**Blockchain Layer for Aadhaar System**

* **Drawbacks of Current System**

1. Attacks on Aadhaar Database
2. Authenticating Agencies, employees misusing Biometric Templates/UIDs of Customers
3. Misuse of Aadhaar UID
4. No transparency as to who is using your Aadhaar UID or if anyone has used your biometric template.
5. No proof from where attack has come from.
6. No monitoring or saving of queries.

* **Proposed System**

UIDAI CIDR

ASA

ASA

ASA/AUA

AUA

AUA

AUA

5. SUCCESS: Create a transaction of UID and Employee details

5. SUCCESS: Query CIDR using UID & Biometric Template

**Global Public Ledger**

6. Create Block of Transactions and write block into Ledger

4. User Details and Biometric Template. Employee UID, IP Address, Location & Timestamp

5. FALIURE

6. FALIURE

5. FALIURE

7. SUCCESS: User Aadhaar Card is Generated

4. UID

4. List transactions of given UID. Create Block of Transactions and write block into Ledger

6. SUCCESS: Create a transaction of query, UID and Employee details

Register Node

Query Node (Banks, Network Providers,PAN)

7. Create Block of Transactions and write block into Ledger

Query Node (Banks, Network Providers,PAN)

8. Aadhaar Card

3. UID

3. User Details and Biometric Template. Employee UID, IP Address, Location & Timestamp

**Private Ledger**

8. SUCCESS: Transaction Status

2. Employee UID, IP Address, Location & Timestamp

3. User Details and Biometric Template. Employee UID, IP Address, Location & Timestamp

2. Query Transactions using UID

PoS

2. Employee UID, IP Address, Location & Timestamp

9. Status of Transaction

**Aadhaar Employee**

5. List of Transactions of UID

PoS

PoS

1. User Details & Biometric Template

**Banks, Network Providers, PAN Employee**

1. User Details & Biometric Template

9. Aadhaar Card

6. List of Transactions of UID

1. UID

10. Status of Transaction

**End User**

**End User**

**End User**

**AADHAAR RELATED OPERATIONS**

**REGISTER NEW USER TO AADHAAR**

**QUERY ALL TRANSACTIONS**

The new system will integrate an additional layer that provides security and transparency. The layer would be an intermediate between the POSs and services provided by UIDAI and the UIDAI backend server.

The layer would be made of a blockchain technology (preferably hyperledger). Key features are:-

1. Recording query of API as well as location and IP of query as transactions in ledger for better transparency.
2. Private channels between POSs and intermediate layer (Hyperledger layer) so each PoS can record, distinguish and monitor attacks.
3. Private channels act as proof as to where attack came from.
4. POSs can monitor their own channel and will therefore know the exact system/desktop and employee associated with it.
5. Monitor and regularize from where and number of queries.
6. Attacks on the system would have to be mass produced on all channels, no single point of attack.
7. End user can see who has queried his Aadhaar number and from where.
8. End user will know if anyone is misusing his/her biometric template.
9. End user notifications.

* **System Entities**
* Current Entities
  + UIDAI CIDR Servers
  + Authentication User Agency (AUA) Servers
  + Authentication Service Agency (ASA) Servers
  + Operators at PoS devices

* New Entities
  + Hyperledger Nodes hosted by Government, Banks and Service Providers using Aadhaar APIs (excluding end users)
* **User Stories**

1. Normal User

Any Aadhaar Card holder can use his/her UID and query the system to get a list of all previous transactions (10 transactions at a time) done using his/her UID.

1. Bad Customer

A customer knows another person UID. He/she can use that UID and get various public information and previous transaction details of the UID holder.

1. Bad Employee

An employee working at the PoS can use various means and keep a copy of a customer’s fingerprint along with keeping a track of the customers UID. Using the fingerprint copy and the UID, the employee can pretend to be the customer and access the system.

* **APIs Design**

1. Register API

https://<host>/<ver>/<ac>/<asalk>/<euid>/<cdetails>

1. KYC API (Other operations such as linking Bank account or PAN)

https://<host>/<ver>/<ac>/<asalk>/<euid>/<cuid>

1. Query Transactions API

https://<host>/<ver>/<ac>/<asalk>/<cuid>

API Elements Details

host: Aadhaar authentication server address. Actual production server address will be provided to ASAs. Note that production servers can only be accessed through private secure connection. ASA server should ensure that actual URL is configurable. (For development and testing purposes, public URL “auth.uidai.gov.in” can be used.)

ver: Authentication API version (optional). If not provided, URL points to current version. UIDAI may host multiple versions for supporting gradual migration. As of this specification, default production version is “2.0”.

ac: A unique code for the AUA which is assigned by UIDAI. This is an alpha-numeric string having maximum length 10. (A default value “public” is available for testing.)

asalk: A valid ASA license key. ASAs must send one of their valid license keys at the end of the URL. It is important that license keys are maintained safely.

euid: UID of Employee working at PoS.

cuid: UID of Customer using the service.

cdetails: Demographic details of the customer registering for a new Aadhaar Card.

**Note**: IP Address will be obtained from the request.

Timestamp for a transaction will be obtained from the node system time.

* **Benefits of New System**

1. Fewer attacks on Aadhaar CIDR.
2. Unnecessary database queries won’t reach the server, hence decreasing load.
3. Service Providers and Banks can find fraudulent employees quicker using their private ledger.
4. Service Providers and Banks can use their private ledger to ensure customers of integrity of their system in times of peril.
5. If a single Service Provider or Bank gets hacked and has to shut down operations, it can do so without hampering the operations of other Service Providers or Banks.

* **Why use Hyperledger?**

1. Permissioned Membership
2. Performance scalability and levels of trust
3. Data on a need to know basis
4. Rich queries over an immutable distributed ledger
5. Modular architecture supporting plug in components
6. Protection of digital keys and sensitive datay