Secure and Efficient Key Delivery in VANET using Cloud and Fog Computing

Sanya Chaba
Computer Engineering
National Institute of Technology, Kurukshetra
Haryana, India
sanyachaba26@gmail.com

Rahul Kumar
Computer Engineering
National Institute of Technology, Kurukshetra
Haryana, India
rahulk0311@gmail.com

Rohan Pant
Computer Engineering
National Institute of Technology, Kurukshetra
Haryana, India
rohanpant26@gmail.com

Mayank Dave
Computer Engineering
National Institute of Technology, Kurukshetra
Haryana, India
mdave@nitkkr.ac.in

***Abstract*--- The article proposes the design of the framework for vehicular ad hoc networks (VANET) for delivery of the authentication keys with minimum delay amongst vehicles having high mobility using fog and cloud computing. We propose to introduce fog computing to extend cloud computing by introducing an intermediate fog layer between mobile devices and cloud thus producing a number of advantages. Since the keys are provided directly from an intermediate layer, the latency is considerably reduced. In addition, the number of messages exchanged between vehicle and other elements of the VANET reduces in comparison to the existing approaches. Consequently, the resultant system is highly efficient. The design is implemented and validated using Omnetpp for single and multi vehicle system.**

Keywords--- Fog Server, KGC, Trusted Management Authority, Vehicular Cloud Computing, Vehicular Adhoc Network, Efficient Key Delivery

I. INTRODUCTION

The number of vehicles on the road has increased drastically because of which, the major problem of the society is the existing transportation system. Therefore, we need a safe and a secure transportation system or an Intelligent Transport System (ITS). ITS can be realized in the form of a network which can provide connectivity to the users on the road and facilitate vehicle-to-vehicle communication. One such network is called Vehicular Ad hoc Network (VANET). In this section we will introduce VANET with a brief overview of its applications ameliorated using cloud computing.

A VANET consists of three important entities; firstly, it consists of a Trusted Authority (TA), which is responsible for the maintenance of the entire system. Secondly, it consists of Road Side Units (RSUs) that are deployed on the side of the roads at intervals of fixed distances. They are responsible for forwarding the messages, authenticating the vehicles thereby reducing the burden on TA. Thirdly, the VANET consists of On Board Unit (OBU), which helps the vehicles to collect and process information and use this information to communicate with each other (Figure 1). In addition, a typical VANET consists of three main types of communications; V2V (Vehicle to Vehicle), V2R (Vehicle to RSU) and R2V (RSU to Vehicle). In V2V and R2V, vehicle is acquainted of its driving environment and takes decisions based on the same to avoid traffic accident, jams etc. Whereas, in V2R, a vehicle itself is a source of information. Traffic Management Authority (TMA) receives information from the RSUs and takes the optimized traffic decisions. However, large amount of information received at TMA may result in congestion and storage burden. In a VANET, the TA in the network is generally a Key Generation Center (KGC), which generates Long Term Pseudonym/Long Term Key (LTP/LTK) and Short Term Pseudonym/Short Term Key (STP/STK) for vehicle and message authentication. LTP/LTK keys are used for vehicle authentication when a vehicle enters a VANET. STP/STK are used for message communication, i.e. when an element of the VANET starts a communication.

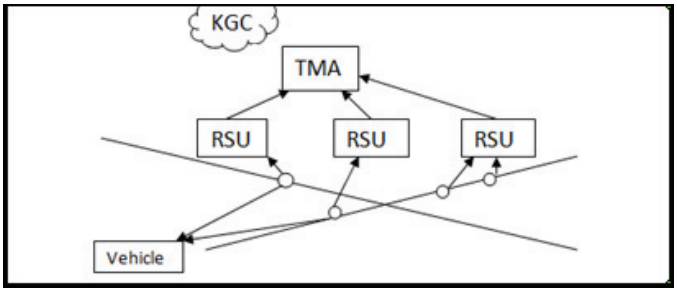


Figure 1: Aggregation of data at TMA

Researchers have envisioned a paradigm shift from the existing VANET and have introduced the concept of Vehicular Cloud Computing (VCC) i.e. merging VANET with Cloud Computing. In VCC, a group of vehicles elects brokers, which in turn elect an Authorization Entity (AE), which in turn seeks permission from the higher authorities to form a cloud. Once the permission is granted, the vehicles pool their resources and act as a cloud service by providing storage and platform for executing applications. In addition, the vehicles on the road might want to use the cloud services. This is facilitated using the RSUs. Lastly, the vehicles might want to use and provide the resources at the same time.

II. MOTIVATION

In a typical VANET, a KGC is deployed on the cloud. A vehicle requests for a STP/STK pair from the KGC via the RSU. This increases the time required for the issuing of the STP/STK pair by the KGC. This delay results an increase in latency in authentication, communication and transfer of

optimized traffic decisions. To mitigate the above issue, in our approach, fog servers are introduced in the network. Fog computing used in our approach extends cloud computing by introducing an intermediate fog layer between mobile devices and cloud server. This accordingly leads to a three-layer Mobile-Fog-Cloud hierarchy. With cloud-like resources, a fog server is able to independently provide pre-defined application services to mobile users in its wireless coverage without the assistance of remote cloud. In addition, the fog servers can be connected to the cloud over internet to leverage the rich computing and content resources of cloud.

III. Related works

There has been a lot of research in the field of Vehicular Ad-hoc network and Vehicular clouds. Following are some salient researches in the same. In [1], VANET Clouds are divided into three main categories: Vehicular Clouds (VC), VANET using Clouds (VuC), and Hybrid Clouds (HC). In VC, vehicles act as a pool of resources. In VuC, the vehicles used the services of clouds e.g. Infotainment. In HC, vehicles rent their services and might use the resources simultaneously. But the following approach had data-storage and privacy issues. In [2], three protocols are proposed for preserving the vehicular communication. The first protocol begins with KGC setting up the system. It generates LTP/LTK pairs for each registering vehicle. Each LTP has a corresponding LTK. The second protocol involved the distribution of STP/STK pairs. These were short lived to facilitate frequent update for regular authentication. In the third protocol, the vehicles got synchronized using common strings while signing. The last protocol exhibited secure as well as privacy-preserving vehicular communications. But the latency in generating STP/STK pairs each time resulted in a delay in optimized traffic decisions. Another approach proposed in [3] introduces the use of MobEyes, which is a mobile sensor middleware that senses event information, processed and classified that data depending upon the interest. That information was then disseminated to the patrolling cars to make optimized traffic decisions. The Navigator Service Agency got the information from vehicular cloud and provided decisions for optimized routes. However, this also lacked secure and privacy aware sharing of the data. [4] introduces the methods to combat the security threats in VANET along with the detailed analysis of the potential threats involved. In order to achieve the same appropriate security architecture is proposed. The approach ensures privacy and is efficient and robust.

IV. System model

A. Vehicle Authentication

When a vehicle enters a VANET [5], it is important that it be authenticated or it should not be malicious. On entering the VANET, the vehicle sends an encrypted message to the RSU requesting an LTP/LTK pair. As VANET is prone to various attacks such as DDOS and Sybil attacks, the RSU checks if the request from the vehicle is valid or not i.e. valid if the vehicle is not attacked. If the request is found to be invalid, the RSU reports the vehicle to the KGC present in the cloud, which in turn identifies the malicious vehicle and keeps record of the same. This ensures security of the system i.e. the system is protected against the malicious vehicles. On the other hand, if the request is found to be valid, it is sent to the KGC where the message (requesting an LTP/LTK pair) gets decrypted and a LTP/LTK pair is generated. The RSU receives the LTP/LTK pair from the KGC and sends it to the vehicle, which requested for it. This process (for better understanding) is clearly illustrated in Figure 2.

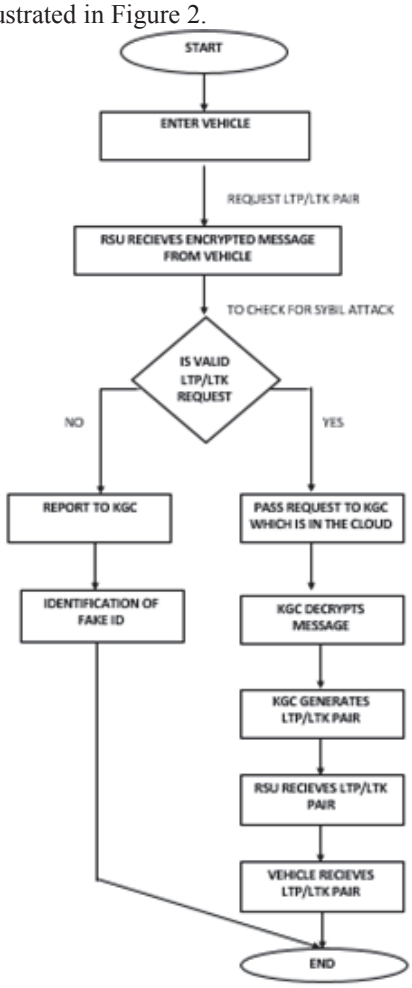
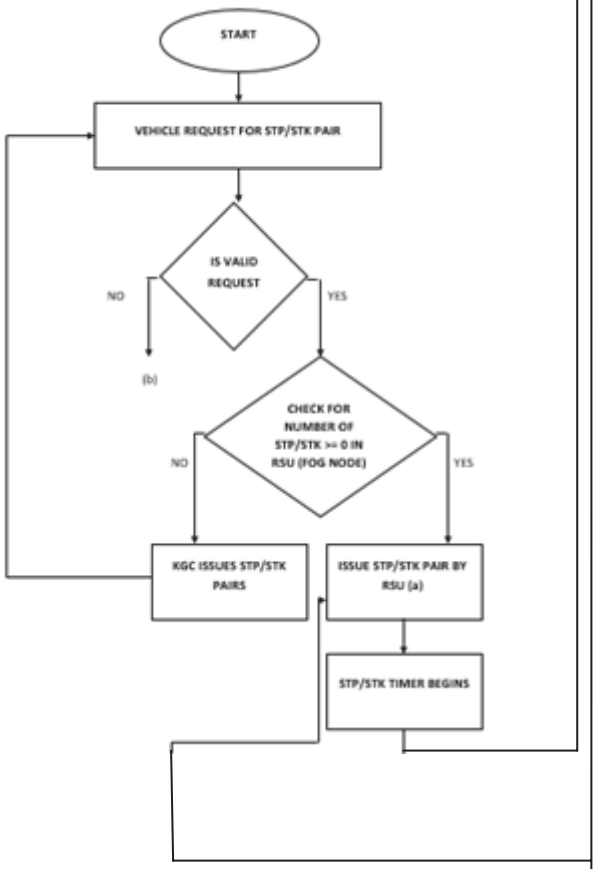


Figure 2: Vehicle Authentication

B. Message Communication

The next step after vehicle authentication is message communication for which the algorithm is stated in the next section. Here each fog[6] server and vehicle is assumed to possess a Fog ID and Vehicle ID for the identification of the server and vehicle respectively. The message communication begins with the request of a STP/STK pair by the vehicle from the RSU (which now is the fog server). If the request is validated(to check from the past records whether the request is from a malicious vehicle or a benign vehicle to ensure security of the system), and availability for STP/STK pairs is checked in the fog server, a pair with a novel ID is allocated to the vehicle requesting for it. If the message is local information for the fog server, it is delivered to the same, otherwise the message is aggregated at fog and re aggregated at TMA,

which then transfers the optimized decision to the vehicle. If the message is a local information for another vehicle, using an appropriate protocol, both the vehicles mutually authenticate and the transfer of message takes place. In addition, if the number of STP/STK pairs were insufficient in the fog server, KGC issues some to the fog server using a transfer of a single message which contains the IDs of the STP/STK pairs issued (here the authenticity of the fog server is assumed) and the entire process is repeated. Each STP/STK pair is assumed to have a timer, which is set at the beginning of the algorithm (Let the optimized timer value by c). This timer ensures the security of the system to a considerable extent i.e. during the transfer of messages, if a vehicle becomes a victim and starts a malicious activity, expiration of its timer, will help check the validity of this malicious vehicle again. At each step, the timer for the pair is checked for expiration. In case the timer value expires, the entire protocol is called again with a new STP/STK pair with a unique ID. If a fog server needs to check whether the information transmitted to the vehicle is an optimized decision coming from the TMA from an older communication or is a message for a new communication with a vehicle, it checks the value of a variable say, 'a' which is assumed to have the value 0 initially. In the former case, the timer value is checked and message is transmitted to the vehicle. In the later, after the availability check as mentioned previously, message in transmitted. We assume a suitable protocol to find the nearest fog server (in terms of range) for a vehicle to communicate. This process is explained in Figure 3



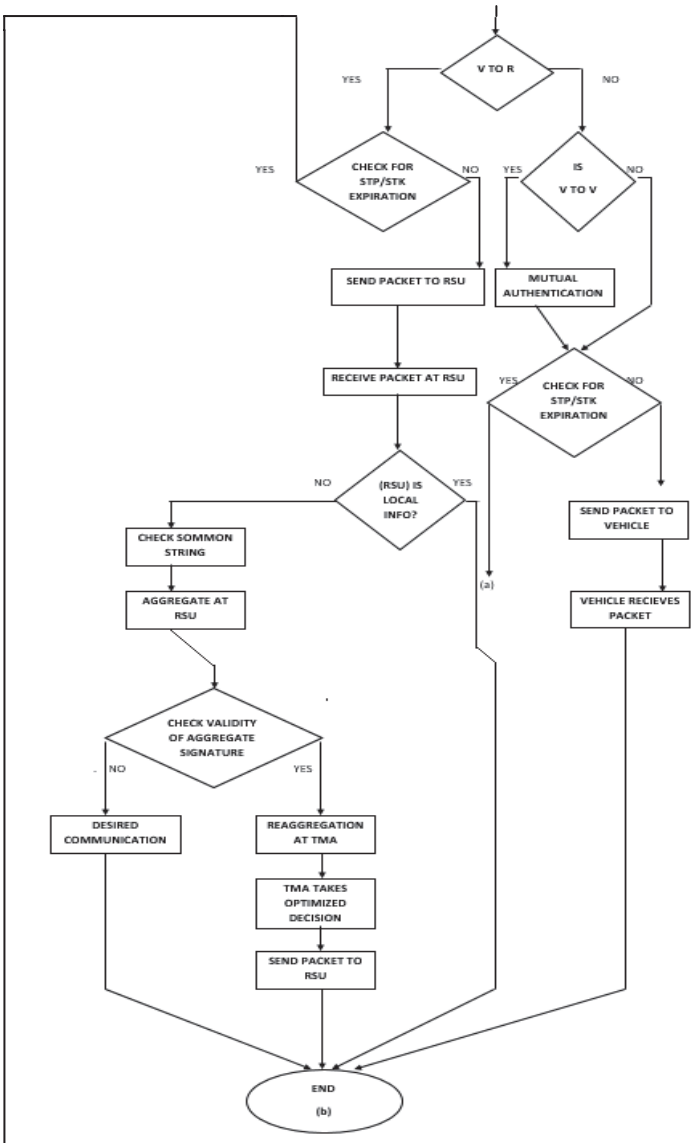


Figure 3: Message Authentication

V. ALGORITHM

As already explained in Section IV, we propose the following algorithm for message communication using fog servers.

Let there be a vehicle v_j and fog server f_i.

```
VehicleFun ( v_j, f_i )
{
If ( IsValid(request)) /*request is validated*/
{
  If(CheckforAvailability(f_i))
  {
    Issue STP/STK_k by f_i to v_j
    STP/STK_k_timer = c /*timer value is set*/
    If(message for f_i)
    {
     If(CheckforTimer(k)) /*where k is the id of STP/STK
*/
     {
```

```
        Send packet to f_i.
        Receive packet at f_i.
        If(!local information) /*directly processed at fog server*/
          {
          /*information to be aggregated*/
           If(common string validated)
           {
            Aggregate at f_i.
            Re aggregate at TMA
            TMA sends optimized decision to fog server
            FogtoVehicle (v_j , 1, f_i. )
           } /* if common string is not validated information is
              discarded  */
          }

         }       Else VehicleFun (v_j)
        }
       else
       {
       Mutual Authentication /*appropriate protocol followed*/
       If(CheckforTimer(k))
      { v_j sends packet to destination vehicle
      Destination vehicle receives packet}else VehicleFun(v_j)
      }
    } else {KGC issues STP/STK pairs to f_i.
                VehicleFun(v_j )}
   }
 }
FogtoVehicle (v_j , a , f_i. )
{
If a=1 /*in continuation of an old communication*/
{
If (CheckforTimer(k))
  {
   Send packet to v_j from f_i.
   Receive packet at v_j
  }
 Else
  {
   If (CheckforAvailability)
   Transmission (v_j , f_i. )
    Else {KGC issues STP/STK pair to f_i.  /*if pairs not available at fog server*/
          FogtoVehicle (v_j, 1 , f_i.) }
  }
}
Else if a=0 /*in continuation of a new communication*/
{
If (CheckforAvailability(f_i.)) transmission(v_j , f_i. )
Else { KGC issues STP/STK pair to f_i.
       FogtoVehicle(v_j , 0 , f_i.) }
}
}

Transmission (v_j , f_i. )
{
Issue STP/STK_k by f_i. to v_j
STP/STK_k_timer = c
Send packet to v_j
Receive packet at v_j
}

IsValid (request)
{
Check authentication for v_j
/*Check for Sybil attack*/
If (authenticated)
   Return true /*authentication checked*/
Else
   Return False /*Request not authenticated*/
}

CheckforAvalaibility (f_i )
{
If (No. of STP/STK > 0)
  Return True
Else
  Return False
}

CheckforTimer( k)
{
 If (STP/STK_k_timer > 0 )
   Return True /*timer expired*/
 Else
   Return False /*timer not expired*/
}
```

VI. IMPLEMENTATION AND RESULTS

The implementation is done on the platform Omnetpp[7] in two steps. First, the STP/STK pairs are issued without the fog node, i.e. via the RSU. The implementation is based on simulations given in [8]. Second, the STP/STK pairs are issued with the fog node, i.e. without the RSU.

A. Key exchange between the KGC, RSU and vehicle(s) without a fog server.

In the simulation involving one vehicle, the vehicle sends a request for a key to the RSU (which takes 0.1 seconds in our simulation), which then forwards the request to the server (which takes 0.1 seconds in our simulation), which is the KGC. On receiving the request, the server sends the required key to the RSU (which takes 0.1 seconds in our simulation) which then forwards the key to the vehicle that requested for it (which takes 0.1 seconds in our simulation). Therefore, this whole process takes 0.4 seconds to complete in our simulation. In the simulation involving multiple vehicles, assuming the multiple vehicles send a request for a key to the RSU simultaneously, which then forwards the request to the server, which is KGC. On receiving the request, the server sends the required key to the RSU, which then forwards the key to the vehicles that requested for it. This whole process takes 0.4 seconds to complete in our simulation. In case of one vehicle

and two vehicle systems, the number of messages exchanged is equal to 4 and 8 respectively. As the number of vehicles increase, the number of messages exchanged increases.

B. Key exchange between the KGC, fog server and vehicle(s).

In the simulation involving single and multi vehicle system, the vehicle(s) sends a request for a key to the fog server (simultaneously in case of multiple vehicles) which takes 0.1 seconds in our simulation, which then instead of forwarding the request to the server, which is the KGC, sends the requested key directly to the vehicle(s) that requested for it (which takes 0.1 seconds in our simulation). This is due to the implementation of a RSU as a fog server. The fog server[9] stores the keys from the KGC beforehand. This whole process takes just 0.2 seconds to complete in our simulation, which is equal to half the time it took without the implementation of RSU as a fog server. In case of one vehicle and two vehicle systems, the number of messages exchanged is equal to 2 and 4 respectively. As the number of vehicles increase, the number of messages exchanged increases but the number is always less than the results without implementation of RSU as fog server.

The above two are clearly explained in the bar graph as shown in Figure 2.

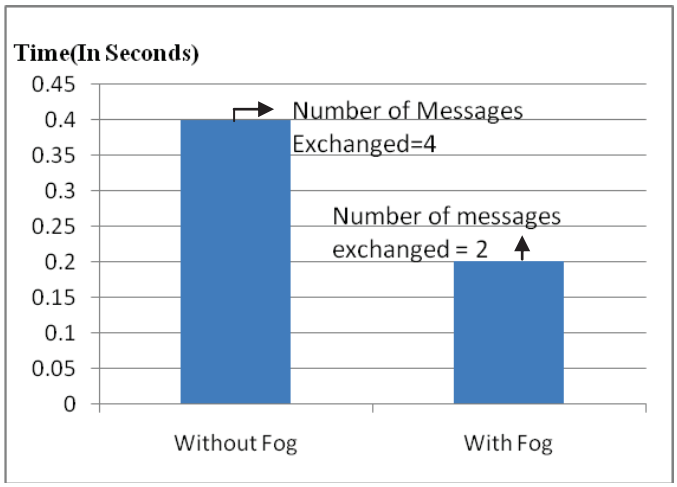


Figure : Implementation Results

VII. CONCLUSION

Our approach ensures lesser delay and hence an efficient approach when compared to the already existing mechanisms [10][11]. This is because; the fog node processes the requests directly, which reduces the latency of the key transfer. This in turn facilitates a faster message communication and hence makes the entire process efficient. In addition, due to the presence of LTP/LTK and STP/STK pairs, the model is secure since each vehicle is authenticated when it joins the network. Take the idea forward; we suggest analyzing the attacks that the network may be prone to in order to make the model securer. In addition, the idea of the paper was to introduce the concept of fog nodes in VANET. To support the same, an implementation was performed. Implementation at a larger scale must be supported along with analyzing the attacks.

REFERENCES

[1] R. Hussain, J. Son, H Eun, S Kim, H. Oh, "Rethinking Vehicular Communications: Merging VANET with Cloud Computing, Cloud Computing Technology and Science (CloudCom)",*IEEE 4th International Conference, 2012*, pp. 607-609.

[2] L. Zhang, J. Domingo-Ferrer,"Privacy-Preserving Vehicular Communication Authentication with Hierarchical Aggregation and Fast Response", *IEEE Transactions On Computers,Vol:65, 2016*,pp. 2563-2565.

[3] M. Gerla, "Vehicular Cloud Computing*", Ad Hoc Networking Workshop (Med-Hoc-Net), 2012 The 11th Annual Mediterranean on 2012*, pp. 152-155.

[4] Raya,Maxim,Hubaux,Jean—Pierre,"Special Issue on Security of Ad-hoc and Sensor Networks", *Journal of Computer Security,Vol.:15,*2007,pp39-68.

[5] R. Shringar Raw, M. Kumar, N. Singh, "Security Challenges, Issues and their Solutions for VANET", *International Journal of Network Security & Its Applications (IJNSA), 2013*, pp. 95-105.

[6] T. H. Luan, L. Gao, Z. Liz, Y. Xiang, G. Wey and L. Sun, "Fog Computing: Focusing on Mobile Users at the Edge", *arXiv preprint arXiv:1502.01815, Vol:5,2015*, pp. 1-10.

[7] A.Varga," Using the OMNeT++ discrete event simulation system in education",*IEEE Transactions on Education,*Vol:42,1999,pp 11

[8] H. Hartenstein and P. Laberteaux, "VANET Vehicular Applications and Inter-Networking Technologies", Wiley, 2010.

[9] C. Esposito, A. Castiglione B. Martini, K.K. Raymond, "Cloud Manufacturing: Security, Privacy, and Forensic Concerns", *IEEE CLOUD COMPUTING, Vol:3,2016*, pp. 16-22.

[10] J. Blum, A. Eskandarian, and L. Hoffman, "Performance Characteristics of Inter-Vehicle Ad Hoc Networks", *Intelligent Transportation Systems,2003*, pp. 114-118.

[11] Hannes Hartenstein et al., "A tutorial survey on vehicular Ad Hoc Networks", *IEEE Communication Magazine June Vol:46,2008*, pp. 164-171.