IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 3, Ver. V (May – Jun. 2015), PP 57-67
www.iosrjournals.org

KURCS: Key Updating for Removing & replacement of Compromised Sensor Nodes from Wireless Sensor Networks

Rohit Vaid[1], Vijay Kumar[2]
[1,2] *(CSE Department, M. M. Engineering College/ M. M. University, Mullana, Haryana, India-133207)*

Abstract : *An energy efficient key management scheme is an important aspect to ensure secure services in resource constrained Wireless Sensor Networks (WSNs). There are two parts of key management scheme, i.e. key distribution and key revocation. Key distribution is the task to manage the key in such a way that if two or more sensors want to communicate with each other for the purpose of sharing the data or control messages, the key is to be made available to them only. The scheme should be energy efficient, as the sensor nodes in the network are resource constrained. However, Key revocation is the task of removing the compromised sensors from the network to restrict such nodes to participate in further communications to avoid any kind of interference or disturbance in communication process. In this paper, we first review and summarize the current key revocation schemes. Then, we present an energy efficient key revocation scheme for removing compromised sensor nodes from the networks. Unlike most key removal schemes focus on removing the compromised node from the network for this purpose these schemes redistribute the keys or keying material to each and every node in the network. But the proposed scheme, Key Updating for Removing & replacement of Compromised Sensor nodes from Wireless Sensor Networks (KURCS), uses key updating techniques to update the keys of few sensors that are uncompromised not in the entire networks thus removes compromised sensor from the network. Simulation result proves that the proposed scheme is energy efficient and performs better than other comparable schemes in the literature without increasing the communication overheads.*

Keywords: *Clustering, Key Management, Key distribution, Key Revocation, Key updating, Sensor nodes, WSNs.*

I. Introduction

Wireless sensor network (WSN) consists of large number of battery-operated sensor nodes. These sensors are very small in size. They have also a built-in processor that is used for the computing functions. In case of wireless sensor network, communication among the sensors is done using wireless transceivers. So every sensor is equipped with a built in antenna that helps them in communication to other sensors in their limited communication range. Each sensor consist of four subsystems: Power Supply subsystem, sensing subsystems, processing sub systems and communication Subsystems. So with the help of these subsystems, sensors are able to sense the environment, compute simple tasks and exchange data among each other. But all the sensors are resource constrained in terms of memory, energy, processing power and communication bandwidth. Every subsystem uses energy for their working. Once the battery is drained, sensor nodes are useless. The situation of network disconnection is also arises if battery is drained in few of the nodes. So energy consumption by a node is a critical aspect, in order to increase the lifetime of the network. In most of the cases it is very difficult to recharge or replace the battery. Thus it is necessary that a protocol in WSN must be energy efficient. Sensor nodes are usually deployed in harsh or hostile environments such as battlefield, environmental monitoring or disaster area where they are operated without any attendance.

1.1. Types of compromised node detection and key revocation schemes

Compromised node detection scheme refers to preparing a list of ID's for those sensors that are performing malicious activity in the network and key revocation refers to the task of updating the key in such a way that the updated key is unavailable to the compromised sensor. There are two types of Compromised Node Detection and Key Revocation Schemes (CNDKRS) as given below:

i. Base station initiated CNDKRS
ii. Group initiated CNDKRS

1.1.1. Base Station Initiated CNDKRS

In Base station initiated Compromised Node Detection and Key Revocation Scheme, a centralized authority (Base Station) is responsible for the detection of compromised sensor node in the network and the same authority updates the key in such a way that the updated key is not available to the compromised sensors in the network. In this scheme, base station monitors the activities of each node in the network using following method:

A. Continuous Monitoring: In this scheme, every node monitors the activities of its neighbors and send this activity in the form monitoring report to the base station. If all the nodes in the network have at least j neighbors then at least j monitoring reports are transmitted by every node to the base station to detect the compromised node. Therefore this scheme is not scalable.
B. Periodic Monitoring: In this scheme, instead of transmitting the monitoring reports to the base station in every round, every node records the activities of each of its neighbor in every round. If some node A in the network is misbehaving to other node B then node B declares node A as a malicious node. In this way, every node monitors the activities of all of its neighbors. A node records the behavior of its neighbors directly by the experience or by the suggestions received from the common neighbors indirectly. On the basis of this direct experience and indirect suggestions, every node in the network gives a positive or negative rank to each of its neighbor. If the behavior and suggestions are good, then rank increases otherwise the rank decreases. If the rank of any sensor is below a threshold value, the node to be declared as a malicious node and its intimation in the form of a report is given to the base station at the same time.

Now using any one technique (continuous or periodic monitoring) the base station collects all the lists containing the ID's of malicious nodes from every sensor in the network. On the basis of these reports, base station declares the nodes as a compromised node based on some criteria applied to all the reports. Finally the base station prepares a list containing the ID's of compromised nodes and broadcast this list in the entire network to to aware all nodes about the ID's of uncompromised nodes. The key used by these compromised sensors is also updated and a new key is provided to all uncompromised sensors in the network. In this way, role of all compromised sensors is ignored in the network and are treated as they have been removed from the network.

If cryptography is used in the network then the list is also broadcasted in the encrypted form. If a single key is shared and used in the entire network, then the updated key is encrypted with this key and broadcasted in the network. Since the key used in the encryption process is already known to the compromised node, this compromised key is never used in the network to encrypt any confidential message. Network with single single shared key is not preferred as if key is compromised; entire network becomes unreliable and unsecure.

On the other hand, if each sensor is using a different key known to a particular sensor and the base station only, for example in a network of 'n' sensors the base station stores 'n' unique keys in its memory. Now if 'p' number of senors is compromised, the base station prepares (n-p) individual lists containing the ID of compromised sensors. Now the base station encrypts first list with the key of first uncompromised sensor and send this list to that sensor. Similarly second list is encrypted with the key of second uncompromised sensor and send this list to second uncompromised sensor & so on. Now all the sensors get the encrypted list containing the ID's of compromised sensors in the network. Every sensor decrypts the list with its own key and knows the IDs of compromised sensors in the network.

In this scheme, total n-p messages unicasted by the base station for list updation. The drawback of this individual key sharing is communication with base station only by all sensors; no communication among the sensor nodes. WSNs with this scheme seem to be efficient in applications based on clustering. For example clustering is used to achieve network performance but in clustering local decisions have to be made in order to cluster formation and cluster head selection where it is necessary that sensors from the local area should communicate with each other, without any interference from the base station.

1.1.2. Group Based CNDKRS

In this scheme, the decision of compromised node is made by several sensors in a fixed group locally without any dependency to the base station. But in this type of schemes, it is necessary for every sensor node to have two keys stored in its memory. One key is used to make a communication with the base station and other key is to communicate with the group members. It is also necessary that the key of one group is different from the key of other groups in the network. But the drawback of this scheme is that if some sensor in a group is compromised, the key of entire group is also compromised. If the system provides a new key to this group then this key is again available to the compromised sensor. On the other hand if we are encrypting the new key with the group key, its useless as the group key is already known to compromised sensor node in the group. One of the solution of this probleto create a new key for encryption by the base station with the individual keys of each sensor and transfers this encrypted key to every member in the group separately.

The advantage of this scheme is that the traffic that flow from sensor to base station decreases as there is no need to handover the monitoring report to the base station. In previous schemes every member of the network is transmitting his own report about each of its neighbor to the base station but in a group based scheme, only one report is given to the base station in a group of sensors. If there are 100 sensors in the network and each group contains ten percent of total sensors then ten such groups needs to formed. Each group contains

ten members. In this scheme instead of transmitting one hundred monitoring reports (one by every sensor) or 100*j (one report by every sensor about its each neighbor) to the base station, only one report per group is given to the base station. Therefore total ten reports needs to be transmitted in the network and given to the base station.

The drawback of this scheme is that network traffic from base station to sensor nodes is not decreasing because unicasting is the only solution to update the group key. Similarly this scheme performs better if compromised nodes are belonging to a particular area in the same group in the network, as unicasting is done only in that group to update the key. Rest of the traffic flow in the network (from base station to the unaffected area) is unaffected because there is no need to update the key in the uncompromised groups. If minimum one node is compromised in each group then base station needs to update the key of every group. Base station sends the updated key to every sensor in every group using unicasting method by encrypting the new updated key encrypted with the key of individual sensor to every sensor in every group in the entire network. In this scenario, this scheme is working in a similar way as a network with the single shared scheme.

The chances of key hacking are greater in this scheme because the same group key is to be used for long time unless any group member is compromised. In net shell, there is a need of a scheme to detect the compromised nodes and update the key for uncompromised nodes without affecting the performance of the network.

1.2. Key Revocation Scheme Requirements

A. Local Decision: The decision of compromised node detection is made locally by the group members instead by the base station.
B. Minimum interference from Base Station: The key updating process should update the group key locally with minimum communication from the base station.
C. Secure key distribution: An updated group key is distributed in such a way that it is unavailable to the compromised sensor in the group.
D. Periodically key updating: The group key is updated periodically irrespective compromised or compromised status of any node.
E. Minimum congestion: The updated group key is communicated to the base station so that traffic of unicasting to update the group key in the network is reduced.
F. Key chain secrecy: The principle of backward and forward secrecy needs to be maintained in the network.

II. Clustering

Clustering is the task of grouping wireless sensor nodes in a cluster. All the nodes in this group communicate with each other to send their sensed data to the cluster heads (CH). CH aggregate this collected data for onward transmission to the base station. Clustering is used for effective data communication. Clustering minimizes the energy consumption in the network by reducing long distance transmission with reduced number of nodes participating in long distance transmission.

There are two types of clustering scheme used in wireless sensor networks, i.e. static and dynamic. In static clustering scheme, clusters are fixed and there is no updation into the size and members of the cluster after cluster formation. But in dynamic clustering scheme, the members of every cluster are decided during cluster formation phase before every round. Thus, the size and members of every cluster are different and depends on the location of the cluster heads elected for the current round. Presented model uses the concept of static clustering, once the cluster is decided for any sensor, it is fixed and permanent in every round throughout the network lifetime. In presented model, the cluster for any sensor is decided with its physical location in the network.

The drawback of using dynamic clustering is that if one sensor is compromised in the network then cluster key of entire network should be updated as shown in Figure 5.1 because it is not known in advance that which sensor should join which cluster under which cluster head. There are several advantages of using static clustering. Some of them are given below:

2.1. Benefits of using Static Clustering

There are several advantages of using static clustering. Some of them are given below:

A. Advance knowledge of network: Cluster members are fixed and known in advance. Advance knowledge lets us know in advance about the affected area in case of any compromised node.
B. Reduce compromise node spoiling area: In static clustering, the members of any cluster are fixed and any node communicates only with the cluster members. If any node within a cluster is compromised, only the members of that cluster are affected not the entire network.
C. Reduce key updating area: If static clustering is used and once a node is compromised then the key need to be updated for that cluster rather updating for entire network as in case of dynamic clustering because

victims are not predicted in advance.

D. Fast key updating: If one sensor is compromised then key revocation scheme is applied only to the member of one cluster to which the compromised sensor belongs.

E. Reduce Network Traffic: As less number of keys are updated to implement key revocation scheme in the network so less number of communications are required in the network.

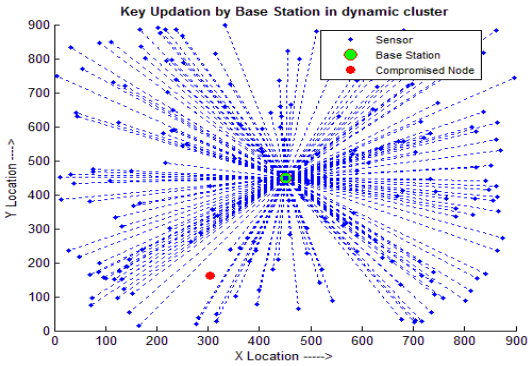

Fig. 1. Key updating by base station in dynamic clustering

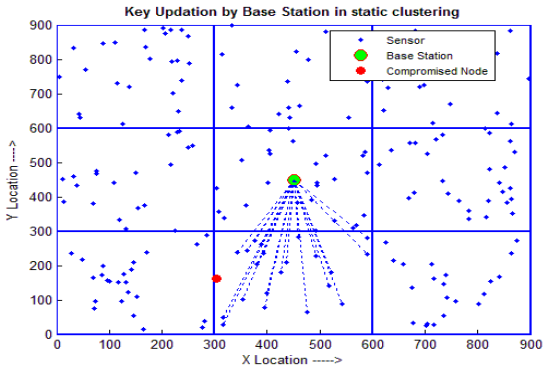

Fig. 2. Key updating by base station in static clustering

As shown in Fig. 2, if some sensor is compromised, there is no need to update the key of entire network. The BS should update the key of all the sensor that belongs to same cluster to which the compromised sensor is belongs to. But in our model, to reduce the long distance transmission between base station and uncompromised sensors, the concept of cluster guard should be introduces that is describe in the section of system model.

2.2. Key Sharing Schemes

Since a sensor node is either forwarding the aggregated data to the base station (Data Forwarding) by collecting it from all cluster members if it is a cluster head or injecting its own data to the cluster head (Data Injecting) if it is not a cluster head. The first scheme of key management is that in which a single key is shared and used in the entire network. Any sensor that is not a cluster head uses this key for data injection and any sensor that is elected as a cluster head uses the same key for Data forwarding process. The drawback of using a single shared key in the entire network is that if the key is compromised by any sensor in the network, entire network is compromised.

Second scheme is that in which a different key is used for all sensors in the network. This key is only known to a particular sensor and the base station only. According to this scheme, compromising a single sensor may not affect the functioning of entire network. But the drawback of this scheme is that in this scheme Data Injecting is completely impossible thus data aggregation scheme fails which is the key factor to achieve network performance. This scheme is only used in the networks where all sensors communicates only with the base station. There is no secure link or communication between the sensors in the network.

The third scheme is that a different key is used for different cluster in the network. All members of the same cluster use this shared cluster key for Data Collection and Data Injecting process. The drawback of using this scheme is that if any sensor in a cluster is compromised, entire cluster is also compromised. The result is that data injected by any cluster member in Data Injecting process is available to the compromised sensor and similarly the aggregated data that is used in Data Forwarding process is also available to the compromised sensor.

In fourth scheme of key management, two different keys are provided to all the clusters and keys of one cluster are different from the keys of other cluster in the network. From these two keys, first key (sensor key) is shared between an individual cluster member and the base station only and the second key (cluster key) is shared between all the cluster members and the base station only. The first key (cluster key) is used in the Data Injecting process and second key (sensor key) is used in the Data Forwarding Process (DFP). The advantage of this scheme is that if sensor key of any sensor within a cluster is compromised, it may not affect the sensor keys of other sensors in the same cluster. But the drawback of this scheme is that if cluster key of any sensor in a cluster is compromised, the cluster key of entire cluster is also compromised as this key is shared between all the members of the same cluster.

The main drawbacks of all these schemes are that in all of the specified schemes, the key or keys that is provided to the sensor nodes is/are fixed throughout the network lifetime and once this key is compromised, all the communications that are used by this node is also compromised. So the system requires a key management

scheme in which a key that is used in encryption or decryption process is updated after a fixed interval of time. The second drawback of these schemes is that in all the schemes the key is provided but not generated, i.e. the secret is distributed, which is not safe for the secret. If key is distributed, the chances of key compromise are more and more in these types of scheme. One more disadvantage of this type of scheme is that it increases the communication overhead due to keys redistribution to each and every node after a fixed amount of time. So a good key management scheme never distributes a key in unsecure nature of wireless medium, instead the key is generated with the help of a keying material that is distributed in the wireless channel. Compromising a keying material should not gives any clue about the key as the seed value that is stored in the memory of a sensor node is also used in key generation process along with the keying material.

III. System Model

We make the following assumptions about sensor network before presenting a general framework for removing and replacement of compromised sensor node from wireless sensor networks.

First, we assume that any scheme from [2, 3, 5, 6, 7, 8, 9, 10, 11] is applied to detect a compromised sensor node from the network because the detection of compromised sensor node is out of the scope of this paper. Once the scheme detects a compromised sensor node from wireless sensor networks, the presented system provides a new key to all the uncompromised sensors in such a way that the updated key is unavailable to the compromised sensors. So key of all those sensors that are affected by compromising the key or compromising the sensor itself should be updated.

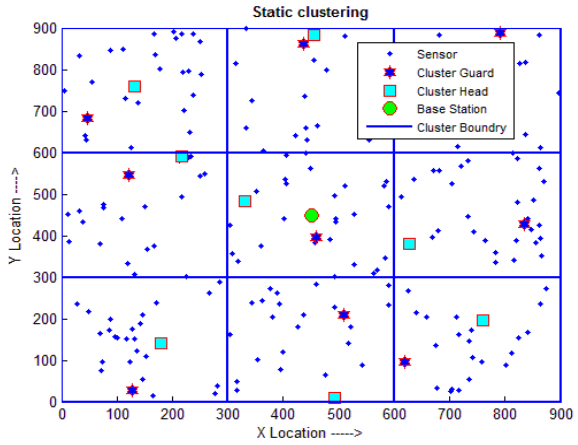

Fig. 3. Static clustering with guard nodes

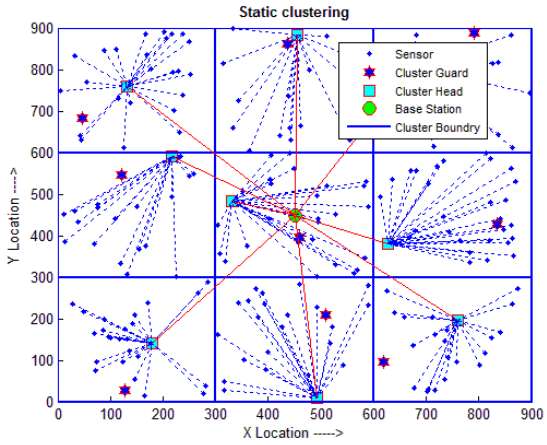

Fig. 4. Sensor to CH to BS Communications

In the proposed scheme, the network is divided into static clusters. Any scheme is chooses for clustering but once the members for a cluster are decided, the members are fixed and permanent throughout the network lifetime. One cluster guard sensor is provided to every cluster as shown in Fig 3. In every round, one cluster head is selected among all the cluster members on rotation basis to balance the energy consumption among all the cluster members. Any scheme from [2, 3, 5, 6, 7, 8, 9, 10, 11] is chosen for cluster head selection. All the members in a cluster sense the environment in the form of a event and send this event in the form of a digital information to the cluster head sensor in the encrypted form using cluster key. This cluster key is dynamic in nature, i.e. the key is updated in every round. Now the cluster head sensor compress the data by applying aggregation function after decrypting the data collected from all the cluster members. Now cluster head sensor transfer this compressed data to the base station as shown in Fig. 4 in the encrypted form with the help of a sensor key that is shared between cluster head and the base station.

Each sensor in the network shares a secrete key with base station known as sensor to base station key. The role of cluster guard is that once a sensor is compromised, it generates a new cluster key generation parameters (seed values) and unicast these seed values to every cluster members except the compromised sensor using cluster guard key that is shared between cluster guard and the individual cluster member as shown in Fig. 5.

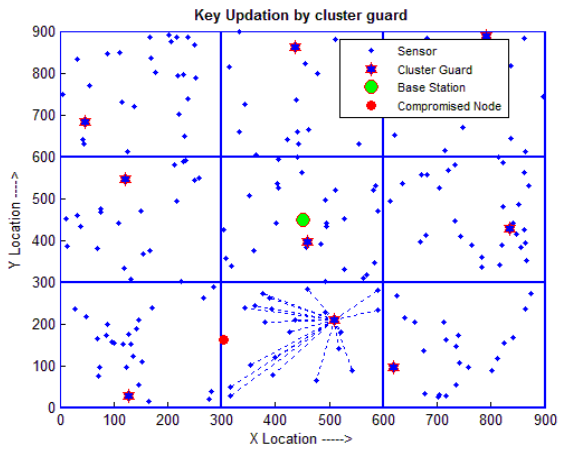

Fig. 5. Key updating by guard node in static clustering

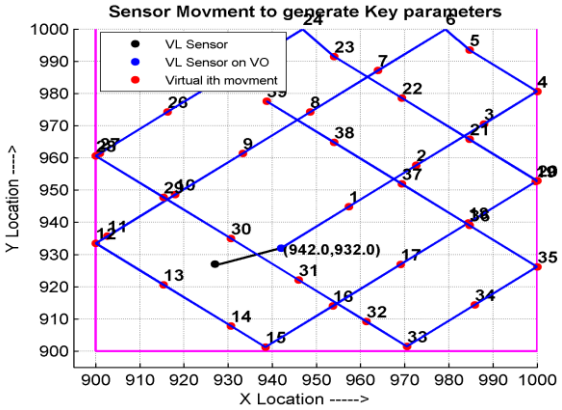

Fig. 6. Virtual movement after origin mapping

3.1. Key Management Scheme

Proposed key management scheme generates the dynamic key based on the concept of (VLKM: Virtual Location-Based Key Management Scheme for Wireless Sensor Networks) [1]. In this scheme, every cluster is given a virtual location. This virtual location is any random location within cluster boundary. This location is stored in the memory of every cluster member. The cluster is also provides a virtual origion. This virtual origin is also stored in the memory of every cluster member. All the cluster members map virtual location of cluster on virtual origin of the cluster. Cluster is also provides a virtual boundary. All the cluster members also store this virtual boundary in their memory. This virtual boundary is a square area where all the sensors move virtually with a specific virtual angle of the cluster in a particular direction with a virtual constant speed to update the virtual location of the cluster. When virtual location hits to the virtual boundary of the cluster by moving virtually, the direction of movement of virtual movement is changed accordingly, i.e. from Bottom Right (BR) to Right Top (RT) similarly from RT to Top Left (TL) and from TL to Left Bottom (LB) and from LB to BR. Whenever any cluster member moves, its current virtual location is updated accordingly. So all the cluster members within a cluster calculates clusters virtual location in every round by moving themselves to a new virtual location as shown in Fig. 6.

3.1.1. Types of keys used in the system

Due to the proper working of the network, several types of keys are used in the network. There are three types of scenario in which sensor communicate in the system. In first scenario, the sensor is communicating with the base station, in second scenario the sensor is communicating with the cluster guard and in last scenario, the sensor is communicating to a cluster head that is different in each round. So a different key is used in different scenario. The keys include:

A. Sensor-BS key: This key is shared and known only to a particular sensor and the base station only. Base station uses this key to make a communication with a particular sensor in the network. Base station knows the key of every sensor in the entire network but an individual sensor knows only a single key that is shared with base station only. All the sensors use this key to make a communication with the base station.
B. Cluster key: This key is used and shared within cluster members of the same cluster. This key is dynamic in nature and is updated after every round in each cluster. The updating of this key is managed in such a way that minimum communication required to update the cluster key. If some malicious sensor has come to know the key in current round then the key that is used in next round is hidden from this malicious sensor so that minimum data generated in the network is compromised.
C. Sensor-guard key: This key is shared and known only to a particular sensor within a cluster and a guard node only. Cluster guard uses this key to make a communication with a particular cluster member only. Guard node knows the key of every sensor in its cluster but a cluster member knows a single key that is shared with the guard node only. Cluster member uses this key to make a communication with the cluster guard of his cluster.

3.2. Key Generation

There are three types of keys that are used in the system. The keys include sensor-BS key, cluster key and sensor-guard key. From these three keys, two keys (sensor-BS key and cluster key) are frequently in the system whereas the sensor-guard key is only used by all the sensors of a particular cluster whenever any sensor

in that cluster is compromised. So both sensor-BS key and cluster key are generated in the network whereas the sensor-guard key is fixed. The process to generate sensor-BS key and cluster key is given below:

3.2.1. Cluster Key Generation

All the sensors within a cluster share a common key known as cluster key. This key is generated by applying one way hash function on current virtual location (CVL: CVX, CVY) and latest initial virtual location (IVL: IVX, IVY) of the cluster given by cluster guard without origin mapping. This way all the cluster members updates their cluster key by updating their current virtual location to produce a same cluster key for all the cluster members without any communication from the base station or the cluster guard. For more details follow [1] for efficient group key management scheme. Cluster guard provides key generation parameters to all the cluster members as shown in Eq. (1).

$$CG_i \rightarrow S_{i,j} : E_{SGK_j}(CVL, CVB, CVM, CVA) \quad \ldots \quad (1)$$

Where CG_i is the cluster guard of ith cluster, S_{ij} is the jth cluster member within ith cluster, ESGK is the encryption function with the help of a sensor to guard key of jth sensor within cluster i, CVL is the cluster virtual location of ith cluster, CVB is the cluster virtual boundary of ith clutter, CVM is the cluster virtual movement of ith cluster, CVA is the cluster virtual angle of ith cluster.

All the cluster members calculates their current virtual location by moving virtually at virtual distance within virtual boundary at virtual angle given by the cluster guard. (By using Algorithm1 Compute Current Virtual Location given in [1] as shown in Eq. (2).

CVDL (VL_{Cr-1}; VB; VA; VS; VD) … (2)

Where VL_{Cr-1} is Virtual Location in previous round (in first round it is IVL), VB is Virtual Boundary, VA Virtual angle of movement in degree (VA), VS is Virtual speed of movement and VD is current virtual direction of movement (in first round it is BL);

Now all the cluster members generates cluster key by applying one way hash function (by using Algorithm2 Compute Dynamic Key for current round with the help of a folding addition) given in [1] as shown in Eq. (3).

ComputeDynamicKey (IVX, IVY, CVX, CVY) … (3)

Where IVX is the latest initial virtual X location without origin mapping, IVY is the latest initial virtual Y location without origin mapping, CVX is the current virtual location on virtual origin and CVY is the current virtual Y location on virtual origin.

3.2.2. Sensor-BS Key Generation

All the sensors within a network share a unique key with the base station known as sensor-BS key. This key is generated by applying one way hash function on sensors initial virtual location without origin mapping and current virtual location after virtual origin mapping. Current virtual location is calculated in the same way as cluster current virtual location is calculated but the procedure is applied on sensor virtual location instead of applying on clusters virtual location. Sensor key is also calculated in the same way as the cluster key is calculated but this time, the procedure is applied onto sensors current virtual location instead of applying it on cluster current virtual location in case of calculating the cluster key.

3.3. Key Updation

To detect a compromised sensor, any scheme from [2, 3, 4, 5, 6, 7, 8, 9, 10, 11] can be used. Detection of compromised sensor node is out of the scope of this paper. The concept of sensor to cluster head and cluster head to base station (S-CH-BS) is used in the network, i.e. all the cluster members sense data from the environment and forward this data to the cluster head using cluster key that is known to all the cluster members. Now the cluster head forward this data collected from all the cluster members to the base station using its sensor key that is known only to this sensor and the base station only. The concept of one key one time (OKOT) is applied on cluster and sensor keys. According to the principle of OKOT, once a key that is used in the system, the key is discarded and this discarded key is never used again in the system in any subsequent rounds. Any time a key is used, the key is updated. As the cluster key is used in every round because all the cluster members elect a different cluster head in each round on rotation basis to balance the energy consumption. Now this newly elected cluster head collects cluster data from all the cluster members with the cluster key that is known to all

the cluster members. That's why the cluster key is updated in every round. But the sensor key is used by a particular sensor whenever this sensor is elected as a cluster head. The sensor uses this key to transmit the cluster data after aggregation in encrypted form to the base station to achieve aggregation secrecy. So the cluster key is updated in every round as this key is used in every round whereas the sensor key is updated by a particular sensor whenever it usess its sensor key, i.e. when a sensor is elected as a cluster head.

Sensor key is only used by a particular sensor. If this key is compromised, there is no loss as this sensor is using a different key in the next time when this sensor is elected as a cluster head. Because every time sensor uses its sensor key, the sensor updates its current virtual location by moving virtually with a virtual angle and virtual speed of movement within virtual sensor boundary to update its current virtual location. Now the new key is produced by applying a one way hash function to update the sensor key. The old key that is compromised is discarded automatically.

If the cluster key is compromised, there is no effect on the functioning of the network as this key is automatically updated in every round to achieve the principle of OKOT. But on the other hand, if a cluster member is compromised then its sensor key and cluster key both are also compromised. Sensor key is used only by this sensor so it may not harm the rest of the system components. The only attack that is possible in this case is that it injects false data within the system. The solution of this problem is that all the uncompromised sensors stops communication with this compromised sensor. So a list is maintained in the cluster to maintain the IDs of compromised sensors.

But as the cluster key that is used by this sensor is also compromised which is also used by all the other uncompromised cluster members. In this case, the cluster key updating procedure may not helps as the key generation parameters (cluster initial virtual location, cluster virtual angle of movement, cluster virtual distance of movement and virtual boundary) and the key generation procedure is also known to this compromised sensor. So this time the key is updated in such a way that this updated key is not available to any of the compromised sensor in the cluster. The solution of this problem is that the cluster key is updated by updating any one or all of the cluster key generation seed parameters, i.e. cluster initial virtual location, cluster virtual origin, cluster virtual boundary, cluster virtual distance, cluster virtual angle that is used in one way key generation hash function. To update the cluster key, the cluster guard of that cluster unicasts a new cluster key generation seed values to all the uncompromised cluster members in encrypted form with the individual sensor guard key. All the uncompromised sensors update their cluster key by using VLKM scheme except the compromised sensors. Now the new cluster key is available to all the uncompromised sensors only. Cluster guard also sends a list that contains the IDs of compromised sensors in the cluster to all the uncompromised cluster members to avoid the false data injection attack in the network. The format of the message given by the cluster guard sensor to the cluster members is given in the following message:

$$CG_i \rightarrow S_{i,j}:E_{SGK_j}(CVL,CVB,CVM,CVA,R,TS_t) \quad \ldots \quad (4)$$

Where CG_i is the cluster guard of ith cluster, S_{ij} is the j^{th} cluster member within i^{th} cluster, E_{SGK} is the encryption function with the help of a sensor to guard key of j^{th} sensor within the cluster i, CVL is the cluster virtual location, CVB is the cluster virtual boundary, CVM is the cluster virtual movement, CVA is the cluster virtual angle of movement, R is the list that contains the ID of compromised cluster members and TS_t is the time stamp when cluster initial virtual location is updated. Now all the cluster members generates a new current virtual location and updates cluster key that is available to only to all the uncompromised sensors.

IV. Performance analysis

To evaluate the performance of a scheme some performance measurements are necessary. For evaluating the performance of KURCS scheme, following measurements is considered:

4.1. Key Updating cost

Key updating cost is measured in terms of energy, i.e. the amount of energy that is consumed in updating the compromised key in the network. Two types of environment are considered in the network as given below:

4.1.1. Compromised nodes spreading from particular area sequentially in the network

In this scenario, a compromised node is assumed to be detected in the first cluster. The second node is again compromised in the same cluster. Then third fourth and so on all nodes are compromised one by one from the first cluster. When all the nodes of first cluster are compromised, then second cluster is chosen as a victim, i.e. one by one all nodes of second cluster are compromised. In this way compromised nodes are spreading from part of the network to the entire network.

Simulation result in Fig. 7 proves that if dynamic clustering is used then it is not known in advance that which node is elected as a cluster head and which nodes join their cluster. That's why key updating process

updates key for all those sensors that are uncompromised in the entire network. The scheme is represented as DBS (Dynamic cluster and key updating by Base Station. But on the other hand if static clustering is used, then there is no need to update the key in entire networks because a sensor is bounded to communicate with other sensor that is the part of the same cluster. So instead of updating the key in entire network, the key in particular cluster is updated to which the sensor is compromised. SBS (Static cluster and key updating by Base Station) represent the scheme when key is updated by the base station and SCG (Static cluster and key updating by Cluster Guard) represent presented KURCS scheme when the key is updated by the cluster guard instead of base station. Simulation result shows that whenever entire cluster is compromised, i.e. all the cluster members are compromised then no energy is consumed in both SBS and SCG schemes as there is no need to update the key of any sensor in that cluster. Simulation results prove that presented SCG scheme is efficient in terms of energy from SBS and DBS schemes.

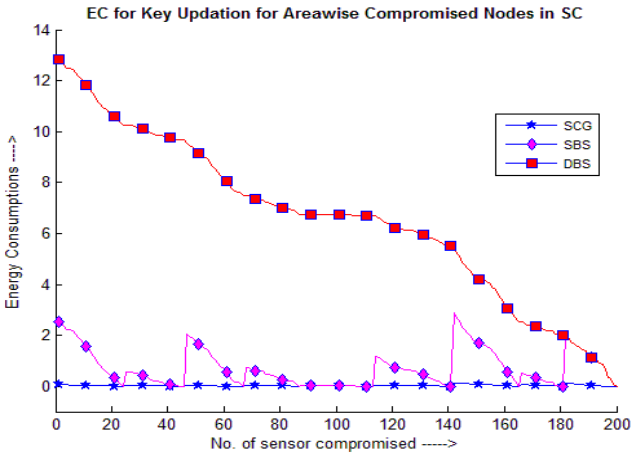

Fig. 7. Energy consumption in static clustering with compromised node from area wise spreading in the entire network

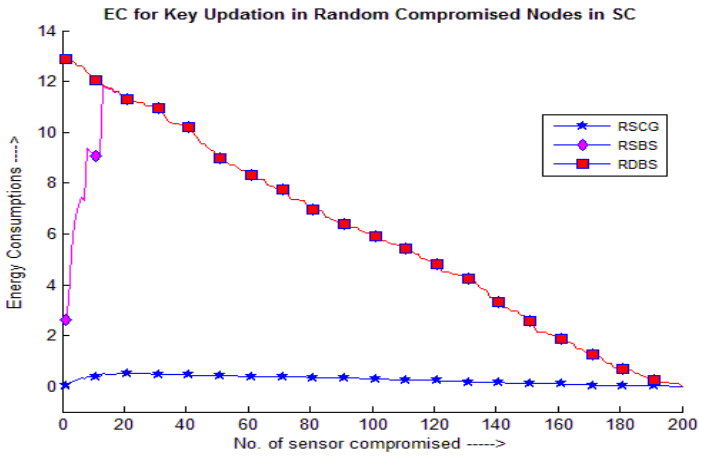

Fig. 8. Energy Consumption in Static Clustering with Random compromised node from Entire Network

4.1.2. Compromised nodes spreading randomly in the network

In this scenario, a compromised node is assumed to be detected at any part of the network, i.e. any node in the network is assumed to be compromised. All nodes in the network are compromised one by one till all the nodes in the network are compromised. In this way compromised nodes are spreading randomly one by one in the entire network.

In Fig. 8, RDBS (Dynamic clustering and key updating by Base Station) scheme is shown when sensors are compromising randomly in the network. But on the other hand when static clustering is used and the key is updated by base station only in the cluster in which at least one sensor is compromised is represented with RSBS. Similarly, when the key is updated by cluster guard in that cluster in which at least one sensor is compromised is represented with RSCG that is the presented KURCS scheme. Simulation result proves that RSBS schemes perform in same way as RDBS scheme when at least one sensor is compromised in all the

clusters in RSBS scheme. But overall the performance of RSCG scheme is efficient in terms of energy from both the other schemes.

4.2. Communication overheads

Communication overhead is measured as the number of control messages that are transferred in the network to update the compromised key after detecting the compromised sensor in the network. The performance of the KURCS scheme depends mainly on cluster key updating process. The cluster key can be updated when a sensor is compromised. This key is updated by unicasting, i.e. cluster guard of the cluster from which the node is compromised unicast key generation parameters to all the cluster members that are not compromised.

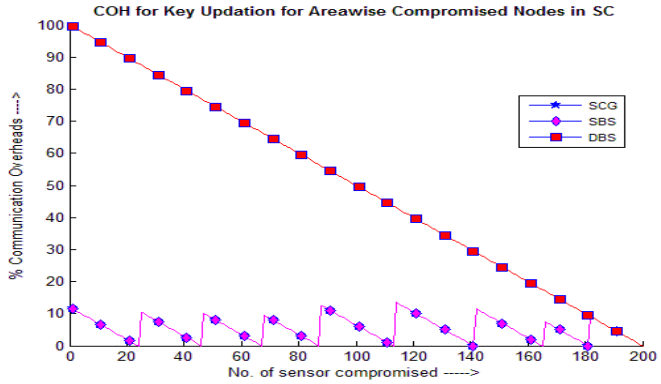

Fig. 9. Communication Overheads in Static Clustering with compromised node from area wise spreading in the entire network

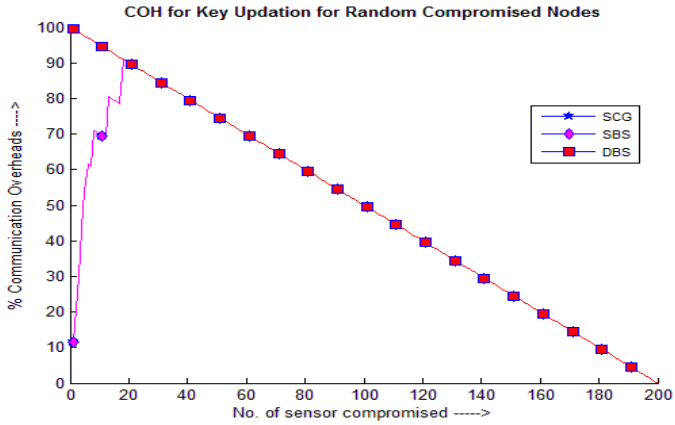

Fig. 10. Communication Overheads with Random compromised node from Entire Network

Simulation result in Fig. 9 shows DBS (Dynamic clustering and key updating by Base Station) scheme. In this scheme, key updating process updates keys for all those sensors that are uncompromised in the entire network by the base station. If static clustering is used and the key is updated by the base station in particular cluster to which the sensor is compromised is represented as SBS. Similarly SCG represent the presented KURCS scheme when key is updated by the cluster guard in particular cluster to which the sensor is compromised. Simulation result shows that both SCG and SBS perform in same way in terms of communication overheads. Communication overheads are 0% when the entire cluster is compromised in both SCG and SBS schemes but in DBS scheme, communication overheads reach to 0% whenever the network is dead, i.e. all nodes in the entire network are compromised. Simulation result proves that both SCG and SBS schemes perform better than DBS scheme in terms of communication overheads.

In Fig. 10, dynamic clustering is represented as DBS when key is updated by base station in the entire network whenever a ransom sensor is compromised in the network. But on the other hand when static clustering is used and the key is updated by base station only in the cluster in which at least one sensor is compromised is represented with RSBS. Similarly, when the key is updated by cluster guard in that cluster in which at least one sensor is compromised is represented with RSCG that is the presented KURCS scheme. Simulation result shows that both SCG and SBS schemes perform in same way in terms of communication overheads. All of the schemes

perform in same way when at least one sensor is compromised in all the clusters. But till then the performance of SCG and SBS schemes are efficient then DBS scheme in terms of communication overheads.

V. Conclusions

In this paper, we presented a key updating scheme for removing & replacement of compromised sensor (KURCS) node from Wireless Sensor Networks. This scheme significantly reduces the number of transmissions needed for removing a compromised sensor node from the network. The scheme also updates the keys of all those sensors which are not compromised but using the key that is stored in the memory of compromised sensor in such a way that the updated key is unavailable to the compromised sensor. This scheme is very suitable for removing & replacement of compromised sensor WSNs. Simulation results proves that this scheme performs better in terms of energy efficiency without increasing the communication overheads.

References

[1]. Rohit Vaid, Vijay Kumar, "VLKM: Virtual Location-Based Key Management Scheme for Wireless Sensor Networks" in IEEE International Conference on Parallel, Distributed and Grid Computing (PDGC-14) held at Jaypee University of Information Technology, Waknaghat, Shimla, H.P. , INDIA, pp. 53-61, December 11 – 13 , 2014.

[2]. Ishmanov, Farruh, Sung Won Kim, and Seung Yeob Nam. "A Secure Trust Establishment Scheme for Wireless Sensor Networks." Sensors 14, no. 1 (2014): 1877-1897.

[3]. Renjian Feng, Xiaofeng Xu, Xiang Zhou and Jiangwen Wan, "A Trust Evaluation Algorithm for Wireless Sensor Networks Based on Node Behaviors and D-S Evidence Theory", open access, Sensors 2011, 11, 1345-1360, ISSN 1424-8220, www.mdpi.com/journal/sensors.

[4]. Yong Wang and Byrav Ramamurthy, Xukai Zou and Yuyan Xue, "An Efficient Scheme for Removing Compromised Sensor Nodes from Wireless Sensor Networks" (2007), CSE Technical reports, Computer Science and Engineering, Department of University of Nebraska – Lincoln, Paper 77. http://digitalcommons.unl.edu/csetechreports/77, 9-7-2007.

[5]. Satya Keerthi, A. Manogna, Yasaswini, A. Aparna and S. Ravi Teja, "Behaviour based Trust Management using geometric mean approach for Wireless Sensor Networks", International Journal of Computer Trends and Technology- volume3, Issue2- 2012.

[6]. Joengmin Hwang, Tian He, Yongdae Kim, "Detecting Phantom Nodes in Wireless Sensor Networks", IEEE INFOCOM 2007.

[7]. Ke Liu, Nael Abu-Ghazaleh, Kyoung-Don Kang, "Location verification and trust management for resilient geographic routing", Journal of parallel and distributed computing, 67 (2007) 215 – 228.

[8]. Yanchao Zhang, Wei Liu, Wenjing Lou and Yuguang Fang, "Location-Based Compromise-Tolerant Security Mechanisms for Wireless Sensor Networks", IEEE journal on selected areas in communications, Vol. 24, No. 2, February 2006.

[9]. Chan-O Hong and Yoon-Hwa Choi, "Proximity-Based Robust Event Detection in Wireless Sensor Networks, International Journal of Distributed Sensor Networks Volume 2014, Article ID 632397, 7 pages, http://dx.doi.org/10.1155/2014/632397.

[10]. Hongjuan Li, Keqiu Li, Wenyu Qu and Ivan Stojmenovic, "Secure and Energy-Efficient Data Aggregation with Malicious Aggregator Identification in Wireless Sensor Networks".

[11]. Shaik Sahil Babu1, Arnab Raha, Mrinal Kanti Naskar, "Trust Evaluation Based on Node's Characteristics and Neighbouring Node's Recommendations for WSN", Wireless Sensor Network, 6, 157-172, 2014.