Securing Medical Records Using Blockchain with Cryptography, Encryption, and Zero-Knowledge Rollups

MSc Research Project
Cloud Computing

Darshika Pongallu
Student ID: x22197222

School of Computing
National College of Ireland

Supervisor: Shreyas Setlur Arun

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Darshika Ravindra Pongallu
Student ID:	x22197222
Programme:	Cloud Computing
Year:	2023-2024
Module:	MSc Research Project
Supervisor:	Shreyas Setlur Arun
Submission Due Date:	12/08/2024
Project Title:	Securing Medical Records using Blockchain with Cryptography, Encryption and Zero-Knowledge Rollup
Word Count:	980
Page Count:	5

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Darshika Ravindra Pongallu
Date:	9th August 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Securing Medical Records Using Blockchain with Cryptography and Zero-Knowledge Rollups

Darshika Pongallu
x22197222

1 Introduction

This manual provides a step-by-step guide for implementing a system to secure medical records using blockchain technology, cryptography, and zero-knowledge rollups. It also includes the essential setup instructions and configurations needed for the tools required in the research. The primary goal of this document is to offer clear instructions and details necessary to execute the code included in the research project submission.

2 Pre-requisites

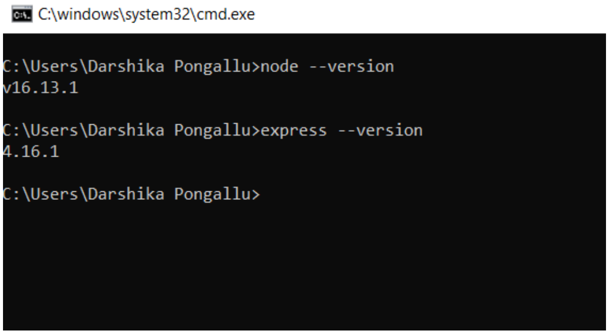
To follow this configuration manual, the user should have a basic understanding of web development programming languages, as well as familiarity with Linux, cloud computing, blockchain, and Solidity. Additionally, it is essential for the user to have the latest versions of Node, Express, Git, and React installed.

3 Minimum System Requirements

- **Operating System-** Windows 10 Home Single Language
- **Processor-** Intel Core I5 with 4GB NVIDIA Graphics Card
- **Installed Memory (RAM)-** 16 GB
- **System Type-** 64-bit Operating System, x64-based Processor
- **Storage Capacity-** 1 TB Hard Disk

4 Software Requirements

- Download and Install Visual Studio Code IDE.
- Download and install NodeJs and ExpressJs on your local machine to run the application locally.
  - `npm i`
  - `npm start`



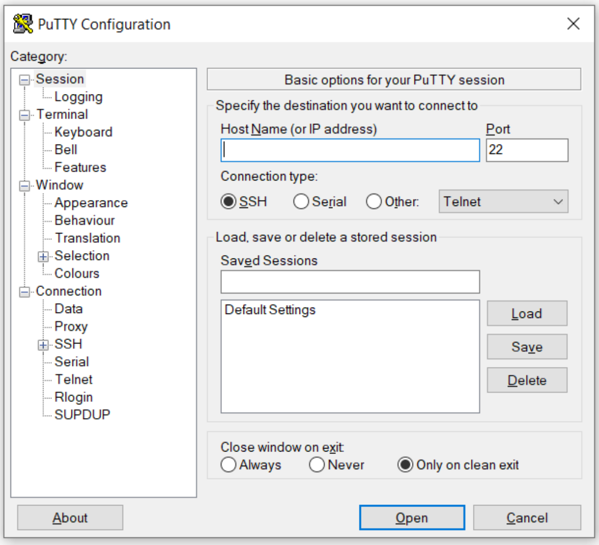
C:\windows\system32\cmd.exe

```
C:\Users\Darshika Pongallu>node --version
v16.13.1

C:\Users\Darshika Pongallu>express --version
4.16.1

C:\Users\Darshika Pongallu>
```

Figure 1: NodeJs & ExpressJS Installation

- `nodemon app.js`

- NTRU lattice based cryptography algorithm and KMS encryption algorithm will be written in JavaScript file and Zero Knowledge Rollups will be written in solidity programming language.
- Create a free tier AWS Cloud account.
- For connecting AWS EC2 instances locally, download and install putty software.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address):

Port: 22

Connection type: ● SSH ○ Serial ○ Other: Telnet

Load, save or delete a stored session

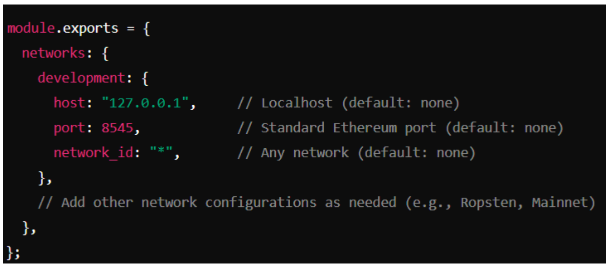
Saved Sessions: Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 2: Establishing connection to EC2 instances

- Create 2 instances i.e. blockchain and backend.
- Connect the backend instance locally using putty software and download MongoDB.
  - `wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc | sudo tee /usr/share/keyrings/mongodb-server-6.0.asc`
  - `echo "deb [ signed-by=/usr/share/keyrings/mongodb-server-6.0.asc ] https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/6.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-6.0.list`

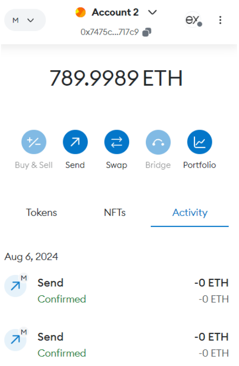
  - `sudo apt-get update`
  - `sudo apt-get install -y mongodb-org`
  - `sudo systemctl start mongod`
  - `sudo systemctl status mongod`
  - `sudo systemctl enable mongod`
- Download ganache locally and test the smart contracts by creating a work space.
- Deploy those smart contracts on blockchain instance.
  - Install Truffle: `npm install -g truffle`
  - Navigate to your project directory.
  - Compile the contracts using the command: `truffle compile`
  - Configure the deployment network.
  - Deploy smart contracts: `truffle migrate --network development`
  - Verify and interact with the deployed contracts: `truffle console --network development`
  - Test deployed contracts: `truffle test`



```
module.exports = {
  networks: {
    development: {
      host: "127.0.0.1",     // Localhost (default: none)
      port: 8545,            // Standard Ethereum port (default: none)
      network_id: "*",       // Any network (default: none)
    },
    // Add other network configurations as needed (e.g., Ropsten, Mainnet)
  },
};
```

Figure 3: SmartContacts Configuration

- Further download docker on blockchain instance.
  - `sudo apt-get update`
  - `sudo apt-get install -y apt-transport-https ca-certificates curl software-properties-common`
  - `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee /usr/share/keyrings/docker-archive-keyring.gpg > /dev/null`
  - `sudo apt-get update`
  - `sudo apt-get install -y docker-ce docker-ce-cli containerd.io`
  - `sudo docker --version`
  - `sudo systemctl start docker`
  - `sudo systemctl enable docker`
  - `sudo usermod -aG docker $USER`



Account 2

0x7475c...717c9

789.9989 ETH

Aug 6, 2024

Send Confirmed -0 ETH -0 ETH

Send Confirmed -0 ETH -0 ETH

Figure 4: Metamask Wallet

- Download and Install Metamask Wallet. Further create an account and add Ethereum test ethers from Ethereum test faucet. Refer Fig. 4
- AWS Services to be used are AWS S3, AWS EC2, AWS Lambda, AWS Cloud Watch, AWS CloudFormation, AWS KMS, AWS IAM.

5 References

HTML.com (n.d.) *HTML*. Available at: https://html.com/ [Accessed 8 August 2024].

W3.org (n.d.) *CSS Overview*. Available at: https://www.w3.org/Style/CSS/Overview.en.html [Accessed 8 August 2024].

W3Schools (n.d.) *JavaScript Introduction*. Available at: https://www.w3schools.com/js/js_intro.asp [Accessed 8 August 2024].

React.dev (n.d.) *React Documentation: Learn*. Available at: https://react.dev/learn [Accessed 8 August 2024].

Node.js (n.d.) *Node.js Documentation*. Available at: https://nodejs.org/docs/latest/api/ [Accessed 8 August 2024].

Express.js (n.d.) *Installing Express*. Available at: https://expressjs.com/en/starter/installing.html [Accessed 15 May 2024].

MongoDB (n.d.) *MongoDB Documentation*. Available at: https://www.mongodb.com/docs/ [Accessed 15 May 2024].

Amazon Web Services (n.d.) *Amazon S3 User Guide*. Available at: https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html [Accessed 20 May 2024].

Amazon Web Services (n.d.) *Amazon EC2 Documentation*. Available at: https://docs.aws.amazon.com/ec2/ [Accessed 20 August 2024].

Amazon Web Services (n.d.) *Amazon API Gateway Developer Guide*. Available at: https://docs.aws.amazon.com/apigateway/latest/developerguide/welcome.html [Accessed 20 May 2024].

Amazon Web Services (n.d.) *AWS Lambda Developer Guide*. Available at: https://docs.aws.amazon.com/lambda/latest/dg/welcome.html [Accessed 20 May 2024].

Amazon Web Services (n.d.) *Amazon CloudWatch Documentation*. Available at: https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html [Accessed 20 May 2024].

Amazon Web Services (n.d.) *AWS CodePipeline User Guide*. Available at: `https://docs.aws.amazon.com/codepipeline/latest/userguide/welcome.html` [Accessed 20 May 2024].

Solidity (n.d.) *Solidity Documentation v0.8.26*. Available at: `https://docs.soliditylang.org/en/v0.8.26/` [Accessed 15 June 2024].

MetaMask (n.d.) *MetaMask Documentation*. Available at: `https://docs.metamask.io/` [Accessed 5 July 2024].

Truffle Suite (n.d.) *Ganache Documentation*. Available at: `https://archive.trufflesuite.com/docs/ganache/` [Accessed 5 July 2024].