Tamper-proof Incentive Scheme for Mobile Crowdsensing Systems

Diogo Calado, Miguel L. Pardal
INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Portugal
{diogo.m.calado,miguel.pardal}@tecnico.ulisboa.pt

***Abstract*—People are increasingly connected to the Internet through their smartphones and each of these mobile devices has a wide range of sensors. The users themselves can be asked short questions about what they see. This *crowdsensing* has the potential to improve our daily lives by providing actual data about the environment and the use of services. However, there are significant obstacles to user participation like resource consumption and privacy concerns. There is a need for *incentives* to motivate the users.**

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 prototype of the system with server components and a mobile application. The proposed incentive scheme implements a tit-for-tat approach: positive user participation is rewarded with points that are stored in a shared record. This incentive ledger uses a Blockchain so that it can be trusted by every participant. The evaluation results show that the proposed scheme is practical and can be used to motivate increased participation in crowdsensing.

***Index Terms*—crowdsensing, participatory sensing, incentive mechanism, blockchain, privacy**

I. Introduction

In the last decade, smartphones have made sensing practical and economical with just one electronic device. Their Internet connection allows sharing the captured data and making it even more useful. Furthermore, given the development of the Internet of Things (IoT) [1], the physical world will be increasingly more connected to the digital world. This extended network of sensing devices can provide people with more awareness of the state of the world and has potential to help them make better decisions in daily life.

Data collection and sharing performed by a large number of regular users is called *crowdsensing* [2]. Before it can be more widely done, there are obstacles: resource consumption and privacy concerns. *Resource consumption* [3], as the user may worry that too much battery power and network bandwidth may be consumed in the sensing activities. *Privacy concerns* [4], [5], 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.

A. End-user survey

To get a glimpse of the relative importance of both concerns we conducted an end-user survey with a universe of 150 smartphone users, from Portugal, with mixed technical backgrounds. 35.8% of the respondents had ages between 18 to 25 and 39.1% had ages between 41 to 65.

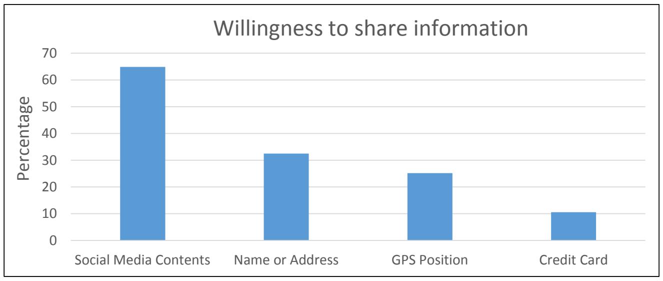

Fig. 1. Survey answers regarding the willingness to share data.

The first relevant result of the survey is that crowdsensing [2] is desired by end-users: 80% of the respondents said that they are available for participation, 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. Even if many users state that incentives are not required, having them will likely boost the use of the system [6].

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 by their devices is a concern, and the majority of the respondents care about the information collected, like GPS positions, and where this information goes. Figure 1 shows the willingness to share different sensitive data, as stated by the respondents. We can see that people are not very willing to share credit card, address and location data from the smartphone's sensors. 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 has a significant impact in the user participation, so protection mechanisms should be present in this type of system. However, over 60% of people are willing to share social media content. These answers indicate that combining crowdsourcing with a social context can influence the willingness of individual users to share data.

978-1-5386-7659-2/18/$31.00 ©2018 IEEE

B. The need for incentives

Having incentives does not directly solve the resource consumption nor the privacy concerns, but it can positively influence the users. If there is a balanced proposition of resource use, privacy protections, and adequate incentives, then the users are more likely to use the system [6]. Some of the common incentives for data sharing are [7]:

- Direct-exchange: do something and expect something else in return;
- Monetary: money in different formats, like coupons, cash or other forms of electronic money like crypto-currencies [8];
- Reputation: earned by having positive behavior in the past [9];
- Gamification: points, medals, trophies, or other game rewards [6], [10].

There is a need for an *incentive scheme* to motivate users to start and continue using the system. The list above shows that there are alternatives to monetary incentives, depending on the targeted user communities [11]. In all cases, incentive records need to be trusted by the participants and the data needs to have quality [12].

C. Overview

In this paper we propose *This4That*, a crowdsensing system where user communities share data captured by their mobile devices. The incentive records are stored in a *tamper-proof* ledger using a Blockchain [8].

The remainder of the paper is organized as follows. In Section II we discuss existing research by presenting existing crowdsensing systems. Section III describes the implemented system, and Section IV details the proposed incentive scheme. The evaluation in shown in Section V and, finally, our conclusions are stated in Section VI.

II. Related Work

There are two approaches to reporting data from smartphones [2]: *opportunist sensing* [13], where the user agrees that her smartphone can be automatically activated on opportunities for data capture and data sharing; and, *participatory sensing* [14] where the user has a direct involvement in the data collection activity because the sensing required for the task may need a human observation of the world. We argue that both opportunity and user participation are important and should be addressed together.

A typical crowdsensing system has the following *work-flow*: 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.

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

A. Medusa

Medusa [15] is a programming framework for crowdsensing applications that provides a 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 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 for entertainment purposes. A Medusa 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 [16]. After the users accept the task, they will execute a sensing task, which in 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 definition of work-flows 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.

B. AnonySense

Ensuring privacy in mobile crowdsensing tasks is crucial to motivate users to share information. AnonySense [17] is a framework that provides security and privacy in mobile crowdsensing tasks. Its main goal is to preserve user privacy in the task execution, distribution and report submission. This system is composed by mobile nodes such as mobile phones, one registration authority, which registers participating nodes and issues certificates to other system components. A task service is accountable for receiving tasks from applications and for distributing the tasks to certain mobiles nodes that satisfy the preconditions defined in the tasks. 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. A report service receives the reports from the users and aggregates them to increase privacy and returns them to the application that previously submitted the sensing task.

Group signatures [18] are used to provide a form of *authentication* through signatures that are made by a group of people and allow the system to authenticate a member of a group without knowing exactly who he is (just that he belongs to the group).

A *mix network* [19] is used to provide *anonymity* through a chain of proxy servers between the mobile devices and the system that shuffles the messages and makes it very hard to discover who is the original sender.

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

The authors also propose a rotation of MAC and IP address to decrease the possibility to associate an MAC address to a device. The application has a certificate issued by the registration authority to communicate with the task and report services, to assure that it is communicating with the right nodes. TS authenticates the MNs as users of the network by using group signature, without revealing their individual identity.

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.

III. Crowdsensing platform

The user community is central to the success of a crowdsensing system, as pointed by the survey results presented in Section I-A.

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. We assume that the users are *both producers and consumers* of information in the same community.

We also scope the user incentives to each community i.e. the incentives given to a user in one community cannot be used by the same user in another community. 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 will have to contribute to the sustainability of the community by providing some resources, namely, computational resources that support the incentive ledger, as detailed in Section IV.

Next we provide an example scenario for crowdsensing, followed by a description of the platform modules.

A. Example scenario

When people get sick they may need to be observed by a medical doctor. Let us consider a user that is feeling sick and lives in a city with three hospitals near her house, at approximately the same distance. The user 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 online, but this information is sensitive because if 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?* The community definition states what is its geographic scope together with a set of classification tags, e.g. 'hospital', 'wait time', to assist in future discovery of relevant communities.

The users that are at the hospitals receive prompts for checking the state of the line, and answer back. After receiving enough results from the community, the user can make a decision and avoid wasting time by going to the less busy hospital. 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, and she will provide information back to the community. Sometimes you help, other times you receive help from others. This positive tit-for-tat approach inspired the name of the system to be *This4That*. For the system to work well, all the participation needs to be dependably recorded and the people that contribute with good information should be rewarded.

B. Architecture

In this section we present the architecture for the task management and privacy protection.

1) Task management: For the management of sensing tasks, we propose a system architecture with six main modules, represented in Figure 2. The *Task Creator*, the *Task Distributor*, and the *Repository* modules are similar to the ones 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 apply privacy techniques while distributing tasks and collecting reports from users, as explained in Section III-B2. 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 and interactive. A sensing task specifies a sensor to be used in a given GPS position (opportunistic sensing). An 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. 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 IV.

2) Privacy protection: Providing sensor data without any protection can expose sensitive personal information. For

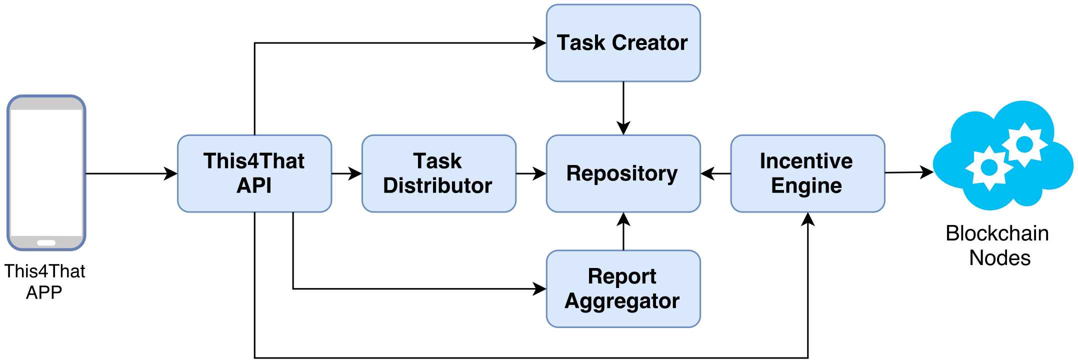

Fig. 2. Crowdsensing architecture modules.

example, answering an interactive task about a physical space may indicate who is the user that is at the place.

A *pseudonyms* mechanism [4] was added to provide a basic privacy protection in the platform: anonymity, i.e., replacing the real names and identifiers with non-related values. Pseudonyms are not enough to absolutely avoid a correlation between the data and the user that reported this data [4] because the IP (Internet Protocol) address that came along with the request is not masked and can reveal the origin. The AnonySense the authors adopted a *mix network* approach 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.

C. Prototype

We implemented a crowdsourcing system addressing both *task management*, based on Medusa, and *privacy protection*, based on AnonySense. The source code is publicly available as described in Section VI-B.

Figure 3 represents the interactions between the platform and the users, including: participate or create a community, create a task, report the results and get rewarded.

We deployed the server which contains the modules in a web server that is part of the platform infra-structure. The web server provides a REST API to the client application in order to communicate and forward each request to the respective module. The incentive scheme runs in the server and coordinates the incentive transactions among the Multichain nodes (as explained in Section IV).

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.

IV. Incentive Management

Incentives are intended to motivate users to keep contributing with new data to the system. If the users do not feel rewarded for their actions, they will eventually stop providing data [20]. As we want a system supported by the community, we want to empower the users to manage the incentive transactions themselves. This is in contrast with Medusa [15] that uses Amazon Mechanical Turk (AMT) [16] to reward the users with money, where all the power of decision to transfer the incentives relies on a central provider (Amazon, in this case).

A. Ledger

The incentive ledger should be *tamper-proof*, meaning that it requires integrity mechanisms to allow its records to be trusted by all users. As this is a community environment, we do not want a central authority to be required to check if the incentives transactions are true or if one particular user is trying to cheat the community by changing the incentives transactions. To do that, we used a distributed ledger between all the participants where they can check the integrity of all the transactions made in the past. To allow this, each user needs to contribute both with computer resources for the ledger – at home or running in a cloud – and with the smartphone resources for the sensing.

Our ledger implementation was integrated with Multichain[2] that allows building private blockchains for assets. The *assets* store points or trophies (instead of money) that are recorded in ledger transactions. The validation process of the transaction is different from *proof-of-work* used in Bitcoin [8], even though there also are miners. Multichain uses the concept of *mining diversity*, which states that there is a percentage of users that is able to validate adjacent blocks without repetitions. For instance, using Equation 1 to determine the interval that a user has to wait to validate the next block.

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

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

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

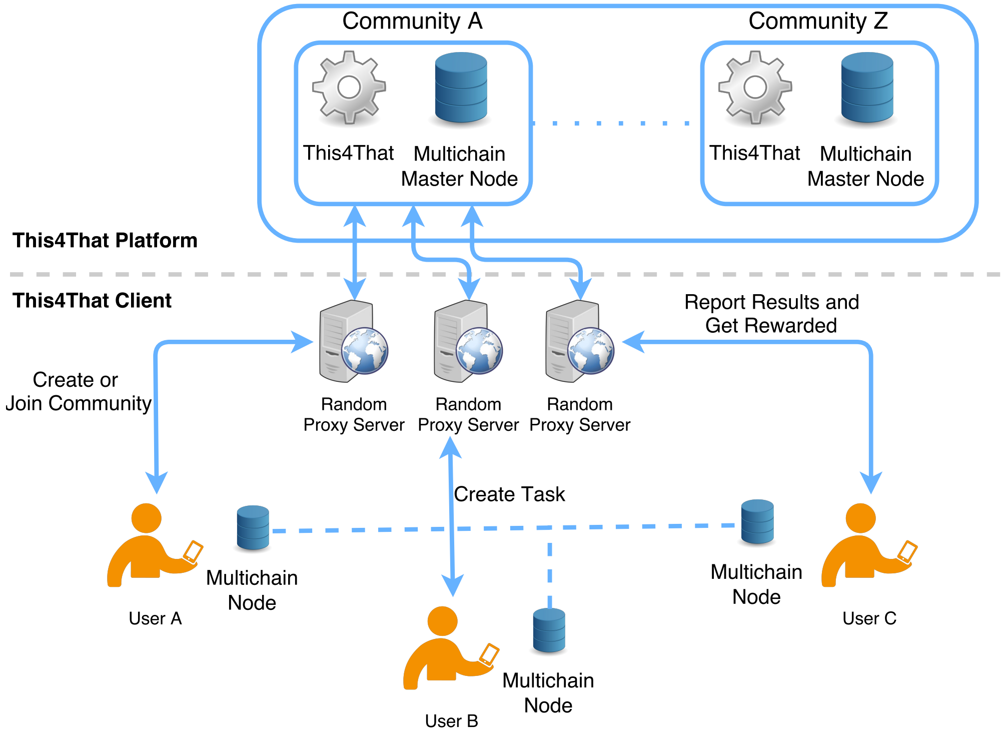

Fig. 3. Crowdsensing platform and client interactions.

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 a user registers herself in a community, she is asked to provide a computer node to participate in the blockchain. This step will authorize the node to participate in the Multichain network and contribute with user resources to sustain the incentive validation process. This validation has a computational cost depending on the security needs. The cost can be adjusted, for example, by requiring the hash result to have less or more zeros, just like in Bitcoin. 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.

The identity of the users is kept private by the use of pseudonyms like in the Bitcoin system, using a asymmetric cryptography key pair: the public key is the user address and private key is used to sign contents.

B. Gamification

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 [10]. Regardless of this option, the data model of our incentive scheme was kept extensible to allow the future implementation of different types of incentives (listed in Section I-B).

In the prototype each user receives 1000 points at the registration phase. A task creation spends 100 points, a task response earns 50 points. At the beginning an asset named *Points* was created in Multichain master node in order to transfer the points to Multichain nodes.

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.

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.

C. Data Quality

The incentives scheme includes data quality procedures [12] to handle outlier values, so that they do not exceedingly bias the incentives. We used a statistical method to distinguish the *outliers* from the majority of the reports.

$$MAD = median(|X_i - median(X)|) \qquad (2)$$

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

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 to calculate the task reward. 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 penalizes the outlier users and encourages them to share more accurate information in the future.

V. Evaluation

In this Section we present the evaluation of the proposed incentive scheme. We analyze performance, to assess the resource consumption, and privacy protections. All the software components ran in independent virtual machines on the same host machine.

A. *Performance*

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 system version where the incentives are stored in the blockchain. A simple incentives database was the baseline for comparison, where the incentive transactions are stored in a single computer node. In the database version there is a computer node containing all the user wallets and the wallet address is the user identification number generated when he registered in the platform. In the blockchain version, the user wallet identification is the address generated by Multichain.

The API module receives the request, the repository saves the new user entity and the incentive engine transfers the incentive to the user wallet. By analyzing the results we could observe a difference of the execution time between the simple database and blockchain incentive engine. With our system configuration, the database approach takes 34 ms and the blockchain 63 ms, as shown in the Table I.

These times include the operations in the transaction node for the database version and the Multichain node for the blockchain version. This happens because in the database version it just checks if the user can make the transaction and records the incentive transaction with simple operations. In the blockchain version, we have also to consider the time spent in the network by the packets 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 mining diversity in Multichain, sending an incentive (asset) amount from one address to another one. When an asset is transferred to another user it generates an transaction and this transaction must be checked.

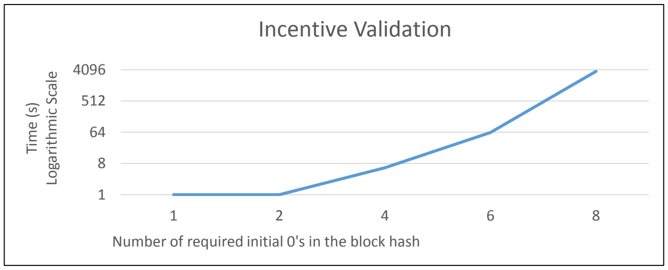

Fig. 4. Validation time for a ledger block.

The identification of the blocks 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. In Figure 4 we can see that the time to create a block is exponential and for a block with 8 initial zeros it takes, at least, 3680 seconds to find the block hash or generate a new block, in these evaluation conditions.

The Multichain allows the difficulty of creating a block to 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 to change previous transactions because if we change a block, all the next blocks must be generated again.

B. *Privacy Protection*

To evaluate the privacy in our incentive scheme we analyzed a possible correlation between the users that are connected to the platform and the transactions performed on Multichain.

We assumed that the crowdsensing platform uses a proxy network to hide the users identity and to do that, we used an API to get random proxies to each request. We deployed the system in our server at Portugal and we analyzed the execution time to each request using different proxy servers around the world and we checked the user IP address that arrives at our server.

Analyzing the results in Table II, we can see that the proxy being used significantly impacts the request performance. The longer the distance to the server, the bigger is the time that it takes to complete the request. The advantage of using proxies is the IP address masking because the proxy server acts as an intermediary and the IP address that reaches the destination server is the proxy IP address. We consider the cost of the protection acceptable given the privacy benefit.

VI. Conclusion

In this paper we presented *This4That*, a crowdsensing system with a cooperative and secure incentive scheme. The developed prototype uses a gamification incentive scheme to

TABLE I
MODULE EXECUTION TIME

Module	Database execution time (ms)	Blockchain execution time (ms)
API	9	9
Repository	13	13
Incentive Engine	34	63

TABLE II
REQUESTS USING DIFFERENT PROXIES

Region / Criteria	Times (s)	Origin IP Address	Destination IP Address
Brazil	1.44	188.140.31.217	189.40.191.95
China	2.06	188.140.31.217	183.222.102.106
Ecuador	2.80	188.140.31.217	181.112.228.126
Kenya	1.86	188.140.31.217	197.232.17.83
Russia	4.35	188.140.31.217	212.192.120.42
Ukraine	0.80	188.140.31.217	95.67.57.54

reward users when they share useful data with the community. This scheme relies on a Multichain blockchain to keep a tamper-proof ledger that uses the computational resources provided by the community members themselves to ensure the integrity of incentive transactions.

To evaluate our solution we did response time measurements comparing the database 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. The absolute time value below 100ms show that it is suitable for use in practical applications. We also analyzed the effectiveness of the privacy protection provided by the use of pseudonyms and proxies, and found it sufficient for the current prototype.

A. Future Work

Further experiments should be done with the blockchain implementation, to assess the *scaling* behavior. This will determine what are the maximum practical dimensions of user communities.

The mining diversity of Multichain avoids the monopoly in the validation process but, if a same user with a lot of computational power creates different Multichain accounts, the mining diversity may not be enough, because the user can generates different blocks, change its content and still respect the ordering. This kind of attack should be addressed in future work.

At this moment the blockchain is only deployed in server nodes because it needs computational resources for it, but with the increasing computational power inside the smartphones today, another improvement to our solution would be the development of a lightweight version of blockchain that makes it possible to run on mobile devices without relying on server resources to support it. Or follow a hybrid approach, with mobile device and server combined. However, there will be an impact on device battery, so this aspect should be carefully addressed.

The current prototype assumes the creditor and debtor of reward value agree on the outcome. To ensure this is the case, a consensus protocol should be put in place, so that the records inserted in the blockchain represent a shared view of a system interaction. A recent system [21] is proposing the use of smart contracts [22] to achieve this same end. We plan to compare the performance of our scheme with this alternative, as soon as it is publicly available.

The hospital monitoring scenario presented in this paper is illustrative of the capabilities of the proposed crowdsensing system. Since the scope and goal of each community is decided by the users that start it, we envision many more useful scenarios appearing in the future.

B. Reproducible research

The source code of our prototype system is publicly available[3] with instructions to install and run it.

ACKNOWLEDGEMENTS

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013 (INESC-ID) and through project with reference PTDC/CCI-COM/31440/2017 (SureThing).

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] 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*. IEEE, 2014, pp. 593–598.
[3] A. Faggiani, E. Gregori, L. Lenzini, V. Luconi, and A. Vecchio, "Smartphone-based crowdsourcing for network monitoring: Opportunities, challenges, and a case study," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 106–113, 2014.
[4] 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.
[5] R. B. Messaoud, N. Sghaier, M. A. Moussa, and Y. Ghamri-Doudane, "On the privacy-utility tradeoff in participatory sensing systems," in *NCA 2016*, 2016, pp. 1–8.
[6] G. Richter, D. R. Raban, and S. Rafaeli, "Studying gamification: the effect of rewards and incentives on motivation," in *Gamification in education and business*. Springer, 2015, pp. 21–46.

[3] https://github.com/inesc-id/This4That

[7] D. Bhattacherjee, “A comparative study of the incentive mechanisms for mobile crowdsensing,” *Aalto University – CSE-E5000 – Seminar on Software Systems, Technologies and Security*, 2015.

[8] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008.

[9] A. Jøsang and R. Ismail, “The beta reputation system,” in *Proceedings of the 15th Bled Electronic Commerce Conference*, vol. 5, 2002, pp. 2502–2511.

[10] 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*. IEEE, 2014, pp. 98–103.

[11] T. W. Malone, R. Laubacher, and C. Dellarocas, “Harnessing crowds: Mapping the genome of collective intelligence,” *MIT Sloan Research Paper*, no. 4732-09, 2009.

[12] 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.

[13] 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.

[14] 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.

[15] 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*. ACM, 2012, pp. 337–350.

[16] W. Mason and S. Suri, “Conducting behavioral research on amazon’s mechanical turk,” *Behavior research methods*, vol. 44, no. 1, pp. 1–23, 2012.

[17] 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*. ACM, 2008, pp. 211–224.

[18] D. Chaum and E. Van Heyst, “Group signatures,” in *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer, 1991, pp. 257–265.

[19] C. A. Ardagna, S. Jajodia, P. Samarati, and A. Stavrou, “Privacy preservation over untrusted mobile networks,” in *Privacy in Location-Based Applications*. Springer, 2009, pp. 84–105.

[20] V. Benndorf, H.-T. Normann *et al.*, *The willingness to sell personal data*. Düsseldorf Institute for Competition Economics (DICE), 2014.

[21] A. Y. W. L. Y. Z. L. H. J. L. Ming Li, Jian Weng, “Crowdbc: A blockchain-based decentralized framework for crowdsourcing,” Cryptology ePrint Archive, Report 2017/444, 2017, https://eprint.iacr.org/2017/444.

[22] N. Atzei, M. Bartoletti, and T. Cimoli, “A survey of attacks on ethereum smart contracts (sok),” in *International Conference on Principles of Security and Trust*. Springer, 2017, pp. 164–186.