Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 89 (2016) 90 – 98

Procedia Computer Science

Twelfth International Multi-Conference on Information Processing-2016 (IMCIP-2016)

Implementation of Node Authentication for WSN using Hash Chains

Ayaz Hassan Moon[a,*], Ummer Iqbal[a] and G. Mohiuddin Bhat[b]

[a] National Institute of Electronics and Information Technology, J & K
[b] University of Kashmir, J & K

Abstract

This paper presents the design and implementation of a lightweight node authentication protocol which comprises of node registration, node authentication and key establishment phases. The solution leverages the low computational overheads associated with cryptographically secure one-way hash chains and Elliptical Curve Crytography (ECC) without using any digital signature algorithm or any public key cryptography. The usage of hidden generator point derived from hash-chains provides defense against a man-in-the-middle attack which is prominent in ECDH (Elliptical Curve Diffie-Hellman) due to lack of entity authentication. The authentication protocol has been simulated on Tossim and its performance bench marking has also been carried out.

Peer-review under responsibility of organizing committee of the Organizing Committee of IMCIP-2016

Keywords: Authentication; Access Control List; Entity Authentication; Hidden Generator; Wireless Sensors Network.

1. Introduction

The resource constrained nature of sensor nodes characterized by limited energy, small memory and less computational power[1] prohibit the use of conventional security primitives as used by other networks to mitigate various threats[2]. Thus, the design of appropriate security solutions for WSN has been an active research challenge. Any potent existing security solution no matter how effective it might be in the case of other networks has to fit itself into WSN framework mainly comprised of resource constraint nodes. A tradeoff between the security services and resources consumed has to be weighed properly before deciding on its adoption. Thus, the design of new protocols and security architecture to implement confidentiality, data integrity, data authentication and data freshness within a resource constraint network like WSN has to be viewed in the correct perspective[3].

This paper focuses on one of the important security services like Node Authentication and Key Generation and Establishment among the broad domain of security concerns faced by the WSN. Authentication is an important security primitive in any class of network both at the entity level as well as at the message level. In message authentication, there is no timeliness guarantee with respect to the message while as entity authentication occurs in real time and both the parties have to be live at that moment though no meaningful message is exchanged[4,5]. All the network entities comprising of WSN including nodes, cluster heads and base station need to be authenticated before sending or receiving any kind of communication within them. Public Key Cryptography offers broad-based solutions to address

*Corresponding author. Tel.: 0194-2300502; Fax: 0194-2300949.
E-mail address: moonah@rediffmail.com

Peer-review under responsibility of organizing committee of the Organizing Committee of IMCIP-2016
doi:10.1016/j.procs.2016.06.013

all the security concerns. However, such solutions are far too expensive to be applied directly to WSN owing to its resource constraints[6]. Key establishment plays a pivotal role in ensuring authentication. An uncomplicated yet efficient method to share secret keys in WSN is based on ECDH. On a MICAZ mote, ECDH operation takes 9.8 seconds to establish shared keys between a pair of nodes[7]. However, ECDH suffers from Man-in-the-Middle Attack. The Man-in-the-Middle-Attack can be overcome by using Hidden Generator Point concept[8,9] and proper authentication mechanism.

In a broader framework, a proper access control mechanism has to be enforced within the network which could broadly have two levels:

- Node Authentication
- Proper Key generation & Establishment.

This paper presents the design and implementation of a lightweight node authentication protocol for WSN on TinyOS[10]. The proposed work facilitates node registration, node authentication and key establishment. The solution primarily leverages the low computational overheads associated with cryptographically secure one-way hash chains without using any digital signature algorithm.

Rest of the paper is organized as following: Section 2 provides an insight into the related work, Section 3 describes the detailed Authentication protocol in 4 phases, section 4 gives the implementation details Section 5 highlights the security analysis of the proposed scheme, section 6 discusses the performance of the proposed protocol, while as Section 7 concludes the paper.

2. Related Work

Authentication mechanisms using Hash Chains are considered to be computationally feasible for resource constraint network like WSN. Hash Chains were first proposed by Lamport who used it for generating one-time password[11]. This involves applying a hash function $h(.)$ repeatedly z times to a seed s to form a hash chain of length z. The hash (.) is easy to compute but hard to invert e.g., $h(h(h(s)))$ gives a hash chain of length 3 and can be denoted by $h^3(s)$. The initial element of the hash chain is called the seed and the last element is called committed value or the tip of the hash chain. The tip of the chain is public and is distributed among the nodes and the elements of the chain are consumed one after other until the secret key is free. Hash chains find exclusive use in data integrity and entity authentication. The ith element of the hash chain denoted as K_i is expressed as:

$$K_i(s) = h(h^{z-i}(s)) \tag{1}$$

The committed value of the chain is made public while as the seed acts as private or secret value. The scheme does not overcome the need for an authenticated initial key-exchange. By the definition of entity authentication, Lamport's one-time passwords do not provide entity authentication as there is no proof of an active communication between the two parties. Some of the relevant schemes offering entity authentication as part of overall access control have been given by Zhou[12] *et al.* which is based on ECC and ECC-based digital signature scheme ECDSA. It has been proved to be an energy efficient than RSA. It achieves node authentication and key establishment for new nodes by including both node identity and node bootstrapping time into the authentication procedure. However, it uses timestamps and assumes that each sensor node can sustain time interval before it can be compromised. Therefore, for practical implementations, it is not thought to be convenient. Huang proposed NACP[13] scheme which is based on Hash chains and ECC. It is simple, energy efficient supports new node addition but has been found to be vulnerable to replay attack and new node masquerading attacks. This is attributed to the absence of any mutual authentication between node and base station. It also lacks hash chain renewability. Other schemes supporting authentication where given under ENACP an enhancement over NACP, PACP and NDACP[14]. Out of the schemes discussed only ENACP provides authenticated broadcast.

3. Proposed Node Authentication Protocol

The proposed scheme presents a comprehensive pair-wise entity authentication protocol with the proper key establishment in 4 phases i.e. Initialization, Node registration, Key generation and Node authentication and Node to Node authentication involving Node identities. The notations used in the protocol are tabulated in Table 1.

Table 1. Notations used.

Symbol	Notation
$h()$	One way-hash function
p	A prime number
Zp	A finite field
Ep	Elliptic curve
G, Ga, Gb	Generator point of order n
N_i	Node identity of i^{th} node
N_j	Node identity of j^{th} node
BS	Base station
K_i	Secret key of i^{th} node
K_j	Secret key of j^{th} node
K_s	Secret key of base station
z	Large integer
$hm(k)$	m cascade hash operations on key k
x_{ij}	Sessions key establishment between i^{th} node and j^{th} node
$\oplus$	XOR Operation
$\|$	Concatenation Operator

3.1 Initialization

Let there be r nodes with $N1, N2, \ldots, Nr$ as their identities constituting the neighborhood of a WSN. The node identities are integer numbers. The base station (BS) selects secret key k_s and computes its hash chain $h^z(k_s)$ by applying the select hash function $h(.)z$ times over k_s. BS also generates r number of secret keys $k1, k2, \ldots, kr$ for each of the node. It calculates $h^z(k_i)$ as the hash-chain commitment of each node with $i = 1, 2, 3, \ldots, r$ by repeatedly applying hash function z times. Further BS initiates following actions:

- It preloads each of the Node N_i with its associated secret key K_i (seed) and one-way hash function $h()$.
- It calculates its own hash chain commitment $h^z(k_s)$ and preloads it in all the nodes.
- It selects an elliptic curve Ep, a cyclic group G and preloads its associated parameters like G, n, a, b, p, H in all the nodes.

3.2 Node registration

In the post-deployment phase, each node has to register itself with the base station before it can communicate with the other nodes in its neighborhood.

Step 1: $N_i \rightarrow \text{Base} h(N_i \oplus ki)$, N_i: Ni hashes the XOR of its node-id Ni with its secret key ki and sends it to the base along with its node-id. BS Verifies the $h(N_i \oplus k_i)$ by using the k_i from its own storage and N_i from the Step 1. If the verification holds, then base station adds node N_i to its access control list.

Step 2: The purpose of this step is to broadcast the hash chain commitment of the registered node $h^z(k_i)$ by the BS in an authenticated manner. The authenticated broadcast of base station to the nodes is achieved in the following steps:

$$BS \rightarrow * : h(h^{z-1}((k_s) \| (h^z(k_i) \otimes N_i) \| nB)) = z_i \tag{2}$$

$$BS \rightarrow * : (h^z(k_i) \otimes N_i), h^{z-1}(k_s), N_i, nB \tag{3}$$

$h^{z-1}(k_s)$ is the secret chain value of BS and can be computed only by the BS. It has the significance of private key to BS. This value is concatenated to hash chain commitment of the registered node and its id as per expression z_i. nB is a nonce and has been added to mitigate replay attack.

Step 3: On receiving the above broadcast as per (2) and (3), nodes in the network including N_i first verify the expression:

$$h(h^{z-1}(k_s)) = h(k_s) \tag{4}$$

If found true, (which implies that it has originated from BS), only then the expression:

$$h(h^{z-1}(k_s))\|h^z(k_i) \oplus N_i)\|nB) \tag{5}$$

is evaluated and compared with the value of BS broadcast z_i. If it holds, then the hash chain commitment $h^z(k_i)$ of Ni is extracted from $(h^z(k_i) \oplus N_i)$ by Ex-oring it with N_i and then registered in their access control list along with the node-id N_i. The broadcast hash chain of BS is updated to $h^{z-1}(k_s)$.

3.3 Key generation and node authentication

Suppose a pair of nodes i.e. N_i and N_j which are in each other's radio range want to communicate with each other. Let z be the hash chain length of both the nodes. Let N_i and N_j have successfully passed through u and v times authentication respectively.

Key generation step using hidden generator:

Pair-wise symmetric key generation between two communicating nodes is computed using hidden generator concept. By this, it is assumed that the two nodes have different Generator points G_a and G_b for the elliptic curve Eq., which has not been made public. This would help in mitigating the man-in-the-middle attack. Node identities are also tied-up to key generation.

Ist Exchange between Ni and Nj

$$N_i \rightarrow N_j : Ga.(h^{z-u-1}(k_i)).N_j \tag{6}$$

$$N_j \rightarrow N_i : Gb.(h^{z-v-1}(k_j)).N_i \tag{7}$$

Assuming that the current hash chain value of N_i and N_j is h^{z-u} and h^{z-v} respectively. The scalar multiplication of hash chain secrets of two nodes i.e, $h^{z-u-1}(ki)$ and $h^{z-v-1}(kj)$) and node identities N_i and N_j with the respective generator points G_a and Gb shall result in a point on the elliptic curve. It will also associate node identity with the process of key generation.

2nd Exchange between Ni and Nj

Nodes will exchange their hash chain secrets which can be generated by them only.

$$N_i \rightarrow N_j : h^{z-u-1}(k_i) \tag{8}$$

$$N_j \rightarrow N_i : h^{z-v-1}(k_j) \tag{9}$$

Verification phase

Ni verifies $h(h^{z-v-1}(k_j)) = h^{z-v}(k_j)$, if true, then it computes

$$Gb.[h^{z-v-1}(k_j)].N_j.[h^{z-v-1}(k_j)]^{-1}.[N_j]^{-1} = Gb \tag{10}$$

Nj verifies $h(h^{z-u-1}(k_i)) = h^{z-u}(k_i)$, if true, then it computes:

$$Ga.[h^{z-u-1}(k_i)].N_i.[h^{z-u-1}(k_i)]^{-1}.[N_j]^{-1} = Ga \tag{11}$$

Shared key between N_i and N_j

Now after 4 exchanges, Ni has the knowledge of G_b that is the generator point of N_j and N_j has the knowledge of Ga that is the Generator point of N_i.N_i and Nj arrive at a Common Generator $Gs = Ga + Gb$. Gs is a point on the elliptic curve $P(xs, ys)$. The shared key between Ni and Nj shall be the x coordinate of point P i.e., xs. The pairwise symmetric key established between N_i and $N_j = x_{ij}$.

3.4 Node-Node Authentication

Node to Node Authentication is based upon verification of hash chain commitment of each node, their node identity and the shared key between them. If either of the verification tests fails, then the authentication will not be completed. Assuming Ni has completed u and Nj has completed v number of successful authentications.

Ni computes the following expressions ai and ak and sends it to Nj:

$$N_i \rightarrow N_j : h(h^{z-u-1}(k_i)\|N_j) = a_i \tag{12}$$

$$N_i \rightarrow N_j : h(h^{z-u-1}(k_i)\|x_{ij}) = a_k \tag{13}$$

N_i broadcasts $h^{z-u-1}(k_i)$ and N_j.

Similarly N_j computes the following expression b_j and b_k and sends it to N_i:

$$N_j \rightarrow N_i : h(h^{z-v-1}(k_j)\|N_i) = b_j \tag{14}$$

$$N_j \rightarrow N_i : h(h^{z-v-1}(k_j)\|x_{ij}), = b_k \tag{15}$$

N_j broadcasts $h^{z-v-1}(k_j)$ and N_i.

Verification phase

Ni computes the following expression: $h(h^{z-v-1}(k_j)) = h^{z-v}(k_j)$. If found correct then N_i verifies the following expression:

$$h(h^{z-v-1}(k_j)\|N_i) = b_j \tag{16}$$

$$h(h^{z-v-1}(k_j)\|x_{ij}) = b_k \tag{17}$$

If both the verifications hold, then N_i authenticates N_j as expressions b_j and b_k would have been computed by N_j only. Similarly, Nj computes the following expressions: $h(h^{z-u-1}(k_i)) = h^{z-u}(ki)$. If found correct then Nj verifies the following expression:

$$h(h^{z-u-1}(k_i)\|N_j) = a_i \tag{18}$$

$$h(h^{z-u-1}(k_i)\|x_{ij}) = a_k \tag{19}$$

If both the verifications hold, then Nj authenticates Ni as expressions aj and ak would have been computed by Nj only. Upon successful authentication, Ni upgrades its hash chain to $h^{z-u-1}(ki)$ and N_j upgrades its hash chain to $h^{z-v-1}(k_j)$ and the same is communicated to BS.

4. Simulation

The protocols were implemented in TinyOS operating system using NesC Language and simulated on TOSSIM[15]. TinyOS is an open-source lightweight operating system specifically designed for low-power wireless sensors. The NesC programs developed were enabled with a highly optimized ECC implementation, TinyECC[16]. The simulation has been carried out in TOSSIM Simulator using DBG flags. The output of various calculation and transmissions in the protocols has been captured. TOSSIM provides configuration of debugging output at runtime. The simulation output of node registration and Node authentication & Key generation are shown in Table 2 and Table 3.

Table 2. Simulation Output of Node Registration.

Operation	Simulator Output
Ni computes h ($N_i \oplus k_i$) and sends h ($N_i \oplus k_i$), N_i to Base	Node 1 is sending Hash(ID ^ K1),ID to Base 16 99 146 162 59 66 173 190 76 205 52 208 221 128 248 208 79 246 220 235
After Verification, base station adds node Ni to its access control list	Base Has Recieved Hash(ID ^ K1),ID from Node 1 Hash(ID ^ K1) generated by Base from its Own Storage is 169914616259661731907620552208221128248208792462202 35 Hashes Match !! Node 1 is registered by Base
Base Broadcasts the following to the Network: B1: $h(h^{z-1}((k_s) \| (h^z(k_i) \oplus N_i) \| nB))$ B2: $(h^z(k_i) \oplus N_i), h^{z-1}(k_s), N_i, nB$	B1: H(H(H(Ks)+ (H(H(H(Ki)^Ni)+ nB) Broadcasted By Base is 156 179 196 78 26 114 61 199 117 21 26 82 38 169 162 171 144 200 58 78 B2: H(H(Ks),(H(H(H(Ki)^Ni), nB) Broadcasted By Base 94 93 72 220 155 198 15 93 230 27 194 180 178 8 104 233 250 182 228 213 34 251 1 82 87 67 130 164 192 63 65 145 23 73 125 113 212 106 121 159 99
Nk receives the broadcast B1 and B2 and verification add Ni to its access Control list	Node 2 is Verifying Hash(B2)?= B1 Node 2 has added N1 in its ACL Node 3 is Verifying Hash(B2)?= B1 Node 3 has added N1 in its ACL

Table 3. Simulation Output of Node Authentication and Key Generation.

Operation	Simulator Output
Compute $Ga.(h^{z-u-1}(k_i)).N_j$ and send it to Nj	Ga.(H(H(Ki)).N2 x value 1d af 25 2d cd 9d 33 f0 cd 6 6f 40 5f c8 d 28 21 6b 6d c8 0 y value e2 b2 c8 22 40 80 99 a1 49 d9 87 da 1d b0 56 83 23 bd fa d 0
Compute $Gb.(h^{z-v-1}(k_j)).N_i$ and send it to Nodei	Gb.(H(H(Kj)).N1 is as follows x value 17 78 ee fb f2 e6 b5 3f 4f f3 bd c4 ff d5 bc 4a 9f 86 83 2c 0 y value 7f 11 20 cb 59 6e 9b 4f fa 0 2e e6 1c 93 fc b2 1d 9c e2 5d 0
Ni verifies $h(h^{z-v-1}(k_j))$?= $h^{z-v}(k_j)$, if true, then it computes $Gb.[h^{z-v-1}(k_j)].N_j.[h^{z-v-1}(k_j)]^{-1}.[N_j]^{-1} = Gb$	Node 1 is Verifying Hash Commintment of the Node 2 The verification Hash Generated by Node 1 is : 95 92 73 221 154 199 14 92 231 26 195 181 179 9 105 232 251 183 229 212 The Public Hash Commitnent of N2 is 95 92 73 221 154 199 14 92 231 26 195 181 179 9 105 232 251 183 229 212 The hashes match !! N2 is autheticated by Node 1 Gb obtained by N1 is x value bc 2b a4 2f 43 37 c7 a2 f4 c9 fd 96 41 37 53 8f 7 58 4d 2f 0 y value c7 31 e1 f3 90 2 9f 3b 81 61 ae c 22 88 9f 79 fa 82 4b f7 0
Nj verifies $h(h^{z-u-1}(k_i))$?= $h^{z-u}(k_i)$, if true, then it computes: $Ga.[h^{z-u-1}(k_i)].N_i.[h^{z-u-1}(k_i)]^{-1}.[N_j]^{-1} = Ga$	Node 2 is Verifying Hash Commintment of N1 The verification Hash of N1 Generated by Node 2 is : 220 47 234 250 12 94 226 106 8 209 249 2 166 86 31 66 106 26 86 23 The Public Hash Commitnent of N1 is 220 47 234 250 12 94 226 106 8 209 249 2 166 86 31 66 106 26 86 23 The hashes match !! N1 is autheticated by Node 2 Ga obtained by N2 is x value 6b 7a 42 13 ed f2 8b f0 f2 e8 5b fc 63 56 30 ca 84 d5 94 68 0 y value 23 ac 5a 61 41 49 86 59 8b a5 6a 20 3c 3 6b ae 20 b4 48 97 0
Ga+Gb=Gs at Ni	The shared Point at N1 (Ga+Gb) is x value 36 27 5f 9 8d 3b e8 3b 91 d3 8f f5 fb f 43 97 71 ae 8 c6 0 y value ba 9d 21 ef 3b ba 8b a f b6 b4 5e 55 aa 49 17 3f 7c 91 58 0
Ga+Gb=Gs at Nj	The shared Point at N2 (Ga+Gb) is x value 36 27 5f 9 8d 3b e8 3b 91 d3 8f f5 fb f 43 97 71 ae 8 c6 0 y value ba 9d 21 ef 3b ba 8b a f b6 b4 5e 55 aa 49 17 3f 7c 91 58 0

5. Security Analysis

5.1 Man-in-the-middle attack

A man-in-the-middle attack is normally launched due to lack of authentication between the communicating entities as is prominent in ECDH. An adversary can also launch MIM by misusing the public disclosure of Elliptical Curve Parameters especially the Generator Point. In our proposed scheme, the concept of hidden generator has been used, wherein communicating nodes make exchanges arrive at a common generator point which is not known to the other entities. Moreover, Shared key generation has been tied to Node identities which make the scheme more robust against MIM.

5.2 Instant authentication

In a Broadcast authentication protocol like μ-tesla[3], nodes cannot authenticate the packets instantaneously because of the delayed disclosure of keys. This can be exploited by an adversary who can inject forged messages into the network and launch denial of service attack. In our proposed scheme instant authentication is provided upfront, by verifying the hash chain secret value before evaluating other expressions. Packets need not be buffered for authentication as done in the case of μ-tesla.

5.3 Malicious node injection

A node is to first register itself with the base station during Registration phase by making use of Node id and secret key k_i. It is only on successful registration, the hash chain commitment is communicated to the node by the base station. This step prevents a malicious node to join the network as it cannot participate in the network communication without forcing its entry into the Access control list of BS.

6. Energy Analysis and Performance Benchmarking

Most of the computational work during initialization phase is shifted to the base station which is supposed to be resource strong. The nodes are kept computationally light by making judicious use of hash functions which occupy around 10 kbytes ROM memory. The ECC operation is used only to generate keys. No computationally intensive techniques involving the use of certifications or digital signature for the purpose of broadcast authentication has been used in the proposed scheme. For capturing computational time of different key operations like Point Addition, Scalar Multiplication, Inverse, Hash (Sha1) a fundamental setup was created using MicaZ[17] mote and an MIB520[17] Gateway. The computational time recorded for different operations is shown in Table 4.

The number of critical operations in various phases of the authentication framework is tabulated in Table 5. For calculation of energy $E = V * i * t$ (joules) is used, where V and I stand for voltage and current drawn respectively,

Table 4. Computational Time Recorded.

S. No.	Operation	Time in Secs.
1	Scalar Multiplication	3.90
2	Point Addition	0.14
3	Inverse Operation	0.12
4	Sha1	0.0091

Table 5. Critical Operation in Node Authentication Protocol.

S. No.	Phase	Scalar Multiplication	Point Addition	Inverse Multiplication	Hash
1	Initialization	–	–	–	–
2	Node registration	–	–	–	7
3	Node authentication and Key exchange	4	1	4	4
4	Node to Node Authentication	–	–	–	12

Table 6. Energy and Time Analysis of Node Authentication Protocol.

S. No.	Phase	Time Taken (Secs)	Energy Consumed (mJ)
1	Initialization	–	–
2	Node registration	0.063	3.72
3	Key generation	16.25	960.375
4	Node to Node Authentication	0.109	6.44
Total		16.42	970.53

Table 7. Comparison of Operation for Establishing Hidden Generator Point.

Scheme	Total Number of Exchanges to Establish Key	Total Number of Scalar Multiplications	Total Number of Inverse Operations	Protection Against MIM Attack	Total Computational Cost
ECDH	02	04	Nil	N	06
Ravi K Scheme	06	08	2	Y	16
Our Scheme	04	04	2	Y	10

t is the execution time for each operation. MicaZ node using Atmel AT Mega128 L is powered by 02 AA batteries. With a voltage of 3 V for 02 AA batteries and a maximum load current of 19.7 mA, the energy and time calculations are indicated in Table 6.

The Hidden generator scheme used for generation and establishment of pairwise keys has been compared with ECDH and the one given by Ravi *et al.*[18]

Our scheme has low computational cost and less number of broadcasts as compared to Ravi *et al.* as shown in Table 7.

7. Conclusions

The proposed entity authentication scheme was implemented by leveraging computationally light Hash chains and Elliptical Curve Cryptography. Shared pairwise key have been derived by using Hidden Generator Points and is tied to the Node identities and hash chain values. This gives a safeguard against MIM attack eminent in ECDH. The Framework doesn't use any digital signature mechanism to achieve authenticated broadcasts. Instead, secret value of hash chains has been used for the purpose. The framework can be embedded into any WSN based application where Entity authentication is a requirement.

References

[1] I. F. Akyildiz, *et al.*, A Survey on Sensor Networks, *IEEE Communications Magazine*, pp. 102–114, (2002).
[2] Adrian Perrig, John Stankovic and David Wagner, Security in Wireless Sensor Networks, *Communications of the ACM*, vol. 47, no. 6, pp. 53–57, June (2004).
[3] Adrian Perrig, Robert Szewczyk and J. D. Tygar, Victor Wen and David E. Culler, SPINS: Security Protocol for Sensor Networks, *Proceedings of 7th International Conference on Mobile Networking and Computing*, vol. 8, no. 5, pp. 189–199, (2001).
[4] D. Hankerson, *et al.*, Guide to Elliptic Curve Cryptography, Springer, (2004).
[5] Bernard Menzes, Network Security and Cryptography, Cengage Learning.
[6] N. Gura, A. Patel, A. Wander, H. Eberle and S. C. Shantz, Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs.
[7] CHES2004, vol. 3156 of LNCS, (2004).
[8] B. Arazi, Certification of dl/ec Keys, *Proceedings of the IEEE*, P1363.
[9] Ravi Kishore Kodali, *et al.*, Implementation of ECC with Hidden Generator Point in Wireless Sensor Network, *IEEE 2014*, 978-1-4799-3635-9/14, (2014).
[10] A. H. Moon and Khan Ummer, Authentication Protocols for WSN using ECC and Hidden Generator, *International Journal of Computer Applications*, (0975–8887) vol. 133, no. 13, (2016).
[11] TinyOS. http:// www.tinyos.net
[12] L. Lamport, Password Authentication with in Secure Communication, *Comm ACM*, vol. 24, pp. 770–772, (1981).
[13] Y. Zhou, Y. Zhang and Y. Fang, Access Control in Wireless Sensor Networks, *Ad Hoc Networks*, vol. 5, pp. 3–13, (2007).
[14] H. F. Huang, A Novel Access Control Protocol for Secure Sensor Networks, *Computer Standards & Interfaces*, vol. 31, pp. 272–276, (2009).

[15] H. Huang and K. Liu, A New Dynamic Access Control in Wireless Sensor Networks, *IEEE Asia-Pacific Services Computing Conference*, DOI 10.1109/APSCC.2008.116, (2008).
[16] P. Levis, N. Lee, M. Welsh and D. E. Culler, *et al.*, TOSSIM: Accurate and Stable Simulation of Entire TinyOS Applications, SenSys (2003).
[17] Liu and P. Ning, *et al.*, Tiny ECC: A Configurable Library for Elliptical Curve Cryptography in Wireless Sensor Networks, *IPSN*, (2008).
[18] Mote Works, Getting Started Guide, PN: 7430-0102-02, March (2013).
[19] Ravi Kishore Kodali, *et al.*, Implementation of ECC with Hidden Generator Point in Wireless Sensor Network, IEEE, 978-1-4799-3635-9/14, (2014).