I J C T A, 9(24), 2016, pp. 109-117 ISSN: 0974-5572

A Scalable and Resilient Dynamic Key Management Scheme for Wireless Sensor Networks

Premamayudu B.*, K. Venkata Rao and P. Suresh Varma*****

ABSTRACT

Today we are using tiny devices to make our life comfortable. These devices are also helpful to alert the society about unwanted happenings and discover the hidden knowledge from the environment/nature. In this process, the communication among the devices must to be reliable, robust, and more secure. But these networked devices (WSNs) are operated with limited resources like computation, communication, memory, and battery power. These types of networks surely require energy efficient and lightweight communication systems/protocols. Wireless sensor network (WSNs) of nodes often handle sensitive information. So, the security/privacy (i.e. confidentiality, integrity, and availability (CIA)) of such networks is a typical baseline requirement.

These lightweight security solutions do not support traditional security solutions, which are used in computer networks and other networking i.e., Asymmetric Cryptography is not preferable approach for this type of networks. Trustworthy cryptographic security solutions to preserve in WSN are possible with good key management. This paper presents a dynamic key establishment and management systemfor WSNs. Each sensor can be pre-loaded with session key before deployment. The session keyensures the establishment of pairwise keys in the network with a small amount of mathematical computation. The analytical study defines the key connectivity parameter of proposed scheme with respect to existing schemes. In addition, the security analysis presents the preventions of different kind of security attacks including node compromising attack.

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

1. INTRODUCTION

Ubiquitous and Pervasive applications such as heal-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). Figure 1 shows the general WSNs structure. Sensor nodes are operated with tiny battery and are deployed in hostile environments to continuously sense real time information such as pressure, light, movement, moisture, temperature, and so on [22, 30, 11, 19].

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 collaborate data to a remote station through the external network such as internet for further processing [26].

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

* Department of Information Technology, Vignan's University, Vadlamudi, Guntur (Dt), Andhra Pradesh, India, *Email: premamayudu@gmail.com*

** Department of Computer Science and Engineering, Vignan's Institute of Information Technology, Gajuwaka, Visakhaptnam (Dt), Andhra Pradesh, India, *Email: vrkoduganti@gmail.com*

*** AdikaviNannayaUniversity, Rajahmundry, W. Godavari (Dt), Andhra Pradesh, India, *Email: vermaps@yahoo.com*

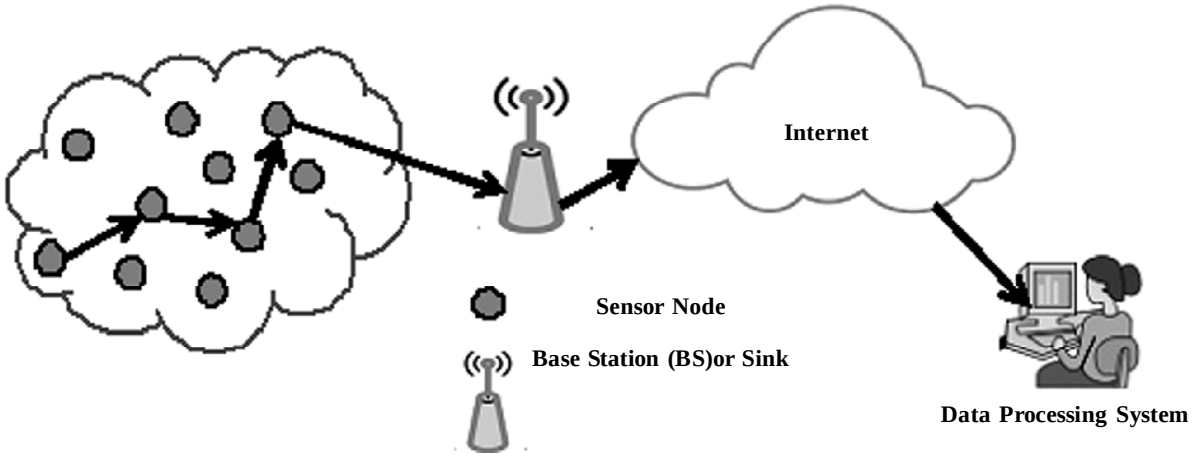

Figure 1: Wireless Sensor Network

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.

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. 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 [9, 12, 15]. The proposed systems cab bed 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 [2, 3, 14, 20, 27, 28, 29]. 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 [21, 25, 28]. 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 in each sensor node. Any pair of sensors in the network finds the common key between sub set of keys and use as pairwise key. This solution prevents the node capture attack, but it can't provide scalability and not suitable for large scale WNSs.

Blom [32], proposed the matrix related key management system where symmetric matrix $K_{N\times N}$ provides all pairwise keys of N sensor nodes in the network. The element in i^{th} row and j^{th}column of this matrix is the key between sensor nodes i and j to establish secure communication. $K = A \times G$ where $G_{(\lambda+1)\times N}$ is a public matrix, $A_{Nx(\lambda+1)}$ is private matrix and $A = (D \times G)^T$. D is $(\lambda + 1) \times (\lambda + 1)$ randomly generated secret matrix available with trusted party like base station or sink. Each sensor node i is pre-loaded with i^{th} row of A and j^{th} 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 $2 \times (\lambda + 1)$ keys.

Blundo et al. [31] 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 [27] 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 any sensor node neighboring nodes have m different keys, then the keys with 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 [18, 23, 24], but they used more memory space than the EG's [27] solution.

In [16, 25], LEAP (localized encryption and authentication protocol) scheme is presented. This scheme is suitable for heterogeneous or hierarchical wireless sensor network. LEAP needs five keys: (i) global key (ii) pairwise key (iii) cluster key (iv) individual key (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 [7, 13, 18] 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. The 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 to calculate the key indexes of i. Then the neighbor checks if any have common shared keys with i. In this scheme, key pool is refreshed using the XOR operation and hash function. For a key chain on m = 150 keys, the sensor should perform 150 XOR hash operations at each key refresh. For instance, $G_w = 5$, each sensor node computes 150×5 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. [1] modifies the deterministic matrix based key management solution [6] to support key refresh and pre-deployment addition of nodes. First phase is same as [32], but public matrix is generated by node itself. Suganthi et al.[4] 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.

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 devoted to the description our proposed key establishment, key refresh/revocation and new sensors addition after deployment. Our scheme minimizes the storage space occupation to stock key material and energy consumption. A session key (K_S) is global value to entire network. For every session a new session key 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 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 definesthe notations used in this scheme.

Table 1
Notations

Notation	*Description of Notation*
S_i	i^{th} 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
$\oplus$	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. Hence, two neighboring sensor nodes S_1 and S_2 compute a pairwise key in the following fashion:

S_1: generates a 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 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 $N_2 = (N_2 \oplus K_S) \oplus K_S$ and then determines the pairwise key $K_{1,2} = H_{KS}(N_1\|N_2\|S_1\|S_2)$

S_2: Computes $N_1 = (N_1 \oplus K_S) \oplus K_S$ and then determines the pairwise key $K_{1,2} = H_{KS}(N_1\|N_2\|S_1\|S_2)$

Base station and all sensors can setup the pairwise keys with other sensor nodes which are in the same communication region. After establishing pairwise with their neighbors, each sensor node removes the session key ($K_{S)}$ from the storage. Therefore, sensor node can't determine the nonce value of its neighbors without session key, because nonce value is XORing with session key.

3.2. Key Refresh/Revocation

Key refresh is accomplished in two cases. In one case, key refresh is done periodically to defend eavesdropping attacks. Meaning that pairwise key is updated time to time. In other case, key refresh is

made to remove captured sensor in to network. BS enables the key refresh in the network. It generates a new session key $K2_S$ and nonce N_{BS}. Then the BS encrypts the new session key and nonce with the current pairwise key and broadcasts to its neighbors.

$BS \rightarrow S_i : \{K'_S \| N_{BS} \| L \| CTR\}K_{i,BS}$, 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^2_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^2_S over the network. After 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 using 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

This section presents the key connectivity in the network. Our scheme, Matrix-based scheme[32], and polynomial scheme [31] are deterministic. Meaning that number of nodes (d) is in the same communication range in the network. The value d defines the key connectivity in the network.

Table 2 presents the key connectivity analytical results of our scheme, Blom's scheme [32], Blundo et al.'s scheme [31], EG Scheme [27] and HaG Scheme [6] for key pools size of S = 10,000.

5. SECURITY ANALYSIS

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

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

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, 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 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 security in the applications. Most of the cases, this value will be decided after applying the many test cases on the sensor network.

5.2. Black Hole and Worm Hole attacks

A block hole and wormhole whole 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 wormhole 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 wormhole 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. 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 of node using hash and MAC functions. Hence, our scheme prevents Sybil attacks.

5.5. Forward and Backward Compatibility

A new session key is loaded in 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 eliminated node identities list also 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 or key refresh operations are initiated by the base station. Therefore, our scheme ensures the forward and backward message security.

6. CONCLUSION

In WSNs, security of the network is indirectly depends on the key management service in cryptographic mechanism. The KMS is the challenge issue in resource constrained networks. In this paper, the proposed KMS is efficient and scalable to minimize the resource utilization. The proposed scheme enables the each sensor to compute the unique pairwise keys between pair of sensor nodes using session key. This scheme requires the minimum storage and communication overhead. The experimental results show that the scheme required less storage and energy when compared with existing KMSs. In addition, it prevents the node capture attack.

REFERENCES

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

[2] Jilna P, Pattathil DP, "A key Management Technique based on Elliptic Curves for Static Wireless Sensor Network", Security communication networks, pp. 3726-3738, 2015.

[3] 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. doi:10.1155/2015/627548.

[4] 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.

[5] Bechkit W, Challal Y, Bouabdallah A, Tarokh V, "A highly scalable key pre-distribution scheme for wireless sensor networks, IEEE Transactions wireless communication 12(2), pp. 948-959, 2013.

[6] Sarimuraat S, Levi A, "HaG:hash graph based key pre-distribution scheme for multiphase wireless sensor networks", IEEE international conference on communications(ICC), pp. 2079-2083, 2013.

[7] 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.

[8] Min X, Wei-Ren S, Chang-Jiang J, Ying Z, "Energy efficient clustering algorithm for maximizing lifetime of wireless sensor networks", AEU-international journal electrical communication 64(4), pp. 289-298, 2010.

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

[10] Chen CL, Lin IH et al, "location-aware dynamic session-key management for grid-based wireless sensor network", Sensors 10(8), pp. 7347-7370, 2010.

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

[12] Alemdar A, Ibnkahla M, "Wireless Sensor Networks:applications and challenges", 9th international symposium on signal processing and its applications", pp. 1-6, 2007.

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

[14] Liu A, Kampanakis P, Ning P, "TinyECC:eliptic curve cryptography for sensor networks", Tiny ECC Software, 2007.

[15] 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.

[16] 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.

[17] Cvrcek D, Svenda P, "Smart dust security-key infection revisited", Electronic notes theory computer Science 157(3), pp. 11-25, 2006.

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

[19] Yoneki E, Bacon J, "A Survey of wireless sensor network technologies:research trends and middleware's role", University of Cambridge TR 646, Cambridge, 2005.

[20] Watro R, Kong D, CutiSf, Gardiner C, Lynn C Kruus P, "Tinypk: Securing sensor networks with public key technology", 2nd ACM workshop on security of ad hoc and sensor networks, pp. 59-64, 2004.

[21] Dutertre B, Cheung S, Levy J, "lightweight key management in wireless sensor network by leveraging initial trust", technical report SRI-SDL-04-02, SRI international, 2004.

[22] Anderson R, Chan H, Perrig A, "Key infection: smart trust for smart dust", 12th IEEE international conference on network protocols", pp. 206-215, 2004.

[23] Chan H, Perrig A, Song D, "Random key pre-distribution scheme for sensor networks", IEEE symposium on security and privacy, pp. 197-213, 2003.

[24] Liu D, Ning P, "establishing pairwise keys in distributed sensor networks", 10th ACM CSS'03 Washington DC, 2003.

[25] Zhu S, Setia S, Jajodia S, "Leap: Efficient security mechanisms for large-scale distributed sensor networks", 10th ACM conference on computer and communications security, 2003,

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

[27] Eschenauer L, Gligor VD, "A key management scheme for distributed sensor networks", 9th ACM conference on computer and communications security, pp. 41-47, 2002.

[28] Lai B, Kim S, Verbauwhede I, "Scalable session key construction protocol for wireless sensor networks", IEEE workshop on large scale real time and embedded systems(LARTES), pp. 1-7, 2002.

[29] Perrig A, Szewczyk R, Tygar J, Ying Z, "Spins: security protocols for sensor networks, Wireless Networks 8(5), pp. 521-534, 2002.

[30] Carman DW, Kruus PS, Matt BJ, "Constraints and approaches for distributed sensor network security", NAI Labs Technical Report #00-010, 2000.

[31] 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.

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