

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research



UNIVERSITY OF ECHAHID HAMMA LAKHDAR - EL OUED

FACULTY OF EXACT SCIENCES
Computer Science Department



Thesis

Submitted in partial fulfillment of the requirements for the degree of

ACADEMIC MASTER

Field: **Mathematics and Computer Science**

Option: **Computer Science**

Specialty: **Distributed Systems and Artificial Intelligence**

Presented by :

- **Bachir Dou**
- **Aymen Bebbi**

Title

Anonymization of data in the IoT-Blockchain environment

Case Study: Electronic Health Record (EHR) in healthcare systems.

Defended on June 21st 2021

Examination Committee :

M. Mounir Beggas	MCA	Chair
M. Kaled Soltani	MAA	Examinator
M. Saci Medileh	MAA	Supervisor

ACKNOWLEDGEMENT

The first thanks is to Allah Almighty, then to my parents for all their efforts since my birth until these moments. You are everything I love you most in Allah. We also dedicate this humble work to the soul of our friend (Ahmed Roun Allah), may Allah have mercy on him. I am also pleased to extend my thanks to everyone who advised me, guided me, directed me, or contributed to the preparation of this research by connecting me to the required references and sources at any stage of it, and I especially thank my honourable professor, Dr. My thanks are also directed to the administration of the Faculty of Exact Sciences at the University of (Martyr Kama Lakhdar) for their good provision and facilitation of services to students and their assistance in all matters that would give them a comfortable space to study and seek knowledge in safety and order.

Aymen Bebbi

Bachir Dou

ABSTRACT

The critical nature of Electronic Health Records (EHR), and the technological advancement in the field of data privacy required researchers to come up with new strategies to handle EHRs in the healthcare industry. Many researchers have been investigating the possibility of integrating Blockchain technology to cloud-based EHR system architecture to increase the security of the health information within the system. However, EHR systems require more security to protect it for different attacks like Sybil Attacks and Linking Attacks. In this thesis, we propose a new data preservation mechanism called Anonymity Preservation Mechanism (APM), which provides another level of protection to health information of EHR systems by implementing a method of a multilayer cryptography to patient records. Moreover, this method splits patient records into two categories; identifiable information, and none identifiable information, implementing complicated hashing mechanism to the identifiable information which results a pseudo identities that do not link to the patient real identity, the none identifiable information go through an AES encryption method which uses a secret key to produce a secure ciphertext of the original data. Additionally, the generated outputs from the previous operations enter another phase of encryption by RSA cryptosystem, this multilayer encryption process produce a secure anonymized patient record. Furthermore, we integrated our proposed APM to our enhanced Blockchain-based EHR management system architecture. Lastly, we provided a prototype of our proposed APM and presented its implementation results in the system.

Keywords: Electronic Health Record (EHR), Privacy preservation, Privacy issues of Blockchain-based EHR, Blockchain.

ملخص

فرضت الطبيعة الخصوصية للسجلات الصحية الإلكترونية (EHR) التي تحتوي على معلومات صحية حساسة بالإضافة إلى التقدم التكنولوجي في مجال حماية البيانات على العديد من الباحثين التقصي عن استراتيجيات جديدة لحماية السجلات الصحية الإلكترونية في مجال الرعاية الصحية. تم اقتراح عدة مخططات مستجدة لأنظمة إدارة السجلات الصحية الإلكترونية القائم على تكنولوجيا السحابة الإلكترونية (Cloud Technology) و تكنولوجيا البلوكشين (Blockchain Technology) مما يعزز من حماية المعلومات الصحية داخل هذه الأنظمة. ومع هذا فإن هذه الأنظمة غير محمية بشكل كامل و عرضة للعديد من الهجمات الإلكترونية كهجمات (Sybil Attacks) و (Linking Attacks). في هذا المشروع, قمنا بإقتراح آلية جديدة للحفاظ على خصوصية وأمان البيانات الصحية (APM), هذه الآلية توفر طبقة إضافية من الحماية للمعلومات الصحية لأنظمة (EHR) من خلال تنفيذ خوارزمية تشفير متعددة الطبقات (Multilayer Cryptography) لسجلات المرضى. أولاً, تقوم هذه الخوارزمية بتقسيم بيانات المرضى إلى فئتين أساسيتين؛ بيانات التعريف الشخصية (identifiable information) و بيانات الغير شخصية (non identifiable information), تقوم هذه الخوارزمية بتشفير بيانات التعريف الشخصية لينتج عن هذه العملية بيانات مشفرة تعطي هوية مستعارة للمريض, كل هذا باستخدام الية تشفير (Bcrypt). يتم تطبيق تقنية (AES) على البيانات الغير شخصية و التي تستخدم مفتاح تشفير سري (SK) لتوليد بيانات مشفرة من البيانات الأصلية. بالإضافة إلى ذلك, تدخل المخرجات الناتجة عن عمليات التشفير السابقة في مرحلة أخرى من التشفير بواسطة الية التشفير (RSA), و التي تقوم بعملية تشفير متعددة الطبقات ينتج عنها سجل بيانات مرضى آمن. إضافة إلى ذلك, قمنا بدمج تقنية (APM) المقترحة في آلية إدارة السجلات الصحية الإلكترونية. أخيراً, قمنا بتقديم نموذج تنفيذ أولي لـ (APM) المقترحة بالإضافة إلى إستعراض نتائج تنفيذها.

كلمات مفتاحية: السجلات الصحية الإلكترونية (EHR), تقنيات حماية الخصوصية, مشاكل

الخصوصية في السجلات الصحية الإلكترونية قائم على البلوكشين, البلوكشين.

TABLE OF CONTENT

ACKNOWLEDGEMENT	II
ABSTRACT	III
Keywords	III
ملخص	IV
كلمات مفتاحية	IV
TABLE OF CONTENT	V
LIST OF FIGURES	VIII
LIST OF TABLES	IX
GENERAL INTRODUCTION	X
OUTLINE OF THE PAPER	XII

CHAPTER I : State of the Art

1 Introduction	1
2 Electronic Health Record Systems:	1
2.1 Features & Functionalities of EHR Systems:	1
2.2 Cloud-Based EHR Systems:	2
2.3 Workflow of Cloud-Based EHR Systems:	2
2.4 Security Risks in Cloud-Based EHR Systems:	3
2.5 Security Requirement for EHR Systems:	3
3 Data Preservation Strategies:	4
3.1 Encryption:	4
3.2 Hashing:	5
3.3 Anonymization:	6
3.4 Differential Privacy:	8
4 Internet Of Things:	9
4.1 Implementations Of the Internet Of Things:	9
4.1.1 Wearables:	9
4.1.2 Finance:	9
4.1.3 Smart cities:	10
4.1.4 HealthCare:	10
4.2 Security and privacy Principles in IoT:	11
5 Blockchain Technology:	12
5.1 Types of Blockchain:	12
5.1.1 Private Blockchain:	12
5.1.2 Public Blockchain:	13
5.2 Working Phenomenal Of Blockchain:	14

5.3	Characteristics of Blockchain:	14
5.4	Challenges of Using Blockchain in IoT Systems:	15
5.4.1	Sybil Attacks:.....	16
5.4.2	Linking Attacks:.....	16
5.4.3	Address reuse:.....	16
6	Challenges & Solutions for Integrating Blockchain to EHR Systems:	17
7	Related Work:.....	17
8	Conclusion:	19
CHAPTER II : Architecture & operation of the proposed model		
1	Introduction:.....	20
2	Basic Concepts and Preliminaries:	20
2.1	Data Preservation Mechanisms:	20
2.2	Smart Contract:	21
2.3	Fetching Query:	21
2.4	Patient Record:.....	22
2.4.1	Anonymized Patient Record:	22
2.5	Transaction:	23
2.5.1	Primary Transaction (PT):	23
2.5.2	Query Transaction (QT):.....	24
2.5.3	Final Transaction (FT):	24
2.6	Transaction Template:	24
3	Proposed System architecture:.....	24
3.1	Anonymity Preservation Mechanism (APM):.....	25
3.2	Network Server:.....	27
3.3	Private Blockchain:.....	29
3.4	Cloud-based EHR Management System:.....	29
4	System Workflow:.....	29
5	Conclusion:	31
CHAPTER III : Implementation & Results		
1	Introduction:.....	32
2	Implementation Scenarios:.....	32
2.1	Case Scenarios:.....	32
2.2	Scenarios Details:.....	33
3	Implementation Details:	33
3.1	Client Application:.....	34
3.2	Anonymity Preservation Mechanism (APM):.....	37
3.3	Network Server:.....	40

3.4	EHR Cloud Database:	41
4	Results:.....	41
4.1	Initial Patient Record (1):	41
4.2	Initial Patient Record (2):	42
4.3	Anonymized Patient Record (1):.....	42
4.4	Anonymized Patient Record (2):.....	43
5	Conclusion:	43
	GENERAL CONCLUSION	44
	FUTURE WORK.....	45
	REFERENCES.....	46

LIST OF FIGURES

Figure 1: Architecture of Cloud-Based Electronic Health Record System	2
Figure 2: Process of encryption and decryption using Symmetric and Asymmetric cryptography.....	5
Figure 3: Implementation of Internet of Things in different fields.....	11
Figure 4: Simplified structure of blocks.	14
Figure 5: Architecture of a multi-layered anonymized patient record.....	23
Figure 6: Proposed System Architecture of cloud-based Electronic Health Record.....	25
Figure 7: Internal Architecture of the Network Server.	28
Figure 8: Sequence Diagram showcases the process of storing patient records in the system.	30
Figure 9: Sequence Diagram showcases the process of fetching patient records from the system.....	31
Figure 10: A general layout description of a client application page.	34
Figure 11: presentation of the “New Patient Record” content.	35
Figure 12: presentation of the “Search for Record” content.....	35
Figure 13: presentation of the “Add a Patient” content.....	36
Figure 14: presentation of the “Export a Dataset” content.....	36

LIST OF TABLES

Table 1: <i>a dataset of patient records in a hospital.</i>	6
Table 2: <i>the dataset of the patients after implementing K-anonymity of $K=3$.</i>	7
Table 3: <i>a dataset of customer spending in a bank.</i>	8
Table 4: <i>a dataset of customer spending in a bank.</i>	8
Table 5: <i>the first type of Initial Patient Record.</i>	41
Table 6: <i>the second type of Initial Patient Record.</i>	42
Table 7: <i>the first type of Anonymized Patient Record.</i>	42
Table 8: <i>the second type of Anonymized Patient Record.</i>	43

GENERAL INTRODUCTION

With the rapid development of information technology and cloud computing, and with the benefits and features that comes with its implementation, more and more people and organization have started integrating these technologies into different aspects of our everyday life providing new electronic solutions to critical problems that ones were known to be impossible to solve, and simplifying complicated tasks that once were hard to complete[12]. Medical institutions are among the fields that rely on technology to manage their data and handle their sectors. In general, Healthcare has become increasingly dependent on information technology to computerize almost all aspects of patient care. Integrating these technologies into the healthcare environment causes a shift in how Healthcare management systems perform their work, provide convenient solutions, and bring new challenges to managing medical information [11].

Electronic Health Record (EHR) is one of the most popular solutions for medical institutions to manage patient data digitally and handle the different operations that occur on patient data .EHR is an electronic database that contains their medical information like medical images, medical treatment, medications, experience reports, family history of genetic disease, etc. Therefore, patient information is considered critical data that needs to be managed in private and secure manure to prevent leakage. On the other hand, with the help of EHR, patient records can provide accurate analysis to researchers to prevent diseases and improve the cure rate [12].

In the early days of this technology, EHR was expensive to acquire and hard to manage; sins needed high costs medical data center and professional technical support. However, with the emerging cloud technology and all the features that it came with, managing EHR has become easier than ever. Cloud technology can provide a cloud computing system with a large storage capacity, which has many features like fast transmission, good sharing, storage capacity, low cost, easy access, dynamic association. These features help us build an integrated EHR System that handles patient records in a private, secure way and efficiently protects it from significant threats like abuse, leakage, loss, or theft. With all of that been said, cloud-based EHR systems can still be venerable to severe threats due to the centralized characteristics of cloud-based systems [14].

Blockchain technology is a decentralized peer-to-peer network first introduced in the scientific paper of the Bitcoin cryptocurrency. Published in 2008 by an unknown author with

the pseudo name Satoshi Nakamoto. After Bitcoin's success, people started showing interest in Blockchain technology and its interrogation on other fields aside from cryptocurrency. In Healthcare, Blockchain is treated as a distributed ledger to store health records for sharing, exchanging, or alternative functions among stakeholders [12]. In a Blockchain-based EHR management system, all the data associated to patients are converted to transactions and stored in the distributed ledger offered by a Blockchain network. Each transaction is evaluated by a group of participants, known as miners, before it get stored in the distributed ledger. No unauthorized entity can change the data in a Blockchain network. A key concept of blockchain, smart contract, empowers trustless features among different EHR management systems. A smart contract involves a computer program that contains a set of agreements and principles. All of the participants in the network must follow the set of agreements and principles. Hence, no trusted third party is required to store data in the blockchain [19].

Integrating Blockchain technology into cloud-based EHR Systems can be challenging, keeping in mind the critical nature of patient data. Therefore, we pinpointed two main questions that we are going to answer throughout this thesis:

- i. How to successfully integrate Blockchain technology into cloud-based EHR systems?
- ii. How to provide privacy and security to patient data and protect it from leakage in a cloud-based EHR System?

We sum up our contribution in this thesis in two main points that we are going to discuss in details later:

- i. Propose an integrated architecture of cloud-based EHR Systems using Blockchain technology.
- ii. Propose a novel data preservation mechanism to protect the patient data anonymity and security in the proposed cloud-based EHR System.

OUTLINE OF THE THESIS

In the first chapter, we provide a theoretical foundation of the technologies and methods which we're going to need in our proposed project. In the second chapter, we present the architecture of the proposed system providing a detailed description of each of its components, and introduce our proposed anonymization mechanism. In the third and last chapter, showcase the implementation process of our anonymization mechanism prototype.

CHAPTER I

STATE OF THE ART

1 Introduction

With the increasing development of IoT technology and data preservation mechanisms and the rise of blockchain technology, new doors of improvement have opened for Electronic Health Record (EHR) systems and provide an excellent opportunity to accommodate e-Health systems in different scenarios effectively. On the other hand, new privacy challenges and hacking techniques have appeared. Creating a new threat for patient privacy and e-Health systems requires advanced development and integrated architecture to secure the patient information within the EHR System. In this chapter, we introduced EHR Systems, describing some of their features, functionalities, and security requirements. Furthermore, we presented some of the most commonly used data preservation strategies. We studied in detail blockchain technology and IoT technology providing characteristics, challenges, and solutions for integrating both of these technologies to provide a direction of improvement for EHR Systems.

2 Electronic Health Record Systems:

Electronic Health Record (EHR) is essentially a digital copy for patient data and information. EHR is a critical part of the medical field because it provides a real-time database of patient records and makes them available easily and instantly for authorized users such as doctors or administrators. Another similar term is Electronic Health Record Systems (EHR Systems), which go beyond storing patients' medical information and treatment histories. EHR Systems can provide all sorts of services and features to the user by implementing different methods and functionalities on patient data [14].

2.1 Features & Functionalities of EHR Systems:

- Contain a patient's medical history, diagnoses, medications, treatment plans, immunization dates, allergies, radiology images, and laboratory and test results
- Allow access to evidence-based tools that providers can use to make decisions about a patient's care
- Automate and streamline provider workflow
- Provide a secure system that protects patient information from leakage and tampering and make it resistant to all kinds of attacks
- Share health information with authorized parties like health organizations or labs for different kinds of medical purposes.

2.2 Cloud-Based EHR Systems:

Many Electronic Health Record systems are being used to manage patient data; one of the most reliable ways to handle sensitive information is Cloud Based EHR Systems, which use Cloud Computing Technology to manage patient information. In Figure 1, there's an example of an Architecture of a Cloud-Based Electronic Health Record System [13-14].

Cloud-Based EHR Systems come with many advantages and features to help manage patient information in the best way possible.

- **Accessibility of information.** It provides the ability to store and access any medical information easily and instantly by using a stable connection to the cloud.
- **Security and privacy.** Using cloud technology to manage medical information is one of the safest solutions as it comes with all different kinds of privacy preservation mechanisms to help protect medical information.

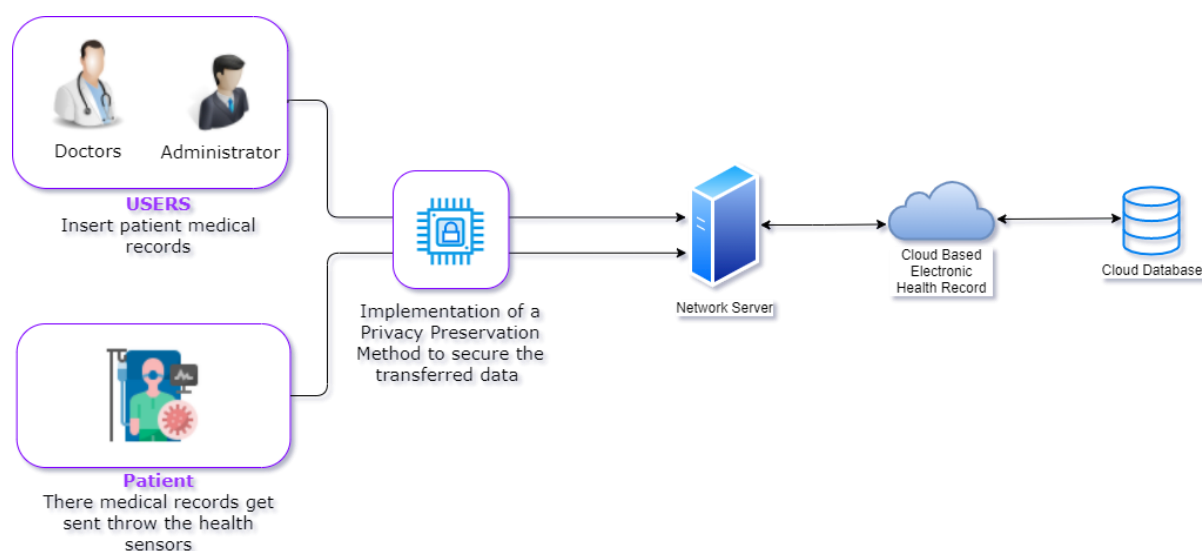


Figure 1: Architecture of Cloud-Based Electronic Health Record System

2.3 Workflow of Cloud-Based EHR Systems:

In cloud-based EHR systems, patient records follow a specific route through the network from the body sensors of the patient or from the client computers of users (doctors, administrator) back to being stored in the database of the cloud-based EHR. **Firstly**, there are two main ways of committing patient records in the cloud database, it either gets inserted from client computers by users, or it could get collected from the body sensors of the patient. **Then**, records get sent to the network server, which accumulates the record in organized datasets and performs specific privacy preservation methods to help secure the patient

information within the dataset records and prevent any leakage of data. **Finally**, the network server uploads the dataset to the cloud-based EHR, which stores it in its database. The organized and anonymized nature of the patient records stored in the dataset provides an infrastructure for many medical research activities and helps researchers perform studies and analysis the patient record [13].

2.4 Security Risks in Cloud-Based EHR Systems:

In spite of having several benefits of cloud-based EHR management systems, security is a critical concern. EHR in cloud-based management systems can be exposed to abuse, leakage, loss or theft. For example, EHRs can be deleted or tampered with by intruders to tamper treatments giving benefits to insurance companies or hiding medical malpractices. EHRs are closely related to health insurance. Dishonest health insurance service providers may hire hackers to delete or tamper the EHRs of patients to prove the existence of pre-existing health conditions. Medical malpractices such as misdiagnosis and delayed diagnosis are a few of the many reasons for medical insurance claims. In most cases, patients cannot prove the medical malpractices due to the mentioned issues. On the other hand, patients modify medical records to get financial advantages in spite of having some pre-existing medical conditions. Several countermeasures are proposed to provide security of EHRs using cryptographic techniques. Unfortunately, security threats remain a great concern due to the centralized characteristics of cloud-based systems[13-11].

2.5 Security Requirement for EHR Systems:

- **Immutability.** The EHR System should prevent any kind of manipulation of health information and ensures integrity to patient data. Hence, health information being shared should be an accurate representation of original information without any form of amendment or alteration.
- **Data privacy.** The privacy of patient information is a critical issue. Therefore the EHR System should put a great effort to protect these pieces of information from attacks, and that's by using different kinds of privacy preservation mechanisms to secure these kinds of data from adversaries.
- **Anonymity.** Patient real identity must be well protected from the public due to this information's personal and private nature, so patient anonymity should be a priority for the EHR System.
- **Access control.** The patient data should only be accessible by an authorized user and kept away from users who should not access it. Therefore, the EHR System must

control who gets access to patient data to prevent unauthorized users from accessing the data.

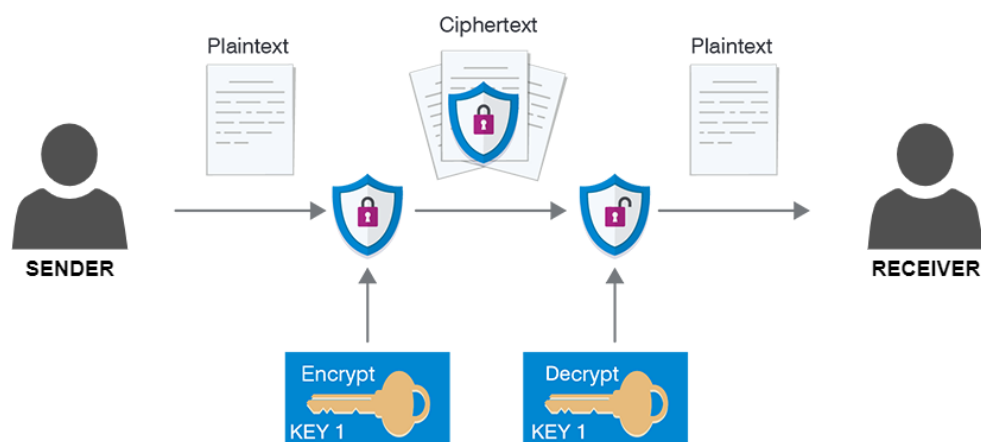
3 Data Preservation Strategies:

As mentioned in the previous section, there are many sensitive data and information that should be secured from numerous attacks. Any leakage of these kinds of information can cause significant damage to the owner of this information. Thus, any data management system should have some kind of privacy preservation mechanism that can provide protection to the data from adversaries. In this section, we provide some privacy preservation mechanisms that are the most used data management systems.

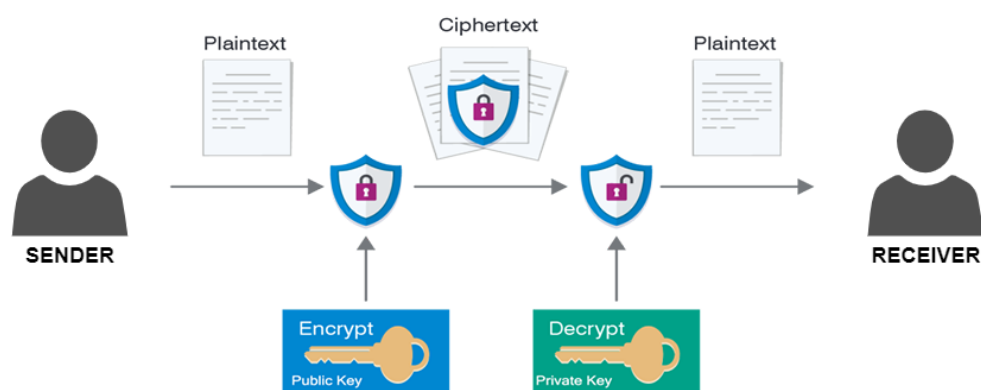
3.1 Encryption:

Encryption is one of the most commonly used strategies to provide privacy and security to the data and hide it from intruders. There are many types of encryption; one type is Symmetric Encryption, where all communication parties have the same secret key for encryption and decryption of the data. Another type of encryption is Asymmetric Encryption or also known as Public Key Encryption (PKE), in it there are two types keys: one type of key is a public key used to encrypt the data before sending it, and it meant to be visible for anyone to use, and the other type of key is a private key, and it used to decrypt the received data, and it should be kept private, a demonstration of the process of Symmetric and Asymmetric Encryption is presented in figure 2. It's worth mentioning that the information before encryption is called PlainText and after encryption is called CypherText. The phenomenal of encryption and decryption protect the data and keep it secure [14-1]. However, many other advanced encryption methods are being discussed by researchers to provide more privacy and security to the data.

Manly, three factors can be considered while calculating the effectiveness of an encryption method; **Computational load:** can be measured by the amount of computational power and resources put into the process of executing the encryption algorithm. **Key size:** The number of bits used to generate the encryption keys (Public/Private). **Size of band:** it matches the number of bits required to transmit a message after encoding or signing.



Symmetric Cryptography



Asymmetric Cryptography

Figure 2: Process of encryption and decryption using Symmetric and Asymmetric cryptography.

3.2 Hashing:

Hashing is the phenomenon of transforming a string of characters into a usually fixed-length value or key works as a unique signature of that string using a mathematical function [17]. When a hashing function receives a document or a file as an input, it generates a hash output that represents the original file. The message to be hashed is called input, the algorithm that performs hashing is called a hash function, and the output is called a hash value. Many formulas can be used to hash a message, but a cryptographic hash function needs to have certain qualities to be considered beneficial. **First**, the hashing process must be one-way, meaning the hash function cannot restore the original input based on the hash value. **Second**, each hash value or output must be unique; this means that it should be impossible to produce the same hash value entering different inputs. **Third**, hashing results should be

deterministic, which means that a given message should not generate different hash values, and therefore the same message must always produce the same hash value. **Forth**, the hashing speed is also an essential factor; the process of producing a hash value should be relatively fast. **Finally**, a hash function needs to be secure; it should be extremely difficult to determent the input based on the hash value, while the slightest change to inputs should generate a hugely different hash. This phenomenon is also known as the Avalanche Effect.

3.3 Anonymization:

Anonymization is a well-known method for providing personal privacy to the data by removing personal identifiable information from the data and ensuring the anonymity of the person's real identity behind the data. There are various advanced types of strategies being discussed by researchers to provide secure layers of anonymization [1].

One is called **k-anonymity**, which might be described as a 'hiding in the crowd' guarantee. If each individual is part of a larger group, then any of the records in this group could correspond to a single person. For k-anonymity to be achieved, there need to be at least k individuals in the dataset who share the set of attributes that might become identifying for each individual[15].

For more explanation, consider the example as shown in table 1; name, Postcode, Age, and Gender are attributes that could all be used to help narrow down the record to an individual; these are considered quasi-identifiers as they could be found in other data sources, disease is the sensitive attribute that we wish to study and which we assume the individual has an interest in keeping private.

Name	Postcode	Age	Gender	Disease
Ahmed	SW1 SE8	22	Male	Cardiovascular
Ali	SW1 WZ9	23	Male	Respiratory
Rima	NW10 V98	18	Female	No Illness
Abby	NW10 FE8	47	Female	Cancer
Marouane	E17 XZ9	42	Male	No Illness
Bachir	E17 KI6	56	Male	No Illness
Omar	E17 O2U	23	Male	Liver
Aymen	E17 L4M	29	Male	No Illness

Table 1: a dataset of patient records in a hospital.

The second table 2 shows the data anonymized to achieve k-anonymity of $k = 3$; this was achieved by generalizing some quasi-identifier attributes and redacting others. In this small example, the data has been distorted quite significantly, but the more significant the dataset, the less distortion is required to reach the desired level of k .

Name	Postcode	Age	Gender	Disease
*	SW1 *	22	Male	Cardiovascular
*	SW1 *	23	Male	Respiratory
*	NW10 *	18	Female	No Illness
*	NW10 *	47	Female	Cancer
*	E17 *	42	*	No Illness
*	E17 *	56	*	No Illness
*	E17 *	23	*	Liver
*	E17 *	29	*	No Illness

Table 2: the dataset of the patients after implementing K-anonymity of $K=3$.

Another type is **Pseudonymization** which is a technique used to de-identify data. When using pseudonymization, sensitive data fields are replaced with pseudonyms to hide the identity of the individuals. The consistency of this technique allows identical pseudonyms to be applied to the same individual throughout the dataset, which is very useful when it is necessary to link data collected at different times relating to the same data subject. Pseudonyms can also retain the original data structure so that the format is retained, which can be useful under some circumstances [16].

Let us take the example of a bank that wants to analyze customer spending patterns over the month of June to determine their high-value customers. In order to do this, they need to use the customer transaction dataset. By looking at the dataset below in table 3, you notice that it contains personally identifiable information such as names, account IDs, and emails.

S/NO	Name	Account ID	Email	Transaction Value	Transaction Date
1	Ahmed	AC4481245	ahmed@gmail.com	59.45	05/06/20
2	Ali	AC1114455	ali@gmail.com	12.50	07/06/20
3	Rima	AC1214445	rima@gmail.com	9.50	11/06/20
4	Ahmed	AC4481245	ahmed@gmail.com	52.50	13/06/20
5	Aymen	AC4545553	aymen@gmail.com	18.63	15/06/20
6	Bachir	AC5698523	bachir@gmail.com	63.65	15/06/20
7	Omar	AC7889633	omar@gmail.com	42.10	15/06/20
8	Ahmed	AC4481245	ahmed@gmail.com	34.50	18/06/20

Table 3: a dataset of customer spending in a bank.

Below in Table 4 is an example of the same dataset, only it has been de-identified using Pseudonymization.

S/NO	Name	Account ID	Email	Transaction Value	Transaction Date
1	DFJFSD	X321343T	idrshdy@gmail.com	59.45	05/06/20
2	LKGJSHF	E896ED5	dkiejnf@gmail.com	12.50	07/06/20
3	LGKKGJD	LM29HY9	qsdufk@gmail.com	9.50	11/06/20
4	FKDHWDD	UI962AZ	idrshdy @gmail.com	52.50	13/06/20
5	FMZICNZS	V15G8PL	qmsfkp@gmail.com	18.63	15/06/20
6	OSCNEPOE	TK35AN7	feojv@gmail.com	63.65	15/06/20
7	EODNEOF	FP8246C	fpzmdg@gmail.com	42.10	15/06/20
8	NVIRSLJFZ	X321343T	idrshdy @gmail.com	34.50	18/06/20

Table 4: a dataset of customer spending in a bank.

Customer names have been pseudonymized to a string of 7 random characters so that the original names are no longer visible. Account ID and Email fields have been pseudonymized consistently; John (in records 1, 4, and 6) has the same values assigned to every occurrence of his record.

3.4 Differential Privacy:

C. Dwork was the first who came up with the idea of differential privacy, and that's by adding noise during query evaluation to protect the database. Differential privacy is an effective privacy preservation method to preserve the privacy of data, at the same time, maintain its usefulness. Thus, researchers are working on new advanced differential privacy techniques to preserve data from leakage and discuss the implementation of this method in different parts of our life like Healthcare, where differential privacy can be an efficient way of protection in such field [1].

4 Internet Of Things:

As identified by Atzori et al. [2] Internet of Things (IoT) refers to the paradigm of connected objects with the ability to collect and share data over a wired or wireless network. Nowadays, IoT systems have reached almost every single object we use in our daily life. In IoT, every object with a sensor, transceivers, and microcontrollers can communicate and share information with other IoT objects through the Internet. The main idea of IoT is to make the Internet more diverse by making it easy for IoT objects to interact and connect with the Internet and with other IoT objects [3].

4.1 Implementations Of the Internet Of Things:

IoT's flexibility and dynamic nature make it easy to integrate into almost every part of our daily lives. In this section, we are going to discuss the different applications of IoT in our daily life. A presentation of some applications of IoT is demonstrated in Figure 3.

4.1.1 Wearables:

IoT enabled Wearables are smart objects that can be used as external accessories and worn; smart wearables gather and examine the data from the environment by using sensors to be able to make a smart decision and offer a helpful and mediate response to the user [4]. Smart wearables can be found everywhere in our daily life, from watches to glasses to even clothes. Mainly, every object that has the ability of transportation and can be attached to a human body can be equipped with special accessories to collect, analyze, and share data with other devices and the Internet. The amount of novelty and renewal in the field of smart wearables has shown how it becomes an essential part of our daily life, and researchers are being carried out to explore the potential of this field further.

4.1.2 Finance:

Thanks to IoT systems, there is a lot of innovation happening in the field of financial services. Financial IoT is a new paradigm, and there is lots of room for development and enhancement in different parts of the financial sector for more efficiency. Researches have begun to provide some improvement in different banking sectors, and there are many examples for that [5]. Nowadays, financial departments can use intelligent gateways to their application layer to efficiently handle their transaction and any other sensitive information. Hans, The integration of IoT in finance systems is playing an essential role in enhancing this field, and researchers are working to make a lot of beneficial solutions in that matter.

4.1.3 Smart cities:

Cities are getting smarter by using different types of electronic methods and sensors to collect, analyze, and share data to facilitate many tasks in different parts of our daily lives and help us achieve the digital promise of the connected world. There are many areas and activities in cities which have lots of space for improvements, such as **Electrical energy:** smart street lighting. **Security:** monitoring, fire control, and alarm systems. **Transport:** smart roads with warnings, messages, and deviations in accordance with the climatic conditions and unexpected events such as accidents or traffic jams. **Structural monitoring:** monitoring of vibrations and conditions of materials in buildings, bridges, and historical monuments. **Parking:** real-time monitoring of the parking spaces' availability. **Waste management:** optimizing the route of garbage collection with trash levels detection in containers [6].

Integrating IoT in all these different areas of cities can improve efficiency while minimizing the effort put into these activities and improve the quality of life in smart cities.

4.1.4 HealthCare:

The integration of IoT in Healthcare systems has great potential and can be very beneficial due to this field's importance to our life. The involvement of IoT in healthcare systems has produced many great applications like fitness, for example, monitoring and controlling heart rate during exercise and monitoring patients' conditions in the hospitals or their homes. Sensors and wearable devices collect and share patient data, Electronic Health Records that store and analyze patients' data [7], and many other implementations that can help in the process of Hospitalization of patients. The prevention of health problems becomes more effective with a real-time collection of information from the patient body, and the medical diagnoses become more accurate. The personal nature of the transmitted data share within the healthcare systems makes it mandatory for researchers to find new secure ways to handle sensitive information like these and protect it from any adversary.

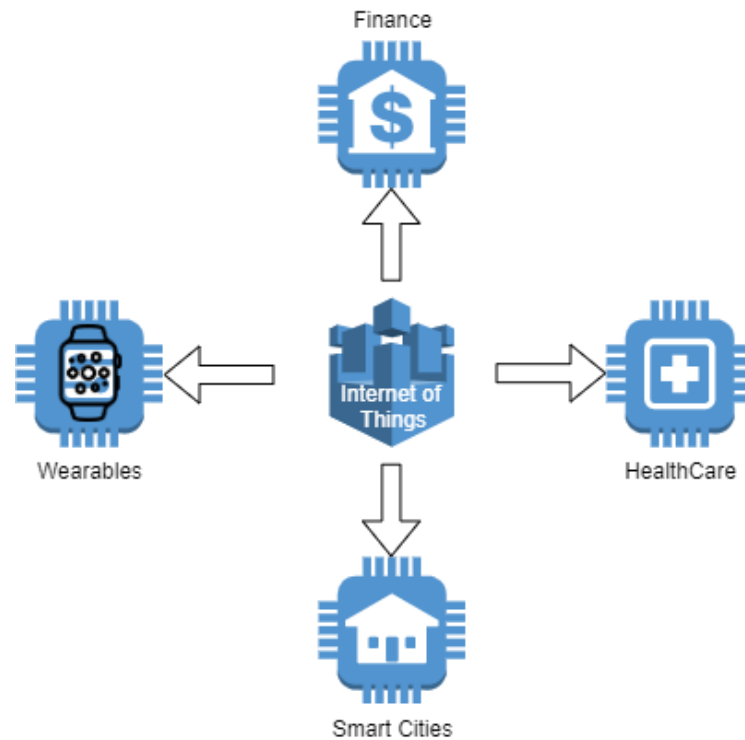


Figure 3: Implementation of Internet of Things in different fields.

4.2 Security and privacy Principles in IoT:

Security and privacy are fundamental principles of any information system. In this section, we discuss some of these principles in details [2]:

- **Integrity:** The certainty that the information has not been tampered with except by those who have the right to make these changes. Commonly, cryptographic mechanisms are used to check integrity.
- **Availability:** ensures that users of a given system are able to use it whenever necessary (the service is always active when requested by a legitimate user).
- **Confidentiality:** it is the guarantee that unauthorized users cannot obtain information. Only those with the rights and privileges cannot access the information. To ensure this principle, the blockchain uses a mechanism for pseudo-anonymization, like hash functions to blind users' identities.
- **Authentication, authorization, and auditing:** this seeks to verify the identity of who performs a specific function in a system, check what rights that user owns, and store usage information for that user. The blockchain ensures these three functions since only users who have the private key can perform transactions.

- **Nonrepudiation:** it guarantees that a person cannot deny an action in a system. The nonrepudiation provides evidence that a user performed a specific action such as transferring money ..., a user cannot deny that he has done it.

5 Blockchain Technology:

Blockchain is a digital ledger of transactions distributed across all the computers of a network; this ledger consists of a group of blocks tied together in sequential order by a hashing mechanism. Blockchain was first introduced after the appearance of the first cryptocurrency globally, BitCoin [8] back in 2008; Blockchain is a decentralized peer-to-peer network that manages the different transaction exchange operations without the need for a centralized third party involved. The massive success of BitCoin caused increased popularity to the technology which BitCoin is based on and pushed researchers to study the different functionalities and components of blockchain deeply. The decentralized nature of blockchain gives it the advantage of transparency, where any node in the network can check and confirm every transaction in the blockchain. That being said, no one can erase or modify any transaction in the network; this feature is called immutability. The transparency and immutability of blockchain make it a secure environment to store and distribute data across the network. A demonstration of a simplified Blockchain structure is presented in Figure 4.

5.1 Types of Blockchain:

Blockchain can be split up into different types based on its operating system and its way of implementation in the network [1].

5.1.1 Private Blockchain:

Private or permissible blockchain is a particular type of blockchain created to manage the different operations for a specific organization or a selected group. Participation in a private blockchain network is restricted and controlled. Any new user cannot be part of the network unless this user has an authorization to join. The execution process of the consensus protocol and maintain the shared ledger. There is typically no native token or incentives to motivate members to join and perform mining; therefore, mining is also handled by predetermined nodes from the network. These characteristics of private blockchain make it safe to work within some particular organizations like financial sectors and healthcare organizations.

There are many different implementation examples of private Blockchain like Hyperledger, which is an open-source framework that offers tools and functionalities to help build custom blockchain network use across various industries [9].

5.1.2 Public Blockchain:

A public blockchain is a peer-to-peer transparent, decentralized type of blockchain that allows any user to join its network. In public blockchains, nodes can have a copy of the distributed ledger to preview and verify all the transactions. They can also be part of the mining mechanism by collecting transaction information and checking its consistency using a consensus mechanism, creating blocks, and adding these new blocks to the blockchain by performing complicated mathematical operations. Nodes receive rewards after a block generated by the node gets successfully added to the blockchain. Since anyone can join the mining mechanism and miners are allowed to create and broadcast blocks, the blockchain network is required to create a mechanism for checking the integrity of these blocks. The most popular consensus mechanism in the public Blockchain is Proof-of-Work (PoW), this mechanism work over the concept of mining power that requires the blockchain to trust whichever block has the most computational work put into it, this mechanism requires miners to compete on who would create blocks first and that makes it harder for intruders to create new blocks the blockchain since they have to have 51% mining power of the network to control the blockchain.

The immutability and transparency of public blockchain make it suitable for many implementations, and maybe one of them is the first-ever Cryptocurrency BitCoin, but not for industries with sensitive nature like financial sectors and healthcare organizations.

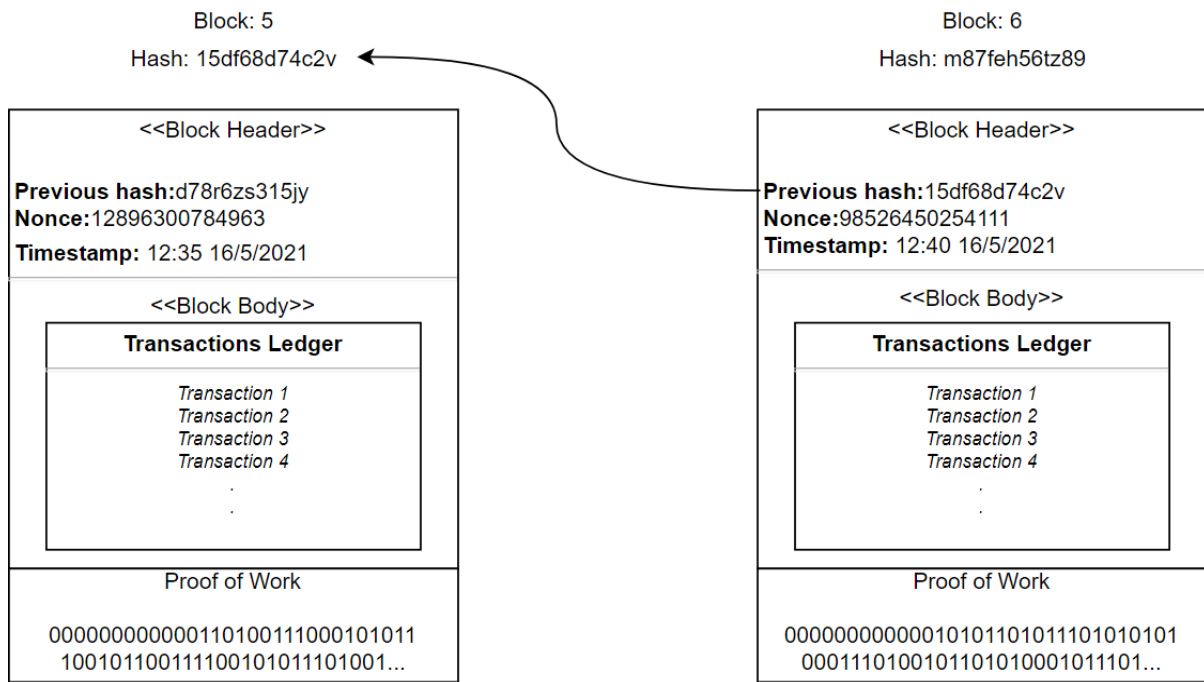


Figure 4: Simplified structure of blocks.

5.2 Working Phenomenal Of Blockchain:

Blockchain is a peer-to-peer network where each node in the network can exchange data using a broadcasting protocol that carried out the data in the network. Miners listen for transactions, create blocks and broadcast these blocks in the network. A block must have a computational power put into it for it to be added to the blockchain. Each block in the blockchain contains the transactional information along with its predecessor hash and other information like creation time; once a block gets validated and added to the blockchain, it cannot be modified or erased. The first-ever block created in the blockchain is known as the genesis block. Blocks are tied together in chronological order using a hash mechanism and are visible and transparent to any user in the network [11].

Blockchain does not have any third party involved to manage the transaction operations; instead, it uses a consensus mechanism to validate transactions. There are various consensus mechanisms out there; for instance, Proof of Stake (PoS), Proof of Importance (PoI), Measure of Trust (MoT), but the most popular one yet is Proof of Work (PoW) which is the first-ever consensus mechanism in the history of Blockchain [1].

5.3 Characteristics of Blockchain:

Blockchain has many advantages and characteristics that make it stand alone as a great technology for many industries and a solution to many problems. In this subsection, we

reveal some of the best characteristics of blockchain.

- **Decentralized nature:** the decentralized nature of blockchain eliminates the concept of trust in the network, so no one has to know or trust anyone else. Each member in the network has a copy of the same data in the form of a distributed ledger. If a member's ledger is altered or corrupted in any way, it gets rejected by the majority of the members in the network. This concept removes the need for a centralized third-party involvement, which helps in many fields where trust is a huge issue.
- **Immutability:** while anyone in blockchain can check all the transaction and verify it, no one can modify or remove any transaction or block which develops a trust in the data contained in the blockchain and make it a reliable ledger of information.
- **Security:** in blockchain, the data is structured into blocks. Each block contains a transaction or bundle of transactions; these transactions can be encrypted using different layers of data preservation strategies to ensure security.
- **Identity backtracking:** in blockchain, data identity can be backtracked to its origin using a unique identifier, which can be helpful, especially when there is a need to identify the source for some information.

5.4 Challenges of Using Blockchain in IoT Systems:

Blockchain systems are well-known for secure and immutable transactions. Specific encryption and authentication strategies are being used in some blockchain systems to ensure the security of data. These security services of blockchain are implemented using various key encryption schemes, in which all nodes of the network have their private key along with the public key via which they manage transactions in the network. These strategies do only serve the purpose to protect transaction security, but on the other hand, integrating blockchain technology into a system as sensitive as Electronic Health Record Systems that need a considerable amount of protection can be a very challenging task. In [8], S. Nakamoto shows that the identity leakage can be a danger as if a person identity gets leaked in a transaction that would leak all the transaction detail of that same person and expose that person to significant threats.

The transparency of blockchain ledgers can be a great feature of blockchain as it can be a significant threat to its integrity. And as it mentioned earlier, the leakage of a person's identity in a transaction can lead to the leakage of all the transaction from that same person. Furthermore, the ease of access to the blockchain ledger can lead to significant information

leakage. In this section, we showcase some of the harmful attacks that would affect blockchain-based IoT Systems [1].

5.4.1 Sybil Attacks:

One of the major attacks considered in blockchain-based IoT systems is Sybil Attacks in which the adversary create a large number of fake users node and try to gain trust among the blockchain network, as mentioned before in a public blockchain PoW consensus the adversary is required to have at least 51% of the mining power to control the network, so the adversary try to create as many fake nodes as possible to gain influence in the network and access the personal data of other users. Researchers are working to defeat this kind of attack by using different mechanisms and algorithms like NetFlow [11]; simultaneously, researchers need to make changes to some of the mechanisms of blockchain to reduce the threat of this attack on IoT Systems.

5.4.2 Linking Attacks:

Linking attacks is an attempt to re-identify individuals in an anonymized dataset by combining that data with background information, in blockchain-based IoT systems linking attack can be carried over a distributed ledger and its mainly directed towards stored data, the ledger contain a copy of the targeted transaction, For example, an anonymized data collected from two separate blockchain databases having records of same individuals; basic re-identification can be carried out by using various linking algorithmic attacks. Therefore, even anonymized data is not 100% safe, so strong privacy protection needs to be maintained while publicizing blockchain private data. Preserving IoT devices' privacy requires different needs as compared to traditional blockchain applications. Because in online payments of traditional blockchain applications, only transaction amounts and identities require privacy. While in IoT applications of blockchain, many parameters need to be preserved before publicizing any information. Keeping in view all these aspects, it can be concluded that privacy preservation of blockchain-based IoT systems is essential. These systems require specific privacy protection strategies to protect individual and device privacy.

5.4.3 Address reuse:

Public addresses of Blockchain users are open to anyone in the network, and adversaries can easily access these addresses via Internet access. These kinds of attacks are called Address reuse, in which the adversary links the address with the original identity, and that allows him to access every transaction or information exchange of that particular user. Sometimes the privacy is protected by using pseudonym addresses that do not necessarily

have a link with actual identities, but still, pseudonymity does not provide complete protection as specific methods can be adopted to link Blockchain data with the original owner.

6 Challenges & Solutions for Integrating Blockchain to EHR Systems:

Generally, blockchain is a public ledger in which everything is easily accessible by every Blockchain member. Leakage of a user identity in any transaction can lead to the point where every transaction or information exchange of that particular user becomes visible to the adversary and cause leakage of private information since transactional records are available for everyone in the public ledger and cause a significant loophole to the privacy of the blockchain network. Moreover, IoT user identity privacy preservation and Transactional privacy preservation among IoT nodes must be taken into consideration in the process of integrating Blockchain into EHR systems. In the previous section, we discuss some of the attacks that can occur to a blockchain-based IoT system. Similarly, integrating Blockchain to EHR System can face the same security challenges, and harmful attacks as blockchain-based IoT system has since EHR Systems are essentially a type of IoT Systems. Furthermore, the sensitive nature of the information that is contained in EHR Systems requires us to examine and overcome all the challenges and issues that come with blockchain technology, mainly the security and privacy issues, to be able to find the best solution for integrating Blockchain in EHR Systems and benefits from blockchain advantages like immutability, security, and decentralized nature.

7 Related Work:

In this section, we are going to discuss different EHR applications presented in earlier work. Showcase and evaluate the mechanisms and approaches used in these applications; in this section, we will provide background knowledge about various existing approaches to protect the security of EHR systems.

Our work is inspired by a few applications that are utilized to provide security and protection to EHR systems. Despite the existing solutions, privacy issues are major obstacles that are limiting the widespread adoption of public clouds across the globe. The main reason for this concern is that the information needs to be published to a broad and possibly anonymous set of receivers and sensitive data are hazardous for outsourcing to the cloud, there is an increasing need to investigate data anonymization techniques applied to this

domain. In [12] the authors introduce a new protocol called BSPP to protect data of consortium blockchain-based e-health system. The authors mentioned that patient identity leakage can cause harmful effect and therefore highlighted the protection of original identity. To protect identity, the authors used the technique of anonymization by implementing the concept of pseudo identities as evidence for conformance proof. Such a strategy is developed in which only the designated doctor can check pseudo identities of patients, and an adversary cannot trace the actual identities. Moreover, the presented algorithm does also provide searchable confidentiality in which the hospital staff can search from patients data without inferring private information. Another important aspect of the presented paper is its unique consensus mechanism named "proof-of-conformance" in which the patients register for a specific doctor in the hospital, and the doctor generates the PHI records of patients and encrypt keys and protect it using pseudo identities.

Another anonymization protocol is presented in [13], in which the authors used common secret keys alongside a PoW consensus mechanism used to generate anonymous ID to prevent blockchain data from Sybil attacks and preserved information safety. the authors used the concept of bilinear pairing to protect identity and location of patients. They took a step further and prevented location leakage during communication and message transmission among patients. The proposed concept enhanced the confidentiality of message and protected e-health hospital records from tracing attacks.

Moving further to the next scenario, the authors in [14] present an architecture of a blockchain-based tamper-proof electronic health record (EHR) management system. They introduce the concept of blockchain handshaker that works as a blockchain wrapper for supporting blockchain integration in the system, the blockchain handshaker work as we wrapper layer integrated mechanism that handles communication between the existing cloud-based EHR management system and public blockchain network. The authors also used an anonymization protocol based on public-key cryptography that generates an anonymized transaction from the submitted patient record by the user (doctor, administrator). This proposed protocol improves the confidentiality of patient information contained within the transaction and protect the system from major attacks. On the other hand, this proposed system is not perfect either and it suffers from certain weaknesses, one of these weaknesses is that the proposed architecture does not emphasize the security of patient records and mainly focused on the structure of the system, which can lead to leakage of data. Furthermore, the authors did not address the case of fetching data which make the architecture incomplete.

8 Conclusion:

This chapter covered the different characteristics and components of EHR Systems, concluding on the privacy and security requirements for its implementation. Moreover, we discussed the different technologies, methods, and strategies that could help us navigate our way to build a secure foundation for an integrated EHR system.

CHAPTER II
ARCHITECTURE & OPERATION OF THE
PROPOSED MODEL

1 Introduction:

After going through introductions and definitions of the different technologies and methods that we need in our study, we now have a theoretical foundation that could help us propose our integrated system. This chapter first presents a description of the main concepts we are going to need in our integrated system. After that, we introduce our proposed system architecture of Blockchain integrated cloud-based EHR System, discussing each of its components in a detailed approach. Furthermore, we discuss the workflow of our integrated system, understanding the process of handling the information flow in the system, and the different components involved in the process.

2 Basic Concepts and Preliminaries:

In this section, we'll go through some of the core concepts that are essential to know before going through our proposed architecture, these concepts are Data Preservation Mechanisms, Smart Contract, Fetching Query, Patient Record, Transaction, and Transaction Template.

2.1 Data Preservation Mechanisms:

The sensitive nature of the healthcare information requires us to follow a methodology that provides protection to patient records within the system. In our system, the data preservation mechanism is a set of rules and principles that used to preserve the integrity of patent records within the system. Our proposed APM anonymity mechanism must follow these methodologies to provide privacy and security for the patient records received from the users or the health sensors of the patient. We define our Data Preservation Mechanism in the following points:

- **Identifying attribute:** in our system, a patient record contains personal information about that same patient, some more sensitive than others. For example, attribute such as names, addresses, or identity card numbers. They permit direct identification to the patient and any leakage of these attribute would reveal that patient real identity and expose him to major threats also these types of information are not needed for statistical or research purposes. Therefore, these types of personal identifiable information should be secured within the system, and removed from the published dataset. Another type of personal information that are not direct identifiers themselves. Attributes like age, gender, city, profession do not individually reveal the patient real identity but when combined with other particular attributes can reveal the identity of the person behind it. For instance, if an

individual of that particular gender, age, and profession lived in that particular city, an intruder may re-identify that person based on these types of attributes. Such attributes are needed for medical and statistical purposes, and should not be removed from the published dataset. Therefore, high anonymization techniques and encryption methods should be applied to some of these attributes to reduce the risk of re-identification to an acceptable level.

- **Re-identification cost:** in the health field, data such as patient information are highly sensitive. And sometimes, acquiring and re-identifying these data can be relatively easy for adversaries, who uses these data to threaten and blackmail patient different kinds of reasons and benefits, and when the cost of re-identification these types of data decreases, the benefits of the adversaries increases. Therefore, using high anonymization techniques and encryption methods can increase the computational cost of re-identification, and the motivation of intruders would typically be much lower as the re-identification cost surpasses the gain benefits. Applying the principle "**The higher the cost, the lower the benefit for intruders**".

- **Data usefulness:** Privacy-preserving patient records resulting from anonymization techniques via generalization and suppression can prevent re-identification of data and protect it from intruders, but using these types of anonymization may also significantly decrease the data usefulness. Hence, techniques like pseudonymization and encryption methods that do not affect the usefulness of data must be used instead of other anonymization techniques that reduce the data usefulness unless it's necessary. Therefore, there must be a balance between anonymizing the data and preserving its usefulness.

2.2 Smart Contract:

Nick Szabo defined the smart contract as “A computerized transaction protocol that executes the terms of a contract” [18]. In our proposed system, we use smart contract with the purpose of maintaining the patient records security and ensure its integrity. Our smart contract validates patient records which is represented as a set of transactions based on the patient pre-existing health records, and check the reliability of the data contained in the received transaction by executing a specific set of instructions.

2.3 Fetching Query:

Fetching queries (FQ) are submitted by the users (i.e., doctors, administrator) when searching for a specific record of a patient. The FQ contains the identity of the user alongside the anonymized identity of the patient. The client computer of users sends the FQ to the APM to add a layer of encryption to secure the user's identity and send it to the network server for further processing.

2.4 Patient Record:

Patient records are the principal repository for information concerning a patient's health care. In our system, patient records are generated and organized by the health sensors of the patient body or by health care authorized users as a direct result of interaction with the patient. After it gets generated, the initial patient record go through an anonymization phase implemented by the APM encryption method to preserve the data integrity and protect it from intruders, a demonstration of the multilayer structure of an anonymized patient record is presented in Figure 5, and a description of the Anonymized Patient Record is provided in the following subsection:

2.4.1 Anonymized Patient Record:

Anonymized Patient Record is generated by implementing the APM encryption method to the initial patient record received from the user or the health sensors of the patient body. A description of the Anonymized patient record (*aPR*) is provided in the tuple below:

$$aPR = PKE ([Hash (Rid, \{ a_1, a_2 \dots a_n \}, SR), SKE(\{ b_1, b_2 \dots b_n \}, SK_{APM})], PK_{NS}).$$

- *Rid*. Represent the ID of the generated patient record.
- *Hash (Rid, { a₁, a₂ ... a_n })*. Represent the hashing message containing personal identifiable attributes in the set $A = \{a_1, a_2, a_3 \dots a_n\}$ of n attributes related to the patient health record and the Rid.
- *SR*. Represent the number of salting rounds used in the process of hashing/encrypting the data with a random salt.
- $(b_1, b_2 \dots b_n)$. Represent the message containing the non-identifiable attributes in the set $A = \{b_1, b_2, b_3 \dots b_n\}$ of n attributes related to the patient health record.
- $aPR = PKE ([Hash (Rid, \{ a_1, a_2 \dots a_n \}, SR), SKE(\{ b_1, b_2 \dots b_n \}, SK_{APM})], PK_{NS})$. is the encrypted *aPR* using the public key of the Network Server, PK_{NS} .

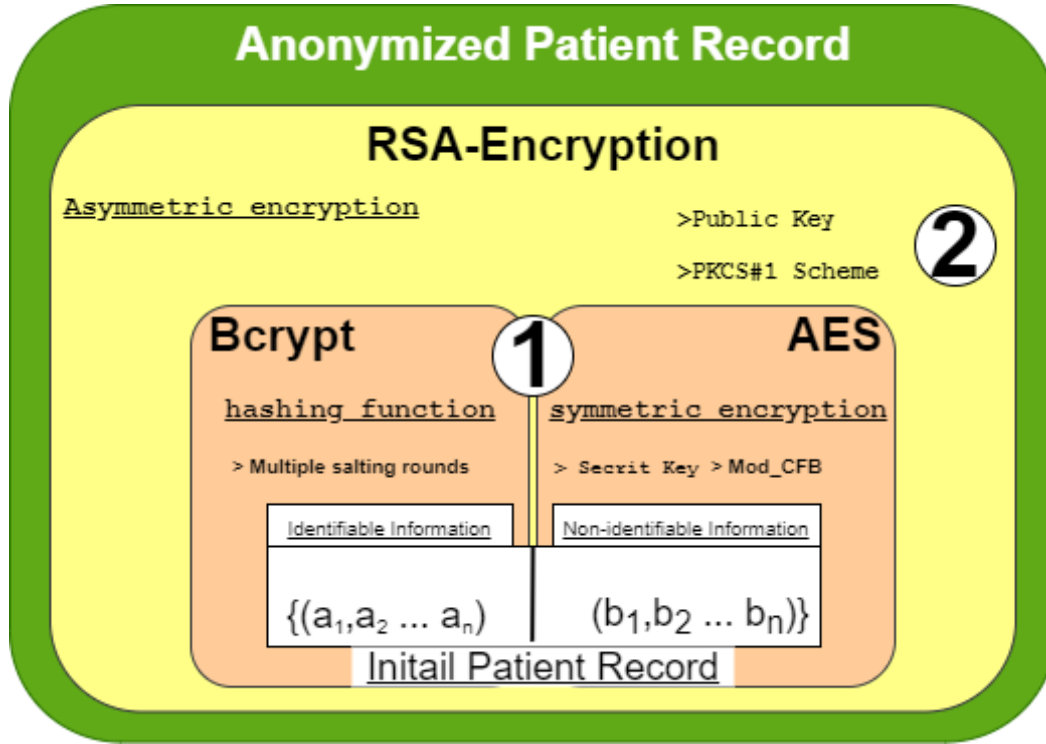


Figure 5: Architecture of a multi-layered anonymized patient record.

2.5 Transaction:

In our system, a transaction is an immutable record that contains information related to patient health records. The entity responsible for generating transaction is the network server which it sends the generated transaction to the private blockchain for validation, and after the transaction gets validated, it gets sent to the Cloud-based EHR management system for further processing. For generating a transaction, the network server follows a transaction template that is created by the system administrator according to the transaction format of the Public Blockchain. There are two types of transactions:

2.5.1 Primary Transaction (PT):

Primary transaction is basically the initial version of the transaction generated from the patient records received from the APM. After generating the transaction, the network server sends it to the private Blockchain for The purpose of validation. Primary transaction contains the identification of the patient, the identification of the same transaction alongside with the attribute of the patient record; a description of the PT is presented in the tuple below:

$$PT = PKE ([Tid, aPR], PK_{PB}).$$

- *Tid* . Transaction ID
- *aPR* . Anonymized Patient Record

- $PT = PKE ([Tid, aPR], PK_{PB})$. encrypt the transaction using the public key PK_{PB} .

2.5.2 Query Transaction (QT):

Query Transaction is generated by the network server based on received anonymized query (aQ) from the APM. The network server sends the user's id of the user alongside the anonymized patient's id. a description of the QT is presented in the tuple below:

$$QT = PKE ([Uid, aPid], PK_{NS}).$$

- Uid . User's ID
- $aPid$. Anonymized Patient's ID
- $QT = PKE ([Uid, aPid], PK_{NS})$. encrypt the transaction using the public key PK_{NS} .

2.5.3 Final Transaction (FT):

This type of transaction is generated by the private blockchain as a result of the transaction validation process. The Final Transaction consists of the validated data contained in the received transaction (PT or QT) alongside the validation result of those same patient records. Similarly, the tuple below describes the FT:

$$FT = PKE ([(PT \parallel QT), VR], PK_{NS}).$$

- PT . Primary Transaction
- QT . Query Transaction.
- VR . The validation result
- $PKE ([PT, VR], PK_{NS})$. encrypt the transaction using Network Server public key PK_{NS} .

2.6 Transaction Template:

The Transaction Template is a formal prototype that defines the outline of each transaction. Transaction Template is built and used by the Network Server for the purpose of simplifying the process of generating transactions.

3 Proposed System architecture:

In this section, we present our proposed cloud-based EHR System architecture. Furthermore, we discuss the different components of the system, diving deep into explaining the fundamentals and demonstrating the role of each component in the system. There are four main components in our proposed architecture: APM, Network Server, Private Blockchain,

Cloud-based Electronic Health Record Manager System. An overview of the system architecture is shown in Figure 6.

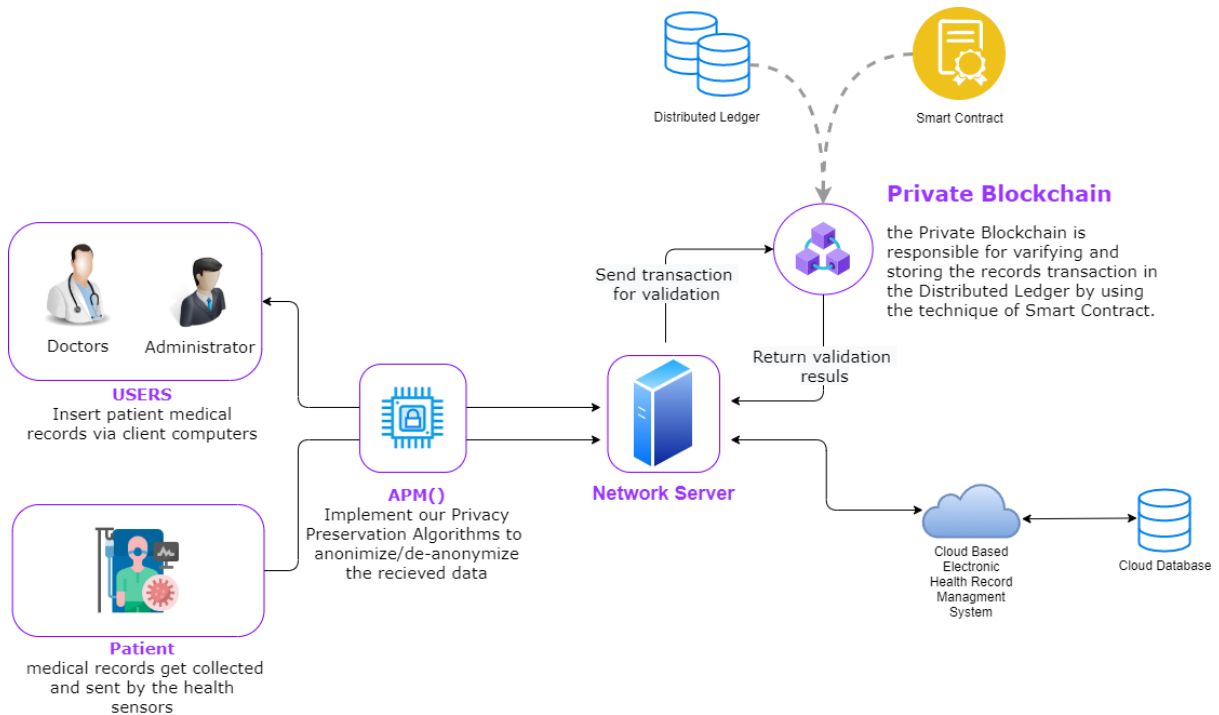


Figure 6: Proposed System Architecture of cloud-based Electronic Health Record.

3.1 Anonymity Preservation Mechanism (APM):

Due to the patient records' vulnerability and sensitive nature, which requires us to use a data preservation mechanism in our system. In our cloud-based EHR architecture, the APM represents the system's core component, responsible for providing protection and anonymization to the patient records before sending them to the network server while preserving its utility and decrypting the received anonymized patient records (*aPR*). In the first case, the APM receives the unencrypted patient records (*iPR*) from the user or the patient health sensors as inputs. Then, implement its privacy preservation method to the patient records and generate an anonymized version of the received records (*aPR*) as outputs, which gets sent to the network server for further processing. A description of the used method for this process is presented in Method 1. In the second case however, the APM received the anonymized patient records (*aPR*) from the network server and generate a decrypted patient record as a result of the query sent from the user. A description of the used method for the second process is presented in Method 2

Method 1 anonymization of the initial patient record

```

1 Require: Bcrypt, RSA, AES
2 Input:  $iPR_a, iPR_b, SR, PK_{NS}, SK_{APM}$ ;
3 For every item (i) in  $iPR_a$  do
4     hash  $aPR_a[i]$  with Bcrypt using (SR);
5 End
6 For every item (i) in  $iPR_b$  do
7     encrypt  $aPR_b[i]$  with AES using ( $SK_{APM}$ );
8 End
9 encrypt ( $aPR_a, aPR_b$ ) with RSA using ( $PK_{NS}$ ) and assign the results to the  $aPR$ ;
10 Output: Anonymized Patient Record ( $aPR$ );

```

- *Bcrypt*: is a hashing function used to generate a signature for a given message a certain number of salting rounds (SR) using a random salt.
- *RSA*: is an asymmetric encryption algorithm that is widely used for secure data transmission. We use this algorithm in our system to encrypt the aPR with the network server public key before sending it.
- *AES*: is a symmetric encryption algorithm that we use to encrypt the non-identifiable attributes in the initial patient record with the APM secret key.
- iPR_a : iPR_a is the initial patient record that contains personal identifiable attributes in the set $A = \{a_1, a_2, a_3 \dots a_n\}$ of n attributes related to patient health records.
- iPR_b : iPR_b is the initial patient record that contains non-identifiable attributes in the set $A = \{b_1, b_2, b_3 \dots b_n\}$ of n attributes related to patient health records.

Method 2 de-anonymization of the received *aPR*

```

1 Require: Bcrypt, RSA, AES
2 Input: aPR, SR,  $PrK_{APM}$ ,  $SK_{PPM}$ , PP;
3 decrypt (aPR) with RSA using ( $PrK_{APM}$ ) and assign the results to the aPR;
4 For every item (i) in PP do
5     hash the ID of PP[i] with Bcrypt using (SR);
6     If the hashed ID == the id of the received aPR do
7         For every item (j) in PP[i] do
8             assign the PP[i][j] item to  $iPR_a[j]$ ;
9         End
10    Break
11 End
12 For every item (i) in  $iPR_b$  do
13    decrypt  $iPR_b[i]$  with AES using ( $SK_{APM}$ );
14 End
15 Output: decrypted Patient Record;

```

- *PP* : Stands for Patients Profiles, and it's a record that contain the identifiable information (ID, First Name, and Last Name) of every patient in the systems.

3.2 Network Server:

The network server is responsible for the communication between the APM, the private blockchain, and the Cloud-based EHR management system in our proposed architecture. The network server act as a wrapper component that consists of a series of protocols that allow it to manage communication between the other entities. A description of the internal architecture of the Network Server is presented in Figure 7.

Due to the centralized nature of this component, the network server should be able to handle the flow of information within the system, and perform specific actions to the received information based on the situation to ensure the success of each operation in the system. The Network Server has two main sub-components that perform these actions:

- **Transaction Generator (TG):** generate a Primary transaction (PT) from the Anonymized Patient Record (*aPR*) that was received from the APM following the specification of the predefined Transaction Template (TT).

- **Transaction Validator (TV):** Transaction Validator is the part responsible for managing the communication between the APM, the Private Blockchain, and the Cloud-based EHR management system. TV receives an Anonymized Patient Record from the APM, decrypts it, and sends its data to TG to generate a Primary Transaction. After receiving a PT, TV sends it to the Private Blockchain for validation and waits for the return. The Private Blockchain sent the validation results as Final Transaction. If the FT is valid, the Network Server sends the *aPR* attached to the FT to the cloud. Otherwise, VR is sent as an invalid transaction and stored for future audit tasks.

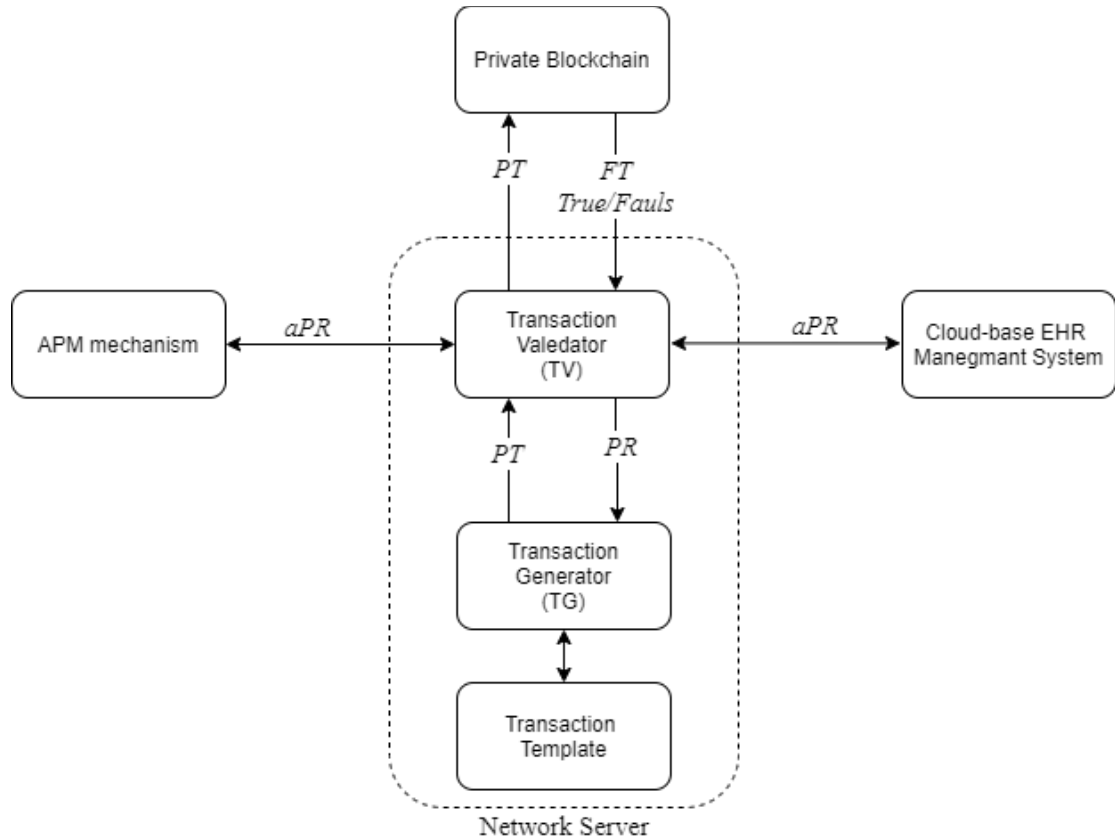


Figure 7: Internal Architecture of the Network Server.

3.3 Private Blockchain:

Private blockchain is the responsible entity for verifying and storing transactions. In our proposed architecture, we use an Ethereum based private Blockchain that consists of a smart contract that handles the process of validating the transaction received from the network server and a distributed ledger that stores the validated transaction. The private blockchain shares the distributed ledger among multiple nodes (computers). Each node has a copy of all the transactions and gets updated each time a new transaction gets validated. The process of validating a transaction is called mining, and it is performed by nodes which in this case are called miners. After miners validate the transaction, the data stored in that transaction are converted into a block and get stored in the distributed ledger. To prevent any tampering during the mining process, a Proof-of-Work (PoW) consensus mechanism guarantees the integrity of the newly mined blocks. After the validation process complete, the private blockchain sends the result as **true** or **false** to the network server.

3.4 Cloud-based EHR Management System:

The cloud-based EHR management system is the final station that patient records arrive to, the main purpose of this component is to store the patient records in a secure database and control the access to this database in which only the authorized users have such permission. The cloud-based EHR system receives the valid transactions from the network server and stores these transactions in the cloud database. Similarly, the cloud also responds to authorized user's query requests with an appropriate reply.

4 System Workflow:

This section discusses the system workflow of the cloud-based EHR system and the integrated Private Blockchain. Going through a detailed explanation of how the system's component interacts with one another and following a step by step description of the sequential path that information follow, first from where it's initially collected and inserted into the system back to its final place where the information gets stored in the cloud database.

Firstly, there are two ways of submitting the patient record into the system. Either the data gets collected by the body sensors of the patient, or they get submitted by the user (i.e., doctors, administrator). Figure 8 shows the architecture of the system workflow in the case of sending patient records from the health sensors. After submitting, the patient record goes through a couple of phases before reaching the last stage. The first phase is anonymization, in which the APM implements an anonymization method to the patient record to preserve its

integrity and protect patient identity from leakage. The APM method returns an encrypted Anonymized Patient Record (aPR) as an output. After that, the (aPR) gets sent to the Network Server, which decrypts it and generates a Primary Transaction (PT) using the Transaction generator. Next, the Network server sends the (PT) to the Private Blockchain for further validation. A Smart Contract is used to validate the integrity of the transaction, and mining nodes generate a new block that contains the data of the validated transaction and adds that block to the Blockchain Distributed Ledger, and a Proof-of-Work (PoW) consensus mechanism control the process of mining to ensures the integrity of new blocks. After the validation phase, the Public Blockchain sends a validation result to the Network Server in the form of a Final Transaction (FT). If the (FT) validation result is invalid, the VR is sent as an invalid transaction and stored for future audit tasks. Otherwise, the Network Server sends the patient record (aPR) attached to that (FT) to the Cloud-based EHR Management System, which finally stores that data in the cloud database.

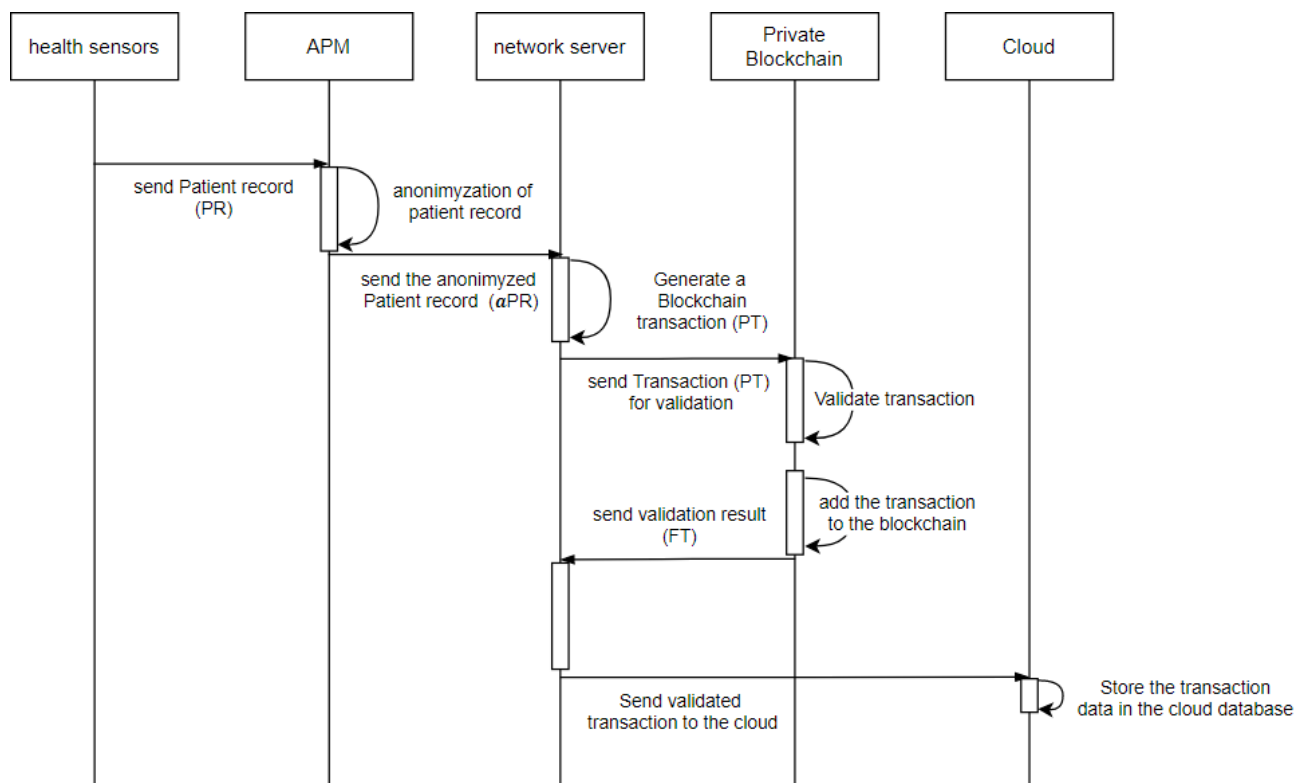


Figure 8: Sequence Diagram showcases the process of storing patient records in the system.

Similarly, when the user's client computer sends a fetching query of a specific patient record to the APM, the query goes through a similar validation process. The APM anonymizes the patient identity in the received query and sends it to the network server,

which generates a query transaction (QT) and sends it to the Private Blockchain for further validation. After the validation phase, the Public Blockchain sends a validation result to the network server. After the validation phase, the network server sends the validated query to the EHR management system, which searches for that specific record in the cloud database using the anonymized patient identity contained in the query and return a α PR, the network server sends the received α PR to the APM which perform its de-anonymization method to it and sends the result back to the user's client computer. Figure 9 shows the architecture of the system workflow in the case of sending patient records from the health sensors.

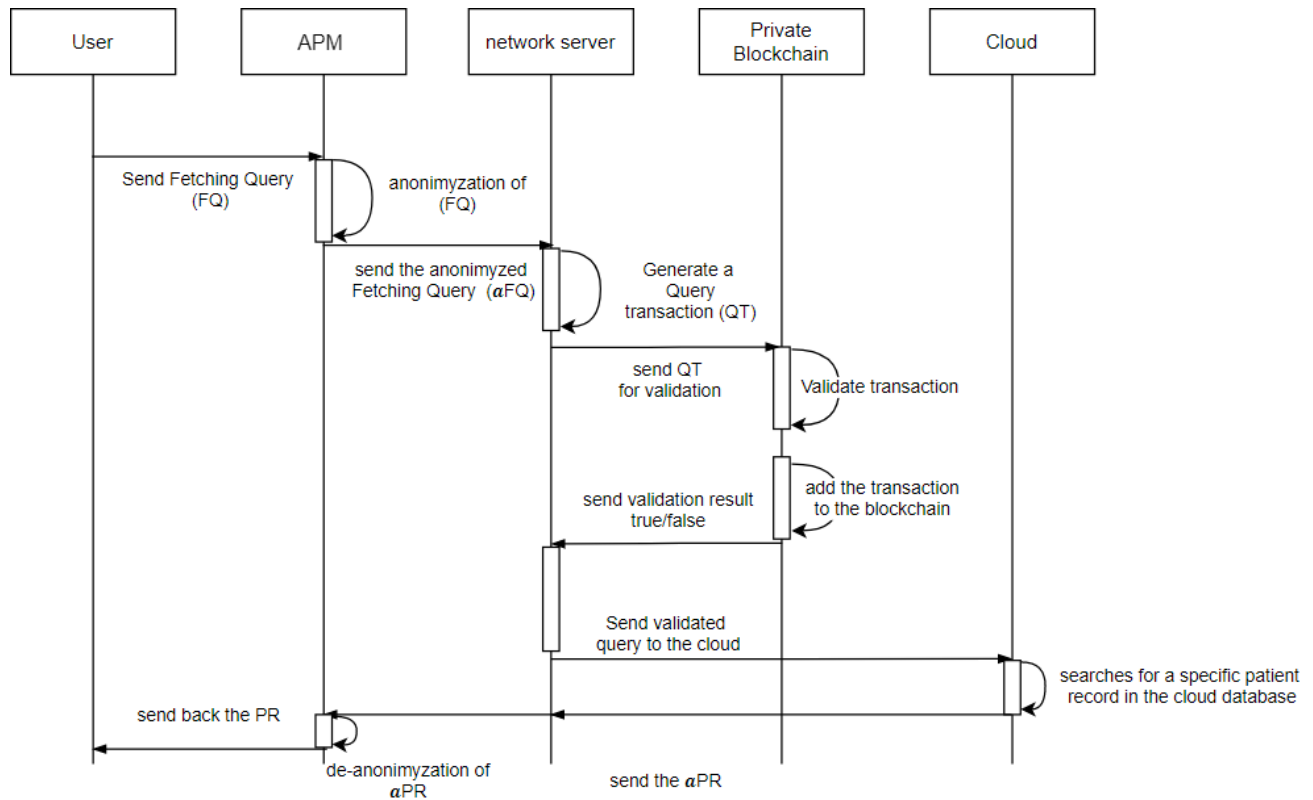


Figure 9: Sequence Diagram showcases the process of fetching patient records from the system.

5 Conclusion:

In this chapter, we proposed our integrated system architecture of Cloud-based Electronic Health Record, describing each component in detail and going through the different concepts and techniques used in our proposed system. Furthermore, we discuss our system's workflow and go over the lifecycle of patient records within the system, which gave us a precise and accurate understanding of the different parts of our system. This understanding makes it easy for us to build our integrated system.

CHAPTER III

IMPLEMENTAION & RESULTS

1 Introduction:

After proposing our integrated EHR management system, highlighting and recognizing each component's role in the system in the previous chapter, we become closer to achieve our ultimate goal, which is to convert our integrated system into an implemented prototype. In this chapter, we are going to showcase our implemented prototype of a cloud-based EHR management system. First, we go through the different implementation scenarios of our prototype, explaining each case scenario in detail. Then, we present our prototype component and discuss the role of each component in our system associated with pictures.

2 Implementation Scenarios:

In this section, we gave a detailed walkthrough of our prototype implementation by covering a case scenario first, in which we are going to present a simple example of an operation in our prototype.

2.1 Case Scenarios:

Our first case scenario example began when a doctor performs a medical assignment to the patient. After that, the user (i.e., doctor or administrator) interacts with the client application to insert the patient record into the system after an actual medical session with the patient. Every time a patient record with new patient information gets inserted into the system, the client application automatically creates a new profile of that patient information. After that, the patient record gets sent to the APM, which performs an anonymization method to the received data and generates an anonymized patient record (*aPR*) which gets sent to the network server for further validation. After the validation phase, the network server sends the *aPR* to the EHR Cloud Database for further processing.

The second case scenario starts whenever a user searches for a particular patient record in the system, the user inserts the id of a patient record into the client application, which sends it to the APM to perform the same anonymization method of the patient record to the received id and generate an anonymized id (*aID*) which get sent to the network server for validation. After validation, the network server sends a fetching query of the *aID* to the EHR Cloud Database. The network server sends the received fetching results (*aPR*) to the APM, which de-anonymizes it and sends the result (*iPR*) to the client application for further processing.

2.2 Scenarios Details:

Our prototype consists of four parts: client application, APM, network server, and EHR cloud database. In this subsection, we are going to present with pictures of each part of our prototype, covering the different components of each part and the role it plays in our system. The client application has two main functionalities; **1.** Insert patient records into the system, **2.** Search for a specific or multiple records in the EHR Cloud Database.

In the first case, the user supplement three input fields in the "New Patient Record" page of the client application ID, Condition, Description. The data inserted into the client application gets sent to the APM, which includes more information to the received data (Firstname, Lastname, Age, Gender) based on the patient ID. Then the APM generates an Initial Patient Record which contains two groups of information; Identifiable information and Non-identifiable information. Next, the Initial Patient Record gets sent as an input to the anonymization method of the APM (**Encryption_fun**), which executes our anonymization method to the received record and generates an Anonymized Patient Record (*aPR*) as an output. After that, the APM takes the output of this process and sends it to the Network Server, which validates the received *aPR* and sends it as an insertion query to the EHR Cloud, which stores it into its database.

In the second case, the client application receives an id inserted by the user in the "Search for Record" page. The client application sends the received ID to the APM, which implements a similar anonymization method (**Encryption_query**). Then the anonymized ID gets sent to the Network Server, which generates and sends a fetching query of one or more *aPR* to the EHR Cloud database and listens for the result. The Network Server sends the fetching query result (*aPR*) to the APM de-anonymization method as an input (**Decryption_fun**), which implements our de-anonymization method to the received *aPR*. Finally, the APM sends the de-anonymized patient record back to the user through the client application.

3 Implementation Details:

In this section, we are going to present the tools and framework used to develop the prototype of our proposed Cloud-based EHR system. These methodologies are described with pictures.

3.1 Client Application:

We developed a client application using HTML5 and Bootstrap framework as a front-end with a Node.JS environment in the back-end. This application is connected to a MongoDB Database program that manages user's authentication. The client application works as a friendly interface that allows authorized users to interact with the EHR system. Figure 10 describes the general layout of an example page in our application.

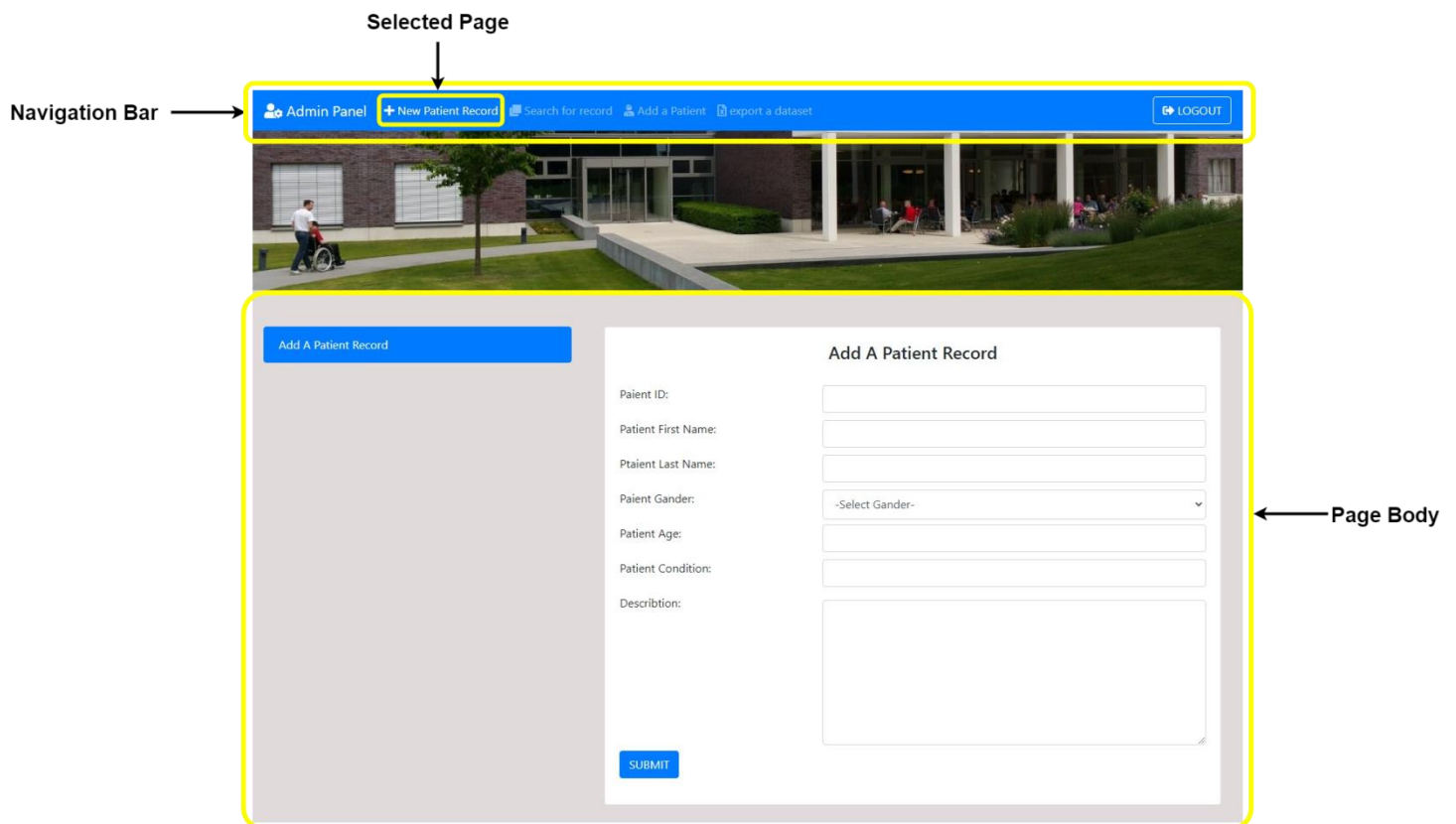


Figure 10: A general layout description of a client application page.

There are four pages in our client application; each page has a specific role to play and a specific input body; these pages are described down below:

- **New Patient Record:** this page allows users to insert patients record into the system, the user fills the page body form with the patient record variables and submits it. If this process was successfully completed, the user gets a successful message. Figure 11 presents a description of this page content.

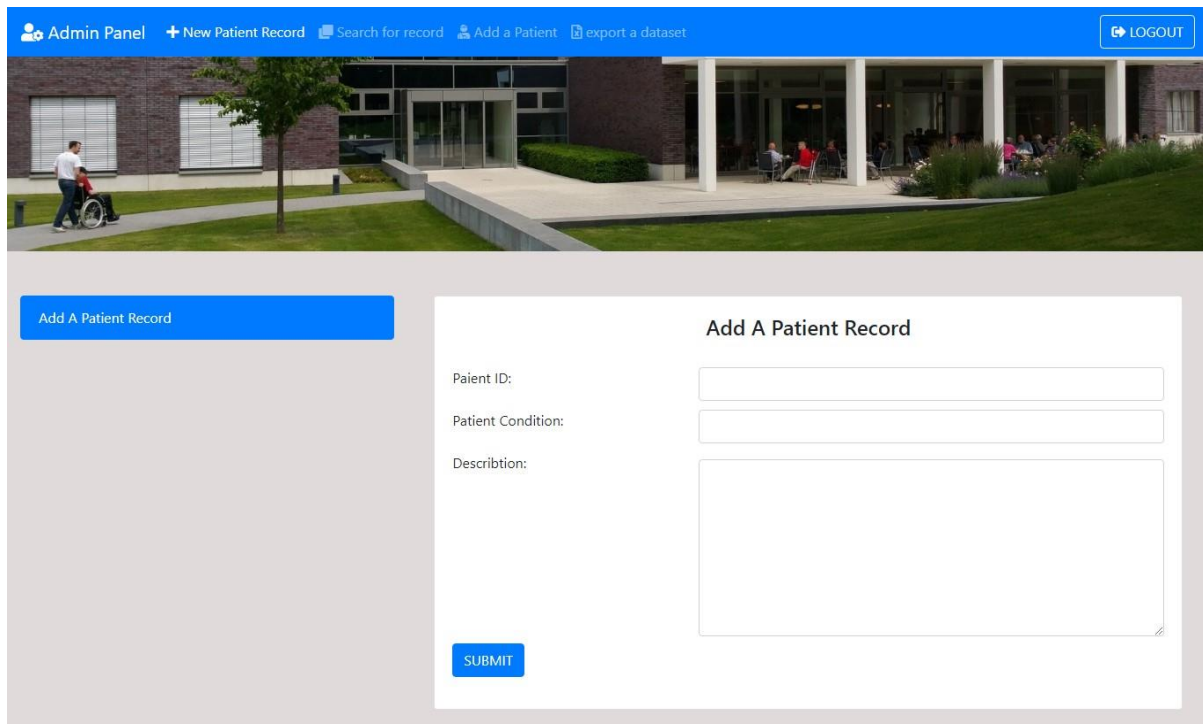


Figure 11: presentation of the “New Patient Record” content.

- **Search For Record:** this page allows users to search for specific or multiple patient records in the EHR cloud database. A description of this page layout is presented in Figure 12.

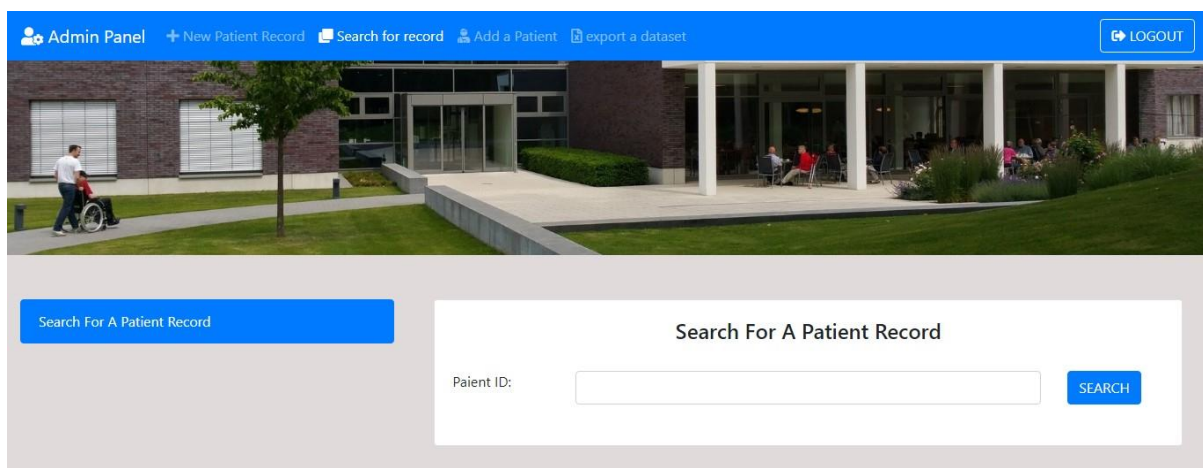


Figure 12: presentation of the “Search for Record” content.

- **Add a Patient:** this page allows users to add a new patient profile into the system. Users fill the page body form with the new patient information and then submit these information. Figure 13 describes this page's content.

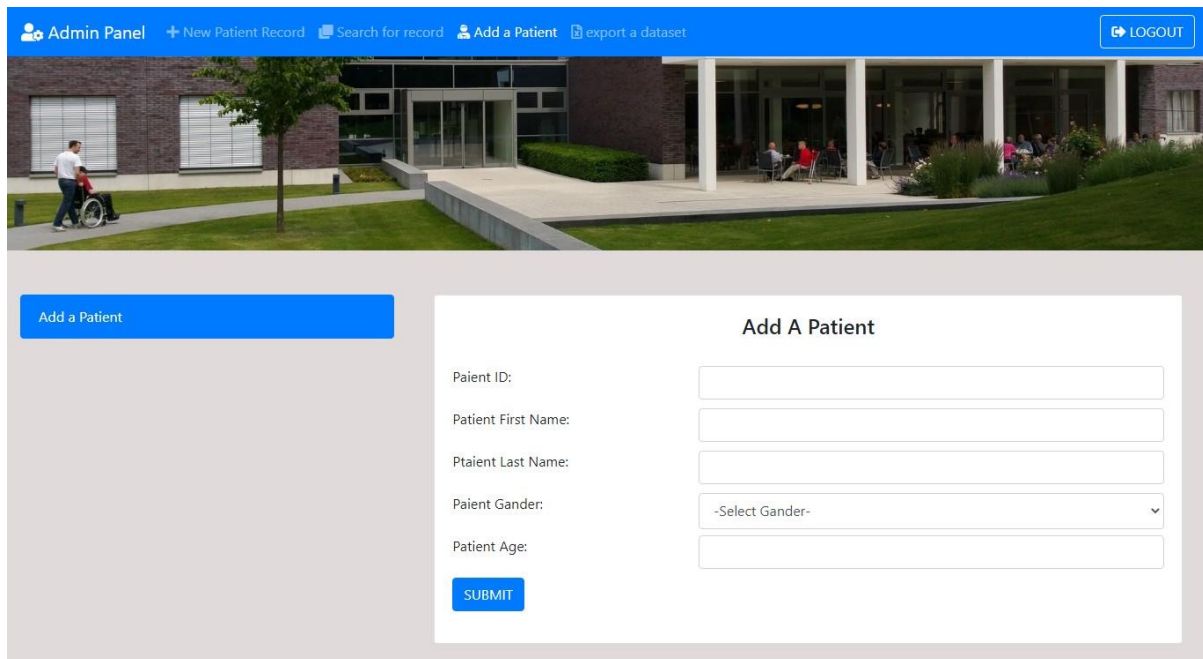


Figure 13: presentation of the “Add a Patient” content.

- **Export a Dataset:** this page allows users to export a dataset with anonymized patient information from the system. The exported dataset consists of all the patient records contained in the EHR cloud database. Figure 14 presents a description of this page's content.

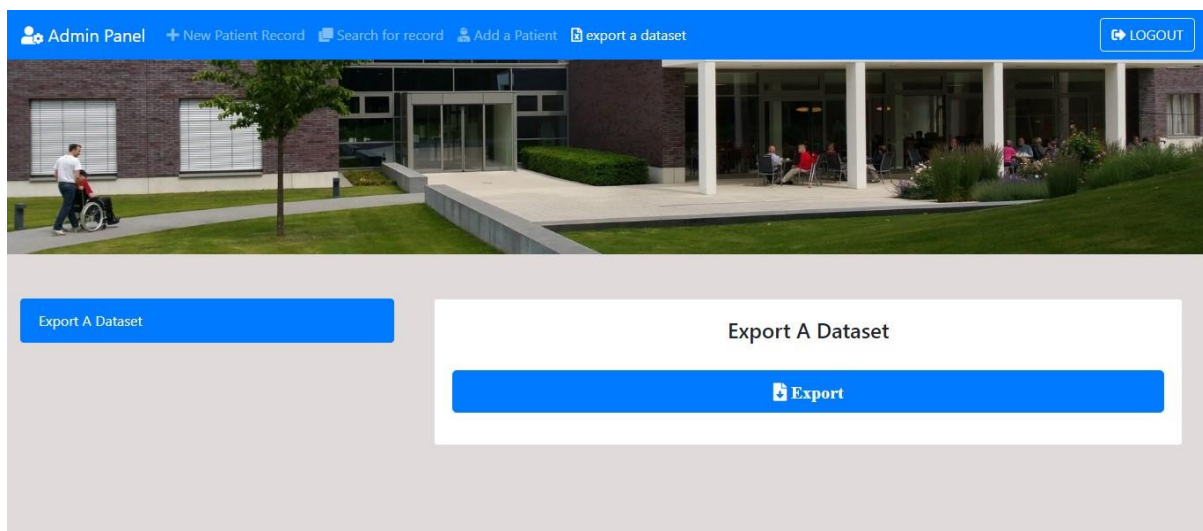


Figure 14: presentation of the “Export a Dataset” content.

3.2 Anonymity Preservation Mechanism (APM):

We wrote a prototype of our APM methods presented in the previous chapter using Python programming language. Our prototype consists of two primary methods; **1. Encrypt_fun():** applies our APM anonymization method to the initial patient record and generates an anonymized patient record as an output, **2. Decrypt_fun():** receive the anonymized patient record as an input and generate a de-anonymized patient record as an output. A demonstration of our APM prototype anonymization and de-anonymization methods is presented the python script below:

```
import bcrypt, rsa, AES          #import the necessary modules for the Method
from networkServer import *      #import the network server file

    //--- THE ANONYMIZATION METHOD ---\\

def encryption_fun(pID,bRecord,sKey,nsPublicKey,salt):

    #the declaration of the new anonymized patient record
    newRecord = json.loads('{ "Rid": "", "Pid": "", "aRecord": {}, "bRecord": {} }')

    #encrypt the generated random id and assign it to the "Rid" attribute of the aPR
    newRecord["Rid"] = b64encode(rsa.encrypt( randomID.encode('utf8'), nsPublicKey)).decode('utf8')

    #import the all the patient profiles in the patient table
    with open ('patientTable.json', 'r') as f:
        patients = json.loads(f.read())

    #Loop over each patient profile in the patient table
    for P in patients['patinetTable']:

        #check if the patient profile id match the received patient id
        if P['Pid'] == pID:

            #hash and encrypt the received patient id and assign it to the "Pid" attribute of the aPR
            pID = bcrypt.hashpw(pID.encode('utf8'), salt)
            newRecord["Pid"] = b64encode(rsa.encrypt(pID, nsPublicKey)).decode('utf8')

            #Loop over every attribute within the "aRecord" attribute of the patient profile
            for val in P['aRecord']:

                #hash the patient profile attribute and assign it to the "hashVal" variable using Bcrypt Hashing Function
                hashVal = bcrypt.hashpw(P['aRecord'][val].encode('utf8'), salt)

                #encrypt the "hashVal" variable and assign it to the aPR using RSA Asymmetric Encryption
                newRecord["aRecord"][val] = b64encode(rsa.encrypt( hashVal, nsPublicKey)).decode('utf8')
```

```

#Loop over every attribute within the "bRecord" attribute of the patient profile
for val in P['bRecord']:

    #assign the patient profile attribute to the received bRecord
    bRecord[val] = P['bRecord'][val]

    #break out from the loop
    break

#Loop over every attribute in the bRecord
for val in bRecord :

    #initialize the AES encryption mechanism and generate new cipher object of the bRecord attribute
    cipher = AES.new(sKey, AES.MODE_CFB)
    ct_bytes = cipher.encrypt(bRecord[val].encode('utf8'))

    #extract the Initialization Vector and ciphertext from the cipher object
    iv = b64encode(cipher.iv).decode('utf8')
    ct = b64encode(ct_bytes).decode('utf8')

    #create a new json object and assign it to the aPR attribute
    newRecord["bRecord"][val] = json.loads('{ "iv":"iv", "ciphertext":"ciphertext"}')

    #encrypt the iv and ct variable and assign it to the aPR using RSA Asymmetric Encryption
    newRecord["bRecord"][val]["iv"] = b64encode(rsa.encrypt(iv.encode('utf8'),nsPublicKey)).decode('utf8')
    newRecord["bRecord"][val]["ciphertext"] = b64encode(rsa.encrypt(ct.encode('utf8'),nsPublicKey)).decode('utf8')

#send the generated aPR to the TV method of the network server and return the result
return transactionValidator (newRecord)

```

//--- THE DE-ANONYMIZATION METHOD ---\\

```
def decryption_fun(Record,sKey,apmPrivateKey,salt):
```

```
    #import the all the patient profiles in the patient table
```

```
    with open ('patientTable.json', 'r') as f:
        patients = json.loads(f.read())
```

```
    #Loop over each patient profile in the patient table
```

```
    for P in patients['patinetTable']:
```

```
        #hash the patient profile attribute and assign it to the "ciphertext" variable using Bcrypt Hashing Function
```

```
        ciphertext = bcrypt.hashpw(P['Pid'].encode('utf8'), salt).decode('utf8')
```

```
        #encrypt the "Pid" attribute of the received received Record and assign it to the "Pid" variable using RSA Asymmetric Encryption
```

```

Pid = rsa.decrypt(b64decode(Record['Pid'].encode('utf8')),nsPrivateKey).decode('utf8')

#check if the "ciphertext" variable match the "Pid" variable
if ciphertext == Pid:

    #assign the patient profile id to the "Pid" attribute of the received Record
    Record["Pid"] = P['Pid']

    #Loop over every attribute within the "aRecord" attribute of the received Record
    for val in Record["aRecord"] :

        #assign the patient profile attribute to the received Record
        Record["aRecord"][val] = P["aRecord"][val]

    #break out of the Loop
    break

#Loop over every attribute within the "bRecord" attribute of the received Record
for val in Record["bRecord"] :
    #encrypt the iv and ct values of the "bRecord" attribute of the received Record using RSA
    iv = b64decode(rsa.decrypt(b64decode(Record["bRecord"][val]['iv'].encode('utf8')), apmPrivateKey).decode('utf8'))
    ct = b64decode(rsa.decrypt(b64decode(Record["bRecord"][val]['ciphertext'].encode('utf8')), apmPrivateKey).decode('utf8'))

    #initializes the AES decryption mechanism using the iv and the sKey
    cipher = AES.new(sKey, AES.MODE_CFB, iv=iv)

    #decrypt the "bRecord" attribute of the received Record using AES
    Record["bRecord"][val] = cipher.decrypt(ct).decode('utf8')

#return the decrypted record
return Record

```

- **Pid:** the patient id submitted from the user in the client application.
- **bRacord:** the Condition and Description attributes submitted from the user in the client application.
- **sKey:** the secret key of the APM used in the AES encryption and decryption process.
- **nsPublicKey:** the network server public key used in the RSA encryption process.
- **salt:** the APM hashing salt used by the Bcrypt hashing mechanism.
- **Record:** the anonymized patient record (*aPR*).
- **apmPrivateKey:** the APM private key used in the RSA decryption process.

3.3 Network Server:

We also used Python programming language to develop our network server. Our prototype of the network server act as a communication entity between the APM and the EHR Cloud Database. The network server receives the anonymized data from the APM and sends it to the cloud for storing. Similarly, the network server fetches data from the cloud database to the APM according to the anonymized query received from the APM.

```
import pymongo #import the mongodb module of python

#connect to the Atlas mongodb cluster
client = pymongo.MongoClient("mongodb+srv://adminEHR:<adminpassword>@ehrdatabase.aqnrs.mongodb.net/myFirstDatabase?retryWrites=true&w=majority")

#connect to the EHR database
db = client.EHR

#select the patient record collection
patientRecord = db.patientRecord
```

//--- THE TRANSACTION VALEDATOR METHOD---\\

```
def transactionValidator (aRecord):
    #check if the code bellow doesn't returns an error
    try:

        #decrypt the "Pid" attribute of the received received aRecord
        aRecord["Pid"] = rsa.decrypt(b64decode(aRecord["Pid"].encode('utf8')),nsPrivateKey).decode('utf8')

        #insert the received aRecord to the patient record collection of the EHR database in Atlas mongodb cluster
        patientRecord.insert_one(aRecord)

        return True

    #if the code above returns an error
    except:

        return False
```

- **aRecord:** the received anonymized patient record (*aPR*) from the APM.

3.4 EHR Cloud Database:

The EHR Cloud Database is developed using MongoDB Atlas, a global cloud database service containing a fully managed MongoDB and deployed to the AWS platform. This database contains a ledger of anonymized patient records, and it's directly connected to the network server. Similarly, the cloud allows the network server to fetch and store data from and into its database. Figur showcases the anonymized patient record contained in the EHR cloud database.

4 Results:

In this section, we are going to showcase the effects and changes that occur to a real patient record example after each phase of our prototype case scenarios.

- **Hash:** Represent the produced hash from Bcrypt hashing function.
- **Cipher Text (CT):** Represent the encrypted text using AES Symmetric Encryption.
- **Initialization Vector (IV):** Represent the random variable used in the encryption process.
- **Public Key Encryption (PKE):** Represent the encrypted text using RSA Asymmetric Encryption.

4.1 Initial Patient Record (1):

This Initial Patient Record (*iPR*) is the first type of initial patient record that is created in the system, and it's basically a copy of the original *PR* generated by the client application from the user after the treatment of the patient, it contains information of ID, Condition, and Description of that patient. Table 5 present the formation of the *iPR* created by the client application.

Table 5: *the first type of Initial Patient Record.*

ID	Condition	Description
AF89XI96FD24	Fever	“Seasonal fever associated with a cold”

$$iPR = \{ (ID) , (Condition, Description) \}$$

4.2 Initial Patient Record (2):

This is the second type *iPR* which is generated automatically by the APM in the first phase of anonymization, the APM includes information of the patient like Firstname, Lastname, Age, Gender to the received patient record using the ID. A demonstration of the formation of this type of *iPR* is presented in Table 6.

Table 6: the second type of Initial Patient Record.

ID	Firstname	Lastname	Age	Gender	Condition	Description
AF89XI96FD24	Ahmed	Ali	22	Male	Fever	“Seasonal fever associated ...”

$$iPR = \{ (ID, Firstname, Lastname) , (Age, Gender, Condition, Description) \}$$

4.3 Anonymized Patient Record (1):

The APM anonymization method generates this type of *aPR* after the first layer of anonymization. This *aPR* contains hashed version of the Identifiable Information using Bcrypt function and an encrypted version of the Non-identifiable Information using AES symmetric encryption. Table 7 present the formation of the *iPR* created by the client application.

Table 7: the first type of Anonymized Patient Record.

ID	Firstname	Lastname	
Hash="\$2b\$12\$zIVOnTr81y64OCP YOTbpHu2fSGG/kwlHwlfGwgLKx Ymh0cZyLYy9a"	Hash="\$2b\$12\$zIVOnTr81y64OCP YOTbpHupiaiFDa8unHHtGVeL8qN R0wXWyBRhFG"	Hash="\$2b\$12\$zIVOnTr81y64OCPYOTbpH uTq1L/gFJimlN60Xy5FiN1luFLtqeOAu"	
Age	Gender	Condition	Description
CT = "f3U="	CT="pzp6xw=="	CT="HCN4TSg="	CT="ubGDpaV2EXXuWw22avP/cgIjPQAhn gXTc+29acPzglWVXPRRNw=="
IV="3otJMbgaZXn/u 6JOmITotA=="	IV="b43ZEgfWl9FmW jC1ML+mFA=="	IV="TN/+s/MBHbE1sKbJ BgkMLA=="	IV=" k2jglkr5UT88H4sX0hWIVA=="

$$aPR = \{ \text{Bcrypt} (\{ID', Firstname', Lastname'\}, SR) , \text{AES} (\{Age', Gender', Condition', Description'\}, SK_{APM}) \}$$

4.4 Anonymized Patient Record (2):

This is the second type of *aPR*, and it's generated after the second layer of anonymization. In this phase, the anonymization method of the APM implements another layer of encryption to the first type of *aPR* using RSA asymmetric encryption. A demonstration of the formation of this type of *iPR* is presented in Table.

Table 8: the second type of Anonymized Patient Record.

ID	Firstname	Lastname	
PKE(Hash)=""TxBYAtom0QRIAsEb h1jCrQ1HncBQf+SAjvVWghOubd8 CmSFvK2Bw..."	PKE(Hash)=""ZsIzdSlS2Z96mDaP8t9lsEy KgtxXkNhyJxcsINTCR9J5cWHd/+zqR... "	PKE(Hash)=""LQj7JuLeuQzKTtHfUy/YI ekWIL0SgipoaqgV+sYXnOfUi+kvC0P 1S..."	
Age	Gender	Condition	Description
PKE(CT)=""gNB6hdCe 0oNXgJE4bbM5Fkpcg LMuDRV8I/st8T..."	PKE(CT)=""WslOUEtB Xf300pg1/PeA3VE6TP BFV0iT6RIj/uov..."	PKE(CT)=""QPWtEVbGFOlp 1xo3SFyXzIQGLzuS91G9SH 000w0f4AOGJ..."	PKE(CT)=""BoVm6b9N4mp3f/PpwoKW vmYGUFL1fTUI8JBzB0i..." PKE(IV)=""Y6Ie88R1rrHG/zsi3hyf9axEo hePV7Po++X1Br56ZLMhp4ymOUc9... "
PKE(IV)=""fXmnXnRu rlrimEWLzNVfUPMI FEJASdiAw..."	PKE(IV)=""YYO62i9im 2Df75HWjBYco6L/LJ VBz0yebSWzjCw..."	PKE(IV)=""cbGOENRatAJr5b psr0FQX1N7TdMFpXyzV00 QQQi5YI..."	

$$aPR = \{ \text{RSA}([\text{Bcrypt}(\{ \text{ID}', \text{Firstname}', \text{Lastname}' \}, \text{SR}) , \text{AES}(\{ \text{Age}', \text{Gender}', \text{Condition}', \text{Description}' \}, \text{SK}_{APM})], \text{PK}_{NS}) \}$$

5 Conclusion:

In this chapter, we presented a prototype of our proposed EHR management system. Our prototype aims to showcase a practical example of implementing our Anonymity Preservation Mechanism (APM). First, we presented the tools and methodologies used to build our prototype. Next, we discussed the different case scenarios of our prototype, covering the implementation details of each scenario. Then, we showcased the implementation results of our prototype.

GENERAL CONCLUSION

By the end of this study, and after performing the required research for this project which we covered in our thesis. We feel confident to say that we have achieved our objective from this study, which was to build and implement a secure privacy preservation mechanism that safely anonymizes patient identity in cloud-based EHR management systems by presenting our proposed Anonymity Preservation Mechanism (APM).

We divided our thesis into three-chapter; chapter one provides a theoretical foundation for us to have a broad understanding of the different technologies regarding this field of study. In chapter two, we introduced our APM anonymization mechanism and merged it into our integrated cloud-based EHR management system architecture. In chapter three, we showcased the implementation of the proposed prototype of our APM anonymization mechanism.

In conclusion of this study, we've managed to build a privacy preservation mechanism that anonymizes patient records in cloud-based EHR management systems.

FUTURE WORK

The proposed prototype on our integrated cloud-based EHR system helped us reach a very advanced stage into achieving our final objective, which is to build a secure and comprehensive platform that manages electronic health records. We would like to look at the possibility of implementing our integrated Private Blockchain network to validate transactions in the system using predefined nodes by executing a custom-made smart contract and a Proof-of-Work consensus mechanism generating the confidentiality of the process, and a distributed ledger stores the validated transaction. This Private Blockchain network can be implemented using an Ethereum Blockchain network. Another future possibility is to implement our APM anonymized mechanism in an electronic chip or small single-board computers such as Raspberry Pi, that way we can get access to the APM from multiple computers without the need to store it on every computer in the system, and we also attach it to the patient body sensors to anonymize the collected data.

REFERENCES

- [1] Hassan, M. U., Rehmani, M. H., & Chen, J. (2019). *Privacy preservation in blockchain based IoT systems: Integration issues, prospects, challenges, and future research directions*.
- [2] L. Atzori, A. Iera, G. Morabito, The Internet of Things: a survey 54 (2010).
- [3] A. Zanella, N. Bui, A. Castellani, L. Vangelista, M. Zorzi, Internet of things for smart cities, IEEE Internet Things J. 1 (1) (2014) 22–32.
- [4] Dian, F. J., Vahidnia, R., & Rahmati, A. (2020). *Wearables and the Internet of Things (IoT), Applications, Opportunities, and Challenges: A Survey*.
- [5] S. Rimer, An IoT architecture for financial services in developing countries, in: IEEE IST-Africa Week Conference, 2017, pp. 1–10.
- [6] Jesus, E. F., Chicarino, V. R. L., de Albuquerque, C. V. N., & Rocha, A. A. de A. (2018). *A Survey of How to Use Blockchain to Secure Internet of Things and the Stalker Attack. Security and Communication Networks, 2018*.
- [7] J. L. Catarinucci, D. De Donno, L. Mainetti, L. Palano, L. Patrono, M.L. Stefanizzi, L. Tarricone, An IoT-aware architecture for smart healthcare systems, IEEE Internet Things (2015).
- [8] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, 2008.
- [9] Hyperledger, 2021, (online), Available: <https://www.hyperledger.org>.
- [10] P. Otte, M. de Vos, J. Pouwelse, Trustchain: A Sybil-resistant scalable blockchain, Future Gener. Comput. Syst. (2017).
- [11] H.-T. Wu, C.-W. Tsai, Toward blockchains for healthcare systems: Applying the bilinear pairing technology to ensure privacy protection and accuracy in data sharing, IEEE Consumer Electron. Mag. (2018).
- [12] A. Zhang, X. Lin, Towards secure and privacy-preserving data sharing in e-health systems via consortium blockchain, J. Med. Syst. (2018).
- [13] Rahman, M. S., Khalil, I., Mahawaga Arachchige, P. C., Bouras, A., & Yi, X. (2019). *A Novel Architecture for Tamper Proof Electronic Health Record Management System using Blockchain Wrapper*.
- [14] Kanwal, T., Anjum, A., & Khan, A. (2020). *Privacy preservation in e-health cloud: taxonomy, privacy requirements, feasibility analysis, and opportunities. Cluster Computing*.
- [15] Mahanan, Waranya, W. Art Chaovalitwongse, and Juggapong Natwichai. "Data anonymization: a novel optimal k-anonymity algorithm for identical generalization hierarchy data in IoT." *Service Oriented Computing and Applications* 14.2 (2020)
- [16] Riedl, B., Graser, V., Fenz, S., Neubauer, T.: Pseudonymization for improving the privacy in e-health applications. In: Proceedings of the Annual Hawaii International Conference System Sciences, pp. 1–9 (2008)
- [17] Donald E. Knuth. 1998. *The art of computer programming, volume 3*.p:513
- [18] A. Reyna, C. Martín, J. Chen, E. Soler, M. Díaz, On blockchain and its integration with IoT. Challenges and opportunities, Future Generation Computer Systems (2018).