RESEARCH REVIEW JOURNALS
Volume-03
Issue-12
December-2018

ISSN: 2455-3085 (Online)
RESEARCH REVIEW International Journal of Multidisciplinary
www.rrjournals.com[UGC Listed Journal]

A Novel Procedure to Bring Forth Security for Static Homogeneous WSN's Using Secret Keying Protocol

[*1]Itfaq Ahmad Mir, [2]Prof.G.M.Mir, & [3]Anwaar Ahmad Wani

[*1]*Research Scholar, Department of Computer Application, Mewar University, Rajasthan, INDIA.*
[2]*Professor, College of Agri. Engineering, SKUAST, Kashmir, J&K, INDIA.*
[3]*Teaching & Research Assistant, Department of Computer Application, Mewar University, Rajasthan, INDIA.*

ARTICLE DETAILS

Article History
Published Online: 10 December 2018

Keywords
SKSP, Wireless Sensor Network, Keying, Secret keying.

ABSTRACT

Wireless Sensor Network(WSN) has no exclusive security solution accessible which can be suitable for all categories of applications. based on the impact of the attacks & threats, the security solution must be designed. Our proposed protocol represents a security solution which has been designed for small scale applications that requires minimum level of security. SKSP uses one-way hash function to dynamically generate the keys that avoid transmission of key during the runtime. Grouping among the nodes is introduced to minimize the memory overhead. The performance of SKSP is carried out on Castalia simulator, which shows evaluated results. The SKSP identifies the attacks such as Relay Attack, Sybill Attack and DoS Attack. The proposed SKSP protocol has been analyzed using the parameters such as Network availability, packet Delivery and Network energy on replay and DoS attack. In SKSP protocol, the scalability can be still increased by introducing the clustering concept in order to reduce the traffic and overhead the base station.

1. Introduction

In WSN, there is no unique security solution available which is suitable for all categories of applications. While designing a security solution for a WSN, analysis must be done on the consequences of the attackers in the application area. Based on the impact of the attacks and threats, the security solution must be designed. WSN applications such as Industrial process control, SMART home, SMART offices, health care monitoring, small scale environmental monitoring etc. require the minimal amount of security. The coverage of the network is also within a small region or a building. The network that consists of the nodes of similar capabilities is also called as homogeneous network. While incorporating the security for the small scale applications, the resource must be utilized efficiently. These kinds of applications do not require complex security operations to be performed. Hence incorporating the security in wireless sensor networks based on the application requirements is very challenging. Sensor Node consists of both volatile and non-volatile memory. In the non-volatile memory the static information such as program, node-ID, routing table, and security related data can be stored. Due to the improvements in the hardware technology, the physical size of memory is reduced by increasing the capacity of memory.

The key distribution and the key management are considered to be the core of secure communication. One of the points to be noticed is that no key distribution scheme is ideal to all kind of sensor network applications. If the key is used during the lifetime of the network, it may be modified or hacked by the adversaries. So it is better to generate the keys for every round or session when ever required. The sensor nodes should not always depend on static keys that are preloaded or generated only once during the entire lifetime of the nodes. The network should have the re-keying facility to change the keys that are identified by the adversaries and the keys whose lifetime is expired.

Many key distribution and management schemes are proposed such as pair-wise shared key schemes and random key pre-distribution scheme. Sharing a single secret key among all the nodes is vulnerable to attack. Instead, having different pair-wise keys for node every is much more secure, but this solution occupies unnecessary storage space on a sensor node. Instead of pre-distributing the keys in each sensor nodes, some parameters can be pre-distributed and that parameters can be used to dynamically generate the keys. Security architectures have been proposed by considering the design issues of WSN.

2. Literature Review

Throughout our everyday lives, cyber-attacks threaten critical infrastructure and small domestic networks. Cyber-attacks are becoming more complex, making it harder to protect our networked networks against them (as far as their attack patterns, forms, and methods are concerned). The security of networked networks is extensively assessed by three key components: (i) safety measures, (ii) security models and (iii) security metrics. Security assessment is a mechanism in which security-related evidence such as detecting vulnerabilities and their related threats are collected in your networked environment. These can be gathered by test bed tests, simulation and honey pots. Security metrics reflect different quantitative and qualitative measures of cyber safety relevant to the networked environment and are developed in a variety of corrections such as the business community. Safety models are established using security metrics, and safety measures are used to determine the safety of a networked device.

SenSec uses a variant of skipjack algorithm called skipjack-X for generating the cipher text by introducing one more secret key without affecting the internal structure of the algorithm. SenSec does not defend against the replay attack.

Kalpana Sharma et al. (Sharma, Ghose and Yadav 2009) have introduced Intelligent Security Agent architecture that uses trust framework which consists of 11 parameters to compute the trust level of all its neighbors. It requires more amount of memory to maintain these parameters.

SPINS is a security framework that does not focus on the implementation efficiency, instead they focus mainly on the security protocol. But TinySec [60] architecture focuses on the implementation efficiency. In this architecture, key scheduling has to be pre-computed in RC5 which requires additional 104 bytes of RAM per key.

KuiRenet. al. have proposed a location-aware-end-to-end security framework which is robust against DoS attack. It uses efficient en-route false data filtering scheme in order to identify the false data injection attack. The major drawback in LEDS is its increased resource consumption due to hop-by-hop authentication, hop-by-hop decryption, processing and encryption.

3. Methodology

3.1. Simple Key Selection Protocol(SKSP)

To provide security which is suitable for small scale WSN, the key management mechanism should be designed simply and securely by efficiently utilizing the resources. Once the nodes are deployed in the field, if the keys are distributed over-the-air, there are possibilities to compromise the keys by the attackers.

In the proposed Simple Key Selection Protocol (SKSP), the keys are not directly distributed over the network at any time. Instead, the parameters that are used to generate the keys are transmitted only during the re-keying. It is significantly hard for an adversary to identify those parameters. The SKSP satisfies the following security properties:

Backward Secrecy: Even if an adversary recovered an adjacent subset of keys, it is impossible to recover the previous keys.

Privacy: Even the node is physically captured by an adversary; the secret information in the node's memory cannot be retrieved.

Data Integrity: Data Integrity ensures that the data during the transmission over the network is not modified by an adversary.

Secure Management: SKSP provides a secure method for key generation as well as for re-keying which is very much necessary in defending against cryptography attacks.

SKSP uses three types of keys which are Data Encryption Keys (DKs) that are generated and shared within a group and BS, Re-keying Key (RK) that is generated and shared between a node and BS which is used during the re-keying and a Secret Key (SK) that is shared between a node and BS. The keys DK and RK were encrypted using SK and maintained in its volatile memory. Due to this little bit of computational overhead, even if the nodes are physically captured, the keys cannot be retrieved from its volatile memory. SKSP assumes the WSN in which the nodes are static with similar

computational and communication capabilities. The network uses skipjack algorithm for encryption and decryption process. We have chosen this algorithm because the memory requirement is very less and encryption/decryption and key setup efficiency is also good. This proposed approach is designed to identify the DoS attack, Packet Replay attack and Sybil attack. Identifying those attacks will help to increase the network lifetime.

3.2. Grouping of Nodes

If all the nodes in the network are using the same set of keys, all the nodes have to participate in re-keying which is an overhead. To reduce this overhead, the nodes are grouped based on the size of the network. After grouping, if any node needs re-keying, the other nodes in that group itself have to participate in re-keying process. This avoids the overhead of re-keying for the remaining nodes which belong to other group(s). Let N be the number of nodes, NK be the types of key, NG be the number of groups and NDK be the number of data encryption keys per node. Let us consider the following example.

N = 100 (Number of nodes in the network)
NK = 2 (Data encryption keys and Re-keying keys)
NDK = 9
NG = Round ((N/NK)/ 9) = 6 groups

3.3. Secret Key (SK) Deployment

Before a node is deployed, a static Secret Key (SK) has to be embedded in the source code. The source code is converted to its executable (.exe) format and loaded in the node's non-volatile memory. Then every node is pre-distributed with 2 pairs of parameters ki, ki-1 and ri, ri-1 which are used for generating the Data Encryption keys and Re-keying keys respectively using one way hash function. A unique seed value Seedi is preset in every node during the deployment. The counter value of ith node (CNTi) used by the key selection protocol for all the sensor nodes is initialized as:

$$\forall_{i=1}^{N} C_i = Seed_i \quad (3.1)$$

After deployment every node generates its 9 data encryption keys as:

DK1 = h(ki, ki-1) DK2= h(DK1, ki) DK3= h(DK2, DK1)
DK4= h(DK3, DK2) DK5= h(DK4, DK3) DK6= h(DK5, DK4)
DK7= h(DK6, DK5) DK8= h(DK7, DK6) DK9= h(DK8, DK7)

The parameters ri and ri-1 will be discussed later while generating a key for rekeying.

3.4. Key Selection Process

Every sensor node maintains a key pool (kp) that consists of 9 keys, which are generated by the node immediately after its deployment. We limit the NDK as 9 since our key selection protocol uses a function SoD that always results in a single digit. To increase the security, NDK can be increased but will lead to increase in computation overhead during the key generation and rekeying. For each data transmission, the node i calculates the key number kni from its key pool as:

Let

$$x = \left((ID_i >> \varepsilon_{i,\tau_i}) \times \varepsilon_{i,\tau_i} \right)$$

$$kn_i = \begin{cases} SoD(x) & if\ x > 9 \\ x & if\ x \le 9 \end{cases} \quad (3.2)$$

$$\tau_l = \tau_{l-1} + T_{\text{sec}} \quad where\ \tau_0 = 0 \tag{3.3}$$

where ε_{i,τ_l} is the counter value at τlth time interval of ith sensor node which is initialized with the Seedi and will be incremented as $\varepsilon_{i,\tau_l} = \varepsilon_{i,\tau_l - 1} + 1$ for each constant time interval Tsec. τlis the time interval which is initialized with 0 and is incremented by Tsec for each time interval. That is,

$$k = kp\,[kn_i] \tag{3.4}$$

Now key k to encrypt the current message msg chosen from kp as: $k = kp\,[kn_i]$

and the encrypted message E_{msg} is

$$E_{msg_i} = E\left(k, (msg_i, ID_i, kn_i, T_i\right) \tag{3.5}$$

Ti is the time stamp at which the ith node transmits a packet, np is the number of packets transmitted by the ith node during the last Q seconds and kni is the key number in kp. The keys that are generated by all the nodes of a group will be same, but the selection of key for the current communication will not be same. How the BS identifies the counter value ε of the nodes a and b which is used for key selection. Every node in the network will maintain a counter value which is initialized with a seed value during the deployment. In our example scenario, counter value of node a and node b are initialized with seed a and seed b respectively. This counter value is incremented by 1 for each constant time interval Tsecs. The nodes can be deployed at anytime interval. During the deployment of sensor nodes, the BS maintains the time interval τdep, node_num at which the nodes are deployed. In this scenario, we assume that node a is deployed in 0th time interval τ0 and node b is deployed in 2nd time interval τ2 of BS, so τdep,a= 0 and τdep,b= 2.Data transmission between node a and BS is occurred in different time slots. Data transmission between node b and BS is occurred in same timeslot. From this scenario we prove that the key selection protocol chooses the right key when the BS receives the packet at the same time slot and different time slots.

Node a transfers a packet at the time Ta during the time interval τ3. It computes the key k to encrypt the msga using the key selection protocol as

$$\tau_{dep,a} = 0,$$

$$\tau_{dep,b} = 2$$

Applying this in Equation (3.2), we get

$$\varepsilon_{a,\tau_3} = 3 + Seed_a \tag{3.6}$$

By applying x in Equation (3.3), we get the value of the key number kna. By using this kna, the key k for the current operation has to be taken from the key pool as kp[kna]. This key k is used to encrypt the msgaas shown in Equation (3.5). The BS receives the encrypted message E_{msg_a} from the node a at the time Ts,a during the time interval τ4. Let $T_{s,a}$, be the time taken to transfer a packet to BS, be the synchronization time difference between a source node and BS. Then BS selects the key k from its key pool as shown below.

$$x = SoD\ \left((ID_a >> (3 + Seed_a)) \times (3 + Seed_a)\right)$$

$$kn_a = x$$

$$k = kp[kn_a]$$

$$E_{msg_a} = E\left(k, (msg_a, ID_a, kn_a, T_i\right)$$

$$\alpha = T_{s,a} - \delta_{d,a} - \delta_{diff}$$

$$\alpha = T_{s,a} - \delta_{d,a} - \delta_{diff}$$

$$T_{s,a} = T_a + \delta_{d,a} + \delta_{diff}$$

$$\therefore \alpha = T_a$$

$$\beta = T_{s,a}$$

$$x = SoD\ \left((ID_a >> \varepsilon_{a,\tau_3}) \times \varepsilon_{a,\tau_3}\right)$$

$$\gamma = \tau_4 \times T_{\text{sec}} = 4T_{\text{sec}}$$

$$\varepsilon_{Ba,4} = 4 + Seed_a - \tau_{dep,a}$$

$$= 4 + Seed_a - 0$$

T s,a is the time at which the BS receives the packet and τ_4. Since the BS has received the packet at the time interval τ4. Now the counter value $\varepsilon_{Ba,4}$ to be used by the BS is calculated as given below:

$$\varepsilon_{Ba,4} = 4 + Seed_a \tag{3.7}$$

The time at which the packet has been transmitted from node a isTa that falls in the 3rd time interval τ3 and it has been received by BS is Ts, a which falls in the 4th time interval τ4. So $\alpha > \gamma \,\&\&\, \beta < \gamma$. By this, it is clear that the packet transmission and reception falls in different time intervals. So the BShas to use previous counter value to obtain the correct key number of the key pool to successfully decrypt the packet. By substituting the value obtained in Equation (3.7) in the Equation (3.2), we get

$$y = SoD\left((ID_a >> \varepsilon_{B_a,4} -) \times (\varepsilon_{B_a,4} - 1)\right)$$

$$y = SoD\left((ID_a >> 4 + (Seed_a - 1)) \times (4 + Seed_a - 1)\right)$$

$$y = SoD\left((ID_a >> (3 + Seed_a)) \times (3 + Seed_a)\right)$$

$$kn_B = y$$

$$k = kp_a[kn_B]$$

$$Payload = D(k, E_{msg_a}) = (msg_a, ID_a, kn_a, T_i)$$

3.5. Re-keying

In the proposed approach SKSP, the re-keying process is initiated by the sensor node only if any two (other than the last two) consecutive keys are invalidated (compromised). Once all the sensor nodes are ready to deploy in the field, two parameters ri and ri-1 have to be preset. A new re-keying key will be generated by one-way hash function to communicate with the Base Station. The re-keying parameters ri, and ri-1 are different for every node. Like the DK, the consecutive re-keying keys are also generated using the previous keys. The protocol for Re-keying mechanism between a node and BS is given below:

Step 1. (Node → BS)

First the node that needs to re-key the existing data encryption keys will send a request to the BS using RKRQ message. This message includes the node ID, its existing group number and the hash value generated by one way hash function. This information is included in this message by encrypting the same using the re-keying encryption key.

RKRQ, BS, SrcID, ERKni (SrcID, GrpNo, H (SK, SrcID ⊕ GrpNo))

Step 2. (BS → Node)

Next the BS has to authenticate the node using RKA1 message before sending the parameters for re-keying the data encryption keys. This hash value is generated by using SK and RDT. This information is included in this message by encrypting the same using the re-keying encryption key. All the upcoming messages regarding the re-keying operation use the re-keying keys itself.

RKA1, DestNode, BS, ERKni (DestID, H (SK, RDT), RDT, T1)

Step 3. (Node → BS)

After receiving the message RKA1, the sensor node generates a hash value using RDT and T1 and compares with the hash value sent by the BS. Then the sensor node authenticates with BS using RKA2 message. This message includes the hash value and the timestamp. The message used for generating hash value is the XOR value of the random text and the time stamp T1.

RKA2, BS, SrcNode, ERKni (H (SK, RDT ⊕T1), T2)

Step 4. (BS → Node)

Now the BS compares the hash value in the RKA2 message with hash value generated by itself using the RDT, T1 and T2. If both are same, the BS sends the parameters for generating the data encryption key to the sensor node using the RKPM message. This message includes the node ID, the parameters, another RDT, timestamp T3 and the hash value generated by SK and $k_i \oplus k_{i-1} \oplus T2$.

RKPM, DestNode, BS, ERKn$_i$ (DestID, k_i, k_{i-1}, H (SK, $k_i \oplus k_{i-1} \oplus T2$), RDT, T3)

Step 5. (Node → BS)

Finally the node that receives the parameters has to send an acknowledgment to BS using RKPA message. This message includes the timestamp and the hash value of RDT ⊕ T3.

RKPA, BS, SrcNode, ERKni (H (SK, RDT ⊕ T3), T4)

The re-keying protocol requires five transactions in order to complete the process for a single node. If there are N numbers of nodes in the network, it consists of N/NG nodes per group. So the number of communications NC in the network during the re-keying is,

*NC = (N/NG) * 5*

Since the re-keying occurs occasionally, it does not increase the communication overhead to the network.

The grouping in the network does not mean that the nodes have to communicate only through the nodes that belong to the same group. The grouping of nodes has been introduced in order to maintain only the different sets of keys by the groups. By this, if any two consecutive keys that belong to a group have been compromised, that group itself has to initiate the re-keying process. The other groups need not initiate the re-keying process. Also, any node that belongs to a group can send the data to the sink via the intermediate nodes that belongs to any other groups. So it is not mandatory to know the key for the current communication by the intermediate nodes to forward the packet to the sink. Hence the key connectivity is not considered to be an issue in our security scheme.

4. Time Synchronization

Since the key management in our method requires time to be synchronized between the nodes in order to maintain the correct counter values in BS as well as in the sensor nodes, we use Gradient Time Synchronization Protocol (GTSP) which synchronizes the clock accurately in decentralized fashion. Using GTSP, the node synchronizes its logical clock by exchanging beacons for every 30 seconds which consists of the timing information such as current logical time and relative logical clock rate with its neighbors. After receiving the beacons from the neighbors, the node updates its absolute clock rate and its logical clock offset. Every node maintains a neighbor table which consists of logical clock value, the relative logical clock rate and last beacon arrival timestamp of their neighbors. This protocol is robust against the link and the node failures. This protocol requires each node to broadcast only the time information during the synchronization period, the communication overhead is minimum. But the GTSP is vulnerable to time synchronization attacks. Any malicious node can send false synchronization messages to the neighboring nodes and claim to be legitimate. To provide security for GTSP, filters have been added into the architecture of GTSP. We have used the filters such as logical clock rate filter; logical time filter and timestamp filter. The frequency of sending beacons by a node is set to 30 seconds which is increased after the synchronization period.

Firstly, when a beacon is received from its neighbor, the node checks the timestamp of the last received beacon. If the time difference is less than 30 seconds, it adds the sender node ID in the timestamp blacklist filter. Secondly, the node computes the logical clock rate. If the difference between the received logical clock rate and the most recent logical clock rate of the neighbor is more than the accepted value, it adds the sender node ID in the logical clock rate filter. Finally, the node verifies the received logical time of its neighbor with its own logical time. If it is less than the current logical, then it adds the neighbor node ID in the logical blacklist filter. Based on these three filters, the legitimate nodes can filter the beacons sent by the nodes in the blacklist filters.

5. Theoretical Analysis

In SKSP, the storage requirement and security analysis have been done theoretically to evaluate the efficiency of the protocol. The details are given in the following sections.

5.1.Storage Requirement Analysis

Storage requirement of SKSP security scheme falls into two categories.

1 .Storage at each node.

2 .Storage at the BS.

Let *Ln* - bit length of the Node identifier

Lsk- bit length of the SK at each node

Lk- bit length of the keying parameter Ki and K i-1

Lr– bit length of the re-keying parameter ri and r i-1

Lc– bit length of the counter

Lnp– bit length of the counter to count the no. of packets transmitted during the last *Q* seconds

Ldk– bit length of the generated keys

Lskpalg– bit length of Skip jack algorithm

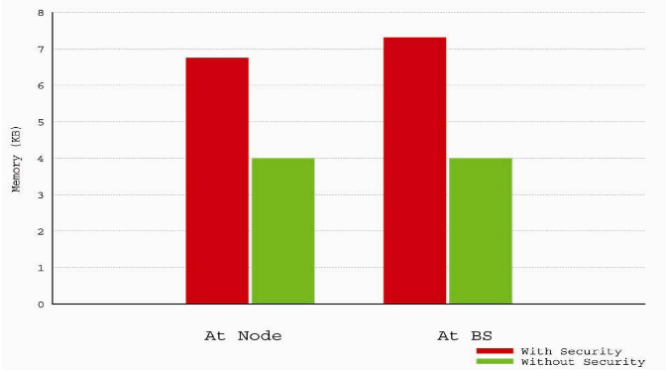
The storage at node in SKSP is given as follows:

SRn = Ln + Lsk + 2Lk + 2Lr + Lc + Lnp + 9 Ldk + Lskpalg

The storage at BS for *m* number of nodes in SKSP scheme is given as follows:

$$SR_{bs} = \sum_{i=1}^{m}(L_{n_i} + \tau_{dep,i} + \varepsilon_{B_i,\tau_l}) + L_{sk} + N_{G'}(2L_k +$$

Considering the key size as 80 bits long, node ID as 32 bits long, the counter size as 8 bits long, the keying and re-keying parameters are 320 bits long, the storage requirement for skipjack algorithm under CBC mode is 21366 bits long and the number of groups in the network as 2, the total storage needed at node SR*n* is 6.751KB and base station SR*bs* is 7.42KB.



5.2. Security Analysis

In this section we explain how SKSP detects various attacks such as Packet Replay attack, Sybil attack and DoS attack.

Replay Attack: Replay attack occurs when an attacker captures the packet at some point of time and then replays the same at later point of time without any modification.

Selective Forwarding Attack: The attacker node in the path of the data flow refuses to forward certain messages from the source node.

Sybil Attack: In Sybil attack, any particular node illegitimately claims for several identities. The Sybil node acts as original node and can introduce false packets into the network and disrupt the purpose of the network.

DoS Attack: In DoS Attack, the attacker captures the key processing request pattern and raises these requests frequently and blocks the service availability to others.

When the base station receives the packet from the *ith* node, it identifies the key number *kni* using the Equation (3.3) to decrypt the message *Emsg*. If the BS receives the packet from the node *i* in the time that falls in the same time interval τl at which the node *i* sends the packet, then the BS uses the counter value *ε Bi,τl*,; otherwise if the packet reaches the BS in the next time interval *τl+1,* it uses the previous counter value *ε Bi τ l-1*. The key k is selected using kni and the *Emsg i* is decrypted by *k* to obtain the payload.

If the key *k* cannot decrypt the received encrypted packet, it will be treated as an illegal packet. Then the base station tries to decrypt the received encrypted packet using the remaining valid keys. If the packet cannot be decrypted by any of the remaining valid keys, then the BS identifies the packet has been corrupted. If any one of the remaining valid keys decrypts the packet, then the BS verifies the timestamp *Ti.* If the packet is not a fresh packet, then the BS declares that this is a replay packet.

If the packet is a fresh one, then the BS declares that this is a Sybil attack and it broadcasts a message to invalidate the key number *kni* of the group where the Sybil node exists and sends a command to that node not to send any data for a configurable period of time. Each node in the network has to maintain another 8-bit counter *np* that counts the number of packets that are transmitted in the last *Q* seconds. The *Q* depends on the frequency of data transmission in the application. The source node and the BS increment *np* by one upon transmission and reception of a packet respectively. The attacker in the data path to the destination refuses to forward certain packets only. In such case, the value of np differs between the source node and the BS. Once the BS identifies the BS sends a command to that node not to send any data for a configurable period of time.

In SKSP, the choice for DoS attack is the re-keying request packet. An attacker can frequently send re-keying request and launch the DoS attack. In our approach, the re-keying request comes from the node only when any two consecutive keys are invalidated or the lifetime of the keys have been expired. Base stations will maintain this information for each node. So, if the rate of rekeying requests comes frequently, then base station can conclude for possible DoS attack and drop the packets from that node. The base station can also send a broadcast packet to stop the processing request from the attacking node for an interval.

6. Comparative Analysis of Proposed Protocol (SKSP) with other Security Solutions

We provide comparison from the perspective of memory requirement, communication and computation overhead and some other basic security parameters such as data integrity, confidentiality, availability, authentication, etc. Our approach provides re-keying, but LEDS regenerates keys only if the nodes are dislocated. SNEP provides security, but attack model is not discussed. The SKSP provides authentication support only during the re-keying process. We provide the computation overhead in terms of number of rounds required by one-way hash function to generate the keys and skipjack algorithm for encryption/decryption process. In our Proposed model, only the outside attackers are considered. The outside attackers launch DoS attack, packet replay attack, selective forwarding attack and Sybil inside the network.

7. Experimental Setup

To evaluate the performance of SKSP protocol, the Castalia simulator has been used.

7.1. Simulation Parameters

The simulation setting to evaluate the performance of SKSP, SPINSSNEP and LEDS is as given.

No. of Nodes 100
Area Size 100 m × 100 m
Simulation Time 300 secs
Traffic Source CBR
Radio Propagation Model Two-Ray Ground model MAC IEEE
Antenna Type Omni Antenna
Data rate 250 Kbps
Transmission Range 50 m
Initial Energy 50 Joules

7.2. Performance Metrics

In the simulation study, 100 nodes have been used with manual deployment in the area 100 × 100 m and initial energy is set to 50 J. We measure the performance using the following metrics:

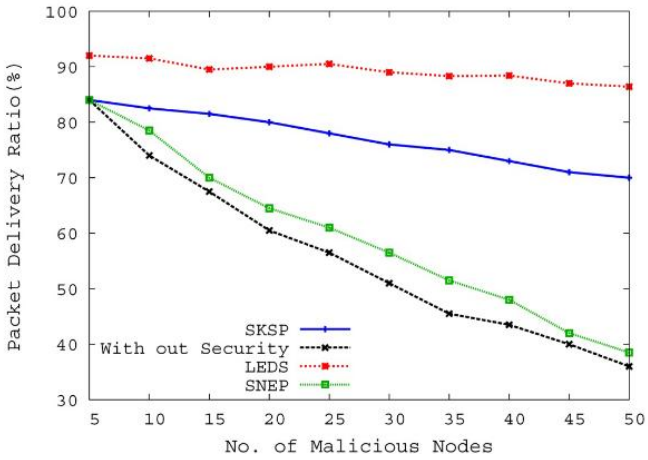
Packet Delivery Ratio: The total number of packets received is divided by the total number of packets sent from the source.

Network Availability: Availability can be measured by means of the lifetime of the secure wireless sensor network under various conditions.

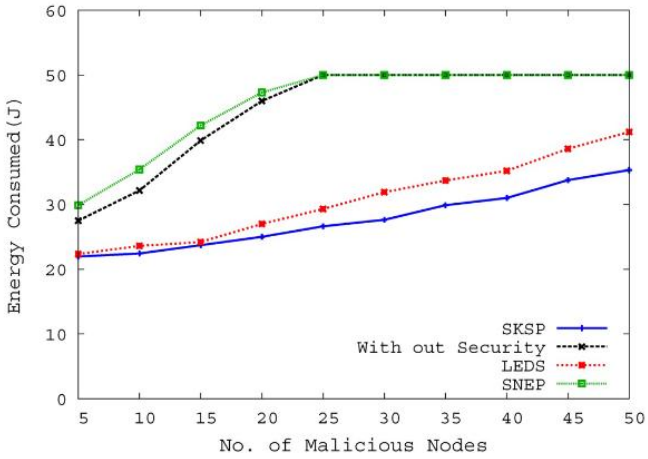
Energy Consumption: It refers the average energy consumed by the network for computation, transmission and reception. This parameter inversely reflects the lifetime of the network. If the energy consumption of the entire network is high, the network lifetime will be low.

8. Results and Discussions

This section presents the simulation and performance results of the proposed SKSP compared with other existing security mechanisms SPINSSNEP and LEDS. The results are measured in terms of packet delivery ratio, energy consumption and network availability by launching various attacks. Figure shows that SKSP maintains good packet delivery ratio of 70% even the number of malicious nodes increases up to 50. The increase in number of nodes will proportionally increase the number of false packets over the network. In SKSP security mechanism, the BS identifies the false messages and broadcasts the command not to forward the false message from the malicious nodes. The node that receives this command will simply discard the false messages, so that the normal traffic in the network is maintained which provides good delivery ratio.



Effect of Packet Delivery Ratio by increasing Malicious Node



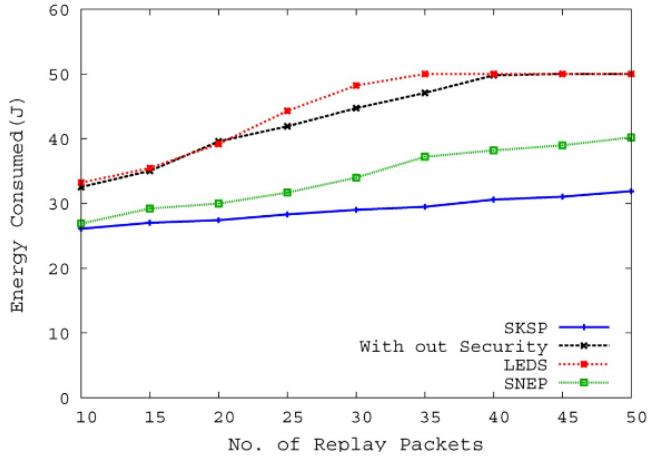
The Effect of DoS attack on network energy

In SNEP, all the security threats are identified and discarded at the BS. So the network is affected with high congestion because of the false messages transmission. Hence the SNEP drops more legitimate packets when the malicious nodes increase. But LEDS reduces the false message transmission and the valid packet dropping by using the en-route-filtering operations. So it achieves the best delivery ratio than all other approaches.

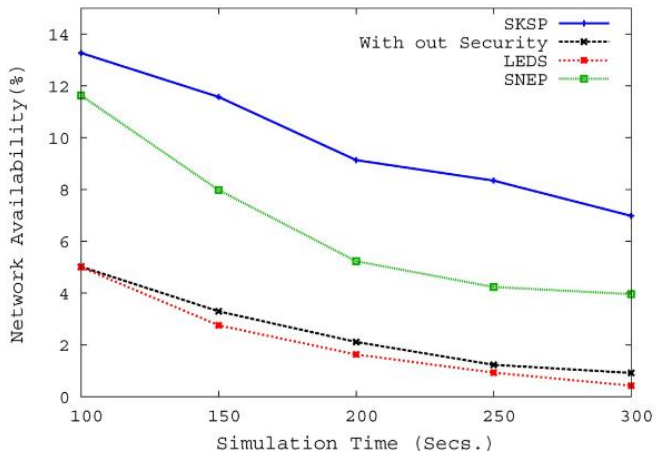
For the DoS attack simulation, we keep a constant attack rate of packets and calculate the average energy at the nodes for over a period of time. Figure shows the effect of DoS attack on network energy. SNEP does not identify the DoS attack. So it consumes greater energy due to DoS attack. In our approach, we stop processing the packets at nodes for some time interval, whenever the DoS attack is detected. In LEDS, DoS attack is prevented by using en-route filtering which needs extra energy than our approach.

Figure shows the effect of the replay attack on network energy when five attackers replay the packets at the rate of 10

to 50 copies per second. SNEP uses implicit counter maintained at both the ends to protect against the replay attack. In SNEP, if the replay packet is identified by the destination, it simply discards the replay packet and the remaining replay packets are still forwarded by the nodes to the destination. But in our mechanism, if the replay attack is identified, the BS sends a command not to forward the packets from the node where the replay attack is launched for a configurable period of time. This will reduce the energy consumption during that time period. Because of the communication overhead to detect the replay attack, SNEP consumes greater energy than the network without security mechanism. LEDS effectively controls the reply attack, but because of its high communication overhead it consumes greater energy than our approach.



Effect of Replay Packets on network Energy



Percentage of Network Availability over the time

The network availability can be improved by means of increasing the lifetime of the secure wireless sensor network under various conditions. Figure shows the percentage of Network Availability over the time. Because of the computation and communication overhead, SNEP drains the nodes energy faster than the without security mechanism. LEDS and SKSP effectively control the security threats, so the node availability in both the methods is maintained for a longer period. But in SKSP, the computation and communication cost is much lesser than LEDS, we achieved the higher network availability than LEDS.

9. Conclusion

The security solution in WSN depends upon the network size and the level of security that the application needs. In this paper we represents a security solution named SKSP which has been designed for small scale applications that require minimum level of security. The SKSP uses one-way hash function to dynamically generate the keys that avoid transmission of key during the runtime. In order to minimize the memory overhead, we have introduced grouping among the nodes in the network that maintains different sets of keys. The SKSP provides a method for re-keying the data encryption keys if any two consecutive keys are compromised by the attackers. By grouping the nodes, the communication overhead during re-keying is also reduced. The packet replay

attack, selective forwarding attack and DoS attack are the attacks that can be easily launched by the attackers without knowing any secret credentials of the network. The SKSP identifies the attacks such as Replay attack, Sybil attack and DoS attack. The SKSP protocol has been analyzed using the parameters such as network availability, packet delivery and network energy on replay and DoS attacks. In SKSP mechanism, the scalability can be still increased by introducing the Clustering concept in order to reduce the traffic and overhead to the Base Station.

References

[1] Ren, Kui, Lou, Wenjing, and Zhang, Yanchao, LEDS: Providing location aware end-to-end data security in wireless sensor networks. IEEE Transaction on Mobile Computing, 7 (5) (2008), 585 – 598.

[2] Rivest, Ronald L, The RC5 Encryption Algorithm. In The Second International Workshop on Fast Software Encryption (FSE) 1994e (1994),86-96.

[3] Rivest, R.L., Shamir, A., and Adleman, L., A method for obtaining digital signatures and public-key cryptosystems. In Communications of the ACM(New York, USA 1978), ACM, 120-126.

[4] Roy, S.D, Singh, S.A, Choudhury, S, and Debnath, N.C, Countering sinkhole and black hole attacks on sensor networks using Dynamic Trust Management. In Computers and Communications, ISCC 2008. IEEESymposium on (Marrakech 2008), IEEE, 537-542.

[5] Ruj, Sushmita, Nayak, Amiya, and Stojmenovic, Ivan, Pairwise and Triple Key Distribution in Wireless Sensor Networks with Applications. IEEE Transactions on Computers, 62 (11) (2013), 2224-2237.

[6] Sahingoz, Ozgur Koray, Large scale wireless sensor networks with multilevel dynamic key management scheme. Journal of Systems Architecture, 59 (9) (2013), 801-807.

[7] Saifan, Ramzi and Al-Jarrah, Omar, Novel Algorithm for Defending Path- Based Denial of Service Attacks in Sensor Networks. InternationalJournal of Distributed Sensor Networks, 2010 (2010), 9 pages.

[8] Sarigiannidis, Panagiotis, Karapistoli, Eirini, and Economides, Anastasios A., Detecting Sybil attacks in wireless sensor networks using UWB ranging-based information. Expert Systems with Applications, 42 (21)(2015), 7560-7572.

[9] Saxena, Himali, Ai, Chunyu, Valero, Marco, Li, Yingshu, and Beyah, Raheem, DSF - A Distributed Security Framework for Heterogeneous Wireless Sensor Networks. In Proceedings - IEEE Military Communications Conference MILCOM. (2010) (2010), IEEE, 1836 -1843.

