Blockchain-based Interoperable Healthcare Using Zero-knowledge Proofs and Proxy Re-Encryption

Bhavye Sharma
MAIT Rohini, India
bhavye.sharma1996@gmail.com

Raju Halder
IIT Patna, India
halder@iitp.ac.in

Jawar Singh
IIT Patna, India
jawar@iitp.ac.in

***Abstract*—The development of a robust, transparent and interoperable E-healthcare infrastructure has been a difficult task due to many regulations and legislatures like HIPAA (Health Insurance Portability and Accountability Act) and GDPR (General Data Protection Regulation). Healthcare service providers prefer to store data about their patients in locked up silos, behind often inadequate layers of security and firewalls. Such an approach results in data breaches and limits the ability to get a holistic view of the medical history of a patient. The obscure cost of treatment is another issue brought to attention recently in media. In this paper, we have proposed a national blockchain framework for managing patients' Electronic Health Records (EHRs) access control and funds in the context of India's National healthcare scheme. We introduce a transparent insurance claim process for healthcare providers and an auditable trail of EHR access using smart contracts. We use a smart card approach allowing beneficiaries to authenticate their identity using zero-knowledge proofs and delegate access to service the providers via proxy re-encryption.**

***Index Terms*—Blockchain, Healthcare, Proxy Re-Encryptionn, ZK-Snarks, Zero Knowledge Proofs**

I. Introduction

Recently, there has been a growing interest around the world to improve healthcare by adopting emerging technologies. These include introduction of Electronic Health Records (EHRs), cloud based storage of medical data, Internet of Things (IoT) for medical devices to monitor the vital signs of patients, data mining to find trends and monitor epidemic outbreaks. Adoption of this "e-healthcare" approach has allowed patients to store their medical records on their smart phones, send real time updates to their primary health care provider through health monitors and get medicines prescribed. These innovations to collect and store data about patients also present new challenges to abide by regulations like HIPAA [1] and GDPR [2], that ensure privacy and security of medical data.

Health insurance claims are another problem that is vexing the healthcare industry. Due to its nature of dealing with monetary value, it is time consuming and prone to fraud. In the USA, third party vendors known as "Clearing Houses" deal with the process of validating insurance claims and ensuring compatibility of data between the insurance company and the hospital. This sharing of medical data to third party vendors often is the source of data leaks if done inappropriately. In India, on the other hand, around a billion have no health coverage for whatsoever reasons according to a study by Ernst and Young [3].

Recently, the Indian Government has launched the world's largest national health insurance scheme i.e. `Aayushman Bharat Yojna` to provide a health coverage of INR 5,00,000 ($6,900) to every family below poverty line as per Socio Econonmic Caste Consensus 2011 [4]. Since, this scheme will be claimable at public as well as private empanelled hospitals it is necessary to enforce a national pricing mechanism for the 1350 medical treatments covered under this scheme.

Blockchain-based solutions are emerging to be a viable option for overcoming the flaws in current healthcare systems by allowing secure data sharing and controlled access between healthcare providers and insurance companies [5], [6]. Research has also been done to allow privacy assured health insurance claims using blockchain technology [7]. A blockchain acts as a distributed and immutable ledger between the stakeholders like Hospitals, Patients, Pharmacies and Government agencies. Its immutability allows for an auditable trail of transactions that are cryptographically linked and transparent. A recent study by IBM [8] showed that 56% of healthcare institutes plan to adopt blockchain technology for EHR sharing. Privacy is a major feature that needs to be built by design into a blockchain architecture for it to be adopted, especially in the case of healthcare data which is one of the most sensitive kinds of personal information that can be exploited by malicious parties and sold online in the black market. Hence, it should not be leaked by any entity (e.g. health service provider, an insurance company, and even the government agencies), otherwise, it would leave them liable to lawsuits and expensive fines by HIPAA.

In this paper, we propose a system that allows automated insurance claim, interoperability and data sharing between different healthcare providers in context of India's new health care Scheme i.e. `Aayushman Bharat Yojna`. This is a challenging task, as more than just creating a unified data schema to be used by everyone, it also needs to manage funds, access control and permissions in a transparent and non-repudiable way. While protecting integrity and privacy of the medical data according to regulations [1], [2].

A. This paper makes the following contributions

- We propose a novel national healthcare framework using Inter Planetary File System (IPFS) and smart contracts for storage and access control of EHRs and other medical documents. Our solution use(s) a proxy cloud server for managing re-encryption to delegate access to the service providers to read and write data on the patient's behalf.
- We introduce a novel authentication mechanism using zero-knowledge proofs of identity from [9] to verify the identity of a patient with his nation's digital identity (e.g. Aadhar in India's context) using Zero-Knowledge Succinct Non-Interactive arguments of knowledge (ZkSnarks) [10] and E-cards. A six-digit pin code or passphrase is used along with a one-way hashing function to authenticate access to medical data from Blockchain Smart contracts.

- We perform a security analysis of our proposed solution and defined it's trust model.

II. Related Works

There has been extensive work for solutions to deal with the problem of Interoperable access control of EHR sharing and Insurance claims individually. Recently, Nguyen *et al.*[11] proposed a cloud-based solution for secure EHR sharing. The flaw in their approach was the presence of a centralized "EHR manager" responsible for decryption and re-encryption of EHR data. This would leave the data revealed as plain text on the cloud, making it a prime target for hackers and malicious parties. Another reason why it would be unsuitable for the Indian context is the tech-illiterate people for whom the national health scheme is made will not be able to manage privileges using smartphone applications.

Another proposed solution for EHR sharing in the context of health governance in Saudi-Arabia [12] gives the responsibility of securely sharing decryption keys of EHRs to a central Trusted Health Authority (THA). If this authority were to be hacked all the decryption keys for EHR stored on the cloud will be compromised. It also used Attribute-Based encryption which suffers from expensive computation and complexity in bi-linear pairing operations as observed in [13].

In [14], the blockchain technology has been utilized as a distributed data storage of medical data. The limitation caused by this approach is of scalability since all the nodes in the network have to replicate the data on blockchain, it would result in a bloated blockchain. Similarly, Adler *et al.* [15] describes the benefits of a paperless EHR system over the traditional paper-based system. It also studied the challenges faced in the adoption of a new IT healthcare system and also concluded a reform was needed in the payment system to promote better healthcare. For authentication using smart cards and zero-knowledge proof, we were inspired by [9] which utilized e-passport based zero-knowledge proofs to create pseudo-anonymous identities for Sybil resistant mining of blockchain. This solution fits well for our proof of identity for healthcare as explained later.

III. Preliminaries

A. Blockchain

Post advent of the Bitcoin by Satoshi Nakamoto in 2008 [16], the Blockchain Technology has emerged as an important innovation showing its immense potential applications in many sectors, including finance, e-governance, supply chain, IoT security, healthcare, and many more. Further, it also allows a peer to peer transaction system without the need for intermediary parties like Banks. It involves the creation of an immutable, distributed, replicated ledger for storage and validation of transactions between pseudo-anonymous identities or "accounts". Such a distributed ledger provides a powerful means to verify records, without any trusted intermediary such as brokers, agents, etc. By utilizing the consensus protocol between multiple nodes it allowed performing computations in a distributed manner. Smart Contracts which allowed developers to build applications by using Turing complete programming languages such as "`solidity`" and code automated programs containing business logic. These smart contracts enable the development of trusted open-source software in the form of DApps and new Blockchains.Efforts from industries as well as the government in the development of proofs-of-concept are moving to the deployment stage. For example, cryptocurrencies linked to fiat money are being launched as 'Stablecoins' world-over.

B. Proxy Re-encryption

Proxy re-encryption is a type of public key encryption that allows an intermediary (proxy) to transform cipher text from one public key to another without being able to observe the plain data. This technique was first realised in [17] and was based on traditional ElGamal scheme. A comprehensive study on multiple variations of proxy re-encryption have been researched, for example, ID-based proxy re-encryption, uni-directional proxy re-encryption, fully homomorphic proxy re-encryption [18].

Definition 1: Any proxy re-encryption scheme can be described by 5 algorithms (KeyGen, ReKeyGen, ReEnc, Enc, and Dec) as follows:

1) *KeyGen*(n): This generates a random public-private key pair. In case of Identity based encryption, we also pass the identity and system parameters as arguments.
2) *ReKeyGen*($\mathsf{pk}_A, \mathsf{sk}_A, \mathsf{pk}_B, \mathsf{sk}_B^*$): This generates a re-encryption key $rencK_{A->B}$ that can be used by proxy to convert cipher text intended for A to cipher text for B. Where, *denotes that private key of B may [17] or may not [19] be required depending on algorithm.
3) *ReEnc*($rencK_{A->B}, C_A$): This is executed by the proxy to convert cipher text C_A to cipher text C_B which can be decrypted by sk_B.
4) *Enc*($data, \mathsf{pk}_A$): This generates the cipher text C_A that can be decrypted by sk_A.
5) *Dec*(C_A, sk_A): This decrypts the encrypted cipher text to reveal the data.

Salient characteristics that distinguish different proxy re-encryption schemes are: Directionality, Transitivity, Interactivity, and Collusion-Resistance.

C. Zero Knowledge proofs

Zero knowledge proofs were first conceived by Goldwasser *et al.* in [20], the question they were asking was "how much knowledge needs to be conveyed to show proof of some knowledge?". A zero knowledge protocol is a probabilistic-based verification method that involves two parties, namely a prover and a verifier, the prover is said to have exponential time-space while the verifier having linear time-space. The objective for the prover to prove knowledge of a witness, W, to the verifier without actually revealing the witness itself. Requirements that need to be satisfied by any zero knowledge proof are:

1) **Completeness**: This property states that as long as both prover and verifier are behaving honestly the zero-knowledge proof will work for "true proofs".
2) **Soundness**: This property states that the prover shall not be able to generate "false proofs" without knowledge of the witness.
3) **Zero-knowledge**: No knowledge about prover is revealed to the verifier other than the knowledge of proof being true or false.

The zero knowledge proof that we will be using in our solution is "Zero-Knowledge Succinct Non-Interactive Argument of

TABLE I: Smart Contracts in proposed solution

Contract	Symbol	Functionality
Registration	RSC	Stores mapping of empanelled hospitals and Beneficiaries
Insurance	ISC	Stores remaining balance, EHR hash, PRE Keys and logs claims for audit
Verifier	VSC	Automatically generated by Zokrates to check proofs on chain

Knowledge" (ZK-snark), which involves a multiparty trusted setup phase before generation and verification of proofs. Zokrates [21] library provides a high level domain specific language similar to Python for the creation of ZK-snarks for the Ethereum Blockchain. It takes care of generation of the Common Reference String in the trusted setup phase and the disposal of the private randomness used in generation of it.

D. PM-JAY scheme

Ayushman Bharat Yojana (PM-JAY) [22] is a national healthcare scheme launched by the Indian Government to provide health insurance coverage to 100 million poorest families in India. Benefits of the scheme are portable across the country and a beneficiary covered under the scheme will be allowed to take cashless benefits from any public or private empanelled hospitals across the country. The government of India has provided a fund of 2,000 Crore rupees and will cover for 5 Lakh rupees for each family entitled to the scheme. The objective of this scheme is universal access to good quality health care services without anyone having to face financial hardship as a consequence. A predefined list of 1,350 medical procedures are covered by this scheme and require no out-of-pocket payment from the beneficiary.

IV. Proposed System

The current flaws in the healthcare system can be overcome by utilizing the transparency, immutability and automation features of blockchain technology. We propose a layered smart contract approach to manage access control permissions, insurance claims, and proof of identity for medical data in the context of India's National Health Scheme. A brief description of the smart contracts in the proposal is shown in Table I.

A. Stakeholders

Since our objective is to promote interoperability between the different healthcare institutes, government agencies and beneficiaries, let us first list out all the stakeholders involved in our proposed solution.

- **Federal Government**: The responsibility of providing funding for insurance under the PM-JAY scheme is of the federal government. They will be responsible for deployment of the ISC and will also have the capability audit the insurance claims made by utilizing event logging and state variables of the ISC.
- **Empanelled Hospitals**: These are hospitals and healthcare service providers (Public and Private) on-board onto the national healthcare scheme. They will be able to request access to EHR's of the beneficiaries while also being able to generate new EHR's and claim insurance amount after being registered onto the RSC by the government.
- **Beneficiaries**: The 10 crore families and 500 million individuals under the poverty line as per Socio Economic and Caste Consensus 2011 will be eligible to claim the benefits under the PM-JAY scheme. They will be given smart cards to store their public and private key generated at the registration desk of any of the aforementioned empanelled hospitals. Their public key will be registered via RSC.

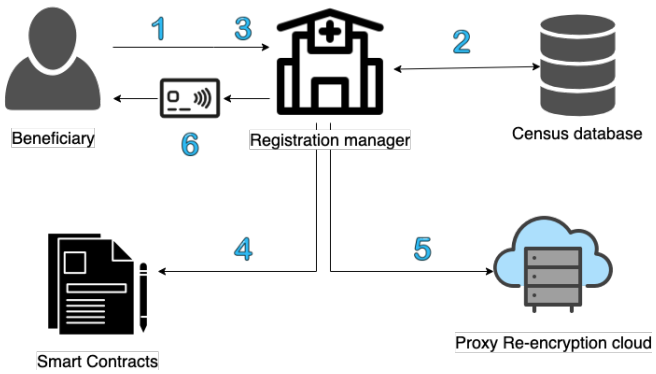


Fig. 1: The flow of beneficiary registration and key generation.

B. Registration of beneficiaries

The beneficiary families eligible for insurance coverage under PM-JAY scheme have to be issued unique digital Health IDs as proposed by NITI Aayog [22] in the form of public and private key. Their public key will be used to identify them on the blockchain. This registration process can be done at any hospital help desk that is included under the scheme. Also, in order to be able to employ a scalable proxy re-encryption framework for access control, a symmetric encryption key (SYM_K) (using 256 bit scheme like AES 256 or RSA) will need to be generated and should be stored in Proxy Re-encryption cloud after encrypting the SYM_K with public key of beneficiary to create C_{SYMK} (Encrypted cipher text of the unique symmetric key): $C_{SYMK} \leftarrow \mathsf{Enc}(\mathsf{pk}, SYM_K)$. This cipher text is stored in cloud and is re-encrypted by the cloud whenever a new empanelled hospital provides a valid proof to the VSC. We employ cloud due to it's high availability, scalability and overcome the privacy limitation of other solutions by never storing decrypted data on it. The work flow of Registration and key generation is illustrated in Fig. 1. The description of each registration step is summarized as follows:

1) The beneficiary uses biometric verification for identification of the individual against his/her national identity or state issued identity such as UIDAI Aadhar, family card to prove identity at registration desk.
2) The registration manager queries the latest SECC (Socio Economic Caste Consensus) database to check eligibility for coverage under PM-JAY after verifying the identity of beneficiary.
3) The beneficiary decides a six digits pin (any number of his/her choice) which will act as passphrase for authentication. This six digit pin along with hash of their national ID are used to generate a public private key pair for the beneficiary. An E-card with a private key will be issued to the beneficiary. This process needs to be deterministic to ensure only one unique key pair may be generated for each beneficiary and it needs to be recovered in case E-card is lost.

$$\mathsf{pk}, \mathsf{sk} = \mathsf{KGen}(KDF(pin, hash(NationalID)))$$

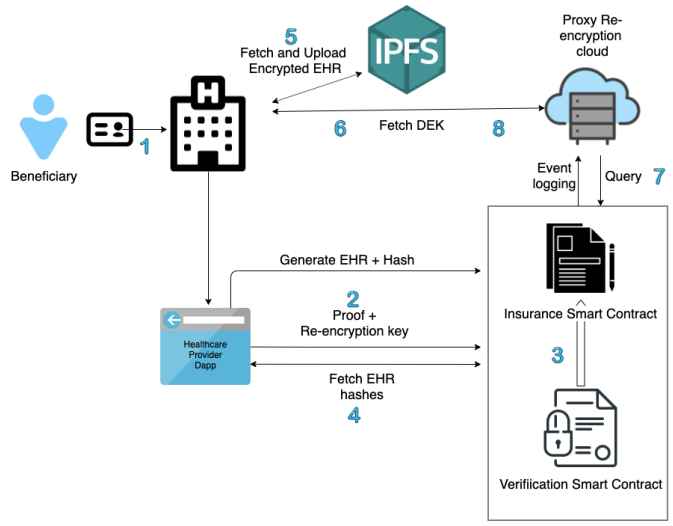


Fig. 2: Flow digram for EHR access and generation to verify insurance claim.

The Password-Based Key Derivation Function 2 (PBKDF2) is a password based key derivation function (KDF) recommended by IETF (Internet Engineering Task Force).

4) The registration manager will register this pkas a beneficiary in the RSC, setting it's balance as the insurance coverage (5 Lakh) in ISC and causing an event to be triggered. It will also store a *passwordHash* along with his public key which will act as authentication mechanism, where *PasswordHash* = H(sk||*pin*). This PasswordHash is stored in the VSC and will come into play when we use zero knowledge proofs for authentication later.
5) The registration manager will generate an unique symmetric key (SYM_K) using entropy in the form of public key of beneficiary, e.g. salt, passphrase etc. and encrypt it using beneficiary's pkbefore uploading to Proxy Re-encryption cloud server: $C_{SYMK} \leftarrow$ Enc (pk, SYM_K), and upload encrypted cipher text to cloud.
6) A smart card will be issued to the beneficiary with his newly generated public/private key pair stored on board.

This smart card will allow beneficiaries to access health services at any of the 15,000+ empanelled hospitals across the country with full coverage and the ability to share medical EHR's generated during the process.

C. Zk-Snark Trusted Setup

We are using ZK-snarks for our access control layer, where the Provers are the Hospitals and the Verifier is the publicly visible Verifier Smart Contract (VSC). The hospital will create a Proof of Private Key Ownership for the beneficiary and knowledge of pre-image of a hash. The pre-image in our case will be —Private key of beneficiary + Pin code passphrase— used by beneficiary. The hash of this value will be stored on the smart contract when registering beneficiary as explained above. VSC's function 'VerifyTX' checks if the witness computed and submitted by the hospital is correct and if so, enables EHR sharing via events logging.

For ZK-snarks to be feasible an initial trusted setup phase is required to generate the common reference string that will be used by everyone to generate proofs and by the VSC to verify said proofs. Zcash has a detailed explanation of how they performed this parameter generation using six geographically separated computers [23].

D. Insurance claim and data sharing

In this Subsection, we introduce two functionalities (a) secure data sharing and access control using proxy re-encryption, and (b) automated insurance claims using smart contracts on blockchain. The smart contracts allow different functions to be executed in a distributed and verifiable manner, providing an audit trail of events and transactions. The flow diagram in Fig. 2 explains how our approach allows for a more transparent and convenient method, explained as follows.

EHR access sharing

1) The beneficiary will visit one of 15,000+ empanelled hospitals and present the smart card issued to him along with the six digit pin code decided by him/her during registration.
2) The hospital uses the pk present on the smart card to generate re-encryption key $rencK_{beneficiary->Hospital}$. The hospital will then act as a prover and compute the proof of private key ownership and knowledge of pre-image of a hash i.e. the *PasswordHash* using pin and pk. This *PasswordHash* is cross checked against the one stored in ISC by the VSC. Hospital calls VerifyTX of VSC with the proof created and the re-encryption key as arguments to initiate this cross-checking.
3) If the proof evaluates as "accept" in VSC the re-encryption key will be added to the mapping of the beneficiary in ISC. This will trigger an event which can be listened to by the cloud and government auditors.
4) The hospital will now have to fetch data stored on IPFS by checking the hashes of EHRs in the ISC.
5) Using these hashes, the hospital will use IPFS gateway to fetch the encrypted EHRs. These EHR files will be all encrypted by a symmetric key (SYM_K) unique to the patient.
6) The hospital signs a token with their private key to send a request for patient data access to Cloud proxy re-encryption server.
7) The cloud server will check whether a re-encryption key exists on the ISC. If so, it will re-encrypt the symmetric key cipher text for the hospitals public key and send it to the hospital.
   $C^i_{SYMK} \leftarrow ReEnc(rencK_{Beneficiary->Hospital}, C_{SYMK})$
8) The hospital will decrypt the received cipher text from the cloud using it's private key and use the obtained Symmetric key to decrypt all the associated encrypted files it has fetched from IPFS and use it to gain better insights for patient treatment.

Health Insurance Claim

1) Once the empanelled hospital has treated the patient for one of the 1350 procedures covered by PM-JAY a new EHR will be generated by them.
2) This EHR will be Symmetrically encrypted using the unique SYM_K associated for the beneficiary and uploaded to IPFS.

TABLE II: Zokrates library metrics

Authenticate.zok	Time Taken	Memory
Compile Time	25.65s	12.2MB
Trusted Setup	17.1s	36MB
Compute Witness	3.0s	492KB
Generate Proof	2.0S	4KB
Export Verifier.sol	1s	29KB

3) The hash of the encrypted EHR along with Public Key of the beneficiary and procedure will be uploaded to the ISC causing an event to be triggered.
4) This event will be logged to all the stakeholders including the federal government which will monitor for fraudulent activity on the blockchain.
5) The balance of the beneficiary will be deducted according to the treatment specified and the equivalent amount will be added to the hospital's balance on the health Insurance smart contract.
6) Federal government monitors for any fraudulent activity being performed by the empanelled hospital it can ask them to present the decrypted EHRs as proof of treatment.

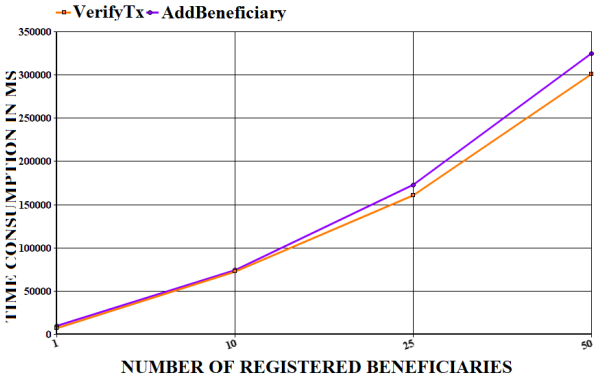
V. Experimental Results

We present the experimental analysis for our proof of concept implementation of an authentication mechanism using Zokrates and Solidity smart contracts. We used a 1.6 GHz i5 Macbook Air with 8 GB of RAM for compiling and deploying our code to the Rinkeby test network. Zokrates library was used using a docker container as recommended by it's documentation and IPFS was deployed on top of Amazon Cloud to compare it to the centralized S3 service provided by the same. For the testing of the scalability of our authentication scheme we used Ganache a tool developed by Consensys for setting up a local RPC to simulate a blockchain without the long block mining times.

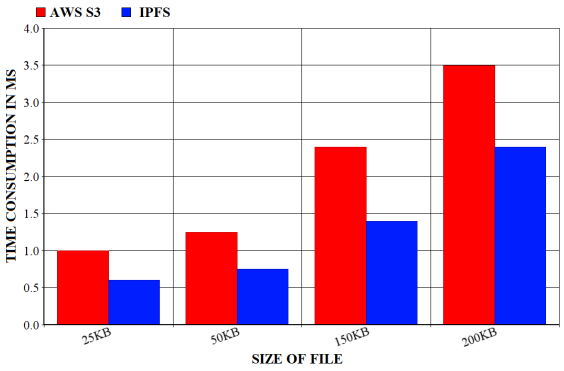
In Table II we present the time taken for Zokrates DSL code (Authenticate.zok) to be compiled and generate proofs using pghr-13 proving scheme [24]. Note that only Compute Witness and Generate Proof needs to be repeated by the hospital when it wishes to gain access to a new beneficiary's EHRs.

In Figure 3(a), we measured the average time consumption for verification of proofs on-chain using the deployed smart contract and also the time taken to add a new beneficiary while checking if the proof submitted is not reused by the hospital. The time taken to verify a proof on chain (VerifyTx) is recorded along with our AddBeneficiary which internally calls verifyTx and also stores PRE key on chain. From the above results it is obvious that our proposed solution remains feasible for implementation as even in the case with 50 consecutive registrations which takes less than 6 minutes.

We also evaluated performance of EHR sharing on IPFS with a centralized storage like Amazon S3 and observed that the IPFS has consistently out-performed. We present the results below in Figure 3(b). Gas cost for storage of more hashes, PRE-Keys and verification of proofs will change with increase in number of users/beneficiaries. We have implemented and studied key functionalities involving authentication as to present a feasibility study for our solution. We observed that gas cost of verification remains constant and does not vary with input, as depicted in Table III. As we can see that the gas



(a) Time to verify a proof on chain versus time to store and check a submitted proof in ISC



(b) Comparison of IPFS versus AWS S3

Fig. 3: Time Overheads

TABLE III: Gas cost of VSC functions and deployment cost on Rinkeby

Smart Contract	Gas Cost
Deploy verifier	1282112 GasUsed
VerifyTx	1896490 GasUsed
AddBeneficiary ISC	1911490 GasUsed

used to verify a proof on-chain is around 1.9 Million units which is much less than the 9 Million block gas limit on the Ethereum Mainnet, proving its feasibility to implement this authentication mechanism.

VI. Security Analysis

Security has paramount importance in an EHR management system, since medical data needs to be handled with privacy and protected against malicious actors. This section proves the security of our proposed system through different threat scenarios.

Theorem 1: Assuming an actor can gain access to all EHRs in IPFS without consent of corresponding beneficiaries, such an attacker will be unable to read medical data.

Proof: In our proposed solution all the medical records (EHRs) are encrypted with unique symmetric keys for each corresponding beneficiary. We use strong 256 bit symmetric encryption like AES (Advanced Encryption Standard) to ensure it is challenging for attacker to guess the key. Even if attacker is able to guess symmetric key for an individual it is

nearly impossible to decrypt all the files. The symmetric keys corresponding to each beneficiary are encrypted and stored in a protected Proxy Re-encryption Cloud. Since, these keys are encrypted with public key of the beneficiary, in order to retrieve keys the actor must first be granted access via a proxy re-encryption key stored on the Blockchain smart contract.

Theorem 2: Suppose an individual loses or misplaces his/her E-card which is obtained by a malicious party. It is impossible for such a party to misuse the E-card to obtain EHR access or make insurance claims.

Proof: In order to use E-card at an empanelled Healthcare provider, the adversary must be aware of a unique pin or passphrase which is used to create the Zero-knowledge proof along with private key on the E-card. Our smart contract implementation also allows for contract owner i.e. government to disable accounts of certain beneficiaries if the card is reported stolen by them. The card may be confiscated the healthcare provider if tried to be used once disabled. Similarly a new E-card may be issued to the legitimate beneficiary by using a new pin and the initial balance of the new account can be set to remaining balance of previous account.

VII. Conclusion

This paper proposes a novel healthcare framework for the implementation of India's recently launched Aayushman Bharat Yojna using Blockchain technology and Proxy re-encryption. We identified the limitations that prevent implementation of 'traditional' Blockchain/Cloud EHR sharing architecture in a country like India and proposed a convenient E-card based solution. We utilized Zero-knowledge proofs as an authentication mechanism in our system which allowed us to efficiently and securely share access to EHR in a privacy-preserving manner. A hybrid encryption model with both Symmetric and Asymmetric encryption is used to overcome the limitations of Proxy Re-Encryption which acts as an access control mechanism to medical data. Security analysis and detailed evaluation of the benefits and feasibility of the proposed architecture was also performed. Studying these merits of our model, it can transform healthcare in India and achieve the target of universal health coverage.

References

[1] U. D. of Health and H. Services, "Summary of the hipaa security rule," https://www.hhs.gov/hipaa/for-professionals/security/laws-regulations/index.html, 2018. [Online]. Available: https://www.hhs.gov/hipaa/for-professionals/security/laws-regulations/

[2] "Gdpr," https://eugdpr.org/the-regulation, 2017.

[3] EY, "Global analysis of health insurance in india," https://www.ey.com/Publication/vwLUAssets/EY-global-analysis-of-health-insurance-in-india/$File/ey-global-analysis-of-health-insurance-in-india.pdf.

[4] M. of Rural Development, "Socio-economic caste census," https://secc.gov.in. [Online]. Available: https://secc.gov.in/welcome

[5] S. J. et al., "Blochie: A blockchain-based platform for healthcare information exchange," in *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*, June 2018, pp. 49–56.

[6] A. A. et al., "Medrec: Using blockchain for medical data access and permission management," in *2016 2nd International Conference on Open and Big Data (OBD)*, Aug 2016, pp. 25–30.

[7] X. He, S. Alqahtani, and R. Gamble, "Toward privacy-assured health insurance claims," in *2018 IEEE International Conference on Internet of Things (iThings)*, July 2018, pp. 1634–1641.

[8] "Healthcare rallies for blockchains," https://www.ibm.com/downloads/cas/BBRQK3WY. [Online]. Available: https://www.ibm.com/downloads/cas/BBRQK3WY

[9] D. C. Snchez, "Zero-knowledge proof-of-identity: Sybil-resistant, anonymous authentication on permissionless blockchains and incentive compatible, strictly dominant cryptocurrencies," Cryptology ePrint Archive, Report 2019/546, 2019, https://eprint.iacr.org/2019/546.

[10] E. Foundation, "zksnarks in a nutshell," https://blog.ethereum.org/2016/12/05/zksnarks-in-a-nutshell/.

[11] D. C. N. et al., "Blockchain for secure ehrs sharing of mobile cloud based e-health systems," *IEEE Access*, vol. 7, pp. 66 792–66 806, 2019.

[12] S. Sharaf and N. F. Shilbayeh, "A secure g-cloud-based framework for government healthcare services," *IEEE Access*, vol. 7, pp. 37 876–37 882, 2019.

[13] Y. et a, "Multiparty privacy protection for electronic health records," in *2013 IEEE Global Communications Conference (GLOBECOM)*.

[14] A. e. a. Al Omar, "Medibchain: A blockchain based privacy preserving platform for healthcare data," in *Security, Privacy, and Anonymity in Computation, Communication, and Storage*. Cham: Springer International Publishing, 2017, pp. 534–543.

[15] J. Adler-Milstein and D. W. Bates, "Paperless healthcare: Progress and challenges of an it-enabled healthcare system," *Business Horizons*, vol. 53, no. 2, pp. 119–130, 2010.

[16] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[17] M. e. a. Blaze, "Divertible protocols and atomic proxy cryptography," in *Advances in Cryptology — EUROCRYPT'98*, K. Nyberg, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 127–144.

[18] D. Nuñez, I. Agudo, and J. Lopez, "Proxy re-encryption: Analysis of constructions and its application to secure access delegation," *Journal of Network and Computer Applications*, vol. 87, pp. 193–209, 2017.

[19] S. S. C. et al., "Efficient unidirectional proxy re-encryption," Cryptology ePrint Archive, Report 2009/189, 2009, https://eprint.iacr.org/2009/189.

[20] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof-systems," in *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, ser. STOC '85. New York, NY, USA: ACM, 1985, pp. 291–304. [Online]. Available: http://doi.acm.org/10.1145/22145.22178

[21] J. Eberhardt and S. Tai, "Zokrates-scalable privacy-preserving off-chain computations," in *2018 IEEE International Conference on Internet of Things (iThings)*.

[22] N. Aayog, "National health stack, strategy and approach," http://www.niti.gov.in/writereaddata/files/document_publication/NHS-Strategy-and-Approach-Document-for-consultation.pdf.

[23] "What are zk-snarks?" https://z.cash/technology/zksnarks.

[24] B. P. et al., "Pinocchio: Nearly practical verifiable computation," https://eprint.iacr.org/2013/279.