Tamper Proof Birth Certificate Using Blockchain Technology

Maharshi Shah, Priyanka Kumar

Abstract--- With the advancement of technology in modern era everything needs to be digitalized to make it more secure and reliable. Nowadays birth certificate is the only age proof of an individual and can be used to apply for a job, for admissions in collages/universities and basis of all the important government document identities like Aadhar card, Pan Card, Passport and other related matters. So identifying the correct birth certificate of any person is a major challenge. In the current system, afterbirth of any individual the birth *has to be registered with the concerned local authorities within 21 days of its occurrence, then it should be filled up the form prescribed by the Registrar and then* Birth Certificate *is issued after verification with the actual records of the concerned hospital. However, current methodology used for birth certificate verification is costly and very time-consuming. Therefore, our objective is to propose a theoretical model for issuing birth certificate and verification of genuine birth records using blockchain technology. This technology uses several functions including hash, public/private key cryptography, digital signatures, peer-to-peer networks and proof of work. So in this paper we have developed an efficient and more secure way of storing birth certificate by using Inter Planetary File System (IPFS) and most demanded "blockchain" technology.*

Keywords--- Birth certificate, birth certificate verification, birth certificate authentication, blockchain technology, IPFS.

I. INTRODUCTION

Birth certificate is mostly in the form of a paper-based document as an electronic document cannot effectively replace a physical birth certificate [1]. However, due to low quality and cheap scanning and printing technologies available, the forgery of birth certificates has increased. This threatens the integrity of the birth certificate holders and the government bodies that issued the certificate. Therefore, birth record of an individual's validation and verification are the major challenge. It is necessary to validate that the birth certificate presented by any person is genuine and the holder is the rightful owner. Moreover, birth record has to be verified to ensure that its content is correct and also to ensure that the birth certificate has been issued from an authentic source [3].

With these motivations we have used blockchain technology for storing each individual birth record. This emerging technology is an asset database that aggregates transactions in blocks, and these blocks are appended to a chain of existing blocks. This is suitable for decentralized and transactional sharing of data across a large network of untrusted participants. By using this technology one can maintain continuously growing list of records called blocks and link them in a distributed manner (blockchain), potentially in such a way that these are secured against tampering. Blockchain is maintained as distributed database of records of transactions (Distributed Ledger) that are shared among participants. This technology uses cryptographic algorithms to validate the logged transactions and ensures that no record is duplicated and also permanent records are updated on each node of the network. This technology allows new type of distributed software architecture where components can establish trust by finding concurrence on their shared states.

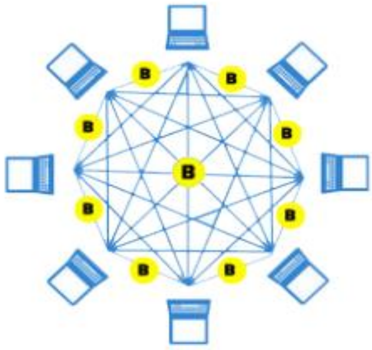


Fig. 1: Distributed Ledger

In recent years digitalization is on extreme demand where vital documents such as passports, birth records, medical records, even transactions are being digitalized. Digitalization has not just improved security but also time and effort to maintain all records has reduced. Despite of being secure several fraud incidents occur such as fake birth certificate rackets which was identified in Delhi, India few months back [2]. We require an efficient technology to store birth records which cannot be tampered as well as easy to maintain, well secured and easily shareable. In the literature, there already exists another way of securing data is using biometric techniques which are the most preferred techniques to establish identity of an individual. Biometrics are nothing but using our biological traits for authentication purposes which cannot be replaced by anyone as they are unique. These techniques are also used to replace passwords, pins, smartcards, keys and tokens which are the means of authentication. But disadvantages of using the pins or passwords are difficult to remember and also that there is a good chance that they might be hacked, corrupt our physical domains. There could be also possibility of smart cards, keys, tokens may be stolen by someone or we might misplace or forget them in an undesired place which can actually lead to many security issues.

Manuscript received February 01, 2019

Maharshi Shah, Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore, Tamil Nadu, Amrita Vishwa Vidyapeetham, India. (e-mail: cb.en.p2cse17015@cb.students.amrita.edu)

Priyanka Kumar, Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore, Tamil Nadu, Amrita Vishwa Vidyapeetham, India. (e-mail: k_priyanka@cb.amrita.edu)

Magnetic cards have a good chance of getting corrupted and make them unreadable. But using biometrics to secure birth certificate is not conventional nor feasible as fingerprints changes slightly over the time. So with these motivation we have proposed more efficient and secure way of storing birth certificates used blockchain technology and IPFS which can easily verified and authenticated [3].

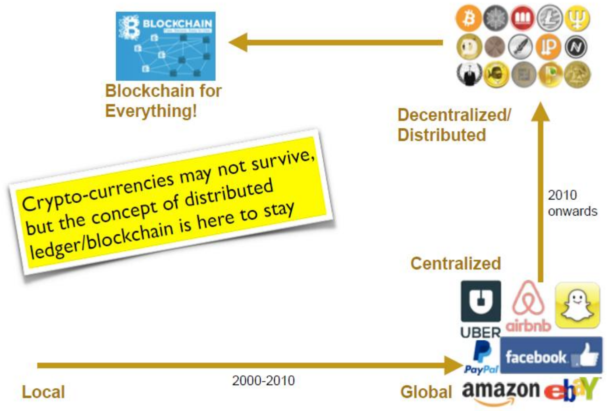

BLOCKCHAIN
Blockchain for Everything!
Decentralized/ Distributed
Crypto-currencies may not survive, but the concept of distributed ledger/blockchain is here to stay
2010 onwards
Centralized
UBER
airbnb
facebook
PayPal
Local
2000-2010
Global
amazon
ebY

Fig. 2: Ever Changing Landscape of Communication

Fig. 3: Blockchains can change a lot of things

Due to immutable and decentralized/distributed nature, blockchain technology has become so popular and it has changed a lot of things.

Roadmap. The paper is organized as follows. We describe our proposed system model in Section 2. In Section 3, we formally show the implementations of our proposed model. In Section 4 we formally describe the results obtained so far and user validation and successfully retrieval of birth record from blockchain by any government agency. Finally we conclude in Section 5.

II. SYSTEM ARCHITECTURE

Our Proposed theoretical architecture is depicted below.

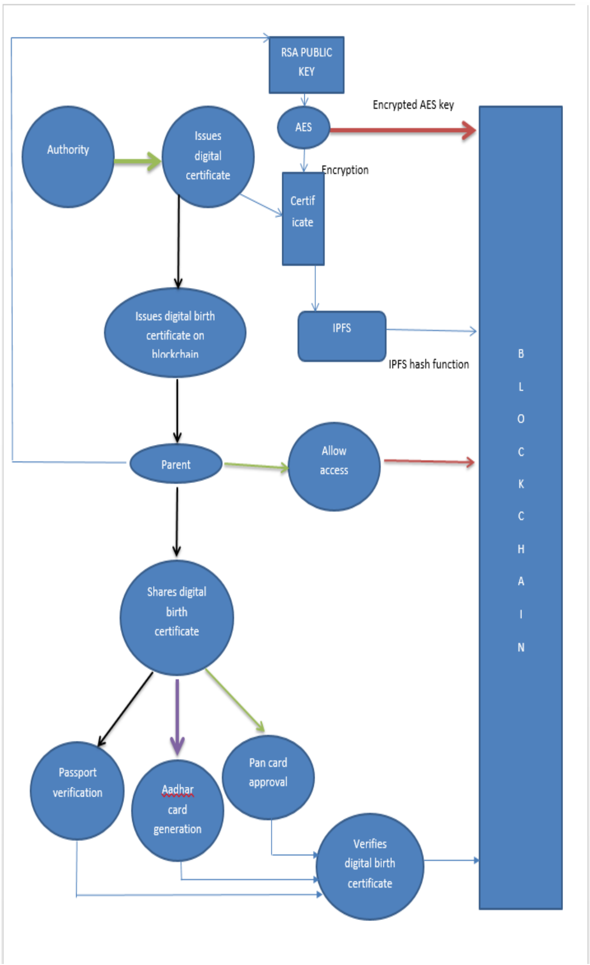


Fig. 4: Proposed Architecture

Blockchain technology is immutable as well as distributed technology where all the data is been replicated to all the participating clients in a blockchain. So if we want to tamper a record, we need to tamper data on each client to whom data has been distributed which is nearly impossible because of the availability of computing power. Hence the blockchain is seen as one of the most secure technology ever the world has witnessed. Integrating blockchain to secure birth records will not just only save paperwork but it will be more secure and all the fake birth certificate rackets would shut down, no more duplicate birth certificate would be valid. Also we can easily share Birth certificate on blockchain which can be used for verification purpose such as passport verification, Aadhar card verification, pan card verification etc. The proposed model consists of the following parts.

III. IMPLEMENTATION

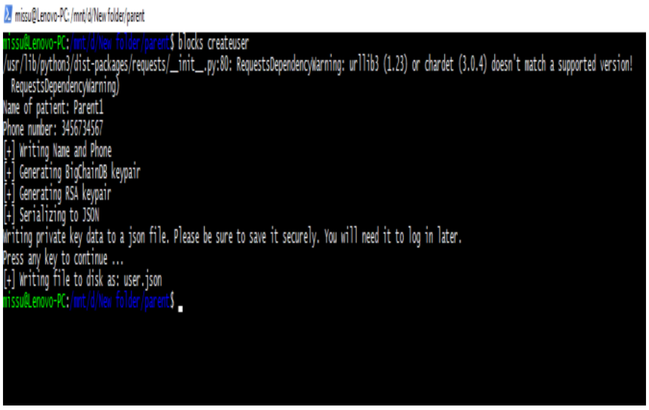
We have implemented the following modules.

A. User Registration

User will register on application using Name and Phone Number. So in this proposed system, parent has to register as user name and their phone number.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

We have used RSA algorithm for user login. During registration, RSA key pair as well as bigchain DB key pair will be generated and this will be stored in JSON format which is shown in the fig 5. For future perspective it can be embedded in RFID chip for quick access.

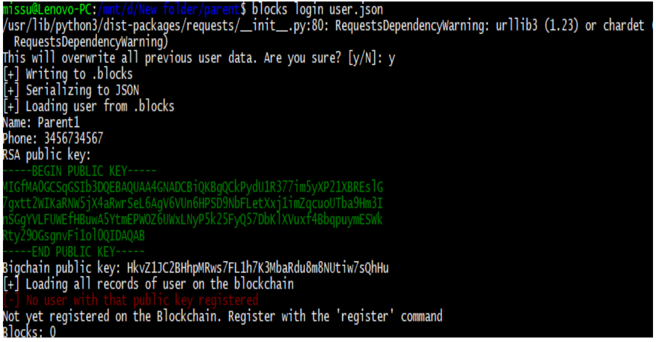


```
missu@Lenovo-PC:/mnt/d/New folder/parent

missu@Lenovo-PC:/mnt/d/New folder/parent$ blocks createuser
/usr/lib/python3/dist-packages/requests/__init__.py:80: RequestsDependencyWarning: urllib3 (1.23) or chardet (3.0.4) doesn't match a supported version!
  RequestsDependencyWarning)
Name of patient: Parent1
Phone number: 3456734567
[+] Writing Name and Phone
[+] Generating BigChainDB keypair
[+] Generating RSA keypair
[+] Serializing to JSON
Writing private key data to a json file. Please be sure to save it securely. You will need it to log in later.
Press any key to continue ...
[+] Writing file to disk as: user.json
missu@Lenovo-PC:/mnt/d/New folder/parent$
```

Fig.5: User registration form

B. *User login*

This module will help user to access his account. While login data would be taken from JSON file which was stored in Computer during login.

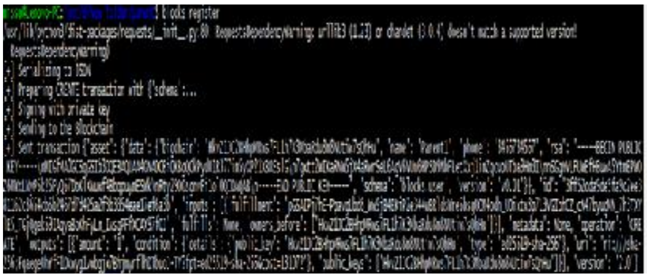


```
missu@Lenovo-PC:/mnt/d/New folder/parent$ blocks login user.json
/usr/lib/python3/dist-packages/requests/__init__.py:80: RequestsDependencyWarning: urllib3 (1.23) or chardet
  RequestsDependencyWarning)
This will overwrite all previous user data. Are you sure? [y/N]: y
[+] Writing to .blocks
[+] Serializing to JSON
[+] Loading user from .blocks
Name: Parent1
Phone: 3456734567
RSA public key:
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCkPydU1R377im5yXP21XBREslG
7gxtt2WIKaRNW5jX4aRwrSeL6AgV6VUn6HPSD9NbFLetXxj1imZqcuoUTba9Hm3I
nSGgYVLFUWEfHBuwA5YtmEPWOZ6UWxLNyP5k25FyQ57DbKlXVuxf4BbqpuymESWk
Rty29OGsgnvFi1ol0QIDAQAB
-----END PUBLIC KEY-----
Bigchain public key: HkvZ1JC2BHhpMRws7FL1h7K3MbaRdu8m8NUtiw7sQhHu
[+] Loading all records of user on the blockchain
[-] No user with that public key registered
Not yet registered on the Blockchain. Register with the 'register' command
Blocks: 0
```

Fig.6: User login form

C. *User Registration on blockchain*

This module will help user to register with blockchain, all the information such as RSA public key, Bigchain DB public key, Name, Phone Number would be registered on blockchain, so it will be easier to find any user on blockchain.



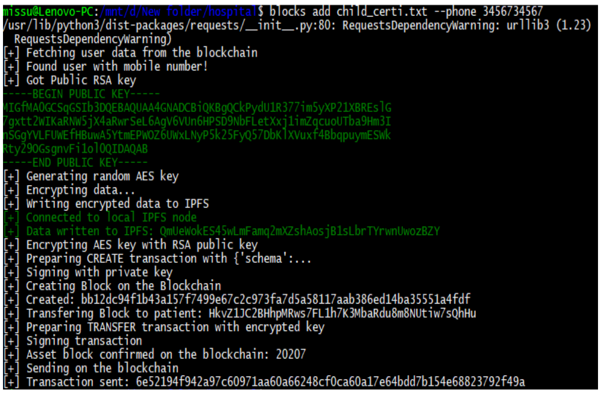
```
missu@Lenovo-PC:/mnt/d/New folder/parent$ blocks register
/usr/lib/python3/dist-packages/requests/__init__.py:80: RequestsDependencyWarning: urllib3 (1.23) or chardet (3.0.4) doesn't match a supported version!
  RequestsDependencyWarning)
[+] Serializing to JSON
[+] Preparing CREATE transaction with {'schema':...
[+] Signing with private key
[+] Sending to the Blockchain
[+] Sent transaction {'asset': {'data': {'bigchain': '[illegible]', 'name': 'Parent1', 'phone': '3456734567', 'rsa': '-----BEGIN PUBLIC
KEY-----[illegible]
[illegible]-----END PUBLIC KEY-----', 'schema': 'blocks.user', 'version': 'v0.1'}}, 'id': '[illegible]
[illegible]', 'inputs': [{'fulfillment': '[illegible]
[illegible]', 'fulfills': None, 'owners_before': ['[illegible]']}], 'metadata': None, 'operation': 'CRE
ATE', 'outputs': [{'amount': '1', 'condition': {'details': {'public_key': '[illegible]', 'type': 'ed25519-sha-256'}, 'uri': 'ni:///sha-
256;[illegible]'}, 'public_keys': ['[illegible]']}], 'version': '2.0'}
```

Fig.7: User registration on blockchain

D. *Adding file to blockchain*

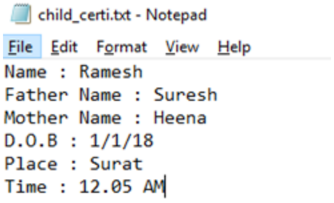
Whenever a birth certificate governing body wants to add certificate, it will find users phone number from blockchain. When the client's phone number is used, it will generate one AES key which will encrypt the content of file and on top of this, AES key will be encrypted using clients RSA public key, so that only dedicated user with RSA private key can decrypt.

In meantime transaction will sent to IPFS and one hash function will be generated, which will be stored inside block of Blockchain.



```
missu@Lenovo-PC:/mnt/d/New folder/hospital$ blocks add child_certi.txt --phone 3456734567
/usr/lib/python3/dist-packages/requests/__init__.py:80: RequestsDependencyWarning: urllib3 (1.23)
  RequestsDependencyWarning)
[+] Fetching user data from the blockchain
[+] Found user with mobile number!
[+] Got Public RSA key
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCkPydU1R377im5yXP21XBREslG
7gxtt2WIKaRNW5jX4aRwrSeL6AgV6VUn6HPSD9NbFLetXxj1imZqcuoUTba9Hm3I
nSGgYVLFUWEfHBuwA5YtmEPWOZ6UWxLNyP5k25FyQ57DbKlXVuxf4BbqpuymESWk
Rty29OGsgnvFi1ol0QIDAQAB
-----END PUBLIC KEY-----
[+] Generating random AES key
[+] Encrypting data...
[+] Writing encrypted data to IPFS
[+] Connected to local IPFS node
[+] Data written to IPFS: QmUeWokES45wLmFamq2mXZshAosjB1sLbrTYrwnUwozBZY
[+] Encrypting AES key with RSA public key
[+] Preparing CREATE transaction with {'schema':...
[+] Signing with private key
[+] Creating Block on the Blockchain
[+] Created: bb12dc94f1b43a157f7499e67c2c973fa7d5a58117aab386ed14ba35551a4fdf
[+] Transfering Block to patient: HkvZ1JC2BHhpMRws7FL1h7K3MbaRdu8m8NUtiw7sQhHu
[+] Preparing TRANSFER transaction with encrypted key
[+] Signing transaction
[+] Asset block confirmed on the blockchain: 20207
[+] Sending on the blockchain
[+] Transaction sent: 6e52194f942a97c60971aa60a66248cf0ca60a17e64bdd7b154e68823792f49a
```

Fig.8: Hospital issuing birth certificate for a user



```
Name : Ramesh
Father Name : Suresh
Mother Name : Heena
D.O.B : 1/1/18
Place : Surat
Time : 12.05 AM
```

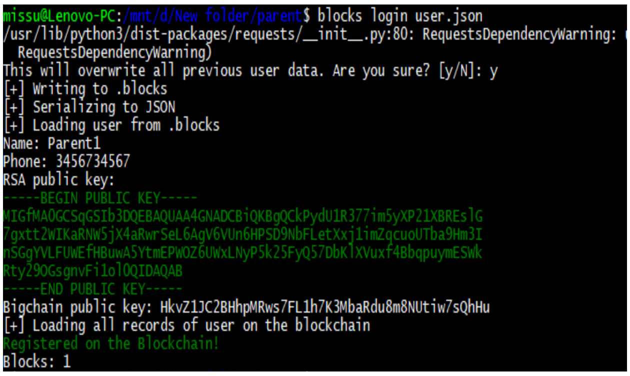
Fig.9: Birth certificate

IV. RESULTS OBTAINED SO FAR

We have implemented our proposed model and found the following results.

E. *Retrieve file from blockchain*

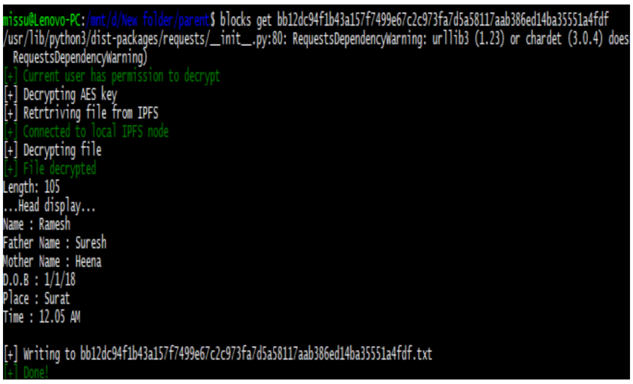
When user wants to retrieve the file, the RSA private key will decrypt the AES key, only the user whose RSA public key was used for encryption can decrypt it using its RSA private key. After obtaining AES key, key will be applied on the file to decrypt data. On successful decryption user will able to see actual content of file. During this process, the IPFS hash function stored inside block will be mapped to IPFS server and according to it file will be downloaded in the system. On successful download of file from IPFS, system will automatically apply RSA private key from the JSON file to decrypt the AES key. Once we obtain AES key, we need to apply this AES key on the file which was used to encrypt data. And on successful decryption we will get the content of file.



```
missu@Lenovo-PC:/mnt/d/New folder/parent$ blocks login user.json
/usr/lib/python3/dist-packages/requests/__init__.py:80: RequestsDependencyWarning:
  RequestsDependencyWarning)
This will overwrite all previous user data. Are you sure? [y/N]: y
[+] Writing to .blocks
[+] Serializing to JSON
[+] Loading user from .blocks
Name: Parent1
Phone: 3456734567
RSA public key:
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCkPydU1R377im5yXP21XBREslG
7gxtt2WIKaRNW5jX4aRwrSeL6AgV6VUn6HPSD9NbFLetXxj1imZqcuoUTba9Hm3I
nSGgYVLFUWEfHBuwA5YtmEPWOZ6UWxLNyP5k25FyQ57DbKlXVuxf4BbqpuymESWk
Rty29OGsgnvFi1ol0QIDAQAB
-----END PUBLIC KEY-----
Bigchain public key: HkvZ1JC2BHhpMRws7FL1h7K3MbaRdu8m8NUtiw7sQhHu
[+] Loading all records of user on the blockchain
Registered on the Blockchain!
Blocks: 1
```

Fig.10: Updated increment of block pointer on user login

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication



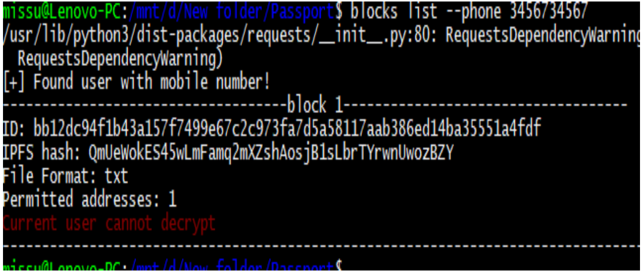
```
missu@Lenovo-PC:/mnt/d/New folder/parent$ blocks get bb12dc94f1b43a157f7499e67c2c973fa7d5a58117aab386ed14ba35551a4fdf
/usr/lib/python3/dist-packages/requests/__init__.py:80: RequestsDependencyWarning: urllib3 (1.23) or chardet (3.0.4) does
  RequestsDependencyWarning)
[+] Current user has permission to decrypt
[+] Decrypting AES key
[+] Retrtriving file from IPFS
[+] Connected to local IPFS node
[+] Decrypting file
[+] File decrypted
Length: 105
...Head display...
Name : Ramesh
Father Name : Suresh
Mother Name : Heena
D.O.B : 1/1/18
Place : Surat
Time : 12.05 AM

[+] Writing to bb12dc94f1b43a157f7499e67c2c973fa7d5a58117aab386ed14ba35551a4fdf.txt
[+] Done!
```

Fig.11: User retrieving birth certificate

F. Permit other user to view file

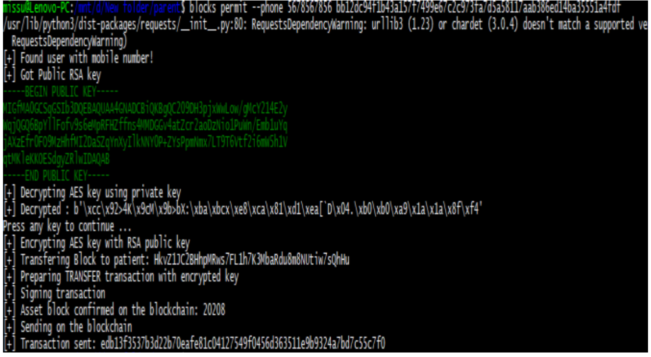
If user wants to share file with other user such as passport authority for birth certificate verification he can simply add file with permit flag and phone number of the user whom we can easily find on blockchain.

In this module all the above steps will be repeated but instead of users RSA key, the third party whom we want to give access will be used to encrypt AES key.



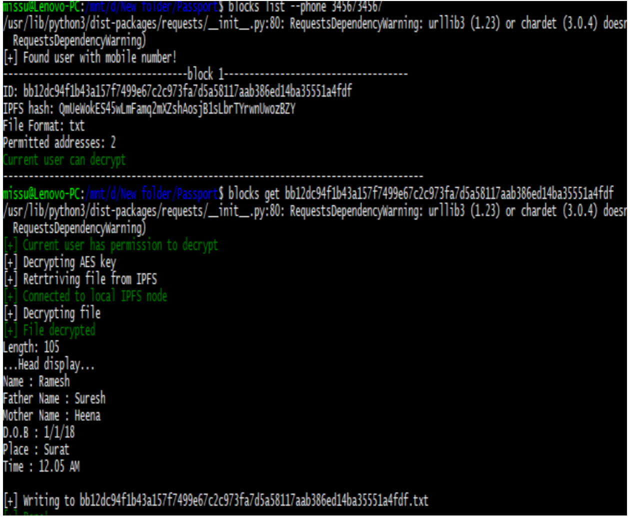
```
missu@Lenovo-PC:/mnt/d/New folder/Passport$ blocks list --phone 3456734567
/usr/lib/python3/dist-packages/requests/__init__.py:80: RequestsDependencyWarning
  RequestsDependencyWarning)
[+] Found user with mobile number!
----------------------------------block 1----------------------------------
ID: bb12dc94f1b43a157f7499e67c2c973fa7d5a58117aab386ed14ba35551a4fdf
IPFS hash: QmUeWokES45wLmFamq2mXZshAosjB1sLbrTYrwnUwozBZY
File Format: txt
Permitted addresses: 1
Current user cannot decrypt
---------------------------------------------------------------------------
missu@Lenovo-PC:/mnt/d/New folder/Passport$
```

Fig.12: Permission required for decrypting file.



```
missu@Lenovo-PC:/mnt/d/New folder/parent$ blocks permit --phone 5678567856 bb12dc94f1b43a157f7499e67c2c973fa7d5a58117aab386ed14ba35551a4fdf
/usr/lib/python3/dist-packages/requests/__init__.py:80: RequestsDependencyWarning: urllib3 (1.23) or chardet (3.0.4) doesn't match a supported ve
  RequestsDependencyWarning)
[+] Found user with mobile number!
[+] Got Public RSA key
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC209DH3pjxWwLow/gMcY214E2y
WqjQGQ6BpYllFofv9s6eMpRFHZffns4MMDGGv4atZcr2aoDzNio1PuWn/Emb1uYq
jAXzEfrOFO9MzHhfMI2DaSZqYnXyIlkNNYOP+ZYsPpmNmx7LT9T6Vtf2i6mWSh1V
qtMKleKKOESdgyZRlwIDAQAB
-----END PUBLIC KEY-----
[+] Decrypting AES key using private key
[+] Decrypted : b'\xcc\x92>4K\x9cM\x9b>bX:\xba\xbcx\xe8\xca\x81\xd1\xea[`D\x04.\xb0\xb0\xa9\x1a\x1a\x8f\xf4'
Press any key to continue ...
[+] Encrypting AES key with RSA public key
[+] Transfering Block to patient: HkvZ1JC2BHhpMRws7FL1h7K3MbaRdu8m8NUtiw7sQhHu
[+] Preparing TRANSFER transaction with encrypted key
[+] Signing transaction
[+] Asset block confirmed on the blockchain: 20208
[+] Sending on the blockchain
[+] Transaction sent: edb13f3537b3d22b70eafe81c04127549f0456d363511e9b9324a7bd7c55c7f0
```

Fig.13: User giving permission to passport authority



```
missu@Lenovo-PC:/mnt/d/New folder/Passport$ blocks list --phone 3456734567
/usr/lib/python3/dist-packages/requests/__init__.py:80: RequestsDependencyWarning: urllib3 (1.23) or chardet (3.0.4) doesn
  RequestsDependencyWarning)
[+] Found user with mobile number!
----------------------------------block 1----------------------------------
ID: bb12dc94f1b43a157f7499e67c2c973fa7d5a58117aab386ed14ba35551a4fdf
IPFS hash: QmUeWokES45wLmFamq2mXZshAosjB1sLbrTYrwnUwozBZY
File Format: txt
Permitted addresses: 2
Current user can decrypt
---------------------------------------------------------------------------
missu@Lenovo-PC:/mnt/d/New folder/Passport$ blocks get bb12dc94f1b43a157f7499e67c2c973fa7d5a58117aab386ed14ba35551a4fdf
/usr/lib/python3/dist-packages/requests/__init__.py:80: RequestsDependencyWarning: urllib3 (1.23) or chardet (3.0.4) doesn
  RequestsDependencyWarning)
[+] Current user has permission to decrypt
[+] Decrypting AES key
[+] Retrtriving file from IPFS
[+] Connected to local IPFS node
[+] Decrypting file
[+] File decrypted
Length: 105
...Head display...
Name : Ramesh
Father Name : Suresh
Mother Name : Heena
D.O.B : 1/1/18
Place : Surat
Time : 12.05 AM

[+] Writing to bb12dc94f1b43a157f7499e67c2c973fa7d5a58117aab386ed14ba35551a4fdf.txt
```

Fig.14: Passport office can view birth certificate after user gave permission.

V. CONCLUSION

There are many methods for securing birth records by using biometrics, cryptography, combination of both cryptography and biometrics etc. But we are using blockchain technology along with Cryptography algorithms and Inter Planetary File System (IPFS) protocol to secure birth records, as well as to access and share records with user permission from anywhere in the network. It also validates or authenticates records within a few moments. This work is still in progress and results will be published soon. In future, we want to extend our work to embed JSON file in RFID chip for easily access of birth records.

REFERENCES

1. M. Warasart and P. Kuacharoen, “Paper-based Document Authentication using Digital Signature and QR Code,” ICCET, 2012.
2. Nicolas Buchmann, Christian Rathgeb, Harald Baier, Christoph Busch and Marian M: Enhancing Breeder Document Long-Term Security using Blockchain Technology, IEEE 41st Annual Computer Software and Applications Conference, 2017.
3. Yongle Chen, Hui Li, Kejiao Li and Jiyang Zhang : An improved P2P File System Scheme based on IPFS and Blockchain, IEEE International Conference on Big Data (BIGDATA), 2017.
4. Sin Kuang Lo, Xiwei Xu, Yin Kia Chiam, Qinghua Lu : Evaluating Suitability of Applying Blockchain , International Conference on Engineering of Complex Computer Systems , 2017.
5. Safdar Hussain Shaheen, Muhammad Yousaf, Mudassar Jalil : Temper Proof Data Distribution for Universal Verifiability and Accuracy in Electoral Process Using Blockchain, IEEE Conference, 2017.
6. T.D. Smith : The Blockchain Litmus Test, IEEE International Conference on Big Data (BIGDATA), 2017.
7. http://startupmanagement.org/blog
8. H. Hou, “The application of blockchain technology in E-government in China,” *2017 26th Int. Conf. Comput. Commun. Networks, ICCCN 2017*, 2017.
9. J. Sidhu, “Syscoin : A Peer-to-Peer Electronic Cash System with Blockchain-Based Services for E-Business,” 2008.
10. N. Smolenski and D. Hughes, “Academic Credentials In An Era Of Digital Decentralization Academic Credentials In An Era Of Digital Decentralization Learning Machine Cultural Anthropologist contents preface,” 2016.
11. S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” *Www.Bitcoin.Org*, p. 9, 2008.
12. Nomura Research Institute, “Survey on Blockchain Technologies and Related Services,” 2016.
13. S. Thompson, “The preservation of digital signatures on the blockchain - Thompson - See Also,” *Univ. Br. Columbia iSchool Student J.*, vol. 3, no. Spring, 2017.
14. C. F. Bond, F. Amati, and G. Blousson, “Blockchain, academic verification use case,” 2015.
15. J.-F. Blanchette, “The digital signature dilemma Le dilemme de la signature numérique,” *Ann. Des Télécommunications*, vol. 61, no. 7, pp. 908–923, 2006.
16. MIT Media Lab, “What we learned from designing an academic certificates system on the blockchain,” *Medium*, no. December, p. 2016, 2016.
17. P. Schmidt, “Certificates, Reputation, and the Blockchain,” *MIT Media Lab*, 2015.