ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 4, April 2018

Implementation of Dynamic Cluster-based Key Management Protocol in Wireless Sensor Networks

Snehal Mankar [1], A.B. Raut [2]

P.G. Student, Department of Computer science & information technology, HVPM Engineering College, Amravati, Maharashtra, India[1]

H.O.D. Department of Computer Science & Engineering, HVPM Engineering College, Amravati, Maharashtra, India[2]

ABSTRACT: wireless sensor networks (WSNs) have been deployed for a wide variety of applications, including military sensing and tracking, patient status monitoring, traffic flow monitoring, where sensory devices often move between different locations Securing information and interchanges requires suitable encryption key conventions. In this paper, we propose a certificateless-powerful key administration (CL-EKM) convention for secure correspondence in element WSNs described by hub versatility. The CL-EKM underpins proficient key overhauls when a hub leaves or joins a group and guarantees forward and in reverse key mystery. A security analysis of our scheme shows that our protocol is effective in defending against various attacks.We implement CL-EKM in Linux OS and simulate it using NS2 simulator to assess its time, energy, communication, and memory performance.

KEYWORDS: Wireless sensor networks, certificateless public key cryptography, key management scheme, elliptic curve cryptography.

I. INTRODUCTION

In this paper, we present a certificateless effective key management (CL-EKM) scheme for dynamic WSNs. In certificateless public key cryptography (CL-PKC) [2], the user's full private key is a combination of a partial private key generated by a key generation center (KGC) and the user's own secret value. The special organization of the full private/public key pair removes the need for certificates and also resolves the key escrow problem by removing the responsibility for the user's full private key. We also take the benefit of ECC keys defined on an additive group with a 160-bit length as secure as the RSA keys with 1024-bit length.

In order to dynamically provide both node authentication and establish a *pairwise key* between nodes, we build CL-EKM by utilizing a pairing-free certificateless hybrid signcryption scheme (CL-HSC) proposed by us in an earlier work [4]. Due to the properties of CL-HSC, the pairwise key of CL-EKM can be efficiently shared between two nodes without requiring taxing pairing operations and the exchange of certificates. To support node mobility, our CL-EKM also supports lightweight processes for cluster key updates executed when a node moves, and key revocation isexecuted when a node is detected as malicious or leaves the cluster permanently. CL-EKM is scalable in case of additions of new nodes after network deployment. CL-EKM is secure against node compromise, cloning and impersonation, and ensures forward and backward secrecy. The security analysis of our scheme shows its effectiveness. Below we summarize the contributions of this paper: • We show the security weaknesses of existing ECC based key management schemes for dynamic WSNs [10], • We propose the first certificateless effective key management scheme (CL-EKM) for dynamic WSNs. CL-EKM supports four types of keys, each of which is used for a different purpose, including secure pair-wise node communication and group-oriented key communication within clusters.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 4, April 2018

Efficient key management procedures are defined as supporting node movements across different clusters and key revocation process for compromised nodes. • CL-EKM is implemented using linux OS and use a NS2 emulator to measure the computation and communication overhead of CL-EKM. Also we develop a simulator to measure the energy consumption of CL-EKM. Then, we conduct the simulation of node movement by adopting the RandomWalk Mobility Model and the Manhattan Mobility Model within the grid. The experimental results show that our CL-EKM scheme is lightweight and hence suitable for dynamic WSNs.

II. RELATED WORK

Symmetric key schemes are not viable for mobile sensor nodes and thus past approaches have focused only on static WSNs. A few approaches have been proposed based on PKC to support dynamic WSNs. Thus, in this section, we review previous PKC-based key management schemes for dynamic WSNs and analyze their security weaknesses or disadvantages. Since ECC is computationally more efficient and has a short key length (e.g. 160 bit), several approaches with certificate [5] have been proposed based on ECC. However, since each node must exchange the certificate to establish the pairwise key and verify each other's certificate before use, the communication and computation overhead increase dramatically. Also, the BS suffers from the overhead of certificate management. Moreover, existing schemes [5] are not secure. This paper proposed a key management scheme by using ECC-based signcryption, but this scheme is insecure against message forgery attacks [6]. Huang et al. [5] proposed a ECC-based key establishment scheme for self-organizing WSNs.

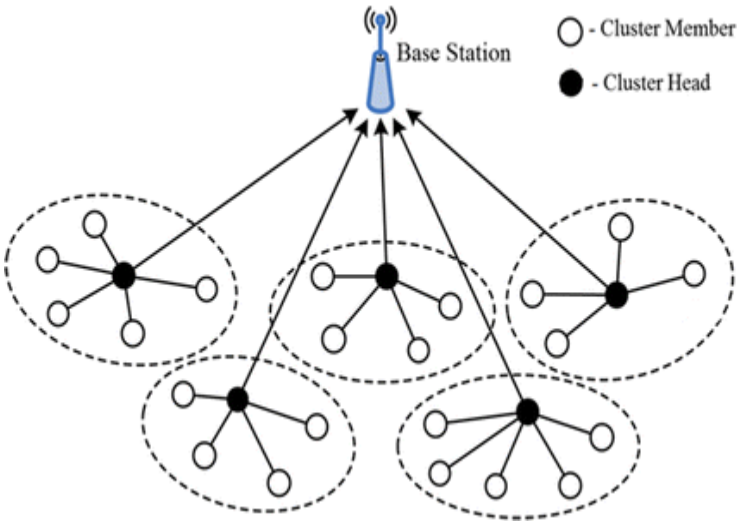

Fig 1 hetrogeneous dynamic wireless sensor network

their scheme is not compromise-resilient against cluster head capture, because the cluster head randomly generates a pairwise key between sensor nodes whenever it is requested by the nodes. Moreover, in their scheme, in order to share a pairwise key between two nodes in different clusters, these two nodes must communicate via their respective cluster heads. So, after one cluster head generates the pairwise key for two nodes, the cluster head must securely transmit this key to both its node and the other cluster head. Thus, this pairwise key should be encrypted by using the shared pairwise key with the other cluster head and the shared key with its node, respectively. Therefore, if the pairwise key between the cluster heads is exposed, all pairwise keys of the two nodes in different clusters are disclosed. The scheme supports forward and backward secrecy by using a key update process whenever a new node joins the cluster or if a node is compromised.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 4, April 2018

- NETWORK AND ADVERSARY MODELS

A. Network Model

We consider a heterogeneous dynamic wireless sensor network (See Fig. 1). The network consists of a number of stationary or mobile sensor nodes and a BS that manages the network and collects data from the sensors. Sensor nodes can be of two types: (i) nodes with high processing capabilities, referred to as H-sensors, and (ii) nodes with low processing capabilities, referred to as L-sensors. We assume to have N nodes in the network with a number N1 of H-sensors and a number N2 of L-sensors, where N = N1 + N2, and N1 _ N2. Nodes may join and leave the network, and thus the network size may dynamically change. The H-sensors act as cluster heads while L-sensors act as cluster members. They are connected to the BS directly or by a multi-hop path through other H-sensors. H-sensors and L-sensors can be stationary or mobile. After the network deployment, each H-sensor forms a cluster by discovering the neighboring L-sensors through beacon message exchanges. The L-sensors can join a cluster, move to other clusters and also re-join the previous clusters. To maintain the updated list of neighbors and connectivity, the nodes in a cluster periodically exchange very lightweight beacon messages. The H-sensors report any changes in their clusters to the BS, for example, when a L-sensor leaves or joins the cluster. The BS creates a list of legitimate nodes, M, and updates the status of the nodes when an anomaly node or node failure is detected. The BS assigns each node a unique identifier. A L-sensor nLi is uniquely identified by node ID Li whereas a H-sensor nHj is assigned a node ID Hj . A Key Generation Center (KGC), hosted at the BS, generates public system parameters used for key management by the BS and issues certificateless public/private key pairs for each node in the network. In our key management system, a unique individual key, shared only between the node and the BS is assigned to each node. The certificateless public/private key of a node is used to establish pairwise keys between any two nodes. A cluster key is shared among the nodes in a cluster.

B. Adversary Model and Security Requirements

- Compromise-Resilience: A compromised node must not affect the security of the keys of other legitimate nodes .In other words, the compromised node must not be able to reveal pairwise keys of non-compromised nodes. The compromise-resilience definition does not mean that a node is resilient against capture attacks or that a captured node is prevented from sending false data to other nodes, BS, or cluster heads.
- Resistance Against Cloning and Impersonation: The scheme must support node authentication to protect against node replication and impersonation attacks.
- Forward and Backward Secrecy: The scheme must assure forward secrecy to prevent a node from using an old key to continue decrypting new messages. It must also assure backward secrecy to prevent a node with the new key from going backwards in time to decrypt previously exchanged messages encrypted with prior keys. Forward and backward secrecy are used to protect against node capture attacks.

- OVERVIEW OF THE CERTIFICATELESS EFFECTIVE KEY MANAGEMENT SCHEME

In this paper, we propose a Certificateless Key Management scheme (CL-EKM) that supports the establishment of four types of keys, namely: a certificateless public/private key pair, an individual key, a pairwise key, and a cluster key. This scheme also utilizes the main algorithms of the CL-HSC scheme [13] in deriving certificateless public/private keys and pairwise keys. We briefly describe the major notations used in the paper (See Table I), the purpose of these keys and how they are setup.

A. Types of Keys

- Certificateless Public/Private Key: Before a node is deployed, the KGC at the BS generates a unique certificateless private/public key pair and installs the keys in the node. This key pair is used to generate a mutually authenticated pairwise key.

- Individual Node Key: Each node shares a unique individual key with BS. For example, a L-sensor can use the individual key to encrypt an alert message sent to the BS, or if it fails to communicate with the H-sensor. An H-sensor can use its individual key to encrypt the message corresponding to changes in the cluster. The BS can also

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 4, April 2018

use this key to encrypt any sensitive data, such as compromised node information or commands. Before a node is deployed, the BS assigns the node the individual key.

- Pairwise Key: Each node shares a different pairwise key with each of its neighboring nodes for secure communications and authentication of these nodes. For example, in order to join a cluster, a L-sensor should share a pairwise key with the H-sensor. Then, the H-sensor can securely encrypt and distribute its cluster key to the L-sensor by using the pairwise key. In an aggregation supportive WSN, the L-sensor can use its pairwise key to securely transmit the sensed data to the H-sensor. Each node can dynamically establish the pairwise key between itself and another node using their respective certificateless public/private key pairs.
- Cluster Key: All nodes in a cluster share a key, named as cluster key. The cluster key is mainly used for securing broadcast messages in a cluster, e.g., sensitive commands or the change of member status in a cluster. Only the cluster head can update the cluster key when a L-sensor leaves or joins the cluster.

- THE DETAILS OF CL-EKM

- Pairwise Key Generation:

After the network deployment, a node may broadcast an advertisement message to its neighborhood to trigger the pairwise key setup with its neighbors. The advertisement message contains its identifier and public key. At first, two nodes set up a long-term pairwise master key between them, which is then used to derive the pairwise encryption key. The pairwise encryption key is short-term and can be used as a session key to encrypt sensed data.

1) Pairwise Master Key Establishment: In this paragraph, we describe the protocol for establishing a pairwise master key between any two nodes nA and nB with unique IDs A and B, respectively.We utilize the CL-HSC scheme [13] as a building block. When nA receives an advertisement message from nB, it executes the following **encapsulation** process to generate a long-term pairwise master key KAB and the encapsulated key information, $\phi A = (UA, WA)$.

• Choose $lA \in_R Z^* q$ and compute $UA = lAP$.
• Compute $TA = lA \cdot h0(B, RB, PB)Ppub + lA \cdot RB \bmod q$ $KAB = h1(UA, TA, lA \cdot PB, B, PB)$
• Compute $h = h2(UA, \tau A, TA, A, PA, B, PB)$ $h\_ = h3(UA, \tau A, TA, A, PA, B, PB)$ $WA = dA + lA \cdot h + xA \cdot h\_$ where τA is a random string to give a freshness.
• Output KAB and $\phi A = (UA, WA)$. Then, nA sends A, pkA, τA and ϕA to nB. nB then performs **decapsulation** to obtain KAB.
• Compute $TA = dB \cdot UA$. Note: Because of $dB = rB + x \cdot h0(B, RB, PB)$ and $UA = lAP \bmod q$, TA is computed as $TA = (rB + x \cdot h0(B, RB, PB)) \cdot lAP \bmod q = lA \cdot h0(B, RB, PB)Ppub + lA \cdot RB \bmod q$,
• Compute $h = h2(UA, \tau A, TA, A, PA, B, PB)$ and $h\_ = h3(UA, \tau A, TA, A, PA, B, PB)$.
• If $WA \cdot P = RA + h0(A, RA, PA) \cdot Ppub + h \cdot UA + h\_ \cdot PA$, output $KAB = h1(UA, TA, xB \cdot UA, B, PB)$. Otherwise,

2) Pairwise Encryption Key Establishment: Once nA and nB set the pairwise master key KAB, they generate an HMAC of KAB and a nonce $r \in_R Z^* q$. The HMAC is then validated by both nA and nB. If the validation is successful, the HMAC value is established as the short-term pairwise encryption key kAB. The process is summarized below:

• nB chooses a random nonce $r \in_R Z^* q$, computes $kAB = HMAC(KAB, r)$ and $C1 = EkAB (r, A, B)$. Then, nB sends r and C1 to nA.
• When nA receives r and C1, it computes $kAB = HMAC(KAB, r)$ and decrypts C1. Then it validates r , A and B and if valid confirms that nB knows KAB and it can compute kAB.

- Key Update

In order to protect against cryptanalysis and mitigate damage from compromised keys, frequent encryption key updates are commonly required. In this section we provide the pairwise key update and cluster key update operations.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 4, April 2018

1) Pairwise Key Update: To update a pairwise encryption key, two nodes which shared the pairwise key perfor Pairwise Encryption Key Establishment process. On the other hand, the pairwise master key does not require periodical
updates, because it is not directly used to encrypt each session message. As long as the nodes are not compromised, the pairwise master keys cannot be exposed. However, if a pairwise master key is modified or needs to be updated according to the policy of the BS, the Pairwise Master Key Establishment process must be executed.

2) Cluster Key Update: Only cluster head H-sensors can update their cluster key. If a L-sensor attempts to change the cluster key, the node is considered a malicious node. The operation for any j th cluster is described as follows:

1) nHj chooses x_ j ∈ R Z ∗ q and computes a new cluster key GK_ j = HMAC(x_ j , Hj). nHj also generates an Update message including HMAC(GK_ j ,Update) and computes C6 = EGKj (GK_ j ,HMAC(GK_ j ,Update)). Then, nHj transmits Update and C6 to its cluster members.

2) Each member nLi decrypts C6 using the GKj , verifies HMAC(GK_ j ,Update) and updates a cluster key as GK_ j . Then, each nLi sends the acknowledgement message to nHj

- Key Revocation

We assume that the BS can detect compromised L-sensors and H-sensors. The BS may have an intrusion detection system or mechanism to detect malicious nodes or adversaries [19], [20]. Although we do not cover how the BS can discover a compromised node or cluster head in this paper, the BS can utilize the updated node status information of each cluster to investigate an abnormal node. In our protocol, a cluster head reports the change of its node status to the BS, such as whenever a node joins or leaves a cluster. Thus, the BS can promptly manage the node status in the member list, M. For instance, the BS can consider a node as compromised if the node disappears for a certain period of time. In that case, the BS must investigate the suspicious node and it can utilize the node fault detection mechanism introduced in [21] and [22]. In this procedure, we provide a key revocation process to be used when the BS discovers a compromised node or a compromised cluster head. We denote a compromised node by nLc in the j th cluster for a compromise node case and a compromised head by nHj for a compromise cluster head case.

1) Compromised Node: The BS generates a CompNode message and a EK0H (CompNode, Lc). Then it sends EK0H j
(CompNode, Lc) to all nHj , (1 ≤ j ≤ N2). After all H-sensors decrypt the message, they update the revocation list of their clusters. Then, if related keys with nLc exist, the related keys are discarded. Other than nLc , nHj performs the Node leave operations to change the current cluster key with the remaining member nodes.

2) Compromised Cluster Head: After the BS generates a CompHeader message and a EK0L i (CompHeader, Hj), it sends the message to all nLi (1 ≤ i ≤ n) in the j th cluster. The BS also computes EK0H i (CompHeader, Hj), (1 ≤ i ≤ N2, i _= j) and transmits it to all H-sensors except nHj. Once all nodes decrypt the message, they discard the related keys with nHj . Then, each nLi attempts to find other neighboring cluster heads and performs the Join other cluster steps of the Node join process with the neighboring cluster head. If some node nLi is unable to find another cluster head node, it must notify the BS by sending EK0L i (FindNewClusteLi). The BS proceeds to find the nearest cluster head nHn for nLi and connect nHn with nLi . Then, they can perform the Join other cluster steps.

III. PERFORMANCE ANALYSIS

- End to end delay: It refers to the time taken for a packet to reach source to target over a network.
- Packet delivery ratio: It means difference between packets sent and packets received.
- Energy consumption: It can be defined as average energy consumed on ideal sleep transmits and received to total energy consumed.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 4, April 2018

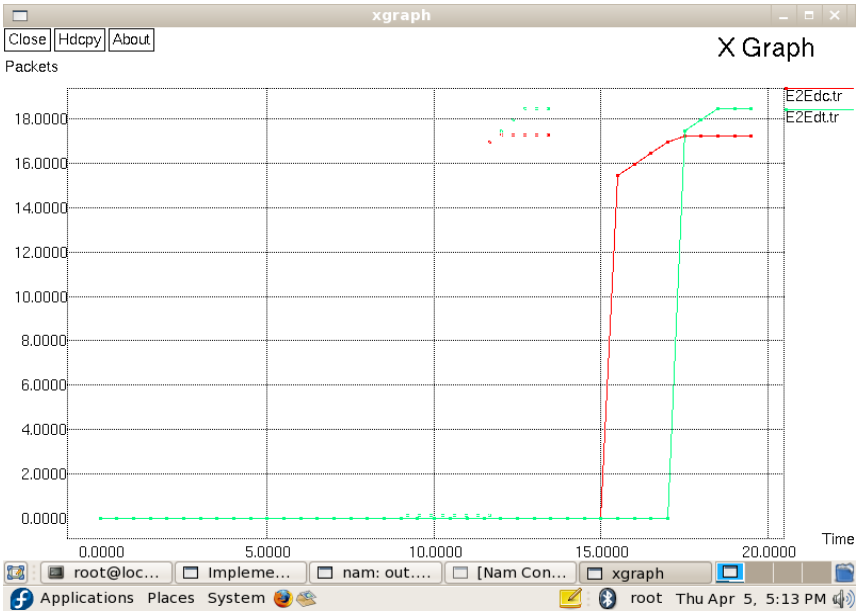

FIG 1. Comparison of End to End delay

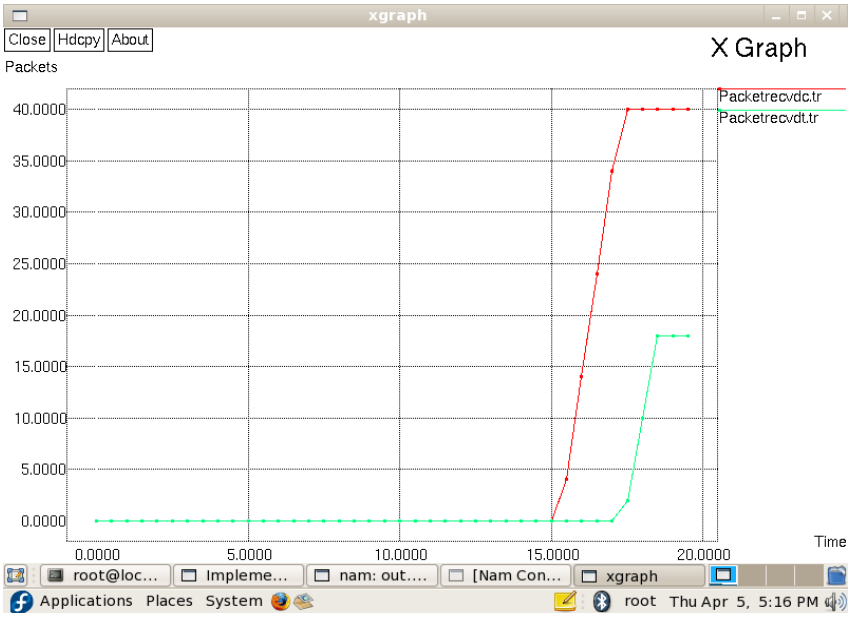

FIG 2. Comparison of Packet Delivery Ratio

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 4, April 2018

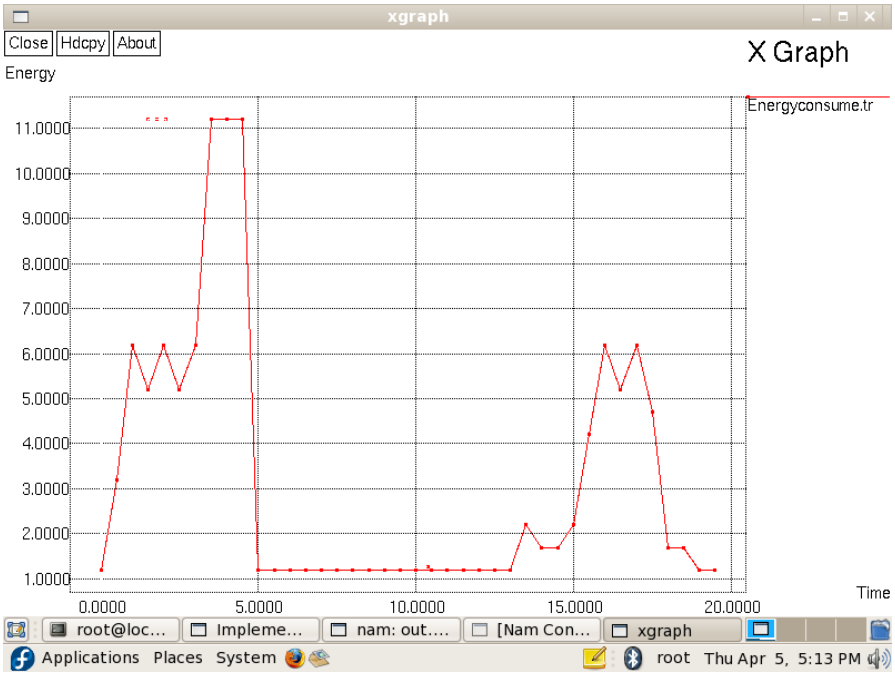

FIG 3. Energy Consumption

IV. CONCLUSION

In this paper, we propose the first certificateless effective key management protocol (CL-EKM) for secure communication in dynamic WSNs. CL-EKM supports efficient communication for key updates and management when a node leaves or joins a cluster and hence ensures forward and backward key secrecy.Our scheme is resilient against node compromise, cloning and impersonation attacks and protects the data confidentiality andintegrity.The experimental results demonstrate the efficiency of CL-EKM in resource constrained WSNs. As future work, we plan to formulate a mathematical model for energy consumption, based on CL-EKM with various parameters related to node movements.

REFERENCES

[1] Seung-Hyun Seo, Member, IEEE, Jongho Won, Student Member, IEEE, Salmin Sultana, Member, IEEE, and Elisa Bertino, Fellow, IEEE, "Effective Key Management in Dynamic Wireless Sensor Networks", IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 10, NO. 2, FEBRUARY 2015

[2] G. de Meulenaer, F. Gosset, O.-X. Standaert, and O. Pereira, "On the energy cost of communication and cryptography in wireless sensor networks," in Proc. IEEE Int. Conf. Wireless Mobile Comput., Oct. 2008, pp. 580–585.

[3] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A key predistribution scheme for sensor networks using deployment knowledge," IEEE Trans. Dependable Secure Comput., vol. 3, no. 1, pp. 62–77, Jan./Mar. 2006.

[4] I.-H. Chuang, W.-T. Su, C.-Y. Wu, J.-P. Hsu, and Y.-H. Kuo, "Twolayered dynamic key management in mobile and long-lived clusterbased wireless sensor networks," in Proc. IEEE WCNC, Mar. 2007, pp. 4145–4150. Prof. A.B.Raut *et al*, International Journal of Computer Science and Mobile Computing, Vol.6 Issue.2, February- 2017, pg. 36-40 © ***2017, IJCSMC All Rights Reserved 40***

[5] Seyed Hossein Erfani1, Hamid H. S. Javadi2, and Amir Masoud Rahmani," Analysis of Key Management Schemes in Dynamic Wireless Sensor Networks", ACSIJ Advances in Computer Science: an International Journal, Vol. 4, Issue 1, No.13 , January 2015

[6] Sagar D. Dhawale Dr. B. G. Hogade Dr. S. B .Patil ," Design and Implementation of a Dynamic Key Management Scheme for Node Authentication Security in Wireless Sensor Networks", International Journal of Science, Engineering and Technology Research (IJSETR), Volume 4, Issue 4, April 2015

[7] Syed Muhammad Khaliq-ur-Rahman Raazi and Sungyoung Lee†,"A Survey on Key Management Strategies for Different Applications of Wireless Sensor Networks," Journal of Computing Science and Engineering, Vol. 4, No. 1, March 2010