Framework for Key Management Scheme in Heterogeneous Wireless Sensor Networks

Kamal Kumar
M.M. Engineering College, Mullana, Ambala, Haryana, India
kkmishra76@yahoo.co.in

[1]A. K. Verma and [2]R. B. Patel
[1]CSED, Thapar University, Patiala, India.
[2]CSED, DBCR State University, Murthal, India.
[1]akverma@thapar.edu, [2]patel_r_b@yahoo.com

Abstract— **Ubiquitous computing environments find their practical implementations through wireless sensor networks, which sense a relationship among themselves and the environment. Presently devised key management schemes are classified namely for homogeneous environments, or heterogeneous environments.In this paper, we propose a deployment conscious security framework supporting, a shift of complex operations to more capable nodes of heterogeneous environment and relieving resource constrained generic sensor nodes of major activities. We introduced a concept of deployment knowledge independent group key generation using a special kind of heterogeneity-multilevel transmission. Performance of proposed key management schemes is evaluated across relevant matrices and concluded to be satisfactory. Findings show that asymmetric key cryptography is comparatively more demanding in resources than symmetric version but ensures maximum security. Through our work we able to conclude that a hybrid of asymmetric and symmetric key cryptography best suits heterogeneous environments.**

Index Terms— **heterogeneous, asymmetric keys, wireless sensor networks**

I. Introduction

Wireless sensor networks (WSNs) are commonly used in ubiquitous and pervasive applications such as military, homeland security, health-care, and industry automation. WSNs consist of numerous small, low-cost, independent sensor nodes, which have limited computing and energy resources.

These systems have traditionally been composed of a large number of homogeneous nodes with extreme resource constraints. This combination of austere capabilities and physical exposure make security in sensor networks an extremely difficult problem. In WSN, the medium of communication is wireless, which is inherently insecure. Thus, each sensor node must know one or more keys to secure its communication. Furthermore, situations might arise wherein an authenticated node is compromised by the intruder, revealing partial or entire keying information to the intruder – making it necessary to remove such node from the network. Needless to say, that the robustness of a security framework relies upon the strength of its key management schemes.

Until now asymmetric encryption or PKI is not seen practical solution in this environment, a number of clever symmetric-key management schemes have been introduced. But Certainly with increasing technological and economical advances comparatively better sensor nodes are available with bit more Energy, Memory and comparatively better computing capabilities. Thus requirements of PKI are not a burden on the WSN.

One well received solution that has been extended by several researchers is to pre-distribute a certain number of randomly selected keys in each of the nodes throughout the network [9], [4], [7], [10]. Using this approach, one can achieve a known probability of connectivity within a network. These previous efforts have assumed a deployment of homogeneous nodes and have therefore suggested a balanced distribution of random keys to each of the nodes to achieve security. Likewise, the analysis of those solutions relies on assumptions specific to a homogeneous environment. A deviation from the homogeneous system model has been increasingly discussed in the research community. Instead of assuming that sensor networks are comprised entirely of low-ability (PKI Compliant) nodes, a number of authors have started exploring the idea of deploying a heterogeneous mix of platforms and harnessing the available “microservers” for a variety of needs. For example, Mhatre et al. [17] automatically designate nodes with greater inherent capabilities and energy as cluster heads in order to maximize network lifetime.

In this paper, we propose a Hybrid Key Management (HKM) scheme for heterogeneous wireless sensor networks. A variation of HKM supporting (Public Key Infrastructure) PKI named HKM-P (Public) and HKM-H(Hybrid) a hybrid scheme supporting both symmetric and public key for divided communications. Even without using deployment knowledge, we harness Location Dependence to generate a group key among geographical neighboring nodes. The rest of the paper is organized as follows. Section 2 and Section 3 discusses the proposed network’s model, network deployment and clustering approach. In section 4 HKM and its variants

doi:10.4304/jetwi.3.4.286-296

has been discussed with section 5 and 6 discuss performance related issues. Section 7 reviews the state of art in key management in WSN. Finally paper concluded in section 8.

II. Network Elements

Basically, two architectures are available for wireless networks, distributed flat architecture and hierarchical architecture. The former has better survivability since it does not have a single point of failure, and the latter provides simpler network management, and can help further reduce transmissions. As we know, WSNs are distributed event-driven systems that differ from traditional wireless networks in several ways such as extremely large network size, severe energy constraints, redundant low-rate data, and many-to-one flows. It is clear that in many sensing applications, connectivity between all Sensor Nodes (SNs) is not required but some applications require explicit connectivity between every pair of nodes. Mostly wireless SNs merely observe and transmit data to those nodes with better routing and processing capabilities, and do not share data among themselves. Data centric mechanisms should be performed to aggregate redundant data in order to reduce the energy consumption and traffic load in WSNs (out of scope of our proposal). Therefore, the hierarchical heterogeneous network model has more operational advantages than the flat homogeneous model for WSNs with their inherent limitations on power and processing capabilities [11][12][13][8] and [12]. Moreover recent trend is towards secure connectivity between geographical neighboring nodes. This phenomenon requires of Group Key which is shared symmetric key among a group of neighboring nodes.

In this paper, we focus on large-scale HWSNs with the same three-tier hierarchical architecture as in [2] [3]. SNs are divided into two categories namely H-Sensors and L-Sensors. H-Sensors are small number of SNs possessing higher memory, transmission range, multiple transmission ranges, processing power and battery life. Our network model has four different kinds of wireless devices on the basis of functionality; sink node/base station (BS), cluster head node (CH), Anchor Nodes (AN) and sensor node (SNs).

- *Sensor node* **(**SNs**):** Sensor nodes are new generation L-Sensors which are inexpensive, limited-capability, generic wireless devices. Each SNs has limited battery power, memory size, data processing capability and short radio transmission range. SNs communicate with its CH, *cluster* SNs and $SINK$. These are assumed to be capable enough to support the PKI. We propose to store two different encryption algorithms i.e. one for asymmetric key cryptography and one for symmetric key cryptography. We propose to use Elliptical Key Cryptography (ECC) for asymmetric key and Advance Encryption standard (AES) for symmetric key.
- *Cluster head node* (CH): Cluster head nodes are a kind of H-Sensors, have considerably more resources than the SNs. Equipped with high power batteries, large memory storages, powerful antenna and data processing capacities(not exploited in this paper). CHs can execute relatively complicated numerical operations and have much longer radio transmission range than SNs. CHs can communicate with each other directly and relay data between its cluster members and the $SINK$. SNs which need to communicate with neighbors in neighboring cluster will relay its data through CHs. CHs are responsible for dividing SNs into clusters of uniform size.
- *Anchor Nodes* **(**ANs**):** Anchor Nodes are a special kind of H-Sensors which have multiple power level for transmission. Thus ANs have capability to transmit in multiple ranges which can be changed at requirement. ANs are placed at triangular/Hexagonal points to realize a new grouping approach.

We introduce a new geographical/Location based grouping of nodes with the help of ANs but without using deployment information. Grouping is post deployment affair in our proposal and thus besides cluster communication we can achieve group communications which finally can be exploited to avoid transmission of redundant data gathered by the geographically neighboring nodes.

Sink node/Base station **(**$SINK$**):** Sink node is the most powerful node in a WSN, it has virtually unlimited computational and communication power, unlimited memory storage capacity, and very large and powerful radio transmission range which can reach all the SNs in a WSN. Sink node can be located either in the center or at a corner of the network based on the application. For PKI $SINK$ node works as Certificate Authority (CA) which can perform the function of generating public-private key pairs for various nodes in the HWSN. $SINK$ node also performs the function of revoking nodes and CHs and redistribute the keys as and when needed. $SINK$ is responsible for maintaining/publishing Central Revocation Lists (CRLs).

In our network model, a large number of SNs are randomly distributed in an area. A sink node ($SINK$) is located in a well-protected place and takes charge of the whole network's operation. After the deployment, ANs partitions the nodes into groups and CHs partition a WSN into several distinct clusters by using a clustering algorithm discussed ahead. Each cluster is composed of CH and set of SNs (distinct from other sets). SNs monitor the surrounding environment and transmit the sensed readings to their respective CH for relay. SNs may use multi-hop or single hop communication pattern for communication with CHs .

III. Grouping and Clustering Approach

SNs are large in number and have limited capabilities. SNs are deployed randomly in the field for deployment like can be dropped from an aircraft. H-nodes which would bear Cluster heads responsibility are also deployed randomly. ANs are placed uniformly and in controlled manner using a manned or unmanned deployment vehicle which is equipped with GPS system to connect with satellite to retrieve exact location for ANs . Using hexagonal/triangular deployment of ANs in the deployment field the network deployment field is roughly divided into *hexagonal/triangular* field using multiple transmission power levels of ANs . As shown in the Figure 1 the lines in dark are transmission radius of ANs placed at triangular points. The higher is the transmission level larger is the transmission radius. Depending upon the number of ANs whose transmission ranges are aligned/covering a small area completely, SNs in that area/cell will receive the equivalent number of nonce, considering that each transmission level of a AN transmits an entirely different nonce. In Figure1 we assumed that intersection of transmission levels results in approximately triangular area/cell thus nodes in a triangular area/cell will receive nonce from the ANs whose transmission level is covering the triangular cell completely. For e.g. Nodes in Blue Cluster receives Selected Nonce but from all the ANs . Each SN in Triangular cell in blue colored cluster receives different set of nonce. Nodes in Triangular cell closer to $AN2$ receives N_{24} , N_{25} , N_{26} from $AN2$, N_{15} , N_{16} from $AN1$, N_{34} , N_{35} , N_{36} from AN_3 , N_{45} , N_{46} from $AN4$, N_{55} , N_{56} from $AN5$ and N_{65} N_{66} , from $AN6$. SNs in the same area/cell receive the same set of nonce thus grouped into same geographical group. Similarly SNs in other adjoining areas/cells receives nonce depending upon their location in the field.

Considering the placement of Nodes as shown in the figure 1, $AN1$, $AN2$ and AN_3, $AN4$, $AN5$ and $AN6$ are able to transmit at different power level and thus can transmit in multiple ranges. We here assume that the Anchor Nodes are able to transmit at six power levels in Figure 1. All this implies that more number of transmission levels results in lower density groups and vice-versa with similar information received.

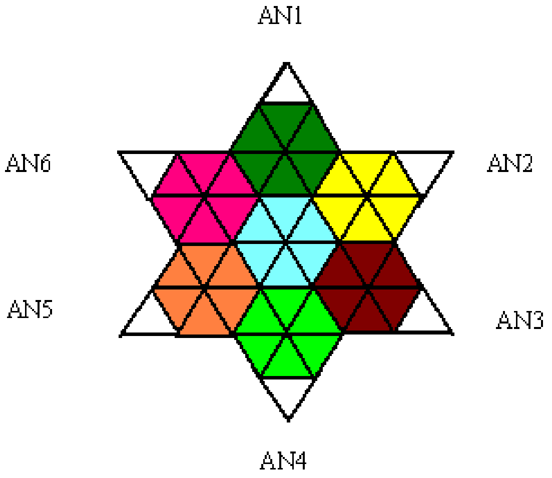

Figure 1: Hexagonal deployments of ANs and Resultant Hexagonal Clusters. For sake of convenience the circular arcs are approximated as straight lines. Transmission ranges from closely placed Anchor Nodes at six corners intersect with each other and resulting into triangular shaped cells. Adjoining cells may be joined to give a hexagonal shaped clusters which are supposed to managed by cluster Heads.

A. CHs Discovery: CHs now can start discovering their neighbor CHs to form second tier in the hierarchical organization of HWSN. *CHs* broadcasts CH-Discover (CH_D) signal which is received by neighboring *CHs*. SNs which receives CH_D replies with CH-Discover Reply (CH_DR) with their $ID.ID_s$ of *CHs* in the CH_DR are sent by the *CH* to $SINK$ which verifies the ID against Database maintained and replies with verification details. If authentic CH , $SINK$ will send some information of corresponding CH ; particular to HKM variant. Otherwise $SINK$ updates its Central Revocation List (CRL) and dispatches the same to all the *CHs* and SNs in the network. Having verified now *CH* can add another *CH* as its neighbor and maintains a neighbor list which can be used for routing decisions.

B. SNs Discovery: CHs Broadcast a Join Request $(J\operatorname{Re}P)$ in its transmission range at some intervals for maximum reception and also not to collide with neighboring *CHs*. SNs receiving $J\operatorname{Re}Q$ replies with Join RePly $(J\operatorname{Re}P)$containing their ID_s. SNs receiving $J\operatorname{Re}Q$ from more than one *CHs* will reply to multiple *CHs* but one who will administer the communications of this SN is one which is in circle with radius D centered at SN . This helps us achieve a uniform coverage cluster. The implementation of this clustering scheme is under process.

As each SN is replying with its ID ;*CHs* forwards ID of SNs to $SINK$ for Authentication against legal SNs Database. If SN is legal $SINK$ will send the information pertaining to a particular variant of HKM. Otherwise $SINK$ Updates its CRL and initiates revocation of the SN. *CHs* stores information sent by $SINK$ for future reference. *CHs* now reply Join Completed to the concerned SNs and some information pertaining to HKM variant.

C. Geographical Group Key Generation ($K_G^{i,j}$): Sensor nodes in the same geographical group i.e. triangular cell, can construct a group key $K_G^{i,j}$ using the broadcast received from AN_s. But all the nodes receiving the same set of nonce need not be covered or administered by the same *CH*. Thus we can have some members be administered by one *CH* and some by other *CH*. Now *CH* in their coverage areas may have nodes with different set of nonce but nodes with same set of nonce co-located. This is the strength of our proposal that without using any deployment information we are able to harness location dependent group of nodes. We are now able to have cluster with possibly several sets of nodes with same nonce thus geographical and logically related nodes. This small set of nodes may be called to form a group. Using the information from *CH* and nonce; nodes in a cluster will form several groups of co-located nodes. Thus we have several groups with different group key with in the same cluster because of different set of nonce and different from groups in neighboring cluster because of different information from their *CHs*. Following equation can be used to generate group key:

$$K_G^{i,j} = H_{K_M}\left(k_{11}, k_{12,}, ..., k_{ij}, ..., S_{G_n}\right)....(1)$$

where K_{ij}'s are key broadcast from AN_i and transmitting at j^{th} power or transmission level, S_G is the seed obtained from *CH* for generation of $K_G^{i,j}$ i.e. group key for j^{th} group in i^{th} cluster. Moreover S_G is the same used for generation of K_{C_i}. H_{K_M} is the keyed hash function which uses K_M; pre-deployed Master Key.

IV. Security Framework

In existing key pre-distribution schemes, two communicating sensors either use one or some of their shared pre-loaded keys directly as their communication key [15][10], or compose a communication key by their pre-loaded secret shares. Although this kind of mechanism has low computational overhead, it could lead to a serious security threat in practice. If some SNs are captured after the deployment, an adversary may crack some or even all the communication keys in the network by those compromised keys or secret shares. This node capture attack is the main threat to a key pre-distribution scheme. To address the limitations of existing key pre-distribution schemes, we proposed to incorporate the location dependence with pre-distribution. Our proposed framework supports three schemes. HKM is proposed in three variants namely HKM, HKM-P (HKM- Public) and HKM-H (HKM- Hybrid).

A. HKM: Before SNs are deployed, setup keys need to be pre-loaded into them. Also each node is assigned with ID. Each sensor node is pre-programmed according to the application requirements for deployment. At the same time, one unique Key (k_{SN_i-SINK}) of size m bit and master key (K_M) of size M bits is written in FLASH ROM of each node. The reason for storing K_M in FLASH ROM instead of hard coding is to utilize this information for later purging the information in corrupted/compromised nodes. $SINK$ stores [ID, k_{SN_i-SINK}] pair for each node and uses it to authenticate and establish identity for each sensor node at the time of node joining in the network. Besides this $SINK$ also stores routing keys (K_{CH}) and cluster keys (K_{Ci}) in a database for an epoch.

Key Assignment and Distribution: Since a single key is inappropriate for securing all communication in a sensor network, our framework supports establishment of three different keys. This helps in limiting the impact of any key's compromise to only a certain number of nodes. $k_{SNi-SINK}$ is a unique pair-wise key of SN_i with the $SINK$. $SINK$ use this key to communicate any interest directly to that SN. K_{CH} is used by *CHs* to communicate with $SINK$ and other CHs. Any *CH* can reach $SINK$ indirectly using K_{CH}. Our next work is to propose a routing protocol which is aware of underlying key management scheme where we will use K_{CH}. k_{CE} is used by CN of i^{th} cluster to communicate with their *CH* and other members of their cluster. Group key ($K_G^{i,j}$) generation is discussed in section 3.3 above which is used for communication among SNs within a group.

After formation of clusters, the $SINK$ broadcasts key generation seed S_{CH} to *CHs*. Each *CH* then computes K_{CH} using a function of S_{CH} and K_M. Once S_{CH} is generated, each *CH* generates a seed S_{C_i} different from S_{CH} and specific to a cluster location and broadcasts it to its CN_s. Each Node of i^{th} cluster then computes K_{C_i} by a function of S_{C_i} and K_M. Initially no SN is physically compromised, an intruder eavesdropping into bootstrap communication; unable to produce any keys as it doesn't have information about K_M . Our scheme generates and distributes keys simultaneously, unlike [16] and [10] which require separate key distribution algorithms for key allocation.

Key Generation: For our scheme, we will use key generation algorithm as proposed in [27]. This algorithm is combination of pre and post-deployment key generation mechanisms. It generates K_{CH} and K_{C_i} and used for specific epochs and are removed from SN after that epoch. These keys will also be generated during re-keying or revocation.

a. Re-keying: Re-keying is optional instead of an obligation. The re-keying helps in refreshing or replacing security keys. It is needed when a compromised node needs to be excluded from the system, or a clustering is required. Re-keying after session expiry and new cluster formation is same as explained in key assignment and distribution section. Keys assigned in last session can now be used for secure seed propagation during re-keying if uncompromised. In the situation of a node compromise, $SINK$ first performs a node revocation REVOKE operation, on the node which is malicious using its $k_{SN_i - SINK}$. The operation is explained in next section. REVOKE clears the keys stored in the FLASH RAM of the node and making it impossible to get new communication keys. For extremely security critical application re-keying could be performed. If the compromised node was CN , then re-keying is required only within cluster and can be done by just generating new K_{C_i} and $K_G^{i,j}$ using new S_{C_i} . In case of CH compromise, re-clustering will be required to updates K_{CH} , $K_G^{i,j}$ and K_{C_i} .

b. Node Revocation: A compromised node reveals all the information to the intruder, including the master key and leaves network vulnerable. A counter control operation must, be triggered to let the network become aware of the problem and resolve it. Such a problem cannot be circumvented using temporary mechanisms such as assigning new keys only to the compromised node and/or its

In HKM Framework we proposed REVOKE operation to be triggered after detection of the compromised/malicious node. For Pre REVOKE operation; existing schemes like [28] can be utilize. $SINK$ knows the keying information in the compromised node. It unicasts REVOKE (step 3) message masked as a normal primitive operation just before the epoch for the next key update phase (step 4). The target node executes the command. This is followed by the resetting of the FLASH ROM, deleting compromised keying information of the sensor node. The RAM is totally reset. At least another epoch passes (step 5), before the compromised node and its intruder recover from the REVOKE that struck. It may take up to several epochs before the sensor board cold starts, initializes variables, and intruder re-deploys the keys (step 6). The sensor node cannot assign itself a new key during the subsequent re-keying processes as it cannot decrypt the new session key (K_{CE_3}) using previous key (K_{CE_1}) and the master key. Hence, such a node is isolated from the network (figure 2).

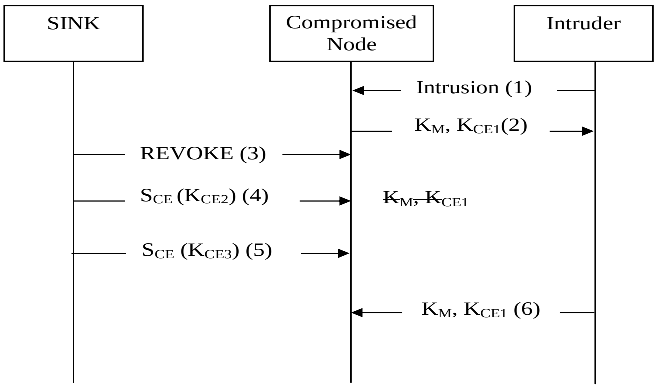
SINK
Compromised Node
Intruder
Intrusion (1)
K_M, K_{CE1}(2)
REVOKE (3)
S_{CE} (K_{CE2}) (4)
~~K_M, K_{CE1}~~
S_{CE} (K_{CE3}) (5)
K_M, K_{CE1} (6)

Figure 2: REVOKE Operation communicating neighbors. Instead of assigning new keys to neighbors or compromised nodes only, more concrete solutions are needed to isolate such nodes from network.

B. HKM-P (public): HKM-P is public key infrastructure (PKI) based. Two keys per node (one private key K_{pub} and one public key K_{pub}) are needed here, as required in PKI besides one group key generated using unique location dependent broadcast.

Key Generation and Assignment: A public/private key pair is generated for each node using Elliptical Curve Cryptography (ECC) prior to network deployment. Base station maintains [ID_j , K_{pub}] pair for each SN and each node is pre-installed with its respective private key K_{pri} , prior to network deployment.

Key Distribution: At network setup, each SN sends $j\operatorname{Re}q$ message as well as encrypted with K_{pri} to the nearest *CH*. As *CH* unable to decrypt the message therefore forwards this message to $SINK$. $SINK$ verifies the node's authenticity by decrypting the message with the public key registered against that node ID stored in its database. If node is authentic, an OK message is sent to the *CH*; otherwise a REVOKE message is directed to the *CH*. If the *CH* receives an OK message, *CH* stores the K_{pub} of the node for future references and adds SN as its CN , otherwise the message is discarded. Having converted all SNs as CN s, $SINK$ broadcast $K_{pub}\left[SINK\right]$ to all the CHs . On receiving $K_{pub}\left[SINK\right]$, each CH broadcast own $K_{pub}\left[CH_i\right]$ to own CN s.

C. HKM-H (Hybrid): Here we try to harness the heterogeneity of nodes. Burden of complex, computationally expensive algorithms is shifted on the devices with extra capabilities i.e. H-nodes. Here we devise to use the asymmetric cryptography during inter-cluster communication and symmetric cryptography for intra-cluster communication. Two types of encryption algorithms are hard-coded on each node: one is public key based encryption algorithm like ECC and the other is symmetric key based encryption algorithm like AES.

HKM-H uses keys similar to HKM but instead of K_M , we have separate public/private key pairs for each CH denoted as $K_{pri}\ [CH_i]$ and $K_{pub}\ [CH_i]$.

Key Assignment and Distribution: In pre-deployment phase each node is provided with unique identity k_{SN_i-SINK} and has both encryption algorithms ECC/AES installed over it. In addition, a master key K_M and basic keying material for ECC is preinstalled on each H-Node only. The key generation process is initiated by the $SINK$, which calculates its public/private key pair and broadcasts its public key, so that each *CH* receives the public key of the $SINK$. Each *CH* after generating its public/private key pair sends its public key along with its ID to $SINK$. $SINK$ saves this public key and also broadcasts it along with $CH's\ ID$ so that all the *CHs* get the public keys of all the other *CHs*. After the completion of the key generation at *CH* level, each cluster head generates a seed and broadcasts it to all the CN s. CN suseseed to generate K_{C_i} and $K_G^{i,j}$.

V. Evaluation Matrices

To determine the efficiency of our key management protocol for HWSN is checked against the following matrices.

Scalability: Our schemes support high scalability. In HKM and HKM-H, whenever a new node wish to enter the network it propagates ID_j and a JOIN message encrypted with k_{SN_i-SINK} to its nearest *CH*. *CH* unable to decrypt the message, routes ID_j and messages to $SINK$; for authentication. $[SINK]$ checks for the node's ID in its database for [ID_j , k_{SN_i-SINK}] pair. It then decrypts the message using k_{SN_i-SINK} . If message decrypted successfully it authenticates the new node as legitimate node. $SINK$ informs the same to *CH*. *CH* propagates session's seed S_{C_i} to the new node allowing the node to join cluster by generating K_{C_i} .

To become a member of the neighboring geographical group node sends request for generation of its group key. Node sends S_{C_i} encrypted with K_M , which neighboring node can decrypt using its own K_M . This node now tries to generate K_{C_i} using received S_{C_i} If newly generated K_{C_i} matches with its own K_{C_i} , it sends $K_G^{i,j}$ key to new node by encrypting with K_M / K_{C_i} . In HKM-P, new node's key generation and registration activities are performed before deployment. After deployment, SN sends JOIN request message encrypted with K_{pri} and its ID_j to closest *CH* which routes the message to $SINK$. Upon authentication by $SINK$, the new node is allowed to be the cluster member and its public key is stored with the $SINK$.

Key Connectivity: Key connectivity is defined as the number of keys needed to be kept per node for specific level of network connectivity. In HKM a common symmetric key K_{CH} for $SINK$-*CH* and *CH-CH* interactions is used which based on session seeds S_{CH} . It uses K_{C_i} for *CH*-CN interaction based on session seeds which is location based i.e. S_{C_i} . This provides 100% network wide connectivity. HKM-H provides good key connectivity on frequent interaction basis. Frequent CN - CN communication is ensured by a common cluster wide symmetric key. Local CN - CN communications are ensured using group keys. Similarly, each *CH* possesses the public keys for all the other *CH's*, hence complete network-wide key connectivity is ensured. In HKM-P communications is via public keys. *CHs* have public keys *CHs* and have public keys of CN and *CH* of their cluster. We are in process to propose the use of group keys in scenario where path key establishment becomes necessary. In actual will be replacement of Path Key. Depending upon the application we may use the group key to reduce the multiple instances of the messages generated from a locality.

Revocation: Each of the three schemes considers the REVOKE operation for the compromised node removal as described in the node revocation section. For HKM-P, although the node compromise does not reveal any important keying information but REVOKE can be used for removal of other CN_s public keys from compromised node. This will prevent compromised node from generating any kind of attacks on CN_s whose public keys it possesses.

For our framework, $SINK$ has been assigned the role of trusted certificate authority (CA) as well as issue certificate revocation list (CRL), containing information about revoked nodes at regular intervals. HKM-P employs hierarchical mode of communication, which makes application of CRL scheme simple. The CRLs are updated whenever a node (*CH/CN*) is revoked after being declared as compromised/malicious. The CRLs are broadcasted periodically from $SINK$ to *CHs*. Each *CH* filters the CRL according to its CN s. According to HKM-P communication architecture, when a *CH* wants to communicate with another *CH* the process takes place via $SINK$ and whenever any CN wants to communicate with another CN, the communication takes place via relevant *CHs*. For the CH-CH communication, CRL is checked at $SINK$ to determine the status of the receiver. If found revoked, the communication request is returned to the sender with an indication that the intended *CH has* been revoked. In a similar manner, for the CN-CN interaction,

Necessary checks are performed at each *CH* on the CRL to check the validity of a node uncompromised status.

Resilience: Network resilience is defined as its resistance against node captures [20]. Resilience has a direct relation with network security i.e. higher resilience of a network means more security. In HKM; if SN compromised reveals its K_{C_i} or K_{CH} and K_M which makes whole network vulnerable. It requires strong methods which could detect node compromise on earliest possible and start REVOKE operation. HKM-H utilizes K_{C_i} for cluster wide communications. For one CN compromise, only CN_s and *CH* of that particular cluster are vulnerable. Any node compromise in HKM-P does not reveal any keying information except its own private key and a few public keys. The maximum harm this compromised node can do is to decrypt messages destined for it. Table 1 gives a comparison of scalability and resilience for the three schemes along with describing the revocation method.

Table 1: Scalability and Resilience

Scheme	**Scalable**	**Revocation**	**Number of keys required to compromise complete network**	**Keys revealed on compromise**
HKM	*Yes*	*REVOKE*	*1*	*CN:* K_{C_i} , K_M ; *CH:* K_{CL}, K_{C_i}, K_M
HKM-P	*Yes*	*REVOKE*	*N+1*	*CN:* $K_{\mathrm{Pr}\,i}[CN_i]$, $K_{Pub}[CH_j]$ *CH:* $K_{\mathrm{Pr}\,i}[CH_j]$, $K_{Pub}[SINK], K_{Pub}[CN_i]$
HKM-H	*Yes*	*REVOKE*	*C keys : 1 from each cluster*	*CN:* K_{C_i} , K_M ; *CH:* $K_{C_i}, K_{\mathrm{Pr}\,i}[CH_j]$, $K_{Pub}[CH_j]$

VI. Performance

The evaluations are based on the simulations carried out in MATLAB environment for the individual implementations of HKM, HKM-H and HKM-P.

We used PROWLER plug-in in MATLAB for analysis (Table 2). Table 3 gives experimental values for memory, energy and time analysis of all three schemes and compared them with [5], as it is the only other key generation and distribution solution based on ECC.

Table 2: Memory and Energy Analysis

		HKM	**HKM-P**	**HKM-H**	**ECC[5]**
Memory Usage	RAM	*12840*	*41480*	*34000*	*34342*
	ROM	*1300*	*2450*	*2000*	*1140*
Energy Usage	Transmiission	*255*	*394*	*317*	-----
	Electronics	*104*	*131*	*129*	-----
	Total	*390*	*500*	*450*	*816*

Table 3: Number of Keys stored per node

	SINK	**CH**	**CN**
HKM	*N+1*	*4*	*4*
HKM-H	*C+N+2*	*C+6*	*4*
HKM-P	*N+1*	*N+2*	*3*

A. *Memory analysis:*

HKM is based on symmetric key cryptography hence occupies the smallest portion of RAM and ROM in the three schemes (Table 2). HKM-P utilizes public key cryptography for all the communications and hence utilizes maximum RAM and ROM space compared to other two schemes. HKM-H uses public key cryptography for $SINK$-*CH* interaction whereas symmetric key cryptography is used for *CH*-CN communication hence it memory usage falls between other two schemes. This memory analysis is done excluding $SINK$ as $SINK$ has no energy/memory constraints and is maintaining public keys for all the nodes involved in the network. The reason that HKM-P takes more memory than [5] being that HKM-P also provides key update and node revocation mechanism in addition to just key generation and distribution provided by [5]. EBS based schemes [15–17] store a key pool (P) of size k+m where k keys are stored per node along with c communication keys. E.g. in [17] key generation nodes store $k+m+1$ keys and other nodes store $k+1$ keys. LEAP [10] stores $3d+2+L$ keys per node where $d =$ number of neighbors and L = number of keys in key chain. Thus we can easily claim from Table 3, that HKM and HKM-H have very less storage requirement for nodes other then $SINK$, which in our case have limitless resources. Although HKM-P consumes little larger memory space but it provides maximum resilience and security.

B. Communication Overhead: For a network of N nodes' having C clusters with n members each, values for message exchange for key setup and re keying are given in Table 4. Table 4 also shows maximum message exchange which is in case of communication between CN of one cluster with CN of other cluster in all three schemes. Table also gives communication overhead for re-keying procedure.

Other dynamic key management systems like LEACH [10] uses $(d-1)^2 / (N-1)$ messages for re-keying [d is number of neighbors]. EBS based schemes [15–17] use minimum m(number of keys not known to compromised node) messages only for transmission of new keys. Number of messages required for generation and assignment of these keys are additional to these m messages. Based on these results can say that our schemes provide optimum solution for storage and communication for key management. Reason is each cluster node select its nearest possible cluster leader so there is not much variation is their distance. The little variation in time for key distribution from $SINK$ to *CH* is due to the distance of *CH* from the $SINK$. In real life scenario this variation could be much noticeable depending on the size of WSN.

Table 4. Message Communication for Key Management Phases

Scheme	**Key Set-up**	**Max. Communication** $\lfloor CN_i - CN_j \rfloor$	**Re-Keying**
HKM	*C+1 broadcast messages ,besides broadcast by* AN_S *for grouping*	*3 encryptions and 3 decryptions*	*3 messages [for CH Removal] and 1 message each at highest transmission levels from* AN_S *for group rekeying.*
HKM-H	*2C+1 Broadcast C unicasts, besides broadcast by* AN_S *for grouping*	*3 encryptions and 3 decryptions*	*4 messages [for CH Removal] and 1 message each at highest transmission levels from* AN_S *for group rekeying*
HKM-P	*C+1 Broadcast Messages, besides broadcast by* AN_S *for grouping*	*3 encryptions and 3 decryptions(for 1st time)* *1 encryption and 1 decryption(next time)*	*1 message each at highest transmission levels from* AN_S *for group rekeying*

Table 5: Energy Consumption Equation for Various Levels of Nodes

Scheme	**CN**	**CH**	**SINK**
HKM	$P_R + P_C$	$P_R + 3P_C + P_T$	$P_T + 2P_C$
HKM-P	$P_R + P_C$	$P_R(C+1) + 3P_C + 2P_T$	$P_R(C) + P_C + P_T(C+1)$
HKM-H	$P_R + P_C + P_T$	$P_R(n+1) + P_C + P_T(n+1)$	$P_R(N-1) + P_C(N-1) + P_T$

C. Energy analysis

a. Simulation Results: For HKM and HKM-H, the energy consumption tends to balance out with the changing number of nodes i.e. there is slight increase in energy consumption with the increase in number of nodes. A linear trend of slight increase is visible in the energy characteristics of HKM and HKM-H. For HKM-P, the energy consumption increases rapidly with the increase in number of nodes. This is because comparatively more number of nodes are involved in the activities of key generation, key registration and key distribution. Thus caused an exponential increase in the energy characteristics of HKM-P which becomes prevalent as the network size increases (Figure 3). In case of HKM and HKM-P, there are no specific energy characteristics corresponding to node role but for HKM-H, the measurements differ in context to node role. For HKM-H the initial hypothesis has been that $SINK$ performs the most energy consuming operations of key management (Figure 4). The role based energy breakdown for HKM-H reveals the same showing that $SINK$ exhibits maximum energy characteristics, *CHs* depict moderate energy characteristics whereas the energy characteristics for CN_s coincide with those specified for HKM.

b. Analytical evaluation: Table 5 gives analytical equations for energy consumption for the proposed schemes. Where P_R is reception energy, P_T is transmission energy and P_C is computation energy. The energy consumed in sending 1 byte is estimated to be 59.2µJ whereas the energy consumed in receiving is specified to be 28.6µJ. Also energy cost of computation is small compared to data transmission. These values are based on actual mote implementation [1] (Table 6). Since we simulated energy for small number of nodes, so here we will calculate energy characteristics of the three schemes. These values indicate that even for larger network energy consumption at *CN* does not increase significantly, except in HKM-P.

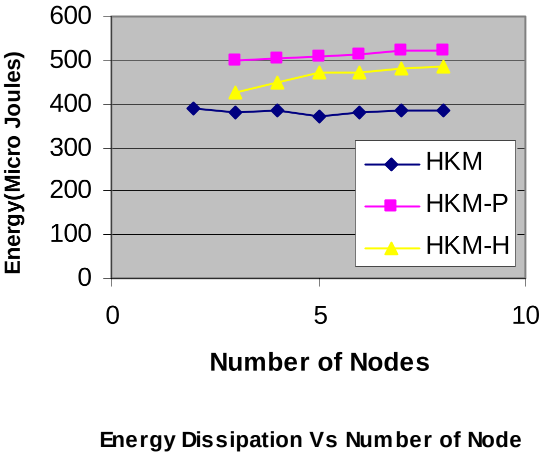

Figure 3: Energy Dissipation Scheme Wise

Table 6: Analytical Values for Energy Consumption

Scheme	CN	CH	SINK	Average
HKM	*143*	*439*	*296*	*292.67*
HKM-P	*286*	*373.8*	*9372*	*4465.33*
HKM-H	*1404.8*	*16400*	*46707.2*	*21504*

For HKM all analytical values remain almost same as noticed in experimental results. Nodes using asymmetric key in HKM-H and HKM-P shows exponential increase in energy consumption with the increase in network size.

VII. Related Work

Typical key-distribution schemes focus on probabilistic key distribution, as in Eschenauer and Gligor [9]. Probabilistic schemes have several undesirable side effects that public-key-based schemes do not: they cannot guarantee that a given node will be able to establish a shared secret with its neighbor(s), and they cannot guarantee security for uncompromised nodes after a number of nodes have been compromised. Research has also shown that Elliptical Curve Cryptography is practical for small sensor nodes [21]. In [22], Du et al. have designed security schemes for HSNs which use public key cryptography.Du *et al.* [7] proposed a key pre-distribution scheme with the objective to improve the resilience of the network if compared to the previous schemes. In [23], Du *et al.* proposed another scheme to utilize node deployment knowledge to improve the Eschenauer-Gligor scheme in [24] in terms of network connectivity, memory usage, and network resilience against node compromise. Their scheme assumes a group based deployment model, in which sensor nodes are deployed in groups around their deployment points and the distribution of deployment points follows a rectangular grid model. In each group, the Eschenauer-Gligor scheme is applied. Zhou *et al.* [9] presented a location-based key establishment scheme, which is a hexagonal-grid-based deployment model combined with a polynomial-based key establishment model to establish a key between two neighboring nodes. We in our proposal introduced the concept of location based key management scheme without using deployment knowledge using special kind of heterogeneity.

VIII. Discussion

We focused on keying methodology in our proposal. The results expectedly show that there exist an inverse relationship between the resource availability and the achievable level of security. cording to the observations, HKM-P utilizes maximum resources but it also provides end to end security and maximum resilience to node compromise, and thus most secure solution. From the resource utilization perspective, the schemes can be arranged from the least expensive to the most expensive

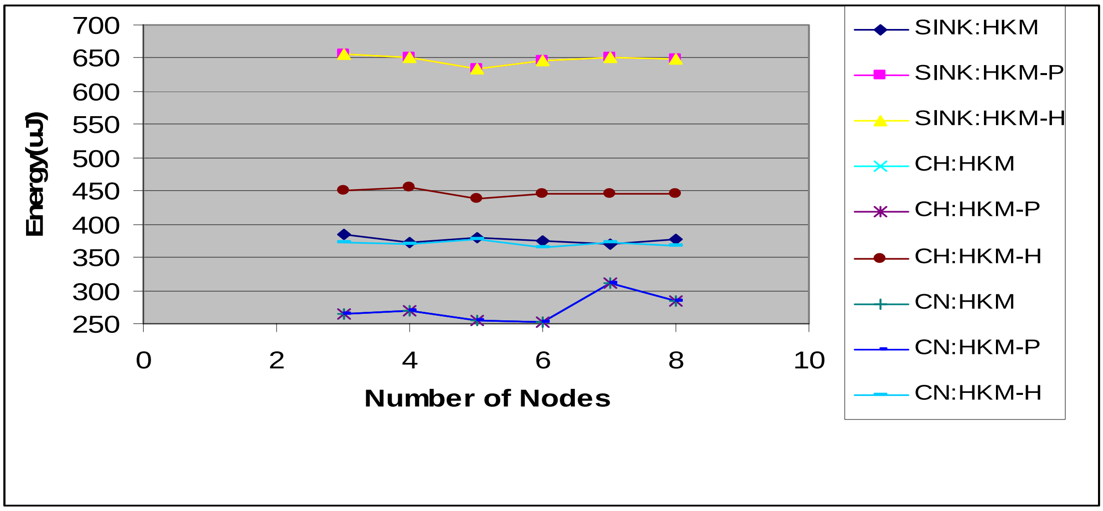

Figure 4: Node Role wise Energy Consumption

as HKM, HKM-H, and HKM-P. From Energy consumption issue HKM and HKM-H show similar energy characteristics. Both follow a linear trend of increase with the increase in the number of nodes. HKM-P is efficient in energy requirements for smaller number of nodes but exhibits exponential increase to support larger number of nodes. From security perspective, the schemes can be arranged from the least favorable to the most favorable as HKM, HKM-H, and HKM-P. Also only HKM-P provides end-to-end security. For example in case of CN_i to CN_j communication, first time message exchange for getting each other public keys will take about four cycles of encryption/decryption. But from next time messages will not need any encryption/decryption on intermediate nodes. In HKM and HKM-H every communication between CN_i to CN_j will require extra encryption/decryption at *CH*

REFERENCES

[1] Wander et al.: Energy analysis of public-key cryptography for wireless sensor networks. In: The Third IEEE International Conference on Pervasive Computing and Communication (PerCom ’05), pp. 325-328, March 2005.

[2] Younis M., Youssef M., Arisha K.:Energy-Aware Routing in Cluster-Based Sensor Networks. In: 10th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS2002), 2002.

[3] Arisha K., Youssef M., Younis M. Energy-Aware TDMA-Based MAC for Sensor Networks. In: IEEE Workshop on Integrated Management of Power Aware Communications, Computing and Networking (IMPACCT 2002), New York(2002).

[4] Chan H. Perrig A., Song D. : Random Key Predistribution Schemes for Sensor Networks. In: IEEE Symposium on Security and Privacy, pp.197, IEEE Computer Society, Washington, DC, 2003.

[5] Malan D. J., Welsh M., Smith M. D.: A public-key infrastructure for keydistribution in TinyOS based on elliptic curve cryptography. In: First IEEE International Conference on Sensor and Ad Hoc Communications and Networks, SantaClara, CA, October 2004.

[6] Firdous K., Sajid H., Laurence T. Y., Ashraf M.: Scalable and efficient key management for heterogeneous sensor network. In: Journal of Supercomput vol. 45, pp. 44–65, 2008.

[7] Du W., Deng J., Han Y. S., Varshney P. K.: A pairwise key pre-distribution scheme for wireless sensor networks. In: 10th ACM conference on Computer and communications security, pp. 42 - 51, Washington D.C., USA, October 27-30, 2003,

[8] Liu B., Liu Z., Towsley S.: On the capacity of hybrid wireless networks. In: IEEE INFOCOM, vol. 2, pp. 1543-1552, San Francisco, CA, April 2003.

[9] Eschenauer L., Gligor V.: A Key Management Scheme for Distributed Sensor Networks. In: 9th ACM Conference on Computer and Communication Security, pp. 41-47, November 2002.

[10] Liu D., Ning P.: Location-based pairwise key establishments for static sensor networks. In: 1st ACM Workshop on Security of Ad Hoc and Security of Ad Hoc and Sensor Networks in Association with 10th ACM Conference on Computer and Communications Security, pp. 72–82, Fairfax, Va, USA, October 2003,.

[11] Gupta P., Kumar P.: Internets in the sky: The capacity of three dimensional wireless networks. In: Communications in Information and Systems, 1(1), pp. 33-50, 2001.

[12] Zhao S., Tepe K., Seskar I., Raychaudhuri D.: Routing protocols for self-organizing hierarchical ad-hoc wireless networks. In: IEEE Sarnoff 2003 Symposium, New Jersy, 2003.

[13] Gupta P. Kumar P. R.: The capacity of wireless networks. In: IEEE Trans. Inform. Theory, vol. 46(2), pp.388–404, March 2000.

[14] Naureen, Akram A., Riaz R., Kim K. H., Ahmed H. F.: An end-to-end security architecture for sensor networks. In: ICIS 2007 Canada, December 9-12, 2007.

[15] Zhu S., Setia S., Jajodia S.: LEAP: efficient security mechanisms for large-scale distributed sensor networks. In: 10th ACM Conference on Computer and Communication Security (CCS ’03), pp. 62–72, ACM Press, NewYork, 2003.

[16] Law Y., Corin R., Etalle S., Hartel P.: A formally verified decentralized key management for wireless sensor networks. In: Personal Wireless Communications (2003).

[17] Eltoweissy M., Heydari H., Morales L., Sadborough H.: Combinatorial optimization of key management in group communications. In: Journal of Network and System Management (2004)

[18] Younis M., Ghumman K., Eltoweissy M.: Location aware combinatorial key management scheme for clustered sensor networks. In: IEEE Transactions on Parallel and Distributed Systems (2006).

[19] Myers M., Ankney R., Malpani A., Galperin S., Adams C.: X. 509 Internet public key infrastructure online certificate status protocol – ocsp. Request for Comments (RFC) 2560 (1999).

[20] Wang Y., Attebury G., Ramamurthy B.: Survey of security issues in wireless sensor networks. In :IEEE Communication Surveys and Tutorials 8 (2)(2006).

[21] Gura N., Patel A., Wander A., Eberle H., Shantz S. C.: Comparing elliptic curve cryptography and rsa on 8-bit cpus. In : 6th International Workshop on Cryptographic Hardware and Embedded Systems, Boston, Massachusetts, Aug. 2004.

[22] Du X., Guizani M., Ci S., Xiao Y., Chen H. H.: A routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks.”In: Ad Hoc Networks, vol. 5, no. 1, 2007.

[23] Du W., Deng J., Han Y., Chen S., Varshney P.: A key management scheme for wireless sensor networks using deployment knowledge. In: IEEE INFOCOM, vol. 1, March 2004, pp. 586–597.

[24] Zhou Y., Zhang Y., Fang Y.: LLK: a link-layer key establishment scheme for wireless sensor networks. In : IEEE WCNC, vol. 4, March 2005, pp. 1921–1926.

Kamal Kumar received his M.Tech. as well as B.Tech degree from Kurukshetra University, Kurukshetra, India. Presently he is working as Associate Professor in Computer Engineering Department in M.M. Engineering College, Ambala, India. He is pursuing Ph. D from Thapar University, Patiala, India.

A. K. Verma is currently working as Assistant Professor in the department of Computer Science and Engineering at Thapar University, Patiala in Punjab (INDIA). He received his B.S. and M.S. in 1991 and 2001 respectively, majoring in Computer Science and Engineering. He has worked as Lecturer at M.M.M. Engg. College, Gorakhpur from 1991 to 1996. From 1996 he is associated with the same University. He has been a visiting faculty to many institutions. He has published over 80 papers in referred journals and conferences (India and Abroad). He is member of various program committees for different International/National Conferences and is on the review board of various journals. He is a senior member (ACM), LMCSI (Mumbai), GMAIMA (New Delhi). He is a certified software quality auditor by MoCIT, Govt. of India. His main areas of interests are: Programming Languages, Soft Computing, Bioinformatics and Computer Networks. His research interests include wireless networks, routing algorithms and securing ad hoc networks.

R. B. Patel received a PDF, Highest Institute of Education, Science & Technology (HIEST), Athens, Greece, 2005. He received a PhD in Computer Science and Technology from Indian Institute of Technology (IIT), Roorkee, India. He is member IEEE, ISTE.

His current research interests are in Mobile and Distributed Computing, Security, Fault Tolerance Systems, Peer-to-Peer Computing, Cluster Computing and Sensor networks. He has published more than 100 papers in International Journals and Conferences and 17 papers in national journal/conferences. Two patents are also in the credits of Dr. Patel in the field of Mobile Agent Technology and Sensor Networks.