KEY MANAGEMENT TECHNIQUES IN WIRELESS SENSOR NETWORKS

Khawla Naji Shnaikat[1] and Ayman Ahmed AlQudah [2]

[1]Department of Computer Science, University of Jordan, Amman, Jordan
[2]Deanship of e-learning, University of Dammam, Ad Dammam, KSA

ABSTRACT

The way that is used to achieve most important security requirements is the cryptography. Cryptography mainly depends on what is called cryptography keys; the cryptographic keys required to be managed by using a robustness technique that guarantees the needed security requirements. So first of all, the necessary keys need to be distributed to the nodes before they are disseminated in the target area, and then let the sensors that need to communicate establish its secure communication by having a deal on what is called pair-wise key, and allow the refreshment process for those keys to be occurred successfully when it is needed, and finally, having the ability to revoke the keys that related to compromised nodes. These phases are performing the process that is called Key Management. ***In this paper we will explain different key management schemes, critique them theoretically, and propose an idea as a way out for the expected problems in one of these schemes.***

KEYWORDS

Wireless Sensor Networks, Key Management, and Cryptography.

1. INTRODUCTION

Wireless Sensor Network (WSN) is the network that consist of many small devices that called sensors. In literature, sometimes it is considered as a special type of the ad hoc networks [1]. These networks are useful in our life; they are widely used in many applications. WSN is used in military, commercial, and in ecological. Thus, the communications in these networks must be secure.Securing the communication in WSN is a very important issue, just because of many security threats and because of the nature of WSN. We can notice this point by studying the communication links in these networks, which is the radio links that is subject to many faulty information and malicious attacks. Sensor devices in general have a limitation in its resources; these limitations can control the nature for WSNs, also affect the security level for this type of networks. As an example for such limitations; limited processing power, battery age, transmission distance, shortage in memory space, random distribution for nodes, and bandwidth [2].

- Limited processing power: Sensor node as any other technical device has its own hardware and software architecture [3]. The key factor when manufacturing a sensor node is to keep the hardware and software architecture for the sensor node simple as much as possible to save node energy [3].
- Battery age: The sensor nodes are fully dependent on its battery, since it is operating in outdoor environment [4]. The only energy resource for sensor nodes is its battery, so once it is out the sensor node is considered as dead node [4].
- Transmission distance: The transmission distance or range for wireless devices is limited. The more distance the sensor node can send, the more energy will be

DOI : 10.5121/ijnsa.2014.6605

consumed [2]. Thus, the transmission range for sensor nodes is restricted regarding to the limitation in sensor resources.

- Shortage in memory space: Sensor nodes just as other related devices have its own operating system software. This software is loaded to the sensor memory and takes part of the memory space. Thus, rest of the memory space needs to be managed in an efficient way [2]. The keys that are needed in the cryptography process in WSN needs a light management mechanism that do not waste the residual memory space, since there is another data in WSN that needs this space, while the network is operating[2].
- Prior deployment knowledge: In general, sensor nodes are deployed in risky areas. For example, war areas. Thus, the deployment process for sensor nodes is a critical issue most time and it could be done randomly or dynamically [2]. This issue contributes in making the location information for the sensor nodes places insufficient or most likely not available. The key management techniques should not depend on the sensor location specially when starting keying process [2].
- Bandwidth: The key establishment process in WSN is limited to sensor nodes constraints. As mentioned before, the hardware architecture for sensor nodes is simple. The transmitter in such nodes has limited capabilities, so it cannot transmit a large set of data at the same time. This means that the bandwidth capacity for sensor nodes is small. Therefore, the key management techniques should take in its account this issue [2].

Since the WSN has such constraints, and is vulnerable to attackers, securing it is considered a great challenge. Many algorithms have been proposed in the literature to achieve this task. WSN needs a cryptography algorithm that must be selected carefully, and the most important factor for those algorithms is solving the Key agreement or management problem, that is needed to provide an encrypted and authenticated data transmission between the sensors nodes to have a secure channel. The cryptography algorithm has to meet many securities requirements such as, confidentiality, integrity, authenticity, and availability (CIAA) [5].

- Confidentiality: This is one of the most important security requirements. It means that the secrecy of the exchanged data between nodes must be guaranteed, and could not be detected by any illicit sides. Therefore, the key establishment mechanism is considered best if it has the ability to guarantee the protection for the keys, so the exchanged data will be secured [2].
- Integrity: This security requirement means that any type of modification for the data must be prevented. The way to guarantee this issue in WSN is to use key management techniques that forbid any illicit node from making any changes on the keys [2].
- Availability: In general, availability means to have the service all the time. In WSN there are many factors that cause a denial of service. The constraints that are found in sensor resources have a great effect on the availability requirement. For example, as mentioned before, the battery life for sensor nodes is short; this is cause an interruption in the service. So the being used key establishment technique should take into its account how to maintain a lasting service in the WSN.
- Authenticity: In security, the authentication means that the node needs to proof its identity. The key management mechanism should not allow for any strange node to participate in any communication process that takes place in the WSN.

With the existence of all previous difficulties that related to the sensor devices nature, the key establishment techniques need to be efficient in a way that can handle such issues. To judge the efficiency and goodness for the key establishment mechanism, many metrics that was mentioned

in the literature, could be used to assess the technique. Here we will brief some of them to see if the key management technique could overcome the obstacles that mentioned before. Theses metrics are resistance, resilience, revocation, the scalability [2].

- Resistance: As we said previously, the sensor nodes are distributed in dangerous and far target areas which make it hard to monitor it all the time. This makes the sensor nodes subject to be physically seized by the attackers. The attacker or adversary once catch the sensor node will try to get any information that could help in revealing the secret keys and data in the WSN. In addition, the attacker may make many changes to compromise some nodes in the network, and place these nodes in the WSN, by this the attacker roles the WSN and the confidentiality for WSN is penetrated [2][4].
- Resilience: If the adversary success in capturing some sensor nodes, the key establishment mechanism should prevent him from making any use of that. This means that the technique should guarantee the robustness for the sensor nodes against the physical attack. The attacker must not success in his trials to get the secret keys and penetrating the confidentiality for the WSN [2].
- Revocation: This comes to get the nodes that were attacked and pervaded by the adversary back. The key establishment mechanism should guarantee this issue in WSN, and should maintain it with respect to the limited resources for sensor devices [2].

The sequence goes as follow; the adversary captures the sensor node and tries to reveal the secret keys to reveal the data. The key management technique could not prevent the nodes from being physically captured, but it should guarantee the resilience. If the key establishment mechanism did not guarantee the resilience, then it should guarantee the revocation for the seized nodes. As a result, through providing and maintaining the resistance, resilience, and revocation. The key establishment technique is considered and evaluated to be efficient.

- Scalability: In general, scalability means that the solution for small size problem should work properly if the problem size gets larger. WSN is subject to be larger while it operates, i.e. new nodes could be inserted to the network. The key management technique should be scalable and allow for the new nodes to work in a secure manner fairly [2].
- Adjustability: In the context of key management, being adjustable means that the mechanism should work properly if any network conditions change. For example, having dead nodes in the networking or losing connectivity.

The previous introduction leads us to the importance of having an efficient key management technique. This technique approach sets a key between whole sensor nodes that aim to exchange data in secure manner, handle the join and disjoin of nodes, the ability to work in undefined deployment environment, and could prevent the unauthorized nodes from establish communication with the network nodes. And it needs to success in fulfilling those goals under WSN constraints, which means considering simplicity in working to not consuming the sensors resources.

2. PHASES FOR THE KEY MANAGEMENT PROCESS

Various schemes have been proposed in the literature for key management techniques, these schemes have focused on many phases that are needed for this process to secure the WSN and to overcome the aforementioned obstacles in WSN. We illustrate here three key management schemes and explain the most important phases for each one of them. The three management techniques that we analysing are:

2.1. The First Technique that mentioned in [6]

As the name implies, this technique is designed for the Heterogeneous Sensor Networks (HSN) that is formed of many clusters. Each cluster is composed of one highly equipped sensor node that is called the cluster head or sink [6], and a number of less equipped sensor nodes, which are the typical sensor nodes. This key management scheme is having the following phases:

2.1.1. Pre-distribution Phase

This step is happened before the deployment of the sensors, there are many mechanisms to do this step, such as, the Pair-wise Key Pre-distribution, the Master Key Based Pre-distribution, the Base Station Participation, and the Probabilistic Key Pre-distribution. The Base Station (BS) mechanism is used in the Key-chain approach. So that, some calculations need to take place prior to the nodes deployment process. These calculations start with the generation of two key chains. These chains generation is done by the Base Station (BS) using the two one-way functions F_1 and F_2:

$\{n_{10}k_{10}, n_{11}k_{11}, n_{12}k_{12}, n_{13}k_{13}..., n_{1n}k_{1n}\}$
$\{n_{20}k_{20}, n_{21}k_{21}, n_{22}k_{22}, n_{23}k_{23...}, n_{2n}k_{2n}\}$

Where $k_{1\,(n+1)} = F_1\,(k_{1n})$, $k_{2\,(n+1)} = F_2\,(k_{2n})$; n_{1n} is the ID of key k_{1n} on the first key-chain, and n_{2n} is the ID of key k_{2n} on the second key-chain. Each sensor, i.e., typical node, is pre-loaded with $n_{1,}$ $n_{2,}$ and an initial key K_{init}.

n_1 and n_2 are the IDs of each key on the two key chains, that have been generated by BS for each sensor, and $K_{init} = K_{1n1} \oplus K_{2n2}$.

The Cluster Head, i.e. the highly equipped node will be pre-loaded with F_1, F_2, K_{10}, and K_{20}, where F_1 and F_2 are the two one-way functions, K_{10} and K_{20} are the first keys on each key-chain.

2.1.2. Pair-wise Key Establishment

This step may happen in different forms depending on who are these pairs. We have pair-wise key establishment and authentication that happens between nodes of the same type, cluster key establishment and authentication that happens between two different types of sensor nodes, and the global key establishment and authentication that happens in what called the distributed WSNs that has only a manager node without the existence of the cluster head.

The Key-chain technique uses two types of key establishment, as a first step, it makes a pair-wise key between the cluster head and sensor, but before doing it, the node that has the ability to play the cluster head role will generate two random numbers N_1, N_2 and calculate the cluster key K_{brod}, as $K_{brod} = K_{1N1} \oplus K_{2N2}$. The cluster head could do this calculation because it is pre-loaded with F_1 and $F_{2,}$ the two one-way functions.

The scenario of forming the cluster head starts with broadcasting a Hello message by the cluster head to the nearby sensors, this message includes the ID of the cluster head and will be called message 1. Each sensor receive a Hello message will join the cluster of the cluster head that sends a Hello message with the best signal noise ratio (SNR). After choosing the cluster head, sensor sends message to the cluster head, this message contains the sensor ID, and the two random numbers n_1 and n_2, and this message is message 2 in this scenario. When the cluster head receives message 2, it calculates the K_{init} for each sensor joined its cluster and find a new key called the pair-wise key K_{pair}, where $K_{pair} = K_{|N1-n1|} \oplus K_{|N2-n2|}$. This step comes to authenticate the sensor's legitimacy. Then the cluster head generates a random number R_1 for the next time communication. After this the cluster head will send message 3 for the sensor, which contains an encryption for the R_1 value, K_{pair}, and K_{brod}, this encryption will be done using the K_{init}. So only the legal sensor has the right K_{init}, and could decrypt message 3 to obtain the information with it. If the sensor is a legal one, it will get the values of R_1, K_{pair}, and the cluster key K_{brod}, and stores those values. Then it will reply the cluster head with message 4, which has the encryption of R_1 with K_{pair}. The cluster head must decrypt message 4, and checks R_1, in case it matches the original value, the cluster head would store the sensor ID and K_{pair}. Till now the cluster head and each sensor belong to its cluster establish pair-wise key for future communication.

As a second step in the Pair-wise key establishment phase, the Key-chain approach uses another type of pair-wise key establishment, the type that happens between two nodes of the same category, i.e. between the clusters heads. Communications between the BS and the cluster heads could be achieved by using the relaying strategy. All cluster heads send data to the BS via multi-hops of other cluster heads. At beginning, the distant or the far away cluster head tries to join into the close cluster, and that means it will be a child node for the cluster head of this cluster. After the successfully joining, both cluster heads broadcast the random number N_2, and this is message 1 for this scenario. Then both cluster heads can calculate the pair-wise keys use the following equation, where N_2' is the other node random number.

$$K_{pair} = \begin{cases} K2N2 & ; N_2 = N_2' \\ K_{2N2} \oplus K_{2N2}; & N_2 \neq N_2' \end{cases}$$

At last, cluster head distribute the cluster key to child cluster head in message 2, after encryption this information using the calculated K_{pair} in the previous step. By now, all the networks keys establishment processes have been finished. In the future communication, the produced keys can be used to encrypt sensitive data, so protecting the WSN from attacks.

2.1.3. Key Renovation

This step means having the ability to re-keying the sensors with new keys as a way to have an intrusion detection mechanism to detect compromised nodes. So the key renovation and revocation phase is an essential component in key management techniques.

Using the Key-chain technique that mentioned in [6], each sensor in the cluster has a unique pair-wise key, if we don't consider the probability of more than two sensors pre-loaded with the same n1, n2. If a node is compromised, we only need to delete the related pair-wise key in its cluster head. This will not affect the other nodes and links. As soon as a node is compromised or the key period expired, the cluster head will renew the cluster key. Then cluster head generate another two random numbers *N1'*, *N2'*, calculate the new cluster key $K_{brod'}$, and distribute it to the cluster numbers encrypting with each pair-wise key. To reduce the communication costs, a piece of message can include several nodes key renovation information. The format of the message is:

$id_1 \| id_2 \| id_3 \ldots \| E_{Kpair\_1}(K_{brod}') \| E_{Kpair\_2}(K_{brod}') \| E_{Kpair\_3}(K_{brod}') \ldots$

Where, *id1*, *id2*...are sensors IDs; *Kpair_1*, *Kpair_2*…are sensors pair-wise keys. Renovation message is broadcasted to all sensors in the cluster. The corresponding sensors receive the message and get the new cluster key. To keep the security of pair-wise key, cluster head must renew the pair-wise key periodically. The process is similar to cluster key renovation.

2.2. The Second Technique that Mentioned in [7]

In this approach, they depended on the idea of a group key management, where the group key is used to secure the communication for each group in the WSN, and it is usually created and updated by a central key server (sink node), and then securely distributed to all members in the same group.

Working under this scheme requires some assumptions that need to be taken, such as that the sensor nodes in the network are uniquely identified, i.e. has a unique ID number i. Each node is innocent before deployment, and cannot be compromised during the first few minutes following deployment. Compromised nodes shall be revoked by the sink. Each node can use the watchdog mechanism [8] to identify the compromised nodes. Sensor nodes are grouping into clusters. Each cluster has a leader, referred to as a cluster head (CH). CHs are elected by specific algorithms [9] and form a second tier network. Each CH stores the identities of other CHs after the CHs elections. In this work, they used the threshold cryptography [10], which is used for distribution of trust in key management and a (n, k) threshold scheme allows n parties to perform cryptographic operations, so that any k parties can jointly perform key discovery whereas (k-1) parties cannot devise any information, even after collusion.

2.2.1. Pre-distribution Phase

The sink node is the node who decides how many groups (clusters) can be formed in the network, each group will take a unique 2t degree bivariate polynomial, and since the count starts from zero, the maximum degree will be equal to 2t–1, as in the following equation, $g(x, y) = \sum_{i=0,j=0}^{2t-1} b_{i,j} x^i y^j$ is constructed over a prime finite field, where t is related to the number of nodes in each group, and y is a variable that represent the current group key version which started with zero value when the node is deployed and increased by one each time the key is updated. Accordingly, each sensor node u gets its own secret g(u, y), as illustrated in Fig.1, node V1 … V6 and CH get their own secrets g(V1, y) … g(V6, y) and g(CH, y).

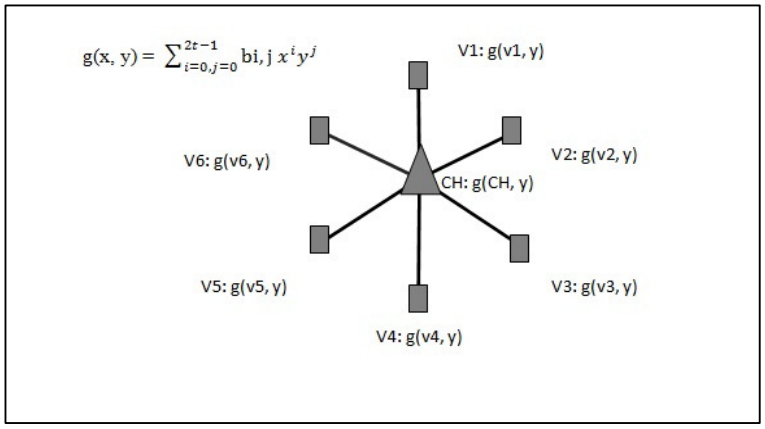

Figure 1. Group Key Pre-distribution [7]

2.2.2. Pair-wise Key Establishment

As mentioned before, the pair-wise key establishment could happened in many forms, depending on the type of the pair nodes. In this approach, the communication is happened between the CH and the member nodes in the cluster, so to secure this communication we need this key, which is called the hierarchical key. To generate this key the CH uses a one-way hash function and the ID of the member node that will communicate with the CH to get $H(ID_1)$ …. $H(ID_n)$, as an output of the hash function, where n is the number of member nodes in the cluster. Then the CH will send this key to the corresponding node through unicast traffic, and it saves all the generated keys in the hierarchical key table in its memory. As a result each member node will have its own secret key, and the CH knows all the hierarchical keys that related to all the member nodes in its cluster.

2.2.2.1. Group Key Recovery

This step represents the reforming of the group key that was given in the pre-distribution phase. This process is done by the CHs. So to get the current group key, the CH shall look for help from t member nodes, by broadcasting the help request to all member nodes in its cluster. In its turn, each member node that receives such request and is willing to trust its cluster leader will return its own secret share, g(u, y). As illustrated in fig.2.

As a result, the CH gets the shares from each member node in its cluster, but there is a possibility to receive incorrect shares from compromised nodes, so if the CH succeeds in getting to correct personal secrets, it can obtain the current group key by using the threshold secret sharing approach, which divide data (D), i.e. the group key here, into n pieces, D_1, …, D_n, in such a way that:

- Knowledge of any k or more D_i pieces makes D easily computable.
- Knowledge of any k-1 or less D_i pieces leaves D completely undetermined (in the sense that all its possible values are equally likely).

The CH can get the group key by estimating (t+1) points using the same polynomial that used in the pre-distribution phase, the CH can evaluate g(0, y) and therefore obtains the current group key $K_g = g(0, y)$.

After calculating the new K_g, the CH needs to send this key to all the member nodes in a secure manner, so it uses the hierarchy key for each member node to encrypt the K_g and send it to each member node.

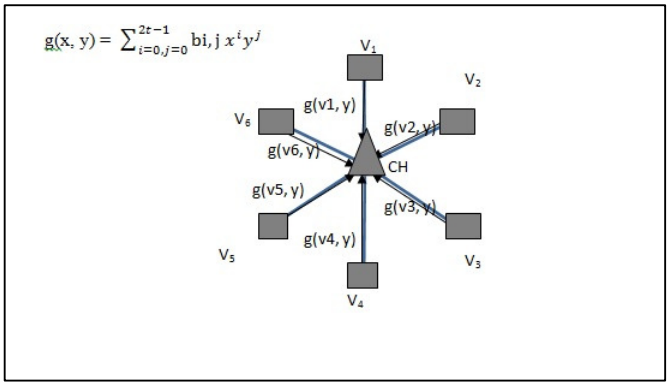

Figure 2. Group Key Reconstruction [7]

2.2.3. Key Renovation

In this scheme, this step means having the ability to re-keying the group key if an adversary captured a node, cause the adversary will be able to steal its key material, therefore the group key needs the update process that is done by following the next steps:

- If a node (A) becomes a compromised node, this approach supposes that the neighbor node (B) is the node who detects the new state for (A), and (B) after detecting this case needs to do a check to see whether (A) is a CH. If yes, (B) sends the identity of (A) to the sink node directly. In its turn, the sink node will inform the nodes who are members at the compromised cluster to do a new CH election. Otherwise, the sink node discards the message. On the other hand, if (B) detects that (A) is a normal node, the identity is send by (B) to the CH, and if the message convinced the CH that (A) became a compromised node and the CH authenticated the message, it erases the hierarchy key that belongs to the node (A) from its memory, then complete the procedure of the re-keying process. Otherwise, it ignores the message from (B).
- The involved CH sends a group re-keying message with timer T to other CHs in the group, where T is for synchronizing group re-keying. In the first place, the other CHs need to check the legitimacy of the message, if it is valid, the system will go back to do the group key recovery operation again, and then follow the re-keying process steps. Otherwise, ignoring the message.
- As a result to the previous steps, the CHs have generated a new version of the group key, the published this key to all the member nodes after encrypted it with the hierarchy key related to each member node, but this distribution for the new K_g does not include the compromised nodes, thus the CH already erase their corresponding hierarchy keys.

2.2.3.1. Node Join and Leave

This step is considering as a form or a type of the key renovation phase, the node joins means the adding of new node to a specific group, and while this node will need the key materials used in this network, so this scheme supported its needs by letting the sink node assigns the new node (U) into a specified group and loaded it with the current group key K_g, and the personal secret g(u, y), as well. So when the new is deployed into the network, it broadcasts its ID with a join requesting message to the neighbors, with the help of the sink node, the CHs within the communication range check the validity of the message, if legal, all the involved CHs replied the new node with a message that has its cluster ID. The new node studies the offers, and picks the offer that related to the nearest CH to save its energy resources, so the winner CH establishes a hierarchical key with the new node. After that, the new node is ready to function in the network just as the already existed once.

The nodes leaving means that the network lost some of its nodes, and that is happened for many reasons, such as, the enervation of the sensor power, or losing the controlling on some node because they become compromised nodes by the adversary. If the leaving node was a normal node, i.e. not CH, its CH should erase its hierarchical key from its memory, and perform the group keying process again as mentioned before. But if the leaving node was a CH, nodes in its cluster must do an election process to select a new honest CH, and that happened with the help of the sink node, then the new CH will repeat the phases of calculating new hierarchical keys for its members, and updating the K_g.

2.2.3.2. Node Isolation

This is needed, because this approach depends on its nodes to function well, so if a large number of nodes become compromised, the CH may not be able to obtain the new group key due to the lack of assistant from the member nodes. As a result, all the compromised nodes in this cluster must be isolated. In this scheme, the CH who fails in reforming the group key sends a cluster dissolving message to the sink node and to the innocent nodes that remains in his cluster, the purpose of this message is to let them know that he is willing to quit, and his cluster is won't be exist anymore, it erases the hierarchy key table, and this CH will also help the innocent nodes in his cluster to join the a new cluster, it did this by sending a joining message to the other CHs in the group, this message has the identities for all the innocent nodes in the old cluster. The CHs check the validity of this message, if so, the replied the CH with a joining accept message. The CH studies all the received offers and picks the one from the nearest CH, then informs all the innocent nodes to join. After the successful join for the nodes, the new CH for those nodes will establishes hierarchical keys with them, and then doing the group key renewing again.

2.3. The Third Technique that Mentioned in [11]

This scheme tries to find a key management approach with less energy consumption, it based on dividing the WSN into three-level models as illustrated in Fig.3. The first level, the network is organized with Normal Nodes (NN), the second level with Cluster Heads (CH) and on the top

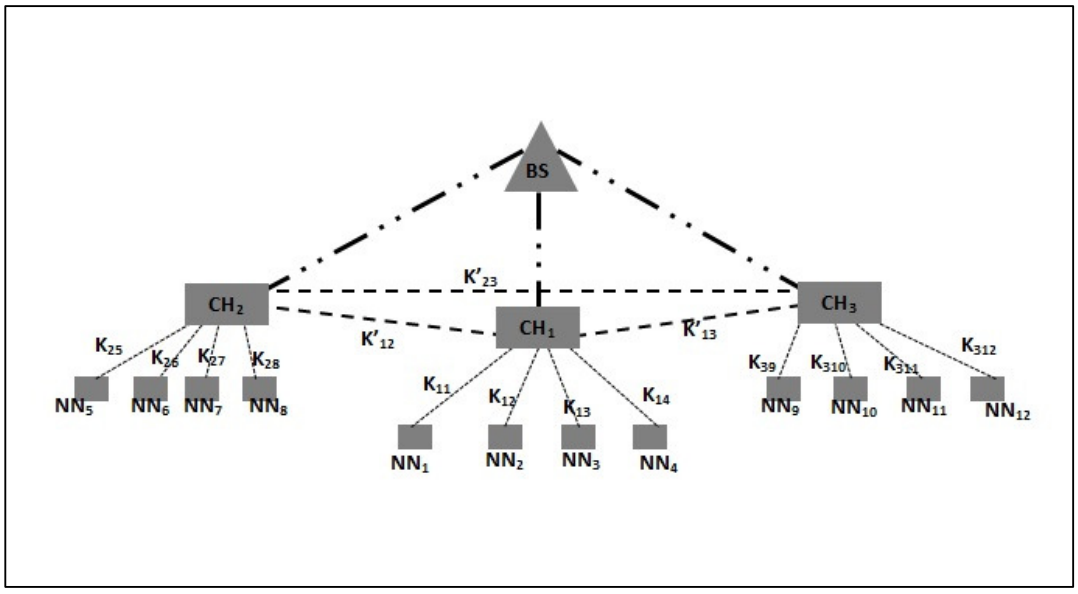

Figure 3. Network Model [11]

level is the Base Station (BS).

As the figure shows, the communications in this network happens between the normal nodes and the corresponding cluster head, between the neighbors CHs, and between the CHs and the BS. The BS has unlimited resources capabilities and strong computational power, the CHs also powerful devices but not as the capabilities for the BS, and the normal node is the one with limited resources and low memory space.

2.3.1. Pre-distribution Phase

The BS is the node that administrates this network, and to ease its task, each sensor in the network is given a unique ID which is equals to the column number in the generated matrix and points to the value of that column, which will be saved in the sensor. All the nodes in this network model, including the BS, are pre-loaded with the following uniform rule, and if needed, more than one uniform rule will be used: $a_i = a_{i-1} \oplus (a_i \vee a_{i+1})$.

Where a_i is the i^{th} bit of a binary string, a_{i+1} is the right bit of a_i and a_{i-1} is the left bit of a_i. The BS also set with an initial state for a binary string, and a variable called KM that represents the number of iteration that will be applied using the uniform rule to generate a key.

2.3.2. Pair-Wise Key Establishment

This is to obtain the keys that are needed for the three types of communications in this network that mentioned before. First the initialization process for the network must take place. The BS will generate a matrix C_{kp} at random in a WSN with N nodes and it is equal to the number of columns in the generated matrix, as well, where p is the number of generated keys, and k is the length of the needed keys, while P-N represents the maximum number of new sensors that could be added to this network. As we said, this phase will include the following steps:

2.3.2.1. Key Establishment between the BS and the CH

After finishing the computing for the BS and the CH, this process begins, with the following sequence:

- Clusters need to be formed, and then the BS broadcasts a message to all CHs in the network, informing them to start the key establishment process.
- Cluster heads send their IDs to the BS.
- The BS will use the IDs values it receives to refer to the matrix C_{kp} and obtains the column value, since the IDs represented the columns numbers in this matrix, then the BS will use the value of KM and the uniform rule to compute the pair-wise key.
- As a reply from the BS, it sends the value of KM and the list of the generated IDs to the corresponding CHs.
- Each CH checks out if its ID included in the list or not.
- CHs calculate their keys upon the column value they have and the KM they got from the BS, and save the keys.

2.3.2.2. Key Establishment between CH and another CH

The cluster heads already have a binary string that represented an initial state that used to generate the keys that are needed to be calculated before; the steps of this process are as follow:

- Thus each CH has a column value that pre-loaded on it; neighbor cluster heads sends its column value for each other's, so every CH will has its column value C_i its neighbor column value C_j.
- Both CHs that did the previous exchange calculate $C_i \cup C_j$ accordingly, and the output of this calculation is used as an initial state values for the uniform rule.
- CHs calculate respectively depending on the uniform rule and the KM value as the number of iterations, and then save the new keys.

2.3.2.3. Key Establishment between Cluster Head and Normal Sensor Nodes

This step is little bit similar to the one happened between CHs and the BS, but here it needs to work in two stages, one between the normal node & the CH, and the second one between the CH and the BS, as following:

- Normal sensor nodes within a cluster send their ID to the corresponding CH.
- In its turn, the CH at this point just passes the list of received IDs to the BS.
- The BS queries the matrix C_{kp}, and then sends the column value according to the IDs list and attaching the value of KM with it to the CH.
- Upon receiving the previous information from the BS, the CH does its calculation, and ate the same time, broadcasts KM and the list of IDs to the sensor nodes within the cluster.
- Normal sensor nodes check out to see if their IDs are in the received list, and also it computes using KM and the uniform rule its key, and save the result.

2.3.3. Key Renovation

As mentioned before, this step is to rekeying and updating the whole keys in the network, and in this approach this is done when a CH is captured by an adversary. This scheme assumes that the BS station is the one who senses that the CH was captured. So when the BS detects the capture of a cluster head, it must notify the other CHs in the network to refresh the keys. And the BS will generate a new random number as a value for KM that represented the number of iterations applied to the uniform rule. So the whole key establishment's stages will be repeated, just by using the new value of KM and the same binary string that was assigned to each sensor.

2.3.3.1. Sensors Addition and Deletion

As usual, this step is considered as a form or part of the key renovation phase, and just as when the network nodes are deployed for the first time, the new node that will to join the network will be pre-loaded with the same needed information, such as the node ID that represented the column number in the matrix C_{kp}, the column value, and a uniform rule. When the node is deployed into the network, it firstly broadcasts its ID; the nearest CH that hears this ID will pass it to the BS, which checks it. After this, the process of key establishment is begun and it is just as the one happened between the CH and the normal nodes.

Nodes deletion in this network model happens for the normal nodes only, because it has less energy level comparing to the CHs nodes. So when the power of the normal node is approximately depleted, it informs its corresponding cluster head which in its turns will erase the ID and the key related to this node.

3. ANALYSIS FOR THE AFOREMENTIONED TECHNIQUES

The Analysing for the previous approaches will be discussed by summarizing the advantages and the disadvantages for each scheme, and by trying to find out some recovery ideas for the weakness point in those techniques from our point of view.

- **First Technique**

This approach uses two one-way hash functions in generating two key chains, this will prevent the adversary from getting a successful trial to discover the key materials, but the expected

problem is with pre-loaded the CHs in this network with the two hash functions and the beginning key for each chain, and since this approach didn't present any solutions for facing the problem when the CH become a compromised node, and if such thing happened, it will be easy for the attacker to control the network that depends on this approach, although the approach assumes that the CH node is equipped with a tamper resistance hardware to protect it. This is not good enough because the tamper resistance is not absolute; an opponent with access to semiconductor test equipment can retrieve key material from a smart card chip by direct observation and manipulation of the chip's components, [10] add to the previous, the cost that it takes in equipped all the CH nodes with this hardware.

Another expected problem with this technique, is the way it used to form the clusters, it uses the SNR mechanism, and this one works by let the CHs send a Hello message for all of the nearby nodes, and the nodes will attend the cluster that belongs to the CH that sends a Hello message with the highest SNR value, this gives the adversary a great chance to attack such approach cause he could put a spoofed CH that propagate a Hello message that has a greater SNR value than the one related to the honest CH, thus ruin the network and let the attacker control the mechanism.
One more thing to the previous comments, the approach, in our opinion, may need another step to be done in the pre-loaded phase, which is using a DB for the random numbers that are generated for the nodes, this is to avoid the probability of having one or more sensors pre-loaded with the same random numbers, it also will help in making a flag for these numbers to let the admin distinguish between the ones for normal nodes (denoted by small n in the original paper) and the ones related for CHs (denoted by capital N in the original paper), and this may be needed, because when the CH try to join another cluster, it broadcasts one of its random numbers in the network, so the normal node will receive this extra message and process it. But if we separate the ranges for both kind of random numbers and used a multicast traffic instead of the broadcast, this may decrease the overhead on the network and contributes in saving the sensor power, thus improve the efficiency of this technique. Another idea to do that is to apply the approach of gateway CHs to achieve the purpose here.

- **Second Technique**

This scheme depends on the idea of hashing the sensor IDs to generate what called the hierarchy key that used between the normal node and the CH. The expected problem is that when sending this key for the normal node, the CH sends H(ID) to each sensor, and this is subject to the attackers attack, because they may steal this key and makes use of it, especially it is the one that used for encrypting another sensitive info which is the group key. As a trial to solve this problem, the scheme could apply the MAC technique for this communication to protect such info.
Another issue with this approach, that it sends a lot of unicast traffic that make an extra overhead on the network, besides to the big assumptions it supposed for the re-keying phase which consumes many of the sensors resources.

- **Third Technique**

This scheme assumes that the BS has unlimited resources capabilities and strong computational power, thus most of the materials that are needed for generating the keys are located in the BS and tied to it.

This approach doesn't allow the nodes, regardless to its category type, to exchange the secret keys; it just exchanged the IDs. These IDs points to binary strings which are used to produce the needed keys after applying the uniform rule KM times on it, but it doesn't handle the situation when the ID that related to a sensor doesn't existed in the received list, this will leave such sensors

without a key, so it won't be able to send its data in a secure manner or it may use the pre-loaded binary string as a key, and this is subject to be stolen by the attacker.

As a proposed solution for such issues, the next section has modifications for the previous scheme, as a trial for a proposed scheme.

4. PROPOSED IDEA AND FUTURE WORK

Here we are trying to find a compromised technique that solves some of the expected problems in the previous techniques. Our technique will use same matrix that mentioned in [10], shown in table 1, but with making some modifications to improve the technique. We will use the idea of pre-loaded the sensor with an initial binary string that is used in forming the clusters in the WSN.

Table 1. Matrix that located in the BS

ID_1=1	ID_2=2	ID_3=3	ID_4=4	ID_5=5	…	ID_n=P
1	0	1	1	…	…	1
0	1	1	1	…	…	0
1	1	1	0	…	…	0
. . .	. . .	. . .	. . .	. . .	. . .	. . .
k	k	k	k	k	k	k

4.1. Cluster Forming

- Divide the nodes into two halves, even half and odd one.
- Divide the target area (field where the sensors will be deployed), into two halves, area O (odd), and area E (even). So when disseminate the sensors, we need to deploy the even half in E, and the odd half in O.
- The CH with an odd ID (3, 5, 7...), will allow for the nodes that has a binary string with an odd parity to join its cluster.
- The CH with an even ID (2, 4, 6...), will allow for the nodes that has a binary string with an even parity to join its cluster.

4.2. Keying the Network

- CHs send their IDs to the BS, the BS applying the uniform rule on the column values related to these IDs with number of times equal to the number of ones in the binary string, so no need for generating KM.
- When the CHs receive back the list after the BS computation, if its ID doesn't exist, it generate its own key by applying the rule with a number of rounds equal to the number of ZEROS in its binary string, then sends a message to the BS in a form of: (ID,0), to let the BS do the same calculation to get the same result for the key value, and if the binary string doesn't have 0s, the CH depends on the number of 1s as the number of rounds, and also sends the message (ID,1) to the BS to do the same.
- Same approach between normal nodes and CH.

Our proposed idea similar to the scheme in [11], that both of us use the binary matrix idea. On the other hand, we differ in preloading the normal sensors with an initial binary string that is used in

forming the clusters in the WSN, and this will help in overcoming the expected problem in [11]. Add on that, we use different technique in dealing with the target terrain that the sensors will be deployed in, and different approach in forming the network clusters and keying the network as explained before.

In the near future we are planning to implement our proposed idea, and conduct some experiments as a trial for improving our point of view, and do some sense comparisons between our proposed techniques and the aforementioned one.

5. ANALYSIS FOR THE PROPOSED IDEA

This approach won't need much cost cause of the efforts that needed to classify the sensors into two groups, and in dividing the network area comparing to predefined location techniques such as the one mentioned in [11], thus no need to the pre-knowledge for each sensor location, all we need to know which part of the network field will has the odd sensors and which one will hold the even ones, thus the group location not individual coordinates for all sensors.

Another point, related to the security part of this approach that if the attacker tried to access the network, by adding a new sensor, he won't know in which part he could add his node, and if he successes by chance, he won't know the rules this technique depends on in making the keying process, which is sending the number of one's or number of zeros, even though he steal the exchanged management messages, he won't be able to get the complete picture of the approach.

6. CONCLUSION

The previous paper focused on a very significant security issue, which is the Key management in WSN. This problem has attracted the researcher's attention and makes them find out many techniques to solve it out. We explained three techniques for key management problem in WSN, and the work went in classifying those techniques upon three phases. Phase one, is the pre-loading stage, then the pair-wise key process, and finally the key renovation phase. Then we made some analyzing for those mechanisms by highlighting their advantages and disadvantages. After that, we presented some modification as a proposed idea to map new key management technique with less overhead by making use of pre-loaded information for security purposes in forming secure clusters and decrease some work efforts such as generating random number to get the number of needed iterations that would applied into uniform rule, and good robustness in facing WSN attacks. Hence, this paper is important to highlight the main steps for the key management process in WSN, and helps in simplifying the idea behind it.

REFERENCES

[1] Mohammed A. Abuhelaleh, and Khaled M. Elleithy, (2010), “SECURITY IN WIRELESS SENSOR NETWORKS: KEY MANAGEMENT MODULE IN SOOAWSN”, International Journal of Network Security & Its Applications (IJNSA), Vol.2, No.4, pp. 67-78.

[2] Yang Xiao, Venkata Krishna Rayi, Bo Sun, Xiaojiang Du, Fei Hu, and Michael Galloway, (2007), "A survey of key management schemes in wireless sensor networks". Science direct, Computer Communications, pp. 2314–2341.

[3] http://nesl.ee.ucla.edu/tutorials/mobicom02/slides/Mobicom-Tutorial-2-MS.pdf (2014), Viewed online on 16th Nov 2014.

[4] Chi-Yuan Chen, and Han-Chieh Chao, (2011), "A survey of key distribution in wireless sensor networks", Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.354.

[5] Ritu Sharma, Yogesh Chaba, and Yudhvir Singh, (2010), "An IPC Key Management Scheme for Wireless Sensor Network, 2010 1st International Conference on Parallel, Distributed and Grid Computing (PDGC - 2010), pp. 251-255.

[6] Guohua Ou1, Jie Huang, and Juan Li, (2010), "A Key-Chain Based Key Management Scheme for Heterogeneous Sensor Network", pp. 358-361.

[7] Yuan Zhang, Yongluo Shen, and SangKeun Lee, (2010), "A Cluster-Based Group Key Management Scheme forWireless Sensor Networks", 2010 12th International Asia-Pacific Web Conference, pp. 386-388.

[8] S.Marti, T.Giuli, K.Lai, and M.Baker, (2000), "Mitigating routing misbehavior in mobile ad hoc networks", Proc. ACM International Conference on Mobile Computing and Networking, pp. 255–265.

[9] K.Kim, Y.Zhang, W.Yang, and M.Park, (2008), "An authentication protocol for hierarchy-based wireless sensor networks", in Proc. IEEE International Symposium on Computer and Information Sciences, pp. 1–6.

[10] A. Shamir, (1979), "How to share a secret",Communications of the ACM, pp. 612–613.

[11] Lanying Li, and Xin Wang, (2010), "A high security dynamic secret key management scheme for Wireless Sensor Networks", Third International Symposium on Intelligent Information Technology and Security Informatics, pp. 507-510.

Authors

Khawla Naji Shnaikat: has a B.Sc. degree in Computer information Systems/ specialist in computer networks with 3.25 GPA, and the M.Sc. degree in Computer Science with 3.94 GPA from University of Jordan. Find more on: Linked in.

Ayman Ahmed AlQudah: has a B.Sc. in software engineering, July 2008 from Hashemite University, Jordan and has an M.Sc. in Information Technology Management (ITM), July 2012, from the University of Sunderland, UK. Find more on: Linked in.