2012 Fourth International Conference on Computational Intelligence and Communication Networks

Wireless Sensor Network Security: Real-Time Detection and Prevention of Attacks

Maneesha V. Ramesh, Aswathy B. Raj and Hemalatha T.
Amrita Center for Wireless Networks and Applications
Amrita Vishwa Vidyapeetham, Kerala, India
maneesha@am.amrita.edu, aswathy.braj13@gmail.com, hemalatha@am.amrita.edu

Abstract— **Security is a critical issue in many real-world applications of wireless sensor networks (WSNs). This research focuses on implementing security mechanisms, against two specific types of attack that occur in a network of MICAz motes. These are 'denial-of-service' (DoS) attacks where unnecessary packets are sent causing services to appear unavailable and thus these services are denied to the legitimate sensor nodes; and 'passive information gathering' attacks where an adversary tries to obtain the confidential information stored in wireless sensor nodes. A machine learning technique named neural network (NN) is used to detect DoS attack conducted by an adversary. We compare and evaluate NN with our new method. The results clearly show an improved performance for our new proposed symmetric-key algorithm compared to NN. Lower encryption computational energy cost (8 µJ vs 16 µJ); lower memory requirements (8896 bytes of ROM and 434 bytes of RAM vs 15848 bytes of ROM and 763 bytes of RAM); and less execution time (0.164835 ms vs 0.6208791 ms) show the significant advantages of using symmetric-key algorithm instead of NN for detecting DoS attack.**

Keywords-Denial of Service; Security; Wireless Sensor Network; Symmetric cryptography; TinyOS.

I. INTRODUCTION

A WSN consists of several individual nodes that are capable of sensing physical parameters. Nowadays, WSNs are widely used for developing real-time monitoring systems. In certain applications, ensuring the security of WSNs is essential for protecting confidential data and ensuring data accuracy. WSNs are vulnerable to a variety of security attacks. The security goals, such as: data confidentiality, data integrity, data availability, data authentication, data freshness, self-organization, time synchronization and secure localization are considered while implementing security mechanisms in WSN [1]. Due to WSN constraints, existing security approaches for computer networks cannot be applied to WSNs. It is necessary to understand WSN constraints before implementing any WSN security mechanism. Resource limitation, unreliable communication and remote location are the major constraints exhibited by WSN [2]. A wireless sensor node has a limited amount of memory for storing programs and data. Therefore the memory space required for a security algorithm must be small. Energy is a major constraint associated with WSN. Sensor nodes are powered through either batteries or solar power. While adding security to a sensor node, the additional operating/energy cost involved for implementing security approach has to be considered. The WSN may be left unattended for a long period of time. Its remote management makes it difficult to detect attacks launched by an adversary.

A DoS attack is an attempt by an adversary to degrade the network's services. In DoS attacks, malicious nodes can degrade the services provided by legitimate nodes, by flooding the legitimate nodes with requests (RTS). One of the characteristics of WSNs is that they are based on 'carrier sense multiple access with collision avoidance' (CSMA/CA) mechanism. This CSMA/CA mechanism relies on the exchange of ready-to-send (RTS) and clear-to-send (CTS) control packets. When a source node has data to send, it initiates the process by sending an RTS packet. When an RTS packet is heard by any node, the node will respond by sending a CTS packet. Therefore when an RTS is travelling the WSN it silences all passing nodes until it reaches its target node, and thus only one CTS packet is returned. Like the RTS packet, the CTS packet silences the nodes in its immediate vicinity. Once the RTS/CTS exchange is complete, the source node transmits data without worry of interference from any other nodes. The data packets are positively acknowledged. In passive information gathering, adversary can use data from multiple sensor nodes to derive sensitive information. This paper introduces a new symmetric-key algorithm capable of detecting both 'denial-of-service' (DoS) and 'passive information gathering' attacks.

The rest of the paper is organized as follows: A survey of different security mechanisms against attacks in WSN is made in section II. The details of real-time detection and prevention of DoS and passive information gathering attacks are explained in sections III and IV. The implementation results, showing the WSN security enhancement against the two attacks, are presented in section V. Finally conclusion and future work are in section VI.

II. RELATED WORK

There is a high demand for adequate WSN security development in both research and commercial applications. The WSN security challenges for collecting and processing data are described by Dirk et.al [3]. Various types of threats and attacks against WSN are categorized by John Paul Walters et.al [4]. Four currently used mechanisms to overcome DoS attacks in WSNs include watchdog scheme, rating scheme, virtual currency and securing routing layer. Afrand Agah and Sajal K. Das [5] identified the disadvantages of these mechanisms. They formulated the prevention of DoS attacks in WSNs as a repeated game

978-0-7695-4850-0/12 $26.00 © 2012 IEEE
DOI 10.1109/CICN.2012.209

between an intrusion detector and nodes of a sensor network. In their proposed framework, the intrusion detector, residing at the base station, keeps track of each node's collaboration by monitoring. If the performance of the nodes is found to be lower than certain thresholds, it means that some nodes are acting maliciously by deviation. These error correcting codes provide a flexible mechanism for identifying malicious collisions but they incur additional processing and communication overheads. In this paper we introduce our WSN security solution, which does not require such large processing and communication overheads. Limited protection against DoS attacks was introduced by Ronald Watro et.al[6] using public key based protocol which allowed the WSN authentication. Symmetric cryptography was chosen instead of the previously used asymmetric cryptography security, which depends on the difficulty of the mathematical problem involved in the algorithm, as it consumes considerably more energy than symmetric key cryptography algorithms. Hence in WSNs, the symmetric key algorithm is typically utilized to encrypt data during the transmission of sensor data, conforming to the limited energy source in the sensor device [7]. This paper uses NN for detecting and counteracting to the DoS attacks launched by adversaries on the medium access control (MAC) layer of WSN. The attack definitions are learned by the sensor nodes and hence on detecting attacks they will send alarm packets to base station. A symmetric-key algorithm is developed for securing confidential data of sensor nodes. Moreover, this algorithm can also be used for detecting DoS attacks.

III. NEURAL NETWORK BASED DETECTION

Article [8] shows that DoS attacks can be detected using the following critical parameters.

- R_c (Collision Rate): R_c is the number of collisions detected by a node in a second.
- R_r (RTS arrival rate): R_r is the number of RTS packets received successfully by a node in a second.
- T_w (Average waiting time): T_w is the waiting time of a packet in MAC buffer before transmission.

A WSN scenario consisting of 25 wireless sensor nodes is simulated using the probabilistic wireless network simulator (Prowler). The above critical parameters at different probability of DoS attack, ranging from 0.1 to 1, are computed for 50 trial runs. From the simulation, it is observed that T_w is negligible compared to R_c and R_r. Hence in this NN based detection scheme, R_c and R_r are used for determining the suspicion of DoS attack.

In the NN based approach, the parameters *Rc* and *Rr* are represented as inputs and the corresponding probability of attack is represented as the targets to the multilayer perceptron (MLP). The MLP is trained by using backpropagation algorithm. At each MICAz mote, a new MLP is implemented with predefined weights and biases which are obtained from trained MLP. Every second, each mote passes its computed values of R_c and R_r to its MLP which produces an output (that is the calculated probability of attack at that particular mote). If the MLP's output (that is the calculated probability of attack at that particular mote) is greater than a preset threshold value S_{TH}, then the mote sends an alarm packet to the base station.

A. *Training Algorithm*

MLP is a feed forward NN in which neurons are arranged in many layers. First layer consists of input units and last layer consists of output units. All other units are called hidden units and they constitute hidden layer. Each neuron is connected to other neurons by directed communication links. Each communication link has a weight associated with it. The weights represent information in the neural net. Each neuron has a state called activation which is a function of all inputs it has received. The structure of MLP used is shown in Figure 1. It has one hidden layer. X_1 and X_2 are the input units. The output unit Y_1 and hidden unit Z_1 have biases. The bias on output unit Y_1 is denoted by W_{01}. The bias on hidden unit Z_1 is denoted by V_{01}. The activation function applied to both hidden layer and output layer is sigmoid function.

MLP is trained using BP algorithm [9]. It involves three stages namely feed forward of the input pattern, calculation and backpropagation of associated error, adjustment of weights. All the weights and biases of the backpropagation neural network are initialized to random values between 1 and -1. The activation of the i^{th} hidden unit for a given input pattern k is described in equation (1), where x_{kh} is the output of the input layer, v_{ih} is the input to output connection and $f_{sig}(.)$ is the sigmoid function.

$$z_{ki} = f_{sig}\left(\sum_{h=0}^{2} x_{kh} v_{ih}\right) \quad (1)$$

The activation of the j^{th} output unit is described in equation (2), where z_{ki} is the output of the hidden unit and w_{ji} is the hidden to output connection.

$$y_{kj} = f_{sig}\left(\sum_{i=0}^{h} z_{ki} w_{ji}\right) \quad (2)$$

The error signal δ_{kj} for output unit is defined in equation (3).

$$\delta_{kj} = y_{kj}\left(1 - y_{kj}\right)\left(b_{kj} - y_{kj}\right) \quad (3)$$

This error value is propagated back to update weights that feed the output layer is described in equation (4).

$$w_{ji}^{new} = w_{ji}^{old} + \eta \sum_k \delta_{kj} z_{ki} + \alpha \Delta w_{ji}^{old} \quad (4)$$

δ is the learning coefficient and α is the momentum factor. Here δ= 0.075 and α=0.15.

The error signal δ_{ki} for the hidden unit is described by equation (5).

$$\delta_{ki} = z_{ki}(1 - z_{ki}) \sum_{j=1}^{q} w_{ji} \delta_{kj} \qquad (5)$$

This error value is propagated back to update weights that feed the hidden layer from the input layer is described in equation (6).

$$v_{ih}^{new} = v_{ih}^{old} + \eta \sum_k \delta_{ki} x_{kh} + \alpha \Delta v_{ih}^{old} \qquad (6)$$

In this off-line learning mode of BP network, the weights are updated once after all patterns in the input pattern set are presented. It is carried out for 10000 epochs. After training, the weights of the trained network are obtained. Theses weights are used to implement a new MLP in MICAz mote.

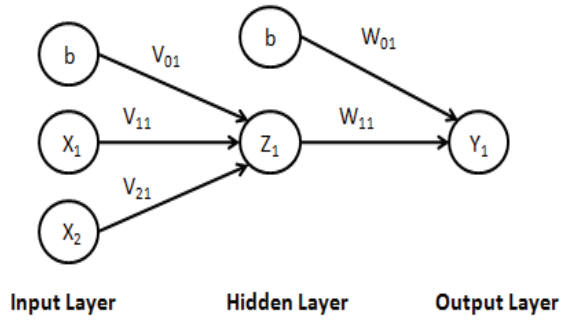

Figure 1. Structure of MLP

IV. SECURITY BASED ON SYMMETRIC CRYPTOGRAPHIC ALGORITHM

In symmetric key cryptography, a single key is used for both encryption and decryption. The sender uses the key to encrypt the plaintext and sends the ciphertext to the receiver. The receiver applies the same key to decrypt the message and recover the plaintext. With this form of cryptography, the key must be known to both the sender and the receiver. All the motes in the WSN will be embedded with a single key k, which is 8-bit. All the motes encrypt the data for transmission by using the Algorithm 1. After receiving the data by a mote, it will decrypt it using the Algorithm 2. After decryption, each mote will perform the DoS attack detection using Algorithm 3. The mote will accept the packet if it is coming from the legitimate mote. Otherwise, the mote will drop the packet. Since the adversary is unaware of the symmetric cryptography used in the WSN, XOR operation in Algorithm 3 always gives a value other than k. Each mote keeps track of the number of packets dropped in this manner and stores it in a variable R_d. Every second, each mote checks the value of R_d. If R_d is greater than a preset threshold C_{TH}, then it generates an alarm packet and transmits it to the base station indicating the occurrence of DoS attack.

Algorithm 1 Encryption

Require: Single shared key, k
1: Generate 8-bit value of plain text, p
2: Take one's complement of the binary number generated
3: Divide the complemented number with $(00001010)_2$ to obtain quotient q and remainder r
4: Left shift the remainder by 5 bits, i.e. r = r << 5
5: Compute s = q ∨ r
6: Cipher text, c = s ⊗ k

Algorithm 2 Decryption

Require: Single shared key, k
1: Generate 8-bit value of cipher text, c
2: c is XORed with k, i.e. c = c ⊗ k
3: Compute q = c ∧ $(00011111)_2$ and r = c ∧ $(11100000)_2$
4: Right shift r by 5 bits, i.e. r = r >> 5
5: Compute s = (q × $(00001010)_2$) + r
6: Take one's complement of s to obtain plain text, p

Algorithm 3 DoS Attack Detection

Require: Single shared key, k
1: Rd ← 0
2: Decrypt cipher text, c to obtain plain text, p
3: Generate 8-bit value of p
4: Take one's complement of p
5: Divide the complemented number with $(00001010)_2$ to obtain quotient q and remainder r
6: Left shift the remainder by 5 bits, i.e. r = r ≪ 5
7: Compute s = q ∨ r
8: **if** s ⊗ c = k **then**
9: Accept the packet
10: **else**
11: Drop the packet
12: Rd + +
13: **if** Rd ≥ CTH **then**
14: Send alarm packet to base station
15: **end if**
16: **end if**

V. EXPERIMENTATION AND RESULTS

A. *Numerical Results*

The WSN scenario used for landslide detection [10] is modeled for DoS attack detection. The topology in Figure 2 is used for performing DoS attack on real nodes. Here 'circle' represents the real wireless sensor nodes and 'edge' represents the communication between the sensor nodes. The WSN follows a two-layer hierarchy; with lower wireless sensor nodes transmit data packets to intermediate nodes. The intermediate nodes aggregate data packets and send them to sink node. The sink node aggregates the data packets from intermediate nodes.

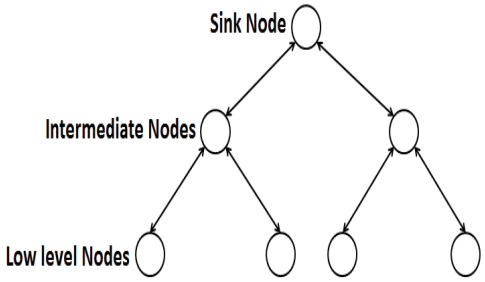

Figure 2. WSN topology

The real wireless sensor nodes used for the implementation are MICAz motes from Crossbow. TinyOS is used as the operating system for WSN. It is a flexible, event-driven and application specific operating system with low memory requirement of 400 bytes. It is implemented in NesC language. A TinyOS program is a set of components, each of which is an independent computational entity that exhibits one or more interfaces. Components have three abstractions namely commands, events, and tasks. Commands and events are used for inter-component communication whereas tasks are used for intra-component communication. A command is a request to a component to perform a service.

The WSN scenario as shown in Figure 2 is implemented using MICAz motes. The implementation of this scenario involves 8 MICAz motes having unique IDs from 1 through 8. If a MICAz mote has data packet to be sent to another Mote, then it sends RTS packet to destination mote. The destination mote responds to the request by sending CTS packet. On receiving CTS packet, data packet is sent by the recipient. The data packet is in turn acknowledged by sending ACK packet. The mote with ID 8 turns into an adversary and launches a DoS attack. It transmits repeated RTS packets with probability of unity in every 0.25 seconds. This transmission collides with the packets on the broadcast medium causing a substantial rise in the collision rate, R_c. Moreover all motes repeatedly send its data packets in response to the requests by adversary. This results in an unusual rise in RTS arrival rate, R_r. The values of the critical parameters R_r and R_c in presence of DoS attack are recorded in 50 trial runs and their average are computed. Table 1 shows the values of the critical parameters in absence of DoS attack.

TABLE I. CRITICAL PARAMETERS AVERAGED OVER 50 TRIAL RUNS IN PRESENCE OF AN ATTACK

Probability of attack	Collision rate	RTS arrival rate
0.1	125.28571	123.2857
0.2	126.28571	127
0.3	128.14286	132.8571
0.4	130	134
0.5	131	135.4286
0.6	132.85714	141.4286
0.7	163.42857	142.4286
0.8	184.57143	144
0.9	186.71429	164.1429
1	206.71429	183.4286

10 set of input patterns (normalized R_r and R_c) as shown in Table I are normalized and presented to the backpropagation neural network. Target value of each pattern set corresponds to their value of probability of attack. After training, the weights of the trained network are obtained. Theses weights are used to implement a new MLP in MICAz mote as shown in Figure 3. This pre-trained MLP in each mote watches the critical parameters collision rate R_c and packet request rate R_r to compute the measure of suspicion (probability of DoS attack).

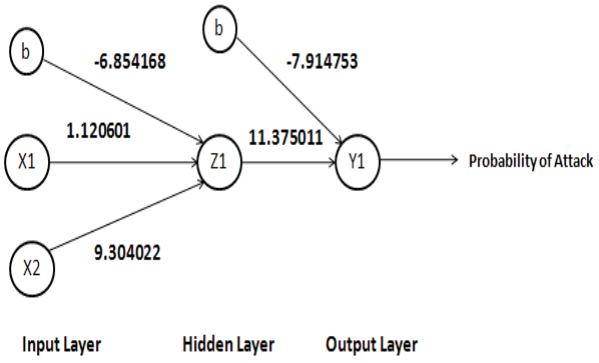

Figure 3. Pre-trained MLP in MICAz mote

B. Performance Analysis

Table II gives the comparison between proposed symmetric cryptographic algorithm and NN. The proposed algorithm has low encryption computational energy cost compared to NN, 16 µJ by NN and 8 µJ by proposed algorithm. Also, the proposed algorithm has low memory requirements, 8896 bytes of ROM and 434 bytes of RAM by the proposed algorithm and 15848 bytes of ROM and 763 bytes of RAM by NN. The results also show that the proposed algorithm needs less execution time, 0.164835 ms by the algorithm and 0.6208791 ms by NN. Hence the proposed algorithm suits best for the resource-constrained sensor nodes.

TABLE II. COMPARISON BETWEEN PROPOSED SYMMETRIC CRYPTOGRAPHIC ALGORITHM AND NN

Performance Metric	Proposed Algorithm	NN
Computational Energy for Encryption (micro joules)	8	16
Memory (bytes)	8896 of ROM 434 of RAM	15848 of ROM 763 of RAM
Execution Time (milli seconds)	0.164835	0.6208791

VI. CONCLUSION AND FUTURE WORK

Each MICAz mote in the network contains a pre-trained MLP which watches the critical parameters collision rate R_c and packet request rate R_r, and computes a measure of suspicion. If the suspicion factor exceeds a preset threshold level, then the mote sends an alarm packet to the base station. A symmetric cryptographic algorithm is developed for securing data from passive information gathering. This method can also be used for DoS attack detection. All the MICAz motes in the network are embedded with a single key, K. Each mote will accept the data packet only if it is from legitimate node; otherwise it will drop the packet. It keeps the count of packets dropped in such manner in a variable, R_d. If R_d exceeds a preset threshold level, then the mote sends an alarm packet to the base station. This method has an advantage that it secures the critical data transmitted over the channel. We compare and evaluate NN with our new method. Lower encryption computational energy cost (8 µJ vs 16 µJ); lower memory requirements (8896 bytes of ROM and 434 bytes of RAM vs 15848 bytes of ROM and 763 bytes of RAM); and less execution time (0.164835 ms vs 0.6208791 ms) show the significant advantages of using symmetric-key algorithm instead of NN for detecting DoS attack. The results clearly show an improved performance for our new proposed symmetric-key algorithm compared to NN. In future, this work will be extended in many ways. One way is to enhance the countermeasure used when DoS attack is detected. This is done by remotely switch off the motes when attack is detected. Another way is to use unsupervised learning or different neural architectures such as generalized neuron, radial basis networks.

ACKNOWLEDGMENT

We would like to express our immense gratitude to our beloved Chancellor Shri. Mata Amritanandamayi Devi for providing a very good motivation and inspiration for doing this research work.

REFERENCES

[1] G. Padmavathi and D. Shanmugapriya, "A survey of attacks, security mechanisms and challenges in wireless sensor networks," *International Journal of Computer Science and Information Security*, vol. 4, 2009.

[2] J. P. Walters, Z. Liang, W. Shi, and V. Chaudhary, "Wireless sensor network security: A survey," Security in Distributed, Grid, and Pervasive Computing, 2006.

[3] Dirk Westhoff, Joao Girao and Amardeo Sarma, "Security Solutions for Wireless Sensor Networks", *NEC Technical Journal*, vol. 1, no. 3, 2006.

[4] John Paul Walters, Zhengqiang Liang, Weisong Shi, and Vipin Chaudhary, "Wireless Sensor Network Security: A Survey", *Security in Distributed, Grid, and Pervasive Computing*, 2006.

[5] A. Agah and S. K. Das, "Preventing dos attacks in wireless sensor networks: A repeated game theory approach," *International Journal of Network Security*, vol. 5, no. 2, pp. 145-153, September 2007.

[6] R. Watro, D. Kong, S. fen Cuti, C. Gardiner, C. Lynn, and P. Kruus, "Tinypk: Securing sensor networks with public key technology," *Proceedings of the 2nd ACM Workshop on Security of ad hoc and sensor networks*, pp. 59–64, October 2004.

[7] X. Zhang, H. M. Heys, and C. Li, "Energy efficiency of symmetric key cryptographic algorithms in wireless sensor networks," *Proceedings of Biennial Symposium on Communications*, May 2010.

[8] Q. Ren and Q. Liang, "Secure media access control in wireless sensor networks: intrusion detections and countermeasures," in *15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*,vol. 4, 2004, pp. 3025 – 3029.

[9] Russell C. Eberhart and Yuhui Shi, "*Computational Intelligence : concepts to implementations*" , Elsevier Inc., 2007.

[10] Maneesha V. Ramesh, "Real-time Wireless Sensor Network for Landslide Detection", 2009 Third International Conference on Sensor Technologies and Application