Implementing Decentralized Digital Identity using Blockchain

Akshay Bakre, Nikita Patil, Sakshum Gupta
Symbiosis International University
Symbiosis Institute Of Telecom Management, Pune, India

Abstract
The internet does not have an embedded identity layer. Each application manages its own digital identity, as per its specific requirements. This decade has seen an explosion in digital identities across applications. Every user has multiple identities that he creates and maintains, to use on different applications, as per the set rules of that application. This has generated in huge volumes of user data with the service providers, resulting in two problems; user owned private data is saved and left to the discretion of such 3rd party applications and the ownership of the user’s data is no longer his. Maintenance of such multiple identities will only become more cumbersome as the digital era unfolds. We attempt to solve this problem by creating a blockchain for such digital identities. The paper explores how such a blockchain could be implemented and its functionality.

Keywords: Digital Identity, Blockchain, data privacy

The concept of identity

The 4 Phases of Identity:

There have been 4 phases of evolution for the online Identity management systems since the inception of the concept.
1. Centralized Identity
In the initial days, the role of issuing and authenticating the digital identity was kept only with the centralized authorities. Some organizations took one step ahead and created hierarchy which was beyond the centralization. A root controller could appoint other organizations to each oversee their own hierarchy. The major point of concern with this root controller was a new system was designed where the users were locked into a single authority who has the authority to deny their identity or even confirm a false identity. Another point of concern was the balkanization of the identities as the number of websites was increasing at a faster pace.

2. Federated Identity
The next major advancement for digital identity took place when a handful of organizations shifted their operations from hierarchy to de-balkanize online identity. It came with the advent of federated identity, which allowed users to use the same single identity on different sites. As this innovation was brought forward by Microsoft, it almost became the traditional centralized authority, which they were not in the favor of. They resisted this idea by creating a federation which then divided the power of centralized authority among several powerful entities. Federation worked on the problem of balkanization too. (users could wander from site to site under the system. However, each individual site remained an authority.)

3. User-Centric Identity
The Augmented Social Network in 2000 started the study on a new kind of digital identity. This was done keeping in mind the scope of a next-generation Internet. They stressed on building “persistent online identity”, which was meant to give an individual or administrative control across multiple authorities while no federation is required.

4. Self-Sovereign Identity
The centralized identities were turned into interoperable federated identities having centralized control, with the use of user-centric designs. Users were taken consent of as how to share an identity and with whom to be shared. This was just another step in the field of user control of identity, the next required user autonomy. The concept of “personal cloud” also came into lime light in the same time, which again proliferated the ideation of self-sovereign identity.

Self-Sovereign Identity
10 Principles of Self-Sovereign Identity
The conceptual model of digital identity explained in this paper draws upon the following concepts:

1. Blockchain
2. SSL (Secure Sockets Layer)
3. SSO (Single Sign-on)
4. Kerberos

In the following paragraphs, we attempt to explain these concepts in brief:

1. **Blockchain**

A Blockchain is essentially a database that runs a software that validates and shares new entries with all the participants. It is a distributed ledger of records, across nodes. These records are maintained in blocks. Each new block has to be mined by miners by solving a complex problem. Every block points to its preceding block. This blockchain is replicated across all its nodes. This makes it immutable. It would take enormous computing power to hack a blockchain, distributed across multiple nodes. Hence, records on a blockchain cannot be changed. The can be cancelled by a following reverse transaction.

Blockchain is the underlying principal of Bitcoin transactions. Thus, blockchain provides the following functionalities:

1. Decentralized distribution
2. Immutable
3. Security
4. Transparency
5. Authenticity

The following parameters determine the type of the blockchain:

1. Who is allowed to participate in the network?
2. Who is allowed to execute the proof of work consensus protocol?
3. Who maintains the shared ledger?

Following the above principles, a blockchain can be either of the following:

a. Permission less or public blockchain: Completely open, hence anyone can join and participate in the blockchain. A great amount of compute power is required to solve the proof of work problem to confirm consensus.

b. Permissioned or private blockchain: A private blockchain requires an invitation to join. This must be validated by the starter of the network. The access control can have the following alternatives:

1. Existing users will decide the future participants
2. A regulatory body will decide the future participants
3. A consortium will decide the future participants

A private blockchain will have a middle man in the form of participants who run the nodes or technology vendors.

2. **SSL (Secure Sockets Layer)**

SSL uses the asymmetric key algorithm to create a secure connection. A public key is used to secure data. This can only be unlocked with the owner’s private key. A private key is only in the custody of a single entity to whom it is issued. It is not shared. A public key is open to the public and generally used to lock data.

SSL is used to create an encrypted, secure connection over browsers. This is usually depicted visually using the lock icon.

Many websites require an user to enter secure information. If the website server is not secure, this data is vulnerable to theft. Hence, an SSL certificate is installed on the web server. This helps to create a secure connection between the user’s browser and the website server. This ensures data security.
3. SSO (Single Sign-on)
SSO is a authentication service that authenticates an user for multiple applications, with a single logon. Once a user signs on for the first time he is not required to sign on again, even if he switches to another application. The applications need to be given such rights in the backend. Action of signing out from a single application terminates the access to the multiple applications that were logged in earlier. As different authentication mechanisms are supported by different resources and applications, SSO internally need to store the credentials put for initial authentication and then translate them to the required mechanisms. SSO mitigate the risk for access to the 3rd party sites as the user passwords are not stored or shared externally.

4. Kerberos Protocol
Kerberos is an authentication protocol for authenticating trusted hosts across untrusted networks such as the internet. It consists of a user, a service that needs to be accessed and a Key Distribution Centre. The KDC is responsible for authentication and for granting session tickets. These tickets enable a user to logon to a service for the duration of the session. This helps to avoid the transaction of passwords over unsecure networks. As the database of passwords (encryption keys) for all the users on the network is maintained by the authentication server, it becomes extremely important to physically protect and secure the hardware of the server.

Implementation of Digital Identity using Blockchain

Participants:
Admin
The admin of the blockchain would be a consortium of all the service providers registered with the blockchain. Consider these service providers to be social networking platforms such as Gmail and Facebook. These service providers have large amounts of compute powers at their disposal and hence can be responsible for maintaining the ledger and writing the blocks.

Nodes
Nodes are service providers. For example, consider Google and Facebook as two separate nodes. The blockchain is replicated at each node. The nodes have the responsibility of maintaining these blocks and validating new entries.

Users
These are users who access the blockchain

Functions of the blockchain:
1. Creating and maintaining Digital Identities
2. Authenticating logon sessions
3. Recording logon transactions

Entities:
a. Block on Blockchain:
This block is unique to every user. It is identified with a unique number. In this example, we have used 123xxx as an example. This block contains private identification data such as name, address, date of birth, phone number, email, biometric details etc. This block will also contain the user’s public key.
b. Service
This is the service for which user needs to be authenticated using his digital identity stored on the blockchain. We have used Facebook as an example in this explanation.
c. User
This is a unique user who wants to get authenticated for using services provided by different applications online. The user owns private identification data such as name, address, date of birth, phone number, email, biometric details etc. that can be stored on his block on the blockchain. The user knows the unique number of his block on the blockchain.
A user has ownership of a private key corresponding to his identity on the blockchain.

### Detailed Process Flow

<table>
<thead>
<tr>
<th>Sequence of Activities</th>
<th>From</th>
<th>To</th>
<th>Activity</th>
<th>Contents of transaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>User</td>
<td>Service</td>
<td>Request access</td>
<td>Block ID of user</td>
</tr>
<tr>
<td>2</td>
<td>Service</td>
<td></td>
<td>Add service ID suffix</td>
<td>Service ID suffix is added to the request</td>
</tr>
<tr>
<td>3</td>
<td>Service</td>
<td>Blockchain</td>
<td>Request access</td>
<td>Block ID of user + service ID</td>
</tr>
<tr>
<td>4</td>
<td>Blockchain</td>
<td>Encryption</td>
<td></td>
<td>Session ticket for service is encrypted using the user's public key</td>
</tr>
<tr>
<td>5</td>
<td>Blockchain</td>
<td>User</td>
<td>Transmit key</td>
<td>Encrypted session ticket is transmitted to user</td>
</tr>
<tr>
<td>6</td>
<td>User</td>
<td></td>
<td>Decrypt ticket</td>
<td>The session ticket is decrypted by the user using his private key</td>
</tr>
<tr>
<td>7</td>
<td>User</td>
<td>Service</td>
<td>Active session</td>
<td>Authenticated session between user and service for period of ticket</td>
</tr>
</tbody>
</table>

### Use Case:

Consider a user, Robert, who wants to logon to his Facebook account. Robert has his identity stored on the blockchain, against the unique id 56789.

The events will occur in the following order:

1. Robert will visit [www.facebook.com](http://www.facebook.com) from his browser and request access to his account using his own unique block id 12345.
2. Facebook’s server receives this request and adds a unique Facebook id to Robert’s request.
3. Facebook server now transmits this request along with its service suffix to the blockchain.
4. Blockchain verifies Robert’s identity and creates a session ticket for Robert to access Facebook for a specific time period. This session ticket is encrypted using Robert’s public key.
5. The Blockchain transmits this encrypted session ticket to Robert.
6. Robert decrypts the received session ticket using his private key.
7. Facebook recognizes a valid authenticated session and allows access to Robert for the time period of his session ticket.

**Recording Logon sessions**

All logon sessions will be recorded in the blockchain. The following parameters should be recorded:
1. Block id of user
2. Service accessed
3. Time stamp of start and end activity

**The Digital Identity blockchain suggested in this paper will thus have two types of blocks:**
1. Identity Blocks
2. Logon Records blocks

**Conclusion**

As we go through life, we leave behind crumbs of digital data. Essentially, a digital avatar of ourselves. This avatar might even be a better representation of ourselves, as it can record and remember our precise information and doings from the past. This digital identity, can predict from past data, what our actions would be in the future. Blockchain, gives the power to own and control this identity. This identity, in part, as required can be shared for exchange of goods and services over a blockchain. Ownership of identity, also enable us to monetize it. This identity is distributed and recorded over millions of peer-to-peer networks across the globe.

The concept suggested in this paper can be used to create such digital identities. This idea can be expanded for the following as well:
1. Link identity documents to the blockchain (Passport, Driving License etc.)
2. Add services that demand such services as nodes.

However, the implementation process of the above points is beyond the scope of this paper.

**Acknowledgements**

We are deeply grateful to Dr. Pramod Damle for his guidance and expertise during the duration of this project.

**References**


