Dynamic Session Key based Pairwise Key Management Scheme for Wireless Sensor Networks

Premamayudu B[1], Dr. K. Venkata Rao[2], Dr. P. Suresh Varma[3]
[1]Department of Information Technology, Vignan's University,
Vadlamudi, Guntur (Dt), Andhra Pradesh, India
[e-mail: premamayudu@gmail.com]
[2]Department of Computer Science and Engineering, Vignan's Institute of Information Technology,
Gajuwaka, Visakhaptnam (Dt), Andhra Pradesh, India
[e-mail: vrkoduganti@gmail.com]
[3] Adikavi Nannaya University,
Rajahmundry, W. Godavari (Dt), Andhra Pradesh, India
[e-mail: vermaps@yahoo.com]
*Corresponding author: Premamayudu B

*Received June 18, 2016; revised September 19, 2016; accepted October 5, 2016;
published December 31, 2016*

Abstract

Security is one of the major challenges in the Wireless Sensor Networks (WSNs). WSNs are more vulnerable to adversarial activities. All cryptographic security services indirectly depend on key management. Symmetric key management is the best key establishment process for WSNs due to the resource constraints of the sensors. In this paper, we proposed dynamic session key establishment scheme based on randomly generated nonce value and sensor node identity, in which each sensor node is equipped with session key on expire basis. The proposed scheme is compare with five popular existing key management systems. Our scheme is simulated in OMNET++ with MixiM and presented experimental results. The analytical study and experimental results show the superiority of the proposed scheme over the existing schemes in terms of energy, storage, resilience and communication overhead.

Keywords*:* Security in WSNs, Pairwise Keys, Session key, Key Establishment.

http://dx.doi.org/10.3837/tiis.2016.12.024 **ISSN : 1976-7277**

1. Introduction

Ubiquitous and pervasive applications such as health care, military, industry automation, and home security are implemented based on Wireless Sensor Networks (WSNs). Generally, the WSNs consist of numerous sensor nodes and Base Station (BS). **Fig. 1** shows the general WSNs structure. Sensor nodes are operated with tiny battery and deployed in hostile environments to continuously sense real time information such as pressure, light, movement, moisture, temperature, and so on[1,2,3,4].

Networked sensor nodes forward the collected information from the field either directly to the BS or via other nodes in the network. The BS forwards the collaborated data to a remote station through the external network such as internet for further processing [5].

In recent years, [6] vehicular ad hoc networks (VANETs) have the potential to integrate wireless communication and networking the vehicles that includes bus, car, traffic signals, cell phones and other devices. This type of communication system among the vehicles can transform the way people to travel. However, the trustworthy of the data and node trust in the network is the major challenge in these types of networks. The trust management schemes are also inherits the properties of cryptographic security services. The effectiveness and efficiency of trust management indirectly depends on the key management system, because cryptographic security services are tightly coupled with key management. Many environmental monitoring systems are implementing with the help of wireless sensor networks to activate the digital filtering methods to stabilize the environmental conditions. Certainly, these types of applications are required security system to prevent security threats [7].

Most of the cases, WSNs are designed to collect the sensitive information from the deployment field. Sensitive information demands the security. Therefore, security is an inevitable issue in WSNs. However, traditional security mechanisms are not suitable directly to WSNs. In addition, WSNs are not competent in the all the resources of traditional network. Hence, traditional security solutions are not suitable to WSNs. Many security solutions depend on cryptographic operation, which needs a key to perform operations such as encryption and decryption. A difficult issue of key management is to generate and control the cryptographic keys between the pair of sensors in the network. Hence, key establishment and management is the building block for all security solutions.

Although, Key management process is primary building block for security solutions, but security solutions do not state how to interchange keys securely over the network. This issue triggers the key management as open research area. The key management system has the following objectives:

i. Efficiency in resource consumption: The key management system must be considered energy, processing capabilities, storage space, and communication overhead of sensor nodes.
ii. Scalability: The system has to allow the additions or deletions of sensors in the network.
iii. Backward and Forward Compatibility: When sensor nodes are added or deleted, the cryptographic key can’t allow to access previous (backward) or future (forward) messages in the network.

Cryptographic key materials are pre-distributed to the sensors in different kind of key management systems before their deployment in the application area. Key pre-distribution eliminates large computations for key preparation at the level of sensors. Though, this solution is not resilient and typically needs storage space for several keys. In addition, it restricts the scalability of the network after deployment. Hence, novel solutions are required to prevent node compromising attacks, do not utilize the more resources and allows the network extension after deployment phase.

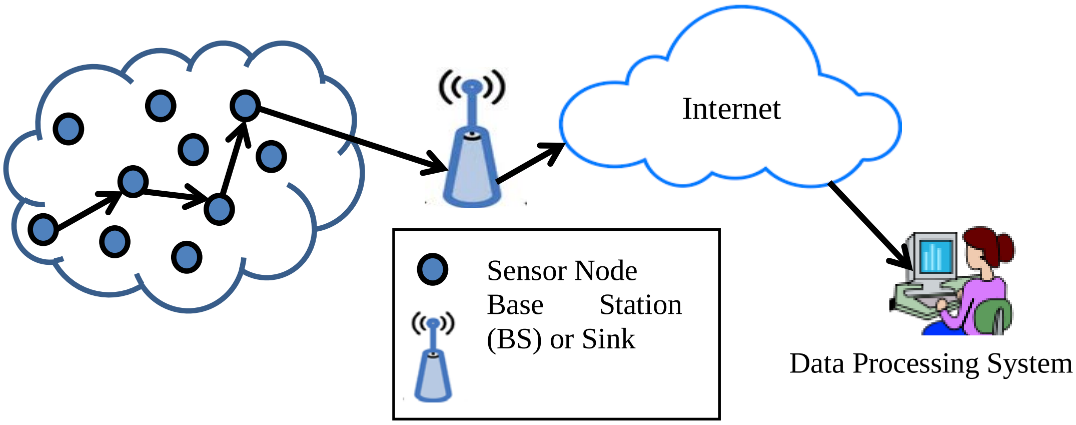


Fig. 1. Wireless Sensor Network

In this paper, the proposed solution on key management system guarantees a balance between resource constraints and security in network. The proposed system depends on session key. This ensures few calculations, less memory space and a lesser number of messages to generate and refresh/update pairwise keys in the network. This system performs less number of computations to create a pairwise key between any pair of sensors in the communication range. Therefore, it ensures the light weight key establishment and management system for WSNs.

The paper is structured as follows. The section 2 presents brief about existing key establishment models and motivation to new proposal. Section 3 defines proposed key management method. Section 4 shows the analytical study and security analysis of the proposed work. Section 5 presents the conclusion on proposed work.

2. Existing Key Establishment Models

Several researchers have been identified key management systems for WSNs. Complete survey can be found in [8, 9, 10]. The proposed systems can be divided into two basic categories based on key distribution, establishment and management: (i) symmetric key management system (ii) asymmetric key management system. Symmetric system offers less resource utilization, but can't resist many attacks and does not provide scalability of the network. In the case of asymmetric system, excellent resistance against many attacks including node compromising attack and offers high scalability, but needs heavy resources in the part of software and hardware of sensor nodes [11,12,13,14,15]. Energy efficiency is also one of the main challenges in WSNs to prolog the network life. In all the applications and protocols used for communication are need to be energy efficient to maintain the stable lifetime of the network. In [16,17,18] presented energy efficient routing protocols and intrusion detection system for WSNs. This section presents solution on the symmetric key management systems. We concisely review and analyze basic and recent existing key management schemes.

In symmetric systems, the basic idea is to pre-store the same key in all sensor nodes before deployment [19, 20]. After deployment, all the sensor nodes use the same key for secure communication. The main problem with this solution is that even one senor node compromised, it leads to compromise the while network. Alternative solution stores some sub set of keys form key pool in each sensor node. Any pair of sensors in the network finds the common key between sub set of keys and use it as pairwise key for secure communication. This solution prevents the node capture attack, but it can't provide scalability and not suitable for large scale WNSs.

Blom [21], proposed the matrix related key management system where symmetric matrix K_{NxN} provides all pairwise keys of N sensor nodes in the network. The element in ith row and jth column of this matrix is the key between sensor nodes i and j to establish secure communication. K=A x G where $G_{(\lambda+1)xN}$ is a public matrix, $A_{Nx(\lambda+1)}$ is private matrix and $A=(D \times G)^T$. D is (λ+1) x (λ+1) randomly generated secret matrix available with trusted party like base station or sink. Each sensor node i is pre-loaded with ith row of A and jth column of G. In key establishment phase, each pair of nodes i and j can locally calculate a pairwise key by interchanging their columns in plain text. This system is λ-Secure. Meaning that more than λ sensor nodes are compromised by the attacker, the whole network can compromise. Since, this system needs to compute a vector multiplication and store 2 x (λ+1) keys.

Blundo et al.[21] presented a polynomial-related key management system. The λ-degree bivariate polynomial $f(x,y)$is randomly generated over F_q(q is a huge prime number). This polynomial allows the property $f(x,y) = f(y,x)$. Before deployment, each node i is pre-loaded a polynomial share $f(i,y)$. In key establishment, any two sensors i and j interchange their identities and can calculate the pairwise key $f(i,j)$. This scheme is also λ-secure. Hence, it requires the memory for polynomial and computation capability based on degree of polynomial.

Eschenauer and Gligor (EG) [15] presented a probabilistic key establishment system to establish pairwise keys between sensor nodes. Their random key pre-distribution solution is divided into three phases: (i) key pre-distribution (ii) shared key discovery and (iii) path key establishment. In first phase, each sensor node randomly loads with m keys form a key pool S. In second phase, each sensor node discovers the shared key among its neighbors in the network. If there is no shared key between two sensor nodes, they can establish shared key through two or more other nodes during third phase. The probability (P) of at least one common key between two sensor nodes having sub sets of size m from the key pool size S. For instance, the probability of 0.5, 75 keys are stored in each sensor from S=10,000. However, when neighboring nodes of sensor have m different keys, then those keys in sensor node are useless. When the sensor node is compromised, all the keys of it will be disclosed. This will affect compromised node communication links as well as non-compromised node links. The challenge issue of this solution is to find the relationship between key pool size (S) and sub set size (m). The large key pool size decrease the probability of shared key between adjacent nodes and smaller key pool size reduces the resilience of the network against node capture attacks. Further, this solution extended in [22, 23, 24], but they used more memory space than the EG's [15] solution.

In [19, 25], LEAP (localized encryption and authentication protocol) scheme is presented. This scheme is suitable for heterogeneous or hierarchical wireless sensor networks. LEAP needs five keys: (i) global key, (ii) pairwise key, (iii) cluster key, (iv) individual key and (v) group key. Each sensor is pre-loaded with a global key (k1), a pseudo-random function $f$and an individual key shared with base station. Sensor node establish its master key using K1 and f. LEAP is highly scalable but not resilient. The global key is common for all the nodes. Since, the total security of the network depends on the global key which is deleted as soon as possible after the pairwise key establishment phase. In case the sensor node is capture before the key establishment phase, an attacker can obtain all pairwise keys. In addition, scalability is not possible in the network.

In [22, 26, 27] new key management solution based on post-deployment knowledge of sensor nodes in network was proposed. These proposals improved the resilience to node compromising attack by regenerating key pool when new sensors are included in the network. These solutions required more computation power and storage space, because they used many hash and XOR operations at sensor node level. Among these solutions, Hash Graph-based key pre-distribution (HaG) [6] uses hash graph of keys to refresh key pool at each post-deployment of sensor nodes. Each sensor has to wake up for a time called generation window G_w. HaG has divided into three phases: (i) key pool generation (ii) key ring pre-distribution and (iii) pairwise key establishment. In key pool generation phase, an initial key pool S is randomly created. S is refreshed for each post-deployment. In key ring pre-distribution phase, key pool S is divided into groups of (g) keys and each sensor node is pre-loaded with m keys from the g key groups. In third phase, each sensor node i exchange its generation value and identifier. Using these two values by the neighbors of node i, calculate the key indexes of i. Then the neighbor checks if any one has common shared keys with i. In this scheme, key pool is refreshed using the XOR operation and hash function. For a key chain (m=150 keys), the sensor should perform 150 XOR hash operations at each key refresh. For instance, G_w=5, each sensor node computes 150x5 XOR-hash operations, which is more energy consuming. This scheme increases the key connectivity but decreases the resilience against node compromised attack, because the compromised nodes disclose m keys that damage links from generation i to i+G_w.

Rahman et al. [28] modifies the deterministic matrix based key management solution [29] to support key refresh and addition of nodes after deployment. First phase is same as [20], but public matrix is generated by node itself. Suganthi et al.[30] presented a key management system with energy saving. In pre-distribution phase, all the sensor nodes are pre-loaded with initial key and a pseudo-random function. After deployment, each node generates the key that it shares with base station. The base station constructs the spanning tree, in which root is base station. This solution consumes less memory, but reduces the resiliency against node compromise attack. The entire security of network depends on the initial key. In addition, when an adversary captures one node, he/she can generate pairwise key with any node in the network. This state creates much vulnerability in the network.

In [31] presented novel key management model based on location deployment knowledge of sensor nodes in the network. Many existing schemes failed to consider changes in the signal range and the deployment error. The location information determined from the signal range and deployment error were exploited a Gaussian distribution. In this model, they defined polynomial pre-distribution scheme in two approaches (i) defined enough number of adjacent cells and length of the grid cell. This model improved the key connectivity and communication overhead. But it is required more storage space to store the deployment information in each sensor.

In [32] defined authentication as a service between IoT devices used as public safety devices and public safety responders. In current machine to machine communication, security is playing key role to carrying useful information about the event scene, making the critical decision in real time, and current status of a mission. In [33] formulated a secrecy-rate-maximization problem to address the communication vulnerabilities to eavesdropping attack.

Finally, the existing solution can't fulfill the all the security needs of WSNs. For instance, when resiliency is achieved, scalability is dropped. In this paper, a new solution is proposed to provide the balance between security and resource usage. The proposed system creates a pairwise key between any pair of nodes using session key, which is preloaded into each sensor node before deployment. This solution offers scalability, resilience, resource awareness, resists node compromising attack and backward and forward compatibility.

3. Proposed key management scheme

In this section, we presented the description of our proposed key establishment, key refresh/revocation and new sensors addition. Our scheme minimizes the storage space occupation to stock key material and energy consumption for key establishment. A session key (K_S) is global value to entire network. For every session a new session key ($K^{'}_{s}$) can be generated by BS and refresh the pairwise keys between the sensors. As soon as the sensor node establish pairwise key, it erases the session key from memory of sensor node. This process resists most of the adversary attacks on sensor nodes including node compromising attack. Assume that the BS is trustworthy or temper resist and can't be compromised. The **Table 1** defines the notations used in this scheme.

Table 1. Notations

Notation	Description of Notation
S_i	ith sensor in WSN, S_i denotes the identifier of sensor
M_K	Key K is used to encrypt information M
$BS \rightarrow *:M$ (or) $S_i \rightarrow * : M$	Message M broadcasted by base station (BS) or Sensor (S_i)
$MAC_K(M)$	Message Authentication code of given information M used k as secret
K_S	Session Key(Global Value) for the entire network
||	Concatenated symbol
$H_{KS}(M)$	One way hash function on information M with session key
T_i	Timer of sensor S_i
T_{Si}	Time stamp of sensor S_i
$\bigoplus$	XOR Operation

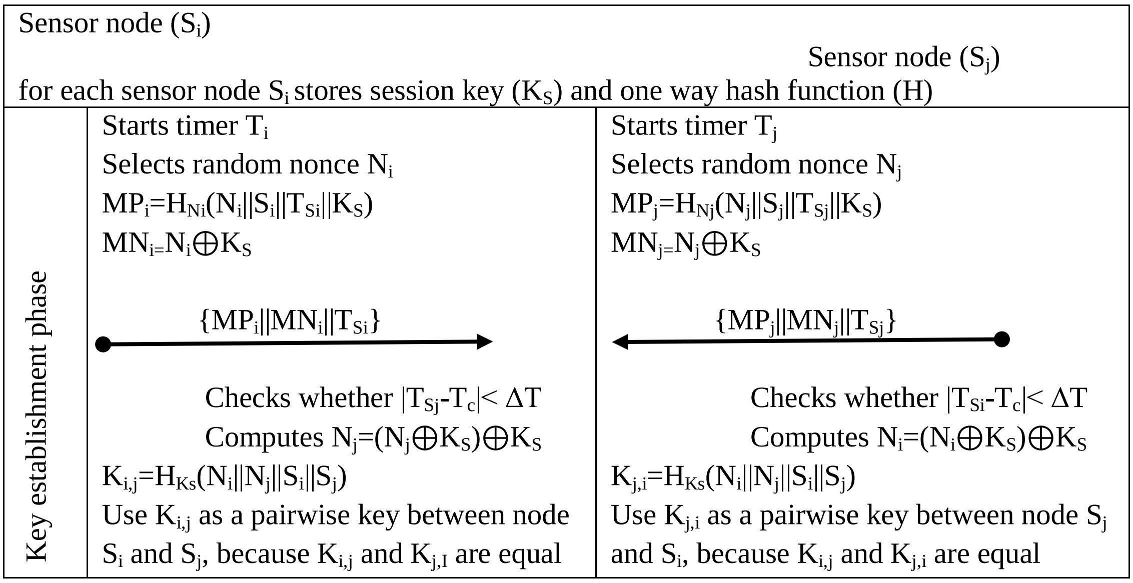
3.1 Key Establishment

Base Station generates a session key (K_S) and pre-stores the session key into each sensor node memory. Each sensor node S_i maintains a timer T_i which is initialized with a value. This value determines how long the sensor node will keep the session key in it. The timer value is decided based on the security level of the application area. If the application needs high level security, than the timer is initialized with smaller value. Otherwise timer is initialized with big value. When the base station identified malicious operation in the network, it will reinitiate new session key and broadcast the session key $K_S^{'}$ to all its neighboring nodes. Moreover, deploying new sensor node can also enables the new session key generation and key refresh operations in the network.

While pairwise key establishment, each sensor S_i starts its timer T_i and generates the message $\{S_i, MP_i, MN_i, T_{Si}\}$. During the message preparation, the sensor node S_i generates nonce N_i and computes the MP_i and MN_i values. The MP_i value is computed using selected nonce N_i and node identifier S_i, current timestamp T_{Si} and session key K_S. For this value (MP_i) one way hash function applied using nonce N_i as secret. $H_{Ni}(MP_i)$ allows authentication and integrity. The value MN_i is calculated using XORing the N_i and K_S. Now, the generated message propagates to all the neighboring nodes of each senor node in the network. Each node is able to establish pairwise key with its neighboring node with a single broadcast message. Upon receipt of broadcast message, each sensor node checks $|T_{Si}-T_c| < \Delta T$. If the timestamp value T_{Si} was within the allowed time interval ΔT, then it agree on the received message. Or else, it rejects the received message. This verification process prevents the replay attacks. The procedure of key establishment phase is depicted in **Fig. 2**. Hence, two neighboring sensor nodes S_1 and S_2 compute a pairwise key in the following fashion:

S_1: generates nonce N_1 and propagates $\{S_1 \,||\, H_{N1}(S_1|| N_1|| T_{S1} \,||\, K_S) \,||\, N_1 \oplus K_S \,||\, T_{S1}\}$
S_2: generates nonce N_2 and propagates $\{S_2 \,||\, H_{N2}(S_2||N_2||T_{S2}||K_S) \,||\, N_2 \oplus K_S \,||\, T_{S2}\}$
S_1: Computes node S_2 nonce $N_2=(N_2 \oplus K_S) \oplus K_S$ and then determines the pairwise key $K_{1,2}=H_{KS}(N_1||N_2||S_1||S_2)$
S_2: Computes node S_1 nonce $N_1=(N_1 \oplus K_S) \oplus K_S$ and then determines the pairwise key $K_{2,1}=H_{KS}(N_1||N_2||S_1||S_2)$

Base station and all sensors can setup the pairwise keys with other sensor nodes which are in the same communication region with the help of above said process. After establishing pairwise with their neighbors, each sensor node removes the session key (K_S) from its memory. The sensor nodes can't determine the nonce value of its neighbors without session key. Hence adversary nodes can't calculate the nonce value of other nodes without getting the session key of the network.



Sensor node (S_i)

Sensor node (S_j)

for each sensor node S_i stores session key (K_S) and one way hash function (H)

Key establishment phase

Starts timer T_i Selects random nonce N_i $MP_i=H_{Ni}(N_i\|\|S_i\|\|T_{Si}\|\|K_S)$ $MN_i=N_i \oplus K_S$	Starts timer T_j Selects random nonce N_j $MP_j=H_{Nj}(N_j\|\|S_j\|\|T_{Sj}\|\|K_S)$ $MN_j=N_j \oplus K_S$
$\{MP_i\|\|MN_i\|\|T_{Si}\}$	$\{MP_j\|\|MN_j\|\|T_{Sj}\}$
Checks whether $\|T_{Sj}-T_c\|< \Delta T$ Computes $N_j=(N_j \oplus K_S) \oplus K_S$ $K_{i,j}=H_{KS}(N_i\|\|N_j\|\|S_i\|\|S_j)$ Use $K_{i,j}$ as a pairwise key between node S_i and S_j, because $K_{i,j}$ and $K_{j,I}$ are equal	Checks whether $\|T_{Si}-T_c\|< \Delta T$ Computes $N_i=(N_i \oplus K_S) \oplus K_S$ $K_{j,i}=H_{KS}(N_i\|\|N_j\|\|S_i\|\|S_j)$ Use $K_{j,i}$ as a pairwise key between node S_j and S_i, because $K_{i,j}$ and $K_{j,i}$ are equal

Fig. 2. Key Establishment phase of the proposed scheme

3.2 Key Refresh/Revocation

Key refresh is accomplished in two cases. In one case, it is done periodically to defend eavesdropping attacks. Meaning that pairwise key is updated time to time. In other case, it used to remove captured sensor in to network. BS only enables the key refresh operation in the network. It generates a new session key K'_S and nonce N_{BS}. Then the BS encrypts the new session key and nonce with the current pairwise key and broadcasts to its neighbors. **Fig. 3** depicted the key revocation process of our proposed scheme.

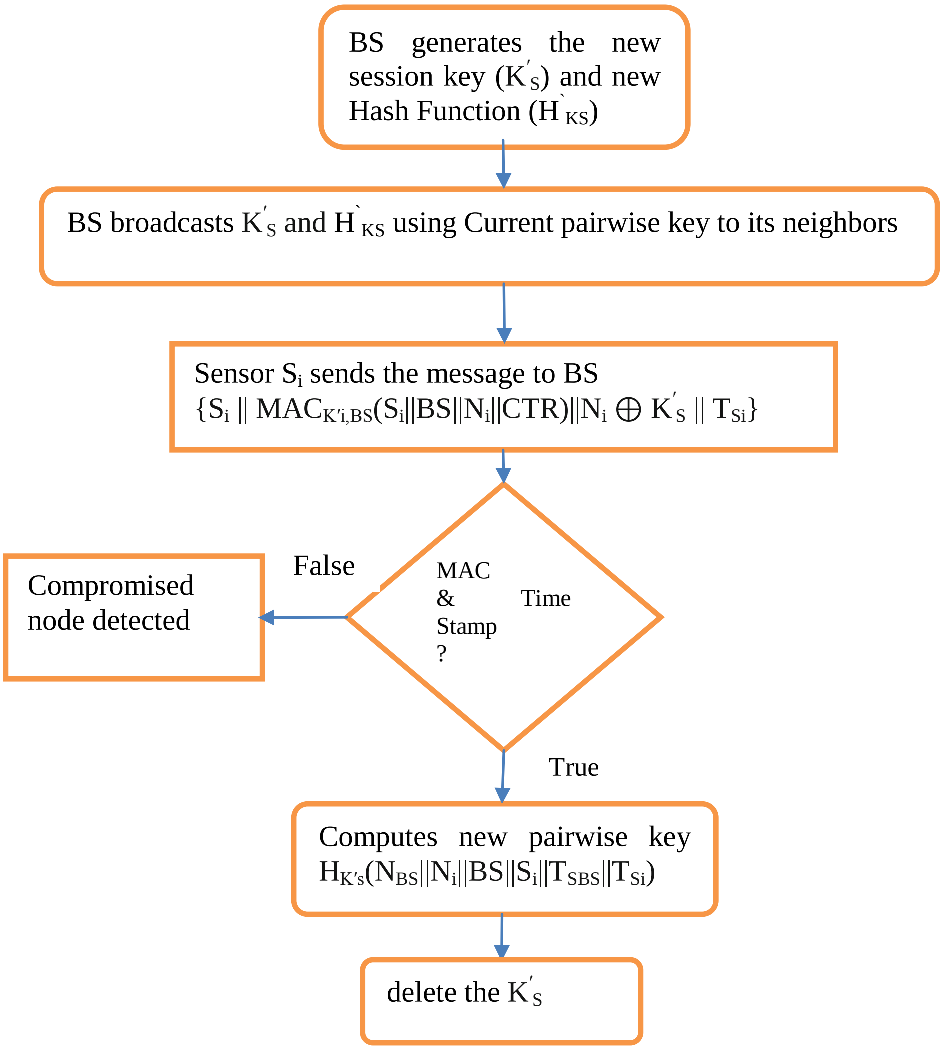

BS generates the new session key (K'_S) and new Hash Function ($H^`_{KS}$)
BS broadcasts K'_S and $H^`_{KS}$ using Current pairwise key to its neighbors
Sensor S_i sends the message to BS
$\{S_i \,||\, MAC_{K'i,BS}(S_i||BS||N_i||CTR)||N_i \oplus K'_S \,||\, T_{Si}\}$
False
Compromised node detected
MAC & Time Stamp ?
True
Computes new pairwise key $H_{K's}(N_{BS}||N_i||BS||S_i||T_{SBS}||T_{Si})$
delete the K'_S

Fig. 3. Key refresh/revocation in the network

$BS \rightarrow S_i : \{K'_S \,||\, N_{BS} \,||\, L \,||\, CTR\}K_{i,BS}$, for all S_i neighbors of base station, where CTR indicate the key refresh counter. It resists the replay attack. Hence, an adversary can't insert old messages in the communication. L provides the list of identities of malicious sensors in the network. When the new session key K'_S propagates over the network, L listed sensors do not

get it. Upon receipt of key refresh message from BS, each sensor node S_i initializes the timer value T_i and then broadcasts the below message to base station:
$S_i \rightarrow BS : \{S_i \,||\, MAC_{K'i,BS}(S_i||BS||N_i||CTR)||N_i \oplus K'_S \,||\, T_{Si}\}$
Note that message contains MAC and timestamp to verify their authentication & integrity and freshness of the message. Now, the pairwise key can be determined using below scenario:
$K'_{i,BS} = H_{K's}(N_{BS}||N_i||BS||S_i||T_{SBS}||T_{Si})$.
The key refresh is accomplished over the network until each pair of sensors refreshed their pairwise keys.

3.3 Adding New Sensor

Scalability property of key management can be improved by adding new sensors to the network. The newly added sensor (S_n) needs to establish pairwise key with neighboring nodes. Base station creates a new session key K'_S and pre-loads in the new sensor S_n. Before deploying the new sensor node into network, base station accomplishes the key refresh operation to propagate new session key K'_S over the network. When a new sensor deployed into network, it broadcasts the below message and initialize the timer T_n
$S_n \rightarrow * : \{ S_n \,||\, H_{Nn}(S_n||N_n||T_n||K'_S)||N_n \oplus K'_S \,||\, T_{Sn}\}$
Upon receipt of above message, all sensor nodes with the range of new sensor S_n compute a pairwise key using the below computation.
$K_{i,n} = H_{K's}(N_n||N_i||S_n||S_i||T_{Si}||T_{Sn})$
Assume that S_i is the one of neighboring node of S_n. Then S_i forwards the below message to the sensor S_n.
$S_i \rightarrow S_n: \{S_i \,||\, MAC_{Ki,n}(S_i||S_n||N_n||T_{Sn})\}$
The MAC value is computed for the values which are using in pairwise key generation with new key $K_{i,n}$ and it is forwarded to S_n to verify the authenticity and integrity. S_n recalculates the MAC value of received message. If both received and calculated MAC values are same, then the message is accepted, otherwise rejected.

4. Analytical study

We have performed analysis on key connectivity and resilience against node compromised attack between schemes presented in the section 2 and our scheme.

4.1 Key Connectivity

This section presents the key connectivity in the network. Our scheme, Matrix-based scheme [20], and polynomial scheme [21] are deterministic. The key connectivity means the probability of any two sensor nodes in the network have a common key. This is equal to the number of neighbors (d) in the communication range for the sensor node. The value d defines the key connectivity in the network. **Table 2** presents the key connectivity analytical results of our scheme, Blundo et al.'s scheme [4], EG Scheme [15], Blom's scheme [20] and HaG Scheme [29] for key pools size of S=10,000.

Table 2. Key connectivity in the network contains N nodes

Name of the Scheme	Key Connectivity in the network
Matrix	1
Polynomial	1
HaG (Key Ring=250, Group Keys=5)	0.72
EG (Key Ring =200)	0.98
Our Scheme	1

In probabilistic schemes, the key connectivity depends on the sizes of key rings (m) and the key pool (s). To provide key connectivity ≃ 1, key ring m=250 for each sensor and key pool size S=10,000 need to maintained in the network. If each key size is 128 bit, each sensor needs to allocate 4000 bytes of memory space for cryptographic keys only. **Fig. 4** shows connectivity in probabilistic key management schemes. In the key connectivity is equal to 1.

4.2 Resilience against Compromised Node

The resilience of the scheme is computed as the ratio of compromised links F_z when z sensor nodes are compromised. In our scheme, the number of compromised links is equal to the number of its neighbors, which is the degree (d) of sensor node. Assume N_z is the number of corrupted links, when the adversary compromised z sensors. Then the resilience ratio is

$$F_z = \frac{N_z}{N * d} \tag{1}$$

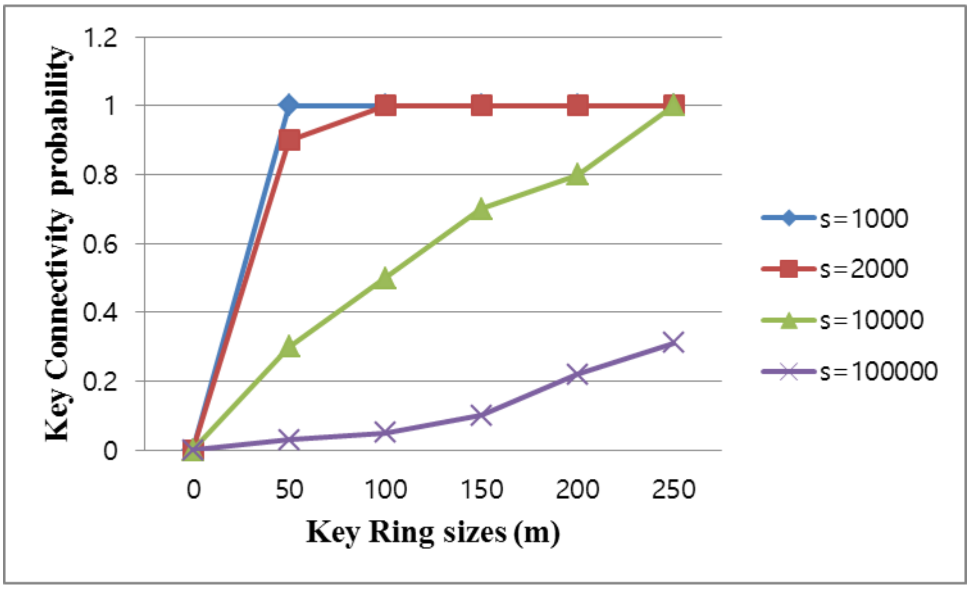


Fig. 4. Key connectivity for different key ring and key pool sizes

In random key management, the compromised node reveals all the keys in the key ring/chain (i.e. m keys) from the key pool S. Hence the ratio of the compromised links can be calculated with following formula when z sensors are captured.

$$F_z = z\frac{m}{S} \tag{2}$$

The **Table 3** shows the sample resilience ratio of the compromised links for the existing schemes discussed in section 2 of this paper.

Table 3. Ratio for compromised links between our scheme and existing schemes

Resilience	**Ratio of compromised links F_z**				
Scheme	**Number of compromised sensor(z)**				
	1	**2**	**3**	**4**	**5**
Our scheme	0.0001	0.0002	0.0003	0.0004	0.0005
EG's scheme(m=75)	0.0075	0.015	0.0225	0.0300	0.0375
EG's scheme(m=120)	0.012	0.024	0.036	0.048	0.06
EG's scheme(m=180)	0.018	0.036	0.054	0.072	0.09
HaG's scheme(m=250, g=1)	0.025	0.05	0.075	0.1	0.125

In this analysis, consider 1000 sensor nodes in the network and degree of neighbor d≃10 of our scheme. **Fig. 3** describes the resilience against the node compromised attack. Moreover, the nodes are compromised randomly in the network. For other schemes (EG and HaG) key pool size S is 10,000 and different key ring sizes (m=75, 120, 180). For HaG, the first generation is used for compromised nodes.

This section analyzes some of the attacks against to sensor nodes and attacks retrieve secret/sensitive information from its storage.

5.1 Node Compromising Attack

Sensor nodes are not physically secure, because they are deployed in open environments. The adversary can physically capture the node and may conduct node replication and false message distribution. After capturing the node, attacker can get only pairwise key information from the node in our key management scheme. Pairwise key can't reveal the session key and nonce values of that node and its neighboring nodes. Since, the compromised node and its neighbors connected with that node can only affected for this attack. The non-compromised nodes and their links are not affected for the node capture attack. If the node is captured before the session key erase from its memory, then attacker can affect the whole network. For this reason, the timer value of the sensor node in our scheme is very important parameter to avoid the node capture attack. The time value is decided based on the level of security in the applications. Most of the cases, this value will be decided after applying the many test cases on the sensor network.

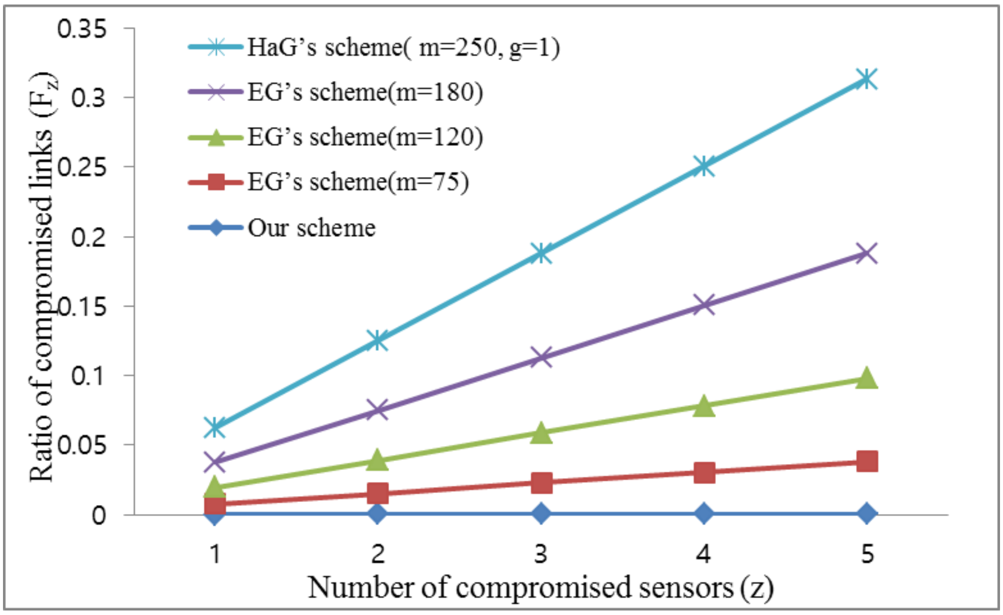


Fig. 5. Resilience against compromised sensor node attack

5. Security analysis

5.2 Black Hole and Worm Hole attacks

A block hole and worm hole attacks are an active DoS attacks. An adversary compromises some nodes and install malicious program to them not forwarding message to the destination. A black hole can be either single node a group of nodes. In worm hole attack, compromised node receives data packets at one point and tunnels them to another compromised node. The tunnel exists between two compromised nodes is called worm hole in network. Both the attacks can be applied easily in WSN, but very difficult to resist. These are applied in routing protocols of WSNs. Our key management protocol provides a strong authentication and integrity mechanism to data packets during transmit using hashing and time stamp. Hence, it can prevent the black hole and wormhole attack in the network.

5.3 Replay Attacks

A replay attack is maliciously or fraudulently repeated or delayed data packets in the network. An adversary intentionally repeats the data packets with malicious information to misguide the sensor nodes. The proposed scheme is using strong timestamp mechanism to avoid the delayed or repeated packets during the key establishment, node addition and key revocation phases.

5.4 Sybil Attacks

A single sensor node represents itself with multiple identifiers. In proposed scheme, each node establishes distinct pairwise keys with its neighbors. Any node can't use other nodes identity without knowing pairwise key of other sensor node. In new node addition phase, neighboring nodes ensure the authentication of new node using hash and MAC functions. Hence, our scheme prevents Sybil attacks.

5.5 Forward and Backward Compatibility

A new session key is loaded into sensor in key refresh phase. In addition, session key is updated in the network periodically base on demand to eliminate the compromised nodes. After session key update, every node had to recalculate new pairwise key with its neighbors to establish secure communication. The new pairwise key can't decrypt the old messages in the network. While key refresh phase, all the compromised nodes are eliminated and also eliminated node identities list propagated with key refresh messages. Hence, non-compromised nodes can't establish pairwise keys with compromised node, because they know the identities of compromised nodes. Therefore, compromised nodes can't encrypt or decrypt the future messages in the network. The periodic key refresh operations are initiated by the base station only. Therefore, our scheme ensures the forward and backward message security.

6. Experimental Results

We experimentally evaluated our scheme with other four schemes described in section 2 using OMNET++ Simulator. OMNET++ (Objective Modular Network Test-bed in C++) is best framework for WSN's simulation environments. The simulation environment is configured in MixiM component using the properties summarized in **Table 4**. MixiM (mixed simulator) is a simulation framework for wireless sensor networks using OMNET++ simulation engine. MixiM is a modeling framework created for wireless sensor network in OMNET++. Let us assume that, the nodes are randomly deployed in the application area. The CSMA/CA is used as medium access scheme with 10% default bit error rate. The key pool size is 10000 and different key ring sizes (m=75, 120, 180) used with EG's scheme. The communication range is 87m and initial power of each sensor node is 2J. Path loss exponent (γ) 2 or 4 was selected for simulation. Energy consumption for each bit processing is 7nJ. And energy consumption for transmission and receiving is 50nJ/bit. The properties sated in simulation are assumptions to make a simulation model among existing key management schemes and proposed scheme. **Fig. 6** shows screenshot of simulation model.

6.1 Energy Consumption of the Sensor Node

Sensor nodes having several operational modules such as microcontroller, transceiver/receiver, sensing unit and power supply unit. The power consumption of sensor node has to consider the energy usage of processor, communication and sensing modules. We assumed that sensing module consumes very less energy compared with processor and communication modules. Hence, we are not included sensing module energy value into sensor node energy consumption equation. Therefore, we defined energy consumption model for sensor node by combing the energy consumption at processor module and communication module.

Energy consumption at communication module: It is responsible for sensor node data sending and receiving. Normally, the communication module has six states (T_x, R_x, off, idle, sleep and CAA/ED).

Table 4. Properties used in simulation

Properties	Value
E_{cpu} per bit	7nJ
E_{elec} per bit	50 nJ
E_{amp} for γ=2	100 pJ/bit/m^2
E_{amp} for γ=4	0.0013 pJ/bit/m^2
Key pool size(S)	10000
Key Ring sizes(m)	75,120,180
MAC Protocol	CSMA/CA
Node deployment	Random
Path loss exponent(γ)	2 or 4
Range of communication	87m
Size of network	500mx500m

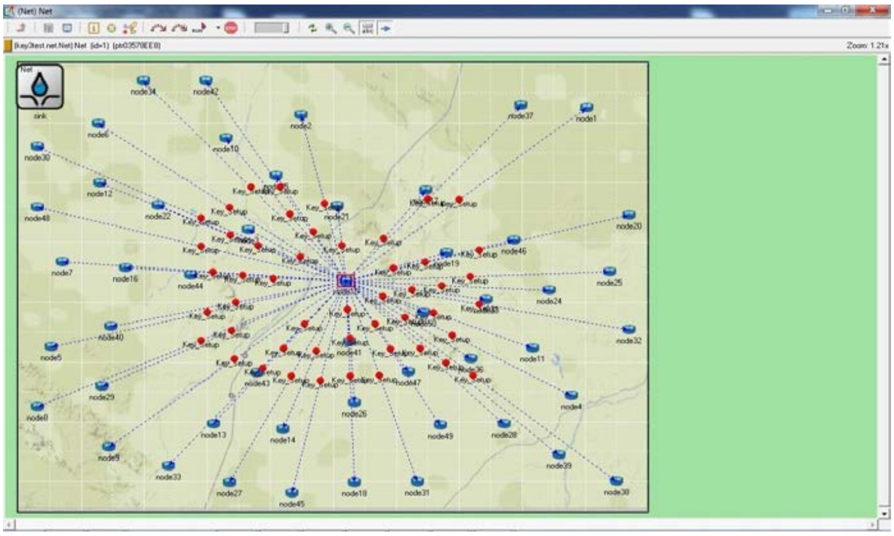


Fig. 6. OMNET++ Simulation Environment Screenshot

Since, T_x and R_x energy consumption values are consider as communication module energy consumption value.

$$E_{T_X}(d) = E_{T_{X-elec}} + E_{amp} dis^{\gamma} \tag{3}$$

$$E_{R_X} = E_{R_{X-elec}} \tag{4}$$

Where 'dis' is the communication distance, $E_{T_{X-elec}}$ is the per-bit power consumption for transmission, $E_{R_{X-elec}}$ is the per-bit power consumption for reception, E_{amp} is the per-bet power consumption for one bit in one m^2 and γ is the path loss exponent that depends on the transmission distance. The communication distance (dis) is less than a computed threshold, and then γ value is 0, otherwise 4. The units for $E_{T_{X-elec}}$ & $E_{R_{X-elec}}$ is nJ/bit and the units for E_{amp} is pJ/bit/m^2.

Energy Consumption at Processor Module: In general, the microprocessor/microcontroller supports three operation states and five state transitions. The operation states of the processor are sleep, run and idle and transitions are sleep to run, run to sleep, run to idle, idle to run and idle to sleep. Energy consumption of processor (E_{cpu}) is the sum of operation state ($E_{cpu\text{-}state}$) and sate transition ($E_{cpu\text{-}change}$) power consumption. Therefore, the formula for energy consumption at processor module is:

$$E_{cpu} = E_{cpu-state} + E_{cpu-change} \tag{5}$$

6.2 Energy Consumption of Sensor Node

The energy consumption of sensor node is sum of the energy consumption at transmission module and processor module. If the sensor node S_i transmits or receives n bits length packets to (dis) distance (in meters), then the energy consumption of a sensor node can be calculated according to formulas 6, 7 and 8.

$$E_{T_X}(n) = E_{T_{X-elec}} \times n + E_{amp} \times dis^{\gamma} \times n \tag{6}$$

$$E_{R_X}(n) = E_{R_{X-elec}} \times n \tag{7}$$

$$E_{cpu}(n) = (E_{cpu-state} + E_{cpu-change}) \times n \tag{8}$$

Where n is the length (bits) of data packets, 'dis' is the transmission distance (meters). $E_{T_{X-elec}}$ (nJ/bit) is power need to run the radio circuitry at transmitter per pit. E_{amp} (nJ/bit/m^2) is the power $E_{T_{X-elec}}$ need by the transmitter for an acceptable receiver's demodulator. $E_{R_{X-elec}}$ (nJ/bit) is the energy need to run the radio circuitry at receiver per bit. E_{cpu} (nJ/bit) is the energy dissipation for processing per bit, this is the combination of $E_{cpu-state}$ & $E_{cpu-change}$. $E_{cpu-state}$ is the energy required for operation state of processor and $E_{cpu-change}$ is the energy need for processor state transition. γ is the path loss exponent that is related to the transmission distance. The sensor node energy power consumption is in direct proportion to the length of data packet or message. If the number of messages for key establishment can be reduced, then the energy consumption can also reduce. In addition, the energy consumption can also depend on the distance of communication. The distance of communication (dis) is in direct proportion to the communication energy. When the transmission is less than threshold, then path loss exponent is 2, otherwise 4.

Table 5 provides the comparisons of five schemes described in section 2 in memory space, communication and processing requirements for key management in WSN.

Table 5. Comparisons of different schemes

Scheme	Method	Memory	Data Exchange	Computation
Our Scheme	Session key	D	d+1	d x XOR Operations
Matrix[20]	Blom	2(λ+1)	d+1	Matrix multiplication
Polynomial[21]	Blom	λ+1	d+1	Polynomial evaluation
EG's scheme[15]	Random selection	m keys	d+1	m key comparison and searching
HaG[29]	Random selection and grouping	m keys	d+1	d x XOR and hash operations

6.3 Communication Overhead

It is the number of data packets used for key management. The key management and key refreshment is in direct proportion to the size of data packet and number of data packets exchanged. In **Table 5**, all the schemes use the d+1 message exchanges for key establishment. Now, we can decide the communication overhead of five schemes described in **Table 5** with the length of the message. **Fig. 7** plots the message sizes in each scheme. Our scheme and Hag [29] are using one way hash function to ensure authentication and integrity of the communicated message. The minimum hash code of any hash function is 160 bits, if it using SHA-1.

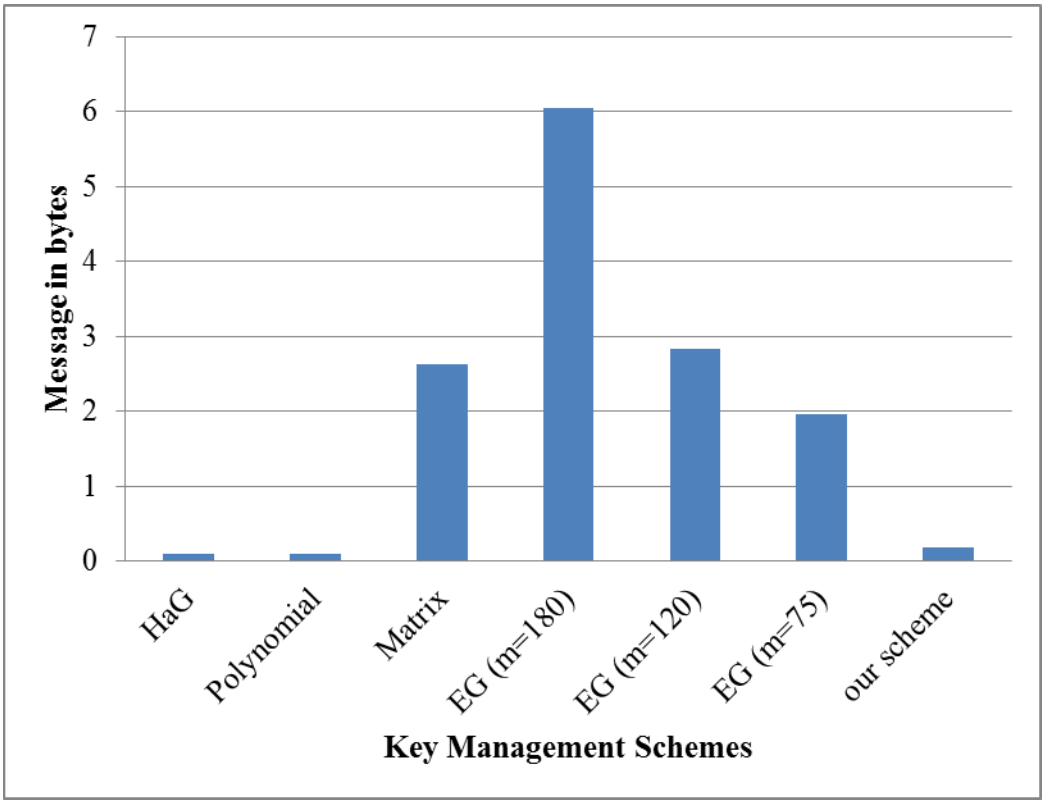


Fig. 7. Length of message in key management schemes

We have also evaluated and plotted in **Fig. 8** about the total energy requirement for key establishment in our scheme, EG (m=75) [15], matrix [20], polynomial [21] and HaG(m=250) [29] schemes. The energy consumption is involved in message transmission and receiving in the key establishment and for calculation of keys. The energy consumption is evaluated for our scheme and other schemes in different network sizes. Energy consumption in our scheme is less than [20], EG [15] and HaG [29].

6.4 Storage Overhead

It is the amount of memory space required to remember pairwise keys and pre-distributed key material such as session key, matrix, polynomial, key chain and so on. **Fig. 9** presents the memory requirement for key establishment schemes in different network sizes. The required storage space of each scheme is defined in **Table 5**. Our scheme consumes less memory than other because each sensor stores keys of its neighbors (d).

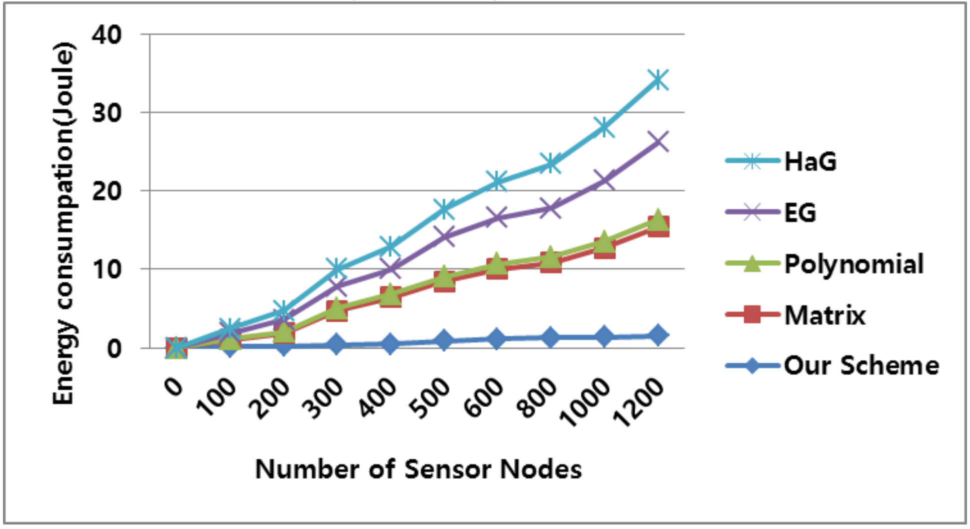


Fig. 8. Energy Conservation in key establishment

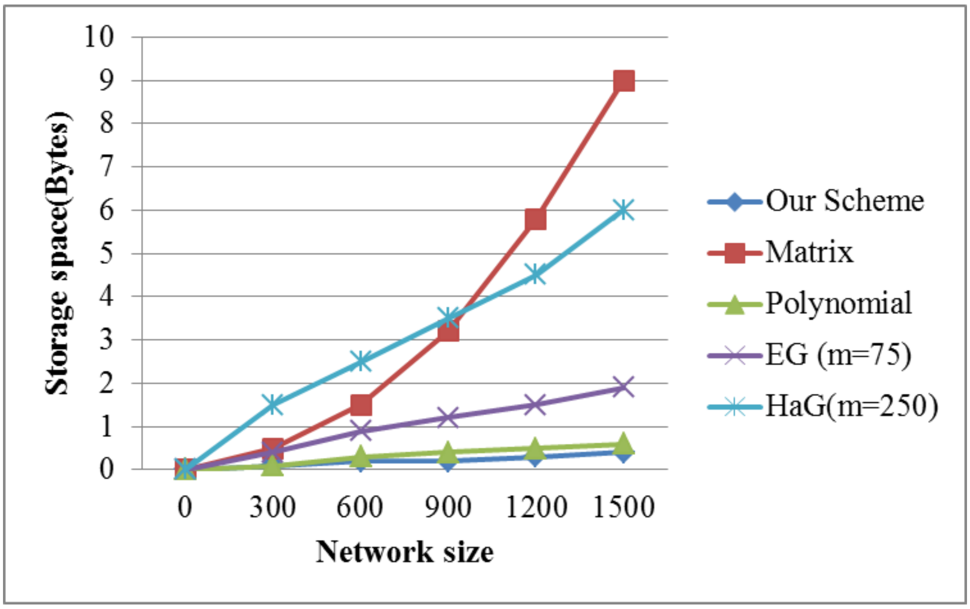


Fig. 9. Memory space required for Key establishment in different network sizes

7. Conclusion

The key management is the challenge issue in rigorous resource constrained wireless sensor networks. In this paper, we proposed key management based on randomly generated nonce values and sensor nodes identity in the WSNs. In this scheme, each sensor node in the deployed network is preloaded with session key based on expire basis to ensure the communication authentication and integrity during key establishment phase. The session key is used to compute hash values of randomly generated nonce and sensor nodes identity. We performed simulation to show the superiority of the proposed scheme over the existing scheme.

The analytical and experimental results show that the scheme required less storage, energy and communication when compared with existing KMSs. In addition, it prevents the node capture attacks, black hole and worm hole attacks, replay attacks, Sybil attack and forward and backward compatibility.

References

[1] Anderson R, Chan H, Perrig A, "Key infection: smart trust for smart dust," in *Proc. of 12th IEEE international conference on network protocols*, pp 206-215, 2004. Article (CrossRef Link)

[2] Carman DW, Kruus PS, Matt BJ, "Constraints and approaches for distributed sensor network security," *NAI Labs Technical Report* #00-010, 2000. Article (CrossRef Link)

[3] Yick J, Mukherjee B, Ghosal D, "Wireless Sensor Network Survey," *Computer networks 52*(12), pp 2292-2330, 2008. Article (CrossRef Link)

[4] Yoneki E, Bacon J, "A Survey of wireless sensor network technologies:research trends and middleware's role," *University of Cambridge TR 646*, Cambridge, 2005. Article (CrossRef Link)

[5] Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E, "A survey on sensor networks," *IEEE communication magazine 40(8)*, pp 102-114, 2002. Article (CrossRef Link)

[6] Li, Wenjia, and Houbing Song, "ART: An Attack-Resistant Trust Management Scheme for Securing Vehicular Ad Hoc Networks," *IEEE Transactions on Intelligent Transportation Systems* 17(4), pp. 960-969, 2016. Article (CrossRef Link)

[7] Yang Jiachen, Jianxiong Zhou, Zhihan Lv, Wei Wei, and Houbing, "A real-time monitoring system of industry carbon monoxide based on wireless sensor networks," *Sensors* 15(11), pp 29535-29546, 2015. Article (CrossRef Link)

[8] Zhang J, Vardharajan V, "Wireless sensor network Survey and taxonomy," *Journal of Network and computer applications* 33(2)", pp 63-75, 2010. Article (CrossRef Link)

[9] Alemdar A, Ibnkahla M, "Wireless Sensor Networks:applications and challenges," in *Proc. of 9th international symposium on signal processing and its applications*, pp 1-6, 2007. Article (CrossRef Link)

[10] Xiao Y, Rayi VK, Sun B, Du X, Hu F, Galloway M, "A survey of key management schemes in wireless sensor networks," *Computer communication* 30(11), pp 2314-2341, 2007. Article (CrossRef Link)

[11] Jilna P, Pattathil DP, "A key Management Technique based on Elliptic Curves for Static Wireless Sensor Network," *Security communication networks*, pp 3726-3738, 2015. Article (CrossRef Link)

[12] Baojiang Cui, Ziyue Wang, Bing Zhao, Xiaobing Liang, and Yuemin Ding, "Enhanced Key Management Protocols for Wireless Sensor Networks," *Mobile Information Systems*, vol. 2015, Article ID 627548, 10 pages, 2015. Article (CrossRef Link)

[13] Liu A, Kampanakis P, Ning P, "TinyECC:eliptic curve cryptography for sensor networks," *TinyECC Software*, 2007. Article (CrossRef Link)

[14] Watro R, Kong D, Cuti Sf, Gardiner C, Lynn C Kruus P, "Tinypk: Securing sensor networks with public key technology," in *Proc. of 2nd ACM workshop on security of ad hoc and sensor networks*, pp 59-64, 2004. Article (CrossRef Link)

[15] Eschenauer L, Gligor VD, "A key management scheme for distributed sensor networks," in *Proc. of 9th ACM conference on computer and communications security*, pp 41-47, 2002. Article (CrossRef Link)

[16] Ahmadi, Ali, et al., "An efficient routing algorithm to preserve k-coverage in wireless sensor networks," *The Journal of Supercomputing* 68(2), pp 599-623, 2014. Article (CrossRef Link)

[17] Naranjo, Paola G. Vinueza, et al., "P-SEP: a prolong stable election routing algorithm for energy-limited heterogeneous fog-supported wireless sensor networks," *The Journal of Supercomputing*, pp, 1-23, 2016. Article (CrossRef Link)

[18] Mostafaei, Habib, and Mohammad Reza Meybodi, "An energy efficient barrier coverage algorithm for wireless sensor networks," *Wireless personal communications* 77(3), pp. 2099- 2115, 2014. Article (CrossRef Link)

[19] Zhu S, Setia S, Jajodia S, "Leap: Efficient security mechanisms for large-scale distributed sensor networks," in *Proc. of 10th ACM conference on computer and communications security*, 2003. Article (CrossRef Link)

[20] Blom R, "an optimal class of symmetric key generation systems," *Brickel EF(ed) Cryptographic LNCS*, vol. 740, Springer, Heidelberg, pp 471-486, 1985. Article (CrossRef Link)

[21] Blundo C, De Santis A, Herzberg A, Kutten S, Vaccaro U, Yung M, "perfectly-secure key distribution for dynamic conference," *Brickel EF(ed) Cryptographic LNCS*, vol. 740, Springer, Heidelberg, pp 471-486, 1993. Article (CrossRef Link)

[22] Liu D, Ning P, Li R, "Establising pairwise keys in distributed sensor network," *ACM transactions information systems Security* 8(1), pp 41-77, 2005. Article (CrossRef Link)

[23] Chan H, Perrig A, Song D, "Random key pre-distribution scheme for sensor networks," in *Proc. ofIEEE symposium on security and privacy*, pp 197-213, 2003. Article (CrossRef Link)

[24] Liu D, Ning P, "establishing pairwise keys in distributed sensor networks," in *Proc. of 10th ACM CSS'03 Washington DC*, 2003. Article (CrossRef Link)

[25] Zuhu S, Setia S, Jajodia S, "Leap+: efficient security mechanisms for large-scale distributed sensor networks," *ACM Transactions sensor networks* (TOSN) 2(4), pp. 500-528. Article (CrossRef Link)

[26] Das AK, "A random key establishment scheme for multi-phase deployment in large-scale distributed sensor network," *International journal of information security 11*(3)", pp.189-211, 2012. Article (CrossRef Link)

[27] Castelluccia C, Spognardi A, "Rok: a robust key pre-distribution protocol for multi-phase wireless sensor networks," in *Proc. of IEEE third international conference on security and privacy in communications networks and the workshop*, pp. 351-360, 2007. Article (CrossRef Link)

[28] Rahman M, Sampalli S, "An Efficient Pairwise and Group Key Management Protocol for Wireless Sensor Network," *Wirel Personal Communication* 84(3):2035-2053, 2015. Article (CrossRef Link)

[29] Sarimuraat S, Levi A, "HaG:hash graph based key pre-distribution scheme for multiphase wireless sensor networks," in *Proc. of IEEE international conference on communications(ICC)*, PP:2079-2083, 2013. Article (CrossRef Link)

[30] Suganthi N, Vembu S, "An efficient pairwise and group key management protocol for wireless sensor network," *International Journal of computer communication and control 9(1)*, pp. 71-78, 2014. Article (CrossRef Link)

[31] Huyen, N. T. T., Jo, M., Nguyen, T.-D. and Huh, E.-N., "A beneficial analysis of deployment knowledge for key distribution in wireless sensor networks," *Security Comm. Networks*, vol.5(5), pp 485–495, May 2012. Article (CrossRef Link)

[32] I. Butun, M. Erol-Kantarci, B. Kantarci and H. Song, "Cloud-centric multi-level authentication as a service for secure public safety device networks," *IEEE Communications Magazine*, vol. 54, no. 4, pp. 47-53, April 2016. Article (CrossRef Link)

[33] Q. Xu, P. Ren, H. Song and Q. Du, "Security Enhancement for IoT Communications Exposed to Eavesdroppers with Uncertain Locations," *IEEE Access*, vol. 4, pp. 2840-2853, 2016. Article (CrossRef Link)

B. Premamayudu is pursuing Ph.D at JNT University, Kakinada in Computer Science and Engineering stream. He is currently associate professor in Department of Information Technology at Vignan Foundation for Science, Technology and Research University, Guntur, India. His research interest focuses on Security in wireless sensor networks, IoT and future networking.

Dr. Koduganti Venkata Rao received the Ph.D degree from Andhra University in Computer Science and Systems Engineering in 2008. He is currently Professor in the department of Computer Science and Engineering and Dean IQAC at Vignan's Institute of Information Technology, Visakhaptnam. His major research interests include key management, authentication protocols and light weight protocol analysis and design in networks.

Prof. P. Suresh Varma M.Tech(CST),Ph.D.(CSE), FIE, FIETE, SMCSI, AMIE, MISTE, SMORSI, MIISA, MISCA Professor of Computer Science and Engineering, Principal, University College of Engineering, & Dean Faculty of Engineering and Technology & Webmaster of University Website Adikavi Nannaya University Rajah Rajah Narendra Nagar, Rajahmundry - 533296,AP, INDIA. His major research interests include Software Engineering, Data Communication, and Computer Networks. Email : vermaps@yahoo.com Web : www.vermaps.com wwww.nannayauniversity.info 09848375600 (Mobile)