GenChain: A Blockchain Based Genetic Diseases Prediction System

Muhammed Esad Simitcioğlu
Department of Computer Science
Ozyegin University
Istanbul, Turkey
esad.simitcioglu@ozu.edu.tr

Emin Sadikhov
Department of Computer Science
Ozyegin University
Istanbul, Turkey
emin.sadikhov@ozu.edu.tr

Aleyna Nur Ölmezcan
Department of Computer Science
Ozyegin University
Istanbul, Turkey
aleyna.olmezcan@ozu.edu.tr

Özge Yılgür
Department of Computer Science
Ozyegin University
Istanbul, Turkey
ozge.yilgur@ozu.edu.tr

Dr. Kübra Kalkan Çakmakçı
Department of Computer Science
Ozyegin University
Istanbul, Turkey
kubra.kalkan@ozyegin.edu.tr

***Abstract*—Genetic testing can be used to learn about the probable genetic disorders a person may have based on their family history, however they are overpriced and out of reach for many people. Our solution to this problem is *GenChain* which is a secure, private consortium blockchain system that calculates the probability of genetic diseases. With our suggested system the blockchain will store the information of the genetic diseases a user and his/her family has and when requested it will calculate the possible genetic disease percentages. The solution involves sending genetic disease data from all hospitals registered in the private system to the *GenChain* along with the family numbers of the patients taken from a trusted authority. *GenChain* uses the Paillier cryptosystem, a probabilistic asymmetric method for public key cryptography, which has the property of homomorphism which we will use in our encryption scheme to protect the confidentiality and privacy of the system. The protocols that have been proposed are tested and verified that they are safe to use in such system. We have implemented our blockchain in Hyperledger Fabric.**

***Index Terms*—Blockchain, medical data, hyperledger fabric, pailler cryptosystem, homomorphic encryption, digital signature, smart contract**

I. Introduction

HEALTHCARE data management has been constantly innovating throughout the years with the developing hardware software and networking technologies [1]. Since the health information of a patient is confidential these healthcare records request safe systems to be protected rather than being kept in centralized third parties, such as a hospital database or clouds which is highly the case of how they are being stored today, this poses a safety concern for users to lose control of their health data, which can easily result in privacy leakage and a single point of failure [2].

Genetically transmitted diseases are a part of our lives, everyone either has them or has people around them suffering or dealing from such diseases. However genetic disease tests are not performed in a lot of hospitals and they are not very common. Some private hospitals or genetic diagnosis centers are generally not preferred by people due to their high costs. So people often realize they have a genetic disease when they start to show symptoms and go to hospitals even though early diagnosis is very important in their treatments.

The aim of this project is to provide a reliable, easy to reach and cost-effective solution to the problem of users not being easily accessed to the information of whether they can inherit a genetic disease from their family or not.

We have created security protocols for our system which were Genetic Disease Protocol, Family Tree Protocol and Prediction Result Protocol which are expounded comprehensively in our solution approach. We have tested if these protocols were secure enough to protect our *GenChain* system via the tool AVISPA and implemented our system in Hyperledger Fabric and tested it for throughput and latency.

Our contributions are:

- Providing private healthcare and data transfer.
- Making genetic disease detection accessible.
- Calculating without revealing patient data.
- Offering a decentralized healthcare system.

The remaining part of the paper is organized as follows. Section II explains the blockchain technologhy and the encryptions we have used along with Hyperledger Fabric and smart contracts. Section III includes similar studies that we benefited from and used as a source during our study. Section IV presents *GenChain*, objectives of this project and our assumptions and constraints. Section V provides the response time, security, privacy and accessibility of the system along with the performance analysis. Section VI gives discussions about our proposed our system. Section VII concludes the paper and presents the future work.

II. Background

A. Blockchain Technology

Blockchain [3] is a decentralized general ledger with a unique data format. This data structure uses cryptographically

verified tamper-resistant and unforgeable decentralization to connect data blocks in chronological order in the form of a chain. Public, private, and consortium blockchains are the three types of blockchains [4]. In this work, byzantine fault tolerance is designed as the consensus mechanism for consortium blockchain to guarantee system's availability. The blockchain's ledger is made up of multiple blocks. As shown in Figure 1, each block contains a block header and a block body. Block header contains multiple meta-information about the current block. For example, timestamp, a hash value for the blockchain body, and a hash value for the previous block. Block body is usually used to record the real data of the current transactions [5].

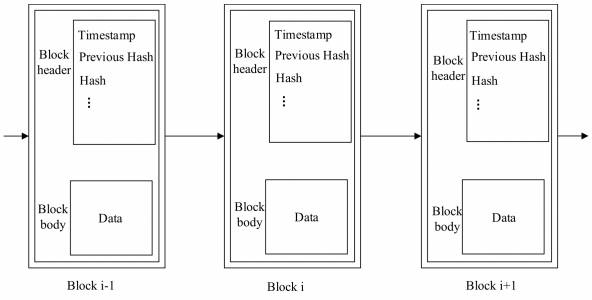

Fig. 1. Blockchain that is constructed up of a series of blocks [6]

Blockchain has a massive potential to improve the healthcare industry [4]. It has important features that will provide sensitive information, disease data, in a much more reliable way compared to traditional hospital systems. The following are the significant properties of the blockchain [5]:

1) Decentralization: There is no central node and every node is equal. Transaction records are kept by numerous nodes spread across the network, and each node records and maintains a comprehensive account. All nodes can control the operation.
2) Tamper resistance: Each block holds the hash of the previous block. When one of the blocks is changed, all subsequent blocks are recalculated. In this way, it is not possible to make changes by a single node in the chain.
3) Openness: The data of the blockchain is available to anyone, in addition to the private information of all parties engaged in the transaction being encrypted. Through the public interface, anybody may query block-data and construct useful applications.
4) Autonomy: The blockchain uses a consensus process that allows all nodes in the system to exchange data freely and securely. As a result, no human intervention is required.
5) Anonymity: Anonymity ensures that the identities of senders and receivers in transactions remain anonymous. Smart contracts are self-executing programs based on customer-supplier agreements that help in the control of the execution of detectable and irreversible transactions [4].

B. Byzantine Fault Tolerance

A system that is susceptible to intrusions and vulnerabilities is one that maintains its properties such as confidentiality, integrity, and availability, even though components and data are compromised by malicious attackers [7]. Byzantine Fault Tolerance ensures the confidentiality and reliability of a system whose components are attacked and intruded by a malicious attacker [8].

C. Homomorphic Encryption and Paillier Cryptosystem

Homomorphic encryption is an encryption change technique for ciphertext in which the result is also a ciphertext as shown in Figure 2. The output of the operations on the related plaintext data shall match the decrypted result after the resultant ciphertext is decoded [9]. Homomorphic encryption is used for privacy preservation.

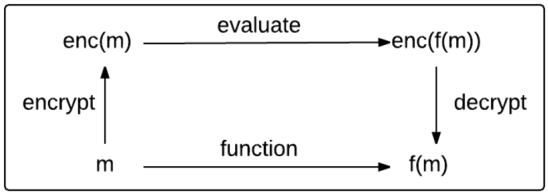

Fig. 2. Homomorphism and encryption [6]

In fully homomorphic encryption systems which we will be using for our project any kind of mathematical operations can be performed on the ciphertext. So the homomorphic system should support any number of arbitrary computations.

Pascal Paillier invented the Paillier cryptosystem in 1999. Paillier cryptosystem is a modular, public key encryption scheme which is also a partial homomorphic encryption technique that permits two forms of computation: addition of two ciphertexts and multiplication of a ciphertext by a plaintext integer [10].

Addition of two ciphertexts: When two ciphertexts are multiplied, the result decrypts to the sum of their plaintexts:

$$D_{priv}(E_{pub}(m_1)\times E_{pub}(m_2) \bmod n^2) = m_1 + m_2 \bmod n$$

Multiplication of a ciphertext by a plaintext: When a ciphertext is raised to the power of a plaintext, the result decrypts to the product of the two plaintexts:

$$D_{priv}(E_{pub}(m1)^{m2} \bmod n^2) = m_1 \times m_2 \bmod n$$

D. Hyperledger Fabric

Hyperledger Fabric is a permissioned blockchain system implementation and open source blockchain. Since it is built on a permissioned blockchain, it only permits interested stakeholders to be participant members and restricts others from accessing the network, altering the ledger, or initiating transactions.

Hyperledger Fabric network is made from various types of nodes such as client nodes, peer nodes, and ordering nodes associated with various organizations. A membership service provider supplies each node an identity that is associated with an organization. The identity of all parties are recognized and approved by all the nodes in the Hyperledger Fabric network. The fabric also has smart contract functionality which we will add into our system [11].

1) Peers: Peers are essential network nodes because they store copies of ledgers and smart contracts. The blockchain, which is an immutable log of transactions, is stored on peers as part of the ledger. An organization would typically have numerous peers to ensure data redundancy and to handle a heavy load. A peer is associated with a single organization, although it can be associated with many channels and host several ledgers and smart contracts. Smart contract calls are forwarded to dedicated chaincode containers in the network, and smart contract outcomes are used to update the network state. It is possible for peers to connect to orderers in order to receive new blocks of transactions and update the local blockchain copy, but it is not required.Rather of being managed by a single organization, fabric blockchain networks are managed by a group of organizations. Because they are controlled by — and serve as network connection points for — these companies, peers are critical to the way this type of distributed network is formed [12].

E. Smart Contracts

Smart contracts are self-executing contracts that are named and conditioned between trusted parties. Independent of trusted third parties, the smart contract is based on the blockchain and consensus submitted by the participants [13]. Smart contracts increase security as it is not connected to a single center. The start of the second generation of Blockchain is characterized by smart contracts based on the Ethereum Blockchain. Other systems, such as Hyperledger Fabric, Stellar, Waves, NXT, and others, appeared after that, allowing the creation of smart contracts. Each platform has its own set of functionality for creating smart contracts [14].

III. Related Works

Our study enables people who want to calculate the rate of genetic diseases that can be transferred from their families and that children can have with blockchain technology. When the literature and previous studies were examined, no other study with the same aim as our study have been found. However, there are different studies that use the tools and systems we used while creating our project, as well as using blockchain in the field of health. We tried to be inspired by other articles that are similar to our project and that benefit us and also find solutions to different problems in the health sector.

Singh et al. [15] have worked on this subject and tool also. They build a patient-centric design of a decentralized healthcare management system with blockchain-based EHR using javascript-based smart contract. Their transactions design between entities and protocols have shown us a great way. In this research they achieved maximum throughput at 186 TPS with about a 10 TPS (Transaction Per Second) decrease while moving larger and larger organizations which is inevitable for us. Because, we're planning to build this network with many hospitals.

After we decided to use Hyperledger Fabric in our study, we started to explore how we could test our study. Spengler et. al. [16] proposed a study about the performance of the network made by Hyperledger Caliper benchmarking. Although not directly related to our project purpose, this study showed that it can be beneficial for us to use Caliper in our performance analysis. It contributes to future application developers as their work demonstrates the impact of database usage on blockchain, as they themselves mention.

Health data is one of the most private and protected data people have. To maintain this confidentiality [17], this study used the Paillier homomorphic cryptosystem along with blockchain technology to conduct a statistical analysis on health data in a distributed blockchain network. Thanks to this study, which is one of the studies we inspired, we decided to calculate the data we need to use when calculating the disease rate, without revealing it at all. In addition, healthcare providers, insurance companies and similar companies will use the Paillier encryption technique to encrypt the results obtained from the queries in the proposed [17] system.

In Kuzlu et al. [18] research includes a study focusing on Hyperledger Fabric and evaluating the impact of network workload on the performance of a blockchain platform. Performance metrics that were specifically examined are throughput, latency, and scalability. Like our study, Hyperledger Caliper is used as a benchmark tool for performance analysis. When they performed performance analysis by transaction rate, blockchain throughput decreased and latency increased significantly when transaction speed increased above 200 TPS. When we want to reconcile this research with our own study, we can think that the procedure rate will increase as new disease status is added. At the same time, the authors emphasize that these metrics will also depend on smart contract complexity.

Dabbagh et. al. [19] compares performance between two of the blockchain platforms, Hyperledger Fabric and Etherium. The benefit that touched our work is how it analyzed the Hyperledger Fabric platform, not the difference between them. This study uses a smart contract with three functions: Open, Query and Transfer provided by Hyperledger Caliper. At the same time, since the codes of the mentioned functions are shared, this research has guided us in writing with Go.

IV. *GenChain*

Our project, *GenChain* aims to predict genetic disease probability of a patient based on his/her ancestors' data. These medical datas are so sensitive and private so it's important to do these calculations of probability without revealing anything. Based on our research, there is no other project that does what we want to do. To achieve this, we are developing a blockchain that contains all of the genetic disease information about patients and when requested will calculate the possibility

of a user's potential genetic diseases going through the family tree of the person.

The first objective of our project is to do genetic disease calculation without revealing anything. Even in the same family, we don't want to show who has a disease or who has not. In order to prevent this information from being exposed, we want to encrypt this data when its firstly released and not decrypting it again until the end of our calculations. In solution approach we have stated our security protocols in order to do private healthcare data transfer safely.

Second objective is to control this big data and to use it efficiently. We decided to add each user to the blockchain. This blockchain would cover all of the users in the country. All of the relatives of the users will also be in one blockchain. However, this approach has one main drawback. Performing search operations to create a family tree will be very complex.

Third objective is to find out from which family the patient is and to keep the information accordingly. We will not be able to keep it by surname since it wouldnt be distunguishive enough. Since we cannot ask the patient, we need help from a greater authority, in our case the government.

Our assumptions and constraints:

- For this project we need long-term data. If we start this project now we can use it efficiently a few generations later. So initially, In the chain, there is information about which genetic diseases the ancestors of each patient had.
- Patients, Hospitals and Government have agreed to use and support this blockchain.
- The probability of genetic disease transmission decreases from first-degree relatives to other relatives. For example, the transference effect of the father will be greater than that of the grandfather. We assumed that the rate of genetic disease transmission would decrease by 25% when moving up the pedigree.
- When we are doing our calculations, as genetic diseases, we have only used diseases inherited from both the mother and the father.

A. System Architecture

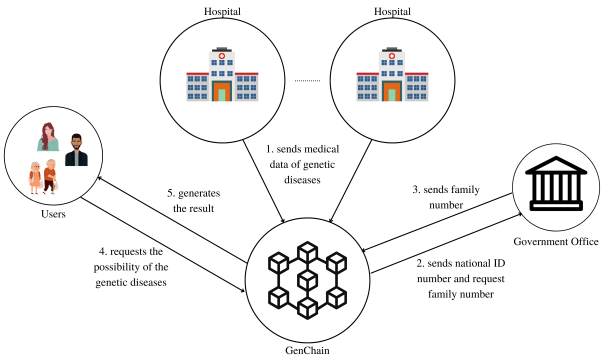

Fig. 3. System Architecture

We have designed a permissioned blockchain. The created *GenChain* will be in touch with hospitals and the government which are trustable authorities. All of the entities in our network will be authorized. With the help of the permissioned blockchain we don't need to check entities for fraud. Best tool for this purpose in the market is Hyperledger Fabric. Hyperledger Fabric is a permissioned blockchain system implementation and open source blockchain. In our model, we assume four entities will be enough for our implementation. Figure 3 shows entities and brief information about the transactions between them. The brief summary will be given below:

1) Hospital Sends Medical Data Of Genetic Disease: The patient will go to the hospital for a specific genetic disease test or for examination. If there is a genetic disease result for that patient, the hospital will send the patient's information and the genetic disease result to the *GenChain*. Even if it is positive or negative. Patients will not get any interaction with *GenChain* in this step. Only Hospital and *GenChain* will have a transaction.

2) GenChain Sends NationalID Of Patient And Request Patient's Family Number: When the hospital sends a message to *GenChain*, the smart contract will be triggered before the block is created. *GenChain* will send a patient's national ID to the government's General Directorate of Population and Citizenship Affairs.

3) GenChain Sends Family Number: They will send the family number of that national ID to the *GenChain*. After receiving the family number, *GenChain* will create the block of that patient. You can see the block in Figure 6. It will look to the network for that specific family number. If there is a family tree that we defined before in the network, these credentials of the patient will be created or updated (it depends on the patient's register). If there is no family tree for that family number, *GenChain* will create it.

4) Patient Requests The Possibility Of The Genetic Disease: Patient will send a request to the *GenChain* for a specific genetic disease. The only thing a patient needs to send is his/her national ID. Then *GenChain* will calculate the possibility of the genetic disease.

5) GenChain Generate Result And Sends It To Patient: After calculating the possibility of the genetic disease prediction, *GenChain* will send only the result of the calculation to the patient.

B. Protocols

In the proposed system, there are 3 protocols which are Genetic Disease Protocol, Prediction Result Protocol and Family Tree Protocol as seen in Figure 4.

Genetic Disease Protocol is used between hospitals and *GenChain*. Regardless of positive or negative, the patient's genetic disease result is shared with *GenChain*.

Family Tree Protocol takes place between the government office and *GenChain*. *GenChain* receives family number information from the government office with the help of national ID number in order to use it in its calculations.

Prediction Result Protocol takes place between the user and *GenChain* who want to learn the genetic disease rate. The user

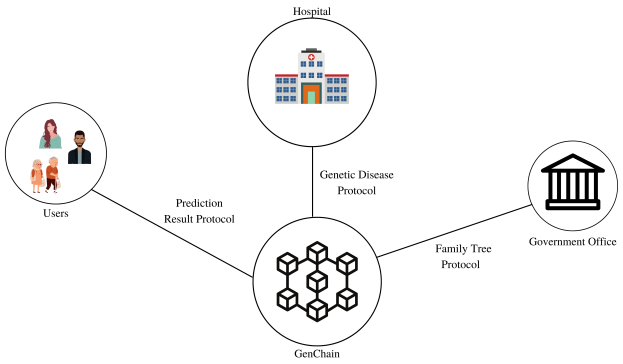

Fig. 4. Protocols

makes a request to see the genetic disease rate and the result is shared by *GenChain*.

1) Genetic Disease Protocol: This protocol takes place between the hospital and *GenChain* as seen in Figure 5. First hospital will pick a nonce. Then it will transfer a message which contains the signature, nonce and national ID of the patient. This message is encrypted with the *GenChain* public key. After receiving this message, *GenChain* will decrypt this message with its private key. *GenChain* needs to check that a patient's family tree exists or not in the network. If there is a family tree in the network, then it will collect the key values for Homomorphic Encryption, if not necessary key values will be generated.

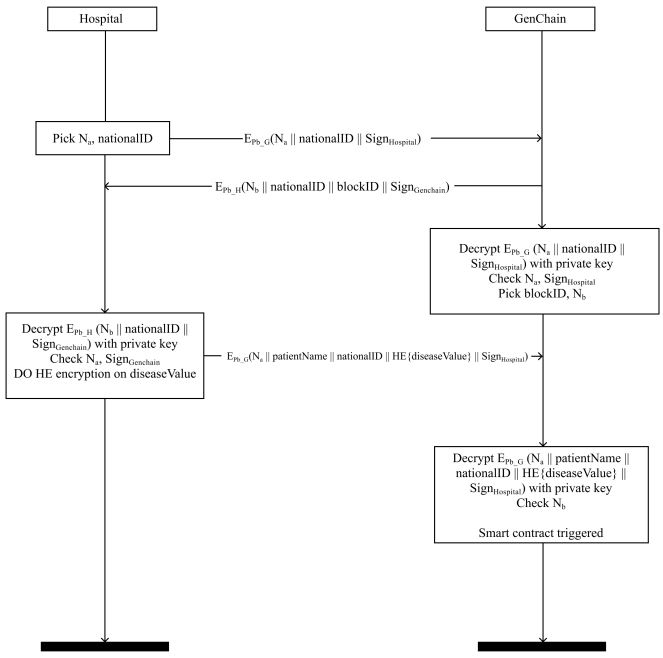

Fig. 5. Genetic Disease Protocol

Key generation works as follows:

1) Pick two large prime numbers p and q, randomly and independently. Confirm that gcd(pq, (p1)(q1)) is 1. If not, start again. (gcd = Greatest Common Divisor)
2) Compute n = pq
3) Define function L(x) = x1 / n
4) Compute as lcm(p1, q1). (lcm = Least Common Multiple)
5) Pick a random integer g in the set Zn2 (integers between 1 and n2)
6) Calculate the modular multiplicative inverse =(L(g mod n2))1 mod n. If does not exist, start again from step 1
7) The public key is (n,g). Use this for encryption
8) The private key is . Use this for decryption

After picking nonce, *GenChain* transmits a message containing signature , nonce and blockID which has got key values , and this message is encrypted with the Hospital's public key. Hospital will decrypt this message with it's private key then do a Homomorphic Encryption with the given key values on disease result. Encryption works as follows:

Encryption can work for any m in the range $0 \leq m < n$:

1) Pick a random number r in the range $0 < r < n$
2) Compute ciphertext c = gm $\times rnmodn2$

Then the Hospital will pick nonce again and transmit a message containing a sign of the *GenChain*, nonce, name of the patient, national ID of the patient and blockID. Next, the hospital will fetch the key value information from the *GenChain* and with those key values it will encrypt disease value. Important part is that we only use Homomorphic Encryption only on disease value. Then we will encrypt this message with the public key of *GenChain*. After receiving this message, *GenChain* will decrypt this message with its private key and a smart contract will be triggered. The reason is that we need more information about patients that hospitals don't have. After successful completion Smart Contract will start the Family Tree Protocol.

2) Family Tree Protocol: This protocol takes place between the government office and *GenChain* as seen in the Figure 6.

RSA encryption is used for encryption. *GenChain* will pick a nonce, then transmit a message that contains the signature , nonce and national ID of the patient. This message will be encrypted with the public key of the Government Office. After receiving this message, the Government Office will decrypt the message with its private key. Then it finds the family number corresponding to that national ID number. After finding this family number it will send a message to the *GenChain* that contains signature, nonce and familyNumber and this message will be encrypted with the public key of *GenChain*. *GenChain* will decrypt this message with its private key and add this family number of the patient to the block. We provide mutual authentication with signatures and confidentiality with encryption. We also prevent replay attacks with nonces.

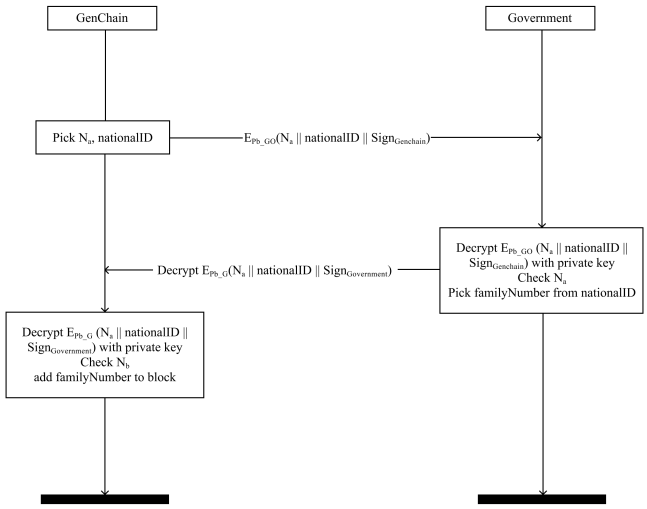

Fig. 6. Family Tree Protocol

3) Prediction Result Protocol: In Figure 7, RSA encryption is used for encryption.User will pick a nonce, then transmit a message that contains the signature, nonce and national ID of the patient. This message will be encrypted with the public key of the *GenChain*. After receiving this message, the *GenChain* will decrypt the message with its private key. Smart-Contracts will be triggered and it will find the block corresponding to the patient's national ID. Then calculate disase probability. With signatures and encryption, we enable mutual authentication and confidentiality. We also use nonces to avoid replay attacks.

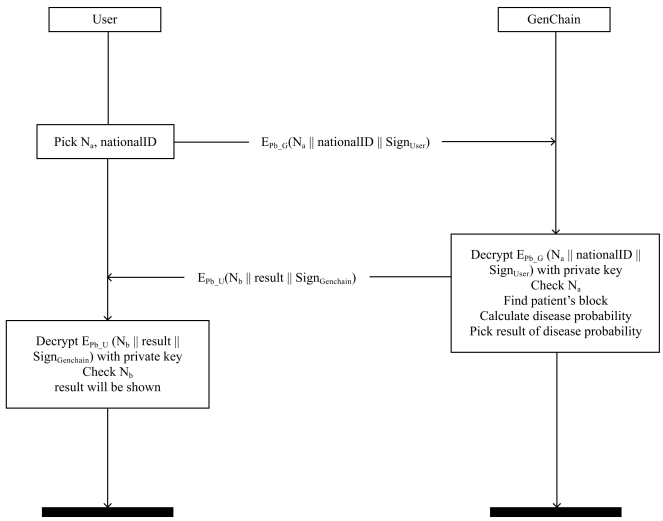

Fig. 7. Prediction Result Protocol

C. Disease Probability Calculations

The homomorphic properties used in the presented encryption schemes are as follows.

Addition of two ciphertexts: When two ciphertexts are multiplied, the result decrypts to the sum of their plaintexts:

$$D_{priv}(E_{pub}(m_1) \times E_{pub}(m_2) \bmod n^2) = m_1 + m_2 \bmod n$$

Multiplication of a ciphertext by a plaintext: When a ciphertext is raised to the power of a plaintext, the result decrypts to the product of the two plaintexts:

$$D_{priv}(E_{pub}(m1)^{m2} \bmod n^2) = m_1 \times m_2 \bmod n$$

In order explain how the calculation of the percentage for a genetic disease work in a persons family tree in the *GenChain* an example will be given below. The calculation can be done by starting with applying the RSA encryption, if we choose p as 17 and q as 19 n is calculated as 323. Result is 0 meaning they are not sick and the encrypted value that represents the sickness status is 45793. Percentage of the disease is assumed to be 25% and the index of the specific disease is 2.

If we assume that a father is not sick and he is a relative level 1 and apply the homomorphic multiplication to 45793, the number above, which represents that he is not sick so the it is the encrypted value of 0 and the percentage we gained according to his level, 25%, our calculation is:

- Compute $c_{product} = 45793^{25} = 43797 \bmod 323^2$

Later we continue our calculation with homomorphic addition of 43797 and 45793;

- Compute Result = $43797 \times 45793 = 79654 \bmod 323^2$

If we decrypt the result we get 0 which shows that the father is not sick.

Then we move to the mother whose relative level is also 1 and who is assumed to be sick on our case so her sickness status is 1 and its encrypted value is 1158. When we apply homomorphic encryption to the encrypted number 1158 and the percentage of heredity coming from her relative level %25 we do the calculation:

- Compute $c_{product} = 1158^{25} = 10202 \bmod 323_2$

After that we move on to the Homomorphic Addition of 10202 which is coming from her homomorphic multiplication and 79654 coming from the fathers result and do the calculations:

- Compute Result = $10202 \times 79654 = 11527 \bmod 323^2$

If we decrypt the result we have got here we get 25 which means the mother is sick and she has a %25 chance to transfer the disease to her offspring.

After, if we move onto the aunt on the family tree we get the relativity level of 2 which is equivalent to %12 transfer possibility this time and in our case she is not sick which is 0 and encrypted as 45793. If we do the same calculations with her data:

- Compute $c_{product} = 45793^{12} = 43388 \bmod 323_2$

Just as the above steps moving on the homomorphic addition of 43388 and 11527 (coming from the mothers calculation):

- Compute Result = $43388 \times 11527 = 84579 \bmod 323_2$

When we decrypt her result we get 25 which means her niece still has the %25 chance of getting sick we can see that percentage haven't increase so we can obtain that the aunt is not sick.

Lastly we go through the grandfathers data, his relative level is 2 so his percentage of heritage is %12 too and he is sick so sickness value is 1 decrypted and 1158 encrypted. Applying the homomorphic multiplication:

- Compute $c_{product} = 1158^{12} = 72011 \bmod 323^2$

Moving on to the homomorphic addition with the value we gained from multiplication and aunts result:

- Compute Result = $72011 \times 84579 = 100007 \bmod 323^2$

When the result is decrypted we have gained %37, which we can interpret that the grandfather is sick and since we come to the end of the family tree according to the information we have the persons chance of having this genetic disease is %38.

After finding the result, *GenChain* will send a message to the User that contains signature, nonce and result and this message will be encrypted with the public key of User. User will decrypt this message with its private key and can see the result which the *GenChain* calculated with high privacy preservation.

1) Create Asset: Create asset function issues a new asset to the *GenChain*. Using the identity and family information of the patient, it is questioned whether they exist in *GenChain*. In case an asset exists in *GenChain* it is returned. Otherwise, new Pailler keys are created and assigned to a new asset with the provided identity and family information. After all of the disease information are encrypted and assigned to the asset, asset is saved in the ledger.

Algorithm 1 Create Asset

$pName \leftarrow PatientName$
$pID \leftarrow PatientNationalID$
$pFID \leftarrow PatientFamilyID$
$pDiseases \leftarrow getDiseases(pID)$
if $asset$ exists **then**
    return already existing asset
else
    $patient \leftarrow Patient(pName, pID, pFID, pDiseases)$
    $paillerProps \leftarrow generateKey(pFID)$
    for diseases in pDiseases **do**
        $d \leftarrow paillerProps.encryption(diseases)$
    end for
end if
$ledger \leftarrow patient$

2) Change Asset: To fetch the patient's information from the *GenChain*, a read asset function is called with the patient identity and family information. A search for the patient is done in the ledger, and if successful requested patient's information is retrieved.

Algorithm 2 Change Asset

$pID \leftarrow PatientNationalID$
$d \leftarrow PatientDisease$
if $asset$ doesn't exists **then**
    return error
else
    $patient \leftarrow getPatient(pID)$
    $patient.d \leftarrow changePatientDiseaseValue(d)$
    $ledger \leftarrow Patient$
    return Patient
end if

3) Calculate Asset: Calculation of the disease probability is done by assessing the family tree of the requested patient. Provided identity of an asset, according family members with encrypted diseases values are fetched and Homomorphic Calculation is done for each of the family members.

Algorithm 3 Calculate Asset

$pID \leftarrow PatientNationalID$
$dIndex \leftarrow DiseaseIndex$
$patient \leftarrow fetchFromLedger(pID)$
$diseases \leftarrow fetchDiseasesFromTable()$
$diseaseProbability \leftarrow diseases[dIndex]$
$familyMemberLevel \leftarrow 1$
$sum \leftarrow paillerEncryption(0)$
for member in patient.familyMembers **do**
    $father \leftarrow member.father$
    $fatherDiseaseValue \leftarrow father.getDiseaseValue()$
    $mother \leftarrow member.mother$
    $motherDiseaseValue \leftarrow mother.getDiseaseValue()$
    $result1 \leftarrow fatherDiseaseValue/familyMemberLevel$
    $result2 \leftarrow motherDiseaseValue/familyMemberLevel$
    $sum \leftarrow paillerAddition(result1, result2)$
    $familyMemberLevel++$
end for
return sum

The workflow of the our implementation described by order in the below

1) Genchain creates asset for the patient.
2) All of the available diseases are created and assigned as false (doesn't experienced) for patient.
3) Patient goes to hospital and gets an illness. This illness is being updated for patient's disease list is being sent back to ledger.
4) Patient requests a disease calculation by assessing his/her family tree. Using calculate asset function, all of the patient's family members are fetched from the ledger based on the patient's national id, and using paillier encryption algorithm, disease rate is calculated level by level (1 being mother and father, 2 being grandparents etc.) for the patient.

5) Calculated result is sent back to patient and patient's request is completed.
6) Patient can decrypt the provided result with his/her private key and see the estimated disease rate.

V. Performance Analysis

In this section, the performance of the proposed architectural framework is tested and examined using benchmark and assessment measures. Latency, throughput, and other factors are used to do performance analysis for various instances. The Hyperledger caliper is a benchmark tool used for analyzing blockchain-based apps developed over the network. It can handle a variety of hyperledger platforms, including fabric, Indy, composer, sawtooth, and Iroha, among others. The caliper is used in this article to verify and execute the performance of the framework. For the framework's performance evaluation, many parameters such as latency, throughput, successful transaction count, and memory usage are measured. The performance analysis done with 2 organizations that has 1 peer (2org1peer) and 1 organization that has 1 peer(1org1peer). *GenChain*'s functions response time and memory usage are shown in the Table I

TABLE I
RESPONSE TIME AND MEMORY USAGE

Transaction	*Response Time*	*Memory Usage*
Create Asset	2.4 second	29.2 KB
Calculate Asset	0.3 second	9.7 KB
Change Asset	2.0 second	10 KB

Create Asset function and Change Asset function takes much more time other than the Calculate Asset. Since asset creation and changing requires a much more search, the response time

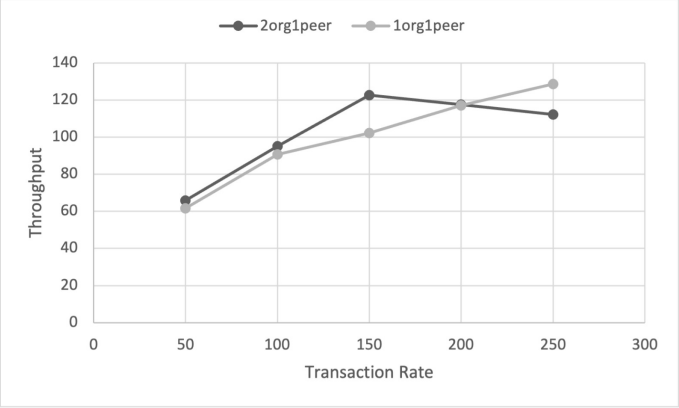

Fig. 8. Throughput with varying transaction rate

Figure 8 shows the throughput against transaction rate. Throughput of 1org1peer network is measured as highest at 130, whereas it is very low for 2org1peer (120). The gap between 1org1peer and 2org1peer in terms of throughput will grown as shown after the 200 transaction rate.

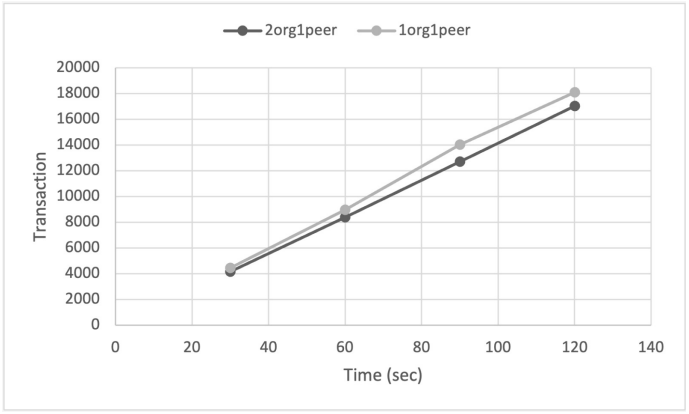

Fig. 9. Required time to complete the transition

Figure 9 shows the time under various rounds for successfully completing the transaction. 1org1peer takes about 62 s to reach 10000 transactions where, at the same time, 2org1peer completes 9000 transactions. So it is observed that the transaction time often increases with growth in the organization and peers. Successful transaction rate is important for a network like *GenChain* because this network working with patients and hospitals. Hence the transaction count per day will be larger and larger when we add more peer and organizations.

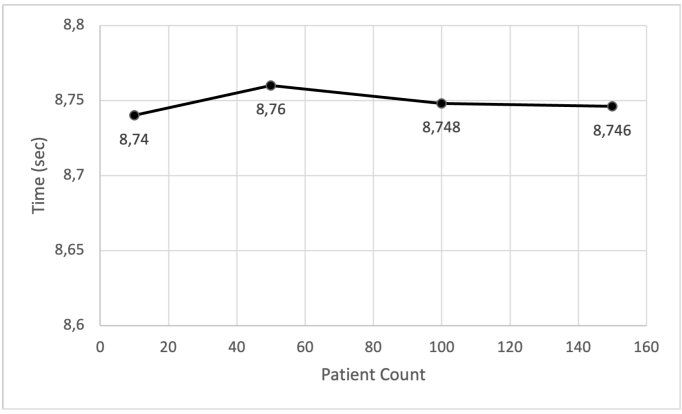

Fig. 10. Genetic Disease Prediction Calculation with varying Patient Count

Figure 10 depicts the *GenChain*'s genetic disease calculation response time after each different patient count. The response times are slightly same and all of the prediction results are correct for every asset count. *GenChain*'s genetic disease prediction algorithm works slightly same when we have a different patient count.

VI. Disscusion

While offering genetic disease prediction by family assessment, *GenChain*'s main goal is to offer a secure system keeping users' privacy and being accessible. Below provided subsections addresses these concerns. The latency of the calculation doesn't make any huge difference when we add

the new assets to the network. This is a vital advantage since calculate asset function will be called several times

A. Security

GenChain is a permission blockchain and all the peers are in the network after an approval step. These peers can be hospitals or governments. It works based on a decentralized network topology, which does not have a single point failure. This property is very important for our application because health data is important so we shouldn't lose or manipulate the data.

B. Privacy

Thanks to our homomorphic encryption, all the disease values are stored in an encrypted form. Even in the disease calculation we don't need to decrypt the data.With this way, all patients' private health data are kept private.

C. Accessibility

Our solution offers high speed while securing the patients private data. All of the calculations that are done for obtaining the final result are assured to be fully encrypted. By using blockchain technology, we offer a decentralized solution which removes the necessity of storing data in one place by guaranteeing data availability.

VII. Conclusion

This article aims to offer secure and private genetic disease calculation with the use of the blockchain technology while ensuring the certainty of the privacy and security of these calculations. The Hyperledger Fabric is used for the implementation. Hyperledger Caliper is used to evaluate the proposed model's performance and resource utilization. The performance results obtained for parameters such as successful transaction count, throughput, memory utilization, and response time, as well as their key findings, show that the genetic disease calculation problem is solved within the 8.5-8.8 seconds. The response time of the calculation is not affected much when we increase the asset count in the network. It is also observed that it exceeded 10000 successful transactions within 62 seconds. Even if we add more organizations, the time difference is not a noticeable difference. Moreover, the people who share their health data can be sure about the security and anonymity of their health data. With this project we have achieved a secure and reliable blockchain system which allows patients to store genetic diseases. Even though this is a small part of human health, blockchain technology has a huge way to improve and get into our lives. We believe that in the name of security it is possible to keep sensitive information such as health data through blockchain and in the future this technology will be used widely.

References

[1] L. Ismail, H. Materwala, and S. Zeadally, "Lightweight blockchain for healthcare," *IEEE Access*, vol. 7, pp. 149 935–149 951, 2019.

[2] J. Xu, K. Xue, S. Li, H. Tian, J. Hong, P. Hong, and N. Yu, "Healthchain: A blockchain-based privacy preserving scheme for large-scale health data," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8770–8781, 2019.

[3] Y. Cao, Y. Li, Y. Sun, and S. Wang, "Decentralized group signature scheme based on blockchain," in *2019 International Conference on Communications, Information System and Computer Engineering (CISCE)*, 2019, pp. 566–569.

[4] M. A. Bazel, F. Mohammed, and M. Ahmed, "Blockchain technology in healthcare big data management: Benefits, applications and challenges," in *2021 1st International Conference on Emerging Smart Technologies and Applications (eSmarTA)*, 2021, pp. 1–8.

[5] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *2017 IEEE International Congress on Big Data (BigData Congress)*, 2017, pp. 557–564.

[6] X. Liu, Z. Wang, C. Jin, F. Li, and G. Li, "A blockchain-based medical data sharing and protection scheme," *IEEE Access*, vol. 7, pp. 118 943–118 953, 2019.

[7] J. da Silva Fraga and D. Powell, "A fault- and intrusion- tolerant file system," 1985.

[8] A. N. Bessani, "From byzantine fault tolerance to intrusion tolerance (a position paper)," in *2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 2011, pp. 15–18.

[9] S. S. Sathya, P. Vepakomma, R. Raskar, R. Ramachandra, and S. Bhattacharya, "A review of homomorphic encryption libraries for secure computation," *ArXiv*, vol. abs/1812.02428, 2018.

[10] M. O'Keeffe, "The paillier cryptosystem," *Mathematics Department April*, vol. 18, pp. 1–16, 2008.

[11] A. R. Rajput, Q. Li, M. Taleby Ahvanooey, and I. Masood, "Eacms: Emergency access control management system for personal health record based on blockchain," *IEEE Access*, vol. 7, pp. 84 304–84 317, 2019.

[12] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukoli'c, S. W. Cocco, and J. Yellick, "Hyperledger fabric: A distributed operating system for permissioned blockchains." New York, NY, USA: Association for Computing Machinery, 2018.

[13] G. Sawant and V. Bharadi, "Permission blockchain based smart contract utilizing biometric authentication as a service: A future trend," in *2020 International Conference on Convergence to Digital World - Quo Vadis (ICCDW)*, 2020, pp. 1–4.

[14] V. Aleksieva, H. Valchanov, and A. Huliyan, "Smart contracts based on private and public blockchains for the purpose of insurance services," in *2020 International Conference Automatics and Informatics (ICAI)*, 2020, pp. 1–4.

[15] A. P. Singh, N. R. Pradhan, A. K. Luhach, S. Agnihotri, N. Z. Jhanjhi, S. Verma, U. Ghosh, D. S. Roy *et al.*, "A novel patient-centric architectural framework for blockchain-enabled healthcare applications," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5779–5789, 2020.

[16] A. C. F. Spengler and P. S. L. d. Souza, "The impact of using couchdb on hyperledger fabric performance for heterogeneous medical data storage," in *2021 XLVII Latin American Computing Conference (CLEI)*, 2021, pp. 1–10.

[17] M. Ghadamyari and S. Samet, "Privacy-preserving statistical analysis of health data using paillier homomorphic encryption and permissioned blockchain," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 5474–5479.

[18] M. Kuzlu, M. Pipattanasomporn, L. Gurses, and S. Rahman, "Performance analysis of a hyperledger fabric blockchain framework: Throughput, latency and scalability," in *2019 IEEE International Conference on Blockchain (Blockchain)*, 2019, pp. 536–540.

[19] M. Dabbagh, M. Kakavand, M. Tahir, and A. Amphawan, "Performance analysis of blockchain platforms: Empirical evaluation of hyperledger fabric and ethereum," in *2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, 2020, pp. 1–6.