WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Decentralized Task Offloading with Block Chain Technology in Vehicular Networks

Dr. M.V. Vijaya Saradhi[1], Fahmida Ashraf[2], M. Namratha[3], Jimka Varshini[4], Rithvik kodisela[5]

[1]Professor and Head of CSE Dept, ACE Engineering College Hyderabad, India hodcse@aceec.ac.in
[2]Student CSE ACE Engineering College Hyderabad, India fahmidaashraf84@gmail.com
[3]Student CSE ACE Engineering College, Hyderabad, India namrathashrutisha@gmail.com
[4]Student CSE ACE Engineering College Hyderabad, India varshinivarshini783@gmail.com
[5]Student CSE ACE Engineering College, Hyderabad, India rithvikkodisela001@gmail.com

ABSTRACT :

Smart cities are developing rapidly. Secure data transmission between different objects is a vital component of the modern smart city. Therefore, communication between different entities, such as vehicle and smart devices, can be considered an important element of contemporary smart cities. Vehicle Ad Hoc Network is a mobile ad hoc network for vehicle environments in smart cities. As the requirements for convenient, safe and efficient transportation continue to increase, the mutual communication between connected vehicles in VANET plays an irreplaceable role. First, to achieve consensus in the vehicular environment, we propose a distributed hierarchical software-defined VANET framework to establish a security architecture. Secondly, to improve the security of offloading, we propose to use blockchain-based access control, which protects the cloud from illegal offloading actions. Finally, to solve the intensive computing problem of authorized vehicles, we determine task offloading via jointly optimizing offloading decisions, consensus mechanism decisions, allocation of computation resources and channel bandwidth. The optimization method is designed to minimize long-term system of delays, energy consumption, and flow costs for all vehicles. To better resolve the proposed offloading method, we develop a new deep reinforcement learning algorithm via utilizing extended deep

Keywords: vehicular network router, VANET network, DQN algorithm, SDV framework, Connected Autonomous Vehicles

I. INTRODUCTION

The decentralized and reliable block chain combined with the distributed SDVs system to ensure security such as secure access control and resource allocation management between vehicle system. In particular, smart contract [15] is a computer program that runs on the block chain background. Its feasibility has been confirmed by various vehicle network security issues. For instance, smart contracts have been proven to have access control capabilities in vehicle networks, provide access verification and data auditing [16]. In addition, smart contracts can protect cloud resources from malicious access [17]. Therefore, blockchain and smart contracts are considered to be applicable to vehicle networks, especially ECCO systems that can achieve the security goal of mobile task offload.

II. OBJECTIVE OF THE PROJECT

First, to achieve consensus in the vehicular environment, we propose a distributed hierarchical software-defined VANET (SDVs) framework to establish a security architecture. Secondly, to improve the security of offloading, we propose to use blockchain-based access control, which protects the cloud from illegal offloading actions. Finally, to solve the intensive computing problem of authorized vehicles, we determine task offloading via jointly optimizing offloading decisions, consensus mechanism decisions, allocation of computation resources and channel bandwidth.

III. METHODOLOGY

Review inputs and outputs for project activities. Information will be collected and prioritized. An appropriate algorithm or framework has been selected. Several estimation algorithms will be compared and the best method will be selected. Software and hardware selection will be made according to the needs. Data will be used as a process or framework

IV. LITERATURE SURVEY

V2V has the goal to facilitate efficient and reliable communication without predominantly relying on global system for mobile communication (GSM) network. It is widely recognized that tradition communication systems like Vehicle-to-Infrastructure (V2I) relies its core communication using third party infrastructure like GSM. Third party infrastructure bandwidth is limited and sometimes not adequate for users' needs and security constraints with regard to their unreliability and not available everywhere. [1]

Everyone nowadays requires the assurance of safer transportation. Vehicle-to-vehicle communication is becoming more popular as computer technology progresses. Vehicle-to- Vehicle Connectivity might assist in obtaining it. The primary objective for car-to-car communication systems is safety and preventing crashes. [2]

This paper specially focuses on the special features and applications supported by connected vehicles (CV) for intelligent transportation systems. CV systems use connectivity (via wireless communications), positioning (via Global Positioning System and digital maps), and data processing to enable vehicles, smart roadway infrastructure, and personal mobile devices to exchange information with each other and to provide road users with both safety and mobility advisories, warnings, and alerts.[3]

Cloud computing has a decentralized architecture in which virtual machine migration is Connected and autonomous vehicles (CAVs) are enabled by various wireless communication devices, processing, and storage modules to enable a safe, efficient, and comfortable experience. In addition, CAVs are embedded with various sensors, navigation systems, data acquisition, and big data analytics to make them more intelligent.[4]

This paper proposes cooperation between MEC and central cloud decisions for different vehicular application offloading. We formulate a new resource allocation problem to guarantee the required response time. To solve such an NP-hard problem, we utilize deep reinforcement learning.[5]

V. PROPOSED SYSYTEM

The system proposes a new secure computation offloading framework for a blockchain-based VANET network, in which a mobile vehicle can offload its tasks to a cloud or edge server to perform computation under an access control mechanism. The system has designed a hierarchical architecture of controllable programming derived from SDN. The system has proposed a trusted access control mechanism that can use smart contracts on the blockchain to effectively detect and prevent illegal offloading of VANET devices. The system proposes a dynamic offloading solution that considers offloading data size, available MEC computing power, throughput and bandwidth resources to offload its resource to the cloud or edge server. The system verifies the proposed ECCO system via simulation experiments, and then investigates the access control and offloading performance.

VI. HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENTS:

- Processor: Min. Core i3 processor
- RAM: 4GB(min)
- Hard Disk Space: 20GB

SOFTWARE REQUIREMENTS:

- Programming Language: JAVA
- Operating System: Windows 7 & above
- FRONT END: J2EE
- BACK END: MySQL

VII. MODULES USED

Source

In this module, the Source browses the required file, initializes nodes with Power and uploads to the destination via Vehicular Network Router

Vehicular Network Router

The Vehicular Network Router is responsible for forwarding the data file in shortest distance to the destination; the Vehicular Network Router consists number of blocks and the power of node is checked in each block and then forwards to destination. If block finds any malicious or less power node in the router then it forwards to its corresponding block. In Vehicular Network Router the system can assign the power for the node and can view the node details with their power status and attacked status.

Destination

In this module, the destination can receive the data file from the Vehicular Network Router which is sent via Vehicular Network Router, if malicious or less power node is found in the Vehicular Network Router then it never forwards to the Destination to filter the content and adds to the attacker profile.

Attacker

In this module, the malicious node or the node details can be identified by a threshold-based classifier is employed in the Attack Detection module to distinguish DoS attacks from legitimate power of each node. The Attacker can inject the less power and generates the node to drop from the corresponding block node.

VIII. IMPLEMENTATION

We propose a distributed hierarchical software-defined VANET (SDVs) framework to establish a security architecture. Secondly, to improve the security of offloading, we propose to use blockchain-based access control, which protects the cloud from illegal offloading actions. Finally, to solve the intensive computing problem of authorized vehicles, we determine task offloading via jointly optimizing offloading decisions, consensus mechanism decisions, allocation of computation resources and channel bandwidth.

IX. SOURCE CODE:

```
import java.awt.Color;
import java.awt.Container;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedInputStream;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.PrintStream;
import java.math.BigInteger;
import java.net.Socket;
import java.security.DigestInputStream;
import java.security.MessageDigest;
import java.sql.Connection;
import java.sql.Statement;

import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;


public class Source extends JFrame implements ActionListener
{
        Container c;

        JButton browse,submit,init;

        int y=0;
        String dest;
```

```
JTextArea tf;
JScrollPane jp;

String filename,mac;
String content;
String n111;
String n112;
String n113;

Source()
{

        setTitle("Source::Blockchain Based Secure Computation Offloading in Vehicular Networks");
        c=getContentPane();
        c.setLayout(null);
        c.setBackground (Color.WHITE);

        tf= new JTextArea();
        tf.setRows(20);
        tf.setColumns(20);
        jp = new JScrollPane();
        jp.setViewportView(tf);
        jp.setBounds(50,200,300,200);
        c.add(jp);
        browse = new JButton("Browse");
        browse.setBounds(50,500,150,30);
        browse.addActionListener(this);
        c.add(browse);

        ImageIcon banner = new ImageIcon(this.getClass().getResource("ServiceProvider.png"));
        JLabel title=new JLabel();
        title.setIcon(banner);
        title.setBounds(5, -10, 900, 100);

        ImageIcon banner1 = new ImageIcon(this. getClass().getResource("SenderBack.png"));
        JLabel title1=new JLabel();
        title1.setIcon(banner1);
        title1.setBounds(375, 100, 500, 500);




        submit = new JButton("Encrypt and Upload");
        submit.setBounds(210,500,200,30);
        submit.addActionListener(this);
        c.add(submit);
        c.add(title);
        c.add(title1);
        setSize(700,600);
        setVisible(true);
}

public void actionPerformed(ActionEvent e)
{
```

```
if(e.getSource()==submit)
{
        String routerip = JOptionPane.showInputDialog("Enter Router IP-Address");
        String destip = JOptionPane.showInputDialog("Enter Destination IP-Address");

        try
        {
                AES enc=new AES();
                String keyWord = "ef50a0ef2c3e3a5fdf803ae9752c8c66";
                int len=content.length();
                String length=Integer.toString(len);

                Socket sc = new Socket(routerip,202);
                DataOutputStream dout = new DataOutputStream(sc.getOutputStream());
                dout.writeUTF(enc.encrypt(tf.getText(),keyWord));


                DataInputStream din = new DataInputStream(sc.getInputStream());
                String msg = din.readUTF();
                JOptionPane.showMessageDialog(null, msg);

        }
        catch(Exception e1)
        {
                e1.printStackTrace();
        }




}



if(e.getSource()==browse)
{
        try
        {
                //MessageDigest md = MessageDigest.getInstance("SHA1");
        JFileChooser jf = new JFileChooser();
        jf.showOpenDialog(browse);
        File f=jf.getSelectedFile();
        String fname=f.getName();


        FileInputStream fin = new FileInputStream(f);
        byte[] b = new byte[fin.available()];
        fin.read(b);
        String cont=new String(b);

        PrintStream pin = new PrintStream(new FileOutputStream("Owner/"+fname));
        pin.print(cont);


         content = new String(b);
        tf.setText(content);
```

```
                        filename=f.getName();

                        }
                        catch(Exception e1)
                        {
                                e1.printStackTrace();
                        }

                }


        }

        public static void main (String [] args) {
                new Source ();
}}
```

X. OUTPUT SCREENS

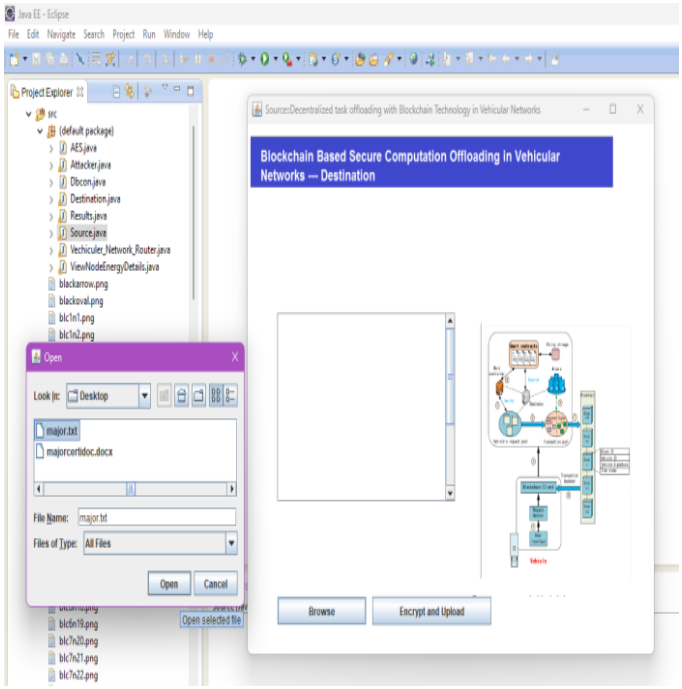

Fig 10.1: Browsing file

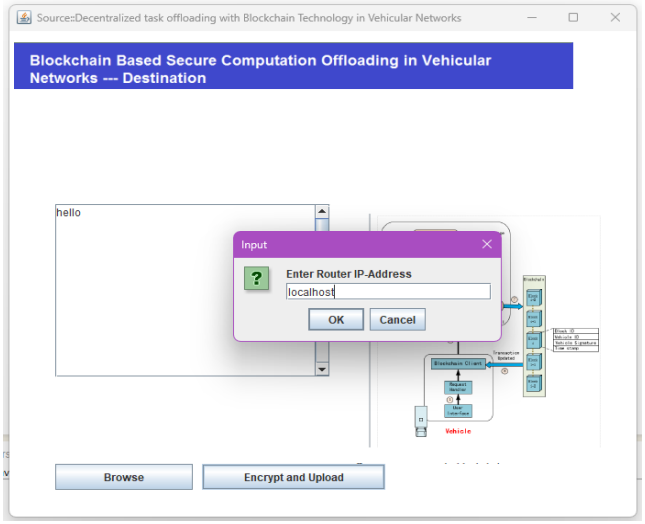

Fig 10.2: Entering router IP address

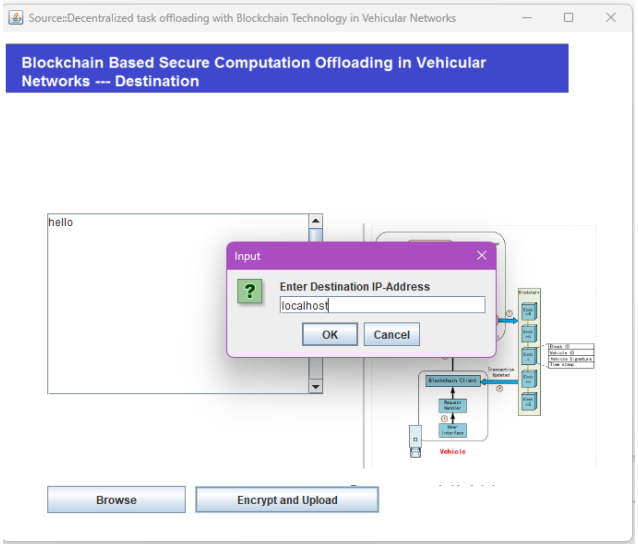

Fig 10.3: Entering destination IP address

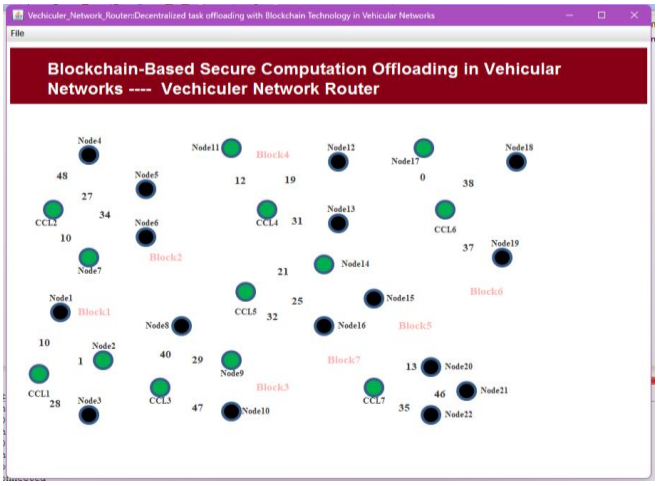

Fig 10.4: Data transfer between nodes

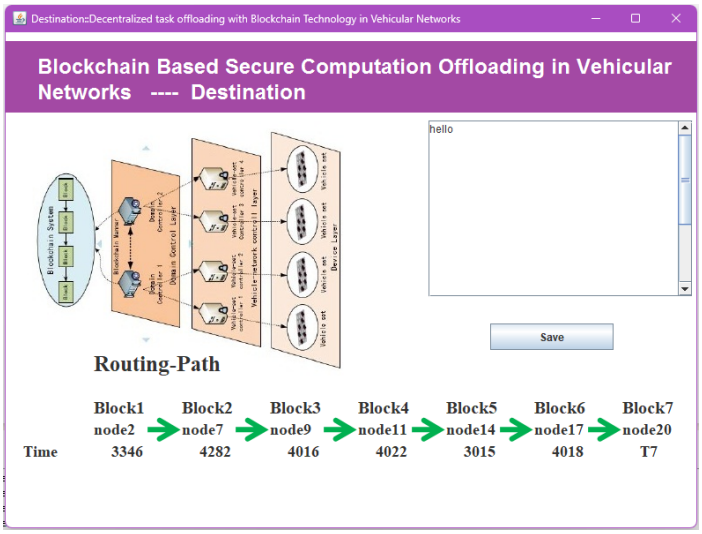

Fig 10.5: Displaying time consumed to send data

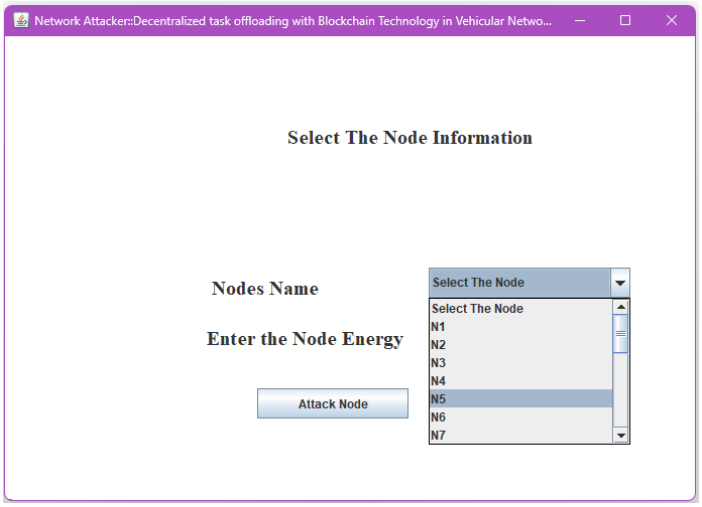

Fig 10.6: Attacking the node

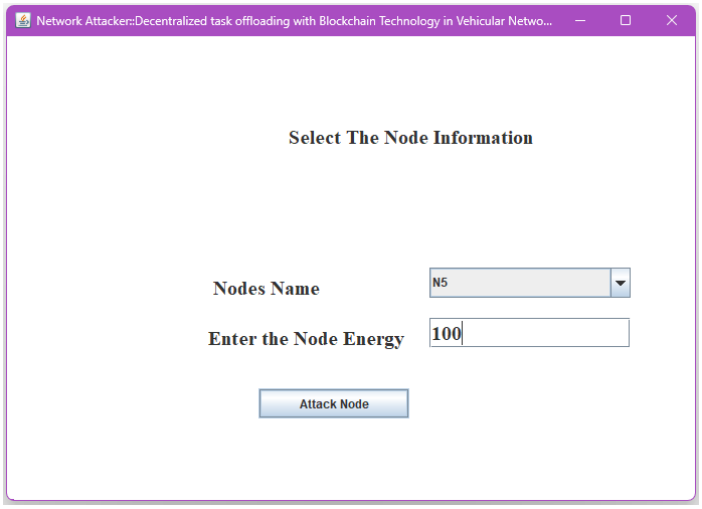

Fig 10.7: Entering node value

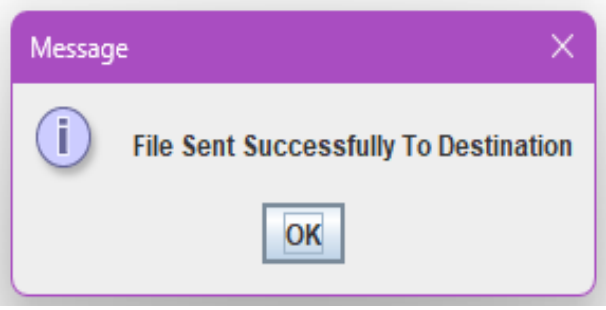

Fig 10.8: File sent to destination

XI. CONCLUSION

In this, we combine block chain and DRL for the ECCO system in the VANET network, and jointly investigate access control and computation offloading. We consider a general VANET scenario where multiple vehicles can offload their tasks to an edge or cloud server for collaborative performance. Then, we designed a hierarchical distributed software-defined VANET (SDVs) framework based on the block chain. First, to improve the security of task offloading, we propose an access control enabled by smart contracts and block chain to manage vehicle access to prevent malicious offloading access. We then propose a new DRL-based offloading scheme to achieve the optimal offloading strategy for all vehicles in VANET. We use the extended DQN algorithm to formulate task offloading decisions, consensus mechanism decisions, and edge resource as well as bandwidth allocation as joint optimization problems to minimize the total offloading cost of computation latency, throughput and energy consumption. We conducted an experimental simulation to evaluate the effectiveness of the proposed scheme. The results show that, compared with other benchmark methods, our scheme provides high security for the ECCO system and achieves performance improvements with minimum offloading costs. In the future, we will consider designing light-weight block chains so that the access control architecture is devised and arrayed directly at the edge side. It will hopefully support time-sensitive network management services for offloaded systems.

XII. FUTURE SCOPE

The Mobile vehicle initializes the demand task as an offloading transaction and performs computation offloading to the edge cloud server. The Block chain control end processes the demand information and sends it to the storage pool for smart contract verification. The main controller collects demands for mobile vehicles in the storage pool on a first come first served basis. The main controller verifies the demand through a smart contract with a control strategy. When the demand is received, the reaction is returned to the mobile vehicle to offload the data.

XIII. REFERENCES

[1] Demba, A., & Möller, D. P. (2018, May). Vehicle-to-vehicle communication technology. In 2018 IEEE international conference on electro/information technology (EIT) (pp. 0459-0464). IEEE.

[2] Ahmed, Manzoor, et al. "A survey on vehicular task offloading: Classification, issues, and challenges." Journal of King Saud University-Computer and Information Sciences 34.7 (2022): 4135-4162.

[3] Karimi, Elham, Yuanzhu Chen, and Behzad Akbari. "Task offloading in vehicular edge computing networks via deep reinforcement learning." Computer Communications 189 (2022): 193-204.

[4] Takai, I., Harada, T., Andoh, M., Yasutomi, K., Kagawa, K., & Kawahito, S. (2014). Optical vehicle-to-vehicle communication system using LED transmitter and camera receiver. IEEE photonics journal, 6(5), 1-14.

[5] Torrent-Moreno, M., Mittag, J., Santi, P., & Hartenstein, H. (2009). Vehicle-to-vehicle communication: Fair transmit power control for safety-critical information. IEEE transactions on vehicular technology, 58(7), 3684-3703.

[6] Matolak, D. W. (2008). Channel modeling for vehicle-to-vehicle communications. IEEE Communications Magazine, 46(5), 76-83.

[7] Samantaray, Rashmi Rani, et al. "Vehicle-to-Vehicle Communication Using RF Technology." 2023 International Conference on Smart Systems for applications in Electrical Sciences (ICSSES). IEEE, 2023.

[8] Yu, F. Richard. "Connected vehicles for intelligent transportation systems [guest editorial]." IEEE Transactions on Vehicular Technology 65.6 (2016): 3843-3844.