RESEARCH ARTICLE OPEN ACCESS

Secure Clustering Algorithm for Wireless Sensor Networks in the Perspective of Confidentiality

Antai Uduak-obong Ekpo [1], Dr Magesh [2]

Department of information and Technology
Faculty of Engineering
SRM University, kattankulathur
Tamil Nadu - India

ABSTRACT

In this paper we propose a means of attaining confidentiality of information broadcasted to child nodes from cluster heads (CH) and also information sent from the cluster heads to the Base Station (BS). One of the main mechanisms used for this purpose is encryption and decryption but due to resource constraints of the sensor nodes these mechanism may affect their performance and increase the computational complexity of the nodes. This research however is focused on finding an appropriate encryption and decryption algorithm which is most suitable for providing confidential communication in a wireless sensor network.

Keywords:- Wireless Sensor Network, Cluster Head, Confidentiality, SLEACH**.**

I. INTRODUCTION

In any form of communication whereby information is passed from one end to another the protection of such security is of paramount effect to the senders. The level of protection for this information offered by the communicating devices varies in strength for different reasons therefore finding the right mechanism for providing this protection if an important task.

Security is one of the most important issues in all the fields, wireless sensor networks is not an exception to this. Wireless Sensor Networks (WSNs) consists of low power, low-cost smart devices which have limited computing resources. With a widespread growth of the applications of WSN, the security mechanisms are also be a rising big issue. A lot of real world applications have been already deployed and many of them will be based on wireless sensor networks. These applications include geographical monitoring, medical care, manufacturing, transportation, military operations, environmental monitoring, industrial machine monitoring, and surveillance systems. This paper discusses typical constraints, security goals, secured clustering and confidentiality with regards to sensor networks and their defensive techniques or countermeasures relevant to the sensor networks, including security methods. The most critical area prone to attack is nearby the base station as the data is more aggregated, that should be kept secure using a number of defensive techniques as stated. Security in sensor networks can be achieved by data confidentiality, authentication, freshness, and integrity. Sensor networks have some special characteristics compared to traditional networks such as the limitation of the available resources, especially the energy.

A flat structure of a large number of sensors often provides low scalability and makes network wide coordination difficult. To solve this problem, hierarchical architectures (clusters) have been proposed to solve the scalability problem. Appropriate clustering can reduce the need for global coordination and restrict most of the sensing, data processing and communication activities within clusters, thus can improve resource utilization and prolong network lifetime. In this kind of organization, nodes are organized into clusters. Cluster heads (CHs) pass messages from their members to the base station (BS). LEACH is the earliest layer architecture routing protocol of WSN. The other ones include PEGASIS， TEEN and so on. They all develop on the basis of LEACH which comparing with general flat multi-routing protocol and static clustering algorithm, LEACH can prolong the network lifetime with a proportion of 15%. To LEACH, because the nodes join in the corresponding cluster according to the strength of signals, the malicious attackers can adopt HELLO attack, Sybil attack, selective report, and modifying data packets to attack easily. According to the generating cluster heads, nodes can be modified to increase the chances of being chosen as cluster heads.

LEACH is a clustering-based protocol that utilizes randomized rotation of local cluster BS (CH) to evenly distribute the energy load among the sensors in the network. LEACH uses localized coordination to enable scalability and robustness for dynamic networks, and incorporates data fusion into the routing protocol to reduce the amount of information that must be transmitted to BS. LEACH rearranges the network's clustering dynamically and periodically, making it difficult for us to rely on long lasting node-to-node trust relationships to make the protocol secure. LEACH assumes every node can directly reach the BS by transmitting with sufficiently high power.

CONSTRAINTS AND ATTACKS IN WSN

Resource constraints: Sensor nodes have limited resources, including low computational capability, small memory, low wireless communication bandwidth, and a limited, usually no rechargeable battery.

Small message size: Messages in sensor networks usually have a small size compared with the existing networks. As a result, there is usually no concept of segmentation in most applications in WSN.

Addressing Schemes: Due to relatively large number of sensor nodes, it is not possible to build global addressing schemes for deployment of a large number of sensor nodes as overhead of identity maintenance is high.

Sensor location and redundancy of data: Position awareness of sensor network is important since data collection is normally based on location. Also there may be common phenomena to collect data, so there is a high probability that this data has some redundancy.

ATTACKS

Attacks against wireless sensor networks are categorized as invasive or non-invasive. Non- invasive attacks generally consist of side channel attacks such as power, timing or frequency based attacks. Invasive attacks are much more common and the more important of these are described in the following sections. Several attacks on sensor networks are listed as follows:

A. Denial-of-Service(DoS) attack In the denial-of-Service(DoS) attack, the hackers's objective is to render target machines inaccessible by legitimate users. There are two types of DoS attacks: Passive attack: Selfish nodes use the network but do not cooperate, saving battery life for their own communications, they do not intend to directly damage other nodes. Active attack: Malicious nodes damage other nodes by causing network outage by partitioning while saving battery life is not a priority. Dos attacks can happen in multiple WSN protocols layers. At physical layer, the DoS attack could be jamming and tempering, at link layer, collision, exhaustion, unfairness, at network layer, neglect and greed, homing, misdirection, black holes and at transport layer, this attack could be performed by malicious flooding and desynchronization. The mechanisms to prevent DoS attacks include payment for network resources, pushback, strong authentication and identification of traffic.

B. Attacks on Information in Transit The most common attacks against WSNs are on information in transit between nodes. Information in transit is vulnerable to eavesdropping, modification, injection, that can be prevented using well established confidentiality, authentication, integrity and replay protection protocols. Traffic analysis can potentially be a big problem in WSNs allowing an attacker to map the routing layout of a network, enabling very tightly targeted attacks to disrupt chosen portions of a network for greatest effect.

C. Node Replication Attack A node replication attack involves an attacker inserting a new node into a network which has been cloned from an existing node, such cloning being a relatively simple task with current sensor node hardware. This new node can act exactly like the old node or it can have some extra behavior, such as transmitting information of interest directly to the attacker. A node replication attack is serious when the base station is cloned. However, as for many deployments, the base station is both in a secure location and much more powerful than the rest of the sensor nodes, so cloning it is much more difficult.

D. Routing attack As with almost all networks there are a number of attacks that target the routing protocol of WSNs, all of which are necessarily insider attacks. Some are as follows:

a. Selective forwarding: Selective forwarding is a way to influence the network traffic by believing that all the participating nodes in network are reliable to forward the message. In selective forwarding attack, malicious nodes simply drop certain messages instead of forwarding every message. Malicious or attacking nodes can refuse to route certain messages and drop them. If they drop all the packets through them, then it is called a blackhole attack. However, if they selectively forward the packets, then it is called selective forwarding. Effectiveness of this attack depends on two factors. First the location of the malicious node, the closer it is to the base station the more traffic it will attract. Second is the percentage of messages it drops. When selective forwarder drops more messages and forwards less, it retains its energy level thus remaining powerful to trick the neighboring nodes.

b. Sinkhole attacks: In sinkhole attacks, adversary attracts the traffic to a compromised node. The simplest way of creating sinkhole is to place a malicious node where it can attract most of the traffic, possibly closer to the base station or malicious node itself deceiving as a base station. One reason for sinkhole attacks is to make selective forwarding possible to attract the traffic towards a compromised node. The nature of sensor networks where all the traffic flows towards one base station makes this type of attacks more susceptible.

c. Sybil attacks: In Sybil attack, a single node presents multiple identities to all other nodes in the WSN. This may mislead other nodes, and hence routes believed to be disjoint w.r.t node can have the same adversary node. Sybil attacks can be used against routing algorithms and topology maintenance; it reduces the effectiveness of fault tolerant schemes such as distributed storage. Another malicious factor is geographic routing where a Sybil node can appear at more than one place simultaneously.

d. Wormholes: In wormhole attacks, an adversary positioned closer to the base station can completely disrupt the traffic by tunnelling messages over a low latency link. Here an adversary convinces the nodes which are multi hop away that they are closer to the base station. This creates a sinkhole because adversary on the other side of the sinkhole provides a better route to the base station.

e. Flooding: Sometime, the malicious node can cause immense traffic of useless messages on the network. This is known as the flooding. Sometimes, malicious nodes replay some actual broadcast messages, and hence generating useless traffic on the network. This can cause congestion, and may eventually lead to the exhaustion of complete nodes. This is a form of Denial of Service attack.

2.2 SECURITY REQUIREMENTS

The goal of security services in WSN is to protect the information and resources from attacks and misbehavior. The security requirements in WSN include:

a. *Availability*: Ensures that the desired network services are available even in the presence of denial of service attacks.

b. *Authorization:* Ensures that only authorized sensors can be involved in providing information to network services.

c. Authentication: Ensures that the communication from one node to another node is genuine. That is, a malicious node cannot masquerade as a trusted network node.

d. *Confidentiality*: Ensures that a given message cannot be understood by anyone other than the desired recipients.

e. *Integrity*: Ensures that a message sent from one node to another is not modified by malicious intermediate nodes.

f. *Non-repudiation*: Denotes that a node cannot deny sending a message it has previously sent.

g. *Data Freshness*: Implies that the data is recent and ensures that no adversary can replay old messages.

h. *Robustness*: When some nodes are compromised the entire network should not be compromised.

i. *Self-organization*: Nodes should be flexible enough to be self-organizing (autonomous) and self-healing (failure tolerant).

j. *Time Synchronization:* These protocols should not be manipulated to produce incorrect data.

II. RELATED WORKS

SECURITY SOLUTIONS IN SENSOR NETWORKS

Security schemes can be applied to provide security in wireless sensor networks, but keeping in view their resource starved nature it is very difficult to do so. Some researchers are striving to develop improved WSN protocols, others are attempting to improve node design; still others are working to resolve security issues including the main WSN security threat of insecure radio links with eavesdropping and information corruption possible. Most security mechanisms that exist today require intensive computation and memory which is the limiting factor in wireless sensor networks. Many security mechanisms require repeated transmission/communications between the sensor nodes which are further drawn in their resources. The number of security suites already exist that are at least some way appropriate for use in WSNs, and combat some of the threats to these networks. This section review some of the more popular and more suitable solutions here. 3.1 SPINS: Security Protocols For Sensor Networks Adrian Perrig et al.[5] proposed "SPINS" a suite of security protocols optimized for sensor networks. SPINS has two secure building blocks: SNEP and μTESLA. SNEP includes: data confidentiality, two-party data authentication, and evidence of data freshness. μTESLA provides authenticated broadcast for severely resource-constrained environments. 3.1.1 SNEP: Sensor Network Encryption Protocol SNEP provides a number of following advantages. 1. It has low communication overhead as it only adds 8 bytes per message. 2. Like many cryptographic protocols it uses a counter, but avoids transmitting the counter value by keeping state at both end points. 3. SNEP achieves semantic security, which prevents eavesdroppers from inferring the message content from the encrypted message. 4. Finally, SNEP protocol offers data authentication, replay protection, and weak message freshness. However, sending data over the RF channel requires more energy. So, SNEP construct another cryptographic mechanism that achieves semantic security with no additional transmission overhead. It relies on a shared counter between the sender and the receiver for the block cipher in counter mode (CTR). Since the communicating parties share the counter and increment it after each block, the counter does not need to be sent with the message. To achieve two-party authentication and data integrity, SNEP uses a message authentication code (MAC).The combination of these mechanisms form Sensor Network Encryption Protocol SNEP. SNEP offers the following properties: • Semantic security: Since the counter value is incremented after each message, the same message is encrypted differently each time. The counter value is long enough that it never repeats within the lifetime of the node. • Data authentication: If the MAC verifies correctly, the receiver can be assured that the message originated from the claimed sender. • Replay protection: The counter value in the MAC prevents replaying old messages. Note that if the counter were not present in the MAC, an adversary could easily replay messages. • Weak freshness: If the message verified correctly, the receiver knows that the message must have been sent after the previous message it received correctly (that had a lower counter value). This enforces a message ordering and yields weak freshness. • Low communication overhead: The counter state is kept at each end point and does not need to be sent in each message. 3.1.2 μTesla: Authenticated Broadcast Asymmetric digital signatures are impractical for sensor networks for the authentication, as they require long signatures with high communication overhead of 50-1000. Earlier TESLA protocol provided efficient authenticated broadcast However, TESLA was not designed for sensor networks. Adrian Perrig et al. proposed μTESLA to solve the following inadequacies of TESLA in sensor networks: • TESLA authenticates the initial packet with a digital signature, which is too expensive for our sensor nodes. μTESLA uses only symmetric mechanisms. • Disclosing a key in each packet requires too much energy for sending and receiving. μTESLA discloses the key once per epoch. • It is expensive to store a one-way key chain in a sensor node. μTESLA restricts the number of authenticated senders.

III. ARCHITECTURE USED FOR CLUSTERING AND CLUSTER HEAD SELECTION

Sub grouping of a sensor network is usually done for the efficient use of network resources like battery power energy consumption routing etc. In some sensor network applications clustering is often done for load balancing among several part of the network to increase the network's lifetime. Existing work on clustering mainly resolves routing of data efficiently
to the sink node. Here distributed region based clustering is followed in order to implement the security services. Our approach has two phases; a cluster formation and cluster head selection process. The following sections explain more in detail about the two step process.

1. Cluster Formation

The sink node is responsible for initiating cluster formation. For this the sink node broadcasts a control message to the sensor nodes in the network. The neighbouring node receives the control message and identifies the region based on its location in the sensor field. The control message consists of the details about the region and the group id for each cluster. The received sensor node then forwards the control message to the neighbouring sensor node. If the received neighbour node previously forwarded the same control message then it is ignored there by unnecessary transmission of control messages by the node is avoided.
Also as we have established, the cluster head monitors the activities in its cluster and if a node has low energy / battery life or is even compromised, the cluster head can request for a mobile node to join its cluster so as to maintain load

balancing. This request is done through the base station which broadcast the request to surrounding clusters. This broadcast however must be confidential as to prevent adversary from injecting malicious nodes.

2. Cluster Head Selection

The main role of a cluster head is to maintain the state of its members and aggregate data from its members and forward genuine data to the sink node. The cluster head is selected by the sink node based on the location in a region. Once a node receives the control message for the formation of cluster, it also checks whether it is present in that sub region based on the information available in the control message. If it is present in that sub region it becomes a cluster head and sends an advertisement message to its members. The message consists of the cluster head node id and its location. The received members then send back an ACK message to the cluster head. The ACK message is routed back to the cluster head by the path where it received the advertisement message.

The cluster head performs complex tasks such as data aggregation, authentication, and generation of keys for decryption and so on. So the energy of the cluster head depletes due to complex computational and communication process. Hence re - clustering can be done so that different nodes will get a chance to become a cluster head thereby conserving the energy of the sensor nodes in the network. Re - clustering can be done either on a periodic basis or based on the no of times a data has been received from a particular cluster.

(3) The allocating of TDMA time slot

Ever cluster-head allocates TDMA time slot according to the number of nodes registering in his cluster. The clusterhead send to the cluster members the scheduling information, and thus to ensure that every member has its own time slot to send its data. In the meantime, the cluster head packs the variable ch(r) in the datagram.

{ID_H|[sequence|CDMAcode|TDMAschedule|chr |]

(4) The transmitting of data

After the building of cluster-heads, the cluster members starting data acquisition in its own TDMA time slot, then encode the data and send them to other cluster-heads. Once a frame is completed, the heads decode the data, run data fusion algorithm, and send the fused data to the base station.

IV. COMMUNICATION ARCHITECTURE

Generally the broadcast is the fundamental communication primitives because the sensor nodes communicate over a wireless network. Due to this broadcast protocol on one hand they affect the trust assumptions and on another hand they minimize the energy usage.

A clustered sensor network forms around one cluster head where a collection of various clusters form around one or more base station which interfaces between the sensor network and the outside world. The base station therefore forms the roots and fundamentals for the cluster and a periodic broadcast and transmission of state packets allows the node to form a routing topology.

Each node forwards a message towards its clusterhead and each clusterhead forwarding the message towards a basestation where the clusterhead collects all the messages from its cluster and forwards them to the basestation.

The nodes can also find packets addressed to it while the clusterhead also being a node should be able to forward messages to the basestation, find the packets forwarded to it from the cluster nodes and also the basestation. It should be able to handle message forwarding and broadcast. We assume the clusterhead has capabilities same as the sensor nodes only with more battery life to surpass the other nodes lifetime, enough memory to store cryptographic key and a means of communicating with the basestation.

Due to the fact that most communication involves the basestation to clusterhead and not between two cluster nodes we have acknowledge a communication pattern within our network and it entails:

1. Node to Clusterhead communication
   E.g. Sensor Reading

2. Clusterhead to Node communication
   E.g. Specific Request

3. Clusterhead to Basestation
   E.g. Sensor Reading Report

4. Clusterhead to All nodes communication
   E.g. Routing beacons, queries and Re-programming the entire network

TRUST REQUIREMENT

A sensor network might be kept in untrusted locations while it may be possible to guarantee the integrity of each node through dedicated secure microcontrollers, we feel that such architecture is quite limiting and doesn't cover the general use and implementation of the wireless sensor networks so instead we assume that individual sensor nodes are untrusted.

Basic wireless communication isn't secured and also because its broadcasted any adversary can eavesdrop on traffic, insert new message or replay old messages. Therefore one shouldn't place any trust assumptions on the communication infrastructure except that the messages are delivered to the destination with zero probability.

Understanding that the clusterhead is the gateway for communication between the outside world and the clustered sensors, compromising the clusterhead or even the basestation would lead to the infiltration of the entire sensor network. All sensor nodes should trust the clusterhead which in turn trust a basestation wherefore at certain time, each node gets a certain master secret key from the basestation which it shares with it and all other key would be derived from this key (master key).

V. PROPOSED SYSTEM

The low computational power implies that special cryptographic algorithms that require less powerful processors need to be used. The combination of both problems leads us to a situation where new solutions to security protocols need to be taken. These types of new approaches take into account basically two main goals: reduce the overhead that protocol imposes to messages, and provide reasonable protection while limiting use of resources. With these limited computation resources available on our plat-form, we cannot afford to use asymmetric cryptography and so we use symmetric cryptographic primitives to provide the security. Due to the limited program store, we construct all cryptographic primitives (i.e. encryption, message authentication code, hash, random number generator) out of a single block cipher for code reuse and to decrease communication overhead we exploit common state between the communicating nodes.

Design Guidelines and notations

1. A, B are principals, such as communicating nodes.
2. NA is a nonce (A nonce is an unpredictable bit string which helps to achieve freshness)
3. XAB denotes a directionless master secret key (symmetric) which is shared between basestation and the sensor nodes such that XAB=XBA=AXB=BAX
4. KAB and KBA denote the encryption key shared between A and B which is derived
5. from the master secret key XAB based on the direction of the communication.
6. KAB = FXAB and KBA= FXAB where F is a pseudo random function (PRF) and it depends on the direction of the communication
7. M[XAB] is the encryption of message M using the encryption key XAB.

VI. CONFIDENTIALITY IMPLEMENTATION

To achieve small SNEP messages, we assume that two communicating Parties A and B know each other with counter values CA and CB and so the counter does not need to be added to each encrypted message. However, if any messages are lost then the shared counter state can become inconsistent. Now present protocols to synchronize the counter state. To bootstrap the counter [2] values that initially, we use the following protocol:

1. A --> B: CA,
2. B --> A: CB, MAC (K'BA CA| | CB)
3 .A --> B: CB, MAC (K'AB CA| | CB)

Note that the counter values are not secret, so there is no need of using encryption method. However, this protocol requires strong freshness, so both parties A and B use their counters as a nonce (assuming that the protocol never runs twice with the same counter values, hence incrementing counters if necessary) and here the MAC does not need to include the names of A or B.

Since the MAC keys K'AB and K'BA implicitly bind the message to the parties, and that ensure the direction of the message. If party A realizes that the counter CB of party B is not synchronized any more, A can request the current counter of B using a nonce NA to ensure strong freshness of the reply:

1. A --> B : NA
2. B --> A : CB, MAC (K'BA NA | | C B)

To prevent a potential denial-of-service (DoS) attack, where an attacker keeps sending false messages to lure the nodes into performing counter synchronization and the nodes can switch to sending the counter with each encrypted message they send. To detect such a denial-of-service attack, there is an another approach where one attaches another short MAC to the message that does not depend on the counter.

Using this model for confidentiality assurance in wireless sensor networks we ensure some key properties in the network being ;

1. ***Semantic security:*** Since the counter value is incremented after each message [7] that means the same message is encrypted differently at each time. The counter value is sufficiently long enough, So never repeat within the lifetime of the node.
2. ***Data authentication:*** If the MAC verifies correctly, a receiver knows that the message is send from the claimed sender.
3. ***Replay protection****:* The counter value in the MAC prevents replay of old messages. If the counter were not present in a MAC, an adversary could easily replay messages.
4. ***Weak freshness:*** If the message is verifies correctly, the receiver knows that a message must have been sent after the previous message it received correctly. This leads a message ordering and yields weak freshness. Low communication overhead. The counter state is kept at each end point and does not need to be sent in each message.

VII. CONCLUSIONS

All energy spent on sending or receiving messages is to be compared with our limited platform energy spent for security is negligible. It is possible to encrypt and authenticate all sensor readings. The communication costs are also very small. Data authentication, freshness, and confidentiality properties use 6 bytes out of 30 byte packets. So, it is feasible to guarantee these properties on a per packet basis. we have identified and implemented useful security protocols for sensor networks: authenticated and confidential communication, and authenticated broadcast. We have implemented applications including and a secure node-to-node key agreement protocol.

In addition we have also implemented a hierarchical management of clusters using selected cluster heads. This also reduces the overhead in communication and resource conservative, which is key to cluster performs.

REFERENCES

[1]. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey", Computer Networks Journal, Elsevier Science, Vol. 38, No. 4, pg. 393– 422, March 2002.

[2]. J. M. Kahn, R. H. Katz, and K.S. Pistcr, Mobile Networking for Smart Dust, ACM/IEEE International Conference on Mobile Computing (MobiCom '99), Seattle, WA, 1999, 217 – 278.

[3]. J. Staddon, D. Balfanz, and G. Durfee. "Efficient tracing of failed nodes in sensor networks", presented at the first ACM International workshop on Wireless sensor networks and applications (WSNA), ACM Press, 2002, 122-130.

[4]. Chris Karlof and David Wagner, "Secure routing in networks: attacks and countermeasures", Elsevier's AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols. 2003.

[5]. Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, and J. D. Tygar. SPINS: Security protocols for sensor networks presented in Seventh Annual ACM International Conference on Mobile Computing and Networks (MobiCom 2001), July 2001.

[6]. B. Veeramallu, S.Sahitya, Ch. LavanyaSusanna Confidentiality in Wireless sensor Networks. International Journal of Soft Computing and Engineering (IJSCE), January 2013.

[7]. Adrian Perrig, Robert Szewczyk, J.D.Tygar, Victor Wen, and David Culler SPINS: Security Protocols for Sensor Networks, university of California, Berkeley.

[8]. Yanfei Sun et al: secure leach routing protocol based on low-power cluster-head selection algorithm for wireless sensor networks. Proceedings of 2007 International Symposium on Intelligent Signal Processing and Communication Systems Nov.28-Dec.1, 2007 Xiamen, China.

[9]. M. Bellare, A. Desai, E. Jokipii and P. Rogaway, A concrete security treatment of symmetric encryption: Analysis of the DES modes of operation, in: Symposium on Foundations of Computer Science (1997). R.L. Rivest, the RC5 encryption algorithm, in: Workshop on Fast

[10]. Llanos Tobarra, Diego Cazorla and Fernando al Analysis of Sensor Network Encryption Protocol".

[11] HU Xiangdong, WEI Qinfang, CUI Ping, CAI Jun, LIU Guangcai, LUO Wei, Modelling on Secure and Efficient Clustering for Wireless Sensor Networks.. 2008

[12] Mona El_Saadawy, Eman Shaaban, Enhancing S-LEACH Security for Wireless Sensor Networks: Computer Systems Department, University Cairo, Egypt.

[13] L.Eschenauer and V.Gligor, “A key-management scheme for distributed sensor networks.” In ACM on Computer and Communications Security, V. Atluri, Ed. ACM, 2002.