2017 15th Annual Conference on Privacy, Security and Trust

An Identity Management System Based on Blockchain

Yuan Liu
Software Colledge
Northeastern University
Shen Yang, Liao Ning, China, 110169
Email: liuyuan@swc.neu.edu.cn

Zheng Zhao
Software Colledge
Northeastern University
Shen Yang, Liao Ning, China, 110169
Email: neumrzz@hotmail.com

Guibing Guo
Software Colledge
Northeastern University
Shen Yang, Liao Ning, China, 110169
Email: guogb@swc.neu.edu.cn

Xingwei Wang
Software Colledge
Northeastern University
Shen Yang, Liao Ning, China, 110169
Email: wangxw@mail.neu.edu.cn

Zhenhua Tan
Software Colledge
Northeastern University
Shen Yang, Liao Ning, China, 110169
Email: tanzh@swc.neu.edu.cn

Shuang Wang
Software Colledge
Northeastern University
Shen Yang, Liao Ning, China, 110169
Email: wangs@swc.neu.edu.cn

***Abstract*—In this paper, we propose a decentralized identity management system based on Blockchain. The function of the system mainly includes identity authentication and reputation management. The technical advantages of the Blockchain makes the data in the system safe and credible. In addition, we use smart contracts to write system rules to ensure the reliability of user information. We bind the user's entity information with the public key address and determine the true identity of a virtual user on the Blockchain. We use the token to represent the reputation which is shown to be an effective reputation model, making the participants in the system prefer to maintain and manage their personal reputation. Our system makes it possible for users to securely manage their identity and reputation on the Internet.**

I. Introduction

In recent decades, it has been brought to people's attention that the Internet security issue is crucial and challenging. Many sensitive personal information is often misused or leaked and financial assets are hacked etc. These security events directly or indirectly cause the economic losses for the Internet users, even destroy the whole Internet transaction environment. Thus, how to manage an identity over the Internet becomes an important problem for both the Internet companies and academic researchers [5]. Many efforts have been taken to seek effective approaches in protecting the personal data security. However, the personal data is traditionally stored in a centralized server, which makes it possible for hackers or attackers to achieve their malicious goals by stealing/misusing/manipulating these data in this centralized server [5]. In 2008, Satoshi Nakamoto proposed the concept of Bitcoin [7], where users trade freely on the Internet without a credible/trusted third party. Because the popularity and fast development of Bitcoin, Blockchain that the technology supports Bitcoin begins to take the public attention. In other words, the Bitcoin starts a new era for the Blockchain technology [13], where it is possible to create and transfer values without trusted medium on the Internet [8]. The biggest feature of the Blockchain is its decentralization where the whole database is maintained by all the nodes on the network. A consensus mechanism ensures that the creation and modification of data are agreed by all the nodes or the majority of the nodes. In this way, the Blockchain has the features of high security, not-easy to be tampered with, and so on [12]. These features make it become a potentially ideal solution for authenticating and protecting identify management system. Meanwhile, the users are able to store their personal information in the Blockchain without worrying about anyone to illegally steal or modify their data, ensuring the information security requirement of an identity management.

In this paper, we propose an identity management system based on the Blockchain technology. In our model, we combine the identity authentication technology and reputation management together, then establish a personal online reputation data file on the blocks. From the technical perspective, we store personal identities and reputation information in the blocks, a distributed database system. Thus, there is no central management organization in the system, which ensures the system data be safe and credible. From the management perspective, the reputation of users are constructed based on the process of building the blocks, which is the first attempt in the literature and points a new direction for reputation management systems based on the Blockchain.

II. Related Work

A. Blockchain

Blockchain is a distributed database system, which can also be treated as a public ledge that is maintained by lots of independent users [8]. Once a transaction is written in a block, the transaction data has to be agreed by all the nodes in the system and the data cannot be further modified by any node. If the data in a block on the chain is illegally changed, it will affect the entire chain after this block and other nodes will not acknowledge the validity of the data on the chain. The participants or nodes in the system are not necessary to

978-1-5386-2487-6/17/$31.00 ©2017 IEEE
DOI 10.1109/PST.2017.00016

trust each other, and they may be strangers in the Internet. Moreover, all the system rules are public and transparent, and are also agreed by all the nodes. Each node holds the information abstraction of all the data stored in the whole block based chain. In other words, each node holds a copy of the database, which guarantees that the database cannot be manipulated by a single node individually. The architecture of the blocks in the form of a chain, linked by cryptography hash, provides the promising distributed security [13].

B. Ethereum and Smart Contract

Ethereum is an open source Blockchain project [2]. It proposes a common Blockchain solution that allows anyone to create a distributed application based on Blockchain. The biggest feature of the Ethereum that it links the smart contract and the Blockchain. A smart contract is a numerally defined contract which is similar to a traditional paper contract. A smart contract specifies the participants, the execution conditions, other detailed transaction rules. The difference with the traditional contract is that the smart contract is completely controlled by the computer program and does not depend on any agency. Once the contract is deployed on the chain, no individual node is able to change it. Meanwhile, once the execution conditions of the contract are satisfied, the contract is automatically executed [12]. Therefore the projects based on the Ethereum and smart contracts are decentralized and credible.

C. Identity Authentication based on Blockchain

An identify system is necessary for business and social affairs [1]. For example, one has to start a bank account to enjoy the services provided by the bank, and people have to be assigned with an identity number so as to achieve the social benefits provided by a country. Traditional identity management systems have a centralized database to store the personal data. The higher of the security level the database requires, the more cost the system has to pay. Thus the centralized database system becomes the security bottleneck of an identity management system. It is always subject to the attacking hackers, system downtime, unavoidable software license fees and upgrades, as well as hardware limitations and network traffic restrictions [4]. So personal identity information is often leaked or even tampered with. What's worse, a user's identity information on the network is not only the user's representation in the virtual network, but also contains the user's behavior and credit information. Therefore, secure and serious management of identities on the Internet is essential. Taking advantages of the Blockchain technology, an identity management system can benefit from the decentralization feature as there is no need for the data centers or dedicated servers and Blockchain can store all the information in a non-manipulable way. One can also verify your identity anywhere on the Internet without the media of an administrator.

At present, there are a number of identity authentication systems based on the Blockchain. ShoCard [9] is a service that keeps the data fingerprint of the documents of entity identity in the Blockchain. The data fingerprint on the Blockchain is protected by a private key. Only the users who hold the private key can modify the data. OneName [11] provides an identity service that allows users to bind their names and bitcoin addresses to social accounts, which is equivalent to provide a public Bitcoin address and digital signature for each social account. The Bitnation project [10] allows users to register as a Bitnation"Citizens" and get Bitnation's "World Citizen Identity Card". With the assigned citizenship, a user is able to access the self-accredited services in the Bitnation .

III. The Main Concepts in the Proposed Module

In this section, we introduce the new concepts proposed in our module, to help the readers understand our module proposed in the next section.

A. Token

In the system tokens are used as a reflection of a user's reputation, which is presented in the form of $RpCoin$. $RpCoin$ cannot be transfered or transacted between users in any form. The $RpCoin$ can only be generated through executing a smart contract and assigned to the designated user, thus no one can achieve $RpCoin$ by purchasing in transactions. In the system, the users are able to accumulate $RpCoin$ by actively participating in the Blockchain reputation tasks and incentive tasks. The negative participants are penalized by deducting a certain amount of $RpCoin$. Therefore $RpCoin$ can be used as a digital certification for the user's reputation.

B. Reputation Task

In the system, a reputation task is a type of crowdsourcing tasks with binary choices, targeting at establishing one's reputation. It has two stages: publishing stage and consensus stage. In the publishing stage, the task publisher creates a mission to prove their reputation (reputation task is usually beneficial to the publisher), and publishes the task to the nodes/workers of the whole network. After the number of task participants satisfying the minimum number requirement which is specified by the system, the task consensus stage begins. In the task consensus stage, the task receives the votes from the participating worker within a certain period of time. Over a specified time, the tasks without votes are abandoned. The workers can vote between "Agree" and "Not Agree" with the reputation statement in the task. When a worker votes "Agree" in a task, it means that the worker is agree with the statement of the task in improving the reputation of the publisher. On the contrary, when a worker votes "Not Agree" with the task, it indicates that the reputation statement in the task is not approved by the worker. If the number of voters satisfies the minimum number requirement, the consensus stage can be finalized in advance of predefined voting time. Based on the votes in a reputation task, a smart contract is executed to aggregating the votes so as to come up with the final task result. The task result is also binary, "Approved" or "Rejected".

After a reputation task is completed, the $RpCoin$ of the task publisher and workers are updated. For task publishers, when the result of the reputation task is "Approved", the publisher will be given $RpCoin$, otherwise the publisher's $RpCoin$ will be deducted. For workers, if the final aggregated result is consistent with their votes, they will receive $RpCoin$ in proportion to their weight, otherwise they will be punished by deducting a certain amount of $RpCoin$, which will be specified in the following sections.

C. Incentive Task

As the reputation system requires the truthful participation of users, our system also sets a type of special tasks which is called incentive tasks. The incentive tasks are necessary, as there exists two types of undesirable behaviors of participates, as specified as follows.

- Negative worker: in a reputation task, a participant may not take their efforts in evaluating the reputation statement in the task, instead the participating workers may provide misleading votes.
- Repetitive task: if a task publisher repeatedly publishes the same task which has been approved previously, a certificate of reputation then will be misused repeatedly. This phenomena is harmful for the reputation system in maintaining its effectiveness.

To avoid the influence of the above two undesirable behaviors, an incentive task can be created by any individual user. An incentive task can be a claim about any other users who are negative workers or the proofs that another user ever published repetitive tasks. The main difference between reputation tasks and incentive tasks is the content of the tasks.

After an incentive task is initiated by a worker, then other workers can participate to complete in the task. The participating and voting rules are similar to that in a reputation task. The rules of reward and punishment $RpCoin$ are as follows.

- For incentive tasks against negative workers, if the final result is "Approved", then the negative behavior of the worker in the task is recognized. The respective worker in the task and the workers who vote "Not Agree" will be penalized by deducting his $RpCoin$, meanwhile the task initiator and the worker who voted "Agree" will be rewarded with $RpCoin$. If the final result is "Rejected", the statement in task is not regarded as the truth and the respective worker mentioned in the task will not be rewarded nor be punished. However the initiator of the task and the workers who vote "Agree" will be subject to the loss of $RpCoin$, and the workers who vote "Not Agree" will get $RpCoin$ reward.
- For incentive tasks against repetitive tasks, if the voting result is "Approved", indicating that the task was identified as a repetitive task, the respective task creator would lose $RpCoin$. For the task initiator and workers, the reward and punishment rules are the same as those of the incentive tasks against negative workers.

We believe that the proposal of incentive tasks are greatly beneficial for our system. On the one hand, the incentive task creates a mechanism for issuing tokens, which provides $RpCoin$ rewards for those who try to find the bad participants in the system. On the other hand, the incentive task is constantly punishing the negative users in the system. This facilitates the good development of the system

IV. The Proposed Module

In this section, we propose an identity management system based on the Ethereum Blockchain and the operating mechanism of this system. Unlike other identity management systems, our system consists of two parts: identity authentication module and reputation management module. The purpose of the identity authentication module is to allow an identity to correspond to an Ethereum public key address, ensuring that an entity in the real world can and only can create an unique corresponding virtual identity in our system. The reputation management module aims to record the behavior of an identity behaving in the system. In our system, we use the Blockchain to store the system data in a completely decentralized mode. No one can modify the user's data so as to ensure the safety of the identity related data. We use smart contracts to write authentication and credit rules. When user behavior meets the contractual execution conditions, the contract is automatically executed with the data accurately written or modified, ensuring the security of the reputation related data.

A. Authentication

In the Ethereum, a real user can create multiple public key addresses as his identity. If the address of the public key is used as the system identity, the reputation results can be manipulated or controlled by attackers as long as the number of votes takes more than 50%. Therefore, the reputation calculated based on the reputation tasks will not be convincing or trustworthy. In order to solve this problem, we propose an identity solution, which establishes a virtual identity and real identity mapping relationship, where a unique public key address is assigned to a single virtual identity, and any real identity can and only can has the unique virtual identity.

1) Identity Creation: For the users who participate in the tasks, it is a pre-requirement for them to provide their personal information. We define real user information as $IdentityInfo$ which uniquely identifies a realistic user. The $IdentityInfo$ is usually in the form of text, image and biometric data, which is Managed by the user and encrypted by the user's private key in the Blockchain. We use $IdentityInfo$'s hash value of a user as the ID of the entity user information, and ID is unique for an entity user in our system.

$$ID_i = Hash(IdentityInfo_i) \tag{1}$$

Then we define $Identity_i$ as the identity of the user i in the system. $Identity_i$ consists of three parts: the user's ID, Ethereum public key address and his $RpCoin$.

$$Identity_i = (Address_i, ID_i, RpCoin_i) \tag{2}$$

where $Address_i$ is the user's Ethereum address for accepting or transferring information and values, $RpCoin_i$ is used to

identify the user's reputation. In this way, an entity user in a physical world is uniquely identified by the combination of his personal information, public key address, and his $RpCoin$

2) Identity modification: In practice, an entity user may update his personal information, and his Ethereum address may also change. In order to ensure that the modification of identity information will not affect the user's reputation, while preventing the creation of multiple identities, the system executes $RpCoin$ transformation when an user changes his identity information. The specific rules are as follows:

$$Identity_i' = (Address_i, ID_i', RpCoin_i) \tag{3}$$

where $Identity_i'$ represents the modified identity, and ID_i' represents the new ID value after the identity information is updated. Therefore, when an identity changes his personal information which will causes his ID to be changed, and the system will re-create the hash ID' based on the new identity information, while the old ID information is still maintained in the Blockchain. In this way, the attackers are unable to create multiple identities by personal information modification.

Another type of identity modification is the change of users' Ethereum address. When an user requests to change his Ethereum address, the system will also generate a new $Identity$, and the old address will remain stored in the Blockchain. In this way, the identity information and $RpCoin$ are transfered from the old identity to the updated one, avoiding the attackers hide their identity by changing their addresses. It is noted that the modification of an user's address requires the user provide his old address of the ID to ensure the legitimacy of the address modification operation. We use $Address_i'$ to represent the changed address, the new identity is defined as follows:

$$Identity_i' = (Address_i', ID_i, RpCoin_i) \tag{4}$$

B. Reputation Model

The reputation of a user in our system is an quantitative representation of his behavior credibility. When a user always behave truthfully and actively, the user's reputation should be high, and verse vice. Thus, the reputation score of a user in our system reflects the user behavior change with time, which is innovative in the literature.

With the help of the $RpCoin$ which is the token associated with the reputation tasks and incentive tasks, we propose a new concept $RpCoinDay$ which is an accumulated parameter capturing the total number of days a user holds $RpCoin$. Suppose a user has a certain number of $RpCoin$ at time t, then the user's $RpCoinDay$ increase by $RpCoin$ at time $t+1$ (the unit of time is day which can be customized as any minimal predefined time unit). In other words, a user's $RpCoinDay$ is a non-decreasing function of time, and it increases faster when the user has more $RpCoin$. As long as the $RpCoin$ of a user is positive, the user's $RpCoinDay$ will consistently increase day by day. The accumulation process of a user's $RpCoinDay$ is demonstrated as in Fig. 1.

Therefore $RpCoinDay$ does not only reflect the number of tokens that users holding, but also reflect the days that

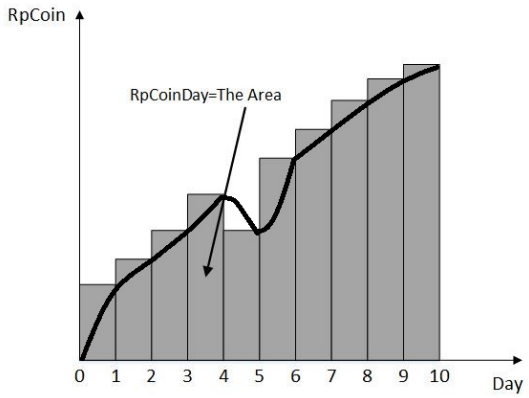

Fig. 1. The demonstration of $RpCoinDay$ calculation

user holds the tokens. However, the value of $RpCoinDay$ is still unable to fully reflect the user's reputation behavior. Let's consider the following three cases with the same value of the $RpCoinDay$:

- The user holds a fixed amount of $RpCoin$, and the user's $RpCoinDay$ increases linearly. As Line 1 in Fig. 2 shows that the $RpCoinDay$ increases uniformly with time, the slope of the Line does not change from time $t1$ to time $t2$, illustrating that the user has a fixed number of $RpCoin$ held during this time period, and thus infer that the user may not participate in the tasks or the user is inactive during the specific time period.
- The user holds a smaller amount of $RpCoin$ initially and obtain more and more $RpCoin$ with time. In this case, the user's $RpCoinDay$ increases convexly. As Line 2 in Fig.2 shows that the growth rate of $RpCoinDay$ increases with time. The slope of Line 3 increases gradually from time $t1$ to $t2$ and reaches its maximum at time $t2$. It indicates that the number of $RpCoin$ held by the user increases during this period, thereby inferring that the user may actively participate in tasks or actively work in the system during this period.
- The user hold a larger amount of $RpCoin$ at first and lose it gradually. In this case, the user's $RpCoinDay$ increases concavely. As Line 3 in Fig. 2 shows that the growth rate of $RpCoinDay$ decreases with time, the slope of Line 2 becomes smaller and reaches the minimum at time t2. It indicates that the number of $RpCoin$ held by the user has been reduced during this period, then it is inferred that the user may work negatively in this period.

The above three cases show that users may accumulate $RpCoinDay$ through different ways. The process of an user in accumulating $RpCoinDay$ reflects the behavior dynamics of users, which is further cased by the dynamics of the user's $RpCoin$.

When a user has a large number of $RpCoin$, its daily $RpCoinDay$'s increment will be great, even if the user does not participate in any tasks, and the user will be presented in

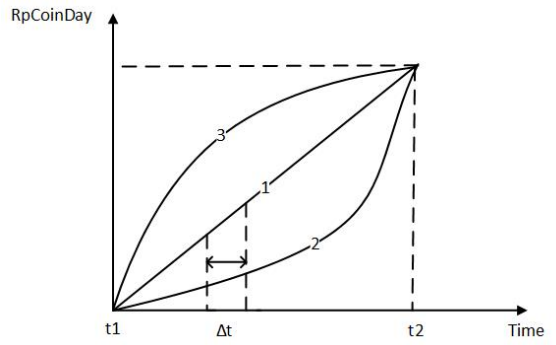

Fig. 2. Three type of user's reputation behavior

a good reputation status with the advantage of his $RpCoin$. In another case, the user $RpCoin$ increases considerably over a short period of time, causing $RpCoinDay$ increasing violently in a short time (several days). In both cases, although the user's $RpCoinDay$ value should be high, we can not explain the user is a reputable user. We believe that a reputable user should continue to participate in the system to actively complete the system tasks, and $RpCoin$'s increments should be maintained stable with time, that is considered as long-term stability.

Therefore, we propose another parameter *reputation fluctuation factor* which is denoted as Rpf to capture user behavior reflected in the user's $RpCoinDay$ accumulation process, and Rpf satisfies $0 < Rpf < 1$. In the above three cases, we denote Rpf_1, Rpf_2, and Rpf_3 for the three lines in Fig. 2, respectively. We believe that the reputation of the user represented by Line 2 is better than the user represented by Line 1 which is even better than the user represented by Line 3, as shown in Eq. (5).

$$Rpf_2 > Rpf_1 > Rpf_3 \tag{5}$$

With this inspiration of Eq. (5), we will discuss how to calculate Rpf. In our system Rpf is a time-dependent parameter, and Rpf varies with the user's $RpCoin$ dynamics. Assuming that we take the time interval for calculating Rpf as T, which is determined by the system according to the user characteristics. In the T time interval, a user i's $RpCoin$ change is expressed as $\Delta RpCoin_i$, which is discussed in the following three cases:

- $\Delta RpCoin_i = 0$, indicating that the number of RpCoins held by the user i in T time remains unchanged. We set $Rpf_i = 0.5$.
- $\Delta RpCoin_i < 0$, indicating that the number of RpCoins held by the user i in T time is decreased. We set $Rpf_i < 0.5$.
- $\Delta RpCopin_i > 0$, indicating that the number of RpCoins held by the user i in T time is increased. We set $Rpf_i > 0.5$.

Therefore, Eq.(5) could be always true.

Next, we discuss how to calculate Rpf_i for each user. Consider the $RpCoin$ change of each day in the period of T. Suppose, T is composed by n days, then the $RpCoin$ differences of the n days are denoted by Δr_{i1}, Δr_{i2}, ..., Δr_{in}. In a single day k, $\Delta r_{ik} > 0$ means that the user is $RpCoin$ increases, and verse vise. Then, the following equation is true.

$$\Delta RpCoin_i = \Delta r_{i1} + \Delta r_{i2} + \cdots + \Delta r_{in} \tag{6}$$

We define $\overline{X}_i$ as the mean value of $\Delta r_{i1}, \Delta r_{i2} \cdots \Delta r_{in}$.

$$\overline{X}_i = \frac{\sum_{k=0}^{n} \Delta r_{ik}}{n} \tag{7}$$

We define S_i as the standard deviation of $\Delta r_{i1}, \Delta r_{i2} \cdots \Delta r_{in}$.

$$S_i = \sqrt{\frac{\left(\Delta r_{i1} - \overline{X}_i\right)^2 + \left(\Delta r_{i2} - \overline{X}_i\right)^2 + \cdots + \left(\Delta r_{in} - \overline{X}\right)^2}{n}} \tag{8}$$

The standard deviation S_i reflects the fluctuation of the data $\Delta r_{i1}, \Delta r_{i2} \cdots \Delta r_{in}$. When user i's $RpCoin$ change is more consistent, the value of S_i should be smaller, and verse vise.

Note that the reputation is a relative concept, which indicates that the reputation of a user should also consider the behavior of other users. Therefore, in calculating Rpf_i, we involve the concept of deviation ranking score Rs_i. Considering the fact that users with the same standard deviation should be scored in the same way, we assign the Rs_i for each user i in two steps. Suppose there are m users in the system. In the first step, the users are ranked in ascending order, and we denote the user id of the user with the smallest S_i as S_1, and so on. In the second step, we give 1 to user 1, and 2 to user 2, and so on. Here, when the users with the same standard deviation, the score will sustain the same. In other words, if $S_i = S_{i+1}$, then $Rs_i = Rs_{i+1} = i$, which resulting the maximum value of the ranking score k is less or equal to m.

TABLE I
DEVIATION S_i RANKING SCORE TABLE

S_i **(in ascending order)**	S_1	S_2	$\cdots$	S_{m-1}	S_m
Rs_i	1	2	$\cdots$	k-1	k

For the three different cases of $\Delta RpCoin_i$, the value of Rpf_i will correspond to three different intervals, and the value of Rpf is as follows:

$$Rpf_i = \begin{cases} 0.5 - Map(Rs_i) & \Delta RpCoin_i < 0 \\ 0.5 & \Delta RpCoin_i = 0 \\ 1 - Map(Rs_i) & \Delta RpCoin_i > 0 \end{cases} \tag{9}$$

where the function $Map()$ aims to map the value of Rs_i between $(0, 0.5)$, which is specified as follows.

$$Map(Rs_i) = \frac{Rs_i}{k} \times 0.5 \tag{10}$$

From Eq. (9), we can see that when a user i's $RpCoin_i$ consistently increases, then the reputation fluctuation factor Rpf_i will approach to 1, and when the user's $RpCoin$ consistently decrease, Rpf_i will be close to 0.

Finally, based on the $RpCoinDay_i$ and Rpf_i, we establish the reputation score of the user i as follows.

$$R_i = RpCoinDay_i \times Rpf_i \tag{11}$$

It can be seen that the reputation score of a user is his $RpCoinDay$ discounted by the user's reputation fluctuation factor. Thus, it may happen that users with the same $RpCoinDay$ have different reputation score due to their behavior differences. In this way, the proposed reputation score can more effectively reflect users' behavior credibility.

C. $RpCoin$ Reward and Punishment

In this section, we specify how the $RpCoin$ are rewarded and penalized in the reputation and incentive tasks.

1) Reputation Task: In a reputation task, both the task publisher and the workers participating in the tasks are rewarded or penalized by a certain number of $RpCoin$. The rewarding or punishing operation is determined by the task result which is determined as follows.

$$TaskResult = \begin{cases} Approved & \sum\limits_{V_i=Agree} R_i \geq \sum\limits_{V_i=NotAgree} R_i \\ Rejected & \sum\limits_{V_i=Agree} R_i < \sum\limits_{V_i=NotAgree} R_i \end{cases} \tag{12}$$

For the task publisher, the system rewards the user with a certain number of $RpCoin$ when the final result of a task is "Approved", and a certain number of $RpCoin$ is penalized when the task result is "Rejected". During the voting process, the workers are not only required to vote between "Agree" and "Not Agree", but also provide a credit rating Cr in a predefined range. In our experiments, the range of a credit rating range is set as $\{1, 3, 5\}$. With the credit ratings with the "Agree" votes, the $RpCoin_{reward}^{Publisher}$ is calculated as follows.

$$RpCoin_{reward}^{Publisher} = \frac{\sum\limits_{V_i \text{is consistent with TaskResult}} Cr_i}{\sum\limits_{V_i \text{is consistent with TaskResult}} 1} \tag{13}$$

In other words, the number of $RpCoin$ rewarded to the task publisher is the average credit ratings given by the voters whose votes are consistent with the final result of the task.

When the task result is "Rejected", a fixed number of $RpCoin$ is deduced from the tasks publisher.

$$RpCoin_{punish}^{Publisher} = \alpha \tag{14}$$

The set of α should also fall in the range of the credit ratings, and in our experiments, we set $\alpha = 5$.

For the workers, when a voter's opinion is consistent with the TaskResult, then the worker should be rewarded with a certain number of $RpCoin$. Otherwise the workers are penalized with $RpCoin$. We set the total number of $RpCoin$ for a task is equal to the number of voters or workers participating in a reputation task. For the voters whose votes are inconsistent with the TaskResult, the total number of $RpCoin$ penalized is also equal to N. Thus, the total number of $RpCoin$ rewarded to the voters whose votes are consistent with the TaskResult is $2N$. Then, for each worker i, if his vote is consistent with the TaskResult, $RpCoin_{reward}^{i}$ will be rewarded, otherwise the worker i is penalized with $RpCoin_{punish}^{i}$.

$$RpCoin_{reward}^{i} = \frac{R_i}{\sum_{V_k=V_i} R_k} \times 2N \tag{15}$$

$$RpCoin_{punish}^{i} = \frac{R_i}{\sum_{V_k=V_i} R_k} \times N \tag{16}$$

It can be seen from the above equations that the rewarded $RpCoin$ obtained by the workers is in proportional to the worker's reputation R_i, which means that the higher of a worker's reputation, the more rewards will be obtained. When the worker's vote is inconsistent with from the final result, the punishment for the workers with higher reputation is also more serious.

2) Incentive Task: The final task result of an incentive task is also determined as that in a reputation task. Different from a reputation task, an incentive task involves three type of users: task publisher, workers, and the objective user (i.e. the worker who is proposed to be a negative worker or posting repetitive tasks).

For the task publisher and workers in an incentive task, the $RpCoin$ reward and punishment is the same as that in a reputation task. Next we will specify how to deal with the $RpCoin$ reward and punishment for the objective user in an incentive task. Since the incentive tasks either aim to reveal the negative workers or discover repetitive task publishers, it is unnecessary to reward the objective user. We only need to properly punish the objective user according to the task result in an incentive tasks.

- When the $TaskResult = Approved$, the objective user should be penalized as the negative statement in the incentive task is agreed by other workers.

$$RpCoin_{punish}^{ObjWorker} = Rpf_{publisher} \times 2\alpha \tag{17}$$

- When the $TaskResult = Rejected$, the objective user is not rewarded or punished.

It is noted that the punishment of an objective user is related with the reputation fluctuation factor of the task publisher, and the punishment for the objective user is more serious than that for the general task publisher as shown in Eq.(14).

V. EXPERIMENT

In order to verify the feasibility of our proposed identity authentication and reputation model, we conduct a set of experiments by building a simple system in a local Blockchain. For the reason of demonstration purpose, we do not consider many users in a large scale and the number of users considered in the experiments is 10. In the experiment, we use the Blockchain development environment Tesstrpc, the smart contract framework Truffle, and solidity programming language to write smart contracts. The table II shows the public Ethereum address of the 10 users which is automatically generated in the development environment.

TABLE II
USER ADDRESS INFORMATION TABLE

1	**0xaae163c5e4419d7d866ae9459c9adba6229d1ca5**
2	**0x34265c223f1daa46d017e23a7854698d50a9b682**
3	**0x3c0be237278422177fb472b902ea29d6c25c52f0**
4	**0xe06489af1a5e48dfde9dd0600bcb59b817c44bd6**
5	**0x4f34346e05257744adbf398240e8eb9eccff07d5**
6	**0x44ded7b95fca0b9fd3328d0072b1ffdc7b5a2fbe**
7	**0xbb42bb7bd39087abd7f2c92831cacf16dda3a191**
8	**0x74782c71e17468b8f837ea12efaee91116cbea45**
9	**0xe33c4563f6585389bd4bbe07ba75cdbf1eb941e6**
10	**0xc0edc57e717c958c6248cb0a55010a51b960cfae**

A. Identity Authentication

In the identity authentication model we propose that the creation of an identity requires the provision of unique physical information. The same physical information and the Ethereum public key address does not allow to create a second identity. We use three sets of input data to verify the identity authentication

1) Case 1: Try to crease a user with different identity information with any users in the dataset but has the same public key addresses with one of the existing users. The experimental data are shown as in Table III

TABLE III
CREATION A NEW IDENTITY WITH THE SAME ADDRESS

Serial	Identity	address	System Output	Block Number
1	name:zz	0x654f5661fbe35ec98df5968d2a2f06c0c74b2dfd	0xd08c5ef6683cb375a8b0e75d74a114da2a8f7ff7bbfe2ed3ad1ec1912b18266d	16
2	name:bj	0x654f5661fbe35ec98df5968d2a2f06c0c74b2dfd	NULL	16

Conclusion: From the experimental data shown in Table III, in the two sets of data input, although the identity information are different, the new identity failed to be created by not recorded in a block in the system. It indicates that the system can not create a new identity with the public key address the same as one of users already existing in the database. Therefore, our system can ensure that users of the same public address can not be created repeatedly.

2) Case 2: Try to crease a user with different identity information with any users in the dataset but has the same public key addresses with one of the existing users. The experimental data are shown in Table IV

Conclusion: The new identity creation is failure as the identity data is not recorded in block. Even though the new identity has different public address with any existing users, however, the system still rejects to create a new identity if the identity information is the same with one of the existing users in the system. It indicates that the users with the same identity information can not be created repeatedly.

TABLE IV
CREATION A NEW IDENTITY WITH THE SAME IDENTITY INFORMATION

Serial	Identity	address	System Output	Block Number
1	name:zz	0x654f5661fbe35ec98df5968d2a2f06c0c74b2dfd	0xd08c5ef6683cb375a8b0e75d74a114da2a8f7ff7bbfe2ed3ad1ec1912b18266d	16
2	name:zz	0xff1b0d9a5b4234130ba6d313778034a0efe71f09	NULL	16

3) Case 3: Try to create an identity with different identity information and different the public key address, and the experimental data are shown in Table V

TABLE V
CREATION A NEW IDENTITY WITH DIFFERENT IDENTITY INFORMATION AND DIFFERENT ADDRESS

Serial	Identity	address	System Output	Block Number
1	name:zz	0x654f5661fbe35ec98df5968d2a2f06c0c74b2dfd	0xd08c5ef6683cb375a8b0e75d74a114da2a8f7ff7bbfe2ed3ad1ec1912b18266d	16
2	name:lx	0xff1b0d9a5b4234130ba6d313778034a0efe71f09	0xeee86070ce470993cc1e1213bd3c3a146b92feab713569352b4a43b90353d0f0	16

Conclusion: The identity creation is successful by recording the new identity in a block. It indicates that the system is able to create identities with unique identity information and unique address, which is consistent with the objective of the identity authentication.

The above three experimental cases demonstrate that once an identity information and address are binded successfully through creating an identity, the system will not allow the same identity information or address to be utilized in any other identity.

B. Identity modification

According to our identity authentication model, if a user needs to modify his address or identity information, it should provide his identity ID or address, respectively. We will use two cases to verify this property of the identity modification model.

1) Case 1: Try to modify an identity's identity information, by providing his true original ID and wrong ID, respectively.

Conclusion: As we can see in Table VI, when the provided ID information is correct, the system successfully executes the modification operation. On the contrary, the system fails

TABLE VI
IDENTITY INFORMATION MODIFICATION WITH TRUE AND FALSE ID

Serial	ID	ID accuracy	System out
1	0xd08c5ef6683cb375a8 b0e75d74a114da2a8f7ff7 bbfe2ed3ad1ec1912b18266d	True	Success
2	0xd08c5ef6683cb375a8 b0e75d74a114da2a8f7ff7 bbfe2ed3ad1ec1912b182660	False	Failure

to perform identity modification operation. It indicates that the identity information modification requires the provision of correct ID information.

2) Case 2: Try to modify the address of an existing user by providing (1) wrong ID information but correct original address; (2) correct ID information but wrong original address; (3) the correct ID information and the correct original address. The experimental results are shown in Table VII

TABLE VII
ADDRESS MODIFICATION

Serial	accuracy	address	System Output	Result
1	address is right	0x654f5661f be35ec98df5 968d2a2f06c 0c74b2dfd	0xd08c5ef6683 cb375a8b0e75d74 a114da2a8f7ff7b bfe2ed3ad1ec191 2b182660	false
2	ID is right	0x654f5661f be35ec98df5 968d2a2f06c 0c74b2dff	0xd08c5ef6683 cb375a8b0e75d74 a114da2a8f7ff7b bfe2ed3ad1ec191 2b18266d	false
3	both right	0x654f5661f be35ec98df5 968d2a2f06c 0c74b2dfd	0xd08c5ef6683 cb375a8b0e75d74 a114da2a8f7ff7b bfe2ed3ad1ec191 2b18266d	true

Conclusion: It can be seen from Table VII that the identity address can only be modified when both the public address and identity information are correct.

C. Rpf Validation

In this experiment, we simulate 10 user's behavior within 10 days, and the parameter T is also set as 10 days. We then observe how the reputation fluctuation parameter is impacted after the 10 days. We consider the following five types of user behavior:

1) Users' $RpCoin$ remains unchanged in the 10 days, as user 1 and 10 in Table VIII, which is explained in Case 1.
2) Users' $RpCoin$ increases evenly in the 10 days, as user 5 in Table VIII, which is explained in Case 2.
3) Users' $RpCoin$ increases non-uniformly in the 10 days, as user 6,7, and 8 in Table VIII, which is explained in Case 2.
4) Users' $RpCoin$ decreases evenly in the 10 days, as user 2 in Table VIII, which is explained in Case 3.
5) Users' $RpCoin$ decreases non-uniformly in the 10 days, as user 3,4,and 9 in Table VIII, which is explained in Case 3.

The standard deviation S_i and Rpf_i is calculated in the last column of Table VIII.

TABLE VIII
USER'S REPUTATION BEHAVIOR

Users	RpCoin	RpCoin Changes	S_i	Rpf_i
1	100	100-100	0	0.5
2	200	200,190,180,170,160,150, 140,130,120,110,100	0	0.5
3	200	200,190,170,160,155,120, 115,110,105,103,100	93.8	0.15
4	200	200,150,145,135,132,129, 125,119,112,111,100	186.6	0.1
5	50	50,55,60,65,70,75, 80,85,90,95,100	0	0.95
6	50	50,51,54,59,65,70, 74,76,80,84,100	15.4	0.75
7	70	70,71,76,79,83,86, 89,90,92,94,100	2.4	0.8
8	20	20,24,27,35,45,55, 67,71,78,80,100	26.56	0.6
9	500	500,478,364,256,200, 187,156,143,132,110,100	1500.4	0
10	100	100,150,149,176,132, 190,50,80,45,90,100	282	0.5

1) Case 1: $\Delta RpCoin = 0$

We can observe that the Rpf for user 1 and user 10 is 0.5, which is determined by Eq (9). For user 1, his $RpCoin$ is always the same, which indicates that the user does not participate in tasks and the system is reasonable to assign the middle value of Rpf for this user. For user 10, his $RpCoin$ is changed unpredictably but still sustain the same $RpCoin$ in the 10th day with the first day. Due to his behavior uncertainty, the system also assign $Rpf = 0.5$ for this user.

Conclusion: Rpf is always 0.5 when the number of $RpCoin$ held by the user is unchanged in the predefined T period.

2) Case 2: $\Delta RpCoin > 0$

The users 5, 6, 7, and 8 belong to this case. We can observe that their reputation fluctuation factor is always larger than 0.5, as their $RpCoin$ increases comparing Day 10 to Day 1. The user 5 achieves the highest Rpf, as the user's $RpCoin$ stably and consistently increase, indicating that the user maintains a good and reputable behavior in this ten days. On the contrary, the greater of the user behavior variance is as user 8, the lower his Rpf will be, indicating that the user's reputable behavior is still not stable in this period.

Conclusion: When $\Delta RpCoin > 0$, the range of Rpf is $(0.5, 1)$, which is consistent with our expectation in the model analysis. The smaller the incremental variance is, the larger of the value of Rpf should be, and verse vice.

3) Case 3: $\Delta RpCoin < 0$

The user 2, 3, 4, and 9 belong to this case. We can observe that their reputation fluctuation factor is always smaller than 0.5, as their $RpCoin$ decreases comparing Day 10 to Day 1. The user 2 achieves the highest Rpf in this case, as the user's $RpCoin$ stably and consistently decrease, indicating that the user maintains a predictably bad behavior in this ten days, the reputation of user 2 is negative impacted as his accumulated $RpCoinDay$ is slower and slower with his $RpCoin$ consistently decreasing. Meanwhile, the greater of the user behavior variance is as user 9, the lower his Rpf will be, which is the lowest value 0

Conclusion: When $\Delta RpCoin < 0$, the range of Rpf is $(0, 0.5)$. The smaller the decremental variance is, the larger the value of Rpf will be, and verse vise. It is also fit for our expectation.

D. Reputation Task

In this experiment, we simulate how a reputation task is processed in our system, and how users are rewarded or punished. We consider two cases with the task result being "Approved" and "Rejected", respectively.

1) Case 1: TaskResult="Approved": The user 1 acts as the task publisher in the task, and other 9 users act as worker. In this case, most people vote "Agree" which overweight "Not Agree". Thus, the task result is "Approved". The credit rating of each vote is listed in the 5th column of Table IX, and the rewarded or punished $RpCoin$ is listed in the 6th column of Table IX where the presented data omits decimal fractions smaller than 0.5 and count all others including 0.5 as 1.

TABLE IX
A REPUTATION TASK WITH TASKRESULT AS "APPROVED"

Users	Votes V_it	Rpf_i	R_i	Cr_i	$\Delta RpCoin$
1 (Task Publisher)	NULL	0.5	50	NULL	+2
2	Agree	0.5	50	3	+3
3	Agree	0.15	15	3	+1
4	Not Agree	0.1	10	0	-1
5	Agree	0.95	95	1	+6
6	Not Agree	0.75	75	0	-4
7	Agree	0.8	80	1	+5
8	Agree	0.6	60	1	+4
9	Not Agree	0	1	0	-1
10	Not Agree	0.5	50	0	-3

Conclusion: From the experimental data we can see that when a reputation task is "Approved", the task publisher will be rewarded. The amount of rewarded RpCoin depends on the average rating score given by the voters whose vote is "Agree". The workers who vote "Agree" will receive rewards, and the amount of the reward will be related to their reputation. On the contrary, the workers who vote inconsistently with the task result are punished in proportional to their reputation.

2) Case 2: TaskResult="Rejected": In this case, the user 1 also performs as the task publisher and other workers act as workers. In this case, most people vote "Not Agree" so that the task result is "Rejected". The credit rating of each vote is listed in the 5th column of Table X, and the rewarded or punished $RpCoin$ is listed in the 6th column of Table X.

TABLE X
A REPUTATION TASK WITH TASKRESULT AS "REJECTED"

Users	Votes V_i	Rpf_i	R_i	Cr	$\Delta RpCoin$
1 (Task Publisher)	NULL	0.5	50	NULL	-5
2	Not Agree	0.5	50	3	+3
3	Agree	0.15	15	0	-1
4	Not Agree	0.1	10	5	+1
5	Not Agree	0.95	95	1	+6
6	Not Agree	0.75	75	3	+5
7	Agree	0.8	80	0	-5
8	Agree	0.6	60	0	-3
9	Not Agree	0	1	5	+1
10	Not Agree	0.5	50	1	+3

Conclusion: from the experimental data we can see that when the task result is "Rejected", the task publisher will receive the penalty which is predetermined by the system as $\alpha = 5$. The workers who vote "Not Agree" will receive rewards, and the amount of the reward will be related to their reputation. On the contrary, the workers who vote "Agree" which is inconsistent with the task result, will be punished in proportional to their Reputation.

E. Incentive Task

In this experiment, we simulate how incentive tasks are processed in our system and how the task publisher, participating workers, and objective user are rewarded and punished. We will consider two cases with the task results as "Approved" and "Rejected" , respectively.

1) Case 1: The TaskResult="Approved": In this case, user 1 acts as the task publisher, and user 2 is the objective user, and the other 8 workers act as voters. As can be seen from the votes listed in Table XI, the vote "Agree" take greater weight than "Not Agree", which result the incentive task result to be "Approved".

TABLE XI
AN INCENTIVE TASK WITH TASKRESULT AS "APPROVED"

Users	Votes V_i	Rpf_i	R_i	Cr_i	$\Delta RpCoin$
1 (Task Publisher)	NULL	0.5	50	NULL	+3
2 (Objective User)	NULL	0.5	50	NULL	-5
3	Agree	0.15	15	3	+1
4	Agree	0.1	10	1	+1
5	Agree	0.95	95	3	+5
6	Agree	0.75	75	5	+4
7	Agree	0.8	80	3	+4
8	Agree	0.6	60	1	+3
9	Not Agree	0	1	0	-1
10	Not Agree	0.5	50	0	-8

Conclusion: According to the incentive task model we know that the incentive task is used to punish the negative users in the system, so when the incentive task result is "Approved", the objective users will be punished. In order to motivate the

task publisher to actively find the negative users in the system, the task publisher is rewarded.

2) Case 2 TaskResult="Rejected": In this case, user 1 also acts as the task publisher, and user 2 is the objective user, and the other 8 workers act as voters. From the votes of voters in Table XII, we can see that the vote "Not Agree" dominates "Agree", which results in the task result as "Rejected". The task publisher is penalized for his wrong statement, and the objective user is not impacted.

TABLE XII
AN INCENTIVE TASK WITH TASKRESULT AS "REJECTED"

Users	Votes V_i	Rpf_i	R_i	Cr_i	$\Delta RpCoin$
1(Task Publisher)	NULL	0.5	50	NULL	-5
2(Objective User)	NULL	0.5	50	NULL	0
3	Agree	0.15	15	0	-1
4	Not Agree	0.1	10	1	+1
5	Not Agree	0.95	95	5	+7
6	Not Agree	0.75	75	1	+5
7	Agree	0.8	80	0	-4
8	Agree	0.6	60	0	-3
9	Not Agree	0	1	1	+1
10	Not Agree	0.5	50	3	+3

Conclusion: We can see in Table XII that the incentive task for the task publisher has a positive encouragement, if the task publisher works hard and finds the negative workers, he will be rewarded. We have reason to believe that the involvement of incentive tasks will help the system attract more active participants and meanwhile avoid negative workers.

VI. DISCUSSION

In this paper, we have proposed an identity management and reputation system based on blockchain technology, and our system is more secure in management virtual identity in the following three aspects. First of all, the proposed model is executed in a complete distributed mode, which outperforms the traditional centralized identity management in the perspective of information security. Secondly, the system prevents the possibility of creating multiple identities for the same physical entity, which is potential to address the challenging Sybil attack existing in most online systems [3]. In the proposed system, a physical entity can only create a single virtual identity due to the feature of blockchain's immutability. To further control the identity creation, we also design the identify modification procedure which avoids the of the white-washing attack [6]. Thirdly, the unique identity is associated with reputation information which captures the user behavior dynamics in the process of completing reputation tasks and incentive tasks. In the two types of tasks, the users perform voting between "Agree" or "Not Agree" with the statement in the tasks, and their reputation is constructed meanwhile. It is promising to consider votes in a more complex manner which will be investigated in our future work.

VII. CONCLUSION

In this paper, we propose a Blockchain-based identity management system. We introduce the working principle of the system and discuss the identity authentication model and reputation model in detail. In the identity authentication model, we clarify how to bind the BlockChain public key address to the entity user information and how to modify the binding identity. In the reputation model, we propose the concepts of $RpCoinDay$ and reputation fluctuation factor Rpf, making a user's reputation reflects the user behavior change with time. Finally, we demonstrate the feasibility of our system through conducting sets of experiments, and the experimental results validate the proposed module.

In future work, we plan to conduct large scale real-data based experiments in public Ethereum to further evaluated the proposed system and improve it.

ACKNOWLEDGMENT

This research is partially supported by National Natural Science Foundation of China under Grant No.61572123, No. 61402097 and No.61602102; the National Science Foundation for Distinguished Young Scholars of China under No. 71325002; the Natural Science Foundation of Liaoning Province of China under Grant No.20170540319, No.201602261; and the Fundamental Research Funds for the Central Universities under Grant No. N162410002, N161704001, N151708005, N161704004, N151704002.

REFERENCES

[1] Joerg Abendroth. Identity management system. *University of Nairobi*, 2014(9):21, 2014.

[2] Vitalik Buterin. A next-generation smart contract and decentralized application platform. *Available: https://www.ethereum.org/pdfs/EthereumWhitePaper.pdf/*, 2014.

[3] John R. Douceur. The sybil attack. In *Proceedings of the First International Workshop on Peer-to-Peer Systems*, pages 251–260, 2002.

[4] Ji Fang, Cao Yan, and Chen Yan. Centralized identity authentication research based on management application platform. In *Proceedings of the First International Conference on Information Science and Engineering*, pages 2292–2295, 2009.

[5] M B Ferreira and Kenji Alonso. Identity management for the requirements of the information security. pages 53–57, 2014.

[6] Weimin Luo, Jingbo Liu, Jiang Xiong, and Ling Wang. Defending against whitewashing attacks in peer-to-peer file-sharing networks. In *Proceedings of the 4th International Conference on Computer Engineering and Networks*, pages 1087–1094, 2015.

[7] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system[online]. available: https://bitcoin.org/bitcoin.pdf, 2008.

[8] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, 2016.

[9] Pete Rizzo. Blockchain identity startup shocard raises 1.5 million. http://www.coindesk.com/blockchain-identity-startup-shocard-1-5-million/, July 2015.

[10] Melissa Jun Rowley. Aiding refugees? yes, the blockchain can do that and more. http://www.huffingtonpost.com/melissa-jun-rowley/aiding-refugees-yes-the-b_b_8149762.html, September 2015.

[11] Sebastian. What is Onename? how does the web application work? https://onename.zendesk.com/hc/en-us/articles/202288932-What-is-Onename-How-does-the-web-application-work, May 2016.

[12] Melanie Swan. *Blockchain: Blueprint for a New Economy*. O'Reilly Media, Inc., 2015.

[13] Yong Yuan and Yuefei Wang. Development status and prospect of blockchain technology. *Journal of Automation*, 42(4):481–494, 2016.