Chapter - VI

6.1 Introduction

The objective of this chapter is to provide a secure schema for energy streaming in WSN within the application and transport layer and to make the schema as energy efficient as possible. The process of achieving the objectives will be discussed through the chapter.

The key challenge in securing sensor networks is how to maximize the life of sensor nodes due to the fact, as it is not feasible to replace the batteries of thousands of sensor nodes. Therefore, computational operations of nodes and communication protocols must be made as efficient as possible in the energy consumption. Among internet protocols, data transmission protocols in application layer have much more importance in terms of energy, since the energy required for data transmission takes 60% of the total energy consumption of a wireless sensor network. The second objective of the research is to find a solution to minimize the number of data packet delivery to make the network energy consummated while adapting the security features as proposed.

Network energy plays a very important role in the decisions of choosing a security mechanism, thus the following issues should be taken in considerations while designing a security schema:

Untethered

The sensor nodes are not connected to any energy source. They have only a finite source of energy, which must be optimally used for processing and
communication. To make optimal use of energy, communication should be minimized as much as possible.

**Availability**

Sensor nodes may run out of battery power due to excess computation or communication and become unavailable. The requirement of security affects the operation of the network.

### 6.1.1 Energy Efficient Layer Attacks:

Attacks can be happened also on two layers of the WSN Stack layers. It can be summarized and their security solution approach in two layers with respect to WSN layers attack in the Table 6.1.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Attacks</th>
<th>Security Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transport Layer</td>
<td>De-Synchronization attack</td>
<td>Complex Puzzles</td>
</tr>
<tr>
<td></td>
<td>Flooding Attack</td>
<td></td>
</tr>
<tr>
<td>Network Layer</td>
<td>Sybil Attack</td>
<td>Authentication</td>
</tr>
<tr>
<td></td>
<td>Sinkhole Attack</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Wormhole</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hello flooding Attack</td>
<td></td>
</tr>
</tbody>
</table>

### 6.2 Proposed Security Schema

The security schema for WSN will be introduced. It will construct its security features within the application and transport layer and made the schema as energy efficient as possible. The proposed security schema consists of three security level:

- The first level is encrypting the packet data using Cryptography Encryption Method (CEM)
The second level is generating the log files within a specified path maintains for analyze the data value. The third level is Key distribution method for aggregating the data value.

- Message Confidential Code (MCC) using Key Management code technique.
- Distance path routing to avoid the high traffic loads.

**Design Assumptions**

Traffic flow follows a pattern determined by the application. Most of the traffic is assumed to be directed from the nodes to the base station. The reason for that assumption is derived from the ultimate goal of a WSN. Assumed that all the sensor use circuit format. Thus, in case if a node is captured the attacker is unable to extract data from the sensor especially keys.

Energy is the biggest constraint to wireless sensor capabilities. Assume that once sensor nodes are deployed in a sensor network, they cannot be easily replaced.

**Packet format modifications:**

The proposed secured packet format was constructed on the current packet format of communication across network shown in Figure 6.1.

![Figure 6.1 WSN normal packet format](image)
Chapter VI Secure Energy Efficient Schema for Wireless Transport
Layer Sensor Network

The modification can be summarized by the following points shown in Figure 6.2.

- Message encryption using Symmetric Encryption Method (SEM)
- Generating Message Confidential Codes (MCC) code using encrypted message, RTP header and UDP header. The generated code will be stored in MCC header field (5 bytes).

The new Pairwise Symmetric Key (PSK) field has been added to the proposed secured schema.

**PSK 1:** Symmetric key, before the network deployed. N is large storage requirements and to shared key very large Sensor size of the storage small.

Number of sensors = n

Storage of key = (1-n)

Each sensor to store exactly (1-n)/2 [symmetric key, before network deployed]

**PSK 2:** Each sensor need to store at least (n/2) [symmetric key],

If n is even n=2, 4, 6, 8 ……

If n is odd n=1, 3, 5, 7 ……

A new Symmetric Key (SK) field has been added to the proposed secured schema. The SK field holds 3 bits and it would contain the following values {000, 001, 010, 111}. The values of SK would notify the secured schema for which key password would be used to encrypt and decrypted the message. The SK field made the proposed security schema more computationally secured against known attacks other than any solutions for security.
6.2.1 Schema Design

The secure schema key management framework is needed to establish and update the cryptographic keys, which are used to secure the two ways of communication design. When designing a key management protocol it analyses the time sequence. The proposed time scheduler discard packet by the sender node from the beginning according to the following process as illustrated in figure 6.3

- Initialize the time of encryption, decryption as $T_{ENC}, T_{DEC}$
- Calculate time needed to secure a hop $T_{SEC} = T_{ENC} + T_{DEC}$
- Buffer two hop, hop A and hop B and calculate the time between them as $T_{AB}$
- If $T_{AB} > T_{SEC}$ then scheduler pass hop A to the proposed security schema else discard hop A and continue until the end of all hops.
Devising a key management schema for WSN is not trivial and in particular may not be successfully accomplished by simple adaptation of security solutions designed for wired networks. This is because of limited resources that a sensor node has such as energy lifetime, slow computation, small memory, and limited communication capabilities, as discussed in Chapter 1.

In this section, describes a key management schema Protocol which secures communication between sensor nodes and base station by considering vulnerabilities that is associated with WSN.

### 6.3.1 Matching Key Protocol

The Proposed solution for Match key rekeying also exploits the idea of key solution using a hash chain in order to achieve key secrecy. The network consists of sensor nodes deployed randomly multipath way over an adversarial region. The

---

**Figure 6.3** Security schema time state scheduler flow chart design
key pool is constructed of keys. The key are denoted by unique ID number. The protocol uses a hash chain, \( h^i(s_0) \), where \( s_0 \) is a key component pre-installed in the pre-deployment phase and \( i > 0 \) denotes the index of key, \( K_{Pi} \) denotes the index for key pool update phase. The key pool is constructed of keys. The key are denoted by unique ID number.

Initially \( k \) keys are distributed in each sensor from a key pool consisting of \( N \) keys. During sequence key establishment, any node \( h^i(s_0) \) randomly generates the secret key \( S \) and derives shares, \( S_{key1}, S_{key2} \ldots S_{keyn} \) from \( S \) such that \(|S1|, |S2| \ldots |Sn+1|=|S1|+|S2| \ldots |Sn+1|\), where + is the SUM \( (\text{Agg}) \) operation. Sender sends each of shares \( S_{keyi} \) through a different path. According to key SUM \( (\text{Agg}) \) operation scheme, two nodes a and b wanting to communicate securely, establish a unique sequence key \( Kab \). Nodes broadcast only their identifier. An algorithm, (for example algorithm 6.1) is build into the sensor, which enables a node a, to calculate sequence keys with another node b. If sequence key are ‘a’ and ‘b’, the sequence key between a and b is given by \( K_{ab}=\text{hash (P_{ab}||id_a||id_b)} \).

Algorithm 6.1: Sequence key establishment algorithm

```plaintext
if a(1)=a'(2), b(4)=b'(3) then
    Print “a, b is a different key”;
    Return empty
else
    y = ((m-m')^2+4(n-n')) mod q
    P = (m-m') + \frac{\sqrt{y}}{2} \text{ (mod q)}
    if \( P < k \) then
        \( K_{ab}=\text{hash (P_{ab}||id_a||id_b)} \)
        Return \( K_{AB} \)
    else
        if a(1)=a(1), b(2)=b(2)
            Print “a, b is a sequence key”
```
The key mapping from combinatorial designs \((X, A)\) to sensor networks is done in the following way. \(X\) is the pool of key identifiers. \(A \in A\) is a sensor node which contains a set of key identifiers from \(X\). The number of keys in each sensor node may be different, but normally is assumed to be equal.

**Algorithm 6.2: Hash Security Algorithm**

1. Node sink sends a broadcast of CCM message with the value of one way sequence number
   
   \[
   \text{ClusterToHash=}\text{oneway sequence number}
   \]
   
   // \(x\) is the set of nodes that received the message CCM

2. foreach \(i \in x\) do
   
   if \(\text{ClusterToHash}(i) > \text{ClusterToHash(CCM)}\)
   
   First sending (i) then
   
   Next cluster \(i \leftarrow \text{IDCCM};\)
   
   ClusterToHash \(i \leftarrow \text{ClusterToHash}(\text{CCM}) + \text{sequence value}\)
   
   // Node (i) receive the value of the ClusterToHash field in the message CCM
   
   \[
   \text{IDCCM} \leftarrow \text{IDi};
   \]
   
   // Node i updates the value
   
   ClusterToHash \(\leftarrow \text{ClusterToHash}\)
   
   First sending \(\leftarrow \text{false}\)

   End

Else

Node i discards the received message CCM

End;
In this Algorithm 6.2, the distance from the sink to each node is computed in node head. This protocol is started by the sink node sending, by means of an INSENS, the Node Configuration Message (NCM) to all network nodes. The NCM message contains two fields: ID and Packet to Hash. Where ID is Node identifier that started or retransmitted NCM message and Node to Hash in packet by which an NCH message has passed the function create a one way sequence number, the sequence value is started -1 at the sink. This forwards it to its neighbors. Each node, upon receiving the message NCM, Verifies if the value of Packet To Hash in the NCM message is 1315423912 less than value CTH that it has stored and if the value of first sending is true. Compare the one node to another node with the key value in the middle of the key list and continue the process until finding the name key and node ID in the list.

### 6.3.2 Key Setup phase for Encryption

In the Key distribution phase, a large key pool of K keys and their corresponding identities are generated. For each sensor within the sensor network, k keys are randomly drawn from the key pool. These k keys form a key ring for a sensor node. During the key discovery phase, each sensor node finds out which neighbors share a common key with itself by exchanging discovery messages. If two neighboring nodes share a common key then there is a secure link between two nodes. In the path-key establishment phase, a path key is assigned to the pairs of neighboring sensor nodes that do not share a common key but can be connected by two or more multihop secure links at the end of the shared-key discovery phase. In the random key distribution mechanism mentioned above, the Sande-tukey Techniques states that any pair of nodes that possesses at least one common key is O(n).

\[
F_t = \sum_{k=0}^{k=9} (f_{key} w_{\text{pass}k})
\]  

(6.1)
Let the probability be that any other node can overhear the encrypted message by a given key. If $f(t)$ is a discrete time signal with period $N$, Power $p$ is defined by the relation $w_0 = 2\Pi$, $T(w)$ is called the fundamental frequency $K$-key values.

$$w_n = n^{th} \text{ root of values}$$

$$w_n = e^{-2\pi i/n}$$

$e^{-2\pi}$ Series of sines and cosines, $2\pi/time(w)$ is called fundamental frequency the total computation form periodic sequence range $p = 0, 1, 2, 3, 4 \ldots n$

A frequently encountered data to base station and fitting a sum of exponentials of the form is given as

$$B = f(c) = A_1 e^{\lambda_1 x} + A_2 e^{\lambda_2 x} + \ldots A_n e^{\lambda_n x} \quad (6.2)$$

Therefore, assume that $n$ is the sum of all the line in the given range is equal to $1$ $f(c)$ c $(1, 5)$ Therefore, the percentage of key that need to merge is given by

$$F_t = \sum_{k=0}^{4} (f_{key} w_9^{pass}) + \sum_{k=5}^{7} (f_{key} w_8^{pass}) \quad (6.3)$$

This variable can make changes and add value even to components

$$F_p = \sum_{k=0}^{4} (f_{key} w_8^{pass}) \quad (6.4)$$

$$\sum_{k=0}^{4} (f_{key} + 5 w_9^{pass} + 5) \quad (6.5)$$

Computation of the total collection of Key
Let the function $f(t)$ alternative values and let $f_0, f_1, f_2, \ldots, f_7$ be a sequence of values of $f(t)$

$$F_p = \sum_{k=0}^{n} (f_{\text{key}} w^p_{\text{ask}})$$

(6.7)

### 6.3.3 Backward Reply Unique Key Exchange Modification

Let’s assume that the adversary has somehow extracted the current value of the unique exchange key, $U_{i,k}^K$. Head node determines the coordinate for each node $S$ before their deployment as $S$: uniqueID, new_mac () address=$(x, y)$. The keys are generated using hash table, so that key is used for generating the authority Key [A] for that parent node. The head node H first generates a secret Key H [SK] and computes the Diffie-Hellman component $G^{th}$. It then broadcasts Message, which includes the index $i$ of the next backward unique key.

The inclusion of the group key index $i$ in the first message enables each sensor node to check if it has the current value of the exchange modify key; if not, the node can request the head node to send the last key $h^i(k)$. Thus, the unique key rekeying protocol exchange as described in Algorithm 6.3.

The rule of backward reply unique key Modification

1. Parent [A] = id, new_mac ()
2. New_macAdd () = Arrays.CopyOf (mac, mac_length)

S: →keyset (K)

Set SourceID ()

Set Destination ()

Set IntermediateID ()
S: Keyset (K)

3. H-N: i, j

The novelty of this new scheme is that, no need for storing the key formula in a keyset because the key are generated randomly after deployment of nodes.

**Algorithm 6.3: Encrypt the data**

```plaintext
Compute TotalKey (EcnData, neighbour hashIDclr)
begin
itext IDclr
count
if i = 1 then
    Ki-2;
Count n
else
    Ki, F→Ki-1
end if
return Ki;
end.
```

6.3.4 **Security Algorithm**

In security design every sensor node has a Class Commend Letter like Key (CCLLK) at a time of deployments. Initially sensor nodes encrypt the sensed data apply the security Algorithm 6.4 , which makes the data transmission more secure, then send encrypted data to base station. The advantage of this technique is that it increases communication security and requiring very less energy when compared to other cryptography algorithms. After completing a current class, sink will generate a new class key using a Mathematical function (f) and current class letter like key and send to the corresponding gateway. The new CCLLK broadcast to its cluster's sensors by the base station, for data encryption of the new CCLLK. So in this
communication process CCLLK key has change dynamically for every class by the Sink.

**Table 6.2**

<table>
<thead>
<tr>
<th>Notations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_i$</td>
<td>Packet</td>
</tr>
<tr>
<td>$E()$</td>
<td>Encryption function</td>
</tr>
<tr>
<td>$SK$</td>
<td>Session key</td>
</tr>
<tr>
<td>$G_l, G_m, G_n$</td>
<td>Gateways ID</td>
</tr>
<tr>
<td>$D()$</td>
<td>Decryption function</td>
</tr>
<tr>
<td>$f()$</td>
<td>Pseudorandom function</td>
</tr>
<tr>
<td>$T_{GS}$</td>
<td>Logical time stamp (sequence - numbers) of the gateways</td>
</tr>
<tr>
<td>$T_{SG}$</td>
<td>Logical time stamp (sequence - numbers) of the Sink</td>
</tr>
<tr>
<td>$SK_n$</td>
<td>New session key</td>
</tr>
<tr>
<td>$SK_c$</td>
<td>Current session key</td>
</tr>
<tr>
<td>$X$</td>
<td>Random number</td>
</tr>
<tr>
<td>$J_j$</td>
<td>Concatenation operator</td>
</tr>
</tbody>
</table>

In each class, sink sends a verification key $CVK_n$, which is encrypted using the current class key $CK_c$. Gateway nodes broadcast the new class key to its cluster. Sensor nodes received broadcasts the verification key $VK_n$ and update their secret class $VK_n$ key. The propose algorithm is provide data confidentiality by using $CVK_n$ for all the subsequent data encryption and decryption during the class, and each sensor node encrypting data with $CVK_n$ also provide authentication.

**Algorithm 6.4: Class Commend Letter Like Key (CCLLK)**

1. Sensor to Gateway:- A Sensor node $Si$ encrypts the packet $Pi$ using current session key $SK$, which is built-in at the time of sensors deployments and send to it's a local gateway $Gi$. 

---


75 | P a g e
2. The following actions are performed by the sink on receiving packet from the gateway:

(i) For a credible time stamp sink decrypts the encrypted packets using the current session key, DS
    $$\text{if } (T_{GS} < T_{SG})$$.
(ii) Verify gateways IDs in the packets.

3. On expiry of current session, sink increments the value of TSG by 1, and generate the new session key using the pseudorandom function (f) and current session key. The new session key is a function of current session and x.

New class verification key CVKn = f (CKc, x) where x is letter like symbol.

4. Class command key is updated for the next class. CCLLK key is updated as follows.

(i) Sink encrypt the new class key (CKn) using the current class key (CKc) and send to the corresponding base station,
(ii) Base station broadcast the new verification class key (NVKn) in its own head node,
(iii) Sensor nodes update its CCLL keys, with the new class key.

7. End

Changing encryption key in each session is helped to ensure data freshness in the WSGN; besides, it also provides the confidentiality of the transmitted data by preventing the use of same key in every session. Ensuring data freshness avoids the replay attack in the network. During the data communication each gateway node appends its GID and logical time stamp to the packet to ensure data freshness and integrity. During receiving a packet, sink decrypt the data using SKn and find out time stamp with associates GID on the message, and verify the data authentication, then obtain the original message Pi. If it is altered or replayed, then sink discard the data or send a retransmission request to the corresponding gateway node. During
aggregation of data gateway node appends its GID before forwarding data to sink to help the sink in locating the origin of the data and reduce the search time required to find the originating cluster node.

6.4 Summary

In this chapter, an energy efficient security schema based on transport layer designed for WSN was proposed. The significance of the proposal is four level-fold: (i) the first level is encrypting the packet data using Cryptography Encryption Method (CEM) (ii) The second level is generating the log files within a specified path maintains for analyzing the data value. (iii) The third level is Key distribution method for aggregating the data value. Message Confidential Code (MCC) using Key Management code technique. (iv) The fourth level Distance path routing to avoid the high traffic loads.

The chapter proposed the key management schema Protocol which secures communication between sensor nodes and base station by considering vulnerabilities that is associated with WSN. Schema framework was described in three scenarios, depending on the adversary capability to affect the gathering result, as follows (i) Key Setup (ii) Matching Key (iii) key Exchange Modification (iv) security algorithm Finally, the chapter applied the methodology used in chapter 2 in order to analyze the security algorithm.