OPEN ACCESS

International Journal Of Modern Engineering Research (IJMER)

Filtering the Route Dynamically Wireless Sensor Network for Good Data Source and Anamoly Detection

Ms. R.Ramya[1], Prof. P.S.Balamurugan[2]

[1]*(Assistant professor (CSE), Annai mathammal sheela engineering college, Namakkal, Tamilnadu, India)*
[2]*(Research Scholar, Anna University, Coimbatore, Tamilnadu, India)*

ABSTRACT:- The wireless sensor network (WSN) has emerged as a promising technology. In WSNs, sensor nodes are distributed deployed to collect interesting information from the environment. Because of the mission of WSNs, most node-wide as well as network-wide activities are manifested in packet traffic. As a result, packet traffic becomes a good data source for modelling sensor node as well as sensor network behaviors. In this article, the methodology of modelling node and network behavior profiles using packet traffic is exemplified. In addition, a node as well as network anomaly is shown to be detectable by monitoring the evolution of node/network behavior profiles.

I. INTRODUCTION

A wireless Sensor networks are the key to gathering the information needed by smart environments, whether in buildings, utilities, industrial, home, shipboard, transportation systems automation, or elsewhere. Recent terrorist and guerilla warfare countermeasures require distributed networks of sensors that can be deployed using, e.g. aircraft, and have self-organizing capabilities. In such applications, running wires or cabling is usually impractical. A sensor network is required that is fast and easy to install and maintain.

Wireless sensor networks satisfy these requirements. Desirable functions for sensor nodes include: ease of installation, self-identification, self-diagnosis, reliability, time awareness for coordination with other nodes, some software functions and DSP, and standard control protocols and network interfaces [IEEE 1451 Expo, 2001].

There are many sensor manufacturers and many networks on the market today. It is too costly for manufacturers to make special transducers for every network on the market. Different components made by different manufacturers should be compatible. Therefore, in 1993 the IEEE and the National Institute of Standards and Technology (NIST) began work on a standard for Smart Sensor Networks. IEEE 1451, the Standard for Smart Sensor Networks was the result. The objective of this standard is to make it easier for different manufacturers to develop smart sensors and to interface those devices to networks.

I.1.Characteristics

The main characteristics of a WSN include

- Power consumption constrains for nodes using batteries or energy harvesting
- Ability to cope with node failures
- Mobility of nodes
- Dynamic network topology
- Communication failures
- Heterogeneity of nodes
- Scalability to large scale of deployment
- Ability to withstand harsh environmental conditions
- Ease of use
- Unattended operation.

Sensor nodes can be imagined as small computers, extremely basic in terms of their interfaces and their components. They usually consist of a processing unit with limited computational power and limited memory, sensors or MEMS (including specific conditioning circuitry), a communication device (usually radio transceivers or alternatively optical), and a power source usually in the form of a battery. Other possible inclusions are energy harvesting modules, secondary ASICs, and possibly secondary communication devices (e.g. RS-232 or USB).

The base stations are one or more distinguished components of the WSN with much more computational, energy and communication resources. They act as a gateway between sensor nodes and the end user as they typically forward data from the WSN on to a server. Other special components in routing based networks are routers, designed to compute, calculate and distribute the routing tables. Many techniques are used to connect to the outside world including mobile phone networks, satellite phones, radio modems, high power Wi-Fi links etc.

II. PROBLEM SET UP AND NOTATIONS

One type is called **false report injection attacks**, in which adversaries inject into sensor networks the false data reports containing nonexistent events or faked readings from compromised nodes. These attacks not only cause false alarms at the base station, but also drain out the limited energy of forwarding nodes. Also, the adversaries may launch DoS attacks against legitimate repor

In **selective forwarding attacks**, they may selectively drop legitimate reports, while in report disruption attacks; they can intentionally contaminate the authentication information of legitimate reports to make them filtered out by other nodes.

Therefore, it is very important to design a dynamic quarantine scheme to filter these attacks or at least mitigate their impact on wireless sensor networks.

Recent advances in wireless communications and electronics have enabled the development of low-cost, low-power, small, yet reasonably efficient wireless sensor nodes. These tiny sensor nodes, consisting of sensing, data processing, communicating, and power source components, make a new technological vision possible: wireless sensor networks (WSNs).

WSNs combine short-range wireless communication, minimal computation facilities, and some kinds of sensing functions into a new form of network that can be deeply embedded in our physical environment.

They involve deploying a large number of tiny sensor nodes in either hostile or non-hostile environments. The nodes then sense environmental changes and report them to other nodes (usually sink nodes connected to the end user) over flexible network architecture. Figure 1 shows the architecture of a WSN. Because there is no, or only limited, infrastructure, WSNs are usually self-organized.

Based on the vision of WSNs, new types of applications become possible. Possible applications include environmental monitoring such as

"Wildfire and climate change monitoring; structural health monitoring; patient health monitoring; and so on."

Due to the mission of WSNs and the low cost of sensor nodes, individual sensor nodes must forward their sensed data to the base station for final processing. This means most node-wide as well as network-wide activities are manifested in packet traffic. As a result, packet traffic is a good data source for modeling the behaviors of both individual sensor nodes and the whole network.

III. BASICS OF PACKET TRAFFIC MODELING

Packet traffic modeling and classification are found to be important in many areas such as quality of service(QoS) provisioning, traffic analysis, traffic simulation, traffic prediction, and network anomaly detection. The task of traffic modeling is to find statistically invariant properties of packet traffic, which subsequently can be used to identify the types of packet traffic.

The types of different traffic can be differentiated by their associated applications, locations, times, and so on. To model a certain type of packet traffic, traffic features like packet train size and packet train length, inter packet times, and payload size are used to characterize packet traffic.

According to traffic features can be mainly grouped into basic features, time-based features, and connection-based features according to their underlying implementation. Table 1 gives a summary of the frequently used traffic features in packet traffic modeling.

One or a combination of traffic features can be used to statistically model a certain type of packet traffic. In WSNs, packet traffic associated with a sensor node can be used to model the behavior of that specific node; routing traffic can be used to model the behavior of routing protocols; and so on. In the following, examples of modeling the behaviors of sensor nodes and sensor networks using packet traffic are presented.

III.1.Modeling node profile:

In WSNs, sensor nodes cooperate to finish a communication task. Each sensor node functions as a sensing data source as well as a relay for the others. But no sensor node has a full view of network-wide communication. This is partially due to the fact that each sensor node has its unique sets of relay nodes and child nodes. This is also due to the fact that the sensing operation is fully controlled by individual sensor nodes and may be different from sensor node to sensor node.

With this insight, we can uniquely model a sensor node using the unique packet traffic set observed at that specific sensor node.

III.2.Modeling Node Profile Based on Packet Sequence:

Sequence relations exist among some types of packets. For example, a Routing Reply (RREP) message always comes after a Routing Request (RREQ) message, which is specified by a routing protocol. In EARLIER the authors propose to use a finite state machine (FSM) to specify the correct ad hoc on demand vector (AODV) routing behavior. The authors also use an FSM to model the correct routing behavior for another routing protocol.

In addition, the sequence relations among some special types of packets can be modeled according to protocol specifications, and the sequence relations among general kinds of packets can also be learned through training. In our former work, a methodology of automatically learning sequence relations for packets arriving at a sensor node has been presented.

The general idea of this methodology is described in the following. Due to the unique traffic set observed at each sensor node, the learned set of packet sequence relations is unique for each sensor node. Therefore, a sensor node is uniquely modeled by its corresponding packet sequence relations.

Packet Classification: To learn the sequence relations among packets, those packets must first be classified properly. Otherwise, either the class set has an unmanageable size, or the learned sequence relations have no practical use.

We propose to classify the packets in such a way that the whole set of packet categories can be mapped to a set of single-byte ASCII characters, and the sequence relations learned based on the classified packets can reflect the unique behavior of the node of interest.

As a demonstration, we are going to classify packets according to the combinations of the two traffic features Packet Type and {Src, Dest} (i.e. the abbreviation for the pair of source and destination addresses).

In order to control the number of packet categories and make the packet classification scheme scalable, we further map the real node address space to an abstracted address space. The abstracted address space has only five entries: {me; neighbor; local; unlocal; and sink/cluster header}, which are classified from the point of view of the node of interest. In concrete terms, “me” is the node of interest, “neighbor” represents all those nodes within one hop distance of the node of interest, and “local” represents all those nodes that are already known by the node of interest through learning of the source and destination nodes of all its previously observed packets.

During the packet sequence learning, no node is classified as “unlocal.” Once a stable set of all learned packet sequence relationships is acquired, the observation of a packet with its source or destination node classified as “unlocal” is usually a sign of anomaly.

Packet Translation: For simplicity, the classified packets can be further mapped to a set of single-byte ASCII characters. Figure 2 shows the process of packet classification and the process of mapping the classified packets to a character set. Finally, the sequence of packets arriving at a node of interest can be viewed as a large (or asymptotically infinite) string of characters.

Pattern Extraction: To learn the sequence relationships among the arriving packets, we must extract patterns from the large (or asymptotically infinite) string of characters.

The pattern Extraction algorithm first proposed by Forrest et al. in earlier for intrusion detection in a UNIX system is used in this case.

During pattern extraction, the arriving sequence of the abstracted packet events (i.e., the character string) is scanned for all given length, k, unique sub-sequences. Simultaneously, a set of all such unique sub-sequences that have been found (i.e., patterns) is built. Once a stable set of patterns has been constructed, the process of pattern extraction is completed, and a behavior profile for the node under consideration is acquired.

The construction of the pattern set is best illustrated with an example. For k = 4 and the sample sequence AABBDCC, we obtain the following pattern set: AABB, ABBD, BBDC, BDCC. The method of pattern extraction can be more complicated.

If we consider the fact that packets are usually sent based on connections, we can ignore the sequence relations among packets belonging to different connections.

IV. MODELING NETWORK PROFILE

Besides individual sensor nodes, the whole sensor network can also be modeled by identifying the statistical invariants exhibited in network-wide packet traffic

IV.1Monitoring Network-Wide Packet Traffic:

Because communication is wireless inside a WSN, monitoring packet traffic is not a difficult task. A problem may arise when there is a need to monitor all network-wide packet traffic. Due to the lack of infrastructures, monitoring all network- wide packet traffic will require a deployment of monitoring nodes all over the network. This is not always realistic.

Fortunately, there is a slightly downgraded but much easier plan B. In a typical WSN, the task is to collect sensed information from all over the network and forward it to a powerful base station for final processing. As a result, most data packets as well as routing packets are destined to the base station. Therefore, it is possible to monitor most network-wide packet traffic by simply attaching a monitor to the base station.

IV.2.Classification of Network-Wide Packet Traffic:

When there is access to network-wide packet traffic, appropriate classification of the observed packet traffic provides a behavior profile for the network as a whole. We mentioned earlier that selected features can be used to represent one or one group of packets. If the features describing the same (group of) packet(s) is viewed as a data feature vector, we have a set of data feature vectors describing all network-wide packet traffic.

To make a classification of network wide packet traffic, a clustering algorithm can be applied to the set of data feature vectors. By organizing data feature vectors into groups whose members are similar in some way, a clustering algorithm finds the structure of the dataset or of the network. If it is not clear, the following case study gives more intuition.

IV.3. Detecting Anomalies Based On Node/Network Profiles

It has been shown that behavior profiles of individual sensor nodes as well as the whole sensor network can be modeled based on their associated packet traffic features. In this section it is shown that the built profiles can be used as the basis for anomaly detection.

IV.4. Detecting Abnormal Packet Sequences

Above we have shown that the sequence relations among packets arriving at a sensor node can be learned. Actually, the learned set of sequence relations represents the unique behavior profile of the node of interest. Because a sensor node in a WSN has its role assigned, and only performs necessary and specified operations, the number of patterns exhibited in the observed packet sequence is limited. Once a stable pattern set has been built, any unacquainted new pattern or packet subsequence observed at the node of interest is highly suspicious and should signal an anomaly.

V. Pattern Matching and Alarm:

When a node profile consisting of patterns is used for anomaly detection, pattern matching is used to find out whether a new packet sub-sequence is a known pattern or not. Pattern matching is similar to pattern extraction. A buffer window of length *k* is maintained across the sequence of arriving packets during runtime monitoring. Each time a new interesting packet arrives, the buffer window is moved forward by one position and checked for a *match* (i.e., whether there is a pattern that matches the sub-sequence in the buffer window). If no matching pattern exists, this is called a *mismatch*. Let *a* and *b* be two sequences of length *k*. The expression *ai* designates the character at position *i*. The difference *d* (*a*, *b*) between *a* and *b* is defined as

$$d(a,b) = \sum_{i=1}^{k} f_i(a,b),$$

$$\text{where } f_i(a,b) = \begin{cases} 0 & \text{if } a_i = b_i \\ 1 & \text{otherwise} \end{cases}$$

During pattern matching, we determine for each subsequence *u* of the arriving packet sequence the minimum distance *d*min(*u*) between *u* and the entries in the pattern set,

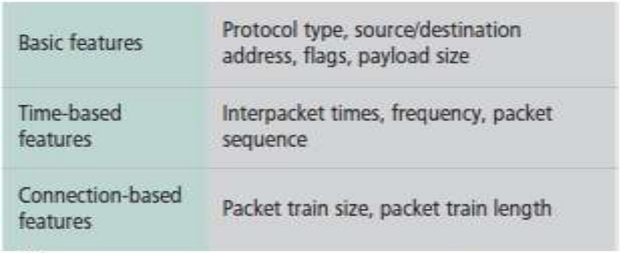
$$d_{\min}(u) = \min \{d(u,p) \quad \forall \text{ patterns } p\}.$$

To detect an anomalous event, at least one of the observed sub- sequences affected by this event must be classified as anomalous. In terms of the above measure, there is at least one sub-sequence *u* for which *d*min(*u*) > 0.

In the ideal case, any *d*min(*u*) value greater than 0 can be considered as a sign of an anomalous event. However, a complete match cannot always be achieved, especially for a network with mobility and a dynamic routing strategy.

Therefore, a threshold can be defined such that only sub-sequences whose *d*min(*u*) value is above this threshold are considered suspicious. Once a packet subsequence is detected as suspicious, an alarm is launched.

When variable-length patterns are used, multiple buffer windows with different lengths should be adopted. The pattern matching method is the same for all pattern lengths. If a packet sub-sequence is considered to be a pattern only after it has occurred for two or more times, the pattern matching should adopt the same strategy. That is, a mismatch of a packet sub-sequence would not be considered anomalous until the packet sub-sequence under consideration has been observed more than two or more times during the pattern matching stage.



Basic features	Protocol type, source/destination address, flags, payload size
Time-based features	Interpacket times, frequency, packet sequence
Connection-based features	Packet train size, packet train length

Table 1. *Traffic features used for packet traffic modeling.*

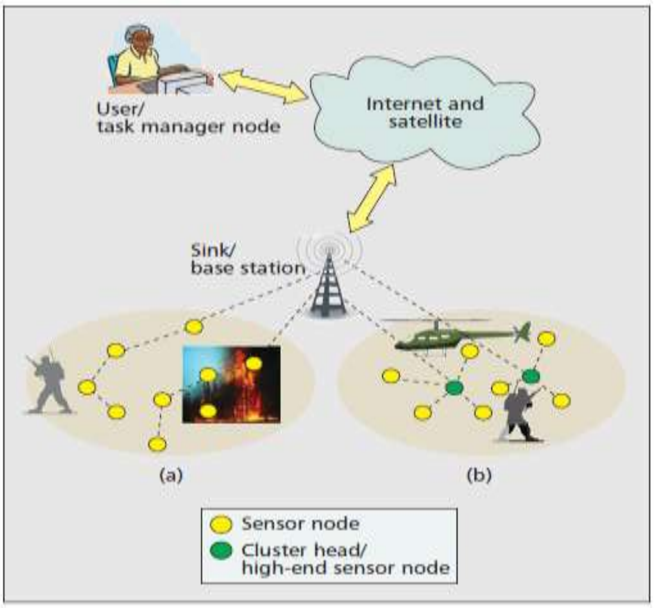
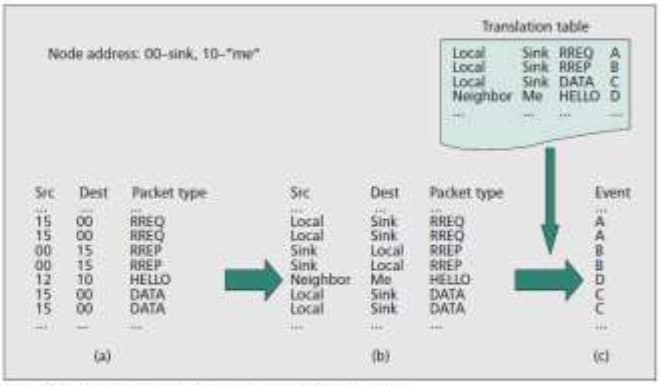


Figure 1. *The architecture of WSNs.*



Node address: 00–sink, 10–"me"

Translation table

Local	Sink	RREQ	A
Local	Sink	RREP	B
Local	Sink	DATA	C
Neighbor	Me	HELLO	D
...	...	...	...

Src	Dest	Packet type	Src	Dest	Packet type	Event
...	...	...	...	...	...	...
15	00	RREQ	Local	Sink	RREQ	A
15	00	RREQ	Local	Sink	RREQ	A
00	15	RREP	Sink	Local	RREP	B
00	15	RREP	Sink	Local	RREP	B
12	10	HELLO	Neighbor	Me	HELLO	D
15	00	DATA	Local	Sink	DATA	C
15	00	DATA	Local	Sink	DATA	C
...	...	...	...	...	...	...
	(a)			(b)		(c)

Figure 2. *Translation of packet arriving events to characters.*

V.1.Detecting Abnormal ON/OFF Periods:

In an event-driven WSN, bursty source traffic due to the discovery of interesting events can originate from any corner of the sensing field. Earlier, an ON/OFF model was used to capture the burst phenomenon of event-driven source traffic. An ON period corresponds to a period of continuous observation of interesting events, while an OFF period is a silent period between two adjacent ON periods when there is no observation of interesting events. The duration of an ON/OFF period is not fixed. However, the probability of observing extremely long ON/OFF periods is low.

When there is an unusually long ON period, it could be due to a compromised sensor node or the result of an energy exhaustion attack. When there is an unusually long OFF period, it could be the result of a link or node failure. Therefore, the appearance of an unusually long ON/OFF period is highly suspicious and should trigger an anomaly alarm to receive special attention. In the following, the methodology of detecting unusually long ON/OFF periods is shown.

V.2.The Methodology of Abnormal ON/OFF Period Detection:

The goal is to detect those unusually long ON/OFF periods so that they can be identified for further analysis. Given that the distributions of the duration of ON/OFF periods can be statistically acquired after training for a certain length of time, a probabilistic upper length limit (e.g., $x \lceil F(x) = 0.99$, where x is the length of an ON/OFF period duration) for the duration of ON/OFF periods can easily be acquired for any node of interest. Our strategy is that an abnormal ON/OFF period is detected whenever there is an unusually long ON/OFF period of duration longer than the specified upper length limit.

We thus describe the new ON/OFF state transition diagram (an old diagram is shown in Fig. 3) for anomaly detection in Fig. 6, where the length of the anomaly ON/OFF timer is set to be a probabilistic upper length limit for any ON/OFF period. For a target tracking sensor network considered in earlier, it has been found that the distributions of the duration of ON/OFF periods have short tails. A short tail of a distribution means that the support ranges of the variable is concentrated in a small region, and there is an extremely low probability that the variable will take a large value. With the short tail property, an anomaly regarding an unusually long ON/OFF period can be quickly detected with high confidence.

V.3.Detecting Changes of Network Profiles

The detection of network profile change is a method of anomaly detection as behavior change of individual nodes or links or other network objects is reflected in an updated network profile. By considering the similarity of the updated and old network profiles, network anomalies are detected. When a network profile is given by an appropriate classification of network objects, the change of network profile can be detected by checking out the similarity between an old classification and an updated classification.

VI. CODING SETUP AND RESULTS

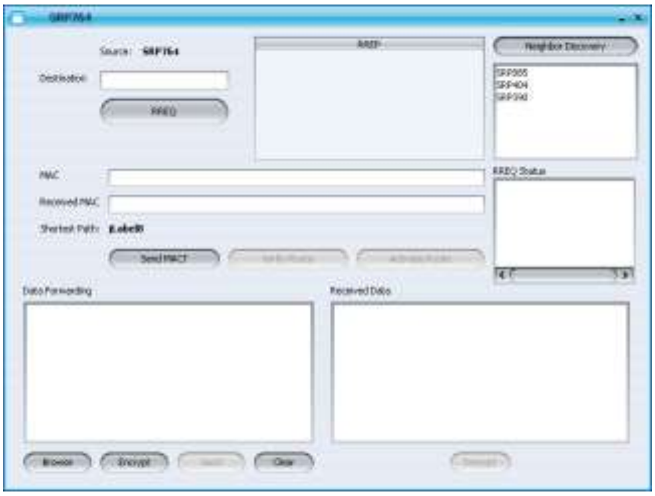
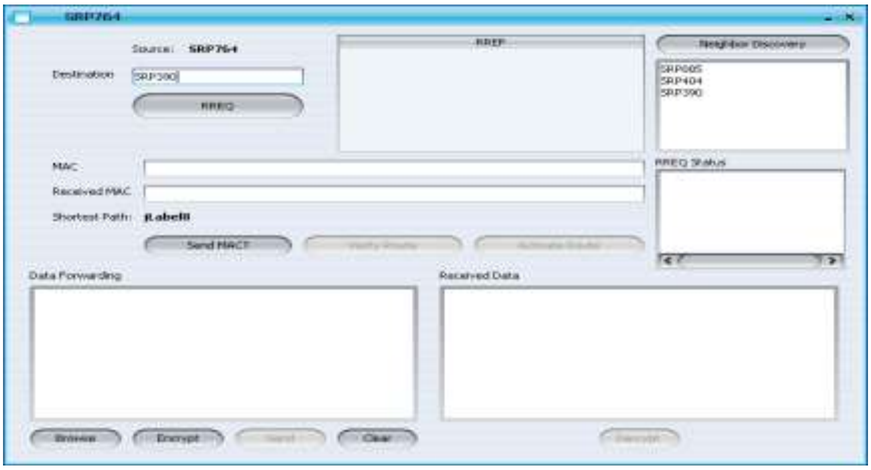


Figure 3: Find The Neighbor Nodes

```
Public class Node extends javax.swing.JFrame {

        /** Creates new form User */
        private static final long serial Version UID = 1L;
        public Action action;
        Receive receive;
        public String source;
        int REQUEST_NUMBER = 0;
        int[] endata;
public Node() {
                initComponents();
                init();}
private void init() {
                // TODO Auto-generated method stub
                action = new Action();
                source = action.getSource();
                setTitle(source);
                lblSource.setText(source);
                int port = action.getPort();
                int dis = action.getDistance();
        action.setProperty("Ports.properties",source, "" + port);
action.setProperty("Distance.properties", source, "" + dis);
                receive = new Receive(this, port, action);
        }
```



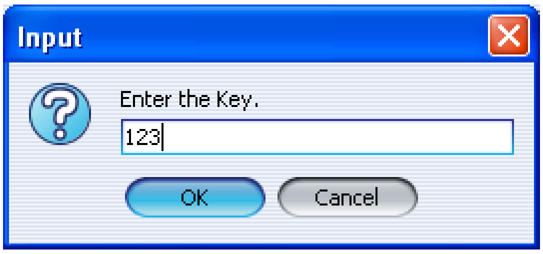
Source: SRP764

Destination SRP390

SRP005
SRP404
SRP390

Figure 4 Request The Destination

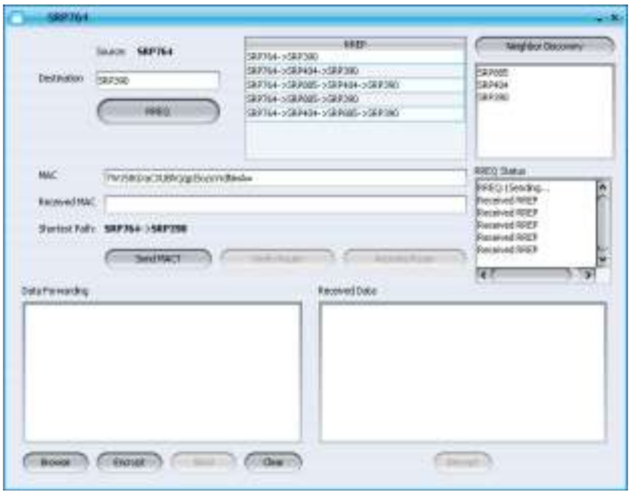
```
lblShortestPath = new javax.swing.JLabel();
btnSendMACT = new javax.swing.JButton();
btnVerify = new javax.swing.JButton();
                btnActivate = new javax.swing.JButton();
                jLabel9 = new javax.swing.JLabel();
                jScrollPane4 = new javax.swing.JScrollPane();
                jtaData = new javax.swing.JTextArea();
                btnBrowse = new javax.swing.JButton();
                btnEncrypt = new javax.swing.JButton();
                btnSend = new javax.swing.JButton();
                btnClear = new javax.swing.JButton();
                jLabel10 = new javax.swing.JLabel();
                jScrollPane5 = new javax.swing.JScrollPane();
                jtaReceive = new javax.swing.JTextArea();
                btnDecrypt = new javax.swing.JButton();
```



Enter the Key.

123

Figure 5 Enter The Key



Source: SRP764

Destination: SRP390

RREP
SRP764->SRP390
SRP764->SRP434->SRP390
SRP764->SRP085->SRP434->SRP390
SRP764->SRP085->SRP390
SRP764->SRP434->SRP085->SRP390

SRP085
SRP434
SRP390

MAC: [illegible]

Received MAC:

Shortest Path: SRP764->SRP390

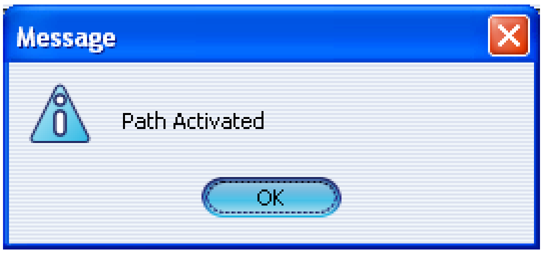
RREQ Status
RREQ-1Sending...
Received RREP
Received RREP
Received RREP
Received RREP
Received RREP

Data Forwarding

Received Data

Figure 6Find Shortest Path

```
btnSendMACT.setText("Send MACT");
                    btnSendMACT.addActionListener(new java.awt.event.ActionListener() {
                              public void actionPerformed(java.awt.event.ActionEvent evt) {
                                        btnSendMACTActionPerformed(evt);
                              }
                    });
```



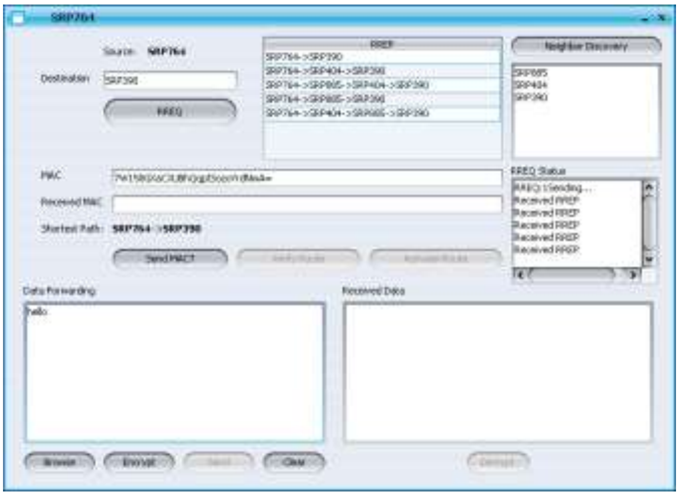
Path Activated

Figure 7 Activate The Path

```
btnVerify.setText("Verify Route");
btnVerify.addActionListener(new java.awt.event.ActionListener() {
publicvoid actionPerformed(java.awt.event.
ActionEvent evt) {
btnVerifyActionPerformed(evt);}});

getContentPane().add(btnEncrypt);
                    btnEncrypt.setBounds(110, 480, 80, 23);

                    btnSend.setText("Send");
                    btnSend.addActionListener(new java.awt.event.ActionListener() {
                              public void actionPerformed(java.awt.event.ActionEvent evt) {
                                        btnSendActionPerformed(evt);
                              }});
```



Source: SRP764

Destination: SRP390

SRP764->SRP390
SRP764->SRP404->SRP390
SRP764->SRP805->SRP404->SRP390
SRP764->SRP805->SRP390
SRP764->SRP404->SRP805->SRP390

SRP805
SRP404
SRP390

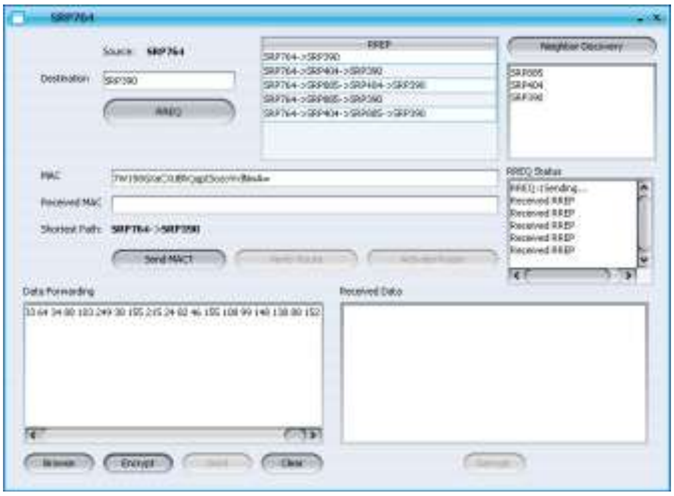
MAC: 7W15lKGlaC3UBfQgjISoozh dMaAw

Shortest Path: SRP764->SRP390

RREQ:1Sending...
Received RREP
Received RREP
Received RREP
Received RREP
Received RREP

hello

Figure 7 Send The Data



Source: SRP764

Destination: SRP390

SRP764->SRP390
SRP764->SRP404->SRP390
SRP764->SRP805->SRP404->SRP390
SRP764->SRP805->SRP390
SRP764->SRP404->SRP805->SRP390

SRP805
SRP404
SRP390

MAC: 7W15lKGlaC3UBfQgjISoozh dMaAw

Shortest Path: SRP764->SRP390

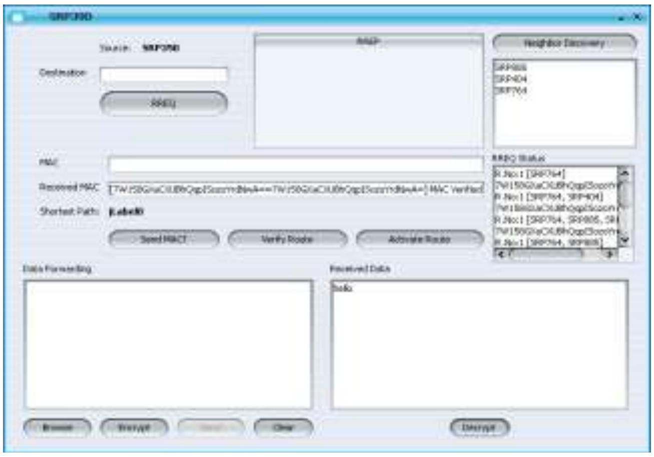
RREQ:1Sending...
Received RREP
Received RREP
Received RREP
Received RREP
Received RREP

33 64 34 80 103 249 30 155 215 24 82 46 155 108 99 140 138 80 152

Figure 8 Encrypt The Data

```
Privatevoid btnDecryptActionPerformed(java.awt.event.ActionEvent evt) {
                        // TODO add your handling code here:
                        receive.decrypt();
            }

PrivatevoidbtnEncryptActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
String str = jtaData.getText();
Encryption encryption = new Encryption();
endata = encryption.ecies_ex(str);
jtaData.setText("");
for (int i = 0; i < endata.length; i++) {
jtaData.append(" " + endata[i]);
}
}
```



Source: SRP350

SRP805
SRP404
SRP764

Received MAC: [TW150GuaCXUBhQgp[Scozm[illegible]==TW150GuaCXUBhQgp[Scozm[illegible]] MAC Verified

Shortest Path: [Label0]

R.No:1 [SRP764]
7W150GuaCXUBhQgp[Scozm
R.No:1 [SRP764, SRP404]
7W150GuaCXUBhQgp[Scozm
R.No:1 [SRP764, SRP805, SR
7W150GuaCXUBhQgp[Scozm
R.No:1 [SRP764, SRP805]

hello

Figure 9 Decrypt the Message

- Compared with others, our scheme can drop false reports much earlier even with a smaller size of memory.
- The uncompromised nodes will not be impersonated because each node has its own auth-keys. Therefore, once the compromised nodes are detected, the infected clusters can be easily quarantined.
- Our Hill Climbing key dissemination approach increases filtering capacity greatly and balances the memory requirement among nodes.
- Each node has multiple downstream nodes that possess the necessary key information and are capable of filtering false reports. This not only makes our scheme adaptive to highly dynamic networks, but also mitigates the impact of selective forwarding attacks.
- Monitored by its upstream nodes and neighbors, the compromised nodes have no way to contaminate legitimate reports or generate false control messages.

VII. CONCLUSION

The packet traffic patterns of WSNs are much simpler and less dynamic than those of more traditional networks (e.g., the Internet). This makes it possible to build precise behavior profiles for individual sensor nodes as well as for the whole WSN based on observed packet traffic. The methodology of building node/network profiles in a WSN based on packet traffic is presented with examples. It is shown that many packet traffic features can be extracted for profile building purposes. Packet arriving sequence, packet arriving interval, and packet arriving frequency are the ones used in this article.

Node/network profiles based on different packet traffic features and packet traffic types reflect different aspects of node/network behaviors. Because WSNs are basically simple networks in terms of node/network behaviors, node/network profiles evolve slowly over time.

Once there is a sign of a change in node/network profiles, there is a high risk that something unexpected is happening. Based on this rationale, it is proposed that anomalies due to malicious threats and reliability defects can be detected by monitoring changes appearing in node/network profiles.

ACKNOWLEDGEMENT

The authors of this paper would thank the reviewers for their valuable comments and suggestions, which help us to improve the quality of this paper.

REFERENCES

[1]. Q. Wang and T. Zhang, "A Survey on Security in Wireless Sensor Networks," Ch. 14, Security in RFID and Sensor Networks, Y. Zhang and P. Kitsos, Eds. CRC Press, Taylor & Francis Group, 2009, pp. 293–320.

[2]. I.-V. Onut and A. A. Ghorbani, "Features vs. Attacks: A Comprehensive Feature Selection Model for Network Based Intrusion Detection Systems," Ch. 2, LNCS, ser. 4779, J. G. et al., Eds., Springer-Verlag, 2007, pp. 19–36.

[3]. C. Tseng et al., "A Specification-Based Intrusion Detection System for AODV," Proc. 1st ACM Wksp. Security of Ad Hoc and Sensor Networks, 2003.

[4]. P. Yi et al., "Distributed Intrusion Detection for Mobile Ad Hoc Networks," Proc. 2005 Symp. Apps. and the Internet Wksps., 2005.

[5]. Q. Wang and T. Zhang, "Detecting Anomaly Node Behavior in Wireless Sensor Networks," Proc. 21st Int'l. Conf. Advanced Info. Networking and Apps. Wksps., May 2007, pp. 451–56.

[6]. S. Forrest et al., “A Sense of Self for Unix Processes,” Proc. 1996 IEEE Symp. Security and Privacy, May 1996, pp. 120–28.

[7]. S. Hofmeyr, S. Forrest, and A. Somayaji, “Intrusion Detection Using Sequences of System Calls,” J. Comp. Security, vol. 6, 1998, pp. 151–80.

[8]. S. Tang, “An Analytical Traffic Flow Model for Cluster-Based Wireless Sensor Networks,” Proc. 1st Int’l. Symp. Wireless Pervasive Computing, 2006.

[9]. Q. Wang and T. Zhang, “Source Traffic Modeling in Wireless Sensor Networks for Target Tracking,” Proc. 5th ACM Int’l. Symp. Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks , Oct. 2008, pp. 96–100.

[10]. X. Zhao et al., “ON/OFF Model: A New Tool to Understand BGP Update Burst,” tech. rep. 04-819, USC-CSD, Aug. 2004.