IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 12, Issue 3, March 2023

DOI: 10.17148/IJARCCE.2023.12305

# Blockchain Architecture for Data Privacy and Security in Healthcare

**Sagar Mal Nitharwal[1], Pragya Chaudhary[2]**

Assistant Professor, Computer Science, BTKIT Dwarahat, Almora, India[1]

Assistant Professor, Computer Science, BCE Bakhtiyarpur, Patna, India[2]

**Abstract**: With the rapid advancement of blockchain technology in recent years, its application in scenarios requiring confidentiality, such as the health sector, has become encouraged and extensively discussed. This paper presents an architecture for protecting the privacy of stored health-related data. This paper presents an architecture to ensure the privacy of health-related data, which are stored and shared within a blockchain network in a decentralized manner through encryption with the RSA, ECC, and AES algorithms. To determine the impact of cryptography on the proposed architecture in terms of computational effort, memory utilization, and execution time, evaluation tests were conducted. The results primarily impact the execution time and the increase in computational effort for transmitting data to the blockchain. This is justifiable in light of the privacy and security offered by the architecture and encryption.

**Keywords:** blockchain; cryptography; DApp; health data; data privacy; encryption.

## I. INTRODUCTION

Several advances related to blockchain technology have been recently consolidated, notably: the advent of blockchain 2.0, the blockchain network Ethereum (major programmable and public blockchain), the Hyperledger Fabric (private and permissioned blockchain), the improvement of smart contracts, and the use of encryption on the data flowing through the blockchain, such as Elliptic Curve Cryptography (ECC). As a result, it has become possible to develop ways to ensure data privacy, integrity, and access control within a specific application in various scenarios and solutions as described. According to Nakamoto the data is publicly visible to everyone on the blockchain network. Consequently, this information must be encrypted before being stored to ensure the confidentiality of the data and to keep the content private, helping to reduce the risk of the pseudonym being linked to the real identity of the blockchain user, which is crucial to promote sharing based on the need-to-know. In addition, blockchain makes it possible to ensure that data cannot be deleted or tampered with. To ensure privacy, in some cases, it is necessary to use cryptography, which has several techniques and algorithms that can be used to implement the security it provides. The article by Aguiar et al. [14] describes creating a blockchain framework using Hyperledger Fabric, a private, permissioned blockchain, and techniques for maintaining privacy when sharing health data. The choice of the blockchain network and the use of anonymization techniques like K-Anonymity distinguish it as a distinct strategy from the one suggested in this paper. Omar et al. [15] use the MediBchain, which is similar in that it aims to give patients and users control over their cloud data while using the ECC algorithm encryption to accomplish pseudo-anonymity and privacy. The authors do not present the use of a different encryption algorithm, the use of IPFS to reduce the expense of storage in the blockchain, and the possibility of granting access to the files. The developed system's cost and protocol are given with in-depth analysis.

Proposed access control is made in the work of Gan et al. [16], where patients will be in charge of their medical data, and institutions can access it without permission, but access can be revoked at any moment. The paper examines an incentive system where patients who contribute their data are rewarded based on an evaluation criterion. A private, permissioned consortium blockchain and an improved version of the Paillier homomorphic encryption mechanism are the two methods Liang et al. [17] suggest for storing personal data. They also use IPFS for off-chain storage in the experiments, which measure the effectiveness of the Paillier encryption mechanism. Even artificial intelligence (AI) has been used in some blockchain apps. According to Dinh and Thai, blockchain and AI have a disruptive combination [19]. Artificial intelligence has numerous uses, including time series and computer vision. In Ref. [33], a data protection scheme with attribute-based encryption (CEC-ABE) combined with a blockchain is proposed to protect electronic health records in edge cloud environments. In the tests, the proposed algorithm (CEC-ABE) is compared with other algorithms, for example, CP-ABE, to evaluate the performance in the proposed architecture by checking the computational cost of each stage (global initialization, key generation, plaintext encryption, outsourced decryption, and final decryption). Shahnaz, Qamar, and Khalid [34] presented a framework for the health sector to securely store electronic records in the blockchain with access rules, using IPFS to store the files.

  

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 12, Issue 3, March 2023
DOI: 10.17148/IJARCCE.2023.12305

However, the work does not focus on privacy and encryption of data to ensure security since the security of the presented system depends on the very security provided by the technologies used, unlike this work, which aims to ensure privacy by adding one more encryption scheme. Tests are also performed simulating a scenario with several users using different framework functions, where three criteria are evaluated: execution time, transfer rate, and latency.

## II. TOOLS FOR THE DEVELOPMENT OF ARCHITECTURE

In this section, we have discussed the tools that are used to design the architecture for the data privacy in healthcare data.

### 1. Truffle:

Truffle is a development environment, test framework, and asset pipeline for blockchains using the Ethereum Virtual Machine (EVM). According to the website [47], the features that the developer gets to enjoy when using Truffle are:

- Integrated binary smart contract compilation, binding, deployment, and management
- Automated contract testing;
- Programmable and extensible deployment and migration framework;
- Network management for deployment in public and private networks;
- Package management with EthPM & NPM, using the ERC190 standard;
- Interactive console for direct contract communication;
- Configurable build pipeline;
- External script executor that runs scripts in a Truffle environment.

### 2. MetaMask:

Users can manage their accounts, keys, and tokens in a variety of ways, including hardware wallets, with the help of MetaMask, an encrypted (digital) wallet and gateway to blockchain apps that separates them from the context of the website. Both a browser extension and a mobile software are accessible. Developers can communicate with Ethereum's (universally accessible) API, which recognises users of Web3 compliant platforms. Every time a call for a transaction signature occurs, MetaMask will ask the user to confirm the transaction and show how much it will cost. Through the Infura API, MetaMask is already set up with some links to the Ethereum blockchain network and to a number of test networks. Aside from Ethereum, MetaMask is presently compatible with any other blockchain (public or private) that provides a JSON RPC (Remote Procedure Calling) API [49].

### 3. React:

React is a JavaScript library used for building user interfaces. This library is declarative, which makes your code more predictable and easier to debug. It is component-based, making it easy to pass various types of data throughout your application and maintain a state outside of the Document Object Model (DOM). React components implement a ”render()” method that will receive input data and return what should be displayed. In addition, a component can maintain internal state data. React makes it easy to interface with other libraries and frameworks [50].

### 4. WEB3:

WEB 3.0 focuses on decentralization, unlike WEB 1.0 and WEB 2.0, and also brings some additional features such as being verifiable, self-governed, permissionless, and distributed. Web3 applications (DApps) run on decentralized networks, blockchains or even a combination of the two forming, for example, a crypto economic protocol, as cryptocurrency plays a big role in many of these protocols, providing a financial incentive (tokens) for nodes that want to participate in the creation, governance, contribution, or enhancement of a project.

### 5. Infura API:

The Infura API is supported by a microservices-oriented architecture that scales dynamically and offers instant access to the Ethereum network via HTTPS and WebSocket’s, creating an infrastructure for DApps swiftly and simply.

## III. ARCHITECTURE OF THE MODEL

An application architecture was created that enables the user to access and communicate with the Ethereum blockchain and IPFS in order to obtain their health-related data, which has been securely stored through encryption. This was done in order to achieve health data privacy protection and verify the effectiveness of encryption techniques. The user can utilise the application by adding a digital wallet to MetaMask. Users of the application can share documents and look for Adocuments they've stored. IPFS is used for the persistence of the mass data. The functional and non-functional requirements for the proposed solution were defined. As functional requirements, there are:

  

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 12, Issue 3, March 2023

DOI: 10.17148/IJARCCE.2023.12305

• Login: the application will only be available after the user login through MetaMask; Enter encryption keys: every time the user logs in to the application, his keys (public and private) will be generated. The application has a space for the user to enter his keys. The first time he accesses the application he can save his keys as he prefers, and the next time he logs in he can enter his keys;

• Store information: the user can send information to be stored in the IPFS, securely through AES encryption, which generates a hash to access the content on the network. This hash and the private key of AES, randomly generated, will be stored encrypted in the Ethereum blockchain, through one of the asymmetric key algorithms (ECC or RSA), to ensure privacy and access control to data;

• Confirm or reject transactions: the user, after sending a request to store information or a permission on the respective contracts, can accept or reject the transaction through MetaMask;

• Fetch information: the user can fetch the information they have stored in the IPFS through the hash stored in the blockchain, according to their logged in user;

• Grant permission: user A can grant access permission to a file that he has stored in the IPFS to user B, through some information such as the hash stored in the blockchain, the specific time that the permission will be valid, the address of user B, and the public key of user B;

• View files with permission: a user that has been granted permission by another user to a file, can have access to view that file within the time that was set by the user that granted the permission. Regarding the non-functional requirements, the following were listed:

• DApp: the application will have the characteristics of a decentralized application;

• Application storage: the application does not store any data permanently and centrally, only temporarily;

• Data security: files sent to IPFS will be encrypted by the AES algorithm. The hash, which identifies the location of the data, and the AES private key (used to encrypt the file) will be encrypted using RSA or ECC before being stored in the blockchain, thus allowing the user to securely reaccess their data. AES will encrypt the private key shared in the permission. Figure 1 shows the solution architecture with the information exchange flows between the six components.

The following describes the components of the architecture:

1. User: will access the application, send and receive data (information, files and permissions), as well as confirm transactions and pay the persistence fee for this data in the blockchain;
2. DApp: responsible for communicating with the digital wallet interface (MetaMask), with the blockchain (Ethereum) sending and receiving the data, with the decentralized bank (IPFS) sending the files and handling the return hash, and with the user receiving and showing the requested data, and, finally, performing the encryption/decryption of the hash, the AES private key, the file, and the shared AES private key when granting permission;
3. Digital Wallet Interface (MetaMask): responsible for managing the transactions performed in the DApp, requesting user confirmation and debiting the fee for persisting the data in the blockchain;
4. Blockchain (Ethereum): responsible for keeping the smart contracts that govern how the pertinent information of the file and the permission will be stored, such as, for example, the hash that is used to access the file in the IPFS, the address of the user who performed a transaction and the address of the user to which permission is being granted, thus performing the link of the respective user to certain information within the smart contracts;
5. Decentralized file system (IPFS): responsible for receiving the encrypted files from DApp, storing them and returning a respective unique hash that indicates where this certain information is in the IPFS network; Encryption: this is the key component needed to ensure the security and privacy of the data stored in the proposed architecture, because through the RSA or ECC encryption algorithms the application encrypts the hash and the private key before sending this information to the blockchain. With AES the file and the shared private key are encrypted when granting permission. In this way, only the user who has access to the private key to decrypt this hash and the AES private key will be able to access the stored information.
6. This process ensures that the information on the blockchain and IPFS is secure.

  

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 12, Issue 3, March 2023

DOI: 10.17148/IJARCCE.2023.12305

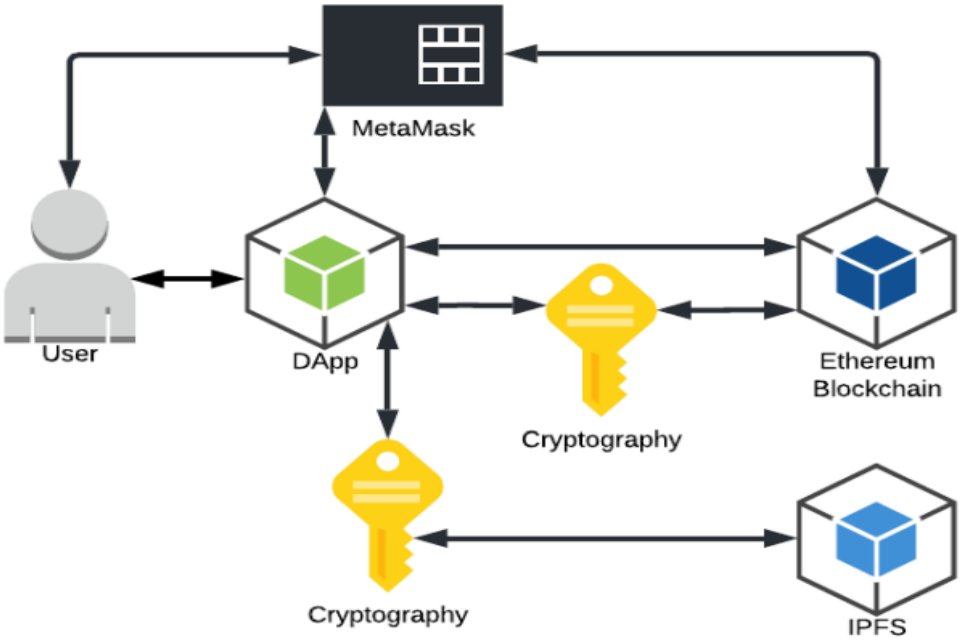


Fig. 1: Architecture schema.

## IV. CONCLUSION

In addition to putting the control of these data under the user's control, this paper presented an architecture that allows the management of user health data storage in a decentralized manner through blockchain and using cryptography to guarantee data privacy.

Two cryptographic strategies were put into practice using the RSA and ECC algorithms to see how they would affect the design. The data sent to the IPFS platform were encrypted using the AES algorithm. Due to the use of a public blockchain, the creation of the architecture using the outlined set of tools and technologies, along with the use of encryption on sensitive data, can be used and implemented in real-world situations at low initial implementation costs. This study analyzed the encryption methods employed, attempting to draw parallels between two of the most popular asymmetric key algorithms to aid in selecting the best method for the application being developed.

## REFERENCES

[1]. de Aguiar, E.J.; dos Santos, A.J.; Meneguette, R.I.; De Grande, R.E.; Ueyama, J. A blockchain-based protocol for tracking user access to shared medical imaging. Future Gener. Comput. Syst. 2022, 134, 348–360.
[2]. Omar, A.A.; Bhuiyan, M.Z.A.; Basu, A.; Kiyomoto, S.; Rahman, M.S. Privacy-friendly platform for healthcare data in cloud based on blockchain environment. Future Gener. Comput. Syst. 2019, 95, 511–521.
[3]. Gan, C.; Saini, A.; Zhu, Q.; Xiang, Y.; Zhang, Z. Blockchain-based access control scheme with incentive mechanism for eHealth systems: Patient as supervisor. Multimed. Tools Appl. 2020, 80, 30605–30621.
[4]. Liang, W.; Yang, Y.; Yang, C.; Hu, Y.; Xie, S.; Li, K.C.; Cao, J. PDPChain: A Consortium Blockchain-Based Privacy Protection Scheme for Personal Data. IEEE Trans. Reliab. 2022, 1–13.
[5]. Dinh, T.N.; Thai, M.T. AI and Blockchain: A Disruptive Integration. Computer 2018, 51, 48–53.
[6]. Shahnaz, A.; Qamar, U.; Khalid, A. Using Blockchain for Electronic Health Records. IEEE Access 2019, 7, 147782–147795.
[7]. Shahnaz, A.; Qamar, U.; Khalid, A. Using Blockchain for Electronic Health Records. IEEE Access 2019, 7, 147782–147795.
[8]. TruffleSuite. Truffle. 2022. Available online: https://trufflesuite.com/docs/truffle/index.html
[9]. MetaMask. Introduction. 2021. Available online: https://docs.metamask.io/guide/
[10]. REACT. React. 2021. Available online: https://pt-br.reactjs.org/