Noname manuscript No.
(will be inserted by the editor)

Reliable Data Analysis through Blockchain based Crowdsourcing in Mobile Ad-hoc Cloud

Saqib Rasool · Muddesar Iqbal · Tasos Dagiuklas · Zia Ul-Qayyum · Shancang Li

Received: date / Accepted: date

Abstract Mobile Ad-hoc Cloud (MAC) is the constellation of nearby mobile devices to serve the heavy computational needs of the resource constrained edge devices. One of the major challenges of MAC is to convince the mobile devices to offer their limited resources for the shared computational pool. Credit based rewarding system is considered as an effective way of incentivizing the arbitrary mobile devices for joining the MAC network and to earn the credits through computational crowdsourcing. The next challenge is to get the reliable computation as incentives attract the malicious devices to submit fake computational results for claiming their reward and we have used the blockchain based reputation system for identifying the malicious participants of MAC.

This paper presents a malicious node identification algorithm integrated within the Iroha based permissioned blockchain. Iroha is a project of hyperledger which is focused on mobile devices and thus light-weight in nature. It is used for keeping the track of rewarding and reputation system driven by the malicious node detection algorithm. Experiments are conducted for evaluated the implemented test-bed and results show the effectiveness of algorithm in identifying the malicious devices and conducting the reliable data analysis through the blockchain based computational crowdsourcing in MAC.

Keywords permissioned Blockchain · Iroha · Hyperledger · Reliable Crowdsourcing · Mobile Ad-hoc Cloud · MAC · Reputation System

Saqib Rasool and Zia Ul-Qayyum are working with University of Gujrat, Gujrat, Pakistan.
Emails: Saqib@ieee.io and ziaqayyum@gmail.com
Muddesar Iqbal and Tasos Dagiuklas are working with London South Bank University, UK
E-mail: m.iqbal@lsbu.edu.pk, E-mail: tdagiuklas@lsbu.ac.uk
Shancang Li works with University of the West of England E-mail: Shancang.Li@uwe.ac.uk

1 Introduction

During the first era of computing, the computation was mainly confined to the central mainframe computers. It was then shifted to the personal computers at the edge of the network. During the last one decade, cloud computing again shifted computation back to the centralize remote locations and now the latest wave of technical advancements, computation is again shifted back to the edge devices through MEC (Multi-Access Edge Computing) [1]. Researchers are working in multiple directions to shift the computation at the edge of the network and one of the research direction is to exploit the under-utilized resources of the mobile devices. Researchers have found that the per hour average utilization of resources of mobile devices is equal or less than 25% [2] and therefore, it is important to device some mechanism for effectively utilizing the resources of mobile devices.

Similar to the computers, the computation of mobile devices was also initially shifted to the remote cloud through a concept known as the MCC (Mobile Cloud Computing). It focuses on shifting the heavy computation from resource constrained mobile devices to the remote cloud [3]. MCC not only helps in preserving the resources of mobile devices but also quickly accomplish the required task by using the resource rich remote platform of cloud. With the tremendous increase in the capabilities of mobile devices, the latter becomes the more prominent feature of MCC as compare to the former. Hence, the efforts have been made to tackle the latter by using resources of nearby mobile devices and this concept is termed as the Mobile Ad-hoc Cloud (MAC) [4].

MAC focuses on shifting the computation to a shared pool of resources contributed by multiple mobile devices. Research shows that the edge cloud formed in MAC can provide services in equally productive or near to the productivity of the remote cloud [5]. Although MAC is considered as a form of MCC [6] but it has two different types of mobile devices; one that offload their computation and the other that perform the same offloaded computation. From the prospective of devices that are offloading the computation, MAC supports both features of MCC. However, from the prospective of mobile devices that performs the computation for executing the computation offloading for other devices, MAC falls under the MEC (Multi-access Edge Computing). MEC was previously known as the Mobile Edge Computing and it was also focused on utilizing the underutilized edge resources for performing the computation which was previously confined to the remote server [7].

Both MEC and MAC exploit the underutilized resources of mobile edge devices for sharing the computation of nearby mobile devices. Many of the studies considered the computation sharing devices as volunteer devices while few have identified the problem to encourage the users for allocating their resources to support the computation offloading for other devices [8]. Researchers also considered the resource allocating nodes as self-interested and rationals and emphasize on devising the processes for incentivizing the resource contributing users to actively participate in MAC [9]. Incentivizing the mobile devices participating in MAC improves the scalability of the system by attracting the masses and encouraging more devices to join the MAC.

Crowdsourcing is a popular way of involving masses towards a collective effort for achieving the desired results [10]. Crowdsourcing has been already used for improving the QoS during the computational offloading [11]. Researchers have also focused on efficiently distributing the computation among the collaborating devices of crowdsourcing [12], [13], [14]. It is claimed that the presence of a strong rewarding system is the most powerful incentive for motivating the participants of the crowdsourcing [15]. Although, the rewarding system motivates the devices for sharing their resources [9] but there still exist the challenge of achieving the reliable results from the participants. The problem of reliable data analysis become more evident during the presence of a rewarding system as it encourages the malicious nodes to submit fake results for claiming rewards, without using their actual resources for computation.

Contribution of this paper is to improve the reliability of the data analysis through a blockchain based rewarding and reputation system in MAC. Blockchain has already been proven for establishing the trust among multiple independent entities. Without loss of generality, the Iroha based permissioned blockchain has been used along with the proposed malicious node identification algorithm for achieving the reliable data analysis in MAC. A test-bed has been implemented to verify the algorithm and results are collected through the real-time experiments on the deployed test-bed to show the effectiveness of our algorithm in identifying the malicious mobile devices and to achieve the reliable data analysis through blockchain based credit and reputation system for crowdsourcing in MAC.

Section two presents the knowledge required for understanding the details of data analysis at different levels of cloud ranging from remote cloud to edge cloud along with our proposed approach of hybrid resource sharing through blockchain. Section three explains the blockchain based credit and reputation system for achieving the reliable data analysis. Section four covers the experimental setup along with the results that shows the effectiveness of our algorithm in identifying the malicious nodes. Section five covers the related work and last section concludes the paper along with some future research directions.

2 Resource Sharing in MAC

This section covers different options of resource sharing given in Fig. 1. It also explains our approach of blockchain based hybrid resource sharing in MAC.

2.1 Horizontal and vertical resource sharing

As given in Fig. 1, each mobile device can offload its computation task to the nearby mobile devices and it is known as the horizontal resource sharing.

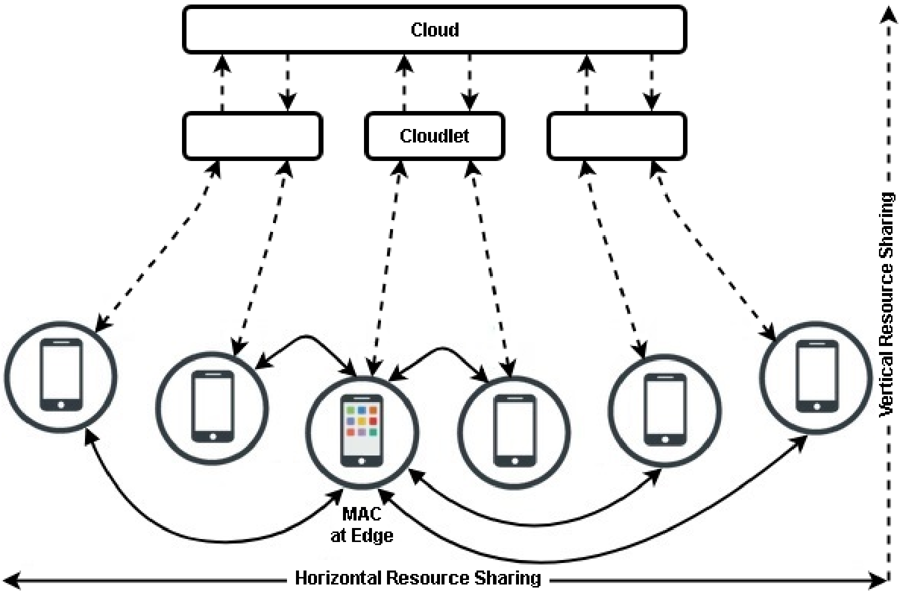

Fig. 1 Vertical, Horizontal and Hybrid Resource Sharing

It is achieved in MAC by maintaining a pool of computation by combining resources of nearby mobile devices. The MAC participants can communicate through either WiFi or Bluetooth with low energy consumption and low transmission delay [16]. Fig. 1 also contains multiple cloudlets and a remote cloud which represents another dimension of resource sharing. Mobile devices can also offload their computation to the nearby cloudlet which can perform the computation by itself or can also forward the same computation to the remote cloud. This process of offloading the computation from mobile devices to the local cloudlets or even to the remote cloud is considered as the MCC [3]. Cloudlet usually establishes the connection with mobile devices using WiFi with medium energy consumption and medium level of transmission delay [16]. Cloudlet also has more resources than mobile devices but less resources than the remote cloud and thus can also offload the computation to the remote cloud for supporting the heavy computational requirements [16].

2.2 Hybrid resource sharing in MAC

MAC can use the hybrid resource sharing which refers to the usage of both neighbouring mobile devices and dedicated infrastructure (like cloudlets or remote cloud) to achieve the computation offloading. Researchers suggest to use the dedicated infrastructure as a backup option alongside the neighbouring mobile devices of MAC [1]. This paper uses the hybrid resource sharing by primarily utilizing the neighbouring mobile devices for data analysis and a local cloudlet is just coordinating the process of reliable data analysis.

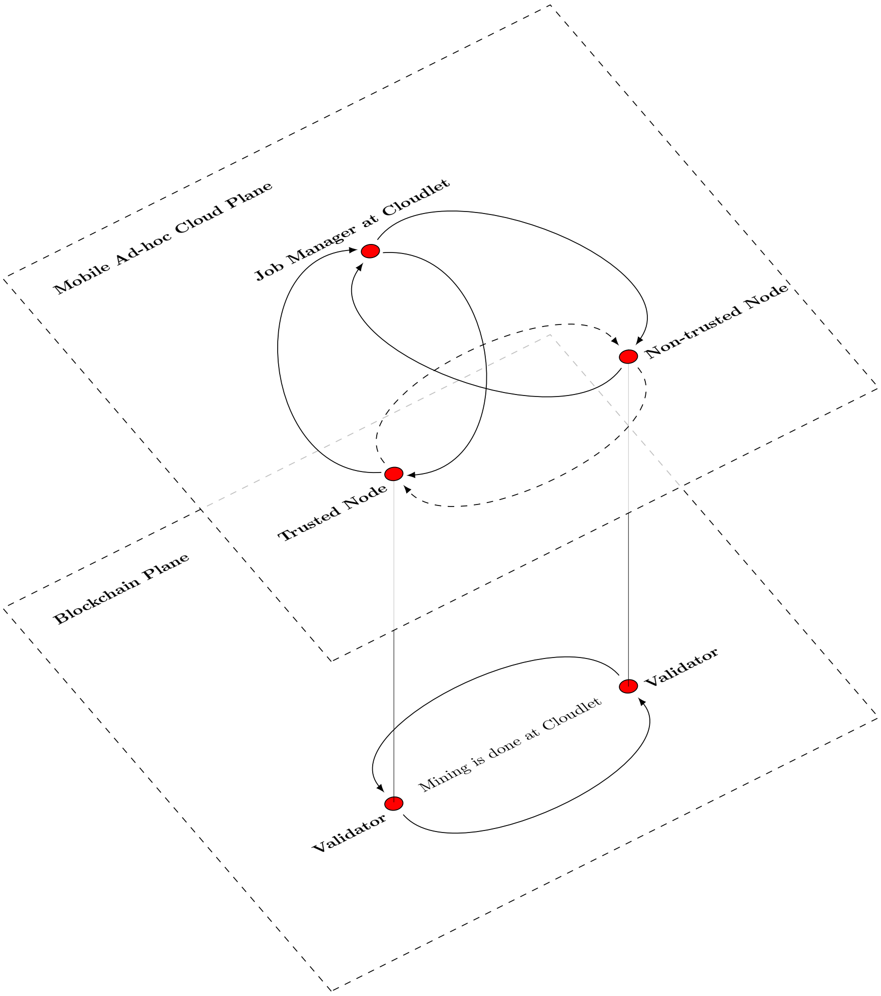

Fig. 2 Collaboration between planes of Mobile Ad-hoc Cloud and permissioned blockchain

2.3 Blockchain based Hybrid Resource Sharing in MAC

Iroha (details are covered in section 3.1) has been used for blockchain implementation which supports both validating and non-validating nodes. Hence, the implementation is also based on two planes for each of these type of nodes. Blockchain plane consists of the validating nodes while the MAC plane consists of the non-validating nodes. Validating nodes control the growth of the distributed ledger and share it with all non-validating nodes. Fig. 2 shows the collaboration between both of these planes. A local cloudlet hosts the validating nodes of Iroha and also coordinates the reliable data analysis among mobile devices (acting as non-validating nodes) in MAC.

3 Blockchain: An Enabler for Reliable Crowdsourcing in MAC

Fig. 2 shows a cloudlets which hosts a blockchain miner and coordinates the whole blockchain based MAC. It also maintains a credit based rewarding system and a ranking based reputation system for maintaining the reliability of the data analysis performed by the participants of the MAC. This section covers the details of blockchain along with its importance in implementing the rewarding and reputation system in MAC.

3.1 Distributed ledger of blockchain

Blockchain [17] has already been proven for removing dependency on a single entity and distributing authority among multiple independent entities. Bitcoin is the first and most popular application based on the blockchain [18]. However, now it has been used in many different type of applications [19]. There are many important features in blockchain that make it unique and effective in comparison to other application development techniques. Among all these features, immutable shared ledger is considered as one of the salient feature of blockchain. Blockchain achieves the immutability by linking multiple blocks of the blockchain through one-way hash, which cannot be modified or decrypted.

First block of the blockchain is known as the genesis block and its hashed is included in the second block as a data element. Hash of second block is included as data element in third block and thus any change in any of the previous blocks alters the hash of the final block of blockchain. This ensures the immutability of blockchain and it is shared with all the participants so that everyone can keep an eye on the growth of the blockchain. Blockchain has been used for storing the earned credits and reputation score of the participating mobile devices. Since the blockchain is shared with all the participants as an immutable ledger therefore, even the owner cannot modify or remove the existing data of the blockchain. This feature of blockchain gives confidence to the participants of the MAC that their earned credits and reputation rank will remain intact.

3.2 Types of blockchain based on read/write access

In contrast to other storage systems, data can only be created or extracted from the blockchain. Fig. 3 refers the data creation rights under the names of blockchain types while the data reading rights over the same names. Data is written in blockchain by adding new blocks with the existing blocks of the blockchain and it is the responsibility of the miners to decide which block is legitimate that can be added in the blockchain. Miners run a consensus algorithm to check if a block is accepted as a legitimate block or not. Following are the details of each blockchain category with respect to the access of data reading and writing:

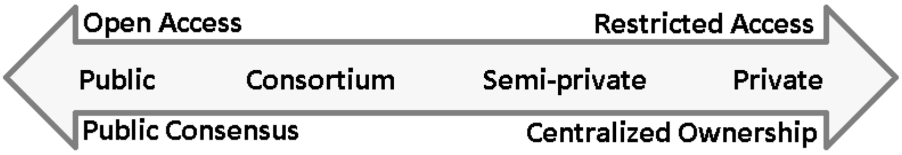

Fig. 3 Types of blockchains based on ownership and access

1. **In Private blockchain** only the central authority decides which nodes can get the blockchain ledger. Private blockchain restricts the ledger to the participating nodes only. With respect to the data writing feature, only the owner hosts the miner and no other participant can claim the mining rights. In the case of MAC, blockchain is heavy [1] therefore, we have used the private blockchain (also known as permissioned-blockchain) to restrict the mining at the cloudlet only.
2. **In public blockchain** anyone can join the blockchain network and can get the blockchain ledger for reading purpose. Similarly, anyone can also join as the miner as well.
3. **In consortium blockchain** some of the predefined members can read or write the data to the blockchain.
4. **Semi-private blockchain** is in-between the private and consortium blockchains. In semi-private blockchains, there is an owner which selects the consortium members and can also update these members as well. Selected consortium members act similar to the members of consortium blockchain.

3.3 Iroha based private blockchain for MAC

Iroha [20] is a project of hyperledger by Linux foundation which was proposed in 2016 by Colu, Hitachi, NTT Data and Soramitsu. It is developed in C++ and is specifically targeting the mobile devices by offering the libraries for iOS and Android platforms. Some of the other libraries come with Iroha are ed25519 library for digital signatures, SHA-3 hashing library, a serialization library for transactions, a P2P library, an API server library etc. Main strength of Iroha is its lightweight nature which makes it a perfect solution for mobile applications.

Following are the three main objectives that we are achieving through the integration of blockchain within MAC:

- As permissioned-blockchain has been implemented using Iroha, only the authenticated mobile devices can join as the participant of MAC.
- The participants of MAC have been incentivized by offering the credits against the computation performed by these devices and the utilization of blockchain for storing the details of earned credits (covered in section 3.4). This is done in order to gain the confidence of the participants.
- Malicious device detection algorithm (covered in section 3.5) generates a reputation rank for each of the participant of the MAC and uses the

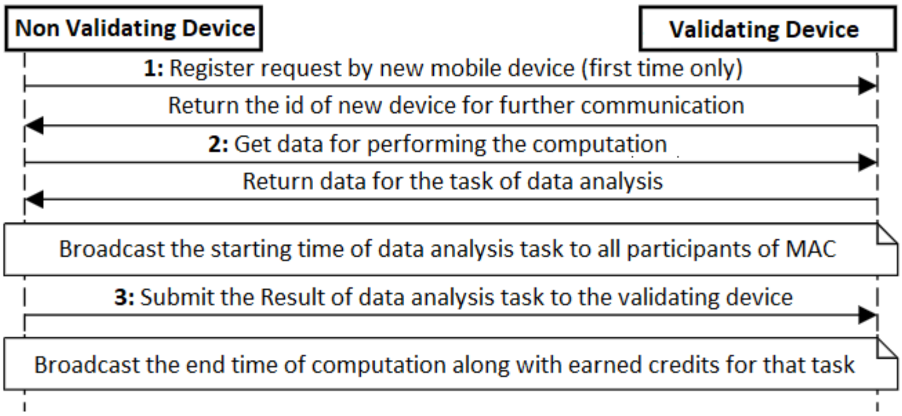
Non Validating Device
Validating Device
1: Register request by new mobile device (first time only)
Return the id of new device for further communication
2: Get data for performing the computation
Return data for the task of data analysis
Broadcast the starting time of data analysis task to all participants of MAC
3: Submit the Result of data analysis task to the validating device
Broadcast the end time of computation along with earned credits for that task

Fig. 4 Procedure of credit earning through blockchain enabled MAC

blockchain for retrieving the previously stored rank and storing the updated rank. All of the generated ranks are shared with all the participants of the MAC and this policy of transparency again employeed to gain the trust of the participants.

3.4 Blockchain based credit system for Incentivizing the crowdsourcing

Crowdsourcing is an important way of offloading the computation and the computation based crowdsourcing is sometimes termed as the crowd computing [21]. Since the participants of crowdsourcing are self-interested and rational therefore, a proper incentivizing system is required to convince them [9] for sharing their resources to join as the participant of the crowdsourcing. Research also suggests that a rewarding system is the best option for motivating and getting the better results from participants of crowdsourcing [15]. Credit based rewarding system not only used for incentiving the participants but also using blockchain for earning the trust of the participants. This section presents the importance of blockchain for implementing the reliable rewarding system in MAC.

Fig. 4 shows the procedure of earning credits through our private blockchain network. Cloudlet hosts the validating device and it controls the growth of the blockchain. All the participants of MAC are acting as non-validating devices of MAC. Step 1, 2 and 3 in Fig. 4 show the requests made by the non-validating devices and following are the details of each of these steps:

- **Step 1:** A new non-validating device make a request for joining the MAC. This request is received by the validating node of Iroha based blockchain network which generates a unique id for the requesting node along with the unique private key (which will act as its signature) and returns it to the requesting node.

- **Step 2:** An already registered non-validating device uses its id and signature for getting the data for analysis from the validating device. Validating device not only gives the computation task to the non-validating device but also assign an id to the task and broadcasts the starting time of each task to all the participants of MAC so that these nodes can store this information in their distributed ledgers of blockchain.
- **Step 3:** Non-validating device uses the id of task, along with its signature, for submitting the response to the validating device which again broadcasts the end time of the computational task along with the details of earned credits against that task to all the participants of MAC.

Before this final broadcast, at the end of step 3, validating device first confirms the accuracy of the submitted result by the non-validating device. This is to ensure that reward can only be claimed after legitimate computational efforts. However, for some some of the scenarios, the results can be easily verified with almost negligible computation effort like the popular way of finding nonce, during the mining process of bitcoin [22]. However, for the results of other computational operations (e.g. mapReduce), equal amount of computation is required for both producing and verifying the results. In that case, there is a need to reduce the computation for verifying the results and this is what we have achieved in this paper. More details of the proposed approach are given in the next section.

3.5 Blockchain based Reputation System for Reliable Crowdsourcing

Berkeley Open Infrastructure for Network Computing (BOINC) is the popular computational crowdsourcing project and it uses the task replication for the verification of the results [23]. However, this task replication takes the duplication of computational and we have developed an algorithm to reduce the number of reanalyses for verifying the results of data analysis through crowdsourcing. Our algorithm is based on the blockchain based reputation system and this sub-section covers its details.

Fig. 5 illustrates the algorithm for providing the reliable data analysis by identifying the malicious nodes. The main idea of the algorithm is to categorize the non-validating devices in to following four categories (also given in the Fig. 3.5 of evaluation scenario):

- **Non-trusted Devices:** When a new non-validating device joins the MAC, it is considered as the non-trusted device and a threshold, defined by the validating device, is required to convert the non-trusted device in to a trusted device. Whenever a result is submitted by a non-validating device, it is being verified through the reanalysis by the trusted devices. Upon each correct submission, the device gets the increment in the rank and it is being shifted to trust device after gaining the rank more than the threshold.
- **Trusted Devices:** As a trusted device has already submitted the correct results after many iterations therefore, no more verification is performed for the results submitted by the trusted devices.

- **Malicious Devices:** Each device which will submit the fake results in the future is nominated as the malicious device and it can be a trusted or a non-trusted device. Algorithm give in Fig. 5 is focused on finding the both trusted and non-trusted malicious devices and results given in Fig. 7 have shown the effectiveness of our proposed algorithm in finding the malicious devices.
- **Blocked Devices:** Once a malicious device is identified with a fake result submission, it is shifted to the group of blocked devices and no further computational tasks are granted to that device.

The steps of algorithm illustrated in Fig. 5 are presented below:

1. Response1 is the response generated after the analytics.
2. Status of Response generating node is collected to determine if this response needs the validation or not?
3. If the score of response generating node is greater than the threshold value then it is considered as the trusted node else it is marked as the non-trusted node.
4. If the node is trusted then both of its credit and score are updated in the blockchain.
5. If the node is non-trusted then the response is pushed to an empty array of RESPONSES.
6. The next step is to find the trusted node which is having score more than the threshold value.
7. Existence of trusted node is confirmed to verify the data generated in Response1.
8. If there is no existing trusted node in the MAC network then Miner of cloudlet performs the analysis by itself to verify if the non-trusted node has submitted the correct response.
9. Miner updates the blockchain by increasing the score and credit in case of correct response or shifts the response generating node to malicious group, if the generated response of miner does not match the generated response by the non-trusted node.
10. If the trusted node found then the least trusted node is selected which is having the least score, after passing the given threshold value.
11. Same analysis of non-trusted node is performed by the trusted node N to confirm if the generated response is correct or not.
12. Trusted node N returns the ResponseN against the same data and analysis of non-trusted node.
13. Responses array may contain one or many responses and the ResponseN is compared with the existing responses in the array of RESPONSES.
14. If the ResponseN matches any of the responses in the RESPONSES array then it not only validates the matching results but also prove its trustfulness.
15. If the ResponseN generated by the trusted node N does not matches any of the existing responses then its result also needs to be validated and thus being pushed to the array of RESPONSES.

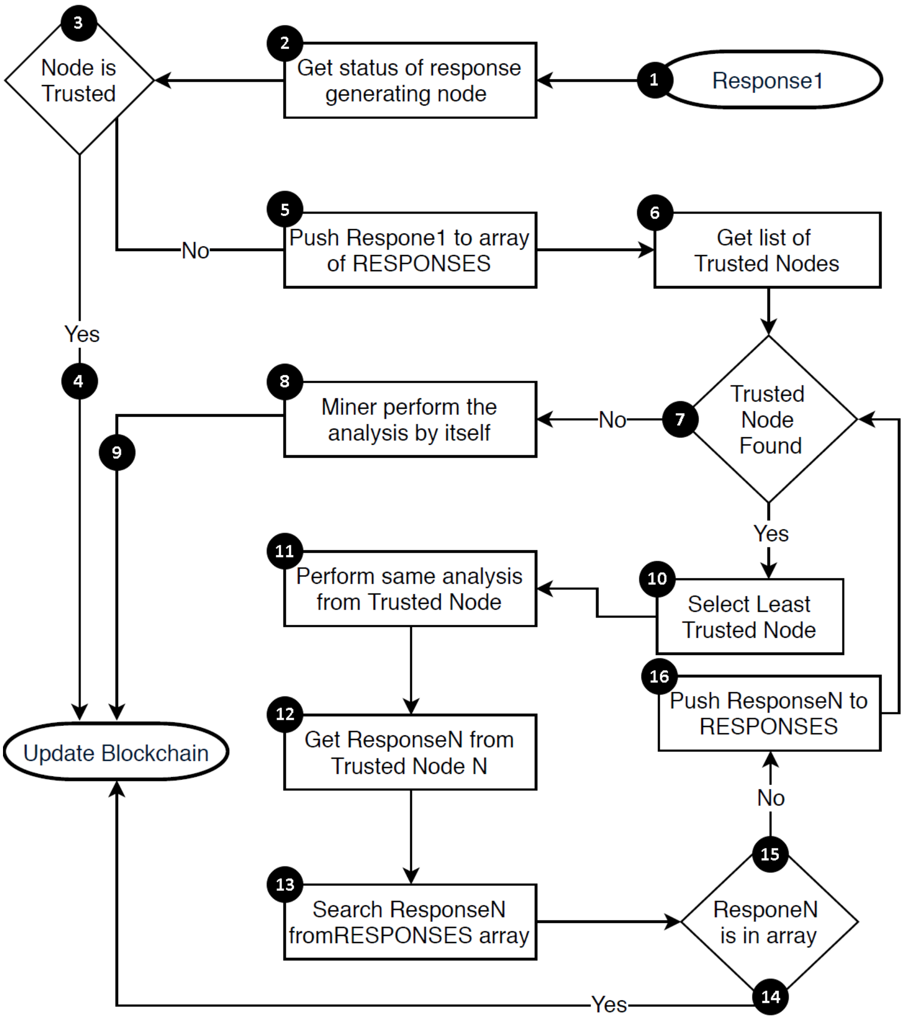
1
Response1
2
Get status of response generating node
3
Node is Trusted
No
Yes
4
5
Push Respone1 to array of RESPONSES
6
Get list of Trusted Nodes
7
Trusted Node Found
No
Yes
8
Miner perform the analysis by itself
9
10
Select Least Trusted Node
11
Perform same analysis from Trusted Node
12
Get ResponseN from Trusted Node N
13
Search ResponseN fromRESPONSES array
14
15
ResponeN is in array
No
Yes
16
Push ResponseN to RESPONSES
Update Blockchain

Fig. 5 Algorithm for detection of malicious nodes

16. This is not the last step of the algorithm as it again starts the validation of all of the existing responses in the RESPONSES array. This loop will keep on repeating until it exits in one of the following two ways:
    (a) **At point 15**, after being verified by a more trusted node.
    (b) **At point 9**, after being verified the cloudlet based miner. It only happens if none of the results in RESPONSES array are matched after being iterating the all trusted nodes. In that case, miner will update the blockchain by placing all the nodes of non-matching results in the category of malicious nodes.

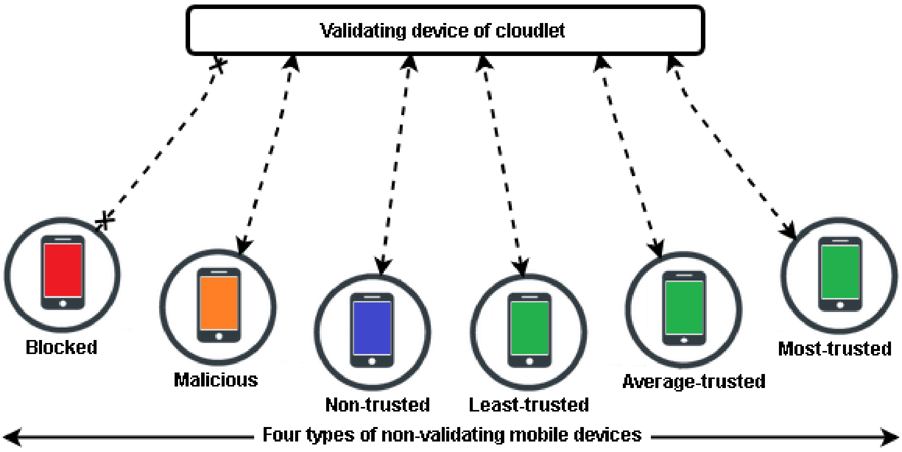
Validating device of cloudlet
Blocked
Malicious
Non-trusted
Least-trusted
Average-trusted
Most-trusted
Four types of non-validating mobile devices

Fig. 6 Validating device of cloudlet is collaborating with different types of non-validating mobile devices to reliably accomplish the tasks of mapReduce

4 Experimentation and Results

Big data technologies such as mapReduce help in thriving the agile businesses [24] and therefore the test-bed has been implemented for the mapReduce operation. Another reason for choosing the mapReduce is the Java Stream API which was introduced in the 8th version of Java and it simplifies the implementation of mapReduce operation [25]. This section explains the evaluation scenario along with the discussion on experimental results.

4.1 Evaluation Scenario

Fig. 6 presents the evaluation scenario which contains six mobile devices. Following is the detail of each mobile device given in Fig. 6 from left to right:

- **Blocked device** is the left-most device in Fig. 6 and it has been permanently blocked as it submitted the fake results for the assigned task of mapReduce.
- **Malicious device** is given at second from left in Fig. 6. This malicious device will submit the fake result of assigned task of mapReduce for claiming the reward without performing the required computation. Our algorithm is supposed to identify the submission of fake result by the malicious device which will result in the blockage of the malicious device. Both non-trusted and trusted devices can act as the malicious devices and our algorithm has to find the both types of malicious devices. In the evaluation scenario, different malicious devices have been used under different circumstances and the proposed algorithm has successfully identified the fake results with different number of reanalyses attempts. The details of these experimental

results are given in Fig. 7 along with description of each result in next sub-section of results and discussion.
- **Non-trusted device** is the third from the left in Fig. 6 and both haven't crossed the threshold to become the trusted devices. The threshold value of ten correctly validated results has been used for converting a non-trusted device to a trusted device.
- **Least-trusted device** is given at fourth from left in Fig. 6. This device has crossed the threshold of ten but has the least score of eleven correctly submitted results.
- **Average-trusted device** is given at fifth in Fig. 6 and its score is twelve in our evaluation scenario.
- **Most-trusted device** is given at the right most of the Fig. 6 with the reputation score of thirteen.

4.2 Results and Discussion

Fig. 7 presents the results for the evaluation of our algorithm to find the malicious nodes. As our algorithm uses the dynamic number of reanalyses for finding the fake result submissions therefore, the y-axis shows the total number of reanalyses performed for finding the malicious devices under different circumstances. While the x-axis shows the malicious devices in the MAC that will also report the fake results. There are four lines in the Fig. 7 that are labelled with non-trusted, least-trusted, average-trusted and the most-trusted devices and these lines are also representing the malicious devices. For the sake of simplicity, we can classify the malicious devices in to following two sub-categories:

1. **Active malicious device** is the one that we are trying to find and these devices are labelled against four different lines of Fig. 7. The number of reanalyses are evaluated for the active malicious devices.
2. **Passive malicious device** is another malicious device in the MAC and these devices are labelled at x-axis. The number of reanalyses are not evaluated for these passive malicious devices and we are just using the passive malicious devices for finding their impact on the number of reanalyses for identifying the fake results by the active malicious devices.

Black solid line shows the number of reanalyses done for finding the active malicious device from the group of non-trusted mobile devices. It took two to three number of reanalyses attempts for finding it under following four scenarios:

1. **When only non-trusted device is the malicious device** then it took two reanalyses attempts that are performed by the least-trusted and average-trusted mobile device for identify the malicious devices.
2. **When least-trusted device is the passive trusted device and the non-trusted device is the active malicious device** then it took three reanalyses and finally the average-trusted and the most-trusted mobile devices has found the both malicious devices.

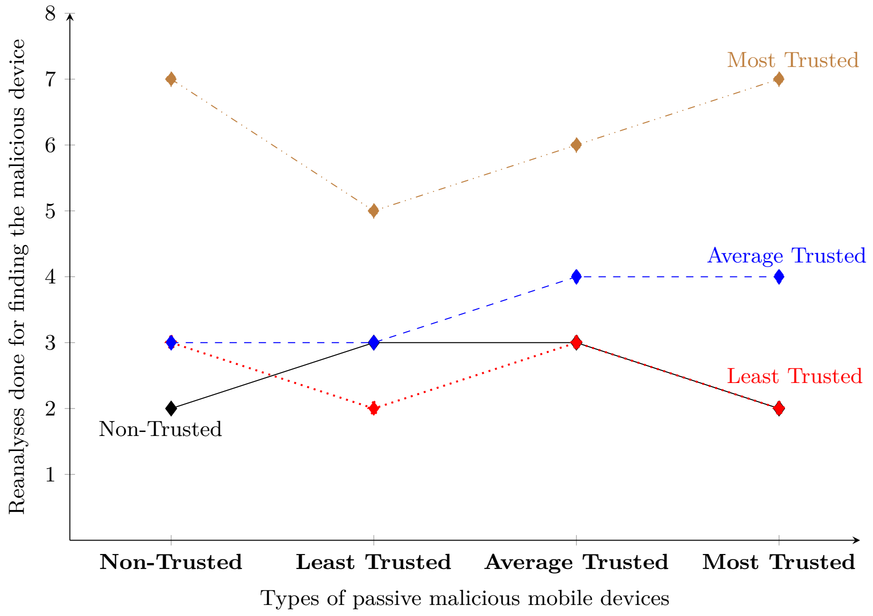

Fig. 7 Number of reanalyses required for finding the malicious devices

3. **When average trusted is the passive trusted device and the non-trusted device is the active malicious device** then it took three reanalyses and finally the least-trusted and most-trusted will identify the both malicious devices.
4. **When most-trusted is the passive malicious device and the non-trusted device is the active malicious device** then it took two reanalyses by both the least-trusted and average-trusted mobile device to find the non-trusted active malicious device.

Red dotted line shows the number of reanalyses done for finding the active malicious device from the group of least-trusted mobile devices. It took two to three number of reanalyses attempts for finding it under following four scenarios:

1. **When non-trusted device is the passive malicious device and the least-trusted device is the active malicious device** then it took three reanalyses attempts and finally the average-trusted and most-trusted mobile devices have identified the both malicious devices.
2. **When only the least-trusted device is the malicious device** then it took two reanalyses and finally the non-trusted and average-trusted mobile devices has found the active malicious device of least-trusted group.
3. **When average-trusted device is the passive malicious device and the least-trusted device is the active malicious device** then it took

three reanalyses and finally the non-trusted and most-trusted devices has identified the both malicious devices.

4. **When most-trusted device is the passive malicious device and the least-trusted device is the active malicious device** then it took two reanalyses by both the least-trusted and average-trusted mobile device to find the both malicious devices.

Blue dotted line shows the number of reanalyses done for finding the active malicious device from the group of average-trusted mobile devices. It took three to four number of reanalyses for finding it under following four scenarios:

1. **When non-trusted device is the passive malicious device and the average-trusted device is the active malicious device** then it took three reanalyses and finally the least-trusted and most-trusted mobile devices have identified the both malicious devices.
2. **When least-trusted device is the passive malicious device and the average-trusted device is the active malicious device** then it took three reanalyses and finally the average-trusted and the most-trusted mobile devices has found the both malicious devices.
3. **When only the average-trusted device is the malicious device** then it took four reanalyses attempts in finding the both malicious devices.
4. **When most-trusted device is the passive malicious device and the average-trusted device is the active malicious device** then it also took four reanalyses attempts for finding the both malicious devices.

Both dotted and dashed line of brown color shows the number of reanalyses done for finding the active malicious device from the group of most-trusted mobile devices. It took five to seven number of reanalyses for finding it under following four scenarios:

1. **When non-trusted device is the passive malicious device and the most-trusted device is the active malicious device** then it took seven reanalyses attempts for finding the both malicious devices.
2. **When least-trusted device is the passive malicious device and the most-trusted device is the active malicious device** then it took five reanalyses attempts in finding the both malicious devices.
3. **When average-trusted device is the passive malicious device and the most-trusted device is the active malicious device** then it took six reanalyses attempts in finding the both malicious devices.
4. **When only the most-trusted device is the malicious device** then it again took the seven reanalyses for finding the both malicious devices.

Results can be summarized as the two to three number of reanalyses attempts are required for finding the active malicious devices from both non-trusted and least-trusted mobile device and reanalyses attempts of three to four and five to seven are required for finding the active malicious devices from average-trusted and most-trusted mobile devices. Although our algorithm took more reanalysis attempts for finding the malicious devices from average and most-trusted devices but it is still identifying the malicious mobile devices even in the worst case scenario.

5 Related Work

Existing mobile devices are using, on average, less than 25% of resources per hour [2]. In order to exploit these resources of mobile devices, researchers have deployed the hadoop on android devices for performing the MapReduce [26]. However, due to extra burden of mobile devices, it is difficult to run hadoop on mobile devices and thus an isolated version of MapReduce is also implemented for Android devices [27]. This performs the mapReduce at the mobile devices and connects it with the remote deployment of the hadoop . Hence, the researchers have already proved the feasibility of MapReduce at mobile devices and claimed that the shifting of MapReduce on mobile devices helps in improving computation performance and job throughput by outsourcing the computation to mobile devices [28]. We have extended the same idea of performing the mapReduce through a MAC and added an algorithm for improving the reliability of the mapReduce performed by random mobile devices.

BOINC [1] (Berkeley Open Infrastructure for Network Computing) is the popular project of crowdsourcing based computation [29] and it uses the recomputations for confirming the reliability of the results [23]. Researchers have also included a special computation task which is designed for testing the reliability of computation results by the participants of the crowdsourcing. This task is randomly given to the participants and results of this task are already known to the task dispatching node. If the returned result matches the already known value then the other results of the same node are also considered correct. We are using an hybrid approach which uses some of the techniques of both discussed approaches. We are also performing the recomputations for finding the accuracy of results but for the non-trusted nodes only and we are using the same results for finding the accuracy of the results by the trusted devices.

We are using blockchain for storing both the earned credits and reputation rank of the participant devices of MAC. GridCoin (GRC) [2] is the custom build blockchain which specifically targets the data analysis by the project of BOINC. There are few other ethereum based blockchain projects like iExec (RLC) [3] for virtual cloud infrastructure, Golem (GNT) [4] for renting CPU, GPU, SONM (SNM) [5] for fog computing etc. All of the listed projects are using blockchain for storing the credit earning details in the form of their specific coins while we are using blockchain for both the details of earned credits and rank. Also none of these are Iroha based which is the lightweight blockchain specifically designed for the mobile devices while we are specifically targeting the mobile devices through the Iroha based blockchain implementation.

[1] https://boinc.berkeley.edu/

[2] https://whitepaper.io/coin/gridcoin

[3] https://whitepaper.io/coin/iexec

[4] https://whitepaper.io/coin/golem

[5] https://whitepaper.io/coin/sonm

6 Conclusion and Future Work

Mobile Ad-hoc Cloud is an effective way of exploiting the underutilized resources of mobile devices. However, there is a problem of motivating the users of mobile devices to share their resources for computation. For the same purpose, we have introduced a blockchain based credit system to incentivize the users of mobile devices.

This credit based system also attracts the selfish nodes and it results in another challenge of denying the malicious nodes to submit the fake results. Our contributions include the realization of reliable data analysis by integrating the blockchain within the MAC. Experimental results show the effectiveness of our algorithm for identifying the malicious nodes.

Following are some of the future research directions that can be focused for further improving the :

- Bitcoin uses an algorithm for dynamically adjusting the difficulty level of the nonce [30]. Test-bed presented in this paper is based on the hard-coded threshold value for shifting a non-trusted device to the category of trusted devices. Similar to bitcoin, an algorithm can be implemented for dynamically adjusting the value of threshold based on the real-time conditions of the MAC network.
- Test-bed presented in this paper is tightly bound to the operation of mapReduce and an alternate implementation can be provided in a decoupled manner. It can expose the APIs to allow the integration for custom logic of data analysis and ensures its execution through the MAC based reliable crowdsourcing.
- Malicious device detection algorithm presented in this paper can be integrated with an existing open source project (e.g. BOINC [29]) to find its effectiveness on a global scale.
- Current implementation hosts the validating node at cloudlet only. An improved implementation can use the idle mobile devices as the validating devices and move from hybrid resource sharing to purely horizontal resource sharing in MAC.

References

1. F. N. Nwebonyi, R. Martins, and M. E. Correia, “Reputation-based security system for edge computing,” in *Proceedings of the 13th International Conference on Availability, Reliability and Security.* ACM, 2018, p. 39.
2. B. Varghese and R. Buyya, “Next generation cloud computing: New trends and research directions,” *Future Generation Computer Systems*, vol. 79, pp. 849–861, 2018.
3. S. Rasool, A. Saleem, and A. N. Mian, “Poster: Rql: Rest query language for converting firebase to a mobile cloud computing platform,” in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking.* ACM, 2017, pp. 567–569.
4. Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “Mobile edge computing: Survey and research outlook,” *arXiv preprint arXiv*, vol. 1701, 2017.

5. V. Balasubramanian and A. Karmouch, “An infrastructure as a service for mobile adhoc cloud,” in *Computing and Communication Workshop and Conference (CCWC), 2017 IEEE 7th Annual.* IEEE, 2017, pp. 1–7.
6. A. A. A. A. Khalifa, “Collaborative computing cloud: Architecture and management platform,” Ph.D. dissertation, Virginia Tech, 2015.
7. S. Lee, K. Grover, and A. Lim, “Enabling actionable analytics for mobile devices: performance issues of distributed analytics on hadoop mobile clusters,” *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 2, no. 1, p. 15, 2013.
8. I. Yaqoob, E. Ahmed, A. Gani, S. Mokhtar, M. Imran, and S. Guizani, “Mobile ad hoc cloud: A survey,” *Wireless Communications and Mobile Computing*, vol. 16, no. 16, pp. 2572–2589, 2016.
9. L. Tang, S. He, and Q. Li, “Double-sided bidding mechanism for resource sharing in mobile cloud,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 1798–1809, 2017.
10. J. Howe, “The rise of crowdsourcing,” *Wired magazine*, vol. 14, no. 6, pp. 1–4, 2006.
11. D. Yao, C. Yu, L. Yang, and H. Jin, “Using crowdsourcing to provide qos for mobile cloud computing,” *IEEE Transactions on Cloud Computing*, 2015.
12. N. Fernando, S. W. Loke, and W. Rahayu, “Mobile crowd computing with work stealing,” in *Network-Based Information Systems (NBiS), 2012 15th International Conference on.* IEEE, 2012, pp. 660–665.
13. ——, “Computing with nearby mobile devices: a work sharing algorithm for mobile edge-clouds,” *IEEE Transactions on Cloud Computing*, no. 1, pp. 1–1, 2016.
14. S. W. Loke, K. Napier, A. Alali, N. Fernando, and W. Rahayu, “Mobile computations with surrounding devices: Proximity sensing and multilayered work stealing,” *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 14, no. 2, p. 22, 2015.
15. K.-J. Stol and B. Fitzgerald, “Two’s company, three’s a crowd: a case study of crowdsourcing software development,” in *Proceedings of the 36th International Conference on Software Engineering.* ACM, 2014, pp. 187–198.
16. J. Hao, M. Xian, H. Wang, F. Tang, and P. Xiao, “Mobile cloud computing: the state of art, application scenarios and challenges,” in *2018 4th International Conference on Computational Intelligence & Communication Technology (CICT).* IEEE, 2018, pp. 1–5.
17. S. Underwood, “Blockchain beyond bitcoin,” *Communications of the ACM*, vol. 59, no. 11, pp. 15–17, 2016.
18. S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008.
19. M. Pilkington, “11 blockchain technology: principles and applications,” *Research handbook on digital transformations*, p. 225, 2016.
20. I. Bashir, *Mastering Blockchain.* Packt Publishing Ltd, 2017.
21. D. G. Murray, E. Yoneki, J. Crowcroft, and S. Hand, “The case for crowd computing,” in *Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds.* ACM, 2010, pp. 39–44.
22. A. Miller, A. Juels, E. Shi, B. Parno, and J. Katz, “Permacoin: Repurposing bitcoin work for data preservation,” in *2014 IEEE Symposium on Security and Privacy (SP).* IEEE, 2014, pp. 475–490.
23. U. of California, “Jobreplication boinc,” https://boinc.berkeley.edu/trac/wiki /JobReplication, April 2015, (Accessed on 10/30/2018).
24. B. Unhelkar, “Big data framework for agile business (bdfab) as a basis for developing holistic strategies in big data adoption,” in *Big Data and Visual Analytics.* Springer, 2017, pp. 85–95.
25. Y. Chan, A. Wellings, I. Gray, and N. Audsley, “A distributed stream library for java 8,” *IEEE Transactions on Big Data*, vol. 3, no. 3, pp. 262–275, 2017.
26. S. Lee, K. Grover, and A. Lim, “Enabling actionable analytics for mobile devices: performance issues of distributed analytics on hadoop mobile clusters,” *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 2, no. 1, p. 15, 2013.
27. ——, “Enabling actionable analytics for mobile devices: performance issues of distributed analytics on hadoop mobile clusters,” *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 2, no. 1, p. 15, 2013.

28. T.-Y. Liang, L.-W. Yeh, and C.-H. Wu, "A visual mapreduce program development environment for heterogeneous computing on clouds," in *Proceedings of the 2018 International Conference on Computing and Data Engineering.* ACM, 2018, pp. 83–87.
29. D. P. Anderson, "Boinc: A system for public-resource computing and storage," in *Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on.* IEEE, 2004, pp. 4–10.
30. J. Göbel, H. P. Keeler, A. E. Krzesinski, and P. G. Taylor, "Bitcoin blockchain dynamics: The selfish-mine strategy in the presence of propagation delay," *Performance Evaluation*, vol. 104, pp. 23–41, 2016.