KDS: Keyless Data Security for Wireless Sensor Networks

Charu Sharma, Rohit Vaid, and Kavita Gupta

Abstract Wireless Sensor Networks (WSNs) are becoming more popular and are also used in a variety of mission-critical applications. Security in these applications plays a significant role. However, these networks are constrained by a number of factors including limited computation capabilities, energy and storage capacity, unreliable communication, vulnerability to physical capturing and unsupervised activities. The main challenge is to retain security in the network despite these constraints. For secure transmission, key management plays a vital role. But re-keying is necessary when the node is compromised or after a specified number of rounds. So key refreshing increases communication overheads in the network which degrades the network performance. To overcome this problem, a Keyless Data Security (KDS) scheme for WSNs is proposed which eliminates the requirement for key management in the network during data transmission. Simulation results prove that the proposed scheme provides better performance without increasing communication overheads in the network.

Keywords Data partitioning · Key management · Wireless sensor networks

1 Introduction

Security becomes one of the major problems in WSNs [1]. During network setup in WSNs, the most important requirement is to establish cryptographic keys for future use. Key management is the mechanism in which pre-allocation of secret keys

C. Sharma (✉) · R. Vaid
CSE Department, M. M. Engineering College, M.M (Deemed to be University), Mullana, Ambala, Haryana 133207, India
e-mail: er.charusharma@mmumullana.org

R. Vaid
e-mail: rohitvaid@mmumullana.org

K. Gupta
University Institute of Computing, Chandigarh University, Gharuan, India
e-mail: 25.kavita@gmail.com

N. Marriwala et al. (eds.), *Mobile Radio Communications and 5G Networks*,
Lecture Notes in Networks and Systems 588,
https://doi.org/10.1007/978-981-19-7982-8_51

to every node is done by which only authorized node can interact with each other [2, 3]. Key management schemes are categorized as static and dynamic methods. In static key management schemes, once the key pre-allocation is done, the keys remain constant during the entire network's lifetime so it guarantees low level of security. In dynamic scheme, regeneration of keys is possible when necessary during the entire network's lifetime and is more secure.

1.1 Problems in Different Types of Key Management Schemes

A high level of security is required to transmit sensitive information over a network [4, 5]. A number of security-critical applications rely on different key management methods to operate. As sensor nodes (SNs) are randomly deployed in unsupervised and inaccessible areas, physical tampering is a major risk [6]. If a single key is used in the network and if this key is compromised by any single SN, the whole network is compromised. If multiple keys are used, then large numbers of keys are required to be managed and refreshed. The WSN must be capable to survive with the compromise of some of the SNs in the network. It is important to find out how many compromised SNs it takes to compromise the security of the entire network. And when any SN is compromised, these security-critical applications also demand a high level of fault tolerance. This is a challenging task as there are many rigid requirements to implement key management and the resources available to execute such methods are highly constrained due to which many highly secured approaches become infeasible to execute. A proper balance between requirements and the number of resources of WSNs decides which key management scheme should be used.

2 Related Work

In [2], authors outline a survey and also summarized critical issues related to different key management schemes highlighted by different researchers such as discovering compromised SNs in the network, making SNs tamperproof without communication overheads and minimizing the bootstrapping time required for WSN. The authors highlight that there is no one-size-fits-all approach to key management for all applications.

Shaik et al. [7] presented a detailed overview of different key management schemes along with the pros and cons of each scheme. The authors also published a table comparing each scheme based on different networks with varied parameters.

Ozdemir et al. [8] proposed a single network-wide key management approach in which a solo key is pre-loaded in the memory of all SNs. Each SN uses this solo key to encrypt and decrypt data, so there is no need to carry out additional key discovery and key exchange processes. All of the SNs send data using the same key that they already have. But this scheme has a major loophole that the compromise of any single SN will compromise the entire network.

For a network of n SNs, ($n - 1$) pair-wise keys are required to be stored in every SNs memory so that each SN can communicate with all the other SNs that are in its communication range. Since each SN has a unique pair of keys for communication with every other SN, this scheme is not scalable and the communication overheads of this scheme are very high as compared to a single network-wide approach. Authors in [9] proposed a key establishment scheme for WSNs that enables both inter- and intra-cluster communication. The goal is to reduce delay, storage and communication overheads while establishing pair-wise keys for these communications.

The major problem with the pair-wise key establishment scheme is that it requires each SN in the network to hold ($n - 1$) key pairs. A solution to overcome this problem is to use a centralized Key Distribution Centre (KDC) approach. The role of BS in this scheme is to supply session keys for communication between any two SNs. This key is saved in the SN memory and acts as an authentication entity for the SN. When compared to pair-wise key setup, this approach uses fewer keys, but the drawback of trusted BS is that it is not scalable and the BS is readily attacked.

In [10], the authors proposed a model in which the network is divided into zones, each with its own intrusion detection system (IDS) and KDC to detect the activity within its zone and communicate with its KDC. The authors tried to reduce the computation and communication overhead of the already overloaded BS by separating the IDS work of the BS with a separate entity in each zone.

In [11], the author proposed various random key pre-distribution methods. Two nodes can communicate with each other only if they have shared key. The drawback of this scheme is that the two neighbouring nodes cannot communicate with each other if they do not share common key and encryption keys of those nodes will be easily revealed which are captured by the attacker.

Erfani et al. [12] proposed a dynamic key management approach that included key pre-distribution and dynamic key establishment techniques. It ensures that any two SNs communicating can share a common key. Each SN memory stores the pre-distributed keys and dynamic keys independently. When a node in its radio range tries to interact with another node, it uses either the common pre-distributed key or the dynamic key. If the communicating SNs do not share a common key, they will compute a dynamic key for safe communication. This approach is more scalable and provides better resilience. This scheme, on the other hand, may not work well for high-mobility WSNs.

In all the above schemes, a lot of keys are required to be managed to broadcast the data packets securely to the BS over the network which increases complexity, communication overhead, energy consumption, time delay, etc.

3 Proposed Work

In some schemes, the keys are pre-loaded in the SNs prior to network deployment. Re-keying is necessary after a specified time interval so that it does not become stale. To establish security priorities such as integrity, confidentiality and authenticity between the connections established between arbitrary SN endpoints, the SNs require the existence of appropriate cryptographic keys at the end points. In a hierarchical scheme, the job of CH is to collect data from all cluster members, so keys are required to be shared between CH and each cluster member for communication. To transmit sensitive data aggregated by the CH to BS, another key is required which can only be shared between the CH and BS. But as SNs are resource constraints, the role of CH changes periodically, so every time new keys are required to be shared between the new CH and its cluster members and between the new CH and the BS which increases communication overhead. The unicast keys are required for SN-SN, SN-CH or CH-BS communication. The broadcast keys are required either by the CH to send messages to its cluster members or by the BS to broadcast a message to all SNs in the network. In other schemes, the keys are required to be refreshed due to changes in topology, the keys are updated periodically or on-demand, or the keys need to be refreshed after key revocation.

A large number of keys are required to be effectively and efficiently managed in different key management schemes for secure data transmission in WSNs which increases complexities, communication overhead, computation overhead and energy consumption. Instead of adopting different key management schemes that share many keys between SNs for secure data transmission, a KDS scheme is presented which eliminates the requirement for key management in the network.

This scheme works as follows: First, the network is partitioned into tracks and sectors. After then, the data packet D of each SN is divided into n overlapping segments such that D cannot be reconstructed from less than k segments. Three neighbours of each SN are selected in the direction of BS only. The segments are transmitted through the selected neighbours in such a way that no more than $(k - 1)$ segments of any SN are collected in a single place, except BS. After receiving k segments of SNs, the BS reconstructs the original packet D.

The flow diagram of proposed scheme is shown in Fig. 1. The model is divided into 4 stages:

- Network Organization
- Data Partitioning
- Neighbour Discovery
- Routing.

A. *Network Organization*: During this phase, the BS organizes the network by dividing the network into concentric circles known as tracks. Tracks are further divided into sectors. The clusters are the regions under the curved strip formed by the intersection of tracks and sectors. All the computations required for the construction should be done in the beginning by the BS.

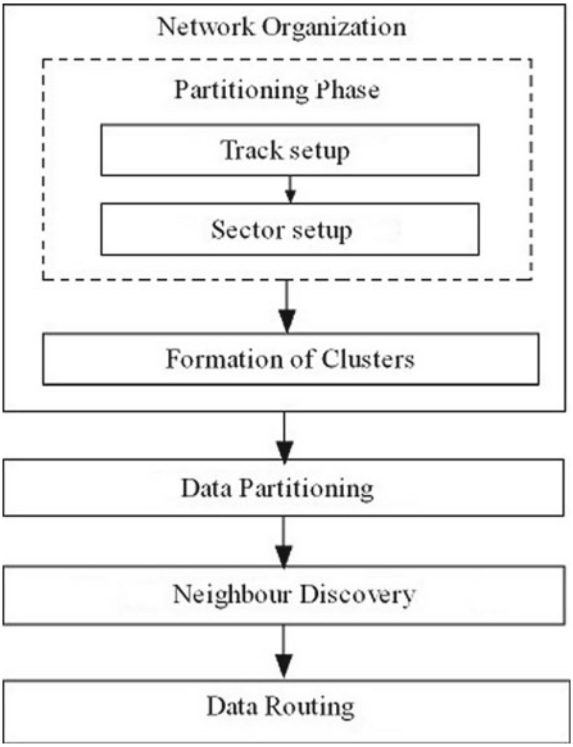

Fig. 1 KDS scheme

B. *Data Partitioning*: During data partitioning, data packet (D) of each SN is divided into n number of overlapping segments $S_1,\ldots,S_n$ such that:

- D can be easily reconstructed from any k segments where k is the threshold value $1 \le k \le n$, i.e. the minimum number of segments required to reconstruct D.
- Even comprehensive knowledge of $(k-1)$ or fewer segments of D provides no information about D and renders D absolutely unpredictable.

As shown in Fig. 2, the data packet D is divided into n number of overlapping segments which are further divided into three groups in such a way that each group contains less than $(k-1)$ number of segments. During data transmission, if any segment is lost, the BS can easily reconstruct D with the help of any k segments.

Algorithm 1 shows how D is partitioned into n number of segments S_1, S_2, S_3, $\ldots S_n$. First randomly choose $(k-1)$ integers. These integers are used to generate a polynomial of a degree $(k-1)$ as given in Eq. 1.

$$f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{k-1}x^{k-1} \qquad (1)$$

After then, the next step is to partition D into n segments using Eq. 2.

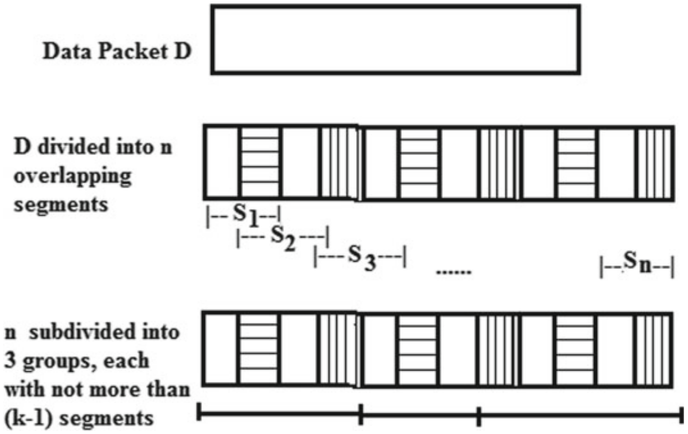
Data Packet D
D divided into n overlapping segments
|-- S1--|
|--- S2 ---|
|---S3---|
......
|--Sn--|
n subdivided into 3 groups, each with not more than (k-1) segments

Fig. 2 Data partitioning

$$S_i = f(x_i) \quad \text{where } i = 1 \ldots n \tag{2}$$

Equation 2 gives the desired number of segments of D.

Algorithm 1: Data Segmentation

Input: Data packet (D), total number of segments (n), number of segments required to reconstruct D is denoted with (k).
Output: n segments of D: $S = \{S_1,S_2,S_3,\ldots S_n\}$
Procedure: Data_Seg (D, *n*, *k*)

```
i    Input $k$ such that
       a)  $k$ should not be zero or less than zero.
       b)  $k$ should not be greater than $n$.
ii   Initialize $(k - 1)$ random integer numbers $(a_1,a_{2,\ldots}a_{k-1})$
iii  Set $a_0 = \text{D}$
iv   Generate a polynomial of degree $(k - 1)$.
         $f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{k-1}x^{k-1}$
v    for (each $x_i$, $1 \le i \le n$)
         {
             Compute $S_i = f(x_i)$
         }
vi   return $S = \{S_1,S_2,S_3,\ldots S_n\}$
```

These n segments $\{S_1, S_2, S_3, \ldots S_n\}$ are then partitioned into three groups comprising no more than $(k - 1)$ segments. These groups of segments are subsequently delivered to selected nodes to transmit data to the BS.

Algorithm 2 shows how D is reconstructed from any *k*-out-of-*n* segments by using Lagrange interpolation polynomial. The formula for the basis polynomials is defined in Eq. 3.

$$l_i(x) := \prod_{\substack{0 \le m \le k \\ m \ne i}} \frac{x - x_m}{x_i - x_m}$$

$$= \frac{x - x_0}{x_i - x_0} \cdots \frac{x - x_{i-1}}{x_i - x_{i-1}} \frac{x - x_{i+1}}{x_i - x_{i+1}} \cdots \frac{x - x_k}{x_i - x_k} \quad \text{where } 0 \le i \le k \tag{3}$$

For *k* segments, the interpolation polynomial in the Lagrange form is a linear combination of Lagrange basis polynomials which is given in Eq. 4.

$$L(x) := \sum_{i=0}^{k} y_i l_i(x) \tag{4}$$

Algorithm 2: Data Reconstruction

```
Input: Total segments ($n$), $k$ number of segments required to reconstruct D.
Output: D.
Procedure: Data_Rec ($n$, $k$)
i    Start
ii   if ($n < k$)
         insufficient segments of information for reconstruction
iii  Select any $k$ segments from $n$.
iv   Compute Lagrange interpolated polynomial using Eq. 4
v    D is the free coefficient after solving the Eq. 4.
vi   end
```

The data packet D of every SN is first divided into n segments for transmission and then BS reconstructs D from any k segments, as shown in Fig. 3.

C. *Neighbour Discovery*: As each SN has limited resources available so it is necessary to lower down transmission power and energy consumption to prolong network lifetime. Neighbour Discovery is the major component of communication. Instead of directly sending data to BS, each node finds its three neighbours in the direction of BS in such a way that.

- One neighbour is selected from the same sector in which the node resides.

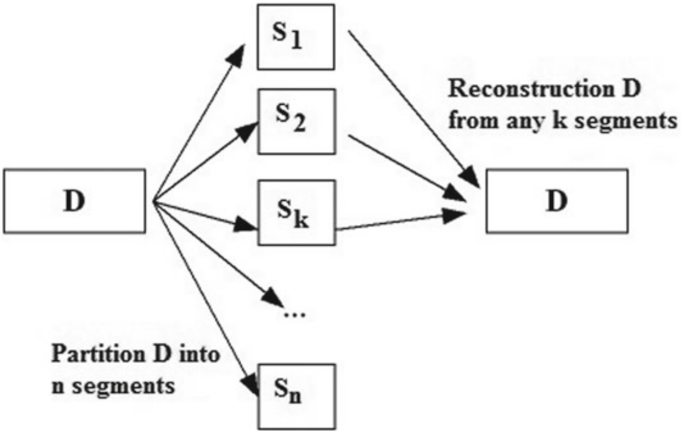
D
S_1
S_2
S_k
...
S_n
Partition D into n segments
Reconstruction D from any k segments
D

Fig. 3 Data partition and reconstruction

- The other two neighbours are selected from the right and the left adjacent sectors to transmit data to BS.

D. *Data Routing Phase*: It is used to transmit the data of each SN to the BS securely.

Track Model:

If the network is divided into tracks only and the SNs are randomly deployed. The information D of each node is partitioned into three groups which should not contain more than (k − 1) number of segments. Every SN in the network finds its three neighbours in the direction of BS to transmit segments of D. Each neighbour selected of a particular SN is allowed to send only (k − 1) or fewer segments of that SN only. When the SNs start transmitting the segments to BS through different routes with the help of selected neighbours, there is a high risk of collecting more than k number of segments of single SN at one place which make that SN compromised. So, in this case data of all the SNs is revealed except those SNs which are present nearest to BS. As the SNs instead of sending packets through neighbours, these nodes send data directly to BS. So to overcome this problem, the presented model partitions the network into circular tracks and triangular sectors.

For data transmission, the sensitive information D of each SN is divided into three groups such that:

- The neighbour selected from left adjacent sector is used to transmit (k − 1) or less segments of D to BS.
- Similarly, (k − 1) or less segments of D are transmitted through the neighbour selected from the right adjacent sector.
- And the remaining segments are transmitted via neighbour selected from the same sector in which the node resides.

During data transmission, the SN transmits its own information D with the help of neighbours selected from same sector and from left and right adjacent sectors, but restricted to send data received from other neighbouring nodes within same sector only. As a result, the information D of each SN is transmitted in such a way that no k or more than k number of segments of every SN can be collected on the single node. In this way, the original information D can never be reconstructed by any intermediate node. Only BS is allowed to receive any number of segments and can reconstruct D. Before attempting to reconstruct D, the BS must check that it has received enough segments (i.e. that it has received at least k segments), otherwise the retrieved information will be of no use.

Algorithm 3 shows the complete working of KDS scheme which defines how the data is securely transmitted over the network without any key requirement.

Algorithm 3: KDS

```
Input: Data packet (D), total number of segments ($n$), number of segments
    required to reconstruct D is denoted with ($k$).
Output: (i) $S = \{S_1,S_2,S_3,\ldots S_n\}$ (ii) Reconstructed D
Procedure: KDS (D, $k$, $n$)

i    Clusters are formed by dividing the network into tracks and sectors.
ii   Call Data_Seg (D, $n$, $k$) to divide D into $n$ segments.
iii  Find three neighbours of each SN in the direction of BS such that
       • First neighbour is selected from the same sector in which the node resides.
       • Second and third neighbours are selected from the next and previous sector
         respectively in which the node belongs.
iv   For every SN, neighbours selected from step 3 is used to transmit ($k - 1$) or
     fewer segments of D to BS.
v    Data_Rec ($n$, $k$)
vi   end
```

4 Simulation Results

- **Symmetric Key**: It relies on a shared key between two parties. The number of keys required to transmit data of p number of nodes is $p(p - 1)/2$. After encryption, the number of communication required to transmit encrypted data of each SN is p. So the number of communication required to transit encrypted data in this scheme is $p + (p\ (p - 1)/2)$.
- **KDS**: In this scheme, no key is required to encrypt or decrypt data. After data partitioning phase, the partition data is transmitted by three selected nearest neighbours. So to transmit data packet D of p number of SNs, $3p$ communications are required.

During key updation and data transfer, Fig. 4 compares the communication overhead of symmetric key distribution scheme with KDS. As KDS scheme eliminates the need for key management, there is no need to exchange keys between communicating nodes for data transmission. As a result, as compared to symmetric key distribution scheme, the number of communication overheads in this scheme is quite low.

The required simulation parameters are given in Table 1.

Figure 5 compares the number of uncompromised packets received by track model and KDS at BS after 200 rounds. In the KDS scheme, compromised nodes in the network are unable to compromise data, resulting in the BS successfully receiving uncompromised data packets, as opposed to other model in which data packets are more likely to be compromised before reaching the BS.

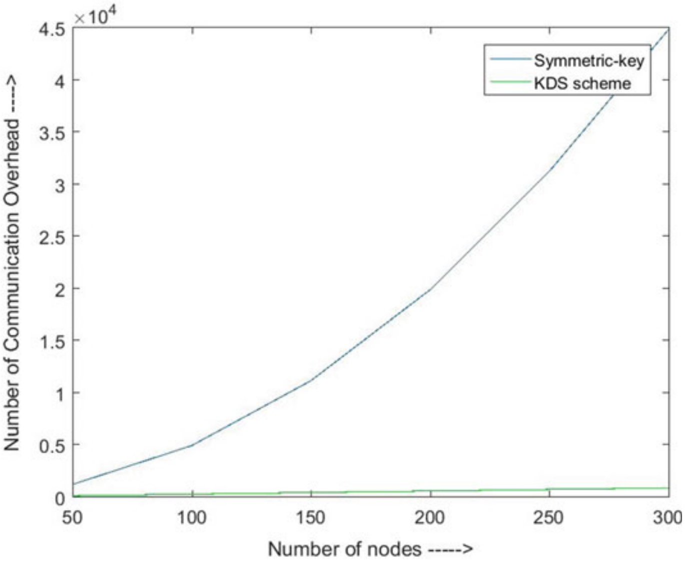

Fig. 4 Communication overhead comparison of symmetric key distribution scheme with KDS during key updation and data transmission

Table 1 Simulation parameters

Parameter name	Value
Network area	1000×1000 m^2
E_{elec}	50 × 10^(−9) Joules/bit
E_{amp}	100 × 10^(−12) Joules/bit/m^2
EDA	50 × 10^(× 9) Joules/bit
No of rounds	200

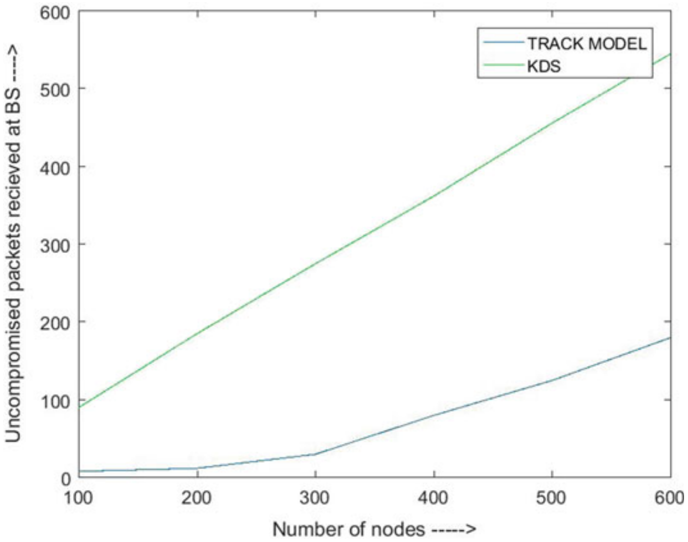

Fig. 5 Number of uncompromised packets received at BS comparison between track model and KDS

5 Conclusion and Future Scope

The proposed KDS scheme overcomes the problem of key management by reducing number of communication overheads in the network. As in this scheme, there is no chance that more than k segments of data packets collect at single place so compromised nodes in the network will not be able to compromise the data using KDS scheme. Comparison graphs show the proposed KDS scheme is secure and more efficient in prolonging network lifetime. The KDS scheme provides security only on static clusters. So in future, one can extend this work and develop a security framework for dynamic clusters where the size and composition of cluster members varied.

References

1. Luqman M, Faridi AR (2022) Security in wireless sensor network: a current look. In: 2022 9th International conference on computing for sustainable global development (INDIACom), Mar 2022. IEEE, pp 385–391
2. Xiao Y, Rayi VK, Sun B, Du X, Hu F, Galloway M (2007) A survey of key management schemes in wireless sensor networks. Comput Commun 30(11–12):2314–2341
3. Yousefpoor MS, Barati H (2019) Dynamic key management algorithms in wireless sensor networks: a survey. Comput Commun 134:52–69
4. Sharma C, Vaid R (2019) Energy-efficient and secure data forwarding mechanism for balancing cluster lifetime for huge size wireless sensor networks. J Comput Theor Nanosci 16(9):3961–3964
5. Bansal S, Juneja D, Mukherjee S (2011) An analysis of real time routing protocols for wireless sensor networks. Int J Eng Sci Technol (IJEST) 3(3):1797–1801
6. Sharma C, Vaid R (2019) Analysis of existing protocols in WSN based on key parameters. In: Proceedings of 2nd international conference on communication, computing and networking. Springer, Singapore, pp 165–171
7. Shaik R, Ahamad SS (2017) Key management schemes of wireless sensor networks—a survey. Fornteiras 6(2):526–537
8. Ozdemir S, Khalil Ö (2012) Performance evaluation of key management schemes in wireless sensor networks. Gazi Univ J Sci 25(2):465–476
9. Prema S, Pramod TC (2018) Key establishment scheme for intra and inter cluster communication in WSN. In: 2018 Second international conference on computing methodologies and communication (ICCMC), February 2018. IEEE, pp 942–944
10. Choudhary V, Taruna S (2016) Improved key distribution and management in wireless sensor network. J Wirel Commun 1(1):16–22
11. Mehta M, Huang D, Harn L (2005) RINK-RKP: A scheme for key predistribution and shared-key discovery in sensor networks. In: PCCC 2005, 24th IEEE International Performance, Computing, and Communications Conference, Apr 2005. IEEE, pp 193–197
12. Erfani SH, Javadi HH, Rahmani AM (2015) A dynamic key management scheme for dynamic wireless sensor networks. Secur Commun Netw 8(6):1040–1049