**Internship Report**

**on**

**TOPIC**

**Design a secured privacy preserving mechanism using, Biometric block chain of each mobile node in VANET for Smart Traffic Management.**

*Submitted in fulfilment of the requirement for the award of the degree of*

**Internship**

in

Computer Science and Engineering

By

Anjali Pathak

(Uni Roll No: 2018006)

Akshat Jain

(Uni Roll No: 1918006)

Under the supervision of

Ms. Priya Kholi



**What is VANET Environment?**

VANETs (vehicular ad hoc networks)

The possibility of enabling communication between vehicles through a wireless network stimulated the creation of new protocols, devices, and diverse utilization scenarios. Due to the difficulties of using a real time execution due to authentication and privacy issues, several majors are taken to provide the wireless network for smart trafficking to minimise the causes of accidents .

VANET aims to ensure safe driving by improving the traffic flow and therefore significantly reducing car accidents. This is possible by providing appropriate information to the driver or to the vehicle.

Due to the easiness of embedding computers in vehicles, it is not farfetched to imagine in the forthcoming years that most of the vehicles will be equipped with an on-board wireless device (OBU), GPS (Global Positioning System), EDR (event data recorder), and a multitude of sensors. However, it is important to note that, since VANET aims to ensure the safety of its users on the road, any delayed communication or defective level of implementation may affect people's lives. Therefore, any feature provided by a VANET protocol must be properly tested and validated.

Vehicles participating in a VANET are equipped with a set of wireless sensors and On Board Units (OBUs). Those units allow wireless communication between the vehicles and their environment. These devices make each vehicle act as packet sender, receiver, and router. It enables the vehicles to send and receive messages to other vehicles or Road Side Units (RSUs) within their reach via wireless medium.Crash avoidance safety system programs. Also known as mobile Ad Hoc Network is a type of local area network is a type that is built spontaneously to enable connection without a central device(router)

Active Security and intelligent transportation are important applications of VANET which needs suitable vehicle-to-vehicle communication technology.

With the Advancement and Maturity of VANET, there will be great revolution in the field of wireless communication in terms of network availability, security, safety with the use of advanced applications etc.

Consisting of a Network of Vehicles, moving at relatively high speed, which communicates among them with different purposes, being the main purpose, that of improving security on the Road.

VANETs can increase road safety by sharing information about both vehicles (e.g., position, speed and direction) and traffic conditions (e.g., accidents, jams, aquaplaning) as beacons. Safety messages, in this case, are the key information to avoid accidents. However, collisions can occur when safety messages and transmission of packets are improperly broadcast from multiple vehicles.

**MOBILE Nodes in VANET**

The Vehicular Ad-Hoc Network, or VANET, is a technology that uses and moves cars as nodes in a network to create a mobile network. VANET is a special form of

MANET (mobile ad hoc network).The Vehicular Ad-Hoc Network, or VANET, is a technology that uses and moves cars as nodes in a network to create a mobile network. VANET is a special form of MANET (mobile ad hoc network).

The vehicle is an entity in VANET and the nodes are limited to a particular type of topology while in motion which is the road topology. The nodes can provide power for data

processing and information transmission to sustain the functioning of the node Hence the communication is provided in two different ways in VANET, there are some fixed node that act as a roadside unit or equipment which enables the ease of VANET to serve as a gateway to the internet and also in accessing geographical data .

Each node in the VANET doesn’t only participate in data transmission and receiving, they also perform as a wireless router of the network as different nodes communicate via their separate communication range, permitting cars in the region of 100 to 300 meters of each other to join the network, and build a network with a wide range. As cars move apart from the signal range and drop out of the network, other cars can join in, connecting vehicles to one another so that a mobile Internet is created.

Traffic can be optimized by the use of sending signals like jam, accidents etc. to the vehicles so that they can choose their alternative path and can save time.

**VANET ALGORITHM**

Read about various algorithms like Consensus Algorithm

RSA ,DSA ,AES SHA-256 ,Hash Functions, PBFT,SD2PA Scheme proxy re-signature technology is proposed.

<https://www.researchgate.net/publication/327366416_A_Research_Paper_on_Vehicular_Ad-Hoc_Network>

<https://journal-bcs.springeropen.com/articles/10.1186/s13173-021-00113-x>

<https://link.springer.com/chapter/10.1007/978-3-642-22543-7_34>

<https://www.sciencedirect.com/science/article/pii/S2352864818303092>

**MODEL**

**Design a secured privacy preserving mechanism using, Biometric block chain of each mobile node in VANET for Smart Traffic Management.**

The three components of this model are Trusted Authority, Static Nodes (Towers), Mobile nodes (Vehicles) which will interact with each other by various mechanisms to let the model work.

Firstly, the information is collected and verified by the Authority by three main Privacy Mechanisms using **Biometric** Technology.

1.By using the **Fingerprinted user** for verification purposes.

2. In Case of any defect or in Exceptional case if the Fingerprint of user does not work Then we use the **Retina of the Eye** for the unique verification of the User Identity.

3. Last for Additional Security we use the **OTP** (one time Password) for the final Verification.

We explore the scope of a block chain for VANET. We make use of a private block chain for authentication and a public block chain for managing messages. The Legal Authority is responsible for making the transactions in the private block chain , the transactions are the identity information required for authentication of vehicles when they join for the first time. TA can identify the message sender vehicle whenever it is required For example, if malicious vehicles start to disturb the system (e.g., sending malicious messages), then the TA should be able to trace this vehicle and take a proper revocation action required.

* The TDE (Transparent Database Encryption Technique) solves the problem of protecting data at rest, encrypting databases both on the hard drive and consequently on backup media. This will work for rest data

So, when the data is stored, it would be encrypted which reduces its chances of any malicious attack.

Secondly, then we have Static nodes which will get the public and Private keys by the Trusted Authority. Static nodes will provide the Infrastructure to Vehicle Interaction for the working for Mobile Nodes in the Vanet. The mobile node which passes the authentication and verification at every static node would only allow entering the VANET.

These Static Nodes will take care of the Mobile Nodes in a specific Distance Range. This will use the Public Keys (Pseudo Identity) for the verification and let the information transfer to other vehicles about the traffic, Accident in that Range For the betterTraffic Management While the private key generator (PKG) generates the private keys.

We use two Algorithms for the Encryption and Decryption of the public and private key.

This is two steps where

Public Key Cryptography- use the linear block cipher where algorithm Used is DES 4.

Private key Cryptography – simple Symmetric Algorithm

For example – AES Algorithm.

Idea- will use the Hybrid of both the Algorithms to improve the Efficiency of the all over model. These Algorithms would provide privacy at both ends.

References:

<https://link.springer.com/article/10.1186/s13673-020-00241-x>

<https://www.researchgate.net/publication/326889573_A_Privacypreserving_Trust_Model_based_on_Blockchain_for_VANETs>

<https://dl.acm.org/doi/abs/10.1145/3377458.3377466>

<https://www.hindawi.com/journals/scn/2021/9942632/>

<https://www.mdpi.com/2073-8994/12/6/1003/htm>

<https://www.sciencedirect.com/science/article/abs/pii/S0306457320309559?via%3Dihub>

DATE: 02-07-21

**Combination of RSA and DSA Algorithm for Privacy and Authentication**

1. **About RSA**

It is an asymmetric cryptography algorithm. Works on Public and Private Key Public Key is given to everyone, and Private key is kept private.

It is based on the fact that it is difficult to factorize a large integer.

The public key consists of two numbers and multiplication of two large prime numbers. And Private Key is also derived from the same two prime numbers.

Encryption is depending on the size of the key. The key size, the security of encryption, increases exponentially.

**2. Digital Signature**

It’s a Mathematical way to validate the authenticity and integrity of a message or digital document or transaction details like email id, credit card info etc. It employs asymmetric cryptography.

It gives the receiver reason to believe the message was sent by the claimed sender.

**3.DSA (Digital Signature Algorithm)**

It also works on two keys- public and private keys. Uses the concept of modular exponentiation and discrete logarithm.

It uses key generating algorithms that selects a private key uniformly at random from a set of possible private keys.

Process:

* Data is processed as a hash function then using a signature algorithm using a private key is made .This process is the Encryption Process.
* Then the received signature and data with Hashing is decrypted.
* Then Hash values at both the ends are matched if they are matched then

The data is authentic and is true.

**IDEA OF COMBINING RSA AND DSA**

**RSA-** Best for Verification and Encryption

**DSA-** For Signing in and Decryption.

To sign a message, just apply the RSA function with the private key to produce a signature to verify, apply RSA Function with the public key to the signature and check that the result is equal which verifies that the message is authenticated and sent by the right user.

This will increase the privacy at both ends when the encryption and decryption is done when the hash functions are generated for verifications.

References:

<https://www.geeksforgeeks.org/rsa-algorithm-cryptography>

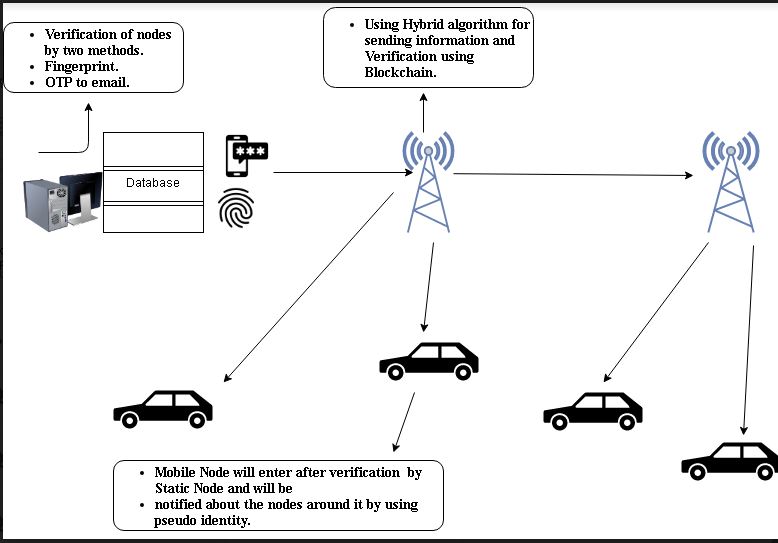
<https://www.geeksforgeeks.org/dsa-algorithm-cryptography>

[www.wikipedia.com](http://www.wikipedia.com)

**IMPLEMENTATION**

The implementation of the Model starts with the verification of the data provided by the user such as the details like phone number, address, car number,personal details of the users is all first verified then stored in the database .All this data is provided or accessible -only to the Trusted Authority . In the whole process only this section is the only one where we are providing the details of the User. This is because Trusted Authority is the one which will start or allow the Mobile Nodes to enter into the VANET. This will validate the information and store all the data into a database which will be sent to the static nodes to verify and continue the whole VANET network. In case if any node proves to have some invalid intention then the Trusted Authority has full rights to use the given information to access them. This will give security and work in a linked manner.

First we start the implementation part by the verification process at various levels then the data verified which will be stored in the Databases then it will generate Public and Private Keys which will be used ahead by the static nodes to allow and verify the nodes with Pseudo Identity Process.



Date: 05/07/2021

**First process of verification is OTP Verification.**

**OTP** is a "One-Time Password" which is randomly generated and sent to your registered mobile number and registered email address for **validation** of your transaction. Here the OTP is sent to the user from where it is seen if the user identity is true and then the further verification process will start.

We can generate OTP using the Random module in python and send it to the user by using SMTPlib module.

We will generate 4-digit OTP.

**Algorithm:**

1. Import modules.
   1. **random**:The **random module** is a built-in **module** to generate the pseudo-**random** variables. It can be used perform some action **randomly** such as to get a **random** number, selecting a **random** element from a list, shuffle elements **randomly**, etc.
   2. **smtplib**:The **smtplib** module defines an **SMTP** client session object that can be used to send mail to any Internet machine with an **SMTP** or ESMTP listener daemon.

<https://www.tutorialsteacher.com/python/random-module#:~:text=The%20random%20module%20is%20a,%2C%20shuffle%20elements%20randomly%2C%20etc>.

<https://link.springer.com/content/pdf/bbm%3A978-3-662-50497-0%2F1.pdf>

1. We will create an empty string and join it with string forming of four random integers using randint() method. And then storing them in the variable **otp**.
2. We will create a server of gmail and set port to 587, using smtplib.SMTP() method.
3. We will start the server using **server.starttls()**

**STARTTLS** is an email protocol command that tells an email server that an email client, including an email client running in a web browser, wants to turn an existing insecure connection into a secure one.

<https://en.wikipedia.org/wiki/Transport_Layer_Security>

1. **Using server.login(email,password)**we will login to the server.

here we will not use the old password but the password generated by the 2-step verification process.

1. We will store the otp in the variable **msg.**
2. Using **server.sendmail(sender mail, receiver mail, msg)**we will send the otp to the receiver.
3. After successfully receiving the otp we will use **server.quit()**  to tear down the connection from the **server**.
4. By this the authentication at first stage would be done and then we can move ahead with our other biometric verification Process.

We have written this code in the Jupyter notebook

**CODE:**

import random

import smtplib

otp=''.join([str(random.randint(0,9)) for i in range(4)])

//we will create a server of gmail and also give the port the passit through tls security layer

server = smtplib.SMTP('smtp.gmail.com',587)

server.starttls()

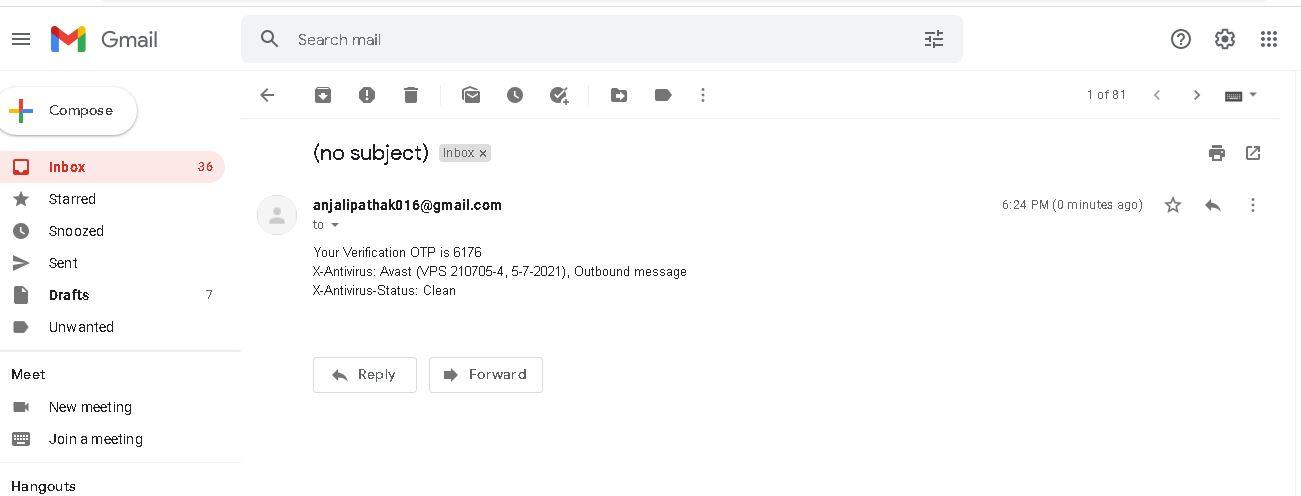
server.login('anjalipathak016@gmail.com','mpocusxhbdpipgkq')

msg= 'Your Verification OTP is '+ str(otp)

server.sendmail('anjalipathak016@gmail.com','devendrapathak015@gmail.com',msg)

server.quit()

//(221, b'2.0.0 closing connection g38sm14638668pgg.63 - gsmtp')

****

**Second method of Verification is Biometric Method**

**Fingerprint Verification**

Detection of fingerprints has become one of the most popular ways of maintaining security systems in today’s world. So, accurate identification of a person through his fingerprints is quite important.

A fingerprint-based biometric system is essentially a pattern recognition system that recognizes a person by determining the authenticity of his/her fingerprint.

***Steps:***

1. **Importing the necessary libraries & modules**

**import** cv2

**import** numpy **as** np

**import** os

1. **Reading the fingerprint to be matched**

test\_original **=** cv2**.**imread**(**"TEST\_1.tif"**)**

1. **Matching with the database**

We are using the SIFT (Scale-Invariant Feature Transform) algorithm here. It is used to extract key-points and detect descriptors for the best-retained features. So, here we will use it to identify the key-points and descriptors of both the test image and database images.

found**=**0

**for** file **in** os**.**listdir**(**"database"**):**

fingerprint\_database\_image **=** cv2**.**imread**(**"./database/"**+**file**)**

sift **=** cv2**.**xfeatures2d**.**SIFT\_create**()**

keypoints\_1**,** descriptors\_1 **=** sift**.**detectAndCompute**(**test\_original**,** **None)**

keypoints\_2**,** descriptors\_2 **=** sift**.**detectAndCompute**(**fingerprint\_database\_image**,** **None)**

After the detection and the computation are over, we start the matching algorithm. We are using the FlannBasedMatcher() functionality here. It helps us in performing a quick but efficient matching using the Cluster and Search algorithm. We will maintain an array of such points that are given as output in the match.

matches **=** cv2**.**FlannBasedMatcher**(dict(**algorithm**=**1**,** trees**=**10**),dict()).**knnMatch**(**descriptors\_1**,** descriptors\_2**,** k**=**2**)**

match\_points **=** **[]**

**for** p**,** q **in** matches**:**

**if** p**.**distance **<** 0.5**\***q**.**distance**:**

match\_points**.**append**(**p**)**

1. **Detecting the ID for fingerprint matched**

keypoints **=** 0

**if** **len(**keypoints\_1**)** **<=** **len(**keypoints\_2**):**

keypoints **=** **len(**keypoints\_1**)**

**else:**

keypoints **=** **len(**keypoints\_2**)**

**if** **(len(**match\_points**)** **/** keypoints**)>**0.5**:**

found**=**1

**print(**"Match found!"**)**

**print(**"Fingerprint ID: " **+** **str(**file**))**

**break**

1. **If fingerprint not found**

**if(**found**==**0**):**

**print(**"Match not found!!!"**)**

**HYBRID ALGORITHM FOR PRIVACY AND VERIFICATION OF DETAILS OF NODES**

1.RSA uses Public and Private Key where Public Key is given to anyone and Private Key is kept only to the authentic sender or Node.

The flow is like

A sends its Public Key to the server and requests for verification so after that it can be entered inside the VANET.

The server encrypts the data using the node Public Key and sends the encrypted data.

The data is received and decrypted.

It uses the key generation method from the multiplication of the two prime numbers. Here the Public key is used for Encryption and Private Key is used for Decryption.

2. Digital Signature is used for the verification of the the message it will add the security to the existing algorithm where an algorithm is used to the generation of the key which will be then check is equal to the message of node and the message that is first encrypted and then decrypted .I that is the case then it is valid and the Node can be proceed further with the process.

**Algorithm**

RSA is used with the additional feature of Digital Signature where the concepts of key generation are taken from RSA . And Signing and Verification is done with the help of Digital Signature Algorithm and concept

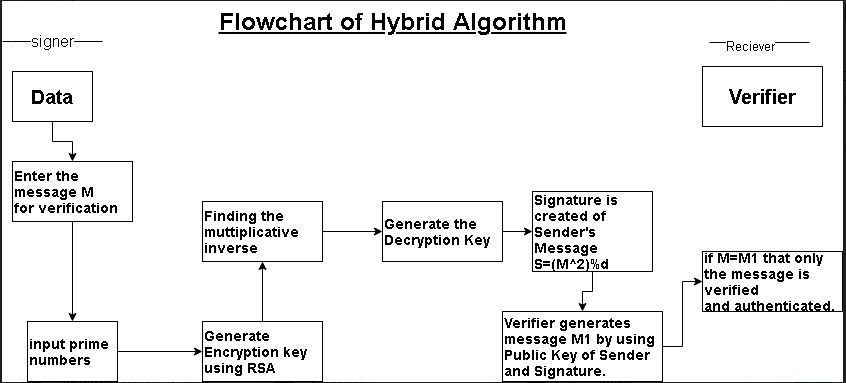
The Digital Signature changes the roles of Private and Public Key.

The Private Key of the sender is used not of the receiver.

The Sender uses his/her own Private Key to sign the document and the receiver uses the Sender Public Key to Verify the authenticity of the Information.

<https://www.geeksforgeeks.org/rsa-algorithm-cryptography/>

<https://www.tutorialspoint.com/cryptography/cryptography_digital_signatures.htm>



**Steps:**

1: Key Generation

This step is same as the key generation of the RSA algorithm

ex: A sends a message to B .A will choose 2 Prime Numbers **p** and **q**.

Calculate **n= p\*q**

**Pn =(p-1)(q-1)**

2 . A chooses private key d such that this conditions are true (d\*e)mod(p-1)(q-1)=1 or d is the valid inverse.

3. public key will be {e,n}

private key = {d,n}

4. Now , let S be the signature A will create the Digital Signature using S = M^d mod n

where M is the message.

5. A sends Message M and signature S to B.

6. B will calculate or decrypt the message M1 = S^e mod n

7. If the message entered M and the message decrypted are the same then the information is authentic and sent by the user A .

8. This node will then be allowed to be in the VANET.

**BENEFITS:**

It will increase the contract speed.

It will enhance Privacy and Security.

It doesn’t involve much complex calculations and is easy to understand.

**Mobile Nodes are connected using Blockchain**

Mobile Nodes are the Vehicles that are used to communicate with each other by sending the message of the Nodes near or around them. This is done by the use of Blockchain where each node is connected with the hash value message of the previous nodes.This will be used by converting the messages of two nodes by concatenating them and converting them into a hash value by using the SHA256 hashing algorithm .

The hash value of the previous node is used in the second node and similarly the message of node three will connect nodes with each other.

There will be a proper authentication as the value differs slightly at any point and will change the hash values of all the nodes ahead. This will also help in the privacy as only the conditions of the previous nodes are shared and not any personal information.

In case of alteration the authority can take the following Vanet network without the interruption of the whole process .As a mobile node in one blockchain is all taken care by one static node and this will work for all the nodes.In this way all the Nodes work together to provide the Smart Traffic Management.

This is how the hash value is generated by passing nodes and hash values int the function H1("aaa",N1,N2,N3)->03gf24

H2(N4,N5,N6,"03gf24")->458kj1.... then we concatenate them all.

**Algorithm**

**Step1**- We will import the library for converting messages into a hash function.

import hashlib

**Step2-** We will create the class to convert ,concatenate and get a hash function.

class VanetNodesHash:

def \_\_init\_\_(self, previous\_block\_hash, transaction\_list):

self.previous\_block\_hash= previous\_block\_hash

self.transaction\_list= transaction\_list

self.block\_data= "-".join(transaction\_list)+"-"+previous\_block\_hash

**Step3-** Hash Algorithm SHA256 is used to convert the message into hash value.

self.block\_hash= hashlib.sha256(self.block\_data.encode()).hexdigest()

**Step4-**These are the nodes with the messages corresponding to them.

N1="Nodes A sends Node B a Message"

N2="Nodes B sends Nodes C a Message"

N3="Nodes C sends Node D a Message"

N4="Nodes D sends Node E a Message"

N5="Nodes E sends Nodes F a Message"

N6="Nodes F sends Node G a Message"

**Step4-** printing the first node with the initial message and hash value

first\_node = VanetNodesHash("initial Message",[N1,N2])

print(first\_node.block\_data)

print(first\_node.block\_hash)

**Output**

Nodes A sends Node B a Message-Nodes B sends Nodes C a Message-initial Message

ab45c850991f3e911ea7447e0b2c862d93d26149f0e3ca33665f7620e120b438

**Step5-** Similarly all the nodes ahead are calculated and received.

second\_node = VanetNodesHash("initial Message ",[N3,N4])

print(second\_node.block\_data)

print(second\_node.block\_hash)

third\_node = VanetNodesHash("initial Message",[N5,N6])

print(third\_node.block\_data)

print(third\_node.block\_hash)

**References**

[**https://www.ibm.com/in-en/topics/what-is-blockchain**](https://www.ibm.com/in-en/topics/what-is-blockchain)

[**https://www.activestate.com/blog/how-to-build-a-blockchain-in-python/**](https://www.activestate.com/blog/how-to-build-a-blockchain-in-python/)