Self-Sovereign Identity

Project Report: Group - 9

Bhavya Jain
Technical University of Delft
Netherlands
B.Jain@student.tudelft.nl

Himanshu Shah
Technical University of Delft
Netherlands
H.Shah@student.tudelft.nl

Umeer Mohammad
Technical University of Delft
Netherlands
U.A.Mohammad@student.tudelft.nl

Shivanand Kohalli
Technical University of Delft
Netherlands
S.C.Kohalli@student.tudelft.nl

Sandesh Manganahalli Jayaprakash
Technical University of Delft
Netherlands
S.MANGANAHALLIJAYAPRAKASH@student.tudelft.nl

ABSTRACT

Current Identity implementations are still struggling to digitize their solution for identity management. Digitization of the highly sensitive private information faces multitude of challenges from security to false identity generation. Further, sharing identity in traditional methods require sharing any and every information present in the identity card. We propose a digital implementation of Identity using BlockChain technology where an individual can share only the required information, for a single use. The proposed solution prevents tampering of personal information once it has been verified by concerned authority. Attack resilience is built in the system as any modification will render the information useless because only verified information can be shared. The solution also utilizes the concept of Zero-Knowledge Proof (ZKP) for providing information in certain cases like age of an individual. We propose an universal solution which handles all the complexities involved but at the same time is simple enough for everyone to use. To enable ubiquitous, global information sharing, requiring minimal user interaction and knowledge about the implementation, we decided to implement the solution for Android smartphone.

1 INTRODUCTION

Identity Information is highly sensitive data which requires frequent sharing and control. Traditional identity solutions involves presenting multiple identity relation documents for an individual entity, each providing certain information about it. With time, the number of identity documents/cards that an individual or business possesses have increased. Identity information is shared for various purposes almost everyday, from entering premises of institutions or offices to restaurant to boarding a train, almost every activity requires user identification. Various countries have multitude of implementation of identity services which issue documents or identity cards with verified information about an individual entity. The most generic form of identity cards contain enough information to ensure identity verification for most daily activities.

However, these solutions require all the information present in the ID card to be shared with the concerned person/authority irrespective of the fact that only a certain amount of information would have satisfied the needs. The common example is of a bar. In order to verify the age of the individual to be above a certain number, let's say 18, one has to produce a legal document that has the date of birth of the user. Here, along with the age, the actual date of birth and other additional information present in the document like complete name, address/nationality etc are also shared while the requirement was a simple yes or no if the user is of the age.

Our application is a step towards implementing decentralized, private blockchain for *Self Sovereign Identity*. Self-sovereign identity is the concept that people and businesses can store their identity data on their own devices, and provide it efficiently to those who need to validate it, without relying on a central repository of identity data.The proposed solution works on the web of trust with information stored in smartphones in form of trusted signed certificates. The solution provides complete information control to user, with options to select and share amount of information required information by the validator. This removes reliance on a central data repository of identity data and provides complete control to the individual. Resilience to false identity is built by using government authorities to verify the information stored in the blockchain by providing digital signatures.

Information sharing should not only be secure, but also quick and easy. For this, we have incorporated verification using the QR code. A QR code is generated for the purposes of identity verification. The application can generate and read QR code and function autonomously with minimal user interaction.

2 PROBLEM STATEMENT

Create a BlockChain based solution that will form a web of trust. This BlockChain will have user identity information stored in form of trusted signatures from concerned authorities. The solution should be stand-alone, decentralized platform with private blockchain implementation. The implementation should provide complete ownership and control over the data to the user or enable *Self Sovereign Identity*. Essentially, to convert paper based identity solution to digital solution, only more secure and reliable. The system developed should be tamper proof and attack resilient. It should not permit fake users or incorrect information to be used for verification.

The solution must be easy to use and ubiquitously accessible. It should involve minimal user interaction on the parts of the individual entity and validator. The information stored should be easily verifiable from concerned authorities and information sharing in the application should be intuitive to support all user demographics ranging from level of education to various age groups.

3 SYSTEM DESCRIPTION

3.1 Functionality

The main functionality of the application can be divided into two parts i.e. Authentication and Verification. Both of these functionalities are built in the same application.

3.1.1 Authentication

The information a user stores in the system needs to be authenticated before sharing. For this purpose, user enters the information in the application. Using the application, user can get his/her information authenticated by the trusted authority in two ways: using normal authentication method or using zero-knowledge proof. Information ranges from Name and Age to Gender and Social Number. Each information is added separately by the user and needs to be individually authenticated by the authority. Once user has entered information for a particular attribute, he/she can request for authentication from the authority. The authority also enters the user's value to be authenticated according to its records and loads the QR code. The user then scans the QR code and obtains the information about the IP address, port number for communication and also authorities public key quickly and effortlessly.

Now, the user's application creates a TrustChain[4] half block with the hash of the entered data as the value in the transaction field, public key of the user, public key of the authority, sequence number of this block in user's blockchain. This half block is sent to the authority using the obtained IP address and port number. Separately with this half block, the actual entered data is also sent to the authority. The authority computes the hash of the data that is sent separately and then checks whether the data in transaction of half block is same as that of computed hash. If it approves of it, then it creates a full block with its sequence number in the blockchain and signs the block and sends it to the user, which is stored in the user's blockchain and can be used for verification with third party vendors later. The authority also appends the block in its local blockchain. The authentication process is shown as a flowchart in figure 1. The procedure for authentication using a zero knowledge proof method follows the algorithm as explained in section 3.2. In the transaction field of the full block instead of hash of the entered data, the

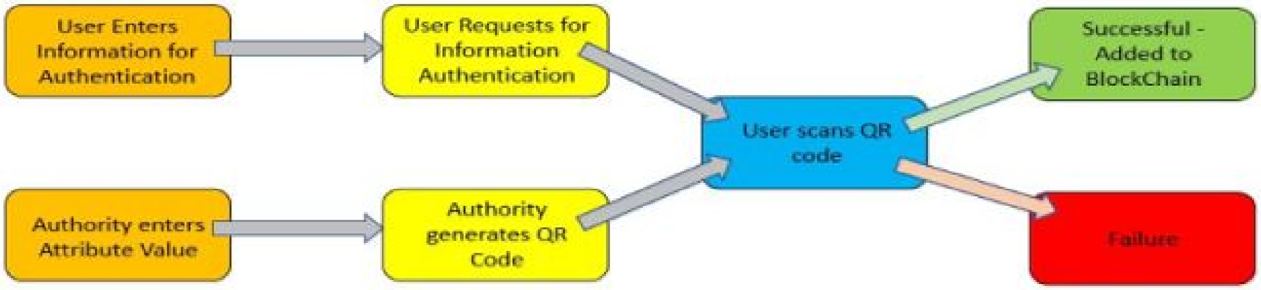

User Enters Information for Authentication
User Requests for Information Authentication
User scans QR code
Successful - Added to BlockChain
Authority enters Attribute Value
Authority generates QR Code
Failure

Fig. 1 : Authentication Process Flow Diagram

proof generated by the authority is stored in this case.

This completes the authentication procedure, and to check whether it was successful, both user and authority can open their Chain Explorer and observe that a new block has been added to their blockchain. The user can verify his authenticated attribute now, as he has a signed block in his blockchain from the authority.

3.1.2 Verification

Once a user has got his personal information authenticated by the authority, he can use it to verify his details against vendors or any third-party service providers. The application facilitates easy and secure validation of the user information which complies with the interests of both the user and verifier. The privacy of the user is the foremost priority but at the same time it is also taken care that the validation process is done authentically and the information to be verified is accurate. In this report the words validation and verification have been used interchangeably.

Similar to authentication, verification can also be done in two ways: Normal verification and verification using Zero-Knowledge Proof. The flow diagram for the validation process is shown in figure 2. In normal verification, the verifier selects the information that he wants to verify and generates a QR code, in which the details to be verified are encoded. The user then just scans the QR code, after scanning the QR code the application checks whether the requested information exists in the user's blockchain. If the information does not exists, user gets a pop-up stating the he cannot provide the requested information. If the information exists, then application provides the user with a choice whether to share the information to the verifier or not. When the user grants the permission to share his information, then the block containing the information is sent to the verifier. Along with this the also sends his actual information in a separate block. Upon receiving the information from the user, the application on the verifier side verifies the signature of the authority, then computes the hash of the information user provided along with the information present in the user's block from his blockchain. If the data and signature matches, then the verification process is termed as successful and user and verifier can exchange services. If either the signature of the authority does not match or the data does not complies with the data provided by the user, then the verification process fails.

In verification using Zero-knowledge proof method, the user need not share his exact data with the verifier instead he can share the proof generated by the authority during authentication. Similar to normal verification process, the verifier generates QR code and user scans it. Upon scanning QR code, system checks if the required information exists in the blockchain and whether it is provable, if it exists the system calculates the $proof$ accordingly as explained in section 3.2 and shares this $proof$ and also the block in which the corresponding hash-chain is stored, and the application now computes the required value at the verifier's side. If the data matches, the verification is deemed successful else failure.

3.2 Zero knowledge Proof

The implemented zero knowledge proof helps to prove that an integer such as age or a balance is greater than a given threshold value without revealing the original information to the verifier. The outline of the algorithm [1] considering Age as the attribute to be verified is as follows.

In the setup phase, the user sends his actual Age to a trusted third party(Authority). The authority if it approves of it, generates a Random number R and calculates a hash-chain [2] as follows $E = Hash^{Age+1}(R)$. It sends the both the values R and E to the user.

In the challenge phase, a verifier requests the user to prove

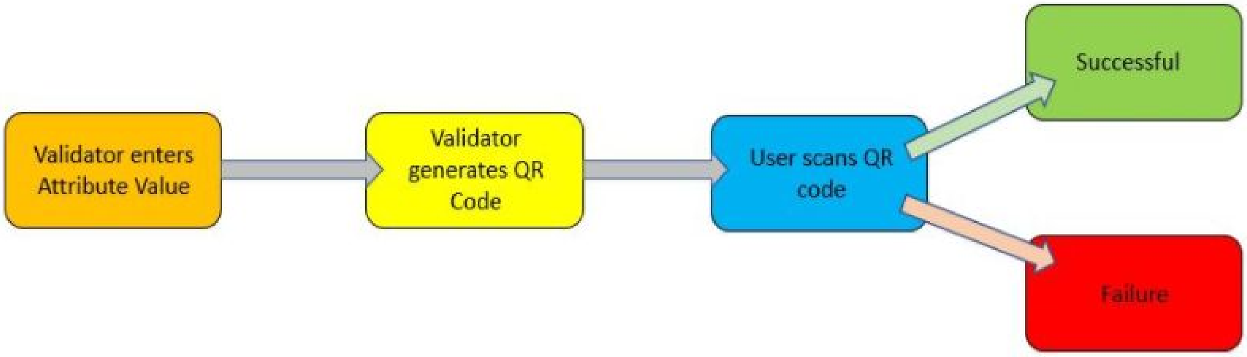


Fig. 2 Validation Process Flow Diagram

if his age is above $toProve$ (some minimum required age).

In the construction phase, the user calculates the proof as follows $proof = Hash^{1 + Age - toProve}(R)$

and sends this $proof$ and $E = Hash^{Age+1}(R)$ received from the authority.

In the verification phase the verifier calculates $verificationProof = Hash^{toProve}(proof)$ and checks if $verificationProof == E$. If this condition matches, it means that the random number R must be received from the signed authority and thus proves that Age is above $toProve$ without the user actually revealing his age. Thus, it ensures the authenticity of the data but at the same time not compromising on the user privacy.

In the block of the blockchain $E = Hash^{Age+1}(R)$ will be stored as this value will not change. For verification, $proof$ will be calculated by the users application and sent to the verifier.

4 CRITICAL DESIGN CHOICES

4.1 Zero-Knowledge Proofs

The idea to incorporate Zero-knowledge proof is to facilitate the user to prove some information about him/herself without revealing his actual detail. Since some application have some generic requirements for which telling the user identity is not necessary, we chose to incorporate zero-knowledge proof authentication and verification methods. As discussed earlier in the example of bar, the requirement that needs to be satisfied is just the user is above 18 years and does not needs any further information about the user. So using ZKP, user can fulfill such requirement by ZKP without actually revealing information about him/her.

We chose the zero knowledge proof as in [1] , because of its simplicity, ease of implementation and efficiency as opposed to other proofs such as the zkSNARK's.

Also, we decided to incorporate non-interactive ZKP because interactive ZKP requires series of questions that must be satisfied whereas non-interactive ZKP is a one time process. Moreover, non-interactive ZKP reduces the communication overhead present in an interactive ZKP proof.

4.2 TrustChain

TrustChain is a permission-less tamper-proof data structure for storing transaction records of agents [4]. It allows every individual to maintain and control his personal chain of transactions, as opposed to traditional blockchains such as Ethereum or Bitcoin where a global chain for all users is maintained. This property allows to reach consensus among only the participants rather than adding block to a global blockchain. This facilitates the users complete control of their identity and personal information.

We chose TrustChain implementation over others because it offered the flexibility to alter the framework as per our needs and did not require a global consensus, instead the user can just get the data authenticated from the authority and get it added to his personal blockchain. This model also resembles with the prevailing processes of identity authentication where government or other identity issuing organization is the ultimate authority. Hence, it helps to digitalize the current process while still letting the control in the hands of same authority as it is presently.

5 RESULTS AND CONCLUSION

In our attempt, we strived to make an application that could abide by as many principles of the self-sovereign identity as much possible[5]. The principles that are fulfilled by the application are listed below:

- **Existence**: "Users must have an independent existence". Since the app enables authentication using existing identity documents which are unique to each user. It ensures users have independent existence.
- **Control**: "Users must have total control over their identity". The application asks for users permission whenever the user needs to share the information with a third party and all the users information is stored in the users device, thereby making user the authority of their identity.
- **Access**: "Users must have access to their data." Users can always access their data using the app.
- **Transparency**: "Systems and algorithms must be transparent"
- **Persistence**: "Identities must be long lived." Identities once added on the app, remain to exist and cannot be tampered.
- **Interoperability**:"Identities must be as widely usable as possible". The implementation of the solution as an mobile application ensures that it could be downloaded from anywhere in the world by anyone and used for identity management.
- **Consent**: "User must agree to sharing of their data". Data is always shared only with the user content.

We were able to build a basic android implementation of self-sovereign identity which was built on top of Trustchain Blockchain.

The operations of Authentication and Verification were integrated into a single application. It enables authorities to authenticate information digitally, on the other hand, it provided the functionality of verification for the users,, all in a single platform.
For certain fields, Zero-Knowledge proof was implemented to prove the existence of certain information, without actually revealing the information.

6 FUTURE IMPROVEMENTS

The possible future extensions that are possible to the application are listed below :

- Extend app to support other modes of communication, such as NFC and bluetooth.
- Currently the app works only on texts, but its functionality can extended to support exchange of digitally signed files.
- It must include mechanism to restore user data in case of device damage or theft, so that the user need not perform all the authentications again.
- The UI of the application could be made more user-friendly so that it could appeal to the a greater range of audiences without the need of intricate details about the processes involved.
- Displaying to the verifiers as to who has authenticated a particular block.
- Application authentication based on biometric information or a pass.

REFERENCES

[1] "Verifiable Auctions for Online Ad Exchanges - UT Computer Science." https://www.cs.utexas.edu/~sebs/papers/vex-sigcomm13.pdf.

[2] "Hash chain - Wikipedia." https://en.wikipedia.org/wiki/Hash_chain.

[3] "A gentle introduction to self-sovereign identity | Bits on blocks." 17 May. 2017, https://bitsonblocks.net/2017/05/17/a-gentle-introduction-to-self-sovereign-identity/.

[4] "TrustChain: A Sybil-resistant scalable blockchain - ScienceDirect." 1 Sep. 2017, https://www.sciencedirect.com/science/article/pii/S0167739X17318988.

[5] “The Path to Self-Sovereign Identity - coindesk” 27 Apr 2016, https://www.coindesk.com/path-self-sovereign-identity/

APPENDIX

A.1 User-Application Interaction Example

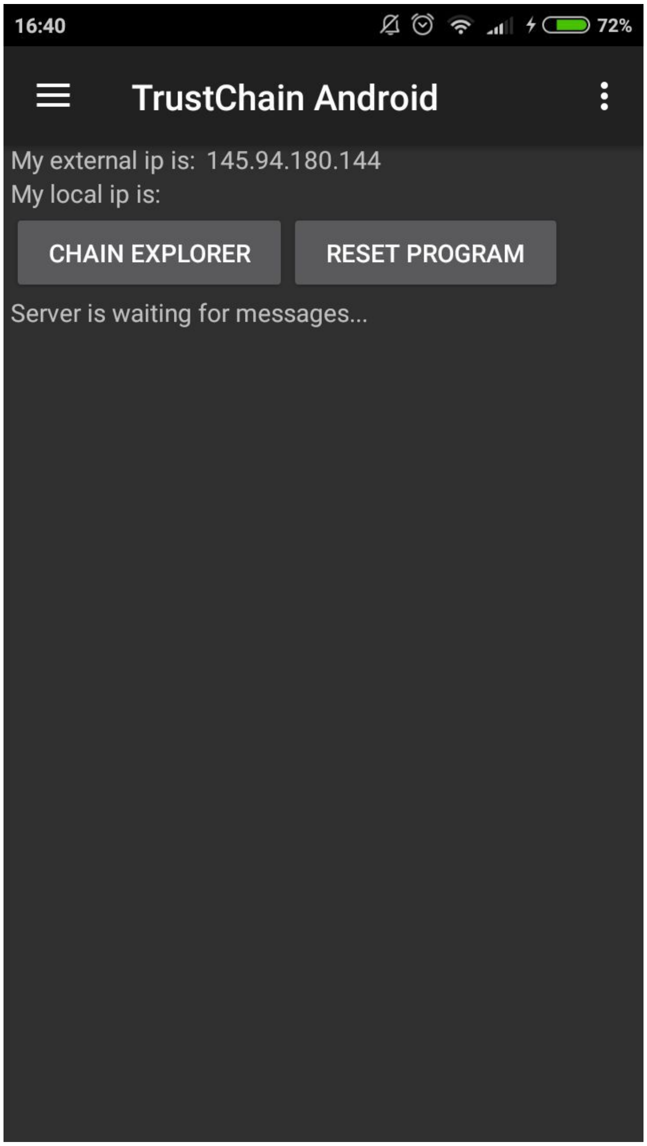
Let's start off with the simple example of a bar. In order to verify the age to be above a certain number, let's say 18, one had to produce a legal document that has the date of birth of the user. Here, along with the age, the actual date of birth and other additional details present in the document like complete name, address/nationality etc are also shared while the requirement was a simple yes or no if the user is of the age.

There can be a solution for the problem where user can carry a bunch of various cards specifying individual identity, or more simply by hiding the non-essential values by hand. But the approach has to be more secure where a mistake of not placing hand properly on card won't cause you to share details you don't want.

Our application is a step towards implementing decentralized, permissionless blockchain implementation of self sovereign identity.

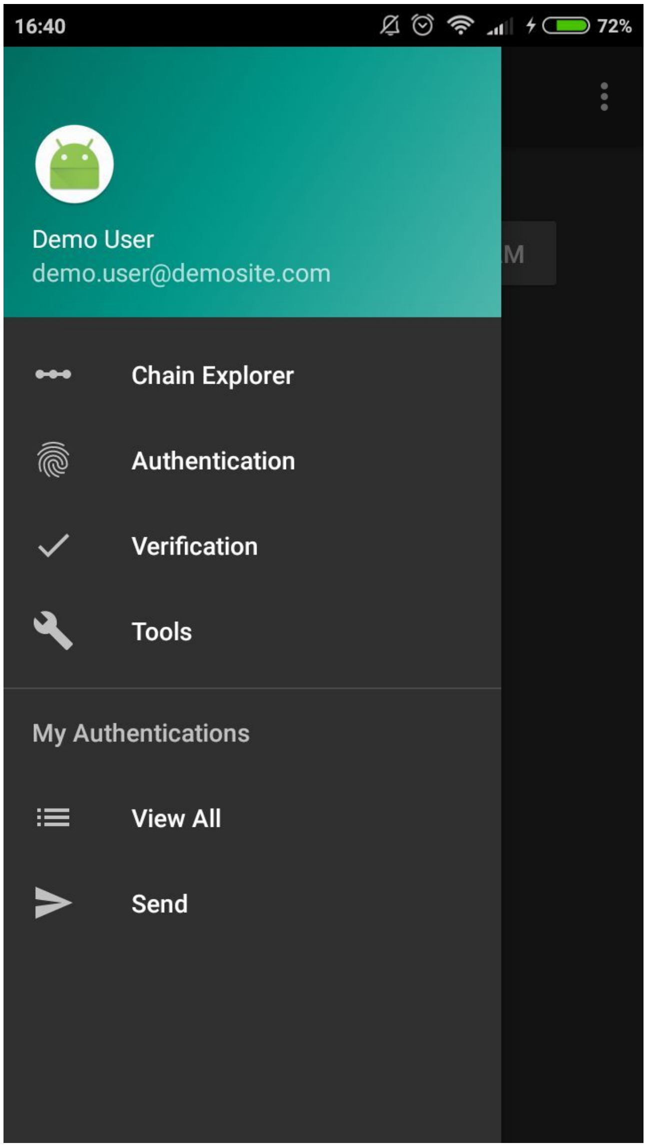
Main functionalities are:
Chain Explorer - Explore the generated blockchain.
Reset the application data - For testing and debugging
Authenticate Identity Details with government
Validate the identity information with a 3rd party seeking it.
View All Authentications - In this part the user can check all his successfully completed authentications done in past.

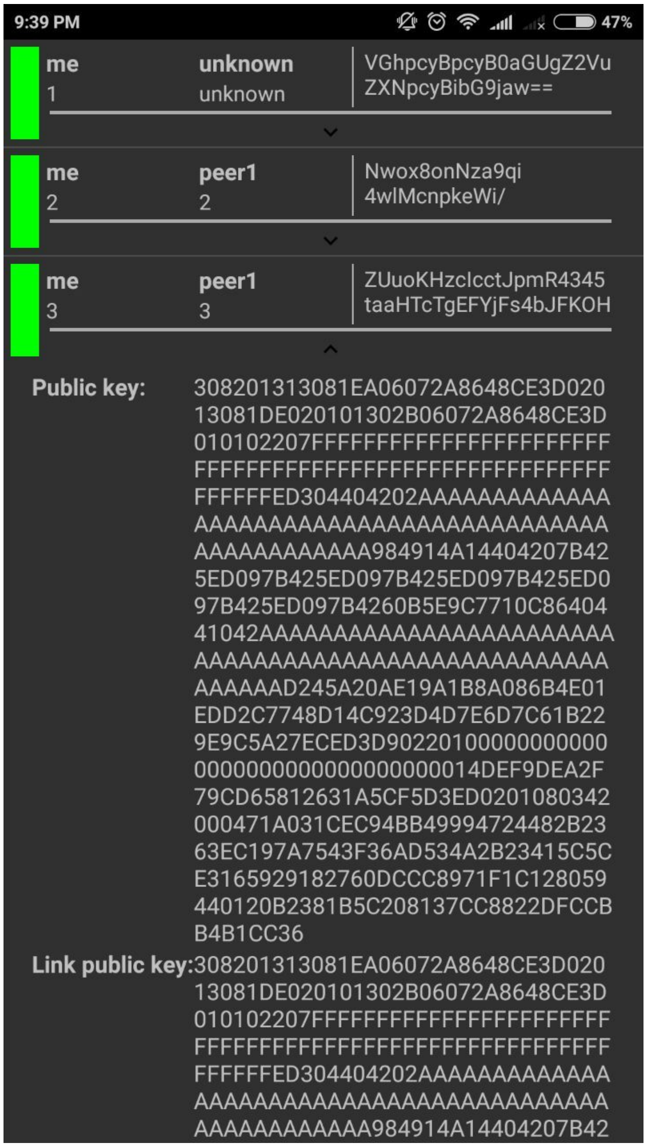
On the left you can see a screenshot of the homepage that can be seen by the application users.



My external ip is: 145.94.180.144
My local ip is:

Server is waiting for messages...



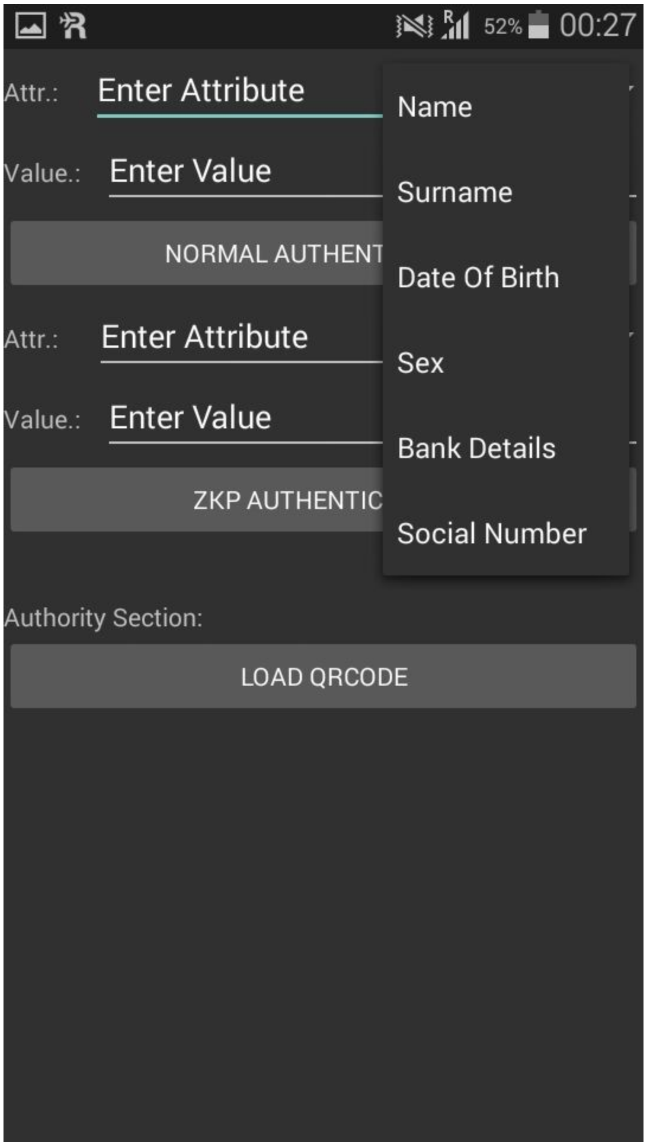


```
me    unknown   VGhpcyBpcyB0aGUgZ2Vu
1     unknown   ZXNpcyBibG9jaw==

me    peer1     Nwox8onNza9qi
2     2         4wlMcnpkeWi/

me    peer1     ZUuoKHzclcctJpmR4345
3     3         taaHTcTgEFYjFs4bJFKOH

Public key:       308201313081EA06072A8648CE3D020
                  13081DE020101302B06072A8648CE3D
                  010102207FFFFFFFFFFFFFFFFFFFFFFF
                  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
                  FFFFFFED304404202AAAAAAAAAAAAA
                  AAAAAAAAAAAAAAAAAAAAAAAAAAAA
                  AAAAAAAAAAAA984914A14404207B42
                  5ED097B425ED097B425ED097B425ED0
                  97B425ED097B4260B5E9C7710C86404
                  41042AAAAAAAAAAAAAAAAAAAAAAAA
                  AAAAAAAAAAAAAAAAAAAAAAAAAAAA
                  AAAAAAD245A20AE19A1B8A086B4E01
                  EDD2C7748D14C923D4D7E6D7C61B22
                  9E9C5A27ECED3D90220100000000000
                  00000000000000000000014DEF9DEA2F
                  79CD65812631A5CF5D3ED0201080342
                  000471A031CEC94BB49994724482B23
                  63EC197A7543F36AD534A2B23415C5C
                  E3165929182760DCCC8971F1C128059
                  440120B2381B5C208137CC8822DFCCB
                  B4B1CC36
Link public key:308201313081EA06072A8648CE3D020
                  13081DE020101302B06072A8648CE3D
                  010102207FFFFFFFFFFFFFFFFFFFFFFF
                  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
                  FFFFFFED304404202AAAAAAAAAAAAA
                  AAAAAAAAAAAAAAAAAAAAAAAAAAAA
                  AAAAAAAAAAAA984914A14404207B42
```

On the left side there is a screenshot of the overlay navigation menu and on the right side you can see the chain explorer activity.

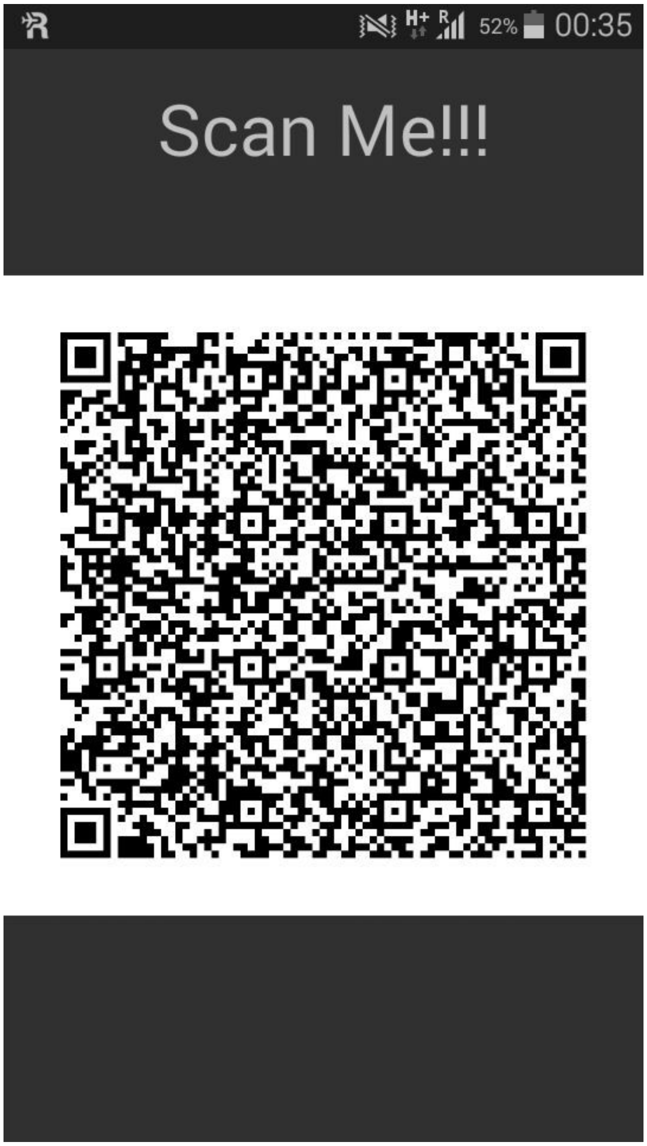
In this page you can get information about who authenticated information with whom, the blocks, public and link public keys, transaction hash etc.

The application supports the authentication and verification of the identity of the user. The user can place his/her identity values in the application and then request an authentication from the concerned authority.

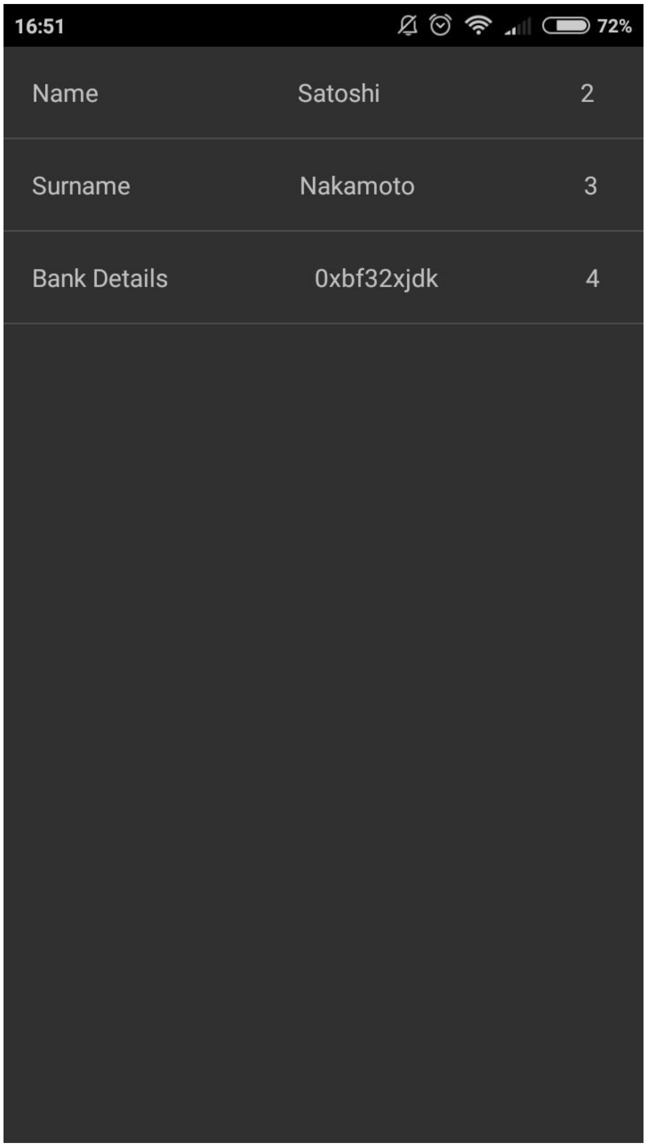


User here enters value in "Attr" and goes for authentication. Authority enters the value in "value" and loads a corresponding QR code. User then can scan the QR code and the application will authenticate the information. For the purpose of ZKP, similar process is followed.

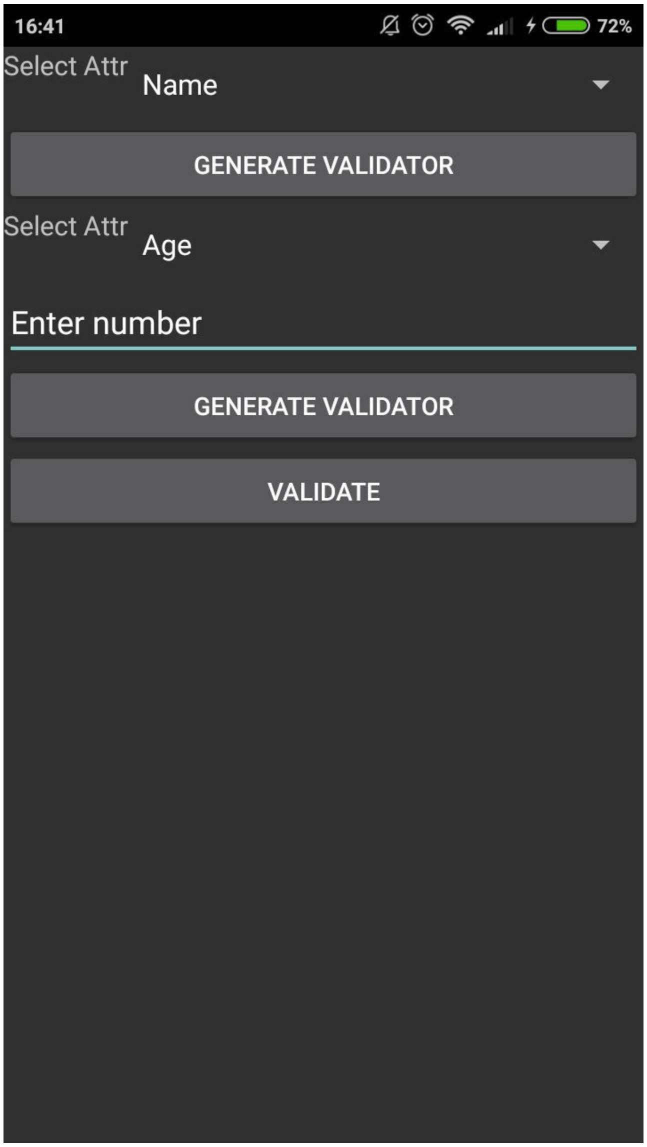
Here is a screenshot of a QR code generated.



Now user can use the app for verification purposes wherever required and information sharing is deemed fit from the user. The user can also check all the successful authentications completed till date by using the view all authentications options. The screenshot for the same is shown below:



Name	Satoshi	2
Surname	Nakamoto	3
Bank Details	0xbf32xjdk	4



Select Attr Name

Select Attr Age

Enter number

For Validation of the information with the third party vendors or service providers, the user can use his authenticated information and serve the task. To accomplish this, user can select the verification option from the navigation drawer menu and then fill in the details required to be validated. The third party then generates QR code which user can scan and share his details for validation. The screenshot for verification activity is shown below: