Intrusion Prevention System for DDoS attack on VANET with reCAPTCHA Controller using Information based metrics

Poongodi M1, Vijayakumar V2, Fadi Al-Turjman3, Mounir Hamdi4, IEEE Fellow, Ma Maode5, Senior Member, IEEE

1 Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Qatar
2 School of Computing Science and Engineering, Vellore Institute of Technology, Chennai, India
3 Artificial intelligence department, Near East University, Nicosia, Mersin 10, Turkey
4 School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore

Corresponding author: Poongodi M (dr.m.poongodi@gmail.com).

ABSTRACT Due to the dynamic in nature, the vulnerabilities that exist in VANET are much higher when compared with that of the wired network infrastructure. In DoS attacks, the legitimate users are prohibited from accessing the services or network resource. The primary goal of the attack to make the desired destination vehicle unavailable or relegate the message all the way through the network affects the reachability. The proposed reCAPTCHA controller mechanism prevents the automated attacks similarly like botnet zombies. The reCAPTCHA controller is used to check and prohibit most of the automated DDoS attacks. For implementing this technique, the information theory based metric is used to analyze the deviation in users request in terms of entropy. Frequency and entropy are the metrics used to measure the vulnerability of the attack. The stochastic model based reCAPTCHA controller is used as a prevention mechanism for the large botnet based attackers. To inspect the efficiency of the proposed method, various network parameters are considered such as Packet Delivery Ratio (PDR), Average Latency (AL), Detection Rate (DR) and Energy Consumption (EC). In the proposed research work, the metric PDR is used to know successful delivery of data packets to the destination vehicle without any interruption. These parameters are used to measure how effectively the data is delivered to the destination from source vehicle.

INDEX TERMS reCAPTCHA Controller, Frequency distribution, Co-variance analyzer, DDoS attack.

I. INTRODUCTION

In recent years Vehicular ad-hoc network (VANET) is considered to be significant and evolving area in networking. VANET is a subclass of Mobile ad-hoc networks (MANET). Vehicular ad-hoc network has three types of communications within the network, a) Vehicle to road-side unit and in sequence b) Road-side unit to road-side unit (R2R) and c) Vehicle to Vehicle communication. Now-a-days there are considerable amount of road accident cases are found. Because of the reason, that roads are overcrowded and hectic.

Here in VANET infrastructure, the dedicated short range communication (DSRC) is to establish the communication between the dynamic nodes which are often fluctuate their directions. Vehicle to vehicle communication among the network is used to communicate the information regarding alert messages and congestion or traffic in the road with the help of RSU. The RSU is placed in the road to help passing information the dynamic vehicles. As discussed earlier, the VANET is subset of MANET, So, VANET environment, with network are of dynamic nodes which moves around the network area with single and multi-hop fashion, to communicate with each other, it also utilizes the RSU further for effective communication for in and around mobile nodes of the network [10]. The advantages in VANET is to enrich the protection of vehicle by alert messages passing between vehicles to ensure the safety. And also there is lot of security vulnerabilities and loop holes to the network which are message modification, sending bogus information, Denial of service attack etc. The crucial need in the VANET is to ensure the safety of passengers by ensuring the information passed among the vehicles is authentic and confidential [11]. In VANET architecture, vehicle communicate each other with the range of communication, if vehicle is away from the range of communication will be passed information in multi-hop.
fashion to reach the desired destination node. Road-side unit (RSU) communication enhances the range of network communication in which the enables vehicles to send, receive and forward data with other vehicles [12]. Identifying the malicious attacks, misbehaving of nodes and faulty vehicles is difficult due to the feature of VANET infrastructure such as decentralization and dynamic topology of network nodes.

To solve the problems mentioned above,

- The reCAPTCHA controller technique prevents the automated attacks similarly like botnet zombies. Each rule in the rule metrics represents the traffic instance for filtering the specific IP addresses or ports.
- The frequency and entropy value is computed for each rule on all the incoming packets at each detection window. The high possible attack is recognized by the deviation occurred between the recent traffic profile and the stored ones.
- The belief scores are calculated from the deviation occurred between the present and prior score in each detection and also updated at the end.
- The frequency distribution matrix is constructed from the rule metrics and entropy deviation is occurred from the covariance matrix for enhancing the sensitivity of detection rate.
- The reCAPTCHA controller is used to check and prohibit most of the automated DDoS attacks.

For implementing this technique, the information theory based metric is used to analyze the deviation in users request in terms of entropy. Frequency and entropy are the metrics used to measure the vulnerability of the attack. The stochastic model based reCAPTCHA controller is used as a prevention mechanism for the large botnet based attackers.

To inspect the efficiency of the proposed method, various network parameters are considered such as Packet Delivery Ratio (PDR), Average Latency (AL), Detection Rate (DR) and Energy Consumption (EC). In the proposed research work, the metric PDR is used to know successful delivery of data packets to the destination vehicle without any interruption. These parameters are used to measure how effectively the data is delivered to the destination from source vehicle.

This paper is systematized into six sections: the literature reviews are presented in Section II. Section III describes Problem definition and discusses solution. Section 4 describes the proposed approach. Results and analysis are provided in Section V. Section VI presents the conclusion.

II. LITERATURE REVIEW

[1] Authentication plays crucial role in safe communication. To secure the messages, the authentication is necessary. In this paper the ECDSA standard is used as digital signature, in spite of authentication, the performance measures need to be considered as well, here in this research the digital signature verification is comparatively high. As a consequence, the attackers of inside and outside the network can utilize this time to use the portion of bandwidth to overflow the network with signatures which are not validated, which results in DoS attack. Here the future work represents that, system tries to mitigate overwhelming attacks with false signature. But, instead if the intruder tries to overflow with false information and bogus signature, the proposed system here in this paper is not so effective. So, future work tries to mitigate this security loophole of this system.

The VANETs are prominently important technology for secured and intelligent transportation system. But the security aspects of VANET get less attention and work towards that so far is not satisfactory. In this article [2], author discussed on technical issues and security challenges in VANET. And, also discussed the major attacks and solutions which mitigates the attack which have been implemented in VANET. The author analysed the results with different parameters. At last they have represented about the open issues and challenges in VANET.

VANETs established as consistent networks through which the vehicles can communicate with each other. The main purpose of VANET is communication on highways and urban environments. Even though, it has many advantages, it also undergoes some challenges and security issues such as high bandwidth and connectivity, vehicle privacy and provisioning QoS. This research paper [3] presents infrastructure of VANET and related challenges.

VANET is being a highly trending area of research due to its importance in societal needs over last few years. Since it is ad-hoc in nature, concept of establishing the network for vehicles and also the communication between them in highways or urban environment is done easy. Along with many advantages, there arise a more performance challenges in vehicular ad hoc networks such as provisioning QoS, security and availability, vehicles individual privacy, and bandwidth. This research paper also represents related challenges [4].

Denial of service attack is the foremost issue which creates risk to the economy. The author proposed the MIPDA Algorithm, which is implemented for detecting the DoS attacks which affects the availability of the network. As a result, attack is ultimately confined with the source domains that it avoids the traffics created by zombies and overwhelming packets that it leads to break the privacy and provides the unavailability of the network. Together the performance of the network also increases by reduces the delay of overhead for processing of information. Which in turn again it increases the communication time, and detect the multiple malicious invalid request, and also detect the attacker in efficient way for an environment [5]

[6] Enhanced packet detection algorithm is proposed which avoids the deterioration of performance of network. The network availability is affected due to DoS attacks, and also in information sharing and communication, the availability of the network is major need. The secure communication can be major security threat now-a-days. The author implement an algorithm for DoS detection algorithm to improve privacy of network and availability.
Vehicular ad hoc networks are targets for various attackers. The game theory for the security in wireless network has high impact on detection of attacks and it considered being good security model through high achieving results. Here, in this research article, author proposes the metrics on game theory models. A novel methodology of game has been proposed to improve the efficiency of communication and availability of network to the vehicles. The proposed method here has been evaluated by simulation results, thus it will be very helpful for solving future DoS attacks in VANET.

The detection methodology is proposed for DoS attack in IEEE 802.11 p. The research focuses on the periodic position of jamming attacks and false alarm rate which has been estimated with different attacker models. Future ongoing work planned is that relaxing the assumption of proposed model and correspondingly enhancing the detection for realistic scenarios.

Author proposed the ECDSA algorithm which describes about the Elliptic curve digital signature to protect the network.

During the survey, some points are explored, where it proposes the drawbacks of existing technique. The proposed research work in this paper, effectively implements defense mechanisms in VANET and it able to enhance the security by distinguishing legit and malicious requests accurately. The same has been claimed using the simulation results in Section 5.

### III. PROBLEM DEFINITION

The automated attack is now growing and very popular among the intruders to perform the Denial attacks in the network without any effort. Due to the tools availability, any normal person can produce this attack in the network without any expertise. The robust mechanism is required to overcome the DDoS attack to the greater extent. Since VANET is highly dynamic and decentralized network with sensitive field of information and also the application of VANET is very extensive in the field of Intelligent Transportation system. It is substantial to protect against the intrusion in the VANET infrastructure.

In most recent distributed denial of service attacks, utmost sophisticated methods are implemented by the intruders. To overcome similar multiple attack events, collaborative approaches need to be carried out. DDoS attacks are difficult to be traced back easily because of its distributed nature. This is even more difficult to locate and to detect the attacks.

Hence, an innovative detection mechanism is needed to prevent the security failure of the network in VANET.

### TABLE I

<table>
<thead>
<tr>
<th>Acronyms</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>MANET</td>
<td>Mobile Ad-hoc Network</td>
</tr>
<tr>
<td>DSRC</td>
<td>Dedicated short range communication</td>
</tr>
<tr>
<td>RSU</td>
<td>Road side unit</td>
</tr>
<tr>
<td>R2R</td>
<td>Road to road unit</td>
</tr>
<tr>
<td>PDR</td>
<td>Packet Delivery Ratio</td>
</tr>
<tr>
<td>AL</td>
<td>Average Latency</td>
</tr>
<tr>
<td>DR</td>
<td>Detection Rate</td>
</tr>
<tr>
<td>EC</td>
<td>Energy Consumption</td>
</tr>
<tr>
<td>ECDSA</td>
<td>Elliptic curve digital signature algorithm</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of service</td>
</tr>
<tr>
<td>DoS</td>
<td>Denial of service</td>
</tr>
<tr>
<td>DDoS</td>
<td>Distributed denial of service</td>
</tr>
<tr>
<td>CBR</td>
<td>Constant Bit Rate</td>
</tr>
<tr>
<td>AODV</td>
<td>Ad-hoc on demand distance vector</td>
</tr>
<tr>
<td>UDP</td>
<td>User data gram protocol</td>
</tr>
</tbody>
</table>

### TABLE II

<table>
<thead>
<tr>
<th>Acronyms</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Number of requests</td>
</tr>
<tr>
<td>dw</td>
<td>Detection window</td>
</tr>
<tr>
<td>Entropy threshold</td>
<td>Entropy threshold</td>
</tr>
<tr>
<td>C_{N×dw}</td>
<td>Covariance matrix</td>
</tr>
<tr>
<td>F_{i,j}</td>
<td>Frequency of rules at detection window</td>
</tr>
<tr>
<td>S_{ip}</td>
<td>Source IP</td>
</tr>
<tr>
<td>D_{ip}</td>
<td>Destination IP</td>
</tr>
<tr>
<td>S_{port}</td>
<td>Source port</td>
</tr>
<tr>
<td>D_{port}</td>
<td>Destination port</td>
</tr>
<tr>
<td>R</td>
<td>Set of rules</td>
</tr>
<tr>
<td>F_{N×dw}</td>
<td>Frequency distribution matrix</td>
</tr>
<tr>
<td>μ</td>
<td>Expected frequency</td>
</tr>
<tr>
<td>σ^2</td>
<td>Variance</td>
</tr>
</tbody>
</table>
from covariance matrix \( (C) \) of data. The entropy of our observations can be calculated from the uncertainty and randomness associated with a variable according to information theory, the metric entropy is a measure of the amount of variation in a distribution. According to information theory, the metric entropy is a measure of uncertainty and randomness associated with a variable (random). It measures the information contained in a slice of data. The entropy of our observations can be calculated from covariance matrix \( (C_{N\times dw}) \). The entropy threshold \( (E_{\text{threshold}}) \) is determined by the entropy distribution. Here we can also consider entries at the time when the DDoS attack occurred in the past. Now this system can be used to filter the attack traffic by using the source side integrity checks.

To implement the proposed reCAPTCHA controller technique, the configuration of rules is the foremost phase to constrain unauthorized access in the network. User behavior is analyzed to detect the malicious nodes. User behavior is represented by the number of requests to access the resources. The frequency matrix of order \( N \times dw \) is computed. Here, \( N \) denotes number of requests and the \( dw \) notation denotes the detection window. Each entry in the matrix is denoted by \( F_{ij} \) to detect the frequency of rule matching incoming packets at each detection window. Next compute a covariance matrix which is necessary to fully describe the variation in a distribution. According to information theory, the metric entropy is a measure of uncertainty and randomness associated with a variable (random). It measures the information contained in a slice of data. The entropy of our observations can be calculated from covariance matrix \( (C_{N\times dw}) \). The entropy threshold \( (E_{\text{threshold}}) \) is determined by the entropy distribution. Here we can also consider entries at the time when the DDoS attack occurred in the past. Now this system can be used to filter the attack traffic by using the source side integrity checks.

### IV. PROPOSED WORK

#### A. reCAPTCHA Controller

To address these issues, the proposed system reCAPTCHA controller is used for filtering the attack traffic by using the source side integrity checks. It is challenge response mechanism which can block the bot generated traffic and non-prominent to the human-generated traffic. Some attacks involve malicious nodes which indulge in least number of interactions to prevent idle times from expiring by lingering on with the partial responses and requests. These attacks consume vehicle resources and slow down the applications by leading eventually to an incapability of handling the messages. The ‘low and slow’ attacks, as comparatively small number of malicious clients can take part in a DDoS attack stealthily and slowly without any consumption of significant bandwidth on the network. In current days such attacks are common. Our proposed system monitors the aggregate and incremental communication thresholds to aggressively close malicious connections to prevent the network from the resource consumption.

To implement the proposed reCAPTCHA controller technique, the configuration of rules is the foremost phase to constrain unauthorized access in the network. User behavior is analyzed to detect the malicious nodes. User behavior is represented by the number of requests to access the resources.

### TABLE III

<table>
<thead>
<tr>
<th>Rule ID</th>
<th>Source IP</th>
<th>Source Port</th>
<th>Destination IP</th>
<th>Destination Port</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><em>.</em>.<em>.</em></td>
<td>Any</td>
<td><em>.</em>.<em>.</em></td>
<td>Any</td>
<td>Accept</td>
</tr>
<tr>
<td>2</td>
<td><em>.</em>.<em>.</em></td>
<td>Any</td>
<td>198.162.5.*</td>
<td>110</td>
<td>Accept</td>
</tr>
<tr>
<td>3</td>
<td><em>.</em>.<em>.</em></td>
<td>Any</td>
<td>198.165.5.3</td>
<td>Any</td>
<td>Accept</td>
</tr>
<tr>
<td>4</td>
<td><em>.</em>.<em>.</em></td>
<td>Any</td>
<td>198.165.5.3</td>
<td>110</td>
<td>Accept</td>
</tr>
<tr>
<td>5</td>
<td><em>.</em>.<em>.</em></td>
<td>21</td>
<td>198.162.5.*</td>
<td>Any</td>
<td>Accept</td>
</tr>
<tr>
<td>6</td>
<td><em>.</em>.<em>.</em></td>
<td>21</td>
<td>198.162.5.3</td>
<td>110</td>
<td>Accept</td>
</tr>
<tr>
<td>7</td>
<td><em>.</em>.<em>.</em></td>
<td>21</td>
<td><em>.</em>.<em>.</em></td>
<td>Any</td>
<td>Accept</td>
</tr>
<tr>
<td>8</td>
<td>192.168.3.*</td>
<td>21</td>
<td>192.168.5.*</td>
<td>110</td>
<td>Accept</td>
</tr>
<tr>
<td>9</td>
<td>192.168.3.*</td>
<td>21</td>
<td>198.165.5.3</td>
<td>Any</td>
<td>Accept</td>
</tr>
<tr>
<td>10</td>
<td>192.168.3.*</td>
<td>Any</td>
<td>198.168.5.3</td>
<td>110</td>
<td>Accept</td>
</tr>
<tr>
<td>11</td>
<td>192.168.3.*</td>
<td>Any</td>
<td>198.162.5.*</td>
<td>Any</td>
<td>Accept</td>
</tr>
<tr>
<td>12</td>
<td>192.168.3.*</td>
<td>Any</td>
<td>198.168.5.3</td>
<td>110</td>
<td>Accept</td>
</tr>
<tr>
<td>13</td>
<td>192.168.3.*</td>
<td>Any</td>
<td><em>.</em>.<em>.</em></td>
<td>Any</td>
<td>Accept</td>
</tr>
<tr>
<td>14</td>
<td>192.168.3.*</td>
<td>Any</td>
<td>198.168.5.3</td>
<td>110</td>
<td>Accept</td>
</tr>
<tr>
<td>15</td>
<td>192.168.3.*</td>
<td>21</td>
<td>198.165.5.3</td>
<td>Any</td>
<td>Accept</td>
</tr>
<tr>
<td>16</td>
<td>192.168.3.*</td>
<td>Any</td>
<td><em>.</em>.<em>.</em></td>
<td>110</td>
<td>Accept</td>
</tr>
<tr>
<td>17</td>
<td>192.168.3.15</td>
<td>21</td>
<td><em>.</em>.<em>.</em></td>
<td>Any</td>
<td>Accept</td>
</tr>
<tr>
<td>18</td>
<td>192.168.3.15</td>
<td>21</td>
<td><em>.</em>.<em>.</em></td>
<td>110</td>
<td>Accept</td>
</tr>
<tr>
<td>19</td>
<td>192.168.3.15</td>
<td>21</td>
<td>198.168.5.3</td>
<td>Any</td>
<td>Accept</td>
</tr>
<tr>
<td>20</td>
<td>192.168.3.15</td>
<td>21</td>
<td>198.168.5.3</td>
<td>110</td>
<td>Accept</td>
</tr>
<tr>
<td>21</td>
<td>192.168.3.15</td>
<td>21</td>
<td><em>.</em>.<em>.</em></td>
<td>Any</td>
<td>Accept</td>
</tr>
<tr>
<td>22</td>
<td>192.168.3.15</td>
<td>Any</td>
<td><em>.</em>.<em>.</em></td>
<td>110</td>
<td>Accept</td>
</tr>
<tr>
<td>23</td>
<td>192.168.3.15</td>
<td>Any</td>
<td>192.168.5.*</td>
<td>Any</td>
<td>Accept</td>
</tr>
<tr>
<td>24</td>
<td>192.168.3.15</td>
<td>21</td>
<td>192.168.5.*</td>
<td>110</td>
<td>Accept</td>
</tr>
<tr>
<td>25</td>
<td>192.168.3.15</td>
<td>Any</td>
<td><em>.</em>.<em>.</em></td>
<td>Any</td>
<td>Accept</td>
</tr>
</tbody>
</table>

The rule shows the packets produced from the tuple as:

\[
< 192.168.3.*, \text{Any}, *.*.*.* \geq 110 >
\]

with the parameters. Source and Destination IP, Ports of Source and destination represented \( S_{ip} \), \( D_{ip} \), \( S_{port} \) and \( D_{port} \) accordingly. The nodes
IP address for the destination and source packets are taken into account in these cases, in which the host as (192.168.3.15), subnet (192.168.3.*), and “*.*.*.*”. Considering, subnet IP address is given, it evaluates the subnet range for center, top and values.

The algorithm makes use of 6 IP addresses in case of “*.*.*.*” is given, addresses to replace it by the approach of all IP addresses are identical but every single selected IP targets to uncover the most predicted IP address range. Those IP addresses are:

- 127.0.0.1
- 88.241.34.41
- 10.0.01
- 215.15.168.23
- 192.168.0.1
- 172.16.0.1

The port number is represented as an integer value or “any”. If represented using an integer value, it is denoted as it is given. But in the case of “any”, the port addresses can be as given here “0, 65535, 23, 80”.

With the set of rules mentioned in the Table III, Each rule in the rule metrics represents the traffic instance for filtering the specific IP addresses or ports. The frequency and entropy value is computed for each rule on all the incoming packets at each detection window. Thus it assumes that it has incremented the counter on the fly for the rules, it is shown in algorithm for analyzing flooding DDoS attack in the next subsection.

C. FREQUENCY DISTRIBUTION MATRIX

In figure 1, the flowchart of algorithm is given clearly represents that the frequency distribution matrix is defined to record the behavior of the users in terms of number of request sent. The request interval values are witnessed along with the request count. The system keeps track of a threshold request interval for the user’s requests. The request count level is verified and flow analysis is initiated with reference to the request levels. The request flow is measured with the frequency distribution at each detection window. The suspected data is verified with other flow sequences. The attack requests are identified by the variations detected by the flow from the covariance matrix computed. The entropy computed from the covariance matrix is compared with the threshold value to detect DDoS attack as shown in below Figure.

![Flowchart of reCAPTCHA Controller](image)

**Algorithm for analyzing flooding DDoS attack**

**Input**: Incoming Traffic  
**Output**: Raise Alert

**Function**: Analyze_FloodAttack()

**Step 1**: Begin

**Step 2**: If (IP_ID = Rule_ID) {
  rate = rate + F  // rate is incoming packet rate
  // Malicious incoming packets are checked via request frequency(F_i)
}

**Step 3**: while (rate_i > Cap_j) then  // Cap_j is capacity of the target node
  BV = False  // Boolean value is set to false
  RaiseAlert();  // DDoS alert

**Step 4**: End While

**Step 5**: else {Forward request to the system}

**Step 6**: End if

**Step 7**: End

Flooding the massive number of packets within a fraction of seconds from malicious nodes in a distributed manner can be prevented by the above algorithm using the rate limiter of incoming packets by using the node capacity.
value. The frequency of incoming packets for each node is monitored and compared with node’s capacity acts a rate limiter for incoming packets.

The slow amplification of attacks is even more complicated to spot the malicious nodes and guard the network. The new innovative mechanism is introduced in our proposed system below

The rule specified in the Equation (1). The set of rules $R$ is specified in the Table III.

$$R=\{r_i \mid i \in (0,N)\}$$  \hspace{1cm} (1)

The each detection window computes the distribution frequency ($F_{ij}$) of incoming packets is given in Equation (2)

$$F_{ij} = \frac{r_{ij}}{\sum_{i,j} r_{ij}}$$ \hspace{1cm} (2)

The frequency distribution matrix ($F_{N\times dw}$) is constructed as follows in Equation (3)

$$F_{N\times dw} = \begin{bmatrix} F_{11} & F_{12} & \cdots & F_{1dw} \\ F_{21} & F_{22} & \cdots & F_{2dw} \\ \vdots & \vdots & \ddots & \vdots \\ F_{N1} & F_{N2} & \cdots & F_{Ndw} \end{bmatrix}$$ \hspace{1cm} (3)

Here $F_1, F_2, \ldots, F_{dw}$ are defined as shown in the Equation (4)

$$F_i = \begin{bmatrix} F_{i1} \\ F_{i2} \\ \vdots \\ F_{iN} \end{bmatrix}$$ \hspace{1cm} (4)

The statistical analysis is used to determine the trust value in terms of entropy deviation using the information based metrics. Trust can be represented as a subjective concept for the trustor regarding the future anticipation, but, it could be influenced by the previous events with the trustee, i.e., experience.

Algorithm for Frequency Distribution

Input:  \hspace{0.5cm} Incoming Traffic
Output:  \hspace{0.5cm} Frequency Distribution

Function $Freq\_distribution$ ( )

Step 1:  \hspace{0.5cm} Begin

Step 2:  \hspace{0.5cm} If (IP_ID = Rule_ID) { Select the rule from the rule sets $R=\{r_i \mid i \in (0,N)\}$

Step 3:  \hspace{0.5cm} Each incoming packets match the at least one rule sets $r_i$

Step 4:  \hspace{0.5cm} The packets matching the rule $r_i$ in the detection window $dw$ is computed in terms of frequency distribution

Step 5:  \hspace{0.5cm} The distribution of each rule on all incoming packets is given by $f_{ij}$

Step 6:  \hspace{0.5cm} The fraction of packets matching the rule in each $dw$ is computed by $F_{ij}$

Step 7:  \hspace{0.5cm} End

The above algorithm is used to define the frequency distribution for each packets on rule $r_i$ at detection windows $dw$. The proportion of packet matching on $dw$ is given as $F_{ij}$. The output of this algorithm is given to the algorithm for distribution matrix to obtain the Frequency distribution matrix.

Algorithm for Distribution Matrix

Input:  \hspace{0.5cm} Frequency distribution
Output:  \hspace{0.5cm} Distribution Marix

Function $Distribution\_Matrix$ ( )

Step 1:  \hspace{0.5cm} Begin

Step 2:  \hspace{0.5cm} Frequency distribution for each rule $r_i$ is obtained $F_i$

Step 3:  \hspace{0.5cm} The frequency distribution matrix of order $F_{N\times dw}$ is constructed with $f_{ij}$

Step 4:  \hspace{0.5cm} The matrix is given as $F_{N\times dw} = \begin{bmatrix} F_1 & F_2 & \cdots & F_{dw} \\ F_1 & F_2 & \cdots & F_{dw} \\ \vdots & \vdots & \ddots & \vdots \\ F_1 & F_2 & \cdots & F_{dw} \end{bmatrix}$

Step 5:  \hspace{0.5cm} The matrix $F_{N\times dw}$ in terms of frequency distribution $f_{ij}$ is given as

$$F_{N\times dw} = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1dw} \\ f_{21} & f_{22} & \cdots & f_{2dw} \\ \vdots & \vdots & \ddots & \vdots \\ f_{N1} & f_{N2} & \cdots & f_{Ndw} \end{bmatrix}$$
Step 6: Each frequency distribution matrix is compartmented with respect to detection window $dw$ is given as

$$
F_1 = \begin{bmatrix}
\begin{array}{c}
f_{11} \\
f_{21} \\
\vdots \\
f_{N1}
\end{array}
\end{bmatrix},
F_2 = \begin{bmatrix}
\begin{array}{c}
f_{12} \\
f_{22} \\
\vdots \\
f_{N2}
\end{array}
\end{bmatrix},
\ldots,
F_{dw} = \begin{bmatrix}
\begin{array}{c}
f_{1dw} \\
f_{2dw} \\
\vdots \\
f_{Ndw}
\end{array}
\end{bmatrix},
$$

Step 7: End

The frequency distribution is obtained in the form of matrix in the order of $N \times dw$ with respect to $r_i$ and $dw$ and it is compartmented with respect to detection window $dw$ as shown in algorithm. The output of this algorithm is given to the algorithm for covariance analyzer to obtain the covariance matrix and entropy.

**D. COVARIANCE ANALYZER**

The covariance matrix analyzes the deviation in the frequency distribution of rules at each detection window. The sensitivity of detection accuracy is increased by computing the changes accurately.

The expected frequency $\mu_i$ for each detection window has been calculated for constructing the covariance matrix as given in Equation (5)

$$
\mu_i = E[f_i] = \frac{1}{N} \sum_{j=1}^{N} f_{ij}
$$

The covariance matrix $C_{N \times dw}$ is constructed.

The matrix elements represent each value as, variance represented by $\sigma^2$ is given in Equation (6)

$$
\sigma^2 = Var(F) = E[(F - \mu)^2]
$$

The covariance for two dimensions can be written as given in Equation (7)

$$
C = E[(F - \mu)(F - \mu)dw]
$$

The covariance matrix $(C_{ij})$ is constructed as follows as given in Equation (8)

$$
C_{ij} = \begin{bmatrix}
\begin{array}{c}
E[f_i - \mu_i](f_j - \mu_j)dw \\
E[f_i - \mu_i](f_j - \mu_j)dw \\
\vdots \\
E[f_i - \mu_i](f_j - \mu_j)dw
\end{array}
\end{bmatrix}
$$

The entropy $(H)$ is calculated from the covariance matrix as given in Equation (9)

$$
Entropy = H = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{dw} C_{ij} \ln C_{ij}
$$

**Algorithm for Covariance Analyzer**

**Input:** Frequency distribution matrix  
**Output:** Entropy

**Function**  
Covariance_Analyzer()  

**Step 1:** Begin  
**Step 2:** The expected value of frequency for each rule $r_i$ is computed based on the $F_{ij}$  
**Step 3:** The $\mu_i$ represents the expected value of frequency  
$$
\mu_i = E[f_i] = \frac{1}{N} \sum_{j=1}^{N} f_{ij}
$$

**Step 4:** The deviation occurred for rule matching frequencies can be identified from the value $\mu_i$  
**Step 5:** The variance is computed from the $\mu_i$ and $F$ values computed for each $dw$  
**Step 6:** The variance is represented as $\sigma^2$ as given

$$
\sigma^2 = Var(F) = E[(F - \mu)^2]
$$

**Step 7:** The covariance matrix is constructed from the variance given for two dimensions is given

$$
C = E[(F - \mu)(F - \mu)dw]
$$

**Step 8:** The covariance matrix $C_{ij}$ is constructed as follows

$$
C_{ij} = \begin{bmatrix}
\begin{array}{c}
E[f_i - \mu_i](f_j - \mu_j)dw \\
E[f_i - \mu_i](f_j - \mu_j)dw \\
\vdots \\
E[f_i - \mu_i](f_j - \mu_j)dw
\end{array}
\end{bmatrix}
$$

**Step 9:** To determine the malicious attacks the entropy is calculated

$$
Entropy = H = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{dw} C_{ij} \ln C_{ij}
$$
Step 10: End

The use of entropy and distributions of traffic features to analyze the traffic patterns to save the network from DDoS attacks.

The reCAPTCHA controller technique is enriched with the parameters frequency and entropy as rule metrics to obtain the entropy deviation. The identification of threats can be achieved by using belief scores obtained.

The reCAPTCHA controller selects a CAPTCHA (Backin 2000) with the required difficulty level dynamically and is sent to the user. User solves the given CAPTCHA within the required time. The difficulty level is higher on adversary condition and the user has to spend more time and resources to solve the CAPTCHA. It is assumed that only the legitimate user spends to solve CAPTCHA and the attacker gives up his attempt.

The reCAPTCHA controller checks for the time taken for solving and the correctness of the solution. If both conditions satisfied, the user allocates with the resources to the client. The successful classification of botnet traffic and legitimate has been achieved. Considerable amount of fluctuation in attack behaviors are identified by the proposed system.

V. RESULTS AND ANALYSIS

A. SIMULATION ANALYSIS

The simulation studies are done using NS-2.28 (McCanne et al. 1997). The different parameters are used to prove the efficiency of system to examine the flooding DDoS attacks. The various network parameters are considered such as Packet Delivery Ratio (PDR), Average Latency (AL), Detection Rate (DR) and Energy Consumption (EC). Here, EC is the rate of energy utilized by the nodes in the data transmission process. Average Latency (AL) denotes the time taken for source vehicle to deliver a packet to the desired destination vehicle. Detection Rate (DR) is used to prove the efficiency of the proposed technique. For example, if DR value is maximized, it is witnessed that the nodes which are with malicious behavior are removed from the network. Hence, the legitimate nodes only participate in the VANET.

- **PDR**: PDR is the measurement requires ensure the reasons for dropping of packets due to weak wireless channel and also to consider malicious behaviors of nodes in the network for intended packet dropping

- **EC**: Energy consumption is the metric used to observe the rate of energy used by the nodes for the data transmission process and the minimum value of parameter denotes the better life time of the system.

- **AL**: This notation measure the queuing time before the message reaches in between from source to destination node can be denoted as average latency and transfer times.

- **DR**: Detection rate is used to determine the potentiality of the detection system to perceive the malicious nodes and attackers.

The Simulation of results includes the various initial parameters used are shown in Table IV. The experimental results of VANET, is projected using the wireless channel is used as the type of channel. In sequence, to predict the radio propagation in VANET infrastructure, the radio wave propagation model is used. Drop tail queue is used to manage the packet delivery. This queue accepts packets till the end of the queue until its empty and packets dropped when it becomes full. The Omni- directional antenna is preferred to use for the reason to cover the varying degree of coverage in all the directions. The high dynamic moving of nodes can be represented by using the random way point model, where it demonstrates the movement of vehicles and locations change over time. In the two-ray ground model predicts the line of sight component.

<table>
<thead>
<tr>
<th>TABLE IV SIMULATION ENVIRONMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameters</strong></td>
</tr>
<tr>
<td>Channel type</td>
</tr>
<tr>
<td>Radio-propagation model</td>
</tr>
<tr>
<td>Antenna type</td>
</tr>
<tr>
<td>Interface queue type</td>
</tr>
<tr>
<td>MAC type</td>
</tr>
<tr>
<td>Maximum packets in Queue</td>
</tr>
<tr>
<td>Topographical Area</td>
</tr>
<tr>
<td>Mobility scenario</td>
</tr>
<tr>
<td>Pause time</td>
</tr>
<tr>
<td>Mobility Model</td>
</tr>
</tbody>
</table>
The traffic parameters are represented in Table V. This attributes are used to set, that helps to implement the application and transport layer stuffs. The Constant Bit Rate (CBR) is used as the application layer traffic agent used in this research work. CBR is represent the bit rate related to which the output data consumed is constant.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traffic Agent</td>
<td>CBR</td>
</tr>
<tr>
<td>Transport Agent</td>
<td>UDP</td>
</tr>
<tr>
<td>CBR Rate</td>
<td>10 kbps</td>
</tr>
</tbody>
</table>

In Table VI the below given parameters are used as variable parameters. The proposed methodology uses AODV protocol and different network sizes. The variable parameters are represented here.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routing protocols</td>
<td>Normal AODV</td>
</tr>
<tr>
<td>Number of Nodes</td>
<td>50, 100, 150, 200</td>
</tr>
</tbody>
</table>

1) PACKET DELIVERY RATIO

In this work, the analysis, and implementation of DDoS are carried out. In order to analyze the attack scenarios clearly and to understand the impact of attacks, the following techniques have been implemented:

- T1-Analysing DDoS AODV
- T2-Analyzing Firecol Technique
- T3-Analyzing reCAPTCHA controller

For all the above techniques, various network sizes are used to make aware of the behavior of the attacks. Table 5 gives the performance of the AODV DDoS. The experiments in the T1, have been carried out with malicious nodes in AODV, with varying network size.

**T1-Analysing normal AODV under DDoS attack**

In Table VII, gives the reference of simulation done for DDoS attack on AODV protocol and figured out the average values measured by executing the scenario for three times each are tabulated.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Nodes</th>
<th>Routed Packets</th>
<th>Dropped Packets</th>
</tr>
</thead>
<tbody>
<tr>
<td>AODV</td>
<td>50</td>
<td>676</td>
<td>7780</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>2422</td>
<td>9112</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>6721</td>
<td>8910</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>11141</td>
<td>9492</td>
</tr>
</tbody>
</table>

1. FIGURE 2. Analyze Packet Drop Ratio in AODV

From the Table V, the following observations are made:

The different network sizes are considered such as 50, 100, 150 and 200. The Figure 2 shows the performance under different network sizes and significant performance metrics of the DDoS attack on AODV.

The dropped packets are increased from 7,780 p/s (packets per second) to 9,492 p/s when the network comprises of malicious nodes.

**T2-Firecol Technique under DDoS**

In Table VI, gives the reference of simulation done for Firecol detection system under the DDoS attack and figured out the average values measured by executing the scenario for three times each are tabulated.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Nodes</th>
<th>Routed Packets</th>
<th>Dropped Packets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firecol Technique</td>
<td>50</td>
<td>1014</td>
<td>7442</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>2652</td>
<td>8882</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>7033</td>
<td>8598</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>15062</td>
<td>5571</td>
</tr>
</tbody>
</table>
The routed packets also increased from 1014 p/s to 15062 p/s when the firecol detection technique is applied under DDoS attack. The Figure 3 displays the performance of the Firecol detection technique under significant performance metrics with different network sizes.

T3- reCAPTCHA controller under DDoS Attack

In Table VII, gives the reference of simulation done for the novel proposed methodology reCAPTCHA controller detection system under the DDoS attack and figured out the average values measured by executing the scenario for three times each are tabulated.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Nodes</th>
<th>Routed Packets</th>
<th>Dropped Packets</th>
</tr>
</thead>
<tbody>
<tr>
<td>reCAPTCHA controller</td>
<td>50</td>
<td>1268</td>
<td>7188</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>3921</td>
<td>7613</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>8597</td>
<td>7034</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>17358</td>
<td>3275</td>
</tr>
</tbody>
</table>

The routed packets increased from 1268 p/s to 17358 p/s when the proposed reCAPTCHA controller detection technique is applied under DDoS attack. Figure 4 displays the performance of the proposed reCAPTCHA controller detection technique under various network sizes and performance metrics.

2) SIMULATION STUDY FOR PROPOSED TECHNIQUE

The simulation results for detecting and isolating DDoS in VANETs are presented in this section. The parameters considered here is EC, AL, DR with the existing techniques:
1. AODV without any detection and DDoS attack
2. Firecol Technique
3. reCAPTCHA controller

The Figure 5 made the following observations are:

The proposed reCAPTCHA Controller mechanism has measured with AL as 33 s when there is malicious nodes in the network. The existing methods such as Firecol and AODV are 57.5 s and 90 s respectively. Hence it is practically proved that the proposed reCAPTCHA controller technique noticeably improved the results compared with an existing techniques.

From Figure 5, the following inferences can be drawn. In general, the AL increases, when the percentage of malicious nodes increase. In the proposed reCAPTCHA controller technique, malicious nodes are isolated without any delay which in turn it reduces the latency in the network. The reduced latency also indicates that the connection established with minimum number of hops. Hence, the maximum number of nodes communicates each other with only via the normal nodes. So, the AL is decreased.
VI. CONCLUSION AND FUTURE WORK

In this work, a novel methodology for detecting and isolating the DDoS attacks in VANET is proposed. The evaluation of network performance done by using density based attack analyzer, when there is an attack. It is shown and practically proved that the proposed reCAPTCHA controller technique noticeably improved the results relatively with an existing techniques. Furthermore, the novel method in this paper minimizes the latency and consumption of energy measured by parameter AL and EC respectively, and increases the PDR and DR. The reCAPTCHA controller is proposed and compared with existing techniques.

The proposed reCAPTCHA Controller mechanism has measured with DR as 94.7%. The existing methods such as Firecol and AoDV are 74.9% and 60% respectively. It is practically proved that the proposed reCAPTCHA controller technique noticeably improved the results compared with an existing techniques.

VI. CONCLUSION AND FUTURE WORK

In this work, a novel methodology for detecting and isolating the DDoS attacks in VANET is proposed. The evaluation of network performance done by using density based attack analyzer, when there is an attack. It is shown and practically proved that the proposed reCAPTCHA controller technique noticeably improved the results relatively with an existing techniques. Furthermore, the novel method in this paper minimizes the latency and consumption of energy measured by parameter AL and EC respectively, and increases the PDR and DR. The reCAPTCHA controller is proposed and compared with existing techniques.

The proposed reCAPTCHA Controller mechanism has measured with DR as 94.7%. The existing methods such as Firecol and AoDV are 74.9% and 60% respectively. It is practically proved that the proposed reCAPTCHA controller technique noticeably improved the results compared with an existing techniques, the PDR is 85% in the implemented scheme, while the PDR for existing technique Firecol is 73% and AODV is 54%. As the result shown, enhanced performance of the proposed reCAPTCHA controller detection technique under various network sizes and performance metrics is evaluated. When the network contains malicious nodes, the AL for AODV is 90 s, for Firecol technique is 57.5 s and for proposed reCAPTCHA controller technique AL is 33 s. The EC for AODV is measured as 130 J with occurrence of malicious nodes, for Firecol technique it is recorded as 82 J and for novel implemented scheme EC is 47 J. The latter has very low EC. From this it is known that the proposed reCAPTCHA controller has very high performance compared with existing systems. Hence, the novel technique proposed in this research work, prefers to send information via genuine nodes by isolating the malicious nodes. The novelty and efficiency of the system has been proved with the extensive experiments illustrated in this paper. In future preventing and isolating the attacks by using the hybrid mechanism is effective in terms of security and performance. The hybrid detection system can be designed based on the information theory based metrics and trust evaluation mechanism to improve the performance and accuracy.

REFERENCES


