Comparison of reward values with Z-Score and with modified Z-Score.

the one that is close to the mean. This indicates that answer is probably the right answer to question, but comparing with the modified Z-Score this shows that option is not correct and the "Overcrowded" answer is the right answer. This divergence happens because the Z-Score uses the mean as the tendency and the modified Z-Score uses the median. The use of the mean in this case can be very influenced by the presence of the outliers and for that reason we chose the modified Z-Score to avoid this disturbance.

Analyzing Figure 8, it shows the points obtained by applying the FinalScore function defined in Section , we can conclude that the majority of users reported "Overcrowded", that for this reason appears likely to be the right answer, using the Z-Score method, are not getting the maximum points because of outliers answers but with the modified Z-Score which are not influenced by the outliers they can reach the maximum points.

So as the reward for contributing are 50 points, the answer with the Z-Score closer to the 0 will receive the maximum points the other answers will receive the points according to the Final Score equation. In this way we can improve the quality of data by making good answers receive good scores.

CONCLUSION

In this paper we presented This4That, a secure incentive scheme to mitigate the problems of user participation and data quality in mobile crowdsensing systems. The developed incentive scheme to reward the users when they share useful data with the community. This scheme relies on blockchain technology to keep a ledger that does not depend on a central authority and that uses the computational resources provided by the community members themselves to ensure the integrity of incentive transactions and the anonymity of users. The data quality is assured by statistical methods that detect outliers answers in crowdsensing tasks.

To evaluate our solution we did response time measurements comparing the centralized approach against the blockchain solution. It reveals that the blockchain will take significantly more time to register the incentives but it offers increased dependability and fault tolerance. However, the absolute time value below 100ms still make it suitable for use in practical applications.

To evaluate the data quality we compared some statistical methods like, IQR, Z-Score and modified Z-Score. Based on our tests, the latter better identifies the potential outliers and does not influence the right answers as the Z-Score influences. This approach proved effective to detect outliers and adjust the granted rewards in those cases.

Future Work

This4That maybe improved in several ways. In the context of the mobile application, the graphic interface should be developed beyond the basic interaction to create tasks and reports results.

Regarding the crowdsensing system, we think that an integration with external services like Facebook or other social networks may allow us to get more accurate information. In this way, we can collect information not only from the users participating in the crowdsensing tasks and we can compare the results provided by the community and the information provided by a external service.

In our solution we still have entities, like users, tasks and results being stored in the server that is supporting the Multichain MasterNode. We think that in a future work, we want to move all these entities to the blockchain and use it not only to store the incentive transactions.

The incentive scheme at the moment uses the blockchain as an open ledger, where everyone is authorized to enter in and this will obligate to use the proof-of-work (PoW) in order to secure the incentives transactions. Our first idea in the beginning of this dissertation was to use a mobile blockchain, but at the moment there is no such implementation to mobile blockchains due to the smartphone's resources consumption to complete the PoW. So, in order to avoid the users having their own computers contributing to the blockchain as implemented in this first version, we want to go further and use the smartphones to do that. This can be possible, by using the blockchain as permissioned ledger, where there is no PoW to do and use of the smartphone resources is reasonable. But the rules of the community model have to change, because a permissioned ledger works based on a circle of trust, so who creates the community has to trust in the people in its circle. It can arise privacy concerns because if we want to maintain the user anonymization, the users should be authenticated in a way, maybe using

the *group signatures*, without exposing their individual identity. Based on these principles we can build *trusted communities*, which does not require to use a personal computer to support an activity that is mainly done on the smartphone.

REFERENCES

1. J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013.
2. D. Christin, A. Reinhardt, S. S. Kanhere, and M. Hollick, "A survey on privacy in mobile participatory sensing applications," *Journal of Systems and Software*, vol. 84, no. 11, pp. 1928–1946, 2011.
3. R. B. Messaoud, N. Sghaier, M. A. Moussa, and Y. Ghamri-Doudane, "On the privacy-utility tradeoff in participatory sensing systems," in *NCA 2016*, pp. 1–8, 2016.
4. B. Guo, Z. Yu, X. Zhou, and D. Zhang, "From participatory sensing to mobile crowd sensing," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on*, pp. 593–598, IEEE, 2014.
5. D. Bhattacherjee, "A comparative study of the incentive mechanisms for mobile crowdsensing," 2015.
6. B. E. Commerce, A. Jøsang, and R. Ismail, "The beta reputation system," in *In Proceedings of the 15th Bled Electronic Commerce Conference*, Citeseer, 2002.
7. Y. Ueyama, M. Tamai, Y. Arakawa, and K. Yasumoto, "Gamification-based incentive mechanism for participatory sensing," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on*, pp. 98–103, IEEE, 2014.
8. T. W. Malone, R. Laubacher, and C. Dellarocas, "Harnessing crowds: Mapping the genome of collective intelligence," 2009.
9. Y. Zhao and Q. Zhu, "Evaluation on crowdsourcing research: Current status and future direction," *Information Systems Frontiers*, vol. 16, no. 3, pp. 417–434, 2014.
10. S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.
11. R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges.," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.
12. J. A. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava, "Participatory sensing," *Center for Embedded Network Sensing*, 2006.
13. M.-R. Ra, B. Liu, T. F. La Porta, and R. Govindan, "Medusa: A programming framework for crowd-sensing applications," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pp. 337–350, ACM, 2012.
14. C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos, "Anonysense: privacy-aware people-centric sensing," in *Proceedings of the 6th international conference on Mobile systems, applications, and services*, pp. 211–224, ACM, 2008.
15. C. A. Ardagna, S. Jajodia, P. Samarati, and A. Stavrou, "Privacy preservation over untrusted mobile networks," in *Privacy in Location-Based Applications*, pp. 84–105, Springer, 2009.
16. D. Chaum and E. Van Heyst, "Group signatures," in *Workshop on the Theory and Application of of Cryptographic Techniques*, pp. 257–265, Springer, 1991.
17. V. Benndorf, H.-T. Normann, *et al.*, *The willingness to sell personal data*. Düsseldorf Institute for Competition Economics (DICE), 2014.
18. B. Iglewicz and D. C. Hoaglin, *How to detect and handle outliers*, vol. 16. Asq Press, 1993.
19. J. W. Tukey, "Exploratory data analysis," 1977.