Security Issues of Solar Energy Harvesting Road Side Unit (RSU)

Qutaiba I. Ali
Computer Engineering
Department Mosul University/Iraq
Qutaibaali@uomosul.edu.iq

Abstract: *Vehicular network security had spanned and covered a wide range of security related issues. However solar energy harvesting Road Side Unit (RSU) security was not defined clearly, it is this aspect that is considered in this paper. In this work, we will suggest an RSU security model to protect it against different internal and external threats. The main goal is to protect RSU specific data (needed for its operation) as well as its functionality and accessibility. The suggested RSU security model must responds to many objectives, it should ensure that the administrative information exchanged is correct and undiscoverable (information authenticity and privacy), the source (e.g., VANET server) is who he claims to be (message integrity and source authentication) and the system is robust and available (using Intrusion Detection System (IDS)). In this paper, we suggest many techniques to strength RSU security and they were prototyped using an experimental model based on Ubicom IP2022 network processor development kit .*

***Index Terms*—Vehicular Ad hoc Network, Network Security, Green Networking, Road Side Unit, Intrusion Detection System.**

I. INTRODUCTION

Many research works suggest that there is a real need for a VANET infrastructure, which consists of various types of fixed nodes performing different actions according to VANET's applications demands. An important class of these nodes are Road Side Units[1,2]. Due to power supply requirements, it was recommended to localize RSUs near to wired electricity sources, such as traffic lights[1,2]. However, such placement limits the area covered by RSUs and thus its provided services. In order to overcome this restriction, it is required to establish a self powered RSUs. In our previous works [3-5], we suggest that RSUs can harvest the energy needed for its work from the surrounding environment, especially solar energy. Such suggestion permits to install RSUs in any place without considering the power supply availability and hence, extensive area is covered by the VANET infrastructure. We also suggest that these RSUs would create an *ad hoc* network in order to assist each other to deliver data packets to their destinations, that's why an ad hoc infrastructure is needed. Each RSU is responsible for providing different VANET services to the vehicles in a certain area of the city, ranging from traffic safety and road monitoring services to Internet access & entertainment services. RSUs, as a part of the VANET infrastructure, receives different packets from vehicles (vehicle status or Internet access request), then forward them to the VANET server via the ad hoc network. As a member in the ad hoc network, RSU also behaves as a router in order to deliver other RSUs traffic to their destinations.

This paper focus on using solar cell energy harvesting in providing an alternative power source to supply RSUs and to manage power provided to these devices. We makes use of our earlier design found in [3] of an efficient ,simple and adaptable energy harvesting module which can be used with different types of embedded RSUs. Although, UBICOM IP2022[16] was selected to be the intended RSU, the adopted energy harvesting circuit can be slightly modified to work with other embedded devices. UBICOM IP2022 is a network processor produced by UBICOM Company and provides the whole solution as a fully integrated platform - the Real Time Operating System (RTOS), the protocol stack, and the necessary hardware. The same device can supports Ethernet, Bluetooth wireless technology, IEEE 802.11, and so on. The key to this approach is Software System on Chip (SOC) technology as shown in Figure(3) [4].

The core of the harvesting module is the harvesting circuit, which draws power from the solar panels, manages energy storage, and routes power to the target system. The most important consideration in the design of this circuit is to maximize efficiency. A DC-DC converter is used to provide a constant supply voltage to the embedded system. The choice of DC-DC converter depends on the operating voltage range of the particular battery used, as well as the supply voltage required by the target system. If the required supply voltage falls within the voltage range of the battery, a boost-buck converter is required, since the battery voltage will have to be increased or decreased depending on the state of the battery. However, if the supply voltage falls outside the battery's voltage range, either a boost converter or a buck converter is sufficient, which significantly improves power supply efficiency. In this work, we used Texas Instruments TPS63000 low power boost-buck DC-DC Converter[3] because it suits our needs. The solar panel is connected to a battery whose terminal voltage determines the panel's operating point along its V-I curve. We ensure the operation at the maximal power point through our choice of battery. Using two parallel AA battery cells with voltage varies between 2.9V and 3.1V, which ensures that the voltage across the solar panel terminals remains close to optimal. To avoid problems such as decreased radio range caused by decreased battery voltage, we use a step up DC-DC converter to provide a constant 3V supply voltage to the battery which provides overcharge and undercharge protection for the batteries[4].

In order to validate the convenience of the security methods suggested in this paper, several practical tests were performed using an experimental network as shown in Fig. 1. The experimental network consists of an ordinary PCs (one of them was programmed to act as a traffic generator to emulate the actions of vehicles and other RSUs and the other PC was programmed to imitate the behavior of the VANET server) supplied with Belkin Dual-Band Wireless PCMCIA Network Card F6D3010 working at different data rates, IP2022 Ubicom platform (i.e., the RSU) was also supplied with the same WLAN NIC, the energy harvesting module and a real time storage oscilloscope. The purpose of performing these experiments is to emulate the real VANET environment in which RSU will be installed.

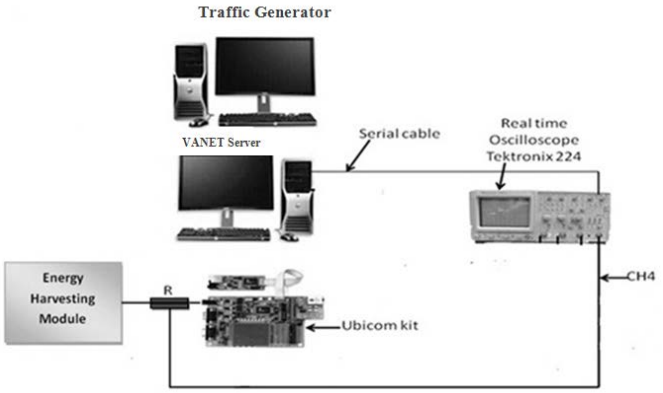


Fig. 1 The experimental network

II. LITERATURE REVIEW

This section presents a survey on the existing works on vehicular network security which had spanned and covered a wide range of security related issues[6]. However RSU security was not defined clearly, it is this aspect that is considered in this paper.

Previous works in[7] and [8] propose the usage of a PKI and digital signatures and presented the problem of certificate revocation and its importance. Also Key management issues in VANET environment was discussed in many papers such as [9] and [10].In [9], the basic structure of VANET and the basic requirement of a key management scheme are introduced, while the authors in [10] proposed a distributed key management scheme with protection against RSU compromise in VANET using group signature.

Some previous works on VANET authentication can be found in [11,12]. These works studies the various methods to achieve secure message and entity authentication and the required methods to obtain flexible, extensible, and efficient VANET verification process.

There have been many papers deals with the different issues related to the VANET Intrusion Detection System (IDS)[13]. Some researches studies the possible attacks that may occur in VANET, their origins, the possible victims. and the anticipated applications[14-17]. On the other hand, there have been a focus on the different IDS strategies to defend against VANET special attacks, such as Sybil attack [18,19],Worm hole attack [20] and Denial of Service (DoS) attack [21].

Security and privacy in VANET are gaining

increasing attention and interest from research communities. The goals of the research works in this field is to provide privacy at different levels such as vehicle to vehicle communication[22] and location privacy[23]. Finally, some research works focus on employing reputation and trustfully based verification methods in order to categorize the different nodes in the VANET[24,25].

III. THREATS MODEL

In this paper we are concentrating on attacks perpetrated against the RSU itself rather than the VANET infrastructure or its users and applications. As shown in Fig. 2, security threats against RSU can take different forms and originate from different sources. These sources can be an insider attackers (attacks (1) & (2) in Fig. 2) which is either a "VANET user" (e.g., vehicles) or a forged RSU. In other words, insider attacker is an authentic user of the network who has some knowledge of network and make use of it for understanding the design and configuration of the RSUs and the whole network. On the other hand, Outsider attackers (attack (3) in Fig. 2) can make use of VANETs' Internet connection to launch their attack from a remote location outside the VANET coverage area.

On the other side, when investigating the possible types of attacks, RSUs are susceptible to a variety of attacks differs in their nature, goals and catastrophic effects. We have made a survey on the possible attacks against RSUs according to their origins and the results of this survey are abstracted in Table I.

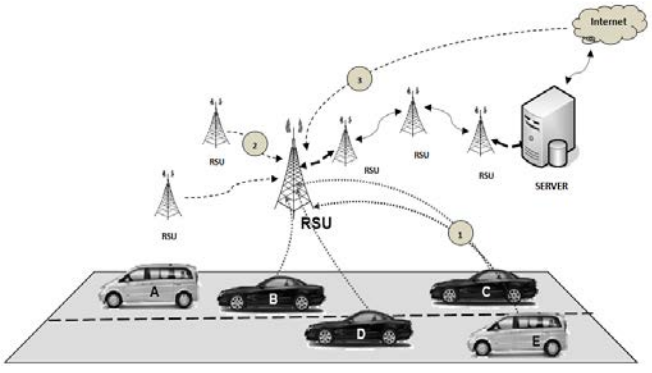


Fig. 2 Threat Model against RSU

TABLE I

SURVEY OF THE POSSIBLE RSU ATTACKS

Attack Source	Attack Type	Description
Insider (Attack 1)	Denial of service (DOS) Attack	The attacker jams the main communication medium and network is no more available to lezitimate users
	Distributed Denial of service (DDOS) Attack	The attackers launch DOS attacks from different locations
	Sybil Attack	The attacker sends multiple messages to multiple RSUs and each message contains different fabricated source identity (ID)
	Administrator Impersonation Attack	The attacker claims the administrator identity
	Application Attack against RSU applications	The attacker changes the content of RSU applications and use it for his own benefits
	Timing Attack	Attacks against the timing of RSU periodic activities
	Monitoring Attack	Passive monitoring of RSU activities
	Energy Exhaustive Attack	Sending high traffic volume to the RSU to exhaustive its stored energy (in the case of battery based RSUs)
Insider (Attack2)	Denial of service (DOS) Attack	The attacker jams the main communication medium and network is no more available to lezitimate users
	Distributed Denial of service (DDOS) Attack	The attackers launch DOS attacks from different locations
	Black hole attack	The attacker can selectively filters or drops traffic from a particular part of the network
	Worm hole attack	Two malicious nodes in a network transfer packets from a private tunnel which they have built by cooperation with each other and if message passes through this tunnel then security breach occurs.
	Application Attack	The attacker changes the content of RSU applications and use it for his own benefits
	Timing Attack	Attacks against the timing of RSU periodic activities
	Energy Exhaustive Attack	Sending high traffic volume to the RSU to exhaustive its stored energy (in the case of battery based RSUs)
Outsider (Attack3)	Denial of service (DOS) Attack	The attacker jams the main communication medium and network is no more available to legitimate users
	Distributed Denial of service (DDOS) Attack	The attackers launch DOS attacks from different locations
	Administrator Impersonation Attack	The attacker claims the administrator identity
	Application Attack	The attacker changes the content of RSU applications and use it for his own benefits
	Monitoring Attack	Passive monitoring of RSU activities
	Energy Exhaustive Attack	Sending high traffic volume to the RSU to exhaustive its stored energy (in the case of battery based RSUs)

IV. THE SUGGESTED SECURITY MODEL

In this section, we will suggest an RSU security model to protect it against the different threats mentioned earlier. The main goal is to protect RSU specific data (needed for its operation) as well as its functionality and accessibility. RSU security model must responds to many objectives, it should ensure that the administrative information exchanged is correct and undiscoverable (information authenticity and privacy), the source (e.g., VANET server) is who he claims to be (message integrity and source authentication) and the system is robust and available. In this paper, we suggest many techniques to strength RSU security.

A. Cooperative-hybrid intrusion detection system

Service availability is an important security issue which means that authorized access of data and other VANET resources is made ready when requested or demanded. This feature could be obtained by protecting the system against different types of attacks using an Intrusion Detection System(IDS). In order to offer a high level of defense against various attacks and to cope against the limited processing and energy resources in the RSU, we suggest a cooperative IDS approach. In this approach, RSUs do not depend only on their local view to make conclusions about the security status of their network, but also cooperate with their VANET server by exchanging security reports to

create a more global and accurate vision about the security situation of the whole network, the possible attacks and their origins. The implementation of the suggested cooperative IDS is shown in Fig. 3. RSUs play the role of an IDS sensors, they generate *"periodically*" their security status reports then forward them to the VANET server. These reports contains the necessary statistics about the number, types and sources of attacks to this RSU at that time. On receiving these reports, VANET server accumulates them, then makes the necessary processing to obtain the final report about the security status of this part of the network. Also, VANET server suggests the necessary IDS reactions to accomplish against these attacks and declares them to the RSUs and the VANET administrator.

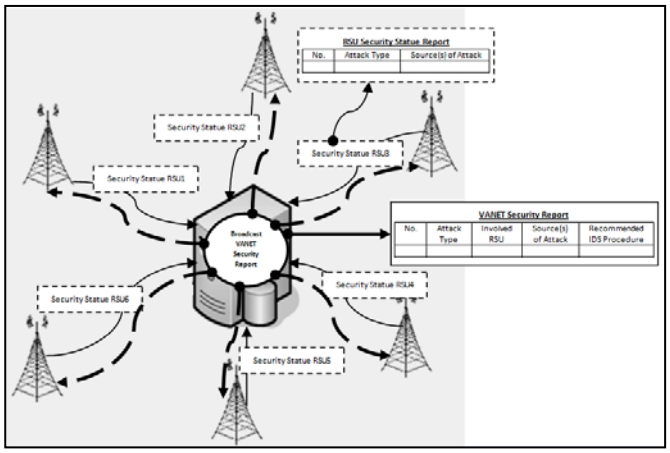


Fig. 3 Cooperative IDS Functionality

Intrusion detection systems must be able to distinguish between normal and abnormal activities in order to discover malicious attempts in time. There are three main techniques that an intrusion detection system can use to classify actions [15]; signature based IDS, anomaly based IDS and behavioral based IDS. One of the main contributions in this paper is the attempt to insert a Hybrid Intrusion Detection System(HIDS) functionality(combines all the three techniques together) into the RSU itself, see Fig. 4. Each one of these IDSs has its own defense strategy against certain classes of attacks. For example, signature based IDS is the best solution against well known Internet attacks, while anomaly based is very effective against Denial of Service (DoS) attacks and finally behavioral based IDS is used to defend against VANET specific attacks[16,17]. As shown in Fig. 4, the input traffic to the hybrid IDS is firstly sampled and processed (in the data processing unit) in order to extract its main features, such as average data rate, maximum data rate and maximum burst size and many others. This input traffic is then converted into a more proper format to be processed by the three different IDSs. The final decision of the suggested hybrid IDS is taken based on the sub-decisions made by the three IDSs. This decision includes the generation of the security report of this particular RSU ,recommending blocking actions against malicious nodes and modifying the routing tables of this RSU in order to avoid insecure routes, see Fig. 4. The details of each of these IDSs is listed below:

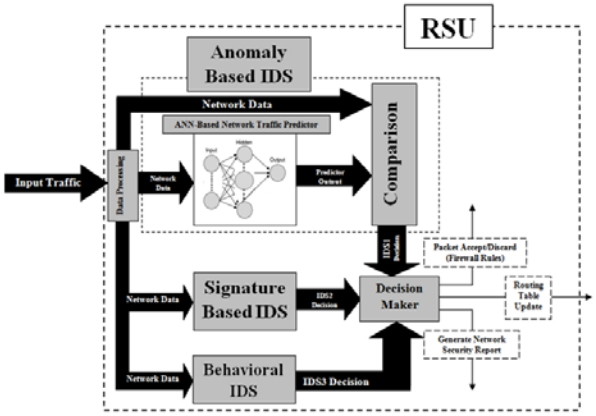


Fig. 4 Hybrid IDS Architecture

1) Signature Based IDS

In signature-based detection systems, the observed behavior is compared with known attack patterns (signatures). Action patterns that may pose a security threat must be defined and stored to the system. Then, the signature based detection system tries to recognize any suspicious behavior according to these patterns. It is already concluded from research in ad hoc networks that severe memory constraints make intrusion detection systems that need to store attack signatures relatively difficult to build and less likely to be effective [13]. To solve this problem, a signature based IDS was achieved by building a light weight, signature based IDS, based on the famous open source SNORT IDS. SNORT is an open source network IDS capable of performing real-time traffic analysis and packet logging on Internet Protocol (IP) networks[13]. SNORT can perform protocol analysis and content searching/matching, and can be used to detect a variety of attacks and probes[13]. SNORT uses a

simple, lightweight rules description language that is flexible and quite powerful. SNORT rules are divided into two sections, the rule header and the rule options[14]. The header contains the rule's action, protocol, source and destination IP addresses including network masks, and the source and destination ports. All options are defined by keywords specifying which fields of the packet should be inspected, such as TTL and content[15].

Based on the previous discussion, intrusion detection can be divided into two procedures: packet filtering based on header fields and string matching over the packet payload. Regarding RSUs, as they have limited processing and energy constrains, the addition of further tasks (such as an IDS program) may affect seriously on its performance, so that, the current design takes these constrains into consideration using the following procedure:

1. The RSUs were loaded with specified rules set (not all rules) which represent the most series attacks at that time. The determination of these rules as "important" is achieved using IDS sensors (i.e., other RSUs) distributed around the network. These IDS sensors monitor the network status (from security point of view) and prepare a report of the most common attacks at that time. These reports are sent to the VANET Server for further processing.
2. VANET Server collects the reports from the IDS sensors and analyzes them to assign the most common risky attacks at that time. Also, it has the classification and processing program which is used to classify the SNORT rules to speed up the searching process at RSU. After that, the server will broadcast the processed rules set to all RSUs that exist in the network.
3. In order to keep the efficiency and performance of the RSUs, a new rules processing algorithm is suggested. The main idea of the suggested algorithm can be abstracted through implementing the preprocessing part of algorithm (the building tree for packet filtering and building tree for string matching) in the VANET server and only the searching part of algorithm (which performs the searching tree of packet filtering and searching tree of string matching) is implemented in the RSU. Aho-Corasick (AC) algorithm is chosen in this paper to act as the classification and processing algorithm. The essence of the Aho-Corasick algorithm involves a preprocessing step (at the VANET server) which builds up a state machine that encodes all of the strings to be searched [13]. The preprocessing part or building trees part will be sent as an update file from the server to the RSUs.
4. During searching phase (at RSU), a match with a SNORT rule is determined if it has prefix match with the source and the destination prefixes, exact match with the protocol, and range match with the source port and the destination port, see Fig. 5. The searching phase in the suggested Tree algorithm is immediately finished without searching the complete trie if an input packet matches a priority rule. This property effectively improves the searching performance. Also, the searching proceeds to the left or right according to the sequential inspection of destination address bits starting from the most significant bit. If there is a match with all the fields in a tree, it is considered as “match” and its priority number is remembered. The searching will be stopped immediately in case if it ends with a match with a priority rule or at a leaf while it is always finished at a leaf in other trie-based algorithms.

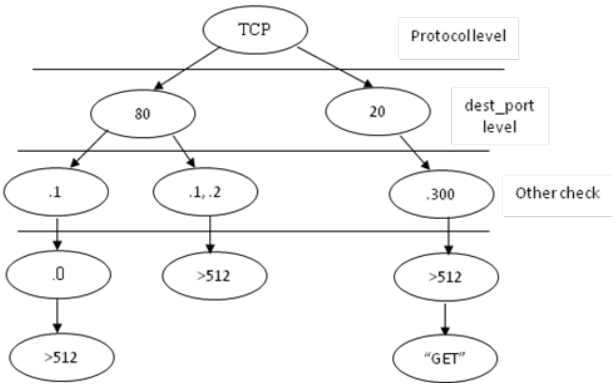


Fig. 5 Tree based searching method

In order to evaluate the performance of the suggested Ubicom based IDS, several tests were implemented. Experimental measurements of the searching algorithms of each phase were performed as two steps. In the first one, results were discussed by computing the necessary memory storage based on the number of rules that can be stored in the memory of the RSU, see Fig. 6. While in the second one, results were collected by measuring the total response time of the proposed IDS when processing packets having different types of internet attacks, see Table II. It is obvious that the suggested signature based IDS has an acceptable performance (with respect to the nature of an ad hoc network)

and its rules set occupies low space of the available storage memory.

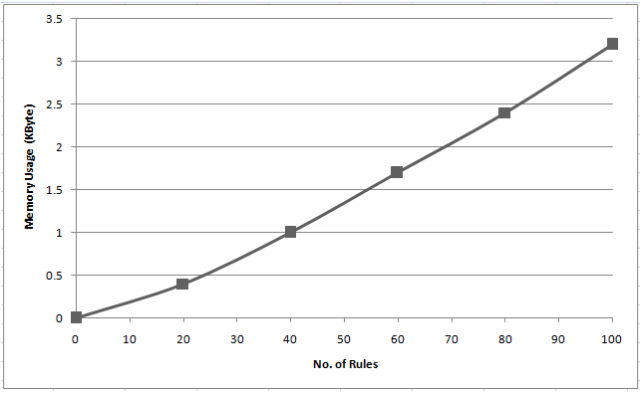


Fig. 6 Ubicom IP 2022 memory Usage vs. No. of IDS Rules

TABLE II
THE TESTED IDS ATTACKS

Attack type	Signature length (byte)	Depth (byte)	Packet length (byte)	Total time (µsec)
1. DDOS	--	--	106	35.822
2. DOS1	6	32	540	196
3. DOS2	14	14	790	113.03
4. DOS3	--	--	400	33.22
5. RESPONCE	8	8	150	281.00
6. EXPLOIT1	6	15	350	98.00
7. EXPLOIT2	3	145	1200	5032.11
8. WEB-CLIENT	7	122	700	2096.2
9. EXPLOIT3	8	92	1400	510.15
10. WEB-COLDFUSION	20	52	900	5607.01

2) Anomaly Based IDS

This type of IDS focuses on normal behaviors, rather than attack behaviors. These systems describe what constitutes a “normal” behavior (usually established by automated training) and then flag as intrusion attempts any activities that differ from this behavior by a statistically significant amount. The intrusion detector learning task is to build a predictive model (i.e. a classifier) capable of distinguishing between bad intrusions and normal connections. Recently, an increasing amount of research has been conducted on applying neural networks to detect intrusions, so that we follow this approach[15]. As shown earlier in Fig. 4, the heart of our anomaly IDS is the prediction algorithm which actually makes use of an artificial neural predictor, which is a three-layer neural network predictor has 20 network inputs where external information is received, and one output layer with one unit where the solution is obtained. The network input and output layers are separated by one hidden layer composed of 10 units. The connections between the units indicate the flow of information from one unit to the next, i.e., from left to right. In order to make meaningful predictions, the neural network needs to be trained on an appropriate data set. Basically, training is a process of determining the connection weights in the network. The final goal is to find the weights that minimize some overall error measure, such as the sum of squared errors or mean squared errors. We have developed a neural predictor and performed experiments to prove its accurate prediction ability with low overhead suitable for dynamic real time settings similar to this system model. Our 20:10:1 network with a learning rate of 0.25 has reduced the mean and standard deviation of the prediction errors by approximately 65% and 73%, respectively. The network needs a 30 minutes to be trained with more than 1000 samples, and then makes accurate predictions without the need to be trained again.

Usually, the traffic data volume is represented by a time series. A time series is a sequence of time ordered data values that are measurements of a physical process. In this particular study, the total network traffic (in bps) received from/transmitted to the different vehicles and other RSUs, are collected every five minutes (it was assumed that each vehicle generates a single 100 Byte/s (safety or status) packet when passing the street in which RSU is installed, while each RSU generates an 1000 Byte packet (traffic report sent to the VANET server) with a packet rate of 10 packet/Minute)[3]. Due to the enormous amount of data, a reduction of data is necessary. With a five minute time interval the data oscillates too much, and the random part is high. Therefore, it is necessary to aggregate them to fifteen minute intervals. The mean value is built with three values. To obtain a smooth time curve an average of three values can be computed. The main advantages of this representation are the great reduction in the amount of data in the database, and the easy handling in the training and testing processes. The network is designed to predict the traffic volume given the past four values of the time series. A set of 1000 consecutive 15 minute samples was extracted from the data available. This is the volume of 7 days. The set was divided in a training

set (5 /7 days of week) and a test set (2 /7 days of week). The model generates a forecast for the next 24 hour period from the daily traffic profile. Fig. 7 shows the temporal variation of the target and output traffic volumes for randomly selected day. The network was extended for the whole set of data, and the results were quite satisfactory. In the figure, the comparison of the original traffic volume with the neural network predicted values for 24 hours can be seen. As shown in Fig. 7, the predicted and measured values are in close agreement. Evaluation of the model performance can be done by the *Mean Square Error*, calculated as the difference between forecasted and actual demand. The average errors for the forecasting up to 24 hours are about 0.007.

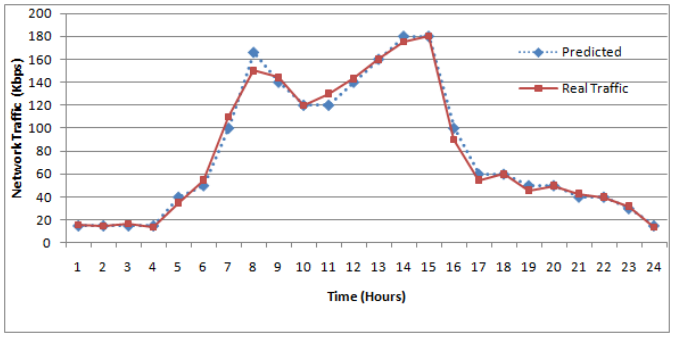


Fig. 7 Outcome of ANN based anomaly IDS network traffic predictor

The performance of this anomaly based IDS is evaluated and monitored by an evaluation algorithm to ensure the ability of the ANN predictor to adapt to load changes over time. This algorithm involves developing a performance monitoring mechanism to sufficiently adapt to meaningful workload changes over time. However, it must also have the ability to avoid overreacting to noise in workload fluctuations. That said, the performance monitoring mechanism must balance adaptability with stability. One unique feature of the neural predictor is that it can be trained with the most up-to-date training data to reflect workload changes when possible. A major duty of this algorithm is to find the suitable training epochs during execution, because the system may oscillate if the epochs are too short, or it may not be able to dynamically adapt if the epochs are too long. The training phase takes place at regular time intervals, making it easier to identify training epochs. As persistent load increase and decrease caused by shifts in vehicles' requests tend to occur on the scale of hours rather than seconds, the training phase can be performed each 24 hour.

The output of the neural predictor is compared with the network data in order to evaluate the abnormality of the input traffic. If the difference between them exceeds a certain threshold (set by the VANET administrator) for a predetermined amount of time (also set by the VANET administrator) then an attack alert is generated and sent to the decision making stage for more assessment, see Fig. 4.

3) Behavioral based IDS

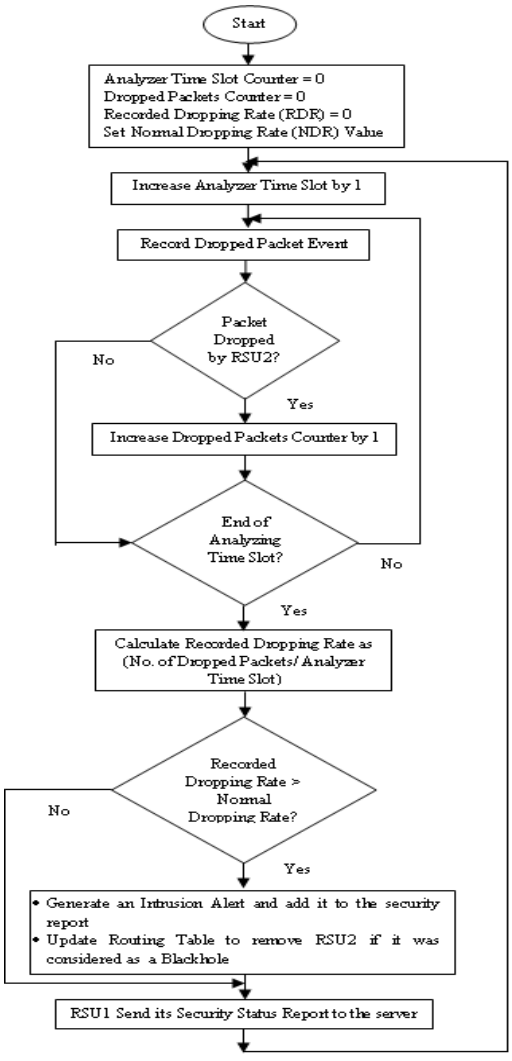
It is also based on deviations from normal behavior in order to detect attacks, but they are based on manually defined specifications that describe what a correct operation is and monitor any behavior with respect to these constraints. This is the technique we use in our approach. It is easier to apply in VANETs, since normal behavior cannot easily be defined by machine learning techniques and training.

In order to clarify the principles of our approach, we build a behavioral based IDS to detect two types of VANET specific attacks: Black hole attack and Energy Exhaustive Attack.

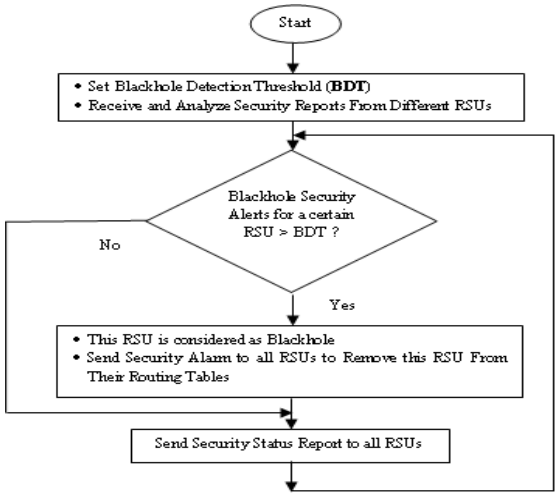
❖ Black hole attack occurs when a compromised node drops a packet that is bound for a particular destination. In this way, an attacker can selectively filter traffic from a particular part of the network. Other possible variations of selective forwarding can involve dropping all packets or randomly dropping packets. Although random dropping is less disruptive, it can also be much harder to reliably detect and trace[20].

These specifications for detecting black hole and selective forwarding attacks can simply be a rule on the number of packets being dropped by an RSU. Each of the RSUs will apply that rule for itself to produce an intrusion alert. The adopted approach would be to increment a counter every time a packet is dropped and produce an alert when this value reaches a threshold. However, we should take into account loss of packets due to other reasons. A better method to detect black hole activity by a certain RSU is to set a threshold of the rate at which packets are dropped (we called Recorded Dropping Rate (RDR)), and when this is reached an alarm can be generated. Therefore we require each RSU to keep track of the packets not being forwarded within a fixed amount of time we called analyzer time slot, during which it creates statistics on the

overheard packets. At the end of each time slot an alert may be produced according to the threshold criterion, which is sent by that RSU to the server as a security report. Then the next time slot is started, and the same process is repeated periodically, for all RSUs, see Fig. 8a. The next design issue we need to solve is who is going to make the final decision that a node is indeed an intruder and the actions should be taken. In our approach, we use a cooperative decision making approach, where the RSUs and their associated VANET server *cooperate* in order to decide whether a certain RSU is launching a selective forwarding attack and take appropriate actions. For example, if the security reports received by the server states that more than half of the RSUs have raised an alert, then the target RSU is considered compromised and should be removed from the routing tables of the other RSUs, see Fig. 8b.



(a)



(b)

Fig. 8 The suggested anti-black hole attack IDS procedure (a)RSU side (b) VANET server side

❖ Energy Exhaustive Attack is a special type of Denial of Service (DoS) attack which is based on sending high traffic volume to the RSU to exhaustive its stored energy (in the case of battery based RSUs) rather than jamming the communication medium. Our approach to defend against this attack is to adopt a proper power management scheme.

In this paper, we suggest that RSU should follow Sleep/Active periods scheme and performs according to its available energy, specifically, the service rate of the RSU is determined as a function of the RSUs' power budget. In this case we need to derive a relation among Duty Cycling periods (Sleep/Active), Average Service Rate (ASR) and the Available Energy(AE). We firstly start by defining the following terms:

- Average Service Rate (ASR) is the average of total traffic (in bps) transmitted from and received by the RSU.
- Duty Cycling Periods: In this paper, time is divided into (1 s) slots. Hence, Duty Cycle is the ratio of the active periods to the total slot time.
- Available Energy (AE) is the summation of the residual energy in the batteries from the last day plus the expected energy in the next day.

Our approach involves the following steps, see Fig. 9:

In the beginning of each *working day*, RSU calculates the Available Energy (AE) as follows:

$$AE = RE + EE \quad (1)$$

Where:
RE is the Residual Energy from the last day
EE in the Expected harvested Energy in the current day

RE of the batteries can be found as:

$$RE = (\text{Initial Energy} + I_{in} \times \text{Effective Charging Time}) - I_{out} \times 24 \quad (2)$$

It is obvious that in order to calculate RE, RSU needs to measure the current flowing to/from the batteries(I_{in} & I_{out} respectively) during the whole working day. We make use of the Ubicom's integrated 12 bit A/D convertor to achieve this task. Our measurement process involves taking a sample every one second, then calculating the average current values in each hour. Effective Charging Time represents the number of hours in which the current drained from the solar panels is greater than zero.
In order to estimate the value of EE, we suggest that the VANET server should broadcast (to all RSUs) the weather forecasts and the effective charging time for this particular day. This weather report includes the expected weather (Sunny, Cloudy or Rainy) and the number of useful charging hours. As a function of current measurement procedure mentioned earlier, RSU can determine the current value expected according to its historically recorded current values in a similar weather conditions, and hence, EE could be calculated as:

$$EE = \text{Average Expected Current} \times \text{Effective Charging Time} \quad (3)$$

The next step is to calculate the Average Service Rate (ASR) of the RSU in this particular day according to the value of AE. The relation between Service Rate (SR) and AE could be derived by determining the power consumed according to RSU activities as follows:

$$AE = E_{TX} + E_{RX} + E_{Proc.} + E_{Sleep} \quad (4)$$

$E_{TX,}$ is the energy consumed during data transmission and can be expressed as:

$$E_{TX} = I_{TX} \times \text{bit time during transmission} = SR(I_{TX}/n \times \text{Data Rate}) \quad (5)$$

Where (I_{TX}) is the current drained by WLAN NIC when working in TX mode and (n) is the ratio between RX to TX periods

E_{RX}, is the energy consumed during data reception and can be expressed as:

$$E_{RX} = I_{RX} \times \text{bit time during reception} = SR(I_{RX} \times (n-1)/n \times \text{Data Rate}) \quad (6)$$

Where (I_{RX}) is the current drained by WLAN NIC when working in RX mode

$E_{Proc.}$ is the energy consumed during data processing and can be expressed as:

$$E_{Proc.} = SR\ (\ I_{Proc.}/\text{Data Processing Speed of the RSU}) \quad (7)$$

Where ($I_{Proc.}$) is the current drained by Ubicom Motherboard during processing

E_{Sleep} is the energy consumed during Sleep mode and can be expressed as:

$$E_{Sleep} = SR((I_{Sleep} \times \text{Data Processing Speed} - I_{Sleep})/ \text{Data Processing Speed}) \quad (8)$$

Where (I_{Sleep}) is the current drained by Ubicom Motherboard in Sleep mode

The next step is to calculate the Average Service Rate (ASR) of the RSU in this particular day according to the value of AE as:

$$ASR = 0.5\ (AE - d) / (a + b + c - e) \quad (9)$$

Where:
$a = (I_{TX}/n \times \text{Data Rate})$
$b = (I_{RX} \times (n-1)/n \times \text{Data Rate})$
$c = (\ I_{Proc.}/\text{Data Processing Speed of the RSU})$
$d = I_{Sleep} \times 24$
$e = (I_{Sleep} / \text{Data Processing Speed})$

The next step, is the mapping procedure of ASR value to suit the different rates of the applied load. ASR represents the service rate for middling number of vehicles, so that mapping is necessary in order to afford variable service rate according to the variation in the number of vehicles. This step requires that RSU has the ability to *predict* the future load according to its historical behavior. Our prediction algorithm actually makes use of the artificial neural predictor mentioned earlier.

The last step is to calculate the Sleep period in each time slot (i.e., 1s) as:

Average Sleep Period =1- (ASR/Data Rate) (10)

After performing the above calculations, RSU can begin its work safely. As each time slot is divided into Active and Sleep periods, Ubicom's enters the sleep period first. At the same time, different types of data packets are accumulated in the WLAN NIC buffers(which is still ON). When Active period starts, Ubicom board wakes up and begin to process the packets received from its WLAN NIC, see Fig. 9.

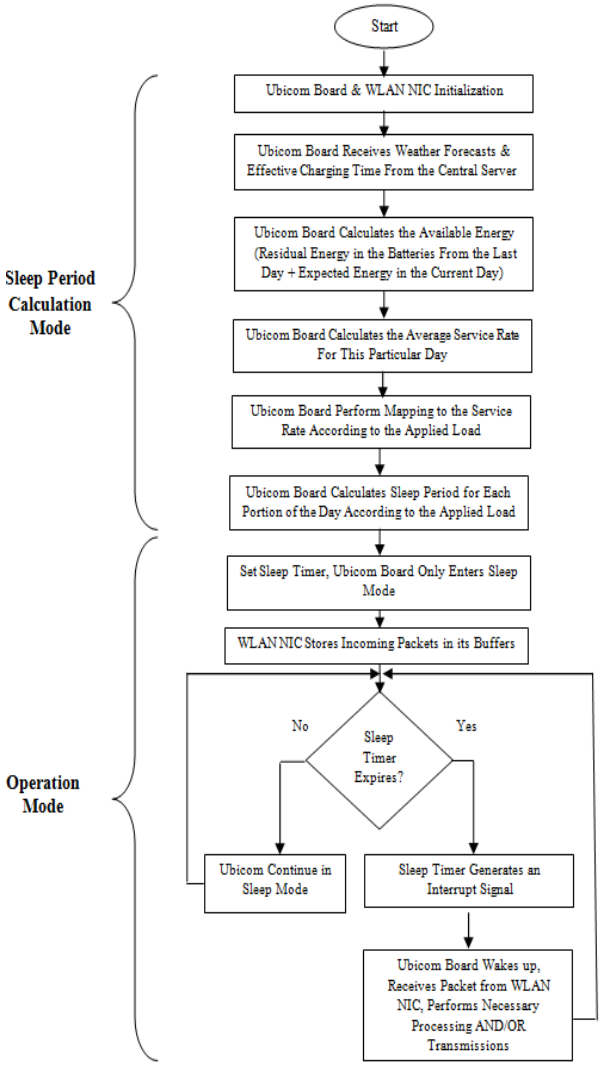


Fig. 9 The suggested anti-energy exhaustive attack IDS procedure

The suggested technique was tested to evaluate its ability to manage the energy consumed by RSU, and hence to defend against unmanaged network traffic conditions (such as those result from the Energy Exhaustive Attack). The initial settings of these experiments could be found in Table III. The Purpose of these experiments is to examine the ability of the suggested method to adapt against different working conditions wherein different Available Energy (AE) levels were assumed. Table IV lists the different values of ASR and ASP obtained from these scenarios, where Residual Energy (RE) stands for a sample case of 50% battery charging percentage and (N) is the number of paralleled solar panels. Fig 10 shows the daily behavior of the RSU while tracking the load offered by its associated vehicles and other RSUs. This Figure was plotted by mapping the ASR (the service rate for an *average* number of vehicles) to be adjusted in response to the number of the daily served vehicles. It is noted that the suggested power management technique was able to adapt its performance according to the available energy levels and hence continue to function in a pre-managed and planned manner.

TABLE III
INITIAL EXPERIMENTAL PARAMETERS

Data Rate	18 Mbps (IEEE802.11a)
Traffic Characteristics	n = (RX Traffic/TX Traffic)= 4
Data Processing Speed of the RSU	24 Mbps *From Ref.*[16]
I_{TX}	150 mA
I_{RX}	120 mA
$I_{Proc.}$	150 mA
I_{Sleep}	1 mA
Battery Characteristics	3 v, 2800mAh
Solar Panel Dimensions	4.25" x 2.5"

TABLE IV
ASR & ASP VALUES UNDER DIFFERENT CONDITIONS

RE (mAh)	N	Weather Condition	AE (mAh)	ASR (Mbps)	ASP (s)
50%	1	Sunny	1910	2.984375	0.834201
50%	1	Cloudy	1666.5	2.603906	0.855339
50%	1	Rainy	1558.4	2.435	0.864722
50%	3	Sunny	2930	4.578125	0.74566
50%	3	Cloudy	2199.5	3.436719	0.809071
50%	3	Rainy	1875.2	2.93	0.837222
50%	6	Sunny	4460	6.96875	0.612847
50%	6	Cloudy	2999	4.685938	0.73967
50%	6	Rainy	2350.4	3.6725	0.795972

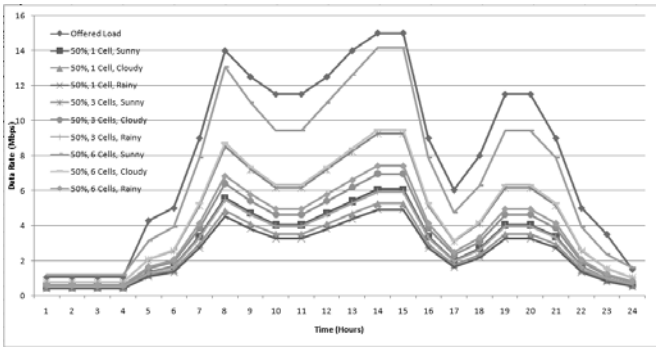


Fig. 10 RSU daily behavior according to power management functionality

B. Secured Embedded Web Server

As an embedded system, RSU settings can be reconfigured remotely by the VANET administrator to adapt against the variant conditions. As shown earlier in Fig. 1, RSU was supplied with an embedded web server so that its web pages create an interface between the remote administrator and the RSU components. It is obvious that the remote RSU reconfiguration procedure made by the VANET administrator is a risky and sensitive task and must be highly secure in order to guarantee the proper functionality of RSUs. In this paper we suggest the following procedure to maintain the security of the RSU web pages access:

1) Bidirectional Entity Authentication

Prior to accepting the remote control request (i.e., accessing the reconfiguration web page of the RSU) made by the VANET administrator, RSU must check his identity. This can be done by adopting a particular challenge response procedure suggested in this paper, see Fig. 11. The challenge is a time-varying value which is a random number and a timestamp which is sent by the server. The RSU applies a function to the challenge and sends the result, called a response, to the server. The response shows that RSU knows the secret. We called this procedure a "bidirectional" because it confirms VANET administrator identity to the RSU and vice versa. This method assumes that the clocks of both sides are synchronized and they also have synchronized and equivalent pseudo random number generators (having the same code functionality, their seeds are equal and generate their outputs at the same time intervals). The challenge/response begin when the VANET administrator sends an encrypted packet contains the username, a generated random number (RND1) and a timestamp (T1). This arrangement proves the identity of the VANET administrator in several aspects:

1. The value of RND1 is already known by RSU because its pseudorandom number generator is synchronized with that of the server. Only the server can generate this value at that time. RSU checks the value of RND1 which is the first prove of the administrator identity.
2. The value of T1 is a time stamp (represents the time value in the server side) which is synchronized with the RSU clock. This arrangement prevents reply attack and can be considered as the second prove of the administrator identity.
3. This packet is encrypted using the Advanced Encryption Standard (AES) algorithm. The 128 bit key, we called Authentication key, is known only by the two sides and is considered as the third prove of the administrator identity.

If the request passed the identity checking procedure, then RSU accepts the connection and sends a similar packet so that its identity is also proved to the server

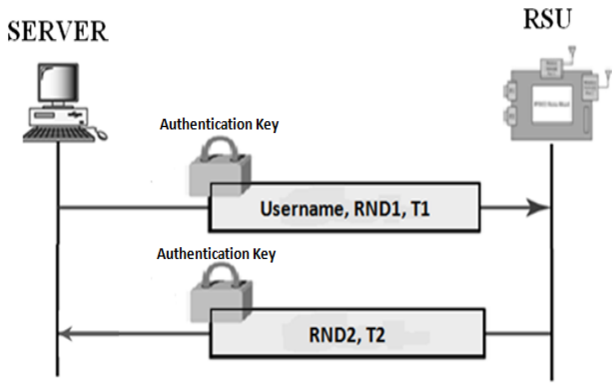


Fig.11 The suggested bidirectional entity authentication

2) Bidirectional Message Confidentiality, Integrity & Authentication

In order to obtain Confidentiality, Integrity & Authentication for the web pages transacted between the VANET server and each RSU, the packets transferred between the Server and the RSU are encrypted (using a secret 128 bit AES key) and sent together with their Hashed Message Authentication Code (HMAC), see Fig. 12. HMAC creates a nested MAC by applying a keyless hash function (Secure Hash Algorithm 2 (SHA2) in our case)to the concatenation of the message and a

symmetric key. A copy of the symmetric key is prepended to the message. The combination is hashed and the result of this process is an intermediate HMAC which is again prepended with the same key, and the result is again hashed using the same algorithm. The final result is an HMAC. The receiver receives this final HMAC and the encrypted message and creates its own HMAC from the received message and compares the two HMACs to validate the integrity of the message and authenticate the data origin. Although Fig. 12 shows the data transaction from the server to an RSU, this procedure was implemented in both directions to provide security to the whole session. In order to measure the additional delay added to the packet creation procedure as a result of applying these security methods, several tests were performed to determine the Ubicom's performance for AES encryption/decryption and HMAC (SHA2) for different packet lengths as shown in Fig. 13.

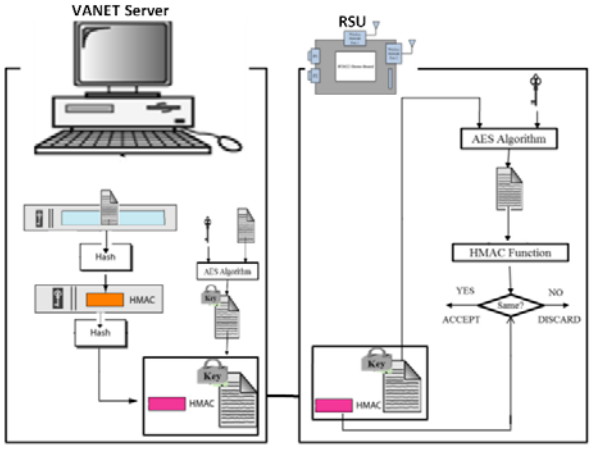


Fig. 12 The suggested bidirectional message confidentiality, integrity & authentication procedure

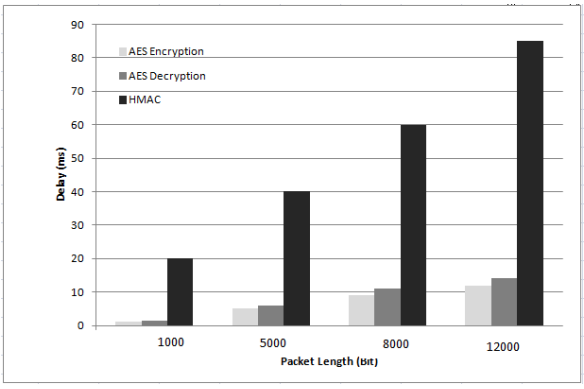


Fig. 13 Ubicom's performance for AES encryption/decryption and HMAC (SHA2) for different packet lengths

3) Key Generation & Distribution

This section discuses the different secret keys used in this paper and the methods adopted to generate and distribute these keys (the public key encryption methods consumes more resources and needs high processing capabilities to perform their calculations, so that we decided to use symmetric key encryption methods only due to the relatively limited processing power of an embedded device such as an RSU).

As a result of their interaction, VANET server and its associated RSUs exchange different types of packets. In order to immunize them, we suggest that AES algorithm is used for encrypting certain types of these packets because of their importance from security point of view. However, different sets (or pairs) of AES keys are needed to encrypt these packets in their various sessions. Table V summarizes the required AES keys, their purpose and source-destination pairs. We assume that an initial values of these keys were pre-injected into each RSU prior to install them in the field (Factory Default Settings).

However, the values of these keys are updated at regular time intervals (remember that we assumed that the Clocks of the VANET server and RSUs are synchronized) to enhance the security of the system. In order to change the values of these keys silently in an efficient and secure fashion, we suggest the following method:

1. In each RSU, there is a long period pseudo random number generator (PRNG) routine. Another identical copy of this PRNG exists in the VANET server, see Fig. 14.
2. Each pair of these PRNGs are firstly synchronized off-line prior to installing the RSU in the field. Synchronization procedure includes feeding the two routines with the same seed values, then beginning the random numbers generation procedure until they produce the same sequence. This initialization point is saved in the RSU and the server and added to the (Factory Default Settings).
3. At this point, the two PRNGs are ready to generate synchronized random numbers which will be used for different purposes such as generating the updated secret keys and the random number used in the challenge response procedure mentioned earlier.
4. In order to maintain the synchronization among them, VANET server and its associated RSUs

perform periodic synchronization tests. These tests begin from the server side and involves sending an encrypted challenge packet (similar to that shown in Fig. 11) to each RSU individually. This packet contains a sequence of random numbers generated by the PRNG routine in the server side and a time stamp. On receiving this packet, the RSU performs the identity check procedure mentioned earlier and generates an equivalent sequence. If the two sequences are equal, then RSU sends a positive acknowledgment to the VANET server.

5. In the case of missing the synchronization between them, VANET server and this particular RSU agree to reset their secret keys values and their PRNG pairs to their Factory Default Settings.
6. The above transactions among the VANET server and its associated RSUs are susceptible to many type of attacks and care must be paid to immunize the messages and their origins against them. We suggest to follow bidirectional entity authentication in order to check the identities for both sides, and all the messages are encrypted and sent together with their HMAC in order to obtain message confidentiality, authentication and integrity.

TABLE V
THE REQUIRED AES KEY GROUPS

Key Name	Purpose	Source	Destination
Keys group1 (Multiple Keys, one for each RSU)	To encrypt RSUs security statue reports	Each RSU has a different key	VANET server
Key2 (One key)	To encrypt VANET server security report	VANET server	All RSUs
Keys gruop3 (Multiple Keys , one for each VANET server_to_RSU connection)	Bidirectional Entity Authentication Key	VANET server	Certain RSU
		Certain RSU	VANET Server
Keys group4 (Multiple Keys, one for each RSU)	Confidentiality, Integrity & Authentication for Web Page Messages Transacted Between VANET Server and an RSU	VANET server	Certain RSU
		Certain RSU	VANET Server

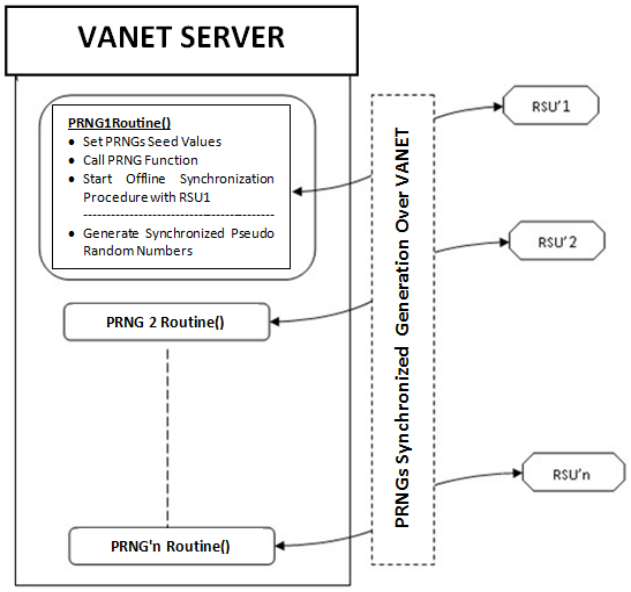


Fig. 14 PRNG pairs in VANET server and RSUs

V. CONCLUSION

In this paper , different methods were suggested to protect solar energy harvested Road Side Units (RSUs) against various types of internal and external threats. The suggested defense strategies took into account the embedded nature of an RSU and hence the recommended solutions make a compromise between highly secured and good performed system. Our solutions aimed to offer the main security features for the message (confidentiality, integrity and authentication), entity authentication and system availability. To the best of our knowledge, the combination of such security methods, algorithms and techniques with solar energy powered system, was not discussed before in any previous works. Although these methods were implemented to serve VANET security, it can be slightly modified to be used to protect other systems such as Mobile Ad hoc Network (MANET) and Wireless Sensor Network (WSN). Our future research work will follow different directions in order to fill the gap in this field. We will make use of our experimental network to study the effect of other attacks, and the defense strategies against them, on the VANET performance in all aspects, especially the power consumption of its nodes. The second step is to propose a secure and green Ad hoc routing protocol so that power management and security techniques will be taken into consideration in the earlier design stages.

REFERENCES

[1] T. Wu, W. Liao and C. Chang, , "A Cost-Effective Strategy for Road-Side Unit Placement in Vehicular Networks", IEEE Transactions On Communications, Vol. 60, No. 8, August 2012.

[2] J. Barrachina, P. Garrido, M. Fogue, F. J. Martinez, J. Cano, P. Manzoni, "Road Side Unit Deployment: A Density-Based Approach, IEEE Intelligent transportation systems magazine, 2013.

[3] Q. I. Ali, " Design, Implementation & Optimization of an Energy Harvesting System for VANETS' Road Side Units(RSU)", IET Intelligent Transportation Systems, Vol.8, Issue3, 2014.

[4] Q. I. Ali, " Event Driven Duty Cycling(EDDC): An Efficient Power Management Scheme for a Solar-Energy Harvested Road Side Unit (RSU)", Energy 2015 Conference, Italy, 2015.

[5] Q. I. Ali, A. Fawzi " Design & Implementation of a High Resolution Navigation System for Intelligent Transportation System", Alrafidien Engineering Journal, 2014.

[6] G. Calandriello, P. Papadimitratis, J. P. Hubaux, A. Lioy, On the Performance of Secure Vehicular Communication Systems, IEEE transactions on dependable and secure computing, 2010.

[7] G. Samara, S. Ramadas, and W.A.H. Al-Salihy, Design of Simple and Efficient Revocation List Distribution in Urban Areas for VANET's. International Journal of Computer Science, 2010.

[8] J.J. Haas, Y.C. Hu, and K.P. Laberteaux. Design and analysis of a lightweight certificate revocation mechanism for VANET. in VANET '09: Proceedings of the sixth ACM international workshop on VehiculAr InterNETworking 2009.

[9] D. K. Nilsson, U. E. Larson, E. Jonsson, Low-Cost Key Management for Hierarchical Wireless Vehicle Networks, IEEE Intelligent Vehicles Symposium, 2008.

[10] Y. Hao, Y. Cheng and K. Ren, Distributed key management with protection against RSU compromise in group signature based VANET, IEEE GLOBECOM, 2008.

[11] A. Studer, E. Shi, F. Bai, A. Perrig, Tracking together efficient authentication, revocation and privacy in VANETs, 7th Annual IEEE Communications Society Conference on Sensor Mesh and Ad Hoc Communications and Networks (SECON 09), 2009.

[12] S. Sivagurunathan, P. Subathra, V. Mohan and N. Ramaraj, Authentic vehicular Environment Using a Cluster Based Key Management, European Journal of Scientific Research, vol. 36, no. 2, September, 2009.

[13] Q.I. Ali, "Design & Implementation of an embedded Intrusion Detection System for Wireless Applications", IET Information Security Journal, Vol.6, Issue 3,2012.

[14] M. Erritali, B. El Ouahidi, A Survey on VANET Intrusion Detection Systems, Proceedings of the 2013 International Conference on Systems, Control, Signal Processing and Informatics, 2013.

[15] S. Sharma , M. Sisodia. "Network Intrusion Detection By using Supervised and Unsupervised Machine Learning Techniques: A Survey". International Journal of Computer Technology and Electronics Engineering. 2011.

[16] N. meyer, J. Njeukam, J. Petit, and K. M. Bayarou, "Central misbehavior evaluation for VANETs based on mobility data plausibility," in Proceedings of the ninth ACM international workshop on Vehicular inter-networking, systems, and applications - VANET '12, New York, USA: ACM Press 2012.

[17] J. Grover, V. Laxmi, and M. Gaur, "Misbehavior detection based on ensemble learning in vanet," in Advanced Computing, Networking and Security, ser. Lecture Notes in Computer Science, Eds. Springer Berlin / Heidelberg, vol. 7135, 2012.

[18] S.Chang, Y.Qi, H.Zhu, J.Zhao, and X.Shen, "Footprint: Detecting Sybil Attacks in Urban Vehicular Networks", IEEE Trans. Parallel and Distributed Systems, vol.23, June. 2012.

[19] M.S. Bouassida, G. Guette, M. Shawky, and B. Ducourthial, "Sybil Nodes Detection Based on Received Signal Strength Variations within Vanet,", Int'l J. Network Security, vol. 9, no. 1, 2009.

[20] H. Kaur , S. Batish and A.Kakaria, An approach to detect the wormhole attack in vehicular ad hoc network in: International journal of smart sensors and ad hoc networks,4,2012.

[21] A. Sinha, S.K. Mishra, Preventing VANET From DOS & DDOS Attack, International Journal of Engineering Trends and Technology (IJETT) – Volume 4 Issue 10- Oct 2013.

[22] Q. Wu, J. Domingo-Ferrer, and U. Gon_zalez-Nicola´ s, "Balanced Trustworthiness, Safety and Privacy in Vehicle-to-vehicle Communications," IEEE Trans. Vehicular Technology, vol. 59, no. 2, 2010.

[23] D. Eckhoff, C. Sommer, T. Gansen, R. German, and F. Dressler, "SlotSwap: Strong and Affordable Location Privacy in Intelligent Transportation Systems," IEEE Communications Magazine, vol. 49, no. 11, 2011.

[24] S.K. Dhurandher, M.S. Obaidat, A. Tyagi, "Securing Vehicular Networks: A Reputation and Plausibility Checks-based Approach", IEEE Globecom 2010 Workshop on Web and Pervasive Security, 2010.

[25] X. Xiaoping, N. DING, J. Yiwen, "A trusted neighbor table based location verification for VANET Routing", IET 3rd International Conference on Wireless, Mobile and Multimedia Networks (ICWMMN 2010), 2010.