International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Bittrack-A Decentralized Trust Based Identity and Access Management Approach

Garv Juneja[1], Raghav Naswa[2]

Submitted: 27/01/2023 **Revised:** 15/03/2023 **Accepted:** 07/04/2023

Abstract: In this paper, a decentralized Identity and Access Management (IAM) application named “Bittrack” has been proposed. The advent of Blockchain technology as a juggernaut in the world of innovations has tagged on with itself revolutionary ways of handling identity on digital platforms. In this paper we propose a high-level architecture of the implemented dapp written in Vuejs that interacts with the Ethereum blockchain on a public test net. With an aim towards building a better model our system follows a more human centric approach for managing of identities using its own consensus mechanism. Later into the paper, we have shown how the reputation of the user can uncover the various undiscussed dynamics of an ideal identity management system. Therefore, the requirements for scalability of our system are far less stringent in comparisons to the current proposed models and thereby making our system to function as an independent model to manage the identities of the user in a highly secure and an efficient manner.

Keywords: *Blockchain, Identity Management, Access Management, Security*

1. Introduction

With the emergence of Bitcoin in 2009 [1] , blockchain was only know to the exchange of cryptocurrencies but later evolved into main stream technology with invention of Ethereum and other similar blockchain frameworks. This technology had revolutionized the concept of traditional centralized solution to a rather more complex but a creative way of transmitting and storing information among the nodes of the network in a transparent and secure manner. Blockchain is essentially a shared, immutable public leger that facilitates the recoding of transactions and tracking its state in a peer-to-peer network. By using the time-stamp of transactions, blockchain provides verified proofs for existence of a transaction in the distributed leger using the underlying cryptographic primitives and digital signatures which makes sure that these transactions are secure and computationally verifiable at any given instance. The invariableness and consensus mechanism of blockchain eliminate the role of intermediaries and appear to appropriate solutions for distributed environment. Below are some steps carried out to initiate a blockchain transaction:

1. A transaction is requested and initiated.
2. The transaction authenticated and is validated against some specific criteria which includes gas limit, time stamp, sender account balance etc.
3. The transaction then enters into a pool of unconfirmed transactions.

1Systems engineer at TCS, Department of Computer science and Engineering
Maharaja Surajmal Institute of technology, Guru Gobind Singh Indraprastha University, Janakpuri, New Delhi
Email ID: garvjuneja98@gmail.com
2Graduate Student, Department of Computer Science and Software Engineering, University of Washington
Email ID: raghavnaswa@hotmail.com

4. The miner picks up these set of verified transactions from the above-mentioned pool and creates a block.

5. The miners then solve the puzzle which is regarded as Proof of work (PoW) to add a newly created block onto the blockchain.

The new block is then broadcasted into the network and is validated before it is added to the leger. The update is distributed among the network where every node adds the new block to its copy of the leger. The blocks which fail to get added are called as uncle blocks. This shows how blockchain maintains transparency and at the same time providing anonymity to its users by allowing them to create pseudo – anonymous transactions without the need of disclosing their personal information.

1.1. *Motivation: Need for identity management and blockchain*

Digital identity is essential in the realm of information technology since the digital society is interconnected and dispersed online [2]. We need a system that can identify who the expected users are and authorize their name, address, and personality because we are moving toward the usage of digital technology [3][4]. A gradual shift from papers to computer for storing identity has surely been a huge leap but the leap still fails to bridge the gap between the reality and the expectations from the system. A leap that, in 2013, resulted in an unprecedented identity theft of nearly 3 billion users of the then one of the biggest tech companies in the world. Digital ID verification confirms that people are the ones who claim to be in the online system. Title verification and sensitivity protection details are the key to being honest with ownership management. Users should change their own details (such as credentials) and organizations in exchange of services. Overcoming theft, misuse or to manage this data in a central way, services providers need to provide multi-item verification and ongoing ID management system problems.

Traditional Identity Management(IdM) systems prioritize Service Provider (SP) convenience of use over that of the users [5].Outside the middle road, combined conditions provide access to multiple sites with the same guarantees. However, data management and ownership is still in the hands of the service provider.

The aforementioned statistics point out that the Blockchain technology is yet to be realized to its fullest potential. This paper, therefore, seeks to take the leap forward and bridge the thence prevailing gap. Also, the paper aims to expand the contemporary view on the use of the Blockchain. The study also entails the use of a prototype built to exemplify the solution this research paper proposes.

1.2.*Organization of the paper*

The remaining paper is presented as: Section 2 presents the related work and its challenges. Section 3 presents the background of IdM and blockchain technology. Section 4 presents the proposed methodology. Section 5 presents an experiment with a working of proposed trust equation. Section 6 describes a comparative study. Section 7 discusses the results and finally Section 8 concludes the paper with the future scope.

2. Related Work

A number of proposals in the field of blockchain based identity management solutions have been proposed. The authors in [6] have proposed a decentralized model combined with an off-chain blockchain storage using key value storage with an encrypted raw data that is the blockchain consisted of only hash pointers to the data. In their work, using the off-chain repository which can be any online data storage (a cloud storage database) enables the users to manage his data independently using the symmetric key encryption without interacting with the blockchain. This concept is overall appealing, but fails to in cooperate to much dependency on off chain repositories thereby making it more prone to single node failures. The authors in [7] have proposed a blockchain based identity management using learning analytics. Their model briefly describes on how a technology layer can embed automated checking rules for data privacy and data sharing. In their work, smart contracts are used to manage the access levels according to the agreements between the two parties and at the same time secure the learning data. However, their work suffers from a major drawback as the major concern for the identity and access management is scalability. On the other hand, our model has been inspired by the consensus mechanism of the blockchain which enables the users to have a trust factor which would enhance the level of scalability. Uport [8] a user's uPort mobile application creates a new asymmetric key pair and sends a transaction to Ethereum that creates an instantiation of a controller that contains a reference to the newly created public key, this leads to the creation of proxy that contains reference to the address of just created control contract and the address of this proxy is composed of uPort identifier of a user. In healthcare sector, another form of identity management is proposed in [9] which enables patient data sharing and incentives for medical researchers to sustain the system. This platform is built on Ethereum and the smart contracts are used as to implement the basic principle of ownership of data, Information integrity and permission by containing the meta data of the above-mentioned aspects. A systematic literature mapping of identity management using blockchain has been done in [10]. The authors in [11] have discussed the vulnerability and mitigation techniques to secure sensitive data in blockchain applications

3. Background

3.1.*Identity and Access Management*

An individual in cyberspace creates a digital identity, which is an online identity. Digital ID verification confirms that people are the ones who claim to be in the online system. Title verification and sensitivity protection details are the key to being honest with ownership management. Users should change their own details (such as credentials etcetera) and organizations in exchange of services. Overcoming theft, misuse or to manage this data in a central way, services providers need to provide multi-item verification and ongoing ID management system problems. Outside the middle road, combined conditions provide access to multiple sites with the same guarantees. However, data management and ownership it is still in the hands of the service provider. IdM on the Internet still relies on what Cameron called a decade ago a "patchwork of identity one-offs comprising several types of IdM systems that are restricted to specific domains and do not interact much with one another [12][13].

The technology Blockchain used for identity and management access confers upon the users a conceivable solution to the obstacles presents in the centralized way of storing data and credentials through warranting users to save the data on the blockchain rather than stockpile the same on manipulatable servers. Particulars once deposited on the chain gets cryptographically secured and cannot be altered or manipulated once it has been tacked on, hence making colossal data breaches a thing of the past. Blockchain, therefore, marks the advent of a revolution in the method of accessing and storing digital identities.

3.2. *Blockchain Technology*

Satoshi Nakamoto first introduced the idea of blockchain in his white paper in 2008. In the white paper, it was explained how peer-to-peer networking may be used to share digital currency in an untrustworthy setting without the need for a central authority. In 2009, Bitcoin became the first application of this technology. It was a decentralized currency that incorporated the idea of public key cryptography with the proof of work consensus algorithm. Through decentralized transactions that take place over a peer to peer network, blockchain assures trust, data security, and integrity. A distributed ledger that is shared among a network of computers makes up this system. Nodes are the components of the network that are actively engaged. The distributed ledger is encrypted to protect the confidentiality of the nodes' information. A blockchain is just a chain of nodes called blocks, with the first block being referred to as a genesis block and lacking any parent blocks. The blockchain block contains a record of all recent network transactions that have taken place. A unique number known as the block's hash is present in each block of the blockchain. Because of how the hash is calculated, it is challenging to reverse engineer. Maintaining the immutability of the blockchain record is a critical function performed by the hash in the blockchain. If all of the network's nodes agree to it, a blockchain transaction is considered to be genuine. A procedure known as the consensus mechanism completes this approval task. The literature has a variety of consensus mechanisms [14]. The chain is known as a blockchain because it is always expanding. The main characteristics of blockchain are represented in Fig 1.

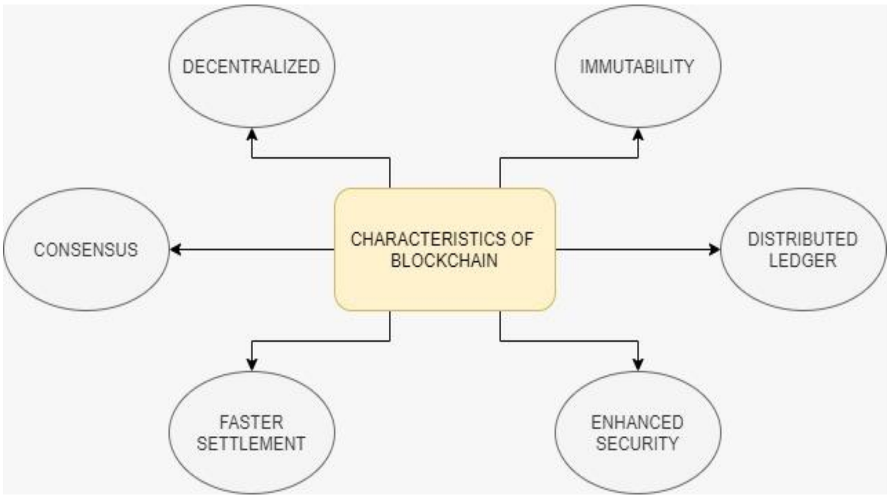


Fig. 1. Blockchain Characteristics

4. Proposed Methodology

The BitTrack Identity Management system is a blockchain-based solution that provides two primary modules: reputation management and identity authentication. As shown in Figure 2, the system is built on the Ethereum blockchain and is designed to ensure that each real-world entity has only one virtual identity in the system. The identity authentication module uses Ethereum public key addresses to enable identity correlation, while the reputation management module monitors an identity's behavior within the system.

To safeguard the security of identity-related data, smart contracts are used to write authentication and credit rules. The system stores data using the blockchain in a completely decentralized fashion, ensuring the protection of reputation-related data. When a user meets the requirements for contractual execution, the contract is automatically carried out, and the data is suitably written or updated. This approach guarantees the protection of reputation-related data.

The system's repudiation model allows users to upvote or downvote an identity, increasing or decreasing the trust score of that identity, respectively. The model ensures complete transparency of the identity for a user, enabling third-party services to examine the identity of a particular user only after sending a request to the user through a blockchain transaction. The user has the right to accept or reject the request, and if the user accepts the request, they can keep track of when the identity was viewed by the third party. The IPFS server sends a notification to the authenticator with the details of the viewer of the identity.

The selective disclosure feature of the system allows users to disclose only the information that the recipient needs to process the data, adhering to the pillars of self-sovereign identity defined in Decentralized and Self-Sovereign Identity [13]. Additionally, users can view the recent changes made by other identities by fetching the details of the blockchain transaction used to change that identity information.

It's worth noting that when a verified identity makes a change, its associated trust score goes down. The list of attesters that previously verified that identity must reapprove the changes to maintain the trust score. This approach ensures that the reputation management module continuously monitors an identity's behavior and encourages responsible behavior within the system. From a user's perspective, they retain ownership of their identity, and our model serves as a single sign-on solution that enables identity verification without the need for individual verifiers to store and verify identities on their end. As a result, users can enjoy a more secure and streamlined experience while maintaining control over their personal information.

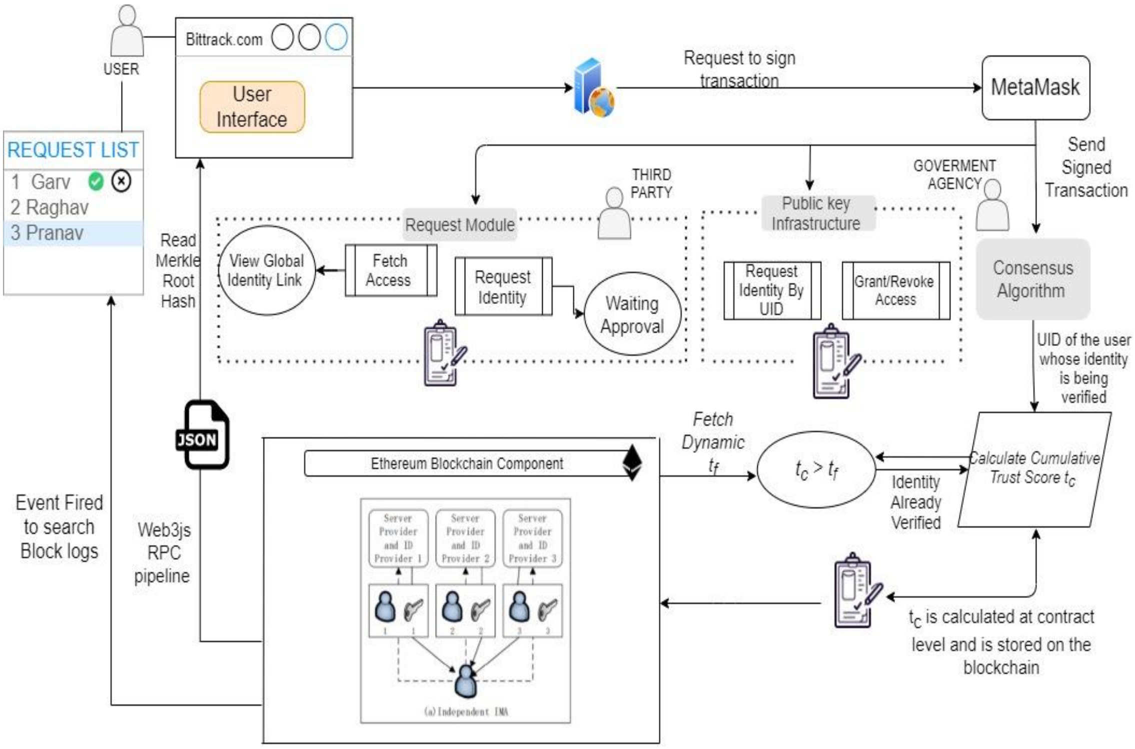

USER
Bittrack.com
User Interface
REQUEST LIST
1 Garv
2 Raghav
3 Pranav
Request to sign transaction
MetaMask
Send Signed Transaction
THIRD PARTY
GOVERMENT AGENCY
Request Module
Public key Infrastructure
View Global Identity Link
Fetch Access
Request Identity
Waiting Approval
Request Identity By UID
Grant/Revoke Access
Consensus Algorithm
Read Merkle Root Hash
UID of the user whose identity is being verified
JSON
Ethereum Blockchain Component
Server Provider and ID Provider 1
Server Provider and ID Provider 2
Server Provider and ID Provider 3
(a) Independent IMA
Fetch Dynamic t_f
$t_c > t_f$
Identity Already Verified
Calculate Cumulative Trust Score t_c
Event Fired to search Block logs
Web3js RPC pipeline
t_c is calculated at contract level and is stored on the blockchain

Fig. 2. Proposed BitTrack Model

4.1. *Trust Model*

The trust model has been drawn in the form of a directed graph which best suits the web of trust model that we wish to achieve. The model elements can be divided into 2 types the vertices and the edges.

Vertices and Edges:

Node U – User: This node is created when the user is begin registered and he/she goes to registration page. The system identifies the user by fetching the real time data from the Ethereum blockchain corresponding to the current wallet holder.

Fig. 3. User Node U

Node I: Identity: Identity created when user uploads his/her identity to IPFS.

Fig. 4. Identity Node

Edge IU: Formed when user copies generated hash to his/her details and registers.

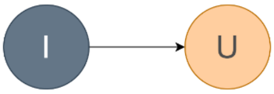


Fig. 5. Representation of a user node in web of trust chain

Verification V: A user can create a verification node that is used to verify an identity. User may create multiple verification nodes but one verification can only point to a single identity. Therefore, the maximum number of verification nodes per user is the total identities the user has access to.

Fig. 6. A verification node

Edge UV & VI: U and I have a one-to-one relation i.e., for each user there exists a bounded identity which is stored in the form of a hash which represents the location of the id on the IPFS server.

V_o = set of vertices

$= I \cup U \cup V$

I, U and V are disjoint, therefore

$$I \cap U \cap V = ø$$

The 3 kinds of edges can be represented as follows:

IU: This edge represents the relation between the user and his identity. A user can only have one identity and a single identity cannot be assigned to two users.

$$\left\{ IU = (I, u) | (i \epsilon I) \wedge (u \epsilon U) \wedge (\nexists (i, x) \epsilon IU : x \epsilon U \wedge u \neq x) \right\} \quad (1)$$

VI: This edge shows the relation between the verification node and he identity node. It is created when a user verifies the identity of another user. A single identity node can be verified by multiple users but one verification node can only verify a single identity.

$$\left\{ VI = (v, i) \ | \ (v \epsilon V) \ \wedge \ (i \epsilon I) \ (\nexists (v, z) \epsilon VI : z \epsilon I \ \wedge \ i \neq z) \right\} \quad (2)$$

UV:

This edge is also created when a user verifies the identity of another user. A single user can verify multiple identities. A single user can be associated with multiple verifications but each verification can point only to a single user.

$$\left\{ UV = (u, v) \ | \ (u \epsilon U) \ \wedge \ (v \epsilon V) \ \wedge (\nexists (y, v) \epsilon UV : z \epsilon U \ \wedge \ y \neq i) \right\} \quad (3)$$

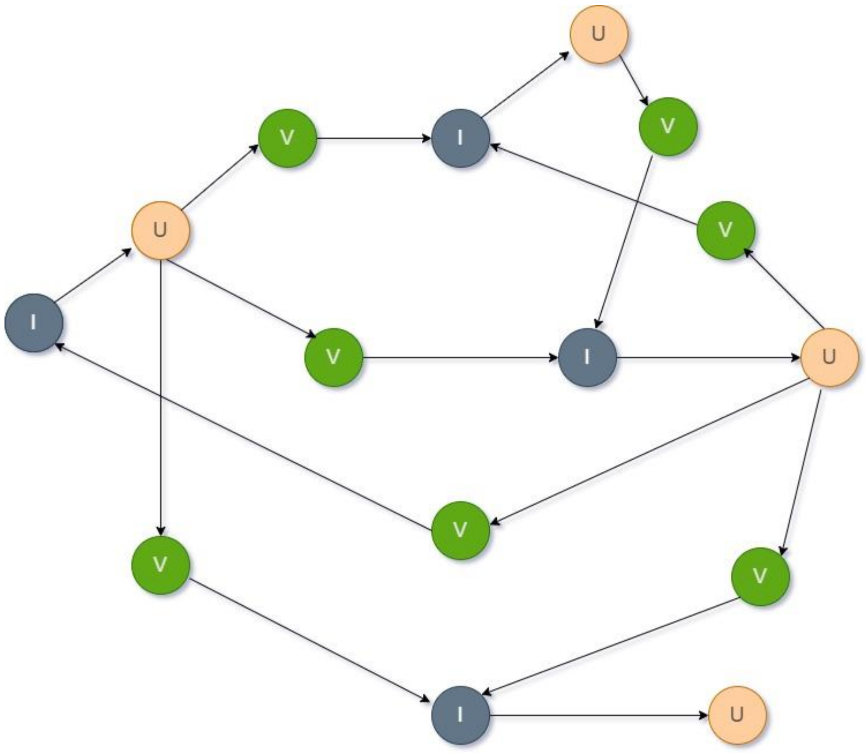


Fig. 7. Web of Trust Model Diagram of 4 users

4.2. *Authentication*

Each individual uses his own Ethereum account to interact with the system. A new user would be identified as a guest and can view the current state of the blockchain. Based on the public address each user is granted a unique temporary guest account. We suggest an identity solution that provides a virtual identity and real identity mapping connection, in which a single virtual identity is allocated a unique public key address, and any real identity may and only can have that virtual identity. The user registers the identity thus, creating the ‘I’ vertex which is linked to unique virtual identity U during its creation.

(1) Creation of user: A new user registers himself after generating an IPFS hash which is associated with his/her identity along with other necessary details. Once the user is successfully registered the system allocates a unique ID to the user which would be used as system reference for the user.

We use the following piece of code to map the user's wallet address with the user ID:

```
mapping (address => uint) public usersIds;
```

(2) Modification of Identity:

We fetch the current user bittrack account using his wallet address which is mapped to his unique ID

```
userId = usersIds[_wAddr]
```

Once the account is fetched changes can be made to it. If an identity's personal information changes, his ID will change as well, and the system will re-create the hash ID' using the new identity information while keeping the old ID information in the Blockchain.

The alteration of a user's Ethereum address is another sort of identity mutation. When a user asks that his Ethereum address be changed, the system creates a new Identity while keeping the old address on the Blockchain. The identification information is transferred from the old identity to the new one in this manner, preventing attackers from hiding their identities by altering their addresses. It should be noted that changing a user's address necessitates the user providing his former ID address in order to guarantee the legality of the address change. The modified address is represented by Address I and the new identity is specified as follows:

$$\text{Identity}_i = (\text{Address}_i, \text{ID}_i, \text{account}_i)$$

(3) Verification: For every identity$_i$ that is a part of the set Identity *i* ϵ I(identity set), there remains a degree of trust that binds to the address of the identity uploaded on the blockchain. The more the identity *i* garners trust through the system from other benefactors, the more the verification score of the identity *i* becomes. This calculation of the trust factor would be explained in the upcoming section.

identity$_a$ - - > identity$_i$ (Address$_i$, ID$_i$, account$_i$)

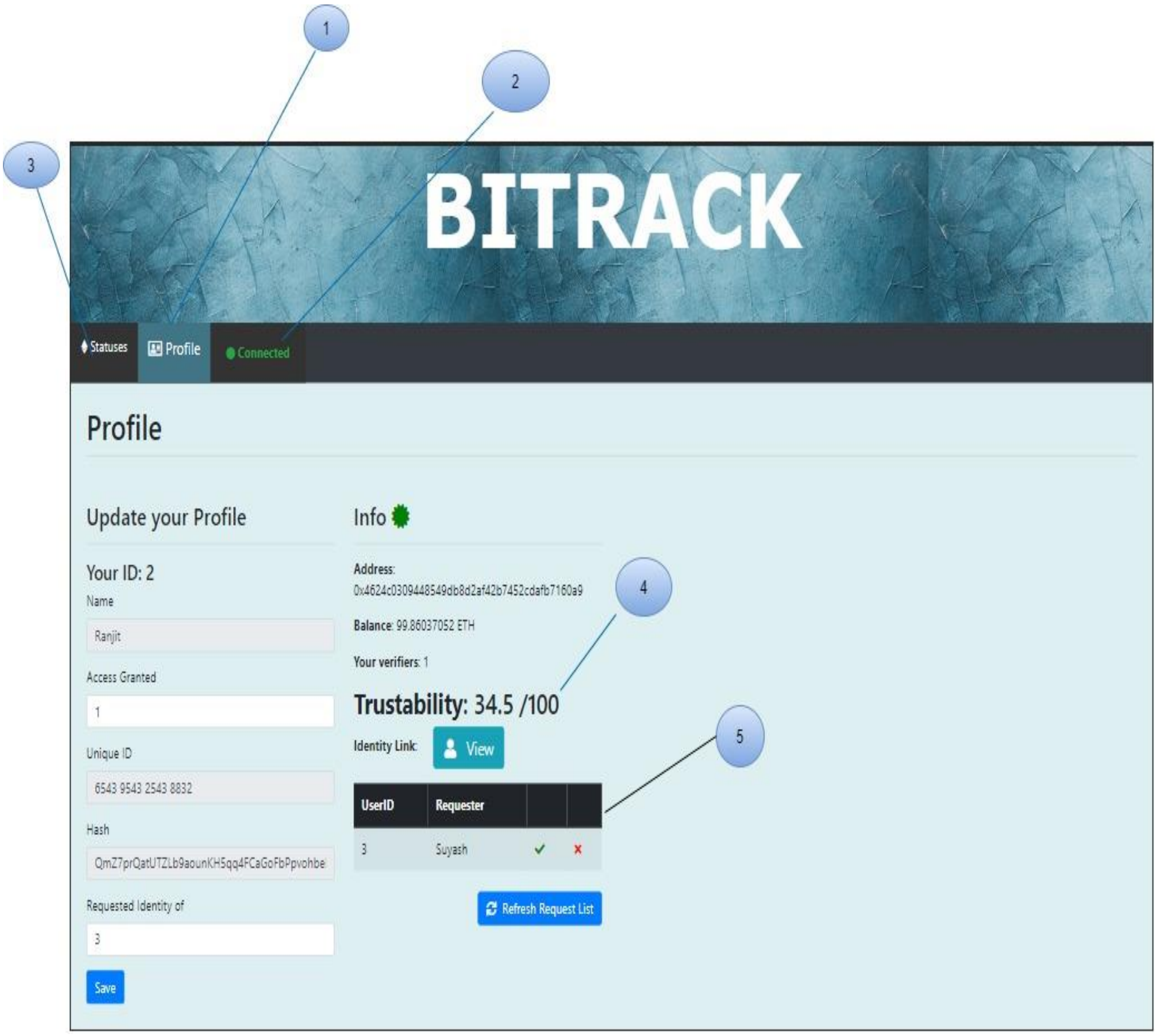
identity$_c$ - - > identity$_i$ (Address$_i$, ID$_i$, account$_i$) …….

(- - > - scaling trust)

Apart from that, a contract owner initially would have a higher voting power.

4.2.1. System design

This section depicts the proposed identity and access management system's implementation-based architecture. It is made up of a client application (App), a blockchain network (which necessitates an Ethereum account), the Interplanetary file system (IPFS) and an application server (DApp-server), By authenticating transactions on the blockchain network, the system authenticates and authorizes users. This client app is responsible to communicate with the Dapp server. It interacts with the blockchain network by authenticating users using credentials saved in the database to the account address corresponding to the Ethereum account. The Appserver then executes the desired operation, such as accessing or changing data in the database, after the user has been successfully authenticated.



Profile

Update your Profile

Your ID: 2

Name

Ranjit

Access Granted

1

Unique ID

6543 9543 2543 8832

Hash

QmZ7prQatUTZLb9aounKH5qq4FCaGoFbPpvohbe

Requested Identity of

3

Save

Info

Address: 0x4624c0309448549db8d2af42b7452cdafb7160a9

Balance: 99.86037052 ETH

Your verifiers: 1

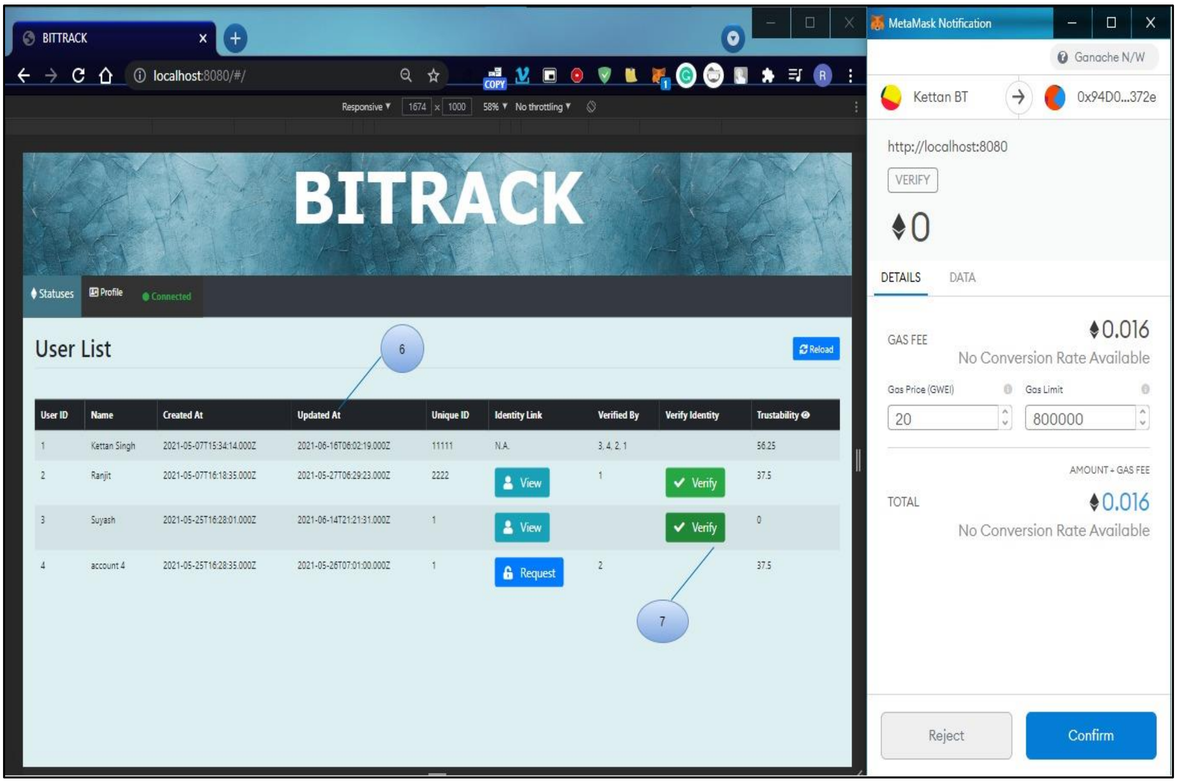
Trustability: 34.5 /100

Identity Link: View

UserID	Requester		
3	Suyash	✓	✗

Refresh Request List

Fig. 8. Depicts Implementation based trust score model 1) Profile page: After registering, the user is directed to the profile page, where he or she can change the credentials. 2) A web3 instance is found locally. 3) Statues Page: On this page, every user can see who else is a part of the blockchain. 4) The cumulative trust score for the present user is determined. 5) User Request list.



User List

User ID	Name	Created At	Updated At	Unique ID	Identity Link	Verified By	Verify Identity	Trustability
1	Kettan Singh	2021-05-07T15:34:14.000Z	2021-06-16T06:02:19.000Z	11111	N.A.	3, 4, 2, 1		56.25
2	Ranjit	2021-05-07T16:18:35.000Z	2021-05-27T06:29:23.000Z	2222	View	1	✓ Verify	37.5
3	Suyash	2021-05-25T16:28:01.000Z	2021-06-14T21:21:31.000Z	1	View		✓ Verify	0
4	account 4	2021-05-25T16:28:35.000Z	2021-05-26T07:01:00.000Z	1	Request	2		37.5

GAS FEE 0.016
No Conversion Rate Available
Gas Price (GWEI): 20
Gas Limit: 800000
AMOUNT + GAS FEE
TOTAL 0.016
No Conversion Rate Available

Fig. 9. Representation of Blockchain state. 6) Real time identity modification update. 7) Verify another user.

4.3. *Scaling trust*

As long as the trust built in a system does not collapse or dwindle, the system works on par with the expectations. The same problem embroils the world of identity management system in Blockchain, wherein new Nodes are added to the user discretion, hence creating a new node identity$_z$ ϵ I(Identity) for every new node added into the system. Hence, the system craves for a parameter that scales accordingly which is a trust model that is introduced in the upcoming section. In this section, therefore, we look at the framework of maintaining and scaling trust throughout the system and even reinforcing it further whenever a new node is added into the chain.

1. Request Access: whenever a user requests an identity the unique ID of the user is added to the list of requests pending for that particular user.
   user.requests = _newRequests

2. Grant/Revoke Access: The user can view the people who has requested his identity. The information is fetched from his account details which is linked to his unique ID which can be fetched from his wallet address. From this list of requester's user m ay grant or revoke access. If he revokes it the unique Id of requesting user is deleted from the list of requesters and if he grants it the unique ID is first deleted, then it's added to another list which contains the unique ID of the user to whom access has been granted

To grant
user.requests = user.requests - RequesterId
user.accessors = user.accessors +
user.accessors

To revoke

user.requests = user.requests – RequesterId

4.4. ***Calculation of trust***

4.4.1. Calculating T_{VX}

T_V represents the trust from one verification node to an identity node.

The X represents the user and hence T_{VX} is the power of trust given to a verification node by a user X.

- Each verification node created by the same user X has equal power of trust which depends on trust factor of the user X.

We introduce a discount factor *d*, which stimulates the decline of trust as we move away from the source.

Where *d* = discount factor

We also introduce a threshold trust value, below which $T_{VX} \rightarrow 0$,

i.e.,

for $t_X < t_t, \quad T_{VX} = 0$

Hence, we define T_{VX} as:

$$T_{vx} \begin{cases} =(d).t_X \;, & t_X \underline{>} t_t \;\; \Lambda \;\; (I, u) \\ 0 \;, & t_i < t_t \;\; V \; (I, u) \end{cases} \qquad (4)$$

4.4.2. Calculating discount factor

Our system incorporates a novel concept of a discount factor, which factors in the attester's trust score when verifying the identity node. The discount factor follows an exponential decay model based on the attester's trust score and the distance of the node being verified from the source node. As the distance between the identity node and the attester increases, the discount factor increases, causing a decline in the trust power. This approach ensures that a node that is farther away from the attester undergoes a lesser change in trust value (Δt) than a node that is closer, enabling the system to maintain, distribute, and regulate the trust score effectively. Moreover, our decentralized system is secure from 51% attacks, as no single user can verify or downvote an identity, thanks to the trust distribution mechanism using the discount factor. This makes the system more robust, scalable, and autonomous, providing a secure environment for identity verification.

The discount factor d_i of user node (U_i) in the web of trust model is defined as:

$$d_i = \left\{ \frac{1}{2^{n_k}} \right\} \qquad (5)$$

Where n_k is the distance of the user node(i) from the attester node k. Hence, change in trust experienced by i^{th} node from a source attester node k:

$$\Delta t = (d_{ik}) * T_f(k) \qquad (6)$$

Therefore, the cumulative change in trust score(Δt_i) for U_i would be defined as:

Cumulative trust score of identity node *i*,

$$\lim_{\Delta ti \to 0^-} = \Sigma_{k=0}^{\infty} (d_{ik}) T_f(k) \qquad (7)$$

Here, $T_f(k)$ is the cumulative trust score of the attester nodes k which verified the identity node *i* and d_{ik} is the discount factor received from k^{th} attester node.

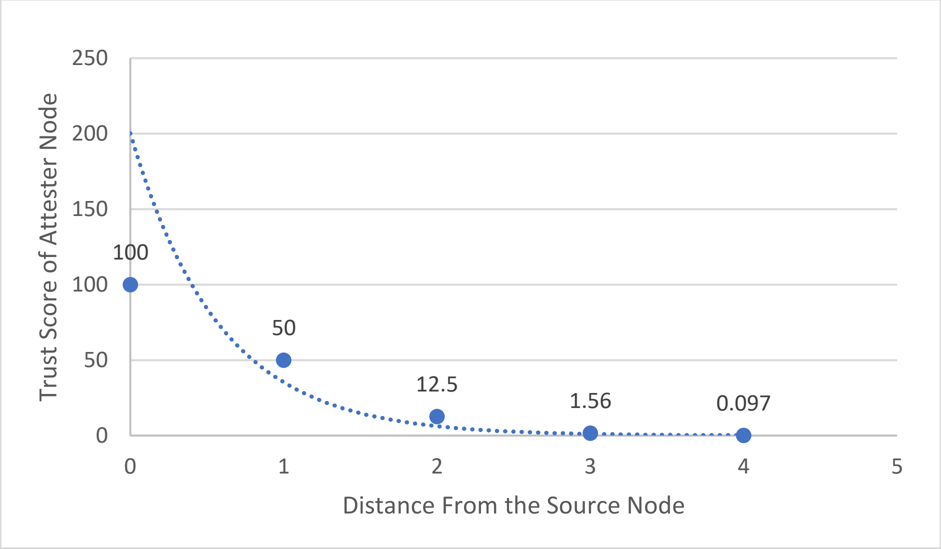


Fig. 10. Depicts a graph where average change in trust declines exponentially as the distance from the source node is increased.

Therefore, the trust equation forms an integral part of the backend subsystem. The trust distribution, which would play a vital role in assisting the system to be self-sufficient, would be in accordance with a trust equation devised, which would ensure that the power to verify other users in the system is traded off in an exact and transparent manner.

5. Experiment

For the initialization of the system there would be a admin that would be the identity issuer that is the government. The admin initially would have the trust score of 100. Now, let us say three user nodes are enrolled into the system. These users are verified from the admin and every time the admin verifies an identity then the change in trust would be a PGP (pretty good privacy) factor described in [15] that is:

$$s = \frac{\ln 0.5}{50}$$

= -0.01386294361119890618834464242916

Therefore, the five users would have a trust score of,

$T_c = T_f(Admin) \div modulus(s).$

(8)

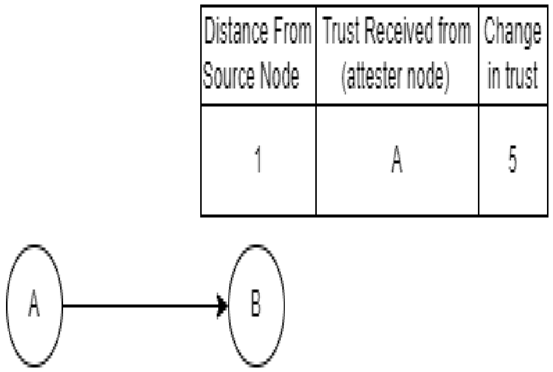
This leads to the next stage of the system where the user verifies each other's identity. Let us say the trust score of three user nodes (*A, B, C*) in the system is 10(T_f) and the identity node *a* verifies another identity *b*, then the trust score would be incremented by the change described in the above "Eq. (6)":

$d_b = \frac{1}{2^n} = \frac{1}{2^1}$ {*n* is the distance from the attester node A which is 1 as the attester node A directly verifies the identity *b* & d_b is the discount factor for node *b*}

Change in trust for Node *B*, $\Delta t_b = d_b * T_f(A) = 0.5 * 10 = 5$

Hence, the final trust score T_f(Initial Trust score of *B*) would be incremented by Δt,

$T_f(b) = T_f + \Delta t_b = 10 + 5 = 15$



Distance From Source Node	Trust Received from (attester node)	Change in trust
1	A	5

Fig. 11. A directed graph in the web of trust model when identity node A verifies node B

Now let us suppose that a node *C* approves/verifies the identity node *A*. Therefore, the change in trust(Δt) would be experienced by identity node *A & B*. Node *B* would also experience this change because node A has verified node B in the last step. Saying this, all the nodes which are verified by node A would experience a change in trust. This trust will be distributed will be distributed until limit Δt tends to 0.Therefore, the new trust scores of Node A & B would be defined as:

Discount factor for identity Node *a* from "Eq. (5)" , $d_a \;=\; \frac{1}{2^n} = \frac{1}{2^1} \quad \{n = 1\}$

Change in trust for Identity Node *a* "Eq. (6)": , $\Delta t_a = \; d_a \;* T_f(C) = \; 0.5 * 10 = 5$

Cumulative trust score for Identity Node *a* "Eq. (7)", $T_f(A) = T_f + \Delta t = 10 + 5 = 15$

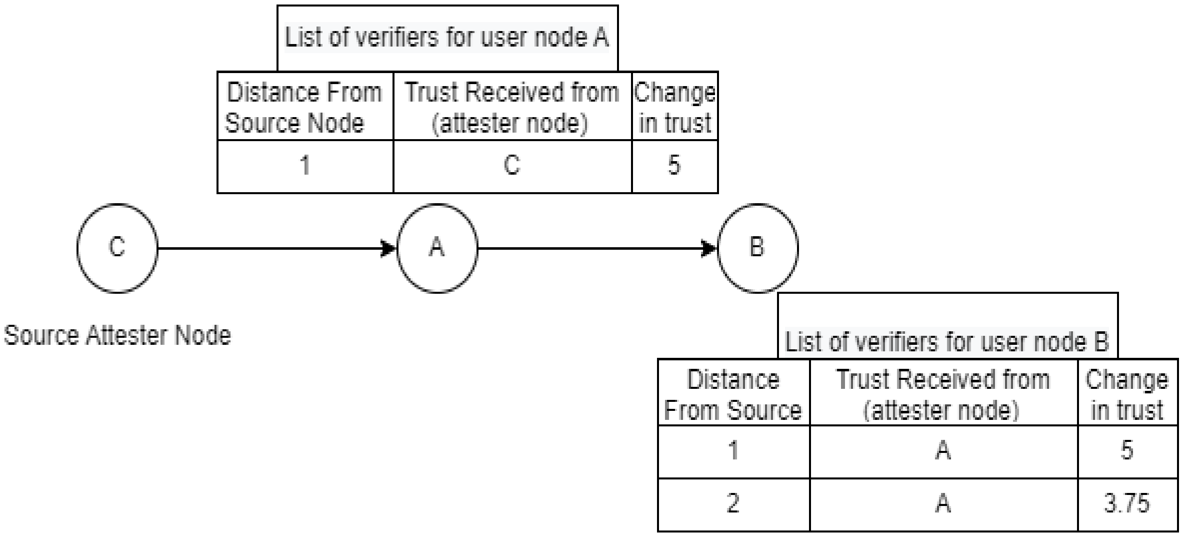
Similarly for user node *b,*

$d_b = \frac{1}{2^n} = \frac{1}{2^2}$ {n being 2 as the distance of node *b* from source attester node *c* is 2}

$\Delta t_b \;=\; d_b \;* T_f(A) = \;\; * 15 = 3.75$

$T_f(B) = T_f + \Delta t_b = 15 + 3.75 = 18.75,$ where T_f was the initial trust score

Since, the trustworthiness of the identity is directly proportional to the trust score hence, Identity node *b* is more trustworthy than identity node *a*.



List of verifiers for user node A

Distance From Source Node	Trust Received from (attester node)	Change in trust
1	C	5

List of verifiers for user node B

Distance From Source	Trust Received from (attester node)	Change in trust
1	A	5
2	A	3.75

Fig. 12. depicts an addition of Node C in the directed graph of web of trust when it verifies node A.

6. Comparative Analysis

The complexity of today's identity and access management challenges, and the need for secure access in today's digital ecosystem, pose significant challenges for information technology service personnel who must keep data safe while meeting users' access requirements. A reputation-based model [16] is introduced which is associated with the integrity of identity and is operated on crowdsourcing tasks with binary choices where a user in the system can approve or disapprove of the claims made by an identity. It involves two stages namely, the publishing stage and the consensus stage. In the publishing stage, a user in the system published a task to prove their reputation and publishes this task to the blockchain network. The second stage, the consensus stage is when the published task receives votes where the votes are between “Agree” and “Not Agree”. When a worker votes for “Agree” then the worker is in agreement with the claims made by the task. Similarly, when the worker votes “Not Agree”, indicates that the worker is debunking the claims made by the published task. When this predetermined voting time is over, the final result of this task is published which would be “Approved” or “Rejected”. For the publisher, if the result is approved, the publisher would experience an increase in number of *RpCoin* which is also equated to the reputation of the user. Similarly, if the result is rejected then the RpCoins would be deducted from the publisher. For the worker, if their votes are consistent with the result of the task, they would also receive *RpCoins*.

The above-described reputation model suffers from various IAM and scalability challenges. In this model, the users have to create a task for the verification of identity and must wait for a predetermined voting time before marking the identity as approved. This would bring in scalability issues as identities would have to wait for verification for a certain time despite being trustworthy. Also, as the number of users in the model increase, there would be a large number of factors that should govern this time factor which would not be feasible in the decentralized world because if this time factor is variable based on the number of users, it would require to query the blockchain state exponentially. Furthermore, a task without votes after this predetermined voting time is abandoned. This could result in a trustworthy identity remaining unverified in the system due to a lack of votes. Whereas in our proposed model, the contract owner who could be the identity issuer (for example government) would determine the Trust score threshold (T_f) and if the trust score of the identity is greater than the trust threshold, the identity would be marked as verified. Also, here since a user with a higher trust score would have more voting power hence, we don’t need to determine a predetermined voting time for the users to vote. This is because, if the claims of the identity have been verified by other user’s nodes who have a higher trust score, then the identity would be marked as verified. This also fastens the process of verification of identities rather than in the former model where an identity node had to wait for a certain time for other identity nodes with equal voting power to come in and upvote i.e., verify the identity. Hence, an identity whose claims are true is not bottlenecked in the system.

In the reputation model stated, as identity claims which are correct could be bottlenecked in the system, it could lead to an increase in the number of dubious identities. Therefore, this makes this model prone to 51% attack where more than 50% of the blockchain network is controlled by dubious identities. Also, there is no factor to differentiate between a trustworthy and dubious identity that is the reputation of the user is not taken into account while performing the task of verification of other identities in the system. In this approach, there is no upper limit on the *RpCoins* which is equated to the reputation of the user and it could increment exponentially if the user has spent a prolonged time in the system being active and this could ultimately result in digressed from decentralization. On the contrary, in our proposed model, since there is a central authority in the initialization of the system which would govern the regulation of the trust score in the beginning, there is a slow transition to a decentralized autonomous identity management system where this trust of the central authority is distributed to identity nodes enrolling in the system such that overall trust score in the system remains in equilibrium that is there is a distribution of the trust. Also, since there is a weight (cumulative trust score T_c) assigned to each identity so, each identity would have voting power based on this weight, and a trustworthy identity when verifying another identity node then, it would lead to a greater increment in the trust value of that identity node thereby the number of trustworthy identities would rapidly increase in the system.

In the reputation model, a user node has to manually publish a task for identity in the system for crowdsourcing and that too for the identities to which it might not be authorized by the owner of that identity i.e., attribute provider. Hence, it falls short of one of the major challenges of IAM which is privacy. In our proposed model, this task is automated, and also another identity node can view and thereby verify another identity only if it is authorized to do so by the attribute provider of that identity. Going one step further, our model provides selective disclosure of the information. But this could lead to a case where a suspicious identity is only giving the power of verification to other dubious identities in the system. This is prevented in the web of trust if an identity in the system is upvoted or approved by too many dubious identities i.e., identities that have a trust score below (trust score threshold) T_f, a push notification would be sent to the central authority which would be the contract owner that is the government agency so as to prevent another dubious user to enter into the web of trust.

In conclusion, while the reputation model has its benefits, it presents several challenges that our proposed model addresses. Our model improves scalability, prevents 51% attacks, ensures decentralization, and resolves privacy challenges, making it a more robust solution for IAM.

7. Results

Our motivation in this study is to develop a framework that allows to maximize transparency as well as control users' personal data. Likewise, a user in our system can track all bits and pieces of his identity and at the same time ensuring a complete integrity of the identities which are assigned a trust score that is equated to how true is the claim made by an identity. Our model follows all the characteristics of self-sovereign identity model (SSI) [15] as follows:

(1) Existence**:** The existence feature of the SSI model requires each entity to have an independent existence. In our model, each identity is bound to a unique public key Ethereum address, ensuring that the same identity cannot have multiple public key addresses, and vice versa. The experimental data in Table 2 shows that our system prevents the creation of multiple users with the same public address, thereby enhancing the security of the system.

Table 1. Users' public Ethereum addresses

User Id	Address
1	0x60A35be415fe2EE4D4b047433D0F7A3DD563D05E
2	0xBC1b961E828F8AA145DDE9084822AB450548AF28
3	0x72AF462Dc93090891C434a41DbF86A9ad0F3c609

Table 2. Show's creation of new identity with same identity information

User ID	Identity	Address	System Output	Block Number
1	Name: Pranav	0x60A35be415fe2EE4D4b047433D0F7A3DD563D05E	0x114FacF171Ee4433Aea8CA7D17a975C518487413	12
2	Name: Raghav	0xBC1b961E828F8AA145DDE9084822AB450548AF28	NULL	13

Conclusion: Based on the experimental data in Table 2, although the identification information was different in the two sets of inputs, new identification information could not be generated because it was not recorded in the system block. This means that the system cannot generate a new ID with the same public key address as an existing user in the database. As a result, our system can prevent users from being created multiple times with the same public address.

(2) Control: Defined as the level of control the user has on his identity which is ensured in our model by actually allowing the user to grant/revoke access to his identity information. Also, our model empowers the user with selective disclosure of identity that is disclose only the information needed by the recipient.

(3) Access: The user should be able to access their identity without any hinderance. They should be aware about all the changes made to the system in which their identity is present. In our model, identities are stored on IPFS, a decentralized file-sharing system that is accessible from anywhere and tamper-

proof. Each user address is associated with an IPFS hash of the identity, ensuring that the identity information is available to the user whenever required.

usersIds[wAddr] = _ipfsHash

(4) Transparency: The systems and algorithms used to manage identities must be transparent in how they operate, manage, and update. So, anyone can see how it works. In our model, we have assigned a trust score to govern the claims made by the identity and the rules for which are coded in the contract published on the public leger and thereby allowing anyone with contract address to view it. Apart from this, the focus here is not on the ownership of the data, but rather on managing the flow of data from the data to the service provider by managing the user's associated consent to each service. In the proposed system, users can grant and revoke access to personal data, as well as control who has access and for what purposes. Furthermore, before marking the identity as verified in the system, there must be consensus within the system of user nodes that the claim made by the identity are true. Lastly, a user node can view any changes made to the other identities node(i_1) which is verified by this user node so as to ensure that the identity claims made by i_1 holds true even after the alteration of identity.

(5) Persistence: The persistence feature of the SSI model requires users' identities to be long-lived, and our model fulfils this requirement using blockchain protocols. All aspects of identity are stored on the blockchain, and real-time details are fetched from the Dapp server and displayed on the client Dapp through blockchain transactions.

(6) Interoperability: Interoperability is another critical aspect of identity management, and our proposed model ensures that identity information is widely usable while remaining under the user's control. Requesters of an identity, such as third-party services, can only view the user's identity if they are authorized to do so, with authorization lying in the user's hands.

(7) Consent & Minimization: In our model, the user can view and track the details of the requester of identity. Also, he has the power for a selective disclosure of identity that is on 'Need to know' basis. Once, the user grants permission to the requester then when the requester actually views the identity is also recorded and a push notification is sent to the authorizer. Also, before marking the identity as verified in the system, there must be a consensus within the system of user nodes that the claim made by the identity is true. Table 3 shows the result of this experiment with a sample request format made to a user that is the attribute provider by the requester of identity:

Table 3. Show's a sample request format made to user while requesting their information

User ID of Requester	Requester Name	IsGranted	Requested At	Viewed At	Attribute Requested

101	Garv	True	DD/MM/YY	DD/MM/YY	Passport
102	Raghav	False	DD/MM/YY	NULL	DL

Conclusion: Based on the above experiment, since the user with ID 102 access is pending with the authenticator of the identity, he cannot view the identity and thereby the system output is NULL. The experimental data presented in Table 3 shows that the proposed model successfully ensures user consent and minimization. Users can view requests made to access their identity, and the authorization lies with them. Additionally, the system records when the requester views the identity and sends a notification to the authorizer. The consensus mechanism within the system ensures that only verified identities are accepted, further enhancing the security of the system.

(8) Protection: The user rights must be protected. In our model, each identity has a trust score and based on which it has the voting power to verify an identity. Hence a user making true claims of their identity would be upvoted in the system thereby protecting the system and its stored identities from fallacious information. Here the only way to change identity information associated with a particular user is by the user himself through his public key address (PKI) which is unique to every user on the blockchain. So, only if the public address and the identification information are accurate then can the identity address be updated.

(9) Table 4. Show’s identity modification with true and false ID

User Id	Address	ID accuracy	System Output
1	0x60A35be415fe2EE4D4b047433D0F7A3DD563D05E	Ture	Success
2	0xBC1b961E828F8AA145DDE9084822AB450548AF28	False	Failure

In conclusion, the proposed model effectively ensures transparency and user control over their identity information, making it a reliable and secure system for identity management. The various features of the SSI model are incorporated in our model, providing users with a robust identity management system.

7.1. *Examination and validation of results*

A variety of studies have been carried out to evaluate the functionality and analyze the performance of use case of blockchain in identity management systems. These trials also contribute to a better understanding of the blockchain's usefulness in real-world applications. Experiments are carried out on blockchains of N, 1, 20, 40, 80, and 100 entries, with 50 measurements collected for each test. Using the Chrome web browser, a total of ten simultaneous queries were made. The average query time for a single record in the blockchain is 54 milliseconds, while the average invoke time is 2196 milliseconds. The tests' scale was limited by the memory resources available on the local computer. While the initial blockchain size is minimal, the amount of RAM required to run a local instance of blockchain and the front-end server at the same time was the limiting factor.

According to Fig. 13, the blockchain query time is comparable to the system's login and update data operations' response times.

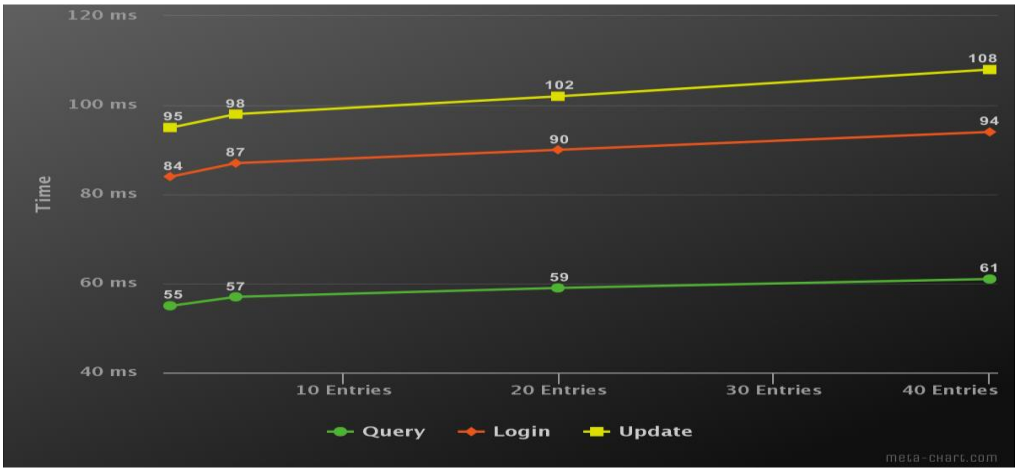


Fig. 13. Performance measurement of blockchain

The invocation time is strongly related to the response time of the registration, grant permission, and revoke authorization functions as shown in Fig. 14. The reason for this is that the blockchain is queried by the login and update data functions. The blockchain is checked when data is updated to ascertain if the actor has authorization to access the object in question. Information is needed in the registration scenario.

Another thing to keep in mind is that system functions like registration, permission-granting, and login times all increase in response time as the number of entries in the blockchain increases.

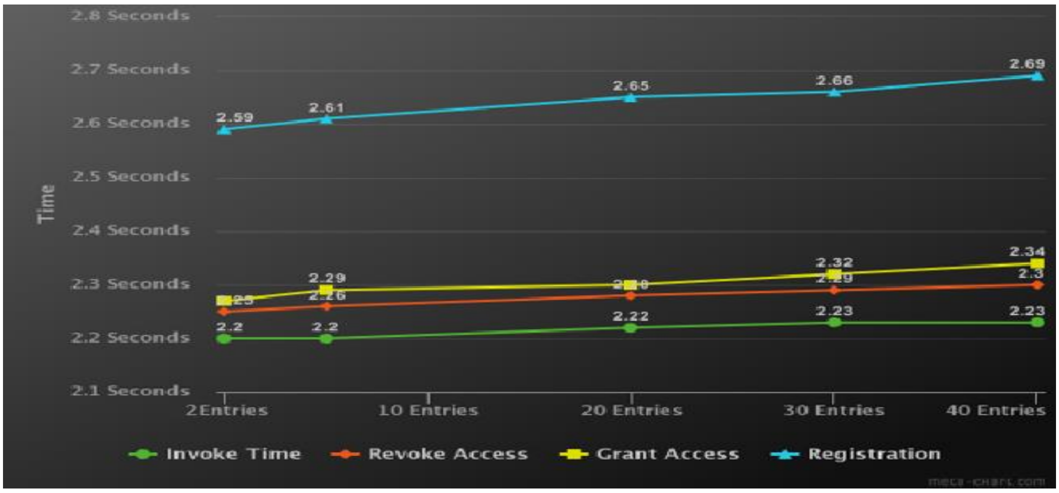


Fig. 14. Performance measurement on basis of blockchain entries

8. Conclusion

In conclusion, the BitTrack system is proposed as a solution to the loopholes present in the current Identity Access system. By functioning as a decentralized application with a consensus mechanism, BitTrack allows for a trust-based verification process that enables users to increment their trust scores and verify other users on the platform. The inclusion of a central authority in the system prevents it from being compromised by the 51% problem. Additionally, the trust distribution in the system is established according to a trust equation that guarantees transparency and accuracy in the power to verify other users.

BitTrack provides a unique closed-chain system that allows users to maintain control over their identities and grant or revoke access to their digital footprints. The system's real-time transaction details from the blockchain make it scalable and improve user monitoring of their identities. Overall, the proposed BitTrack system not only surpasses the current centralized system but also eliminates its loopholes, thereby offering a reliable and secure solution to Identity Access systems. With utmost care and sincerity towards the technology used, the development of the BitTrack system has been completed, providing a viable alternative for identity verification in the digital world.

8.1 Future scope

For future work, we can modify the trust equation to include other parameters which would lead to better distribution of trust score among the users. Also, the current system does not adhere to incentive model which would encourage the users on the system to play by the rules. This model would enable to increment trust score of users who verified other users on the platform once they are categorized as verified identities.

Apart from this, we can also give the user the ability of single sign-on where the user can use Bittrack to authenticate himself on any third-party application preventing the sharing of his identity with third party apps and using a decentralized system to store his identity. Going one step further, the user could also monetize his/her data which is required for analysis for third party vendors. . Hence, this would make our model a one stop solution for all identity management related issues and at the same time ensuring a complete trust-based model to track identities of the user with finest of details.

References

[1] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2019.

[2] J. Lee and S. Member, “BIDaaS : Blockchain Based ID As a Service,” *IEEE Access*, vol. 6, pp. 2274–2278, 2018.

[3] B. Leiding and A. Norta, “Mapping requirements specifications into a formalized blockchain-enabled authentication protocol for secured personal identity assurance,” in *International Conference on Future Data and Security Engineering*, 2017, pp. 181–196.

[4] V. L. Lemieux, “Trusting records: is Blockchain technology the answer?,” *Records Management Journal*, vol. 26, no. 2, 2016.

[5] A. Josang, M. AlZomai, and S. Suriadi, “Usability and privacy in identity management architectures,” in *ACSW Frontiers 2007: Proceedings of 5th Australasian Symposium on Grid Computing and e-Research, 5th Australasian Information Security Workshop (Privacy Enhancing Technologies), and Australasian Workshop on Health Knowledge Management and Discovery*, 2007, pp. 143–152.

[6] B. Faber, G. Michelet, N. Weidmann, R. R. Mukkamala, and R. Vatrapu, “BPDIMS:A blockchain-based personal data and identity management system,” *Proceedings of the Annual Hawaii International Conference on System Sciences*, vol. 2019-Janua, pp. 6855–6864, 2019.

[7] U. La, S. Bcn, D. A. Filvà, F. J. García, and M. A. Forment, “Privacy and identity management in Learning Analytics processes with Blockchain,” 2018.

[8] R. Heck, J. Torstensson, Z. Mitton, and M. Sena, “UPORT : A PLATFORM FOR SELF-SOVEREIGN IDENTITY,” 2017.

[9] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, “Medrec: Using blockchain for medical data access and permission management,” in *2016 2nd International Conference on Open and Big Data (OBD)*, 2016, pp. 25–30.

[10] T. Rathee and P. Singh, “A systematic literature mapping on secure identity management using blockchain technology,” *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 8, pp. 5782–5796, 2022.

[11] S. Khanum and K. Mustafa, “A systematic literature review on sensitive data protection in blockchain applications,” *Concurrency and Computation: Practice and Experience*, vol. 35, no. 1, p. e7422, 2023.

[12] K. Cameron, “The laws of identity,” *Microsoft Corp*, vol. 12, pp. 8–11, 2005.

[13] P. Dunphy and F. A. P. Petitcolas, “A first look at identity management schemes on the blockchain,” *IEEE Security & Privacy*, vol. 16, no. 4, pp. 20–29, 2018.

[14] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun, “A review on consensus algorithm of blockchain,” in *2017 IEEE international conference on systems, man, and cybernetics (SMC)*, 2017, pp. 2567–2572.

[15] Q. Stokkink, G. Ishmaev, D. Epema, and J. Pouwelse, “A truly self-sovereign identity system,” in *2021 IEEE 46th Conference on Local Computer Networks (LCN)*, 2021, pp. 1–8.

[16] X. Wang, Z. Tan, and S. Wang, “An Identity Management System Based on Blockchain,” *2017 15th Annual Conference on Privacy, Security and Trust (PST)*, pp. 44–4409, 2017.