Decentralizing Aadhaar Authentication using a Blockchain

Mehul Agarwal[§]

October 2018

Abstract. A decentralised blockchain layer added to Authentication of the Aadhaar project in India can add to the security of the process reducing the threat of identity theft to a bare minimum. While onion hashing and better encryption techniques may be developed in the current centralised process of Authentication to help mask an identity, it can still be exploited by internal bad actors. By adding an immutable decentralised layer without disturbing the current architecture, this proposal aims to provide a practical and implementable mechanism to include the benefits of a blockchain to a previously centralised process. It also proposes to construct a private blockchain layer at first, slowly building a completely decentralized (public) one. The incentivization of such a layer would involve economic assets to ensure that any public entity may take part in a Proof of Stake consensus protocol. This opens up opportunities for an implementation of a competent Indian state-backed cryptocurrency with smart contracts leading to better business integrity and minimization of disputes related to fraud and non-payment.

1 Introduction

The Unique Identity project of India, Aadhaar, provides a unique biometric identity to 1.2 billion residents of India. Today, it has effectively replaced traditional methods of identity verification to obtain various services across the country. In doing so, it has vastly reduced the time and effort of these processes. The efficacy and need of this change has been debated, and as with any project of such scale, there are many concerns of privacy, surveillance and the overall security of such a system. This document serves to solve a few of these problems by the introduction of a decentralized Blockchain layer to the previously completely centralized process of Aadhaar Authentication. The Blockchain layer will provide extra layer of security that can increase the integrity of existing Aadhaar system. The paper, in its current scope, suggests an entry log

[§] mail@mehul.al

in a Blockchain ledger for each and every Aadhaar Transaction. This extra layer will maintain logs for each authentication with a unique and immutable authentication hash, hence no authentication would go unaccounted and integrity of every authentication will be ensured.

Any Blockchain data is inherently immutable as a Blockchain works on cryptographic hashes and these hashes takes previous hashes (transaction data + a new and unique key) to generate a new transaction hash. These transaction hashes build a chain of hashes and this chain has the records of all the transactions occurring in that system. The hashes on this chain can't be tampered with due to complex algorithms behind the generation of unique keys. Hence with the implementation of this decentralized layer in addition to the existing Aadhaar architecture, no transaction can go unaccounted for and any user/authority can question any stakeholder on any previous authentication. This mechanism would ensure the integrity of system even against the bad actors that get access maliciously by playing the system. The Blockchain's Distributed Consensus model, wherein multiple independent parties participate in a consensus protocol (for eg. Proof of Work, Proof of Stake), is what differentiates it with centralized systems like the Aadhaar. A Blockchain as an independent and parallel layer to the current Architecture inherently ensures that even someone with access to the central servers of Aadhaar would additionally need control over the consensus (A 51% attack) to introduce an illegitimate authentication. For most practical purposes, this makes sure that even people on the inside (i.e. working at the UIDAI) cannot counterfeit an authentication.

2 Aadhaar

On 28 January 2009, the Government of India came up with the concept of a Unique ID for all the residents of the country. This Unique ID is a 12-digit number who issuance is based upon the demographic and biometric data of Aadhaar holder. Now, it is the world's largest biometric based system with 1,227,404,094 billion Aadhaar holders [1] (as of September 2018). Aadhaar enrolment for an individual is a one-time process with the capture of the unique iris and fingerprint biometrics of the individual. Hence, it can't be issued twice to a single person.

Aadhaar works on the principle of enrol once and authenticate frequently. So, Aadhaar data is taken once and only non-biometric fields can be get updated later. This data can be authenticated any number of times by using Aadhaar API's (designated for the purpose)[2].

India is developing country with a population of 1.3 billion people. Thus, there is a need for unique identities and a digital platform to authenticate it anytime and anywhere. Unlike traditional identity proofs that are originally meant for different purposes (e.g. Passport, Voter ID), the Aadhaar exists as a document solely for identity. Instead of verifying an individual under separate systems, Aadhaar ensures everyone can be verified under a single roof. Apart from verifying identity, Aadhaar can also be considered a proof of residence but not as a proof of citizenship.

The UIDAI Ecosystem[3] essentially is divided into three sections: Enrollment, Updation, and Authentication. The Enrolment (and the Updation) Ecosystem consists of Registrars and Enrolment Agencies. Registrar is an entity authorised or recognized by UIDAI for the purpose of enrolling individuals. Enrolment Agencies are appointed by Registrars and are responsible for collecting demographic and biometric information of individuals during the enrolment process by engaging certified Operators/Supervisors.

In co-ordination with the Registrars, the Enrolment Agencies set up Enrolment Centres, where residents can enrol for Aadhaar. Multiple fingerprint scanners, iris scanners, and cameras used for enrolment are certified by Standardisation Testing and Quality Certification (STQC) Directorate and the UIDAI, and all connect to the UIDAI designed standard Application Programming Interface (API). Appointment of multiple registrars, multiple enrolment agencies, and multiple technology providers has created an environment of healthy competition within.

2.1 Aadhaar Authentication[4]

Several service providers require individuals to submit their identity proofs that serve as an enabler for providing consumer services, subsidies or benefits. While collecting such identity proofs, these service providers face challenges in verifying/validating the correctness of identity information documents or proofs submitted by individuals.

The purpose of Aadhaar Authentication is to provide a digital, online identity platform so that the identity of Aadhaar number holders can be validated instantly anytime, anywhere.

This service from UIDAI can be utilized by the requesting entities (i.e. government / public and private entities/agencies) to authenticate the identity of their customers / employees / other associates before providing them access to their consumer services / subsidies/ benefits / business functions / premises.

2.2 Authentication Architecture[5]

In the Authentication process, the Aadhaar number along with the demographic information or biometric information of a Aadhaar number holder is submitted to the Central Identities Data Repository (CIDR) for its verification and such repository verifies the correctness, or the lack thereof, on the basis of the information available with it. To understand Authentication better, we need to discuss the stakeholders involved in it.

AUA (Authentication User Agency) is the party that initialize an authentication request to get a success/fail token from Aadhaar API's. At the time of Account Opening, Bank is an AUA and bank request for applicant's authentication by asking him to verify with fingerprints. In next step, ASA (another stakeholder in Aadhaar System) accepts fingerprints and other data and give a token (true or false) on the basis of authenticity of data. If data matches with the biometric or OTP (alternate) then token shows success else shows fail.

ASA (Authentication Service Agency) is an organisation providing connectivity using private secure network to UIDAI's data centres for transmitting authentication requests from various AUAs. So, one can't bypass any step and every authentication passes to ASAs from AUAs.

Aadhaar Authentication supports multiple factors to increase the security and strength of authentication. These multiple factors include Demographic Information, Biometric Data, PIN, OTP, Possession of mobile, or combinations. AUAs can add their factors too; this will add extra strength to authentication.

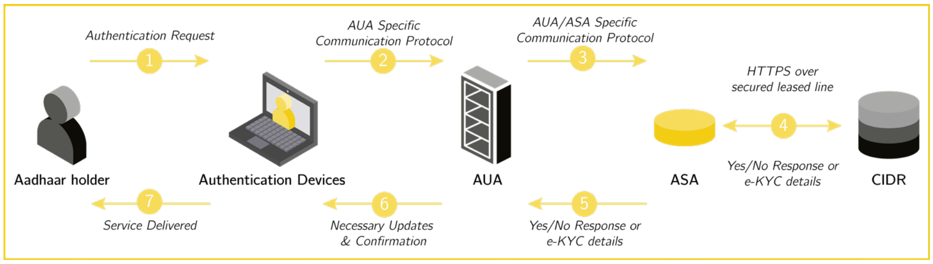

Fig.1: Overview of Aadhaar Authentication

3 Blockchain

3.1 History

The Bitcoin blockchain was introduced by Satoshi Nakamoto (identity unknown) as a way to solve the double spending problem using a peer-to-peer network. Yet, ideas for such a decentralized mechanism was in the works for a long time. Stuart Haber and W. Scott Stornetta are widely considered the first to work on a cryptographically secured chain of blocks in 1991[6]. Their intentions were to implement a system where document timestamps could not be tampered with. In 1992, Bayer, Haber and Stornetta incorporated Merkle trees (The concept of hash trees is named after Ralph Merkle who patented it[7] in 1979) to the design, which improved its efficiency by allowing several document certificates to be collected into one block. Soon after the Nakamoto model, people realized the benefits of the blockchain beyond a currency standpoint.

Blockchain provided the answer to digital trust because it records important information in a public space and doesn't allow anyone to remove it[8]. It's transparent, time-stamped and decentralized. At its core, blockchain is an open, decentralized ledger that records transactions between two parties in a permanent way without needing third-party authentication. Few initial implementations such as the Namecoin used it for a Domain Naming System(DNS). (Provide more egs). the next major step in development of blockchain applications was the introduction of Ethereum by Vitalik Buterin. Ethereum provided a Turing complete language for the blockchain and Smart Contracts, self-executing contracts with the terms of the agreement between users being directly written into lines of code. This allowed development of various applications on top of Ethereum ranging from game-based tokens such as CryptoKitties to Identity systems such as uPort.

3.2 Architecture of Ethereum[9]

Ethereum Accounts: In Ethereum, the state is made up of objects called "accounts", with each account having a 20-byte address and state transitions being direct transfers of value and information between accounts. An Ethereum account contains four fields:

- The **nonce**, a counter used to make sure each transaction can only be processed once
- The account's current **ether balance**
- The account's **contract code**, if present
- The account's **storage** (empty by default)

"Ether" is the main internal crypto-fuel of Ethereum, and is used to pay transaction fees. In general, there are two types of accounts: externally owned accounts, controlled by private keys, and contract accounts, controlled by their contract code. Note that "contracts" in Ethereum are like "autonomous agents" that live inside of the Ethereum execution environment, always executing a specific piece of code when "poked" by a message or transaction, and having direct control over their own ether balance.

Messages and Transactions: The term "transaction" is used in Ethereum to refer to the signed data package that stores a message to be sent from an externally owned account. Transactions contain:

- The **recipient** of the message
- A **signature** identifying the sender
- The **amount of ether** to transfer from the sender to the recipient
- An optional **data** field
- A STARTGAS value, representing the maximum number of computational steps the transaction execution is allowed to take
- A GASPRICE value, representing the fee the sender pays per computational step

The STARTGAS and GASPRICE fields are crucial for Ethereum's anti-denial of service model. In order to prevent accidental or hostile infinite loops or other computational wastage in code, each transaction is required to set a limit to how many computational steps of code execution it can use.

Messages: Contracts have the ability to send "messages" to other contracts. Messages are virtual objects that are never serialized and exist only in the Ethereum execution environment. Essentially, a message is like a transaction, except it is produced by a contract and not an external actor. Like a transaction, a message leads to the recipient account running its code. Thus, contracts can have relationships with other contracts in exactly the same way that external actors can.

Ethereum State Transition Function:

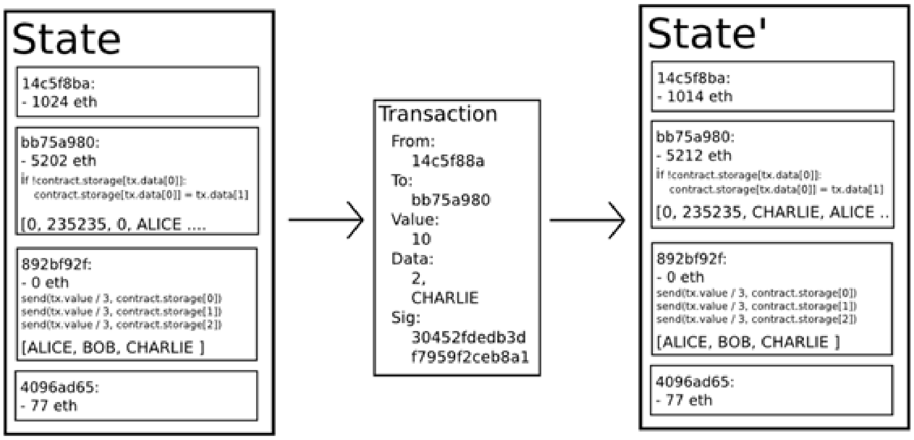
State
14c5f8ba:
- 1024 eth
bb75a980:
- 5202 eth
if !contract.storage[tx.data[0]]:
contract.storage[tx.data[0]] = tx.data[1]
[0, 235235, 0, ALICE ....
892bf92f:
- 0 eth
send(tx.value / 3, contract.storage[0])
send(tx.value / 3, contract.storage[1])
send(tx.value / 3, contract.storage[2])
[ALICE, BOB, CHARLIE]
4096ad65:
- 77 eth
Transaction
From:
14c5f88a
To:
bb75a980
Value:
10
Data:
2,
CHARLIE
Sig:
30452fdedb3d
f7959f2ceb8a1
State'
14c5f8ba:
- 1014 eth
bb75a980:
- 5212 eth
if !contract.storage[tx.data[0]]:
contract.storage[tx.data[0]] = tx.data[1]
[0, 235235, CHARLIE, ALICE ..
892bf92f:
- 0 eth
send(tx.value / 3, contract.storage[0])
send(tx.value / 3, contract.storage[1])
send(tx.value / 3, contract.storage[2])
[ALICE, BOB, CHARLIE]
4096ad65:
- 77 eth

Fig.2: State Transition in Ethereum

The Ethereum state transition function, `APPLY(S,TX) -> S'` can be defined as follows:

1. Check if the transaction is well-formed (ie. has the right number of values), the signature is valid, and the nonce matches the nonce in the sender's account. If not, return an error.
2. Calculate the transaction fee as STARTGAS * GASPRICE, and determine the sending address from the signature. Subtract the fee from the sender's account balance and increment the sender's nonce. If there is not enough balance to spend, return an error.
3. Initialize GAS = STARTGAS, and take off a certain quantity of gas per byte to pay for the bytes in the transaction.

4. Transfer the transaction value from the sender's account to the receiving account. If the receiving account does not yet exist, create it. If the receiving account is a contract, run the contract's code either to completion or until the execution runs out of gas.
5. If the value transfer failed because the sender did not have enough money, or the code execution ran out of gas, revert all state changes except the payment of the fees, and add the fees to the miner's account.
6. Otherwise, refund the fees for all remaining gas to the sender, and send the fees paid for gas consumed to the miner.

Blockchain and Mining:

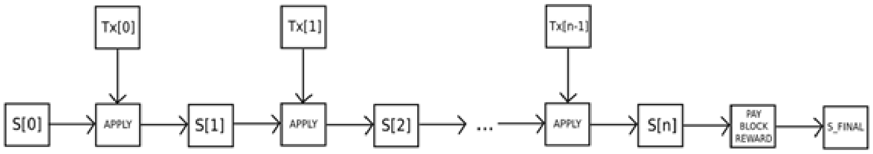

Fig.3: Blockchain Mining

1. The basic block validation algorithm in Ethereum is as follows:
2. Check if the **previous block** referenced exists and is valid.
3. Check that the **timestamp** of the block is greater than that of the referenced previous block and less than 15 minutes into the future
4. Check that the block number, difficulty, transaction root, uncle root and gas limit (various low-level Ethereum-specific concepts) are valid.
5. Check that the **PROOF OF WORK (consensus protocol)** on the block is valid.
6. Let `S[0]` be the state at the end of the previous block.
7. Let `TX` be the block's transaction list, with `n` transactions. For all `i` in **0...n-1**, set `S[i+1] = APPLY(S[i],TX[i])`. If any applications return an error, or if the total gas consumed in the block up until this point exceeds the GASLIMIT, return an error.
8. Let `S_FINAL be S[n]`, but adding the block reward paid to the miner.
9. Check if the Merkle tree root of the state `S_FINAL` is equal to the final state root provided in the block header. If it is, the block is valid; otherwise, it is not valid.

Merkle Trees:

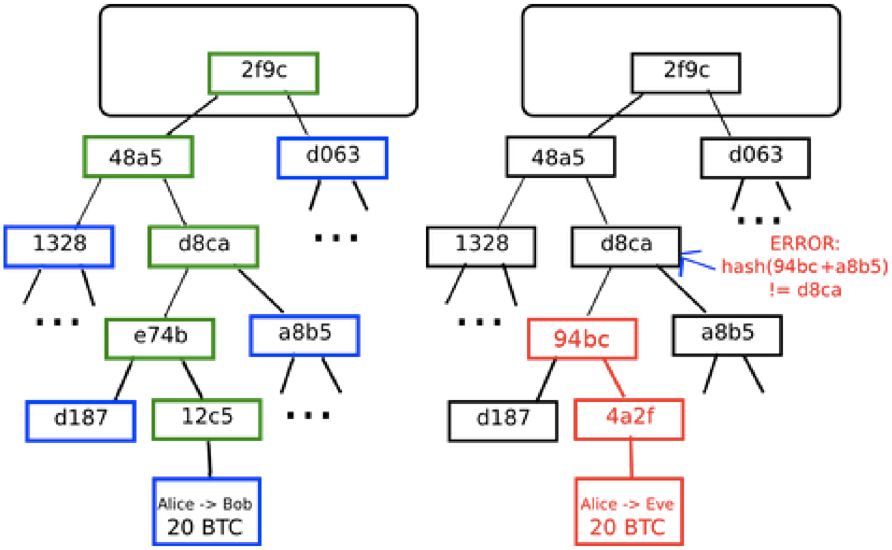

Fig.4: Representation of Merkle Trees

Left: it suffices to present only a small number of nodes in a Merkle tree to give a proof of the validity of a branch.
Right: any attempt to change any part of the Merkle tree will eventually lead to an inconsistency somewhere up the chain.
An important scalability feature of Blockchain is that the block is stored in a multi-level data structure. The "hash" of a block is actually only the hash of the block header, a roughly 200-byte piece of data that contains the timestamp, nonce, previous block hash and the root hash of a data structure called the Merkle tree storing all transactions in the block. A Merkle tree is a type of binary tree, composed of a set of nodes with a large number of leaf nodes at the bottom of the tree containing the underlying data, where each node is the hash of its two children, and finally a single root node, representing the "top" of the tree. The purpose of the Merkle tree is to allow the data in a block to be delivered piecemeal: a node can download only the header of a block from one source, the small part of the tree relevant to them from another source, and still be assured that all of the data is correct. The reason why this works is that hashes propagate upward: if a malicious user attempts to swap in a fake transaction into the bottom of a Merkle tree, this change will cause a change in the node above, and then a change in the node above that, finally changing the

root of the tree and therefore the hash of the block, causing the protocol to register it as a completely different block (almost certainly with an invalid proof of work).

4 Problems in Aadhaar

4.1 Problem Statement

Aadhaar is based on a centralized client server architecture. There exists a lack of trust between Aadhaar Holders and the UIDAI. The AUAs and ASAs remain untrustworthy and multiple instances of Aadhaar "data leaks" at various levels have only made matters worse.

Although data is secured with cryptographic hashes (256 bit – provide reference) but it is still vulnerable to attacks from bad actors inside the system. Such a bad actor can:

I. Edit/Update the Aadhaar data of an Aadhaar holder.
II. Authenticate a service from Aadhaar without the knowledge of Aadhaar holder.

Although the logs of all the authentications of an Aadhaar holder are maintained, it is impossible to trace an Aadhaar authentication and backend triggered transaction in case of some fraud. Blockchain can be a perfect solution in such a scenario.

4.2 Challenges

The following are known challenges of and reservations against Aadhaar.

I. **Integrity:** Integrity is a major concern when Aadhaar holders offers their biometric data for each service, benefit or government subsidy. In a system of this scale, loopholes manifest and bad actors can always take advantage of them. Safeguards against such internal attacks seem few and sparse.

II. **Absolute Centralized power:** The UIDAI, being the central entity of Aadhaar, holds absolute power. On one hand, this power ensures efficient management of services data records at single place. But on other, this same centralized system invites influence from bad actors. It is tough to trace an internal attack back to a bad actor. In a centralized system, integrity is always questionable.

III. **Public Trust:** There is an inherent fear of government surveillance, identity theft and restriction to privacy with Aadhaar in the public mind. The mandatory nature and push by the government to link one's Aadhaar with almost every service certainly doesn't make the situation better.

IV. **Identity Fraud:** To expand on the previous point, there is a legitimate concern of identity theft in the current Aadhaar system. Internal bad actors, silicone finger molds, leaks or improper verification (instead of authentication) all play a part in growing the fear. The UIDAI is working on reducing a few of these threats by introducing Virtual IDs, *Convolutional Neural Networks (CNN) based* liveness detection, but a lot more needs to be done.

4.3 Addressing Challenges with a Decentralised Approach

A decentralized shared ledger like the Blockchain is a solution to most of these challenges of the current Aadhaar Infrastructure. This solution, if adopted in Aadhaar can build integrity and help establish the responsibility of various stakeholders.

One way of doing this is by casting aside the existing central structure in favour of putting all data on Blockchain. Every request for accessing data will make a permanent entry on Blockchain ledger (no matter who is accessing data and for what purpose). The modification of Aadhaar data would only be possible by a two-way (handshake) authentication where user and authority both sign a smart contract.

Doing so would transfer absolute power in users' hand where they are owner of their data and no third party can ever use their data without their permission. Since all the data is stored on Blockchain so any transaction (whether authentication request or update) can't be left unregistered.

However, this implementation is **not practical** as it would require major policy changes and re-shaping the entire centralized architecture. The UIDAI is unlikely to complelety abandon a decade of efforts of creating a centralized repository, forming the right partnerships just to introduce decentralization. Switching would also introduce logical problems such as latency and unforeseen consequences when dealing with a population of over 1.2 billion people.

Instead, it would be better to follow a mixed approach wherein all the data remains on centralized servers but all the logs on immutable Blockchain.

5 Proposed Implementation

5.1 Current Blockchain Based Identification Systems

The concept of a blockchain based identity is not new. Such identity systems form great solutions for providing a decentralized and unique identity to all. Looking at their working can help aid the development of a Blockchain based Aadhaar system. The two most popular Blockchain Identity systems are discussed here.

Civic Protocol: Civic is a Blockchain based secure identity ecosystem. It is called an ecosystem as it provides a couple of solutions for businesses looking at an app-based Blockchain Identity. Users can sign up with their demographic and personal details to generate a unique signature (cryptographic hash- private key) and a unique address key (public address) on the network. Users keep all the data on personal mobile devices only, and no data is shared with any centralized server[10]. Civic also works as a wallet that stores a private key to sign transactions.

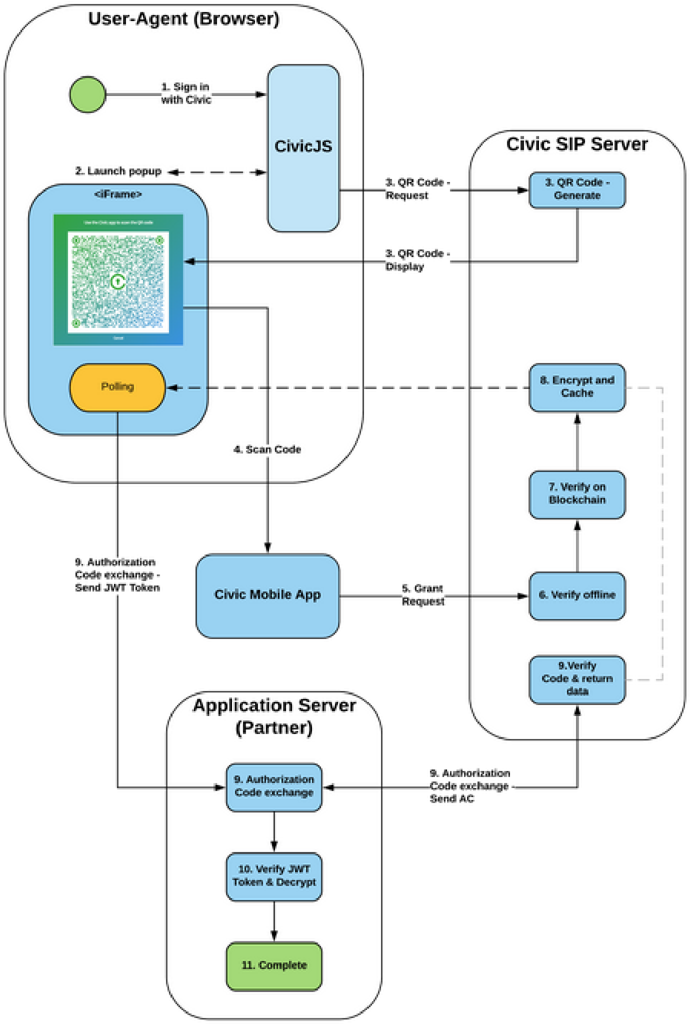

Fig.5: Flow Diagram of Civic

The transaction process is as explained in above diagram. Civic verifies all the transactions with only the local private key and the generated hash is shared at authentication server to test whether the Civic account holder is still authenticated to access the designated service on Blockchain or has been invoked.

uPort: uPort is another Decentralized ID by a company named ConsenSys[11]. uPort is very similar to Civic Id but is a part of ConsenSys ecosystem. uPort can seamlessly integrate with Infura (Blockchain IPFS -InterPlanetary File System Service- by ConsenSys) and Rinkeby (an Ethereum test network). It can also be customized to work as a partially decentralized and partially centralized application. As such, the Aadhaar implementation will be more influenced by uPort than by Civic. Since uPort is an open source project, parts of it may be modified as per our requirements for Aadhaar.

5.2 Private v.s. Public Blockchains

Decentralization forms the very soul of Blockchain technology where all the ledger data is available to all of the nodes and any node can verify block data. The Blockchain in which all the data is accessible to every peer on the chain is called a public Blockchain. In public Blockchain, any node can work as a miner (or validator after submitting stake, if the network is using PoS consensus algorithm). Ethereum, Bitcoin and all other crypto currencies are example of public Blockchain only. A **Public** blockchain has the following benefits:

I. All the data is stored on Blockchain so there is no threat of loss of data and need for the separate storage for the data.

II. All the nodes can participate in mining and it provides absolute power to whole network.

III. It is the most noble way to implement Blockchain, which is truly decentralized and with true integrity.

A **Private** blockchain, or a simple shared ledger, is different in that a blockchain is hosted on private networks and only restricted to desired stakeholders or authorised users. This Blockchain is also immutable. But here, incentivization need not to be that strong and lesser computational power is needed to generate new blocks. Thus, it is more efficient and scalable. However, it puts a lot of trust in the centralized authority. A Private Blockchain can also be hosted in a public-private partnership (PPP) model where public and government can make an agreement to share the resources in a way that optimum utilization of resources can be ensured. The Linux Foundation has developed a special ecosystem to host private Blockchains called the Hyperledger and IBM is already working on this open source technology. Linux Foundation has developed Hyperledger which also can be a viable option for hosting Blockchain.

As the current system of Aadhaar is primarily a centralized dataset, and supports variety of authentication services, a private blockchain is a better option to begin with. Thus, incentivization would not be a major issue to deal with immediately as the UIDAI can make it mandatory for AUAs and ASAs to participate as stakeholders in the consensus protocol. However, it should be noted that the ultimate goal should be complete decentralization of the blockchain layer i.e. building towards a public blockchain. In such a case, incentivization, efficiency and other factors need to be taken in consideration.

5.3 Proof of Work and Proof of Stake

In simple terms, the consensus protocol is a mathematical way to stop fraud over Blockchain. In a blockchain, anyone (any node) can add a new block on Blockchain. Since there is no central authority, there needs to be a way to ensure the authenticity of block (i.e. differentiate between a fraudulent transaction and an authentic one) before adding that to final Blockchain. **Proof of Work (PoW)** and **Proof of Stake (PoS)** are two such protocols. hosting Blockchain.

PoW is the most common protocol in use, currently implemented in Ethereum and Bitcoin and various other cryptocurrencies. In proof of work, a node needs to solve a mathematical puzzle before being eligible to add a new block to the Blockchain. The computational work done to solve that cryptographic puzzle is called Proof of Work. The node called the Miner gets a reward for solving these puzzles or for their Proof of Work. This mathematical puzzle works on a predefined threshold of difficulty and the generated hash works as a parameter to the new block's generation. PoW is a well-tested way to make Blockchain secure but there are serious problems related to power consumption solving these puzzles. This process requires immense amount of energy and computational usage. The puzzles have been designed in a way which makes it hard and taxing on the system[12]. A PoW also puts power into the hands of those with the most computational resources to mine. A common example cited is the Bitcoin chain where 5 mining pools control over 65% of the consensus by hashrate[13] and can “join forces” to launch a 51% attack on the chain, essentially validating any transaction they want, fraud or real.

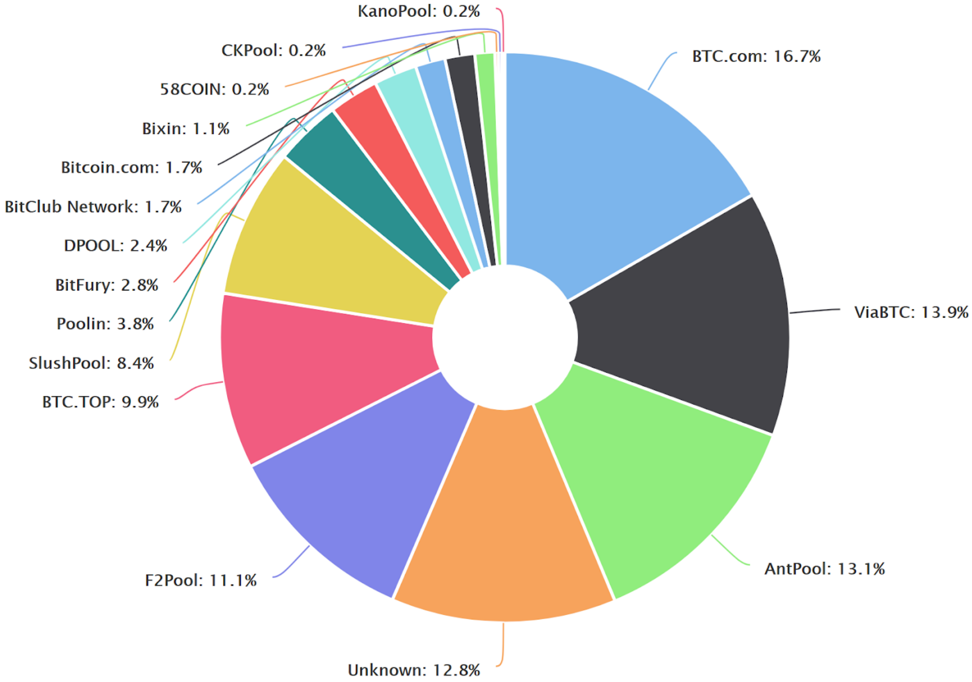

Fig.6: Bitcoin Mining Pools by Hashrate

Proof of Stake is another protocol in which every node (now called validator) needs to put up collateral as stake (instead of solving cryptographic puzzles) to confirm his or her integrity. This adds more decentralization by not restricting people on basis of computational resources. Once the stake is submitted, then the validator can add a new block to the chain. The rewards on the process are dependent on the value of stake and can never surpass the value the stake. So, at any point of time, the validator should have stake in the network more than the actual reward to validate any block. The “one-sentence philosophy” of proof of stake is thus not “security comes from burning energy”, but rather “security comes from putting up economic value-at-loss”[14]. In view of the stake so involved, the validator does not have any incentive to add a fraudulent transaction block to Blockchain. And since the transactions are trackable, if a validator is found committing fraud, then his/her stake and reward can be easily forfeited.

5.4 Blockchain Layer Architecture

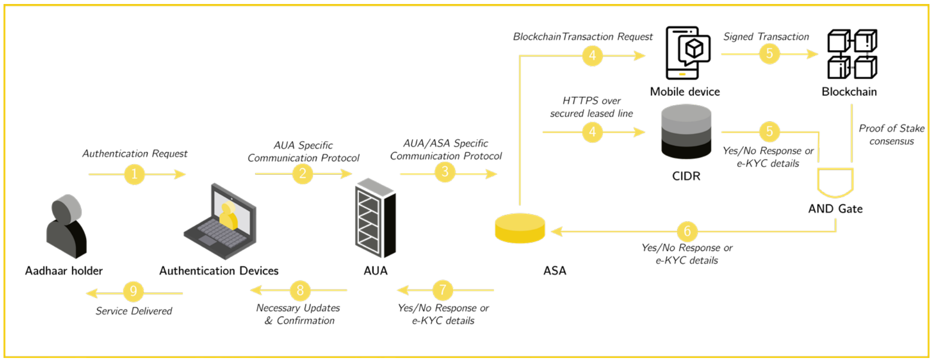

Fig.7: Updated Flow Diagram of Aadhaar Authentication

The various actors with their proposed role in the above solution architecture are as follows.

There are two distinct layers at work here. The first layer is the regular centralized layer with its stakeholders-the AUAs, ASAs, CIDR- at play. The process begins similar to the current architecture, with the Authentication Device processing a request from the Aadhaar Holder. This could be using face, iris, finger biometrics, or an OTP (one-time password) sent to the holder's mobile. This is forwarded to the AUA that forwards it to the ASA.

This is where the proposal differs from the current architecture. Whenever any ASA receives a hash to authenticate Aadhaar information (Yes or No response / eKYC information) from the CIDR, a Blockchain transaction is also invoked. The Aadhaar Holder then receives a transaction request on their mobile device confirming his or her identity. Once the holder signs the transaction with a decentralized private key, it is sent to a validator set to verify and add a block to the ledger. Each authentication request therefore would require the entire process of private key authentication and generation of a new block. All these logs would be shared on a public ledger with enough privacy provided by public key - private key cryptography. In the ideal case, at the same time, the CIDR returns its response (Yes or No / eKYC information). Now, if any one of these processes - the Centralized layer or the decentralized layer result in failure, the authentication is not successful and response not shared (Note: A response would not be generated itself if the central layer yields a negative result). In other words, both the layers share an AND gate relationship. Thus in this way, even internal bad actors in the UIDAI cannot influence the output.

Any block entry on the chain would require a consensus protocol requiring multiple stakeholders. While ideally any person or entity should be able to take part in the consensus mechanism, to begin with the stakeholders can be government entities (including different departments of national and state bodies), registered Aadhaar entities (ASAs, ASAs and Aadhaar's own servers). For the system to be truly decentralized, there needs to be an incentivization mechanism. By introducing tokens (that can be exchanged for money or other services) in a **proof of stake** consensus, third party vendors may be invited to put a stake in chain and to validate blocks. With their credentials and tokens as stake, there can be suitable remuneration for their services and penal provisions for playing foul.

The decentralized private key so discussed can be based on the uPort implementation of the same concept. The demographic, personal Aadhaar information along with biometric (fingerprint) data may be used to generate a unique hash (private key) to sign transactions.

The whole process is summarized in the following steps:

I. The Holder sends an authentication request using an authentication device
II. The AUA transmits the hash to the ASAs
III. ASA queries centralized Aadhaar server to get the data to complete authentication request.
IV. At the same time, a blockchain transaction request is sent by the ASA to the holder
V. Owner can accept or reject the request of authentication. If he/she signs the transaction, it undergoes proof of stake validation
VI. If both the processes are successful, only then the CIDR response is shared

VII. The log for the whole transaction is stored on Blockchain.

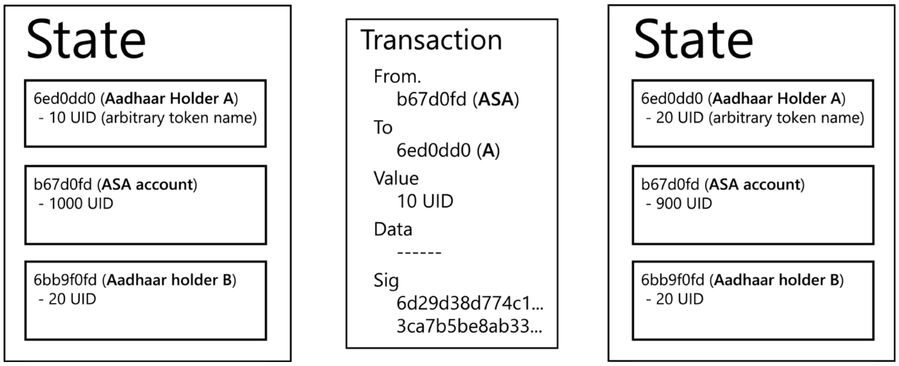

Fig.8: The blockchain layer as a State Transition System

5.5 Challenges with the Architecture

Decentralization: Aadhaar was never planned to incorporate blockchain technology and strictly follows a centralized server architecture. Suddenly introducing decentralization in a proven-and-tested centralized architecture haphazardly can lead to its own set of problems and result in internal resistance. Thus, implementing Aadhaar system on a public Blockchain is a task equivalent to making a Blockchain from scratch and needs months of conceptualization, PoCs and execution to implement properly.

Incentivization: While this doesn't seem like a problem in the early stages when taking part in a consensus protocol may be enforced upon nodes, incentivization needs to be tackled as the blockchain becomes more decentralized. In its most common form, economic incentivization would need careful thought-out introduction of tokens, protocols for introducing nodes and permissions from other government authorities. Certain authorities that are essential to this process such as the Reserve Bank of India especially are not in favor of cryptocurrencies in general and bypassing / convincing them of its benefits may take up considerable human effort and time.

Scaling & Latency: Scaling isn't a problem restricted to this implementation. Many blockchains such as Ethereum or Bitcoin can only process 12-15 transactions per second[15].That becomes a big problem when dealing with a system that has 1.3 billion users, slowing down the entire process. In the short term, therefore, a Blockchain verification cannot be a compulsory part of Aadhaar Authentication. To begin with, this proposal suggests passive verification on Blockchain that would be initiated only on user's request. In the long term, one can look towards scaling solutions on or off the chain such as sharding or plasma chains to increase the number to about 100,000 transactions per second.

Another short term solution can be by simply offering a blockchain log of transactions that outputs its result independently at a later time. If the transaction doesn't go through, then the Aadhaar Holder may get a notification so that he/she can look into an unauthentic transaction and raise a complaint.

Coordinated Attacks: When dealing with national government entities, there exist a lot of outfits meaning to negatively impact its operations. Thus, introducing a truly decentralized layer in Aadhaar Authentication would provide opportunities for all of these outfits to take part in the consensus protocol. Also, since Aadhaar is an identity for Indian residents, it makes sense to put geographical restrictions on the stakeholders (nodes) allowed to take part in the consensus. Thus, without any restrictions put in place, such a system is privy to 51% attacks and intentional slowing down of operations at the least. A PoS protocol does open doors to economic penalties and game theoretic mechanism design to discourage centralized cartels and protect against such attacks[16].

6 Conclusion & Future Scope

To summarize, introducing a second blockchain layer on top of the current centralized architecture of Aadhaar authentication can increase the integrity of the entire process. It would also make the system robust and secure against the malicious intentions of internal bad actors. Having all data on an immutable Blockchain would ensure the responsibilities of all the stakeholders for any questionable update or authentication.

Since shifting the entire Aadhaar database to the blockchain is not practical, the proposal suggests an extra Blockchain layer with all the transaction logs. Authentication in this proposal is akin to a two-step verification process where Aadhaar holders can opt for their prior permission for any transaction/ authentication. Signing such a transaction would involve a private key similar to uPort and incorporate a Proof of Stake consensus protocol. Although this layer would not stop or rollback communication with the CIDR, the Aadhaar holder can allow or disallow the transaction and it will be recorded on a distributed ledger.

Ideally, a fully decentralized Blockchain layer would open doors to many opportunities. A few interesting ones are listed below:

6.1 A State Backed Cryptocurrency

The incentivization of the blockchain layer (if made public) would require tokenization. These tokens can be ones that not only serve the purpose of running the chain but those which can be exchanged for fiat currencies. Thus, instead of transactions between just the ASA and an Aadhaar holder where a standard amount of this token is exchanged for authentication, these tokens (or now coins) may be exchanged between two Aadhaar holders and later "cashed in" for the Indian Rupee or other services. Thus the UIDAI can also provide a complete ecosystem where a state-backed cryptocurrency is launched.

This would better than other implementations (of state-backed crypto & other cryptocurrencies) as there is only partial anonymity here as innately every exchange would be between actual Aadhaar holders (and thus Indian residents) and thus fraudulent transactions can be tracked to a single person. This power would be seldom used however so as not to maintain government surveillance and privacy according to the Aadhaar Act of 2016. With other countries also moving towards state-backed crypto, this can India's moment. This cryptocurrency would comply with centralized authority and would provide an alternate medium to each and every resident to pay and to get paid. Businesses may also use this making a separate set of accounts from public users. This, however, is only possible with required set of permissions from Government entities such as the Reserve Bank of India.

6.2 Development Opportunities

Once backed with an Aadhaar based cryptocurrency, it would be easy to establish a complete ecosystem based on it. This ecosystem can thus also have a Smart Contract Platform to write applications for various business logic running on this cryptocurrency to make payments and to provide incentives.

In fact, almost all applications of Smart Contracts (DAOs, wallets, reputation based question-answering or even token based games) in Ethereum for example can be possible in such an implementation along with a new set of parameters offered through the pre-existing Aadhaar e-KYC parameters (demographic details, etc). One important part of the ecosystem could be data storage and management using large chunks of commodity hardware (available in the country) with Government and with private parties. Using commodity hardware would optimize and fulfil increasing demand for server and computational powers.

Exposing Aadhaar's Blockchain APIs to developers would help in building a universal ecosystem by which companies like Amazon and Walmart can also accept payment in this cryptocurrency and make applications to provide benefits to Aadhaar holders.

Using a state backed Cryptocurrency ecosystem would be highly beneficial for Indian entrepreneurs and start-ups where they can develop applications for Indian workforce. This in turn would be a great reform leading to a self-sovereign digital economy.

References

1. https://uidai.gov.in/aadhaar_dashboard/index.php
2 https://uidai.gov.in/images/FrontPageUpdates/aadhaar_authentication_api_2_0.pdf
3. https://uidai.gov.in/about-uidai/about-uidai/uidai-ecosystems.html
4. https://uidai.gov.in/authentication/authentication-overview/authentication-en.html
5. https://uidai.gov.in/images/the_aadhaar_act_2016.pdf
6. Haber, Stuart; Stornetta, W. Scott (January 1991). "How to time-stamp a digital document". Journal of Cryptology. 3 (2): 99–111. CiteSeerX 10.1.1.46.8740. doi:10.1007/bf00196791.
7. US patent 4309569, Ralph Merkle, "Method of providing digital signatures", published Jan 5, 1982, assigned to The Board Of Trustees Of The Leland Stanford Junior University
8. https://www.forbes.com/sites/bernardmarr/2018/02/16/a-very-brief-history-of-blockchain- technology-everyone-should-read/#4c3722277bc4
9. **Ethereum Whitepaper:** https://github.com/ethereum/wiki/wiki/White-Paper
10. **Civic Documentation:** https://docs.civic.com/
11. **uPort Developer Website:** https://developer.uport.me

[12] https://blockgeeks.com/guides/blockchain-consensus/

[13] https://www.blockchain.com/pools

[14] https://medium.com/@VitalikButerin/a-proof-of-stake-design-philosophy-506585978d51

[15] **Ethereum Yellowpaper:** https://ethereum.github.io/yellowpaper/paper.pdf

[16] https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQs