International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 04 | Apr -2017 www.irjet.net p-ISSN: 2395-0072

DETRACT THE EFFECT OF POWER EXHAUSTING ATTACKS IN HIERARCHICAL WIRELESS SENSOR NETWORKS

Deepa K P

Department of Information Science and Engineering,
THE NATIONAL INSTITUTE OF ENGINEERING,
Mysuru 570008, Karnataka, India

Abstract – *This paper proposes the detailed description and working of Receiver Initiated Media Access Control Protocol (RI-MAC) to alleviate the effect of one of the special type of Denial Of Service Attack called Denial Of Sleep (DOS) attack which drains the battery power of the sensors in the Hierarchical Wireless Sensor Networks (WSN).This paper also addresses a scheme for authenticating the new nodes which try to change the sleep schedule of the nodes. Only transmissions from valid nodes are accepted. The protocols are designed in such a way that they reduce the energy consumption of sensor nodes by keeping the antenna in sleep mode as much as possible. This leads to power saving.*

Key Words **: Sensor, Denial of Service, Denial of Sleep(DOS), Receiver Initiated Media Access Control (MAC) Protocol, Hierarchical WSN.**

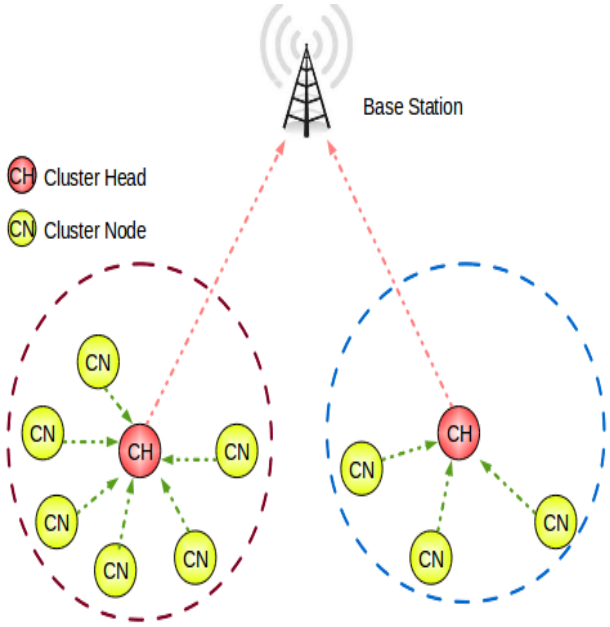
INTRODUCTION

[1] The Denial-of-Sleep is one of the power exhausting attacks of WSNs . This attack is a special type of Denial-of-Service (DoS) attack, which tries to keep the sensor nodes awake to consume more energy of the constrained power supply. An anti-node can send fake data packets to sensor node of unprotected WSNs to initiate unnecessary transmissions repeatedly. Without security mechanism, an anti-node can broadcast a fake preamble frequently in the sender-initiated schemes. If the receiver cannot find the difference between the real preamble and the fake one, the receiver will receive and process the data from the anti-node. Such attack will keep the receiver awake as long as the data transmission sustains, which exhausts the battery of nodes rapidly. Moreover, an anti-node can replay a fake preamble ACK to the sender. Thus, the sender will start to send the data to the anti-node but it will never receive the right data ACK. Similarly, the sender may send data repeatedly and exhausts the battery of node rapidly. In receiver-initiated schemes, an anti-node can broadcast a "fake beacon" to cheat sender to process and send back the data to the anti-node but it will never receive the right data ACK. An anti-node can replay a "fake beacon ACK" to the receiver. Thus, the receiver will start to receive and process the data from the anti-node. If the interval of attack packets is shorter than the sleep period of WSN, then the communication between neighbouring nodes in a WSN could be interfered by attack packets. Consequently, no packets from the attacked nodes can be delivered, which causes a jamming-like scenario. However, unlike the physical jamming attack, no consecutive signals or packets are needed for the packet attack. A well-designed periodical attack packet can be applied to perform such jamming-like attack, which may degrade the performance of a duty-cycle scheme for WSN operating and achieve energy conservation of an anti-node during the attack. As a result, the sender and receiver need mutual authentication schemes to counter such an attack.

In conventional wireless security mechanisms, the transmitted data is encrypted with keyed symmetric or asymmetric encryption algorithm. The wireless sensor networks prefer the symmetric algorithm to avoid the complicated computing and heavy energy consumption. But the encrypted data makes the battery exhaustion even worse under Denial-of-Sleep attack. The anti-node can send the encrypted "garbage" data to receiver. This attack forces the receiver to decrypt the data. Before the receiver identifies that the data is "garbage", the receiver consumes more power to receive and decrypt data. These processes also keep sensor nodes awake longer. Accordingly, an easy and fast mutual authentication scheme is needed to integrate with MAC protocol to counter the Denial-of-Sleep attack.

SYSTEM ARCHITECTURE

The sensors are arranged in hierarchical manner which contains the Sensor Nodes (SN),Cluster Head(CH) and the Base Station(BS).whenever there is some data to be sent from sensor nodes to the base station ,it will be sent though the cluster head which conserves battery power and also enhances the performance of the WSN.



DESIGN PHASE

The Secure Adaptive Topology Control Algorithm (SATCA) is involved to from the hierarchical topology in five phases.

1. Antinode Detection
2. Cluster Formation
3. Key Distribution
4. Data Requisition
5. Message Authentication and Forwarding

Phase 1: Antinode Detection

In order to make the network robust against attacks, an authenticated broadcasting mechanism, a plaintext "Hello" message is encrypted by the pre-distributed key as the broadcasting challenge. If the sensor cannot decrypt the received message successfully, the sender is said to be an anti-node. Thus, the normal nodes and the anti-nodes can be differentiated. Therefore, we keep on the network topology without anti-nodes in order to make the network safe.
Notice that an external attack can be prevented by the operation of Phase I. In this work, we do not have a light weight countermeasure to defend against authenticated malicious nodes. If the authenticated node is compromised and performs malicious activities, a mechanism for evicting the compromised nodes is required.

Phase 2: Cluster Formation

When sensors are first deployed, the [2] Adaptive Distributed Topology Control Algorithm (ADTCA) may be used to partition the sensors into clusters and for the cluster head selection. Each sensor sets a random waiting timer, broadcasts its presence via a "*Hello*" signal, and listens for its neighbour's "*Hello*." The sensors that hear many neighbours are good candidates for initiating new clusters, those with few neighbours should choose to wait. Sensors update their neighbour information (i.e., a counter specifying how many neighbours it has detected) and decrease the random waiting time based on each "new" *Hello* message received. This encourages those sensors with many neighbours to become cluster heads. If a neighbour declares itself to be a cluster head, the sensor cancels its own timer and joins the neighbour's new cluster. If the timer expires, then the sensor declares itself to be a cluster head, a focal point of a new cluster. By adjusting randomized waiting timers, the sensors can coordinate themselves into sensible clusters, which can then be used as a basis for further communication and data processing.

Phase 3: Key Distribution

3.1: Cluster Registration

In this phase the cluster head will register with the base station. The registration is done by sending the packet which contains the IP addresses of all the sensors in that particular cluster and encrypted with the pre distributed key If there are one or more sensor node have the same number of hello count then those sensor node which notify the base station fist will be selected as cluster head and registered.

3.2: Cluster key creation

The cluster key will be created by the base station. Cluster key is mainly used for securing locally broadcast messages. The generation of the cluster key is done by sorting all the IP addresses of the sensor (IP1,IP2……..IPn). The hash function will be applied to this sorted IP addresses the generated key is distributed as a cluster key among the sensor in that cluster.

Variable Clus_key = sort(IP1,IP2….IPn)

Cluster_key = hash(Clus_key)

Phase 4: Data Requisition

The request for the data transmission is done by using [3] RI MAC Protocol in which a hash-chain is created by using the cluster key Kc, which is the shared secret between the valid members and the cluster head. This hash-chain is used for mutual authentication and symmetric encryption key. The session key agreement the receiver-initiated scheme are shown in Fig 1

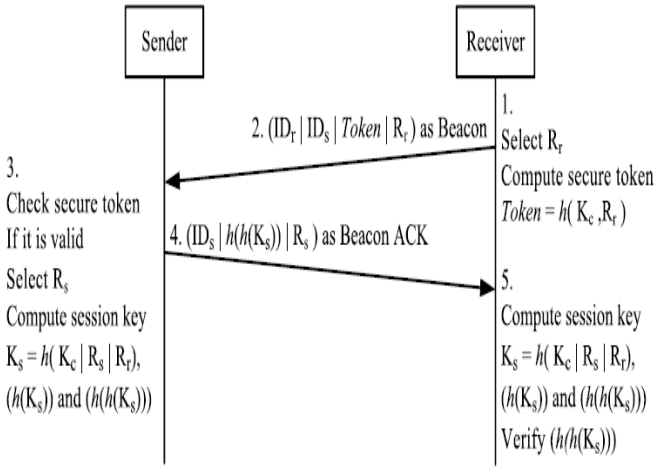


Fig.1. Session Key agreement of Receiver –Initiated scheme

The detailed implementations of these processes are described as follows.

Step 1: The receiver selects a random number Rr and computes the secure token (i.e. *Token* = *h*(Kc| Rr), where *h*(x) denotes a one-way hash function with input x, and the vertical bar | denotes concatenation of strings).

Step 2: The receiver sends its ID, sender's ID, secure token and random number Rr as the beacon.

Step 3: The sender verifies the secure token. If the token is not valid, the sender neglects the beacon and goes to beacon listen mode. If the token is valid, then sender selects a random number Rs and computes the session key Ks = *h*(Kc|Rs|Rr*)*. The sender also computes the hash chain *h*(Ks*)* and *h(h*(Ks*))*.

Step 4: The sender sends the *h(h*(Ks*))* and random number Rs as the ACK.

Step 5: The receiver then computes the required session key Ks = *h*(Kc|Rs|Rr*)* and the hash chain *h*(Ks*)* and *h(h*(Ks*))*. The receiver then verifies the *h(h*(Ks*))*. If the *h(h*(Ks*))* is not valid, the receiver will go back to sleep mode immediately.

Phase 5: Message Authentication and forwarding

With the new created dynamic session key Ks, the sender can encrypt the transmission data via symmetric encryption. A detailed implementation of this process is shown in Fig 2

The implementation steps of authentication and forwarding of the actual data are as follows:

Step 1: The sender sends the h(Ks*)* and EKs(DATA|MACKs(DATA)) to receiver. The EKs(x) denotes encrypts x by using symmetric algorithm with key Ks. The MACKs(DATA) denotes the message authentication function with key Ks, where DATA is the input message.

Step 2: The receiver verifies the h(Ks*)*. If the h(Ks*)* is not valid, the receiver goes back to sleep mode immediately. If the h(Ks*)* is valid, the receiver decrypts the data and checks the MAC of data.

Step 3: The receiver sends the data ACK to sender. Hence, the sender computes h(Ks*)* from known Ks or Kc. To check the received packet valid, the receiver only compares h(Ks*)*. If the h(Ks*)* is not valid, the receiver goes back to sleep mode immediately and discards all the rest processes. It is infeasible to compute the h(Ks*)* from h*(*h(Ks*))*. The sender must compute h(Ks*)* from known Ks or Kc. The hash chains h(Ks*)* and h*(*h(Ks*)* authenticate sender and receiver mutually.

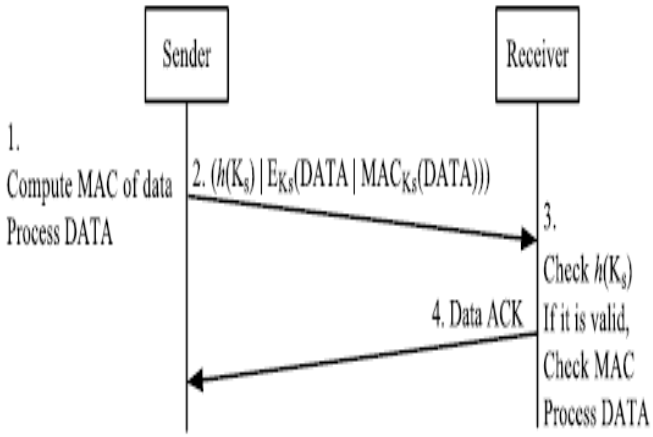


Fig.2. Data Transmission

CONCLUSION:

Security and energy efficiency is the most important concerns in wireless sensor networks (WSNs) designing, since they are prostrate to different types of network attacks and intrusion. The main principle of project is to identify the malicious node and collect the details of the attacker. The MAC protocol tries to reduce energy consumption of sensor nodes by keeping the antenna in sleep mode. The proposed method provides strong authentication which defends denial of sleep attack and triggers the defending mechanism only in the area of attack where the firewalls prevents the attacker from performing the task. The above scheme is effective at transmitter begin side and receiver begin side. The proposed

system can defense against attacks like replay attack and make the sensor nodes return to sleep mode as early as possible to save energy.

REFERENCE

[1] Mechanisms for Detecting and Preventing Denial of Sleep Attacks and Strengthening Signals in Wireless Sensor Networks Chandrakala. P. Goudar1, Shubhada. S. Kulkarni2 1P G Student, Gogte Institute of Technology, Belagavi, Karnataka, India 2Asst Prof, Dept of C S E, Gogte Institute of Technology, Belagavi, Karnataka, India

[2] "A Secure Scheme for Power Exhausting Attacks in Wireless Sensor Networks" Ching-Tsung Hsueh, Chih-Yu Wen and Yen-Chieh Ouyang Department of Electrical Engineering & Graduate Institute of Communication EngineeringNational Chung Hsing UniversityTaichung, Taiwan 40227

[3] "Conservation of battery power by alleviating dos attacks in wireless sensor networks" ,Arpitha R , Chaitra M Department of Information Science and Engineering,
The national institute of engineering, Mysuru 570008, Karnataka, India