The Use of Blockchain for Secure Storage of Data for the Healthcare Field (Group R & D)

Robert Benish
Virginia Tech
rabenish@vt.edu

Deniz Aytemiz
Virginia Tech
denizaytemiz@vt.edu

ABSTRACT

The need for a way to securely store a centralized database for the medical field has been a struggle in the field for years. There has always been the concern of security as there is so much personal and confidential information that needs to be protected in the medical field. The solution could be using a combination of blockchain technologies as well as cloud technologies. The use of blockchain could create a secure and accessible database that could be used in the medical field to speed up processes that have, in the past taken much longer. In this research project the focus is to model a system that can be used securely and efficiently. The process is important because with a universally accessible database there can be faster care to those in need. We reviewed scientific articles that also back up this cause. We examined the designs with blockchain and cloud that also aims for the same objectives. Eventually we implemented a design we concluded that would be the most suitable for this task.

KEYWORDS

Asymmetric Encryption, Decentralization, Third Cloud Services, Medical Data Storage

ACM Reference Format:
Robert Benish and Deniz Aytemiz. 2018. The Use of Blockchain for Secure Storage of Data for the Healthcare Field (Group R & D). In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 7 pages. https://doi.org/XXXXXXX.XXXXXXX

Conference'17, July 2017, Washington, DC, USA
© 2018 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

In the field of healthcare there are many challenges that revolve around accessing patient data securely as well as quickly. The use of blockchain technology is a key stepping stone in innovating the healthcare industry. The protection and secure storage of data in the medical field can cause critical issues because of the importance of accuracy and availability.

For our project we are proposed to create and manage a private secure database that is connected to the cloud for blockchain for the healthcare field application. We have chosen this idea because when doing research for a project idea one of the major challenges of having a secure database for a healthcare field application is the lack of storage on the blockchain and data that is constantly being changed such as patient information. In this proposed project we aimed to design a storage framework that can connect blockchain to some sort of theoretical cloud storage. From other research it is a smart idea to do it this way so no one shareholder can have power over the data.

In the scheme proposed, the large medical data is encrypted and stored on cloud storage off the chain. It stores a hash of the data, records and patients public key also. The new data will be added to the blockchain once it is decrypted with the patients private key. Therefore it is only accessible by anyone but not interpretable to anyone but the patients himself/herself and to the medical intuitions via permission. Blockchain holds the index information of data and transaction records. Once the information is stored by the medical institution it is then immutable and cannot be tampered with. Our proposed project comes into play when we look at the need for a database connection between the blockchain and cloud storage so the files can be accessed.

Overall we believe that this project has a great deal of potential for useful research so that storage of medical information on the blockchain can be more feasible as a secure and effective use of blockchain technology. There are many ways to use blockchain to connect a database and we are going to research in terms of latency which implementation would be best for the use in the healthcare field. Our focus is to implement a blockchain network with cloud storage possibilities built.

2 RELATED WORK

In paper [1] , a service framework and design of a storage scheme with cloud for blockchain to be implemented in the medical data storage process is presented. In this architecture presented, cloud is leveraged as a storage under the chain and it holds encrypted medical data which cannot be decrypted without patients key whereas blockchain holds index of medical data and transaction records. This paper is useful since it comprehensively highlights many vulnerable aspects of currently used system in terms of medical data sharing and emphasizing the importance of building decentralized,secure and private environment in addition to discussing the strengths and weakness blockchain introduces in the architecture. In paper [6] , a similar idea in terms of functionality of cloud is mentioned as in paper[1]. In this scheme, cloud is not a blockchain node but connected via intermediate layer that multiple VMs that gets data from cloud via RESTful API(s) and stores them in blockchain by contacting with smart contracts executed in chain. This paper is useful since it presents different design in terms of blockchain based cloud storage and also underlines the security vulnerabilities and how these are countered within design. Paper [5] supplies many background and key interpretations on blockchain and cloud. It presents existing techniques in both technologies their advantages and disadvantages along with possible future directions they may take. Different designs on blockchain distributed cloud storage techniques are examined and many real-life applications leveraging both technologies are explored. It is very useful in terms of analyzing and choosing the optimal design for our project. Paper[8], proposes another decentralized cloud storage framework with access control using Ethereum blockchain and ciphertext-policy attribute-based encryption.The key information is stored in blockchain and the encrypted data is stored in cloud. This paper is useful since it presents comprehensive overview on how cloud and blockchain technologies are integrated and shares the experiment results in terms of gas used, cost and run time. Paper [3], gives insight on the current security and privacy concerns regarding use of blockchain, analyzing its features that enables privacy in healthcare data storage and the features restricting it. It also highlights potential direction of technologies regarding medical data storage. All these papers gave various insights on storage of medical data, blockchain technology and cloud storage in addition to presenting different architectures with different capabilities. Some of them lacks the performance evaluation or gives different metrics. We tried to find the most feasible and scalable design and aim to integrate cloud with blockchain.

3 OVERVIEW

The approach that we took is to spearhead this project with a scientific mindset. We are going to collect data on all of the viable software components that are going to be needed to complete this. Working as a team we focused on gaining skill in database management as well as how to access the blockchain.

- The first task is how we determined how we are going to access the blockchain
- The second task is that we decided to use an SQL database as the cloud software that we will be using for the data storage.
- We then figured out how the blockchain is implemented as well as the cloud data are going to communicate with each other.

After a long process we decided that the approach that we went with was to implement a simple database connection to show the process for the point of this project. We implemented a simple blockchain structure as well as showed how the request would looking using specific keys linked to the patient. As heavily mentioned in the final presentation out focus for this project was to show through research and data how blockchain can be used to produce a secure and useful centralized database that could be used in the medical field to speed up the process of accessing medical files.

4 PROJECT TIMELINE

The project plan mainly consists of 2 parts; research and demonstration. The research that we conducted is mainly focused on secure storage on the blockchain. This was linked with the research on what kind of cloud storage we determined is appropriate for our research. Amazon Web Services was considered to be one of the larger structures that can be covered, how that is linked to the blockchain is the largest feat. After determining how they will be linked there will me a movement to create a demonstrate-able work of how this research will be effective in the medical field for storing patient information. This is the timeline that was followed for the production of this report and research.

4.1 Milestones That We Accomplished

First of all, with literature review we analyzed the designs of cloud integrated blockchain solutions in medical data, different approaches on managing these two technologies and decide on a design to implement. Then the next goal was tackling the problem of realizing the small-scale demo. We researched online tools and software on how to construct a blockchain integrated cloud system. Finally realizing demonstration. These are the examples of the milestones that we set initially and then slowly accomplished over the course of the semester.

- Milestone 1: Gain knowledge about cloud storage as well as blockchain integration.
- Milestone 2: Decide how we are going to manage these two technologies together.
- Milestone 3: Have a proposal for a demo for our final project.
- Milestone 4: Combine work from both technologies to have rough project complete.
- Milestone 5: Introduce the concept of an SQL database for use in the project as the database example
- Milestone 6: Have presentable database and blockchain to visualize the connection that could be made if further research was implemented.

This is the approach that we took to accomplish the goals that we originally set for this research. Each of these milestones was crucial in explaining and understanding the process that needs to me done to conduct proper research. Hopefully they were conducted correctly and the implementation is understandable.

5 DESIGN

For the approach to the design we needed to take into account many different factors. One of the hardest and most complex issues that is faced is making the connection of a database system and meshing that with the blockchain. We started the design process by trying to find research that had already been implemented and try to adapt our own ideas to what is already being created in today's market. After brainstorming for a while we decided that we need to implement the project in several different sections to be able to show the approach that we want.

After researching many of the different companies that you can implement blockchain structures on we realize that a decent amount of them cost money to be able to access transactions. This steered us away from the use of Ethereum and other platforms. We decided on creating our own implementation in JavaScript in Visual Studio Code so that we could show a simple implementation. We also went with a more simple approach for the design of the database connection with a SQL API request call function to show how it could be done with a larger research budget and team.

5.1 Details and Implementation

Data storage and access control to data are essential parts in the medical blockchain scheme. Medical test results are generated by medical institutions and along with the patients confidential information, they are stored in a third party cloud service. The files are not raw data but encrypted with each patients public key. When the medical file is requested by the patient, they receive the file form cloud storage and decrypt the information with their private key. The access to the data is fully controlled by the patient itself. Once the file is decrypted by the patient, the index value or the hash of the file which refers to the files location in cloud is pushed to blockchain by the intuition. Hence this information and the transaction is made public. This design provides decentralization in storage of medical data such that the medical files are actually reachable by any entity. This is important since a patient may change his/her hospital and patients new doctor would be able to access the medical information history of the patient because it is publicly accessible. However, without the private key of the patient the data can not be decrypted and therefore confidentiality is provided.

We took the approach of creating the basic structure of a system that will interact somewhat like pictured in Figure 1. The structure we aimed for would show that there is potentially a secure way to let patients interact with their data as well as the hospitals and third party insurance companies to access the data.

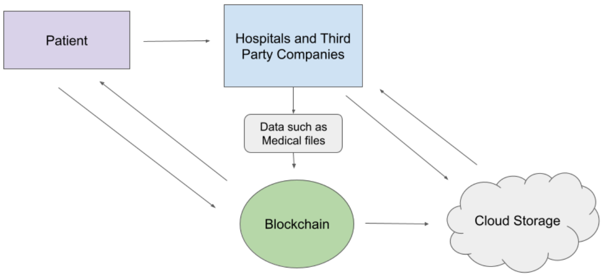
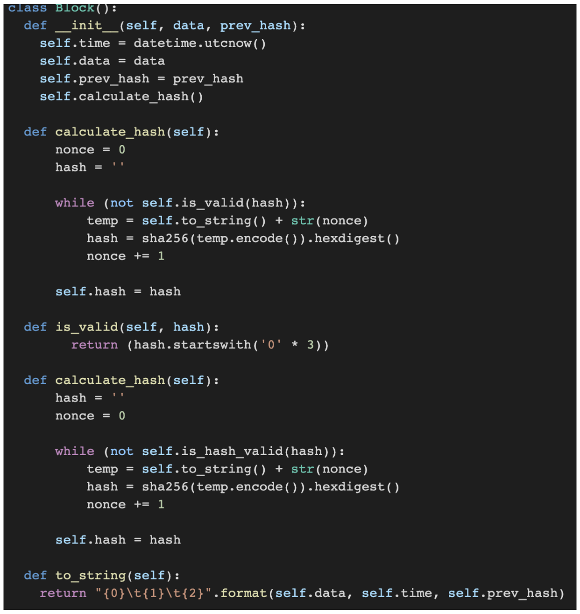

Patient
Hospitals and Third Party Companies
Data such as Medical files
Blockchain
Cloud Storage

Figure 1: This is the basic structure that we are hoping to accomplish.

Block is the constructor class where hash of previous block, data stored and hash of the current block is stored. The class has 3 methods.The to string method returns a string containing data and the previous block's hash. Calculate valid hash method is for hashing the current block. It starts from zero and gets incremented until a valid hash for data is found. This part is for PoW.

How we are using a cloud database is another key factor. Our plan for implementing the structure that we have talked about was needing the use of a cloud database. We have initially decided to move forward and use IBM's cloud function for our projects purpose. In a real world implementation for



```
class Block():
  def __init__(self, data, prev_hash):
    self.time = datetime.utcnow()
    self.data = data
    self.prev_hash = prev_hash
    self.calculate_hash()

  def calculate_hash(self):
      nonce = 0
      hash = ''

      while (not self.is_valid(hash)):
          temp = self.to_string() + str(nonce)
          hash = sha256(temp.encode()).hexdigest()
          nonce += 1

      self.hash = hash

  def is_valid(self, hash):
        return (hash.startswith('0' * 3))

  def calculate_hash(self):
      hash = ''
      nonce = 0

      while (not self.is_hash_valid(hash)):
          temp = self.to_string() + str(nonce)
          hash = sha256(temp.encode()).hexdigest()
          nonce += 1

      self.hash = hash

  def to_string(self):
    return "{0}\t{1}\t{2}".format(self.data, self.time, self.prev_hash)
```

Figure 2: The basic structure for how we are using our block class as well as the methods in it[9]

the medical field we believe that there should be another cloud service used.

The next part of our project revolves around how we want to implement the use of our block in the blockchain. The block needs to store several different things. In each individual block there needs to be a transaction as well as a block header. in the block header there is implemented all of the different features such as the time stamp as well as the hash code and the previous blocks id. In the transaction that is held in each of the blocks this holds the data.

Data owner deploys smart contracts and the hash of medical data is saved in smart contracts. Then the contract address, hash and the encrypted data is uploaded to the cloud and cloud returns the file path to the data owner. Data owner stores the encrypted key of medical data in Ethereum. When a third party company such as an insurance company wants to access the data access request is sent to owner. Data owner sets the period of access permission and records it into smart contract. Data owner the encrypt the secret key of third part and stores it into smart contract.

5.2 Architecture of the Database Connection

Our design considers cloud service as storage and in this design cloud is off-chain meaning it is not considered as a blockchain node. Therefore design requires a layer between blockchain and cloud that would execute API calls.

In the scheme Ethereum blockchain is used. Data users and data owners use smart contracts to store and retrieve encrypted data to process encryption and decryption. Each time smart contacts are called it is saved on blockchain. So the process is traceable.

Each node on the blockchain will be running a virtual Ethereum VM. By using Ethereum we can use existing programming languages to create applications such as Javascript, Python, and more.

Ethereum blockchain uses smart contracts. Smart contracts are programs written in high-level languages such as Solidity and run in a container in the blockchain system. In our scheme Smart Contract stores information about encrypted files.

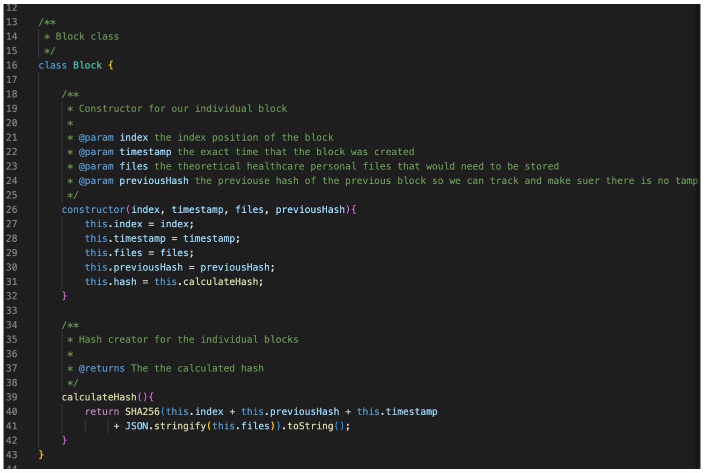
IBM cloud has built in functions for forming connection with blockchain infrastructure. It provides functions for sending any data from any web-enabled application. By the help of those we can store blockchain credentials in the database and use the credentials to make information transaction between cloud and blockchain.

Smart contracts are deployed on the Geth Ethereum client. They run in a container in blockchain and are naturally temper proof and anti-counter-fitting. In our scheme it will hold the path information on encrypted data in cloud.

5.3 Final Implementation

In the final version of our project, we preferred not to use Ethereum blockchain platform or third party cloud services such as AWS or IBM due to the fees they require for their services. Instead we had the approach of implementing the blockchain structure from scratch with JavaScript. This also helped us to see the attributes, parameters and methods associated with blocks and the blockchain. In addition we implemented a file for blockchain and connection of cloud for retrieving of data. Lastly, there is a class for generating asymmetric encryption key pairs.

In Figure 3 & 4 our blockchain class has constructor and other functions such as adding new blocks, creating initial block and returning the last block on the chain. The block class has the function of calculating the hash associated with that particular block along with a timestamp. There is data belonging to each block which will be retrieved from cloud storage. In the real life implementation the data would be the transaction record and the hash of the an encrypted file. The encrypted file referred to is the encrypted medical confidential information of a particular patient. In our implementation we pre-assumed that the database will be implemented in SQL therefore we used SQL queries in the connection file used for bridging cloud and blockchain.

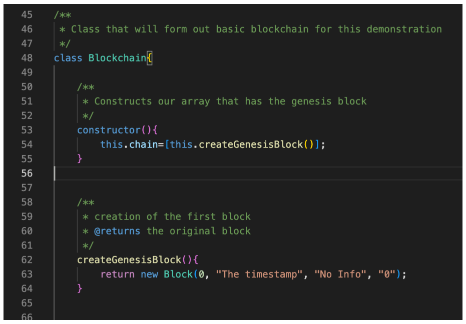


```
/**
 * Block class
 */
class Block {

    /**
     * Constructor for our individual block
     *
     * @param index the index position of the block
     * @param timestamp the exact time that the block was created
     * @param files the theoretical healthcare personal files that would need to be stored
     * @param previousHash the previouse hash of the previous block so we can track and make suer there is no tamp
     */
    constructor(index, timestamp, files, previousHash){
        this.index = index;
        this.timestamp = timestamp;
        this.files = files;
        this.previousHash = previousHash;
        this.hash = this.calculateHash;
    }

    /**
     * Hash creator for the individual blocks
     *
     * @returns The the calculated hash
     */
    calculateHash(){
        return SHA256(this.index + this.previousHash + this.timestamp
            + JSON.stringify(this.files)).toString();
    }
}
```

Figure 3: The basic structure of the block class that takes in several different parameters.



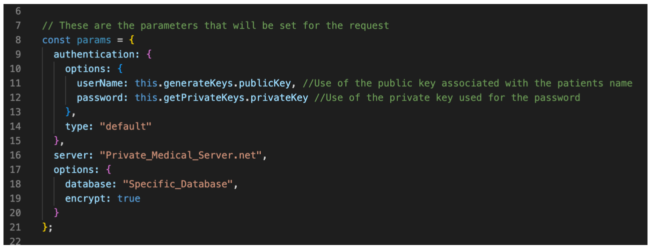
```
/**
 * Class that will form out basic blockchain for this demonstration
 */
class Blockchain{

    /**
     * Constructs our array that has the genesis block
     */
    constructor(){
        this.chain=[this.createGenesisBlock()];
    }


    /**
     * creation of the first block
     * @returns the original block
     */
    createGenesisBlock(){
        return new Block(0, "The timestamp", "No Info", "0");
    }

```

Figure 4: The implantation of the block class that forms the actual blockchain array.[7]

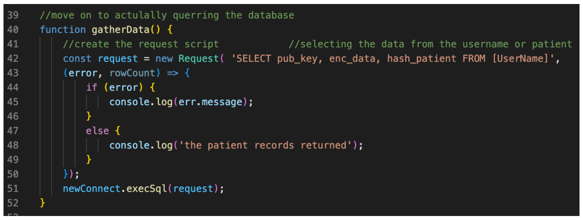


```

// These are the parameters that will be set for the request
const params = {
  authentication: {
    options: {
      userName: this.generateKeys.publicKey, //Use of the public key associated with the patients name
      password: this.getPrivateKeys.privateKey //Use of the private key used for the password
    },
    type: "default"
  },
  server: "Private_Medical_Server.net",
  options: {
    database: "Specific_Database",
    encrypt: true
  }
};
```

Figure 5: With of the use of a connection to a SQL database this is the construction of that.

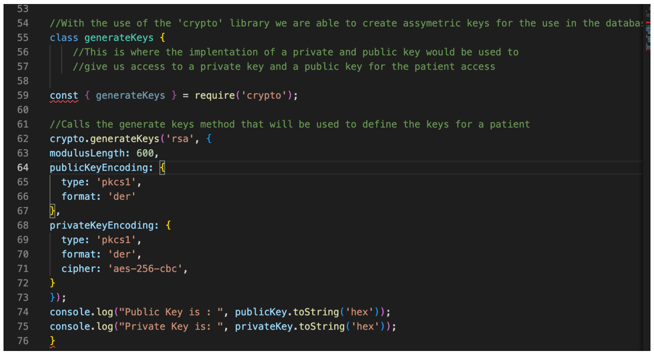
For the asymmetric key generation, we leveraged crypto module. It has method getKeyPair() in which parameters such as encryption scheme, key lengths and padding are specified. We chose RSA encryption knowing its preferred over AES in most encryption and verification designs. As you can see in the figure 5 there are placeholders for the insertion of the private and public key that is used as a getter from the implemented method later in file. In theory this would be implemented to take the keys of an individual patient's data that needs to be accessed. The private key would be generated once and be kept confidential similar to a persons social security number. The public key would just be the key that is linked to the patients name and would be implemented as the username for the system.



```
//move on to actulally querring the database
function gatherData() {
    //create the request script            //selecting the data from the username or patient
    const request = new Request( 'SELECT pub_key, enc_data, hash_patient FROM [UserName]',
    (error, rowCount) => {
        if (error) {
            console.log(err.message);
        }
        else {
            console.log('the patient records returned');
        }
    });
    newConnect.execSql(request);
}
```

Figure 6: The actual SQL request showing the selection.[4]

In Figure 6, this is the representation of how there could be a simple SQL selection request once the private and public information about the patient is input. As we have talked about in other parts of the paper we chose to go with a more simple approach of implementing the database connection so it can easily be visualized. Theoretically a hospital with a secure access point would be able to call on the blockchain and that would trigger the database call and the files or records of that patient would be returned.



```

//With the use of the 'crypto' library we are able to create assymetric keys for the use in the databa
class generateKeys {
    //This is where the implentation of a private and public key would be used to
    //give us access to a private key and a public key for the patient access

const { generateKeys } = require('crypto');

//Calls the generate keys method that will be used to define the keys for a patient
crypto.generateKeys('rsa', {
modulusLength: 600,
publicKeyEncoding: {
  type: 'pkcs1',
  format: 'der'
},
privateKeyEncoding: {
  type: 'pkcs1',
  format: 'der',
  cipher: 'aes-256-cbc',
}
});
console.log("Public Key is : ", publicKey.toString('hex'));
console.log("Private Key is: ", privateKey.toString('hex'));
}
```

Figure 7: The formation of the private and public keys used for accessing database.[2]

For Figure 7, this is where the key generation is created. As heavily mentioned in the presentation we wanted to keep security as one of the major factors that we wanted to focus on due to the risk of private healthcare information being leaked or illegally accessed. As you can see we went with the AES 256 for the encryption of the private key as we though this would be the most secure for this application. Another aspect that was heavily talked about in our final presentation

6 FINAL EVALUATION

In our initial evaluation of the progress on our project I believe that we took a large chunk out of the what we plan to accomplish for this project. Moving now on to our final evaluation of the project our perspective has changed some. We have shown how our progress has been completed as well as how our project makes sense in a real world application. As you can observe from the information given in the implantation section of this paper we are striving for a basic system structure that will help the field of medical data storage using block chain technologies. The data that we will be collected so far shows that this project makes sense.

For our evaluation we are attempting to transfer data from out blockchain interface to the cloud based database. We are still working on the connection between the two so we will explain how the process should be executed to make sense. In later iterations of the paper there will be evaluation and examples of how this is completed.

Overall I believe that the implementation for our project shows the true need for a centralized database system that could be implemented using blockchain for security and uniformity.

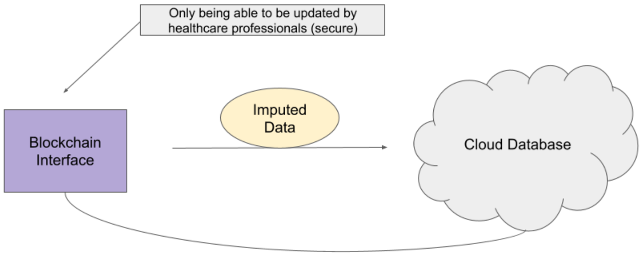
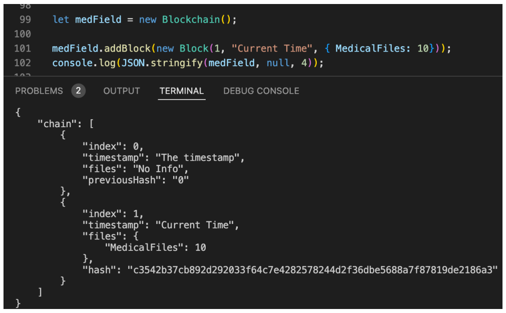

Only being able to be updated by healthcare professionals (secure)
Imputed Data
Blockchain Interface
Cloud Database

Figure 8: Overview of how our project will interact with both blockchain and cloud database.

In Figure 8, this is a representation of our output and what the interaction is like. There is a secure connection between both of them and the data that is inputted from the blockchain can not be changed or tampered with because of the blockchain interaction. In the final presentation of our project we will go more in depth on how this is implemented and how it is key to the medical field.



```
let medField = new Blockchain();

medField.addBlock(new Block(1, "Current Time", { MedicalFiles: 10}));
console.log(JSON.stringify(medField, null, 4));
```

```
{
    "chain": [
        {
            "index": 0,
            "timestamp": "The timestamp",
            "files": "No Info",
            "previousHash": "0"
        },
        {
            "index": 1,
            "timestamp": "Current Time",
            "files": {
                "MedicalFiles": 10
            },
            "hash": "c3542b37cb892d292033f64c7e4282578244d2f36dbe5688a7f87819de2186a3"
        }
    ]
}
```

Figure 9: The given output when blockchain is ran.

Figure 9 shows how the Blockchain class works. In the output the multiple blocks added to blockchain can be observed. They are all timestamped and has an index, hash and data. They are related or chained by holding and verifying their prior blocks hash. The data in this chain is medical file index and is retrieved from the cloud.

As mentioned in the final presentation a huge focus of this project was security. In the world of blockchain, security is a major factor as this is a distributed network. Security is held up by what are called Smart Contracts. When reviewing the code segment that we submitted there is mention of a method call that would check if the blocks were valid as this is what the premise of the smart contract is. Smart contracts or proof of work systems allow for the maintenance of security to make sure that the blocks have not been tampered with to prevent malicious tampering. Smart contracts and proof of work systems are very complex so we were not able to implement a smart contract system for this project, as that would be out of the scope and complexity. If this project and research were to be taken farther, then a complex and very secure smart contract or proof of work system would need to be implemented to make sure that no patients personal and private files would be accessible to the public.

7 CONCLUSION

In conclusion, working on this project we learned about the main practises currently on combining these two technologies; blockchain and third party cloud services which are getting more prevalent over time. Especially use of these technologies in medical data storage introduces better design in terms of confidentiality, security and efficiency. Blockchain contributed to the security in terms of making transaction records public for everyone to see and verify. It also contributed to the decentralization since the medical institutions conducting the test does not have access to decrypted file and the access rights of the data are fully controlled by the patient. Cloud contributes to the efficiency by being storage. Since the data is not stored on the blockchain and only the index of the file referring to it is stored the overall design is more efficient. The asymmetric encryption scheme is also one of the most important properties that contributes to the security of the design. Since the public key is stored on blockchain it is visible and accessible by anyone therefore medical intuitions can encrypt each person's medical file with their own key. Therefore actually anyone can access to anyone's file in encrypted form. But without the right private key associated with that public key they would not be able to decrypt the file.

8 BREAKDOWN OF CONTRIBUTION

In this section we will discuss how the work for this project was broken up between the two students Robert Benish and Deniz Aytemiz. Overall we separated the work on the report, construction of the presentation, research time, and code implementation evenly. We worked on the entire project together and there was not much time spent out of meeting with each other on this project. For the report we split the work into section of implementation. For the Robert worked on the Abstract, Introduction, half of the Related Work section and final code implementation. Deniz worked on the Initial Design, Approach Overview and the initial timeline. She also worked some on the related work. For finishing the project we both split even time working on implementing and proofreading the remainder of the paper and sections such as, The final implementation, details of the implementation, and the final evaluation. Overall the work was split at 50% for each of the two students.

REFERENCES

[1] Ding S. Xu Z. Chen, Y. 2019. Blockchain-Based Medical Records Secure Storage and Medical Service Framework. *Med Syst* 5, 1 (2019). https://doi.org/10.1007/s10916-018-1121-4

[2] GeeksForGeeks. 2021. *Node.js crypto.generateKeyPair() Method Key Generation.* https://www.geeksforgeeks.org/node-js-crypto-generatekeypair-method/

[3] Haripriya Kanagaraj, Brintha NC, and Yogesh Ck. 2021. *A Survey on Securing Medical Data in Cloud Using Blockchain.* https://doi.org/10.3233/APC210150

[4] Microsoft. 2021. *Connection of database.* https://docs.microsoft.com/en-us/azure/azure-sql/database/connect-query-nodejs?view=azuresql&tabs=macos

[5] Malaya Dutta Borah Pratima Sharma, Rajni Jindal. 2021. Blockchain Technology for Cloud Storage: A Systematic Literature Review. *Comput. Surveys* 53, 4 (July 2021), 1–32. https://doi.org/10.1145/3403954

[6] K. Moschou D. Votis S. Theodouli, K. Arakliotis and Tzo-varas. 2018. On the design of a Blockchain-based system to facilitate Healthcare Data Sharing. *17th IEEE International Conference On Trust Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering* 17/12, 1 (2018), 1374–1379. https://doi.org/10.1109/TrustCom/BigDataSE.2018.00190

[7] Savjee. 2021. *Blockchain.js.* https://github.com/Savjee/SavjeeCoin/blob/master/src/blockchain.js

[8] Xu Wang Shangping Wang and Yaling Zhang. 2019. A Secure Cloud Storage Framework With Access Control Based on Blockchain. *IEEE Access* PP (July 2019), 1–1. https://doi.org/10.1109/ACCESS.2019.2929205

[9] Joao Zsigmond. 2020. Creating a Blockchain from Scratch: The key concepts behind this ground-breaking technology, and how you can build one yourself. (May 2020), 1. https://levelup.gitconnected.com/creating-a-blockchain-from-scratch-9a7b123e1f3e