Kirkuk University Journal /Scientific Studies (KUJSS)
Volume 13, Issue 4, December 2018, pp. (17-33)
ISSN: 1992-0849 (Print), 2616-6801 (Online)

Key Management Distribution Scheme in Wireless Sensor Network Based on Knapsack Algorithm

Kameran Ali Ameen
Department of Computer Science, College of Computer Science and Information Technology,
Kirkuk University, Kirkuk, Iraq.
Kameran.ameen@gmail.com

Abstract

Key management in Wireless Sensor Network (WSN) is a complex task due to its nature of environment, limited resources and open communication channel. In addition, wireless communication poses additional threats to the critical information being sent and received over there. WSN are necessary to be protected from different attacks. But, the major problem to secure WSN is a key distribution after deploying the sensor nodes in specific area. This paper examines the design of an efficient key management distribution scheme for WSN. The proposed method based on knapsack algorithm that requires generating a series of vectors to encrypt the private keys only that will be sent to the cluster heads by base station. The same method is used by each cluster head to encrypt only the keys that will be sent to their members. The simulation results showed that the proposed key management method can offers efficient security prerequisites, suitable scalability, and connectivity to achieve authentication.

Keywords: Network Model; key Management; Knapsack Algorithm; Security.

DOI: http://doi.org/10.32894/kujss.2018.13.4.2

مخطط ادارة توزيع المفاتيح في شبكات الاستشعار اللاسلكية بالاستناد على خوارزمية الحقيبة

كامران علي أمين
قسم علوم الحاسوب، كلية علوم الحاسوب وتكنولوجيا المعلومات، جامعة كركوك، كركوك، العراق
Kameran.ameen@gmail.com

الملخص

تعد إدارة المفاتيح في شبكة الاستشعار اللاسلكية مهمة معقدة نظرًا لطبيعة البيئة والموارد المحدودة وقنوات الاتصال المفتوحة. بالإضافة إلى ذلك، تشكل الاتصالات اللاسلكية تهديدًا إضافيًا للمعلومات المهمة المرسلة والمستقبلة. لذا من ضروري أن تكون شبكة الاستشعار اللاسلكية محمية من الهجمات المتنوعة. لكن المشكلة الرئيسية لأمنية شبكة الاستشعار اللاسلكية هي توزيع المفاتيح بعد نشر عقد الاستشعار في منطقة محددة. تبحث هذه الورقة تصميم مخطط فعال لإدارة توزيع المفاتيح لشبكة الاستشعار اللاسلكية. الطريقة المقترحة يعتمد على خوارزمية الحقيبة التي تتطلب إنشاء سلسلة من المتجهات لتشفير المفاتيح الخاصة فقط، والتي سيتم إرسالها إلى رؤساء المجموعات بواسطة المحطة الرئيسية. الطريقة نفسها تستخدم بواسطة رئيس كل مجموعة لتشفير المفاتيح التي سيتم إرسالها إلى أعضائها المرتبطين بها. نتائج المحاكاة أظهرت أن الطريقة المقترحة لإدارة المفاتيح يمكن أن توفر متطلبات أمنية فعّالة، قابلية مناسبة للتوسيع وتحقق المصادقة.

الكلمات الدالة : نموذج الشبكة، ادارة المفاتيح، خوارزمية الحقيبة، الأمنية.

DOI: http://doi.org/10.32894/kujss.2018.13.4.2

1. Introduction:

Recently, the tremendous development in the electronics technology ‘wireless communication’ has enabled the development of low power, low-cost, small memory, multifunctional sensor nodes [1-3]. WSNs are merely defined as a large collection of sensor nodes, each equipped with its own sensors, data processor, and short-range radio transceiver [1,2]. Due to the characteristics of WSNs, they have many applications in battlefield environment, health, disaster, space, environmental threats, and other industries sectors [4], [5]. As, indicated by numerous specialists, the use of WSNs is creating a revolution of the concepts of different day-to-day activities in near future [6].

One of the main problems in WSNs is ensuring communication security, particularly when they are deploying in critical domains where an attacker can be easily captured nodes and manipulate [7]. One security aspect that obtains a great deal of importance in WSNs is the field of key management [2,8]. Key management technique is the method in which keys are generated, protected, stored, transferred, used among the authorized node and can be canceled when they do not needed. Key management builds the keys necessary to afford security requirement that include integrity, confidentiality data, and authentication nodes. Yet, presenting best key management in WSNs is a difficult task due to the unknown network topology prior to deployment. The main aim of key management in WSNs is to distribute the keys by a secure method and creating secured links among the sensors in the formation phase network [1,7].

Moreover, there are two types of cryptography; the first type is symmetric key cryptography that use one key for encryption and decryption and it is faster to execute like Advanced Encryption Standard (AES) [6]. The second type is known asymmetric cryptography or called private key system that use two keys one key for encryption it's called private its secure but the second its public key for decryption like Rivest-Shamir-Adleman algorithm (RSA), Elliptic Curve Cryptography (ECC), and Elliptic Curve Computational Diffie-Hellman (ECCDH) to provide best security [6,8]. Nowadays, WSNs have attracted significant interest in the engineering community and among researchers. In fact, the wireless channels are not secure. In addition, Due to the depending on the keys to make a connection between the nodes in a radio channel (open environment), these keys are easily prone to attack. Thus, the main challenge in the WSNs is the security of the key distribution that is used for the connection. These encryption keys are the critical issue for getting a high security which is the core of this paper by utilizing Knapsack Algorithm [9,10] that requires generating a series of vectors. Finally the roadmap of the paper is organized as follows. Section 2 explains literature survey for WSN. In section 3, discussion the knapsack process. Describing the used network model is depicted in section 4. In section 5, credible preliminaries is explained. While, section 6 illuminates the network phases and including proposed method. The example of the proposed method is explained in section 7. Results and discussion are illustrated in section 8. Ultimately, in section 9, we present our concluding remarks.

2. Literature Survey:

Gianluca Dini and Ida Maria Savino in [11] suggested a key distribution protocol in WSNs based on the key chains by using symmetric ciphers and one-way functions on the next key in the chain. Prior the distribution, these nodes can be either exchanged or installed through a secure channel. During the applying of one-way hash functions, a key authentication can be obtained.

Song Ju in [8] suggested a combination of the Elliptic Curve Diffie-Hellmann with symmetric key cryptography and hash chain to establish a lightweight key protocol in WSNs. It is built on the single-hop network that all the sensors nodes can be communicate with the other nodes. Before the distribution of the nodes in specific area, it is preload same initial key to all nodes as initial trust phase. From the research and analysis, it shows that protocol can less computation and communication complexity compared to other protocols.

Md. Ibrahim Abdullah in [12] suggested technique to the management of keys distribution within cluster nodes and avoid a node-capturing problem update the keys at the orderly interval. The key distribution is entirely local. The network key discards after they are distribute of keys. To prevent node capture their proposal a key update technology. When the keys are updated, the network key is re-formed. Here, technique authenticates a group of cluster nodes rather of each node. This technique has some communication overhead because receiving key update packets from the base station (BS) and it can prevent the common attacks of hierarchical sensor network and decrease the node capturing attacks.

Danyang Qin, Shuang Jia, Songxiang Yang, ErfuWang, and Qun Ding in [13] have proposed a lightweight authentication and key management protocol for WSNs. It solves the problem of wicked nodes happening through the process of networking and to provide high security beside low cost. The mobile sensor nodes need to be authenticated that is the important condition, where, the keys in proposed scheme will be dynamically created and adopted for security protection. When captured of the node or the keys are being compromised by the attackers cannot use the previous keys. The analysis shows that the proposed scheme offers high security with less energy consumption for wireless sensor networks, notably when applying it with mobile sensors.

3. Knapsack Process:

This section describes the knapsack process. Knapsack algorithm involves that we create a series of vectors called (a_i) over positive integers. There are many techniques of creating vectors. For the sake of explanation, we will take the first value as 1 and subsequent values as multiples of n. like

$a_i = (1, n, n_2, n_3 \ldots n_m)$ $(1 \leq i \leq m)$

Now, n may be supposed to be some random positive integer.

Then let us discover how the signed message is subjected to the knapsack procedure. Say, k_i is converted as follows, which can be represented in its binary number format as such:

$k_i = (c_1, c_2, c_3 \ldots, c_m)$

According to the knapsack process, a compute a cumulative sum S .

$$S = \sum_{i=1}^{m} a_i \; k_i \tag{1}$$

In the end, signed message form, k_i value is swapped by its equivalent S. In the final that to send to the consignee. The consignee has all the relevant data for reversing the knapsack process and recovering the bit pattern of S. (For example, the consignee knows the (a_i) series. The consignee reverses the received message S into k_i. Let us discuss how to reverse the knapsack procedure, via using an example. Consider eq. (1) which is the knapsack representation of k_i. The value of k_i recovered in an iterative manner as the following:

$S-n^m$

If the value is positive number i.e., $S-n^m > 0$, then a binary bit 1 is allocated at the $(m)^{th}$ location. If, however, the value is negative number, then a 0 bit is allocated. Now subtract n^{m-1} from the current R. Depending on whether it its positive or negative, allocate 1 or 0 at the relevant bit location. This subtraction continue until the (a_i) series is exhausted. Details of the knapsack algorithm and the reverse knapsack procedure are presented in [10].

4. Network Model:

This section briefly presents hierarchical structure of sensor network as demonstrated in Fig. 1. In our network model.

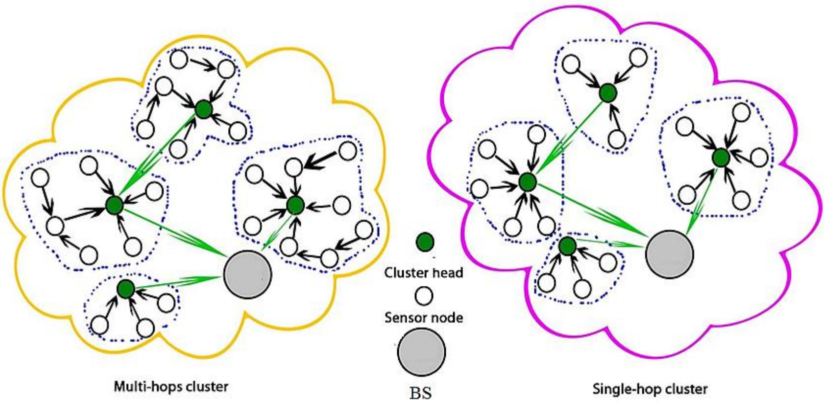


Fig. 1: Hierarchical model

- **The Base Station (BS)** responsible for connecting the wireless sensor network to an outside network. It is considering trustworthy for the network and it performs the highest capabilities in terms of computing power, energy, and storage capacity that are assumed. In addition, it is able to connect directly to each sensor nodes in the network [1,7,14].

- **Cluster Heads CHs** manage members' joining and disjoining procedures. Thus it is responsible for the management of the nodes in the cluster after formation and the data transmission from its cluster members to the base station. CHs can able connect directly to the base station through one hop. Moreover, CHs equipped with a widely higher amount of resources than sensor nodes in the network, such that high processing, high storage, and larger communication [1,7,14].

- The location of the sensor nodes is in the lowest level of the hierarchy. They are low-cost devices with limited computing, energy, storage, and power capabilities. The main task of a sensor node is to collect the information and transmission to the cluster head through one hop or more [1,7,14] as shown in Fig. 1, where, node location is random in sensor arena. It nodes remain fixed after deployment through the network process [14].

5. Preliminaries:

The credible presumptions are used as previously applied in most of the security schemes.

- The BS is known and in secured location, trusted by all members, and with sufficient resources.

- BS has authentication scheme for every node after deployment.
- The BS is qualified for creating the private key for all CHs, while each CH is qualified for creating the private key for all members in the network.
- Each CH can reach the BS and vice versa in the network.
- Every L- node and CHs are static and are randomly deployed in the area moreover are equipped with tamper-resistant hardware and Global Positioning System (GPS). Global Positioning System receivers, such as navigation devices, pick up the signals. It uses to calculate the position, time of and speed.

6. The Network Phases:

In this section the details of the proposed scheme are explained. It has the following 4 phases:

6.1 The Phase of the Key Pre-distribution:

BS has a list of sensors node ID, a share key between nodes and CHs.

- The CH and L-node are preloaded with procedures algorithm1 & 2.
- Every L-node includes its ID with the IDs of all CHs which are deployed in preload duration.
- Every L-node and CHs are pre-loaded with unique shared key with the BS before deploy.
- Every CH is preloaded with the ID of the BS prior deployment.

6.2 Nodes Distribution Phase:

This phase occurs directly after nodes distribution in the selected area that consists of 100 nodes. They have uniformly and randomly distributed exclusively in the area of size 100× 100 m^2 [15], [16] as shown in Fig. 2.

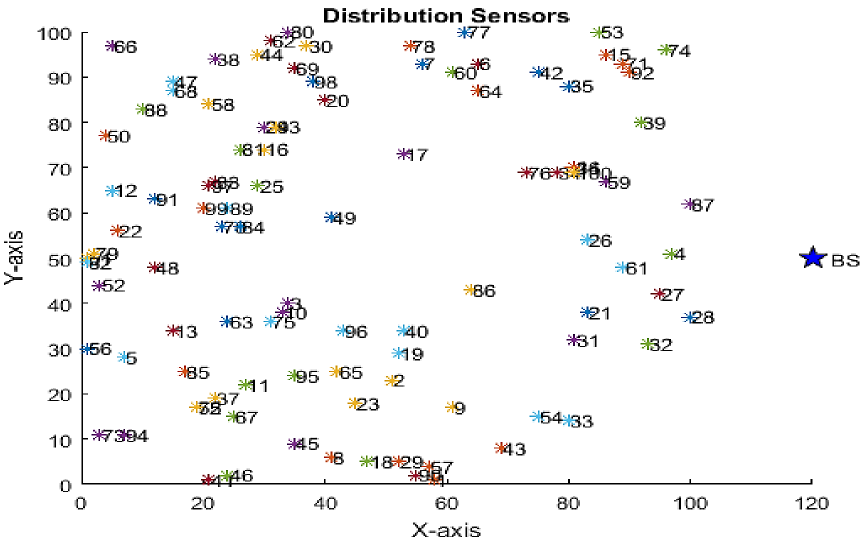
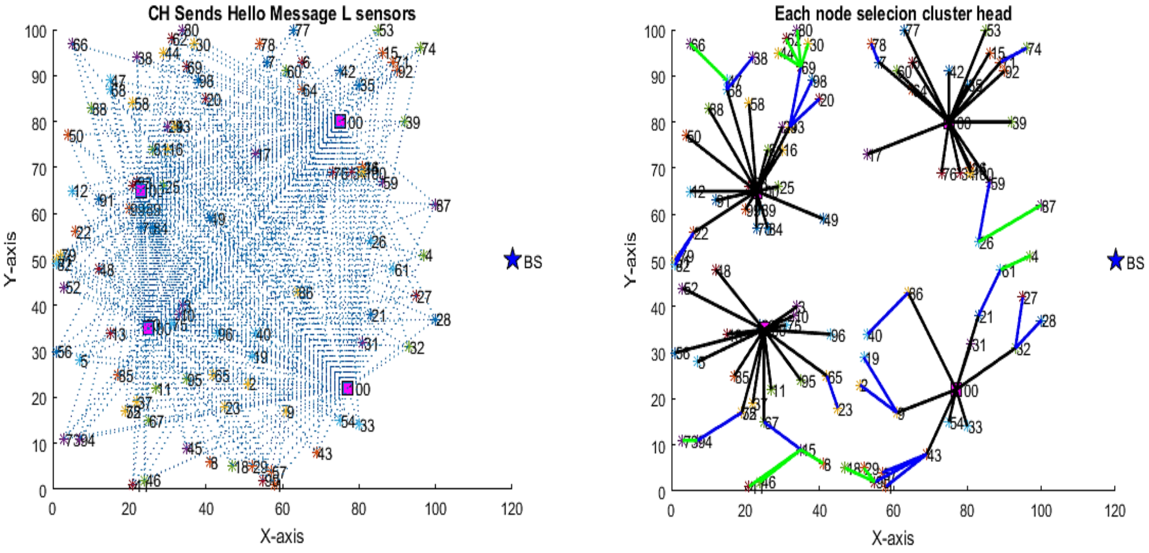


Fig. 2: Distribution of 100 nodes in 100×100 m^2.

6.3 Cluster Establishing Phase:

The cluster formation is started after the sensors deployment in the specific area as depicted in the following steps:

- Each CH-sensor broadcasts message called M which includes the ID and its location with a random delay.
- Each L-sensor may receive messages coming from more than one CH-sensor. Then, it chooses the CH-sensor whose M message has the best signal strength.
- Each L-sensor broadcasts message which includes the ID and its location and stores the information of neighboring nodes. Afterward, each L-sensor sends its ID and location information to the CH-sensor by GPSR [14,17].
- After receiving the information from L-sensors, each CH-sensor constructs different routes based on location information of L-sensors within its cluster (1, 2, or 3 hops). As shown in Fig. 3 a and b [14,17].



(a) (b)

Fig. 3: Cluster Establishing Phase.

6.4 The proposed Key Creation and Distribution:

In this section, the BS generates a private key for each CH by using a random algorithm (only to choose the keys) and send them to the clusters. For the security issues, these keys must be encrypted before sending them by using Algorithm1 and decrypted them after receiving by using Algorithm 2. In addition, each CH is generating the private key for its members by the same procedures. This distribution is shown in Fig. 4.

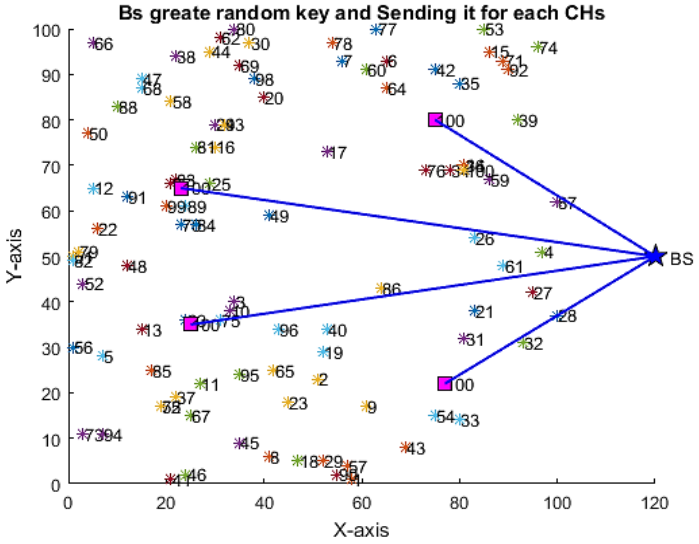


Fig. 4: The BS sends private key to each CH$_i$

Algorithm 1 (Key Generation and Encryption in BS, distribution to CH_i)

```
1.  Input:  i , k , share key , are integer. Where a share key is common between $CH_i$ and BS.
2.  Input: $ID_{CH[i]} = ID_{CH1}$………….$ID_{CHi}$. Where $ID_{CHi}$ is integer.
3.  $CHk_i$ = [ $CHk_1$……..…$CHk_i$ ] using randomly algorithm to choose private  key from [1, $n - 1$] for each $CH_i$.
4.  $a_i$ =Series of vectors, where ($a_i = 1, n_1, n_2, n_3 \ldots n_m$ ( $1 \le i \le m$) ) and m is the length of the binary bit string.
5.  $K_i = b_1, b_2, b_3 \ldots, b_m$ ($1 \le i \le m$) [Binary value of $CHk_i$].
6.  Compute a cumulative sum $K_i$ according to eq.1 which is knapsack algorithm.
7.  Compute $Ks_i = KCH_i$ + share key.
8.  M = [$Ks_i$ || $ID_{BS}$ || $ID_{CHi}$]. //where M is the encrypted message ready to send and '||' is a concatenation.
9.  Broadcast M.
10. End.
```

Algorithm 2 (Key Decryption by CH_i)

```
1. Input:  i , k , share key , $ID_{CHi}$  are integer and k private key.
2. M= message receiver.
3. For I =1 to n
       3.1 Separate message to three parts. ($Ks_i$, $ID_{CHi}$, $ID_{BS}$).
       3.2 Check. If ($ID_{CHi}$ and $ID_{BS}$) is valid.
       3.3 Then accept message M.
       3.4 Save $Ks_i$.
4. Compute $KCH_i = Ks_i$ – share key.
5. Compute if, $KCH_i - n_m > 0$ then a binary bit 1 is assigned at the $(m)^{th}$ position. The current value is $KCH_i = KCH_i - n_m$.
6. Else, $KCH_i - n_m < 0$ then a 0 bit is assigned and the $KCH_i$ remains unchanged.
7. Continue to compute subtract $n_m - 1$ from the current $KCH_i$. Depending upon whether it is > 0 or < 0, assign 1 or 0 at the relevant bit position. Continue this subtraction until the $a_i$ series is exhausted. This will recover the binary bit pattern of $KCH_i$.
8. Save $K_i$ which the private key of CH.
       8.1   Else, ($ID_{CHi}$ and $ID_{BS}$) is not valid.
       8.2   Refuse message M.
   9. End.
```

7. The Example of the Proposed Method:

Assume that the BS choose the integer (99) as a (Key) for one of the CH_i by randomly algorithm. Take with the regard that the applying of Knapsack algorithm requires a series of vectors which are defined by a_i. For illustration “1” represents the first value that is to be taken, and subsequent values are multiples of n such that:

$a_i = (1, n, n_2, n_3\ldots n_m (1 \leq i \leq m))$.

Suppose that n takes some random integer less than 5 like
$a_i = n_m = \{ 1, 5, 25, 125, 625, 3125, 15625 \}$

Where: $n = 5$ and m as (*7*) is the length of the binary bit-string that’s meaning converting the chosen key to binary.
$k_i = (b_1, b_2, b_3\ldots, b_m (1 \leq i \leq m))$

$K_i = 99 \longrightarrow 1100011$ [binary value of 99].
Therefore calculation is done by eq. (1), which yields the results in Table 1.

Table 1: Calculate KCH_i.

a_i	1	n_1	n_2	n_3	n_4	n_5	n_6
k_i	b_1	b_2	b_3	b_4	b_5	b_6	b_7
n=5 / a_i	1	5	25	125	625	3125	15625
$K_i = 99$	1	1	0	0	0	1	1

$KCH_i = 1 + 5 + 0 + 0 + 0 + 3125 + 15625 = 18756$. As illustrated in Fig. 5.

(Where a share key is common between CHi and BS = 200). Then, compute Ks_i according to eq. (2).

$$Ks_i = [KCH_i + \text{share key}] \quad (2)$$

$Ks_i = 18756 + 200 = 18956$.

Next, append Ks_i with ID_{BS} and ID_{CHi}. Finally, BS broadcasts message (M) by depicted by eq. (3).

$$BS \rightarrow CH\text{: } Ks_i \,||\, ID_{BS} \,||\, ID_{CHi} \quad (3)$$

Afterward, each CH_i receives M from BS. CH_i decrypts message (M) to recover the private key of the CH (99). Initially, separate message to three parts (Ks, ID_{CHi}, and ID_{BS}).

Then checks the ID of BS and CH_i if they are matched then accept message elsewhere refusing a message. After that CH_i calculates KCH_i according to eq. (4).

$$KCH_i = Ks_i - \text{share key} \quad (4)$$

KCH_i= 18956 – 200 = 18756, then apply knapsack algorithm as the following:

The KCH_i value is recovered in an iterative method by depicted by eq. (5), which yields the results in Table 2 to calculate K_i that is private key of CH_i.

$$KCH_i - n_m \quad (5)$$

Table 2: Calculate KCH_i.

$KCH_i - n_m$	=	K_i
18756 − 15625	3131	1
3131 − 3125	6	1
6 − 625	-ve	0
6 − 125	-ve	0
6 − 25	-ve	0
6 − 5	1	1
1	1	1

Therefore, 1100011 = Ki = 99 (Read from bottom up).

8. Results and Discussion of the Proposed Scheme:

In this section, the results are evaluated and the security issue is analyzed. A comparison between the proposed scheme and the other schemes is discussed. Firstly, the power of Knapsack algorithm lies in the choosing of the a_i vectors. As said previously, there are many methods for selecting these vectors such as Maclaurin’s series and Taylor’s series [18]. Fig. 5 depicts the knapsack algorithm when the value of the private key is 99. There is a positive relationship between the strength of the security and value of n.

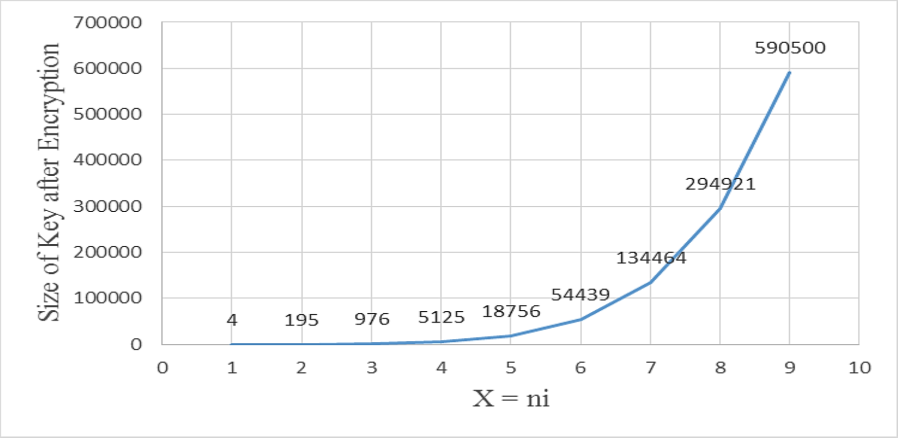


Fig. 5: The results when the value of the private key is 99

In addition, Table 3 shows the comparison between the proposed scheme with other schemes in terms of Security issues.

Table 3: Comparison in terms of Security issues

Issues	Proposed	Scheme [19]	Scheme [20]	Scheme [21]
Cryptographic mechanism	Knapsack Algorithm depends of vectors	PKI with ECC	Self-Certified Keys Cryptosystem and ECC	Symmetric cryptography SKC
Authentication	Maintained	Maintained	Maintained	Maintained
Confidentiality	Maintained	Not maintained	Not maintained	Not maintained
Integrity	Maintained	Not maintained	Not maintained	Not maintained
Scalability	Yes	Yes	Yes	No

Table 4 Indicates the node time consuming measured in milliseconds, show time consuming in each node for encryption and decryption message when sent. The proposed simulation using MATLAB R2013a program for MICAz with ZigBee / IEEE 802.15.4 protocols and transmit data rate 250 kbps.

Table 4: Time spent calculated in millisecond

Process	Time
Encryption	18 ms
Decryption)	13 ms

Finally, the full connectivity is achieved when 100 and 200 nodes are distributed respectively while the number of clusters are remains the same in both cases in the area of size 100 × 100m^2 as shown in Fig. 6 and Fig. 7. This is proving that the network is scalability.

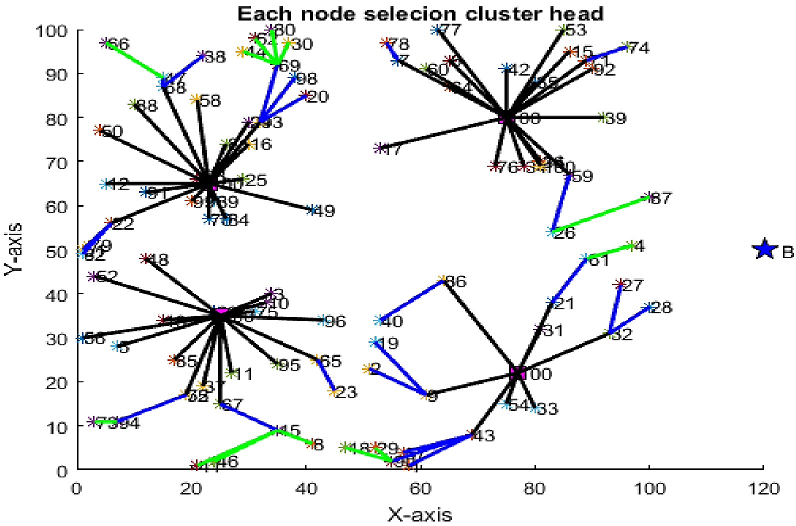


Fig. 6: Distribution of 100 nodes in 100X100 m^2

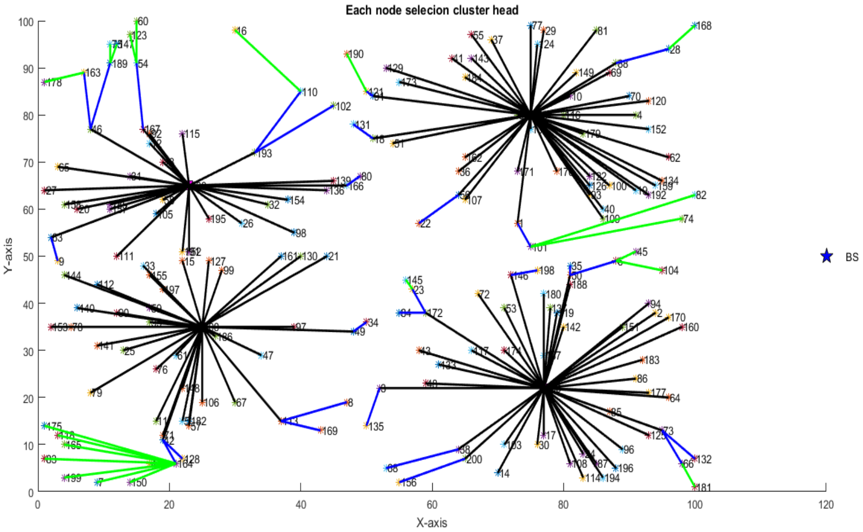


Fig. 7: Distribution of 200 nodes in 100X100 m^2

9. Conclusion:

Key distribution and management in WSNs are a crucial field and essential issue. In this paper we addressed key management problem in WSNs. we proposed a key management distribution scheme to improve network security by creating a secure communication using cryptographic techniques. Hence, providing the powerful key management scheme which meet all challenges is under research and development. Therefore, using Knapsack algorithm can achieve this goal. In this technique, the hierarchical architecture is used to create network depending on Knapsack algorithm based on the series of vectors to encrypt/decrypt keys. Furthermore, the result shown the security strength of Knapsack algorithm lies in the selection of the vectors. Finally, the performance comparison showed that the proposed method has a suitable performance compared to other algorithms in terms of Security issues, dominant scalability and connectivity.

References

[1] L. Gavrilovska, S. Krco, V. Milutinovic, I. Stojmenovic, and R. Trobec, **"*Application and Multidisciplinary Aspects of Wireless Sensor Networks*",** Springer, London, (2011).

[2] Z. Fei, B. Li, S. Yang, C. Xing, H. Chen, and L. Hanzo, **"*A Survey of Multi-Objective Optimization in Wireless Sensor Networks: Metrics, Algorithms and Open Problems*"**, IEEE Communications Surveys & Tutorials, 19(1) (2016).

[3] A. Lambebo and S Haghani, **"*A Wireless Sensor Network for Environmental Monitoring of Greenhouse Gases*"**, ASEE 2014 Zone I Conference, USA, University of Bridgeport, 3 (2014).

[4] S. Kumari, M. K. Khan, and M. Atiquzzaman, **"*User Authentication Schemes for Wireless Sensor Networks: A review*",** Journal of Ad Hoc Networks, 27, 159 (2015).

[5] D. Puccinelli and M. Haenggi, **"*Wireless Sensor Networks: Applications and Challenges of Ubiquitous Sensing*",** IEEE Circuits Systems Magazine, 5(3), 19 (2005).

[6] A. Tufail and K. Kim, **"*A Backbone Assisted Hybrid Key Management Scheme for WSN*",** International Conference on Information Society (i-Society 2011), UK, IEEE, 86 (2011).

[7] W. Abdallah, N. Boudriga, D. Kim, and S. An, "***An Efficient and Scalable Key Management Mechanism for Wireless Sensor Networks,***" 16th International Conference on Advanced Communication Technology, 3(4), 687 (2014).

[8] S. Ju and I. Technology, **"*A Lightweight Key Establishment in Wireless Sensor Network Based on Elliptic Curve Cryptography,*"** Intelligent Control, Automatic Detection and High-End Equipment (ICADE), IEEE International Conf., China, IEEE, 138 (2012).

[9] F. Delicato, F. Protti, L. Pirmez, and J. F. de Rezende, **"*An Efficient Heuristic for Selecting Active Nodes in Wireless Sensor Networks*",** Journal of Computer Networks, 50(18), 3701 (2006).

[10] S. Parvin, S. Han, Z. U. Rehman, A. Al Faruque, and F. K. Hussain, **"*A new Identity-Based Group Signature Scheme Based on Knapsack ECC*",** 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, (IMIS), Italy, IEEE, 73 (2012).

[11] G. Dini and I. M. Savino, **"*An Efficient Key Revocation Protocol for Wireless Sensor Networks*",** International Symposium on World of Wireless, Mobile and Multimedia Networks, USA, IEEE Computer Society, 450 (2006).

[12] I. Abdullah, **"*A Key Distribution and Management Scheme for Hierarchical Wireless Sensor Network*",** International Journal of Multimedia and Ubiquitous Engineering, 6(3), 1 (2011).

[13] D. Qin, S. Jia, S. Yang, E.Wang, and Q. Ding, **"*A Lightweight Authentication and Key Management Scheme for Wireless Sensor Networks*",** Journal of Sensors, 2016, 9 (2016).

[14] S. Hussain, A. Diop, Y. Qi and Qin Wang, **"*An Efficient and Secure Key Management Scheme for Hierarchical Wireless Sensor Networks*",** International Journal of Computer and Communication Engineering, 4(1), 155 (2012).

[15] J. Shen, A. Wang, C. Wang, P. C. K. Hung, and C. F. Lai, **"*An Efficient Centroid-Based Routing Protocol for Energy Management in WSN-Assisted IoT*",** IEEE Access, 5(8), 18469 (2017).

[16] S. G. Susila and J. Arputhavijayaselvi, **"*Innovative Energy Resourceful Merged Layer Technique (MLT) of Node Deployment to Enhance the Lifetime of Wireless Sensor Networks*",** Egyptian Informatics Journal, 16(1), 23 (2015).

[17] Z. Ying and J. Pengfei, **"*An Efficient and Hybrid Key Management for Heterogeneous Wireless Sensor Networks*",** Control and Decision Conference, China, IEEE, 1881 (2014).

[18] L. Liu and G. Mavidi, **"*Hybrid Localization Algorithm in Wireless Sensor Networks and ITS Application in Building Monitoring*",** Advanced Applied Informatics, IIAI 3rd International Conf., Japan, IEEE, 97 (2014).

[19] Z. Benenson, N. Gedicke, and O. Raivio, **"*Realizing Robust User Authentication in Sensor Networks*",** International Workshop on Real-World Wireless Sensor Networks (REALWSN), Italy 14, 52, (2005).

[20] C. Jiang, B. Li, and H. Xu, "***An Efficient Scheme for User Authentication in Wireless Sensor Networks,"*** Advanced Information Networking and Applications Workshops, 21st International Conf., Canada, IEEE, 438 (2007).

[21] S. Banerjee and D. Mukhopadhyay, ***"Symmetric Key Based Authenticated Querying in Wireless Sensor Networks,"*** The First International Conference on Integrated Internet Ad Hoc and Sensor Networks, France, 6 (2006).