International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2019): 7.583

Light Weight WSN Authentication Protocol Suite

Rahul K Drolia

MTech Project Report

Abstract: ***Wireless Sensor Networks (WSNs) consists of low cost, light weight, low processing power, shrinked life sensor nodes known as motes with an ability to communicate with each other over short ranges. The network formed out of sensor motes is controlled by a Base Station which has far more processing power and life as compared to sensor nodes. As WSNs are used for many practical purposes including matters related to na-tional security, it becomes imperative to ensure that the underlying communication is secured. However, the scarcity of resources in a sensor mote makes it highly challenging to ensure secured communication and long life of a sensor network at the same time. Several protocols have been proposed for authentication mechanisms in WSNs but most of them either are weak or have resulted in large energy expenditure. We through this paper have proposed a light weight protocol suite which has tried to cover almost all the authentication requirements of WSN. The proposed authentication protocol suite has been made light weighted by using symmetric encryption/decryption in most of the situations and the security has been made comparable to that of asymmetric encryption/decryption by introducing randomization in symmetric cryp-tosystem. We have proposed protocols for node authentication, broadcast authentication, data transfer and network monitoring using a combination of public key and symmetric cryptography. Previous works have been used for the calculation and comparison of energy expenditures in detail.***

Keywords: Symmetric Key Cryptography, Randomisation, Wireless Sensor Node, Authentication Protocols, Energy Saving

1. Introduction

Wireless Sensor Network is a network of sensors laid down in a particular area for a speci c purpose. Of late, it has found its utility in variety of real life appli-cations which range from simple temperature/pressure sensing to complex IOT network monitoring. Military applications of WSNs certainly does not need special mention.Ability of sensors to be easily deployed in in-hospitable terrain, communicate with each other and form a network gives an edge to the usage of WSNs for several applications (specially in Military parlance). Sensors are low power, low cost devices with very lim-ited resources like memory, battery, processing power etc. These are tiny devices which can communicate in short distances only. A sensor node typically contains a power unit, a sensing unit, a processing unit, a storage unit and a wireless transmitter / receiver.

Even though WSNs has benefits, limited resources of a sensor node brings out a lot of challenges in es-tablishing a secure communication in the network and use the data sensed by nodes. This is because more the network communication is made secure more will the usage of sensor resources which are themselves very scarce. Consequently, the lifetime of sensor net-work is a ected and as a result there has to be trade o between the security requirements and sensors life. In any given network of wireless sensors, there is sup-posed to be one network controller (known as Base Station) which is far more trustworthy, has more com-puting power and manages the entire network. As it is supposed to manage the network, it is the one which is generally responsible for setting up of the cryp-tographic primitives of the network, establish secure communication channel between sensor nodes and be-tween individual nodes and itself, raising queries to the nodes and processing after receiving replies from those nodes.

Several Node Authentication, Broadcast Authentica-tion, Message Transfer and Key Management Proto-cols have been proposed till date; however, all have their own shortcomings. Moreover, in general, all the protocols focus individually on one authenti-cation requirement and many amongst those have been proposed without discussing the energy expen-diture/consumption which is indeed the basic require-ment of any protocol in WSNs. We not only have to look for an increased security but also security proto-col needs to be light-weight. In this proposed work of ours we have attempted to ful l most of the security requirements of WSN through a suite of light weight protocols. We have also done energy calculation and comparison corresponding to sub-protocols and found that our proposal gives considerable savings in terms of energy as compared with other existing work.

The remaining part of the report is organized as follows. Section 2 brings out the Literature Survey conducted by us, in section 3 we have proposed our protocol suite. In section 4 we have calculated and compared energy expenditure. Section 5 discusses Fu-ture Work and nally Section 6 concludes this report.

2. Literature Survey

We surveyed several existing protocols in WSNs which includes protocols of Node Authentication, Broadcast Authentication, User Authentication, Data Transfer etc. and found that all have their own issues. Moreover, in general all the protocols focus individually on one authentication requirement. Our literature sur-vey is focussed on the requirement aspect of secured communication with increased life of sensor node.

In [1] pure MAC scheme has been used to provide data integrity and authentication of communication entities. Disadvantage of this scheme is that the maintenance overhead is high in pure MAC based schemes. Tesla [2] and it's several modi cations have been proposed for broadcast authentication of messages in WSN, however, all of the variants su er from delayed authentication and consequently prone to DOS attack. Localized Encryption and Authentication Protocol (LEAP) [3] in WSN offers multiple keying mechanisms to provide con dentiality and authentication. It comprises of Telsa, one way key chain authentication, key revocation and key refresh-ing. However drawback is again delayed authentica-tion due to Tesla. Secured Network Encryption Protocol (SNEP) in which each node shares a pair of key with BS and other required keys are obtained from master key available with the BS is a useful protocol to provide con dentiality and integrity. However, it is not able to handle node capture and DOS attacks e ectively.Authors of [4] talk about the authentica-tion scheme which is based on key pre distribution however there are two cases and corresponding issues with the scheme discussed. First if in case a single session key is used for the entire network then cap-ture of one node reveals the entire network. Moreover in case each node needs to store a shared secret key corresponding to other node then there has to be n-1 entries in the database of each node which is again a great storage requirement. In [13] A Dynamic User Authentication Scheme for Wireless Sensor Networks has been proposed wherein there are some aws. It cannot provide resistance against replay and forged attacks. It also su ers from stolen veri er attack; both gateway and login-node have the lookup table which contains secret information about registered users. Passwords may be exposed by any of the sensor nodes and the user is unable to alter the password.

In [5] authors talk about multihop node authenti-cation using ECDH wherein they have given four di erent protocols for the same purpose but they do not mention as to how BS understands the ID of node requesting authentication. In [6] authors propose an authentication model that aims at reducing overhead for the re-authentication of sensor nodes. It works only well when the node is in direct range with the base station also the initial authentication phase suf-fers from internal attacks.

In [7][8][9] authors have employed ECC to perform security functions in WSN however each node is sup-posed to be in direct communication with certifying authority and moreover nothing is spoken about node re-authentication. In [10] multi user authentication scheme has been de ned where bloom lter has been used to store user IDs and public keys, however the drawback of bloom lter is that it can be forged and can't prevent DoS attacks.

Several authentication protocols using hash chain have also been de ned in WSN. In [11] node authentica-tion and key establishment for new nodes have been proposed by including node boot strapping time and it's identity in the procedure, however the demerit is that it assumes that each sensor node can sustain time interval before it can be compromised.In [12] authors have proposed authentication scheme again based on hash chain and ECC which is supposedly simple and supports new node addition as well but it has been found to be vulnerable to replay attack and node masquerading attack.

3. Our Proposal

In this report, we propose a set of lightweight authen-tication, data transfer and key management protocols using symmetric/asymmetric key encryption and mes-sage authentication code (MAC) to be used for secure communication in wireless sensor networks (WSN). The proposed protocols address the security issues and gives considerable savings in terms of energy as com-pared with other existing protocols. We rst propose the protocols and then compare the energy require-ment of our proposed protocol with that of existing protocols.

3.1 Notations

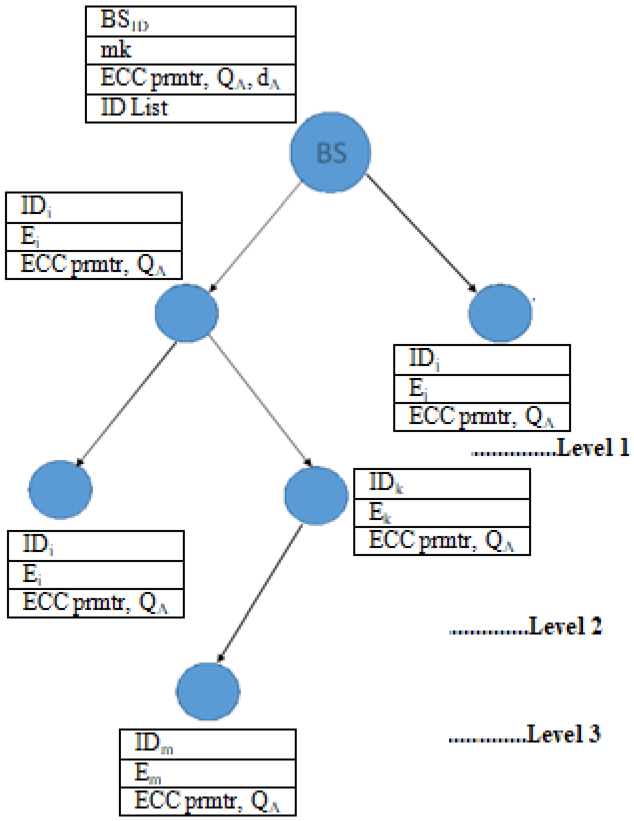
The notations and formulas used in protocols are listed in Table I.

Table 1: List of Notations Used in the Proposed Protocols

Symbol	Meaning
I, j, …t	Nodes
BS	Base Station
ID_i	ID of node i
E_i	$Enc_{mk}[ID_i]$
$N_i, N_{j...}N_{BS}$	Nonces
mk	Master Secret Key with BS
Q_A	Indian Public Key of BS
d_A	Indian Private Key of BS
k_g^n	Group Key in round n
K_{BSi}	Shared secret key between BS and node i
K_{ij}	Shared secret key between node i and node j
MAC	Message Authentication Code
AN	Aggregator Node
DN	Data Client Nodes

3.2 Assumptions and Initialization

1) BS and nodes are initialized and algorithm to form a tree is run.
2) The sensor network is arranged in a tree structure where BS is the root of the tree and sensor nodes are arranged in a hierarchical structure [Fig. 1]. There exists a control room responsible to control the entire network and moreover it is assumed that BS cannot be compromised.



3) The initial tree formed is unauthenticated and the authentication protocol is run to make the entire tree authenticated. After that group key is distributed and broadcast authentication protocol is run.
4) Control room loads BS with the list of IDs of all the sensor nodes who possibly may be the part of network. It maintains the same ECC parameters (Curve, G, n) in BS and all sensor nodes.Initial keys for BS Q_A; d_A are determined and Q_A is preloaded in each sensor node as well.
5) Control room pre-loads a 128-bit master secret key mk and an initial group key k_g^1 in BS. It also pre-con gures each sensor node i with its corresponding ID ID_i and ID_i encrypted using mk, E_i.
6) After the authenticated tree formation, aggregator nodes of the tree are decided by the base station in consultation with control room which may vary as per the application query. Each node gets the application speci c aggregator node corresponding to it and each aggregator node also gets the IDs of all the nodes which are supposed to relay data to it. Each node establishes a shared secret key with it's aggregator node.

3.3 Node Authentication

In the proposed protocol, after an unauthenticated network of nodes is formed, each sensor node sends an authentication request to BS through its parent node.Parent node is supposed to relay the request to BS but for that parent node itself should be authen-ticated. In case parent is not authenticated then the authentication request will have to wait at the parent node till it gets itself authenticated. Consequently the rst set of authentication request is from level 1 nodes which is in direct range of BS.

In order to create an authentication request, a node i will generate a 16B random nonce N_i and use it along with its 16B encrypted ID E_i to create a one-time key

$$K_i = N_i \oplus E_i$$

i will then forward authentication request message consisting of its 7B ID ID_i and N_i. It also calculates the *32B M AC* of message using the one-time key and appends it to the message. Format of the authentication request message generated by *i* is

$$< header > ||ID_i||N_i||MAC_{K_i}[< header > ||ID_i||N_i]$$

Header Encoding
Table 1

First three bits	Message Type	Fourth Bit	Last Four Bits
000	Broadcast Reply	Message Size	
001	Authentication Request	Originator Bit	Message Size
010	Relay Message	Relay Bit	Message Size
011	Authentication Reply	Destination Bit	Message Size
100	Broadcast	Message Size	
101	Group Key	Message Size	
111	Group Key Ack	Message Size	

Message Size: In multiple of 16 B

Encoding of <*header*> is as per Table 1. A node requesting authentication will set its originator bit to '0'. In case i is a level 1 node, last 4 bits of header will be 0100 to represent a size of 64B. Please note that, if required, message is padded with 0^0s to make its size in multiple of 16*B*.

BS on receiving the message will first check the existence of ID_i in its ID list and if found, it will use N_i from message and *mk* from its database to calculate

$$K_i^{/} = Enc_{mk}[ID_i] \oplus N_i$$

BS will use $K_i^{/}$ to verify M AC on the received mes-sage. 0 originator bit indicates that i is in direct range of BS and that BS is supposed to be $i^{/}s$ parent. Post veri cation of MAC, BS generates a shared secret key between itself and *i* and forwards it to *i* using the similar approach which was used by *i* to forward its authentication request to BS.

If *i* is at a level other than level 1 then it will forward it's request to nearest available authenticated node. *M AC* is created and verified at each intermediate node using the shared secret key between the intermediate nodes which they already have by virtue of them being authenticated before *i*. If *i* forwards its authentication request to *j* in the above given format, j from header understands that i is making an authentication request. *j* will add its ID ID_j and a new header to the message with message type as 'Relay Message', Relay bit set to 1 and message size 128B. It then calculates M AC of entire message using the shared secret key K_{jk} between itself and its parent node k and forwards the message along with *M AC* to *k*.

k reads the header to understand that the message type is 'relay'. It then veri es the *M AC* using K_{jk} which exists in its database as well and since 'relay bit' is '1', does no change to message. If the *M AC* gets verified then it simply replaces the previous *M AC* with new *M AC* calculated over the message using shared secret key between itself and its parent. This approach of message relay is followed till the message reaches BS.

BS on receiving message initially verifies the *M AC* using shared key between itself and level 1 node. Once verified, it verifies the *M AC* calculated by i using the same approach as stated before. It computes the shared key between itself and node *i*, BS_i and also between node *i* and node *j*, K_{ij}. Both the keys BS_i, K_{ij} are then relayed to *i* and K_{ij} is relayed to j as well.

Please note that after authenticating each node BS up-dates its database of authenticated nodes with node ID, node's parent, level and shared secret between it and BS. Also it updates the child Node ID of the par-ent node in consideration. Similarly, each sensor node also updates its own database.

In the following we propose a light weight sensor node authentication protocol. We have explained the case of authentication requesting node *i* being in direct range of BS or having several nodes *j; k...t* in between enroute to BS in the same protocol.

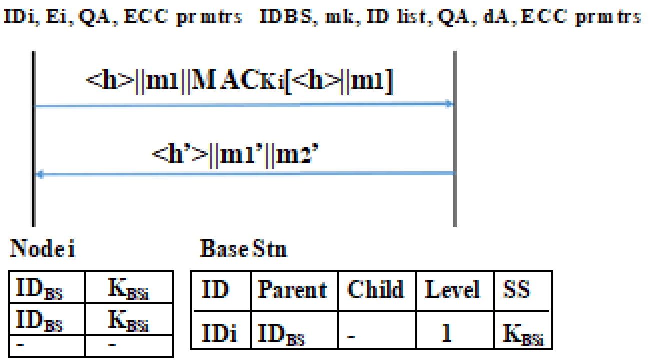
3.3.1 Light Weight WSN Authentication Pro-tocol (LWWAP)

The sequence of messages for LWWAP protocol is as shown in Fig. 2, 3 and 4 and the description of the messages is as given below.

1) Node i → BS/j : < header > $||m_1||$M AC_{Ki} [< header > $||m_1||$

where < header >= 00100100

$m_1 = ID_i||N_i$



<h> : 00100100
<h'> : 01100110
m1 : $ID_i||N_i$
m1' : $ID_{BS}||\ ID_i||N'_{BS}||N''_{BS}||Enc_{K''BS}[K_{BSi}]$
m2' : $MAC_{K'BS}$[<h'>$||m_1$']

Authentication Request and Reply: Level 1 Nodes
Figure 2

2) j → k :

<$header$ 1 > $||ID_j||m_i||M\ AC_{Kjk}$ [<$header$ 1 > $||ID_j\ ||m_i]\ ||ID_j$

Where < $header$ 1> = 01011000

m_i = message received from i

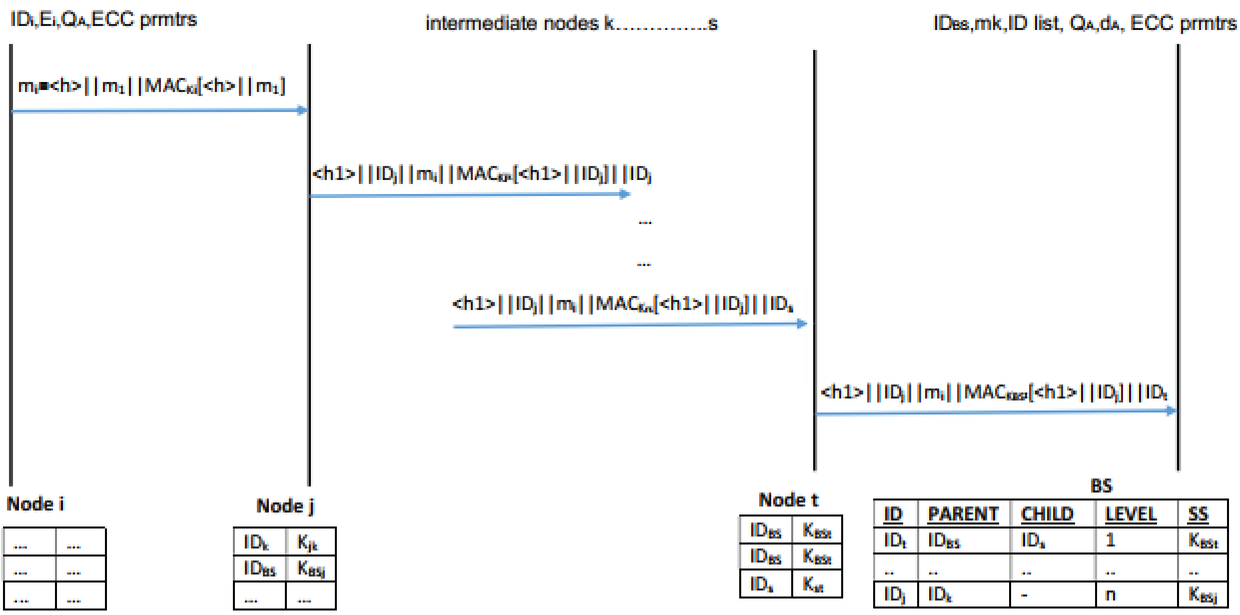
j adds its ID and a new header in the beginning of message.

It computes $M\ AC$ using K_{jk} and appends to the extended message.

It adds ID_j in the end of the entire message to indi-cate that it is the originator of the relay message.

3) k → l ! m.....→ t → BS: < $header$1 > $||ID_j||m_i||M\ AC_{KBSt}$ [< header1 > $||ID_j||m_i]\ ||ID_t$

k on receiving the 'relay' message in (2) veri es the $M\ AC$ using K_{jk} available in its database. On reading header's fourth bit as 1 does no addition to message. Changes ID_j in the end of the message to ID_k. Calculates and adds the M AC on remaining part using K_{kl}. Forwards the changed message to l. Message gets relayed till BS through other nodes using similar approach.



<h> : 00100100
<h1> : 01011000
m1 : $ID_i||N_i$

Authentication Request from Higher Level Nodes
Figure 3

4) At BS:

In case of (3), check ID ID_t at message end. Remove ID_t and verify $M\ AC$ on remaining received message using BS_t

If verified, using m_i or in case of (1), do following:

- Search ID_i in its ID-List
- if found, compute $K_i' = Enc_{mk}[ID_i] \oplus N_i$
- Verify $M\ AC$ on 00100100 $||ID_i||N_i$ using K_i'

International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2019): 7.583

- Store ID_i and corresponding data in authenticated ID table
- Generate nonces N'_{BS}; N''_{BS} and shared secret K_{BSi} between BS and *i*. Also generate K_{ij}, shared secret between *i* and *j* in case of (3).
- Create one time keys $K'_{BS} = Enc_{mk}[ID_i] \oplus N'_{BS}$ and $K''_{BS} = Enc_{mk}[ID_i] \oplus N''_{BS}$ in case of (1)

5) $BS \rightarrow i$: < *header* > $||m_1||m_2$ where <*header*>= 01100110

$m_1 = ID_{BS}||ID_i||N'_{BS}||N''_{BS}||Enc_{K''_{BS}}$

$m_2 = MAC_{K'_{BS}}[< header > ||m_1]$

Destination bit is 0 in case reply is directly for *i* without any relay in between.

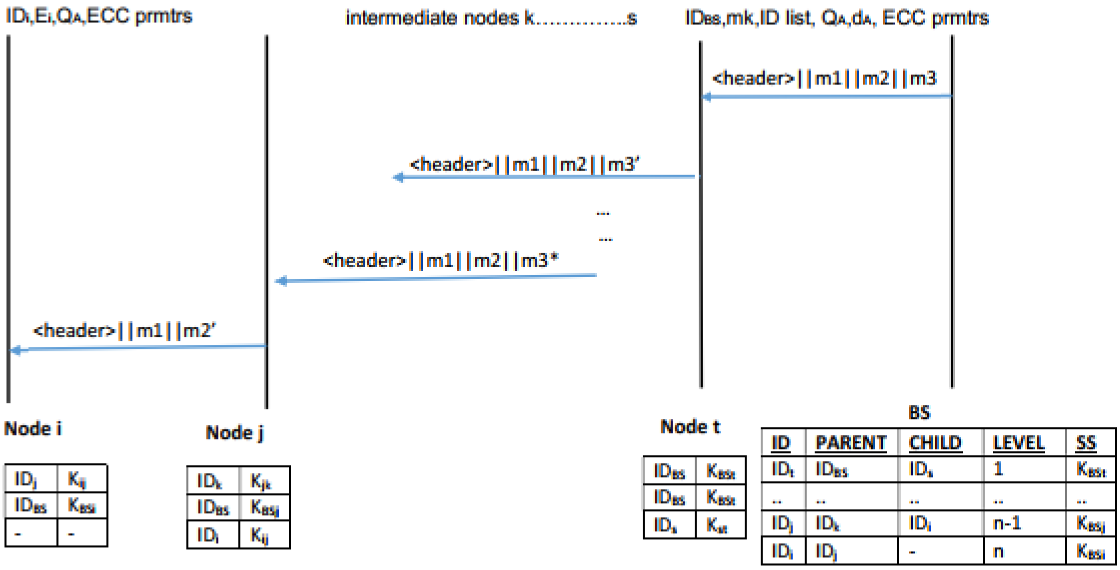
6) At *i*:

i. $K' = E_i \oplus N_{BS}'$

ii. $K'' = E_i \oplus N_{BS}''$

iii. Verify MAC on m_1 using K'. If verified decrypt $Enc_{K''_{BS}}[K_{BSi}]$ using K''

iv. Store ID_{BS} as parent ID and BS ID and K_{BS} as shared key.



<header>: 01111001

$m_1 = ID_{BS}||ID_j||ID_i||N'_{BS}||N''_{BS}||Enc_{K''_{BS}}[K_{BS_i}, K_{ij}]$

$m_2 = MAC_{K'_{BS}}[< header > ||m_1]||Enc_{K_{BS_j}}[K_{ij}]$

$m_3 = MAC_{K_{BS_t}}[m_1||m_2]||ID_{BS}$

$m'_3 = MAC_{K_{st}}[m_1||m_2]||ID_t$

$m_{3*} = MAC_{K_{jk}}[m_1||m_2]||ID_k$

$m'_2 = MAC_{K'_{BS}}[< header > ||m_1]$

Authentication Reply from BS for Higher Level Nodes
Figure 4

in case of (3)

7) $BS \rightarrow t$: < *header*> $||m_1||m_2||m_3$ where <*header* >= 01111001

$m_1 = ID_{BS}||ID_j||ID_i||N'_{BS}|| N''_{BS}|| Enc_{K''_{BS}}$

$m_2 = MAC_{K'_{BS}}[< header > ||m_1||Enc_{KBSj}[K_{ij}]$

$m_3 = MAC_{KBSt}[m_1||m_2] ||ID_{BS}$

Destination bit of header is 1 which indicates that the message needs to be relayed. IDs of m_1 represents originator, last hop and destination IDs in sequence. ID in m_3 represents the originator of relay.

t will verify MAC on $[m_1cm_2]$ using BS_t.

It then compares its ID with last hop ID in m_1, since both are not same it will replace m_3 with m'_3 where $m'_3 = MAC_{Kst}[m_1||m_2] ||ID_t$. It forwards < header > $||m_1||m_2||m'_3$ to s.

8) t → s →......→ j: <*header* > $||m_1||m_2||m_{3*}$ where m_1 and m_2 are same as before. $m_{3*}=MAC_{Kjk}[m_1||m_2] ||ID_k$. *j* verifies the MAC using K_{jk}. Since last hop ID is ID_j it decrypts $Enc_{KBSj}[K_{ij}]$ of m_2 using K_{BSj} from its database. Stores ID_i as child and K_{ij} as corresponding shared key.

9) j → i: <*header*> $||m_1||m_2$ where m_1 is same as before, however m_2 is reduced to $m'_2 = MAC_{K'_{BS}}[< header > ||m_1]$. Steps of (6) are followed to get K_{ij} and KBS_i

3.4 Group Key Distribution and Net-work Monitoring

Once a network gets initially authenticated, BS is supposed to distribute a common group key k_g^n to all the nodes. This common group key is used by BS to carry out a broadcast using broadcast protocol given in section 3.5. It is also supposed to monitor the network and look out for any unwanted/unaccepted changes in the network and if there is any change in the network then the group key needs to be changed for the entire network. We have considered two as-pects to look out for:

i. Node dies due to exhaustion of battery life.
ii. Node clone: It is an attack situation wherein a node is captured and it's credentials are copied into several other nodes. When cloned nodes are placed in network, they communicate with BS as authenti-cated nodes using the credentials of the node which got captured.

For distribution of k_g^n BS will use its authenticated node table and generate message of type <header>||ID_i||K_{BSi} [k_g^n] ||ID_j||K_{BSj} [k_g^n] ||…||ID_t||K_{BSt} [k_g^n] in such a way that IDs of nodes of higher level always come after those of lower level. Each node i will extract its ID ID_i and encrypted group key K_{BSi} [k_g^n] from the message and relay remaining message down the tree.It will then decrypt K_{BSi} [k_g^n] using K_{BSi} from its database.

BS monitors a network by periodically broadcast-ing a ID query message to the network and monitoring the reply from each node. The broadcast message is sent using LWWBA protocol of section 3.5. For any query message, each node is supposed to reply back. BS will monitor the replies and if a reply doesn't come from a node, it considers the node to be dead. More-over, if reply comes from more than one node with same ID, that ID is considered to be cloned. In both cases, BS will delete the IDs of a ected nodes from it database and generate a new group key. This new group key will then be forwarded to the nodes in a way that only the nodes which have not been deleted from BS database gets it. In the following we bring Group Key Distribution and Network Monitor Proto-col (NMGKC). We have considered the tree given in Fig 1 to discuss the protocol

3.4.1 Group Key Distribution and Network Monitor Protocol (GKDNM)

Group Key Distribution

1) BS → i=j:
<header > ||ID_i||Enc_{KBSi} [k_g^1]ID_j||Enc_{KBSj} [k_g^1] ||ID_l|| Enc_{KBSl} [k_g^1] ||ID_k||Enc_{KBSk} [k_g^1] ||ID_m||Enc_{KBSm} [k_g^1]
where <header>= 10101000

2) At *i/j*:
Decrypt Dec_{KBSi} [Enc_{KBSi} [k_g^1]] to get k_g^1 and store it in database.

3) *i* → *k/*l :
<header> ||ID_j||Enc_{KBSj} [k_g^1] ||ID_l||Enc_{KBSl} [k_g^1] ||ID_k||Enc_{KBSk} [k_g^1] ||ID_m||Enc_{KBSm} [k_g^1]

4) At *k/l* :
Repeat step (2)

5) k → m :
<header> ||ID_j||Enc_{KBSj} [k_g^1] ||ID_l||Enc_{KBSl} [k_g^1] ||ID_l||Enc_{KBSk} [k_g^1] ||ID_m||Enc_{KBS0} [k_g^1]

Repeat step (2)

6) Each node i → BS: 11100001; ID_i

Network Monitoring

7) *BS → nodes:* < *header1* >< m' ; ∏ > where

$$m' = Enc_{kg}1\,[m]$$
$$\prod = Enc_{dA}\,[m']$$

$$<header1> = 10000110$$

m is an ID-Query message and nodes verify and de-crypt the message as discussed in LWWBA.

8)Node I → BS : 00000010 || < ID_i, Enc_{KBSi} [ID_i] >

Each node will send it's ID and ID encrypted using shared secret between itself and BS.

8) BS will first verify IDs using shared secret. Post veri cation if any ID comes to BS more than once or if any ID from it's authenticated list doesn't appear, BS simply deletes those IDs from it's authenticated ID table.

9)BS generates new group key $k_g^2 = Enc_{mk}[k_g^1]$.
Steps (1) - (5) are repeated. Deleted IDs do not form the part of message. For instance if *j* is deleted by BS then BS→ i/j: 10100110||ID_i||Enc_{KBSi} [k_g^1] ||ID_l|| Enc_{KBSl} [k_g^1] ||ID_k||Enc_{KBSk} [k_g^1] ||ID_m||Enc_{KBSm} [k_g^1]

3.5 Broadcast Authentication

Once the authenticated network of nodes is formed, the network is used by network users to broadcast messages through BS. For this purpose broadcast au-thentication is required to ensure that the messages broadcast by BS reaches sensor nodes without any manipulation. We propose the usage of Elliptic Curve Digital Signature Algorithm as it gives signi cant en-ergy saving in comparison to the PKC-DSA scheme. This can be simply understood from the fact that in order to achieve 80 bits security the public key size required in ECDSA is 160 bits whereas in PKC it is 1024 bits. So in WSN where we are dealing with energy constrained sensor nodes it is better to work with ECDSA to increase the longevity of nodes.Energy comparison is well explained in [15] and [16].

As assumed earlier, BS has it's initial set of pub-lic/private key pair (Q_A; d_A) and the public key of BS Q_A is pre-con gured in all the nodes of the net-work. Also the ECC parameters have already been agreed upon earlier. BS will first encrypt the broadcast message m using group key k_g and then sign it using it's private key d_A. It will broadcast encrypted message along with signature on it to the nodes.Message can be anything including query messages. Each node will rst verify the signature using the BS public key and then decrypt the message using group key. BS will change it's public/private key pair after every broadcast and will send the new public key encrypted once again by group key and signed by old private key. In the following we present the Light Weight WSN Broadcast Authentication protocol (LWWBA)

3.5.1 Light Weight WSN Broadcast Authen-tication Protocol (LWWBA)

1) BS → nodes: < *header* >< *m'* ; > where ∏

$$m' = Enc_{kg}1\,[m]$$
$$\prod = Enc_{dA}\,[m']$$

First three bits of header is 000 representing broadcast and last five bits is for message size which in this case is 64B

BS encrypts *m* using k_g^1 and broadcasts along with signature on it.

2) Each node will rst verify the signature using method given in [14] and if the signature is verified extract the

message $m = Dec_{kg}1$ [m']. Acknowledgement message specific to the broadcast query is sent back to BS

3) BS randomly chooses a d'_A in [1; n-1] and computes corresponding public key $Q'_A = d'_A \times G$.

4) BS → *nodes*: < *header* >< m', ∏> where

$$m' = Enc_{kg}1\ [Q'_A]$$

$$\prod = Enc_{dA}\ [m']$$

3.6 Data Transfer from Nodes to BS

The data gathered by nodes should be forwarded to BS in a secured manner. Forwarding of data can be done either in aggregated manner or non-aggregated manner. We will explain both the concepts one-by-one.

3.6.1 Data forwarding without aggregation

In this case each node forwards it's data to BS as it is without any changes along with it's ID.In order to make the data transmission secure, a node encrypts the data with the shared secret key between BS and itself which already exists in it's database after initial node authentication. Since the implementation of WSN is over tiny db, BS can make SQL queries to the nodes for e.g. SELECT humidity FROM SEN-SORS. BS makes a DATA-QUERY broadcast using the broadcast protocol and nodes simply replies back.

3.6.2 Data forwarding with aggregation

Data Aggregation is the method of getting data from sensors wherein data from each individual sensor doesn't reach the BS as in the previous case; rather data from sensors reaches BS in summarized form.This reduces the data size and also the number of forwards from network to BS. Thus, it's an e ective way to re-duce communication and bandwidth overheads in a resource constrained WSN.In order to get aggregated data the BS needs to issue queries to the network with the condition. For e.g. SELECT max (temperature) FROM SENSORS or SELECT avg (humidity) FROM SENSORS WHERE ID = 5.Sensor nodes then for-ward their data to the application speci c aggregator node. As already assumed aggregator nodes and their corresponding data nodes are already decided and con-veyed by BS.If the aggregator node of a given node is same as its parent or child, then there already exists a shared secret between them in the node's database. In other cases, nodes do ECDH as mentioned in the proposed protocol given below. In the following we propose a protocol for data forwarding with aggrega-tion in WSN.

Aggregated Data Transmission in WSN (ADTW)

1) $AN :< Q_{AN}\ ; d_{AN} >$

AN uses the ECC curve parameters to generate public/private key pair.

2) $AN_i \rightarrow DN_i$: Q_{ANi}

As the tree is already authenticated, it then simply forwards its generated public key to nodes cor-responding to the query received from the BS.

3) $DN_i \rightarrow AN_i : Q_{DNi}$

DNs on receiving the public key from AN will generate their own public key/private key pair using the same curve parameters and forward their public key AN.

4) AN: $k_{ss} = Q_{DNi}.d_{ANi}$.

DN: $k_{ss} = Q_{ANi}.\ d_{DNi}$.

AN and each DN will calculate the shared key k_{ss} be-tween them.

5) DNs will encrypt their data using k_{ss} and aggregator node will decrypt the data using k_{ss}. Aggregator node will then aggregate the data and forward the same for further processing.

4. Energy Calculations and Comparisons

In this section we will compute the energy require-ment of our protocol and compare it with other exist-ing protocols. For the computation of energy we have following assumptions:

1) We consider a complete binary tree of sensor nodes (as shown in gure) for the ease of calculations. We have considered tree of level 3 consisting of 14 sensor nodes and 1 BS.
2) We assume all the sensor nodes to be TelosB motes of Texas Instruments with operating system tinyOS and tinyECC library [17] implemented. We assess the ECC point multiplications and ECDSA veri-cations involved in ECDH-ECDSA relying on the re-sults of [18]. They implemented ECC and ECDSA in TinyOS for many platforms including TelosB. We use their results for the secp160r1 elliptic curve domain parameters (160-bit keys). The technical speci cation of telos mote and it's comparison with other available sensors is given in [19].
3) We will use precomputed values for energy calcu-lation. Energy required for the calculation of MAC and it's veri cation has been taken from [20].En-ergy required for operations on mote and signature calculation/verification using ECC has been taken from table II and table IV of [21].Energy re-quired for symmetric AES encryption/decryption has been taken from Table 2 and Table 3 of [22].

Energy cost of various operations on TELOS B sensor node with Message Size of 28 B (Llj)

Table 2

SYMMETRIC ENCRYPTION (AES)	207.36
SYMMETRIC DENCRYPTION (AES)	318.72
ECC-160 POINT MULT	17000
ECDSA-160 SIGN	15000
ECDSA-160 VERIFY	19000
RSA-1024 SIGN	304000
RSA-1024 VERIFY	11900
DATA SENT	737.28
DATA RECEIPT	829.44
MAC CALCULATION	410.98
MAC VERIFICATION	410.98

4) In case of LWWAP even though the message length would vary at each stage but for the sake of convenience we will consider the average message length as 128B on the lines of kerberos [21] for relay at

each stage. This average message length is w.r.t. our assumed underlying tree.

5) We will calculate energies corresponding to En-cryption/Decryption, Signature and it's Veri cation, MAC calculation and it's veri cation and nally data transfer and receipt. We will ignore all other oper-ations for e.g. XOR, Sleep etc. because their e ect on energy would be quite negligible. We also assume that each node has a seed and it simply encrypts it everytime to get a new nonce. So w.r.t. nonce we will use the cost of symmetric encryption. The average energy requirement per mote in microjoules for vari-ous operations (considering 128 byte message size) on telosB motes are cumulatively given in the table 2

4.1 Node Authentication

Our proposed protocol uses symmetric encryption in maximum occasions. Symmetric cryptography with randomisation (including nonces in our case) increases the security of protocol and makes it comparable with Asymmetric cryptosystem. Most of the protocols dis-cussed o late has made use of ECC-PKC for the au-thentication of node with argument that ECC provides considerable savings in terms of energy requirements. We will calculate the energy expenditure in our initial node authentication scheme and compare it with an existing node authentication protocol [5].We will do our computations w.r.t. the tree considered.Assuming Nonce selection as one symmetric encryption, the cu-mulative number of operations carried out at nodes at various levels is shown in the table below:

No. of Operations Carried Out at Nodes of Various Level in the Tree Considered
Table 3

Operation → Nodes Level↓	Symmetric Encryption	Symmetric Decryption	MAC Calculation	MAC verification
Level	2	2	2	2
Level	4	8	12	8
Level	8	16	32	32
Total	14	26	46	42

Using Table 2 and Table 3, total energy consumed by our scheme w.r.t. tree formed is

$$E_{tot} = 14X\ 207.36+26\ X\ 318.72+46\ X\ 410.98 + 42\ X\ 410.98$$

$$E_{tot} = 47356\ \mu J$$

On the other hand if we use [5] to carry out simi-lar authentication of 14 sensor nodes of the tree then there would be 14 ECDSA-SIGN and VERIFY opera-tions and 14 ECC-160 POINT MULT operations and thus total energy consumed is given by

$$E'_{tot} = (17000 + 15000 + 19000)\ X\ 14$$

$$E'_{tot} = 714000\ \mu J$$

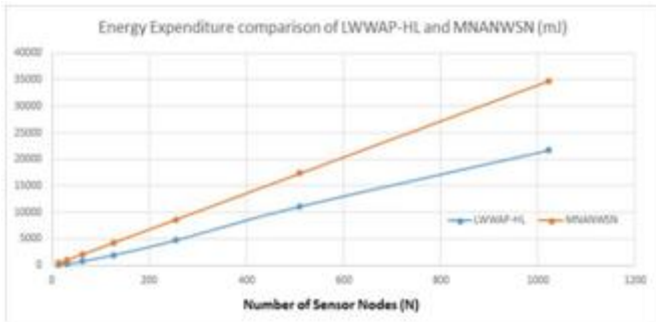
Certainly $E'_{tot} >> E_{tot}$ and even if we consider that sensor nodes have precomputed key parameters and ignore the ECC-160 POINT MULT operations then also total cost is $E''_{tot} = (17000 + 15000)\ X\ 14 = 448000\ \mu J$ which is again approximately 9 times more than our proposal.

Next if we consider the energy cost towards data trans-fer then in our example there would be over 34 sent and 34 receipt operations. Considering the maximum possible message size of 128B, total expenditure (as per [21]) towards data transfer is given by

$$E_{data} = 34\ X\ 737{:}28 + 34\ X\ 829{:}44$$

$$E_{data} = 53268{:}48\ \mu J$$

Overall energy expenditure of our proposal : $E_{overall} = E_{tot} + E_{data} = 47356 + 53264.48 = 100620{:}48\ \mu J$ which is again better than the energy requirement of [5] even without adding the data trans-fer energy expenditure in the later case.Below nd the graph comparison of MNANWSN [5] and our pro-posed node authentication protocol with binary tree as underlying network structure for the ease of cal-culation.In case of MNANWSN we have ignored the data transfer energy expenditure and also considered that each node has precomputed key parameters.



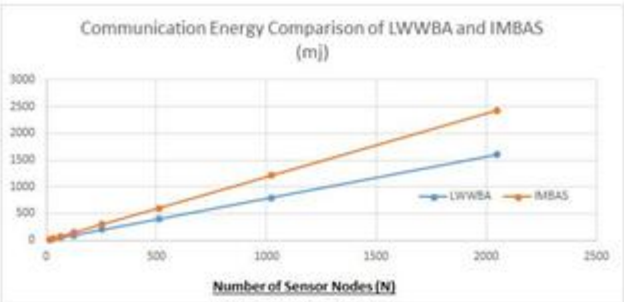
4.2 Broadcast Authentication

In previous case we considered an average message length as 128B but in this case we will try to calcu-late the exact energy required to carry out broadcast authentication by BS. We will compare our energy requirement with the IMBAS protocol proposed in [23].

For BS to broadcast a message it has to send mes-sage encrypted using group key and the signature on it. Considering the original message size to be 16B, the message encrypted using 128 bit group key by AES scheme would be 16B. Signature using ECDSA-160 will be of length 40B. We also have 8-bits header. Consequently the total length of the broadcast mes-sage will become 64B which also includes '0' padding. Also, in our assumed arrangement of nodes, each node will receive the message once and send it twice (both its children) except for the leaf nodes which will have only one receipt. So total number of messages received and sent in our network are 14 X 1 = 14 and 6 X 2 = 12 respectively. Total energy consumed (using Table 2) to-wards message transmit and receipt in the whole network is $12\ X\ \frac{737.28}{128}\ X\ 64 + 14\ X\ \frac{829.44}{128}\ X\ 64 = 10229{:}76\ \mu J$.

Each node on receiving a message will carry out a sig-nature veri cation and symmetric decryption. Total energy spent (using Table 2) by a node in these two operations are $19000 + \frac{318.72}{128}$ X 16 = 19039:84μJ. Total energy spent towards these two operations by entire network = 19039:84 14 = 266557:76 μJ = 266:56mJ.

IMBAS Protocol [23] for broadcast authentication has done all it's energy calculations corresponding to MI-CAz mote and thus we can't directly compare the energy consumption of both protocols because MI-CAz signature veri cation consumes three times more energy [21] (63mJ) than in case of Telos B mote. But with a message (m) size of 10B, the overall size of broadcast message in [23] is 97B whereas in our case the broadcast message is of size 64B even with m of size 16B. So we have a savings of 33B in terms of broadcast message size. Moreover, we have assumed a tree structure which results in 1 receipt and 2 send operations per node whereas in case of [23] it would be N-1 receipts and 1 sent operation per node (where N is the number of nodes). So even if we use same specification sensor node in both protocols, our protocol gives considerable energy saving in terms of message size and number of receipt operations per node. The graph given below compares the two protocols i.e. our proposed LWWBA and IMBAS only in terms of com-munication energy requirement(in mj) with number of nodes plotted on x-axis and energy plotted on y-axis.We have considered that the underlying network has binary tree structure in both cases.



4.3 GKDNM and ADTW

Both NMGKC and ADTW are speci c to our pro-posed protocol suite and thus we have only carried out energy requirement calculations. No energy compari-son has been made in both cases. Again we consider underlying structure to be a complete 3 - level binary tree.

4.3.1 GKDNM

For the distribution of group key in a network tree consisting of 14 nodes, the size of message transmitted by BS would be 336B. Thus, the level 1 nodes would receive message of size 336B. Of which they will re-move their ID and encrypted group key and send the remaining message to their children. Thus, in our assumed underlying tree, nodes of level 2 will receive message of size 304B. Similarly nodes of level 3 will receive message of size of 288B. Each node will also carry out a symmetric decryption to get its group key. Finally, each node will send will 16B acknowledgement to BS. Therefore, total number of operations and data transmissions and corresponding energy expenditures (in microjoule) are:

i. 2 receipts of 336B : $\frac{829.44}{128}$ X 336 X 2 = 4354:6

ii. 4 receipts and 4 transmissions of 304B : $\frac{1566.72}{128}$ X 304 X 4= 14883:8

iii. 8 receipts and 8 transmissions of 288B: : $\frac{1566.72}{128}$ X 288 X 8 = 28201

iv. 14 transmission of 16B: $\frac{737.28}{128}$ x 16 X 14 =1290.2

v. 14 symmetric key decryptions : $\frac{318.72}{128}$ x 16 x 14 = 557.8

Thus, the overall energy expenditure in the considered network for group key distribution is sum of all above i.e. 49.28 mJ.

In the case or Network Monitoring once each node receives a broadcast from BS, simply forwards it's ID and ID encrypted using shared secret between BS and the concerned node. Assuming the ID to be 7 B and it's encryption using 16B shared key by AES-CBC yields 16B cipher text, total length of transmission from one node is 32B which also includes '0' padding. So if there is no change in network, then the energy spent (in μ J) at each node in this phase is

$$\frac{737.28}{128} \text{ x } 32 + \frac{207.36}{128} \text{ X } 16 = 210.24$$

Overall energy expenditure in the assumed network is 14 210:24 = 2943:36 J.This is over and above the energy expenditure of LWWBA.

4.3.2 ADTW

In this phase once an aggregator node and correspond-ing data node establishes ECDH shared key between them, then it's only symmetric encryption and decryp-tion. For example if in the underlying complete binary structure each parent node acts as aggregator node for corresponding children data nodes then there would be 14 symmetric encryptions and 6 symmetric decryp-tion. Considering 7B ID and 16B encrypted data there would 14 send operations of 32B and 6 receipt opera-tions of 32 B. Total energy is given by

$$((\frac{737.28}{128} \text{ x } 32 + \frac{207.36}{128} \text{ X } 16) \text{ X } 14) + ((\frac{829.44}{128} \text{ x } 32 + \frac{318.72}{128} \text{ x } 16) \text{ x } 6)$$

2943:36 + 1483:2 = 4426:56μ J

5. Future Work

In this proposed work of ours, we have assumed that a tree of sensor nodes exists before carrying out the authentication of nodes. However, considering the re-quirement of future, the tree has to be formed while carrying out authentication of nodes. This require-ment falls in line with the current situation at LAC (Line of Actual Control). If the tree gets formed while carrying out the authentication of nodes then we may use a WSN to monitor the intrusion of enemy into our side of LAC. This can be done by dropping sen-sor nodes using drone at the enemy's side of LAC and having BS on our side. BS and sensor nodes would authenticate each other and form a tree with BS at root. Consequently any intrusion may get monitored and actions may be taken accordingly.

6. Conclusion

In this report we rst identifed the problem of au-thentication protocols in WSNs. We pointed out that the asymmetric-key based solutions achieve required security level but is expensive in terms of energy ex-penditure whereas symmetric-key based solutions are inexpensive but weak in security as well. We then came up with an e ective protocol suite which covers almost all authentication protocols of WSNs and also achieves the requirement of secured as well as light weight. Consequently, WSNs may be used for several purposes which have been challenging and wanting till date. Our proposed protocol has attempted to meet both requirements of communication in WSN. Further attempts may be made to improvise the security and reduce the energy requirement for which our report may act as starting point.

References

[1] Tobias Markmann Authentication Schemes for Wireless Sensor Nodes at a Glance Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2011 8th Annual IEEE Communications Society Con-ference on In A WI Report vii, viii, 10, II, 12.

[2] Chen Xiao, Guo Dawei, Li Jiawei, uTESLA broadcasting authentication protocol optimization re-search[J], Sensing technology journal, 2009, Vol.22, No.11, P1623-1625

[3] S. Zhu, S. Setia, and S. Jajodia, \LEAP+: e cient security mechanisms for large-scale distributed sensor networks," ACM Transactions on Sensor Networks, vol. 2, no. 4, pp. 500{528, 2006.

[4] H. Chan, A. Perrig, D. Song Random key pre-distribution schemes for sensor 2003 Innetworks, in: Proceedings of the 2003 IEEE Symposium on Security and Privacy viii, 14, 15

[5] Ismail Mansour, Damian Rusinek, Gerard Chalhoub, Pascal Lafourcade, Bogdan Ksiezopol-ski;Multihop Node Authentication Mechanisms for Wireless Sensor Networks,https://hal.archives-ouvertes.fr/hal-01759853,Apr 2018.

[6] K. Han and T. Shon., "Sensor authentica-tion in dynamic wireless sensor network environments", International Journal of RFID Security and Cryptography, 2012.

[7] Q. Huang, J. Cukier, H. Kobayashi, B. Liu, and J. Zhang, \Fast authenticated key establishment proto-cols for self-organizing sensor networks," in Proceed-ings of the 2nd ACM International Conference on-Wireless Sensor Networks and Applications (WSNA ’03), pp. 141{150, San Diego, Calif, USA, September 2003.

[8] P. Vijayakumar and V. Vijayalakshmi, \E ec-tive key establishment and authentication protocol for wireless sensor networks using elliptic curve cryp-tography," in Proceedings of the Conference on Mo-bile and Pervasive Computing (CoMPC ’08), August 2008.

[9] D. J. Malan, M. Welsh, and M. D. Smith, \A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography," in Proceedings of the 1st Annual IEEE Conference on Communications Society Sensor andAdHoc Commu-nications and Networks (SECON ’04), Santa Clara, Calif, USA, 2004.

[10] K. Ren, S. Yu, W. Lou, and Y. Zhang, \Multi-user broadcast authentication in wireless sensor networks," IEEE Transactions on Vehicular Technology, vol. 58, no. 8, pp. 4554{4564, 2009.

[11] Y. Zhou, Y. Zhang and Y. Fang, Access Control in Wireless Sensor Networks, Ad Hoc Networks, vol. 5, pp. 3{13, (2007)

[12] H. F. Huang, A Novel Access Control Protocol for Secure Sensor Networks, Computer Standards In-terfaces, vol. 31, pp. 272{276, (2009).

[13] K. H. M. Wong, Y. Zheng, J. Cao, and Wang, \A dynamic user authentication scheme for wireless sensor networks," in Proceedings of the IEEE Inter-national Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, pp. 318{327, 2006.

[14] Aqeel Khalique, Kuldip Singh, Sandeep Sood,\Implementation of Elliptic Curve Digital Sig-nature Algorithm" International Journal of Computer Applications (0975 { 8887) Volume 2 { No.2, May 2010

[15] David J. Malan, Matt Welsh, Michael D. Smith, \A Public-Key Infrastructure for Key Distribution in TinyOS Based on Elliptic Curve Cryptography", IEEE SECON, pp.4-6,2004.

[16] Kyung-Ah Shim, \A Survey of Public-Key Cryp-tographic Primitives in Wireless Sensor Networks", IEEE COMMUNICATION SURVEYS TUTORI-ALS, VOL 18, NO.1, FIRST QUARTER,pp. 583-587 2016.

[17] Peng Ning,An Liu;"TinyECC: A Con gurable Li-brary for Elliptic Curve Cryptography",International Conference on Information Processing in Sensor Net-works,pp. 2-6,2008.

[18] A. Liu and P. Ning. TinyECC: A con gurable library for Elliptic Curve Cryptography in Wireless Sensor Networks. Technical Report TR-2007- 36, North Carolina State University, Department of Com-puter Science, 2007.

[19] Joseph Polastre, Robert Szewczyk, and David Culler,"Telos: Enabling Ultra-Low Power Wireless Research"ACM/IEEE International Conference on Information Processing in Sensor Networks,May 2005.

[20] Merad Boudia Omar Ra k,Feham Mo-hammed,"Performance evaluation on TelosB mote of a secure data aggregation protocol using ECC"Conference Nationale sur les Technologies de l‘Information et le Telecommunications CNTIT‘13, pp.4-5,10-11 Decembre 2013.

[21] Giacomo de Meulenaer, Franc ois Gosset, Franc ois-Xavier Standaert, Olivier Pereira;"On the Energy Cost of Communication and Cryptography in Wireless Sensor Networks"IEEE International Confer-ence on Wireless and Mobile Computing, Networking and Communications,2008.

[22] Arvinderpal S. Wander,Nils Gura, Hans Eberle,Vipul Gupta,Sheueling Chang Shantz;"Energy Analysis of Public-Key Cryptography for Wireless Sensor Networks",Proceedings of the 3rd IEEE Int’l Conf. on Pervasive Computing and Communica-tions,2005.

[23] Xuefei Cao, Weidong Kou, Lanjun Dang, Bin Zhao, IMBAS: Identity-based multi-user broadcast authentication in wireless sensor networks", Computer Communications 31 659{667,2008.

[24] Anand d. Dhawale, M.B.Chandak, \Implementa-tion of Rekeying Mechanism for Node Authentication in Wireless Sensor Networks", International Journal of Advanced Smart Sensor Network Systems, vol. 2, No. 4, October 2012
[25] Aqeel Khalique; Kuldip Singh; Sandeep Sood, Implementation of Elliptic Curve Digital Sig-nature Algorithm" International Journal of Computer Applications (0975 {8887) Volume 2 {No.2, May 2010
[26] David J. Malan, Matt Welsh, Michael D. Smith, \A Public-Key Infrastructure for Key Distribution in TinyOS Based on Elliptic Curve Cryptography", IEEE SECON, pp.4-6,2004.