D.I System

Decentralized Identity Verification Platform By Police

■ Description

We use hyperledger sawtooth for building this project . its a identity system which uses decentralized mechanism . Its similar to the working or how we use aadhar in public scenarios . its mainly focus on the security of every persons personal details which cannot access by others without their permission . Only person who was accepted by police can get their verified public key .

By implementing this project , there is no need to carry an hardcopy of identification cards for various purposes , only need the public which is verified by police and users encryption key

The existing Kyc system is not secure enough , hackers even get personal data and they try to sell to others . Using blockchain technology we can solve this issue . The data about each persons added to the block which cannot be altered. Its really take lots of times when we reapply aadhar or voter id even if loss , or it maybe damaged.

In current scenario paper based works are exists like for aadhar , voter id card what if they were damaged or even loss , By implementing this project all personal data are secured through blockchain technology and there is no need to carry identification cards to various office’s for various purposes only the public key and encryption key is needed so the transparency becomes more.

The details only we want to share should be able to see to others. Mainly the details like address , aadhar number and voter id no are in encrypted format to ensure more security. Using blockchain incase of identifications data of peoples we can ensure more security and which is also reliable .The data should not be loss from the blockchain.

■ INSTRUCTIONS FOR SETTING-UP APPLICATION

Requirements:

- OS: Ubuntu 18.04 (Recommended)
- NodeJs version 8.0 stable npm latest
- Docker

Steps:

- Clone & Navigate into main directory
- Run "sudo docker-compose up"
- Open Browser & navigate

LocalHost - http://localhost:3000
Ip Address- http://127.0.0.1:3000

There are three clients in this application

• User
1. View Data
2. Change Enc Key
3. Add Or Edit Data
• Police
1. View Unveried Users
2. Approve them By giving a Verify Tag
3. Delete the from chain by Reject Them
• Client
1. Only A verifed User can Login
2. Check User by their Public Key

■ How to add permissions

In sawtooth if you are giving permission(PERMIT_KEY) to a key, then all other keys are not allowed to submit transaction.

To create keys use the cmd :

sawtooth keygen user
sawtooth keygen police

We hardcoded the private key of police inorder to login , becase the key is always changed each time when we down and up the compose file .

Note : When creating permissions we have to check for the settings-tp , identity-tp is currently present in yaml file and it should be running , if not it shows wait time and pending errors and policy could not be committed .

SETTING UP PERMISSIONS AND ROLES

The permissions are set by using Identity transaction family which defines permissions for sawtooth components by adding policies and roles

Add public key to the list of allowed keys in order to change settings ,

sawset proposal create --key ~/.sawtooth/keys/my_key.priv

sawtooth.identity.allowed_keys=$(cat ~/.sawtooth/keys/my_key.pub) --url

http://rest-api:8008

Here ‘ my_key’ is provided in yaml file .This key is used to create genesis block .Here we are adding the public key ‘my_key.pub’ to the list of allowed keys to change settings.

In order to check the public key has been added to settings , use the cmd below to check the public key is added to allowed keys ,

sawtooth settings list --url http://rest-api:8008

Once the public key is added to settings , then we can create , update roles and policies. Policy is a set of identities which has the permission to perform some action .Policy include a set of permit and deny roles which can be used to permission the access to a validator network and which transactor can participate on the network. PERMIT_KEY to permit transactors and DENY_KEY to block specific transactors. A role simply reference the policy, a role should be assigned as a transactor ,transactor.transaction_signer ,transactor.batch_signer etc .

Each line contains either a PERMIT_KEY or DENY_KEY followed with either a public key for an allowed transactor or the * character to specify all possible transactors. To define multiple permitted or denied keys, use additional PERMIT_KEY or DENY_KEY lines.

For example, running the following command will create a policy that permits all and is named my_policy:

sawtooth identity policy create --key
~/.sawtooth/keys/my_key.priv my_policy "PERMIT_KEY *" --url
http://rest-api:8008

When giving permission to specific keys :-

Suppose we are going to give permission to any users for eg Rahul ,Ramu
we have to create 2 pairs of keys named Rahul and Ramu. Use the cmd

sawtooth keygen Rahul
sawtooth keygen Ramu

But before giving permissions to Rahul or Ramu we should give the public key (my_key) the permission to be a transactor. So we have to use the public key of my_key and give PERMIT to it .

To get public key of my_key use the cmd ,

cat ~/.sawtooth/keys/my_key.pub

We have to create a policy first .use the cmd ,

sawtooth identity policy create --key
~/.sawtooth/keys/my_key.priv policy_1 "PERMIT_KEY
my_key " --url http://rest-api:8008

Here replace the ‘my_key’ in policy file with the public key ,

The next step is to create a role for this policy. Here we are giving the transactor role for the my_key’ .Use the command ‘ sawtooth identity role create‘ to create a role for this policy.

sawtooth identity role create --key
~/.sawtooth/keys/my_key.priv transactor policy_1 --url
http://rest-api:8008

To view the role, use the following cmd

sawtooth identity role list --url http://rest-api:8008

it should display the output as ,
transactor: policy_1

The final step is to give permission to Rahul or Ramu by editing the policy command again by adding the keys of our choice

sawtooth identity policy create --key
~/.sawtooth/keys/my_key.priv policy_1 "PERMIT_KEY
my_ key " “PERMIT_KEY key_of_ Rahul ” --url http://rest-api:8008

were ‘my_ key’ is the public key that we used before to create a policy and key_of_ Rahul is the public key of Rahul .We can provide permission for both Rahul and Ramu like this ,

sawtooth identity policy create --key
~/.sawtooth/keys/my_key.priv policy_1 "PERMIT_KEY
my_ key " “PERMIT_KEY key_of_ Rahul ” “PERMIT_KEY key_of_ Ramu ” --url
http://rest-api:8008

These are the steps required to set up permissions in a sawtooth network