IJCSN International Journal of Computer Science and Network, Volume 4, Issue 3, June 2015
ISSN (Online) : 2277-5420 www.IJCSN.org
Impact Factor: 0.417

Performance Analysis of Multiple Key Management Schemes in Wireless Sensor Network

[1]Amr M. Kishk, [2] Nagy W. Messiha, [3] Nawal A. El-Fishawy, [4] Abd-Elrahman A. Alkafs, [5] Ahmed H. Madian

[1] Reactor department, Egyptian Atomic Energy Authority (EAEA)
Nasr City, Cairo, 9621, Egypt

[2] Electronics and Electrical Communication Engineering, Faculty of Electronic Engineering (FEE), Elmounfya University
Menouf, Elmounfya, 23952, Egypt

[3] Computer Science and Engineering, Faculty of Electronic Engineering (FEE), Elmounfya University
Menouf, Elmounfya, 23952, Egypt

[4] Reactor Department, Egyptian Atomic Energy Authority (EAEA)
Nasr City, Cairo, 9621, Egypt

[5] Radiation Technology Engineering, Egyptian Atomic Energy Authority (EAEA)
Nasr City, Cairo, 9621, Egypt

Abstract - Wireless Sensor Network (WSN) is frequently exposed to attackers. The data transmission in secure channel enhances the data confidentiality. This paper discusses multiple schemes with different approaches. Periodic key updating becomes a main point in the design of each scheme. The key updating enhances the security level because of the difficulties added in the network cracking. The performance of each scheme is discussed to show the drawbacks of the public ideas used in the design of these key management schemes. One of these drawbacks is the drawback in the mathematical use of the modular function to generate a shared key between the two parties. This drawback is used in the new schemes and it is illustrated in a numerical example in this paper. The results show also the preference of key updating with each packet to the other discussed schemes.

Keywords - *Pairwise Key; Shared Key; Key Updating; Public and Private Keys.*

1. Introduction

The data exchange between the Bases Station (BS) and the sensor nodes in secure channels is the main task of the key management schemes. These schemes authenticate both the sensor nodes and the BS with each other and manage the keys distribution between them. Attackers have the ability to analyze the traffic to get the encryption keys [1]. Hence, periodic updating of these keys enhances the security level and leads to increase the difficulties of the network cracking.

Multiple key management schemes are summarized and discussed in this paper to show different aspects used to authenticate the sensor nodes and BS with each others. the drawback of Diffie Hillman Algorithm will be discussed in this paper since it is repeated in many new schemes. This drawback is in the use of modular operation to get the shared key and it will be shown in the discussion of the Diffie Hillman algorithm.

The main problem in the design of key management schemes is the exchange of the encryption keys without cracking. Some of these schemes are to secure the communication channels using preloaded initial key, others use two different keys, public and secret keys, and third category of schemes considered the absence of attackers with the initial communication and no need to the initial key. The performance analysis of these schemes will show the best approach used to secure the communication channels in Wireless Sensor Network (WSN). The organization of the paper is as follows: section 2 introduces the related work of the key management schemes while their performance analysis

discussed in sections 3. Finally, the conclusions of these discussions are shown in the end of this paper.

2. Related Work

Scheme 1: Diffie Hillman Encryption Algorithm

Diffie Hillman algorithm (key-exchange) was developed by Whitfield Diffie and Martin Hillman at Stanford University [2]. It can be used to establish a shared secret key that can be used by the two parties for data encryption algorithm. The Base Station (BS) announces two keys, p_{DH} and g_{DH}, and uses a secret key, x_{DH}. Each sensor node in the WSN has a secret key, y_{DH}. The procedures to generate the shared key-used to encrypt the identities of both sensor nodes and BS during the authentication process-are as follows:

Step 1: the BS uses the two public keys, p_{DH} and g_{DH}, and its secret key, x_{DH}, to generate a new key, K_{BS}, as in equation (1) to send it to the sensor node.
Step 2: The sensor node generates a shared key, K_{DH}, used in identities encryption from the received key, K_{BS}, and its secret key, y_{DH}, as in equation (2).
Step 3: The sensor uses the two public keys, p_{DH} and g_{DH}, and its secret key, y_{DH}, to generate a new key, K_{SN}, as in (3) to send it to BS.
Step 4: BS generates the shared key, K_{DH}, from the received key, K_{SN}, and its secret key, x_{DH}, as in (4).

$$K_{BS} = g_{DH}^{x_{DH}} \bmod p_{DH} \quad (1)$$

$$K_{DH} = g_{DH}^{x_{DH} y_{DH}} \bmod p_{DH} \quad (2)$$

$$K_{SN} = g_{DH}^{y_{DH}} \bmod p_{DH} \quad (3)$$

$$K_{DH} = g_{DH}^{y_{DH} x_{DH}} \bmod p_{DH} \quad (4)$$

Scheme 2: Rivest, Shamir and Adleman (RSA) Encryption Algorithm

In the RSA algorithm, the BS announces two keys and keeps two secret keys. The sensor node uses the two public keys to encrypt the message and the BS uses one of the secret keys to decrypt the message. The RSA steps are as follows [3]:

Step 1: BS chooses two very large prime keys; p_{RSA} and q_{RSA}.
Step 2: it calculates $n_{RSA}=p_{RSA}\times q_{RSA}$
Step 3: it calculates $\Phi_{RSA}=(p_{RSA}-1)\times(q_{RSA}-1)$
Step 4: it chooses a random prime key, e_{RSA}, and then determines d_{RSA} as in (5).
Step 5: BS sends e_{RSA} and n_{RSA} as public keys used in the identities encryption.
Step 6: sensor node encrypts a plain text, P_{RSA}, with the two keys, n_{RSA} and e_{RSA}, as in (6) to get a cipher text, C_{RSA}.
Step 7: BS decrypts C_{RSA} by using d_{RSA} and n_{RSA} to get P_{RSA}, as in (7).

$$(d_{RSA} \times e_{RSA}) \bmod \varphi_{RSA} = 1 \quad (5)$$

$$C_{RSA} = P_{RSA}^{e_{RSA}} \bmod n_{RSA} \quad (6)$$

$$P_{RSA} = C_{RSA}^{d_{RSA}} \bmod n_{RSA} \quad (7)$$

Scheme 3: ElGAMAL Encryption Algorithm

The BS chooses a secret key, x_{EEA}, and broadcasts three public keys (g_{EEA}, q_{EEA}, and G_{EEA}) to all sensor nodes [4] and also broadcasts $h_{EEA} = g_{EEA}^{x_{EEA}}$ to all sensor nodes for authentication. The sensor node computes $C_{1EEA} = g_{EEA}^{y_{EEA}}$ and $S_{EEA} = h_{EEA}^{y_{EEA}}$ and then encrypts its message m_{EEA} by computing $C_{2EEA}=m_{EEA}.S_{EEA}$. The sensor node sends C_{1EEA} attached with C_{2EEA} to the BS. The BS recovers m_{EEA} by computing $C_{2EEA} / (C_{1EEA})^{x_{EEA}}$.

Scheme 4: Elliptic Curve Encryption Algorithm

The BS specifies the selected point, p_E, in the curve and preloads it to all the sensor nodes and generates a public key, Q_E, and $Q_E= d_E.p_E$ which is calculated from its private key, d_E, to broadcast it to all the sensor nodes [5]. If a sensor node senses a message, m_E, and in order to send it to the BS, then this sensor node will specify a point M_E on the curve corresponding to m_E. Then, this sensor node sends two messages to the BS, C_{E1} and C_{E2}, where $C_{E1}=K_E.p_E$ and $C_{E2}=M_E+K_E.Q_E$ and K_E is the private key of the sensor node. BS can recover M_E from C_{E1} and C_{E2} using the relation $C_{E2}-d_E.C_{E1}$ because $C_{E2}-d_E.C_{E1}=(M_E+K_E.Q_E)-d_E.(K_E.p_E) = (M_E + K_E.d_E.p_E) - d_E.(K_E.p_E) = M_E$.

Scheme 5: An Efficient Key Management Scheme for Data-Centric Storage Wireless Sensor Networks

Each node in the WSN is pre-loaded in a unique master key, K_{ai}, and an initial key, K_{ainit}. K_{ai} is used for secure communication between both the sensor nodes and the BS while K_{ainit} is deleted from the nodes memory after generation of pairwise keys, cell key, and Exclusion Basis System (EBS) keys during the key setup phase in each cluster [6]. The pairwise key and cell key are used to secure the communication between each sensor node its neighbors and its Cluster Head (CH) respectively, while

the EBS keys are generated in each CH to update the keys of each sensor node in the cluster. The keys updating depends on notification messages from the CH to update the keys of this cluster and also notification messages from the BS to update the keys of all sensor nodes in WSN with each CHs-election process.

Scheme 6: An Efficient Key Distribution Scheme to Secure Data-Centric Routing Protocols in Hierarchical Wireless Sensor Network

The BS uses a common key, Global key, preloaded in each sensor node to secure the communication channels of the network [7]. The BS updates the Global key to another key, Group key, to send it to all the sensor nodes. The BS uses the Group key instead of the Global key and the sensor nodes use the Group key to communicate with the CH to send its encryption key, pairwise key, to secure the communication channel between them. These keys are updated through CH.

Scheme 7: A Secret Sharing-Based Key Management in Hierarchical Wireless Sensor Network

The Secret Sharing-based Key Management (SSKM) provides various session keys based on different polynomials from BS in different periods which can protect the network and reduce the high probability of the common key [8]. The session keys, a network key and a cluster key, are used to secure the BS-CH and CH-sensor nodes communication channels respectively.

a. The network key management process

1. The BS broadcasts three keys (N, g, Q) to sensor nodes in the network where: N is generated from two prime numbers and $g \in [N^{1/2}, N]$.
2. Each CH sends (ID_{CHi}, y_i) to the BS where ID_{CHi} is the CH identification and y_i is encrypted by the secret key x_i or $y_i=g^{xi} \text{mod} N$ where: $x_i \in [2, N]$, of CH_i.
3. The BS unicasts (y_o, $f_{cin}(ID_{CHi})(y_i)^{xo}$) to each CH_i where: f_{cin} is one of general polynomials in BS shown in (8) and $y_o=g^{xo} \text{mod} N$ which is encrypted by the secret key x_o, $x_o \in [2, N]$, of BS.
4. The BS broadcasts a key, Z_{in}, to all sensor nodes to use it in the generation of the session key.
5. CH can get the network key as in (8 and 9). The network key, K, will be $K=Z_{cin}-S_{cin}$.

$$f_{Cin}(x) = \sum_{j=1}^{t} \left(\coprod_{l \neq j} \frac{x - ID_{CHj}}{ID_{CHl} - ID_{CHj}} \right) f(ID_{CHj}) \bmod Q \qquad (8)$$

$$S_{Cin} = \sum_{j=1}^{t} \left(\coprod_{l \neq j} \frac{ID_{CHj}}{ID_{CHj} - ID_{CHl}} \right) f(ID_{CHj}) \bmod Q \qquad (9)$$

b. The cluster key management process

1. The CH sends a random number, x_{CHi}, to BS.
2. The BS sends (ID_{CHi}, y_i) to the sensor nodes in the cluster where $y_i=g^{xCHi} \text{mod} N$.
3. Sensor nodes send ($ID_{i,r}$, $y_{i,r}$) to the BS where $ID_{i,r}$ is the sensor node ID and $y_{i,r}=g^{xi,r} \text{mod} N$ which is encrypted by the secret key $x_{i,r}$ of the sensor node i.
4. The BS unicasts (ID_{CHi}, f_{CHi} (ID_{CHi})$(y_{i,r})^{xCHI}$) to the sensor node in CH_i and BS sends ($ID_{i,r}$, f_{CHi} ($ID_{i,r}$)$(y_{CHi})^{xi,r}$) to CH_i.
5. The sensor node will compute f_{CHi} (ID_{CHi}) as in (10).

$$f_{CHi}(ID_{CHi}) = \frac{f_{CHi}(ID_{CHi}).(y_{i,r})^{x_{CHi}}}{(y_{CHi})^{x_{i,r}}} \qquad (10)$$

6. The CH will compute f_{CHi} ($ID_{i,r}$) as in (11).

$$f_{CHi}(ID_{i,r}) = \frac{f_{CHi}(ID_{i,r}).(y_{CHi})^{x_{i,ri}}}{(y_{i,r})^{x_{CHi}}} \qquad (11)$$

7. The sensor node sends ($ID_{i,r}$, f_{CHi} ($ID_{i,r}$)) to CH_i.
8. The cluster key can be obtained as the same as the network key.

Scheme 8: An Energy-Efficient Key Pre-distribution Scheme for Secure Wireless Sensor Networks using Eigenvector

This scheme depends on important properties of the matrix to design the key pre-distribution scheme [9]. It depends on the generation of the Eigenvector and Eigenvalues of the matrix to get the session key between two sensor nodes. These sensor nodes exchange their results, M_S and M_d, to generate the session key and updating this session key in the future depends on the new keys, M_s and M_d, and the previous session key.

Scheme 9: A Dynamic Key Management Scheme Based on Secret Sharing for Hierarchical Wireless Sensor Networks

The scheme depends on an initial key, K_D, preloaded in each sensor node [10]. This key is used to distribute IDs suggested by the BS to every cluster in the WSN and also to encrypt the encryption key, r_{D1}. The BS sends the encrypted key, $E_K(r_{D1})$, attached to its hashed code, $h(r_{D1})$. Each sensor node in each cluster decrypts the received

encrypted key to get r_{D1} which is used to generate its cluster key, CK_i, as in equation(12) and also validates the received $h(r_{D1})$ by the stored one, $h'(r_{D1})$, as an authentication process.

To establish secure communication with the BS, the CHs and the BS generate a key by a master key, BK. Firstly, the BS sends $r_{D2} \oplus K_D \| h(r_{D2}) \| E_K(y_{Di})$ to a CH, C_{Di}, where r_{D2} is another random number and $y_{Di}=f(x_{Di})$ is the response of a generated polynomial by the BS due to applying $x_{Di}=h(r_{D2}+ID_i)$ to it. C_{Di} recovers r_{D2} and y_{Di} and also regenerates x_{Di} as shown later. All CHs in WSN exchanges their IDs and their results, $ID_j \| E_K(x_{Dj} \| y_{Dj})$, and each one recovers $(x_{Dj} \| y_{Dj})$ using the shared key, K_D, and generates the master key, BK, as in (13). The BS updates CK and BK periodically to enhance the security level of the WSN. The updating of the CK and BK results in updating r_{D1} and r_{D2} respectively.

$$CK_i = h(r_{D1} + ID_i) \tag{12}$$

$$BK = \sum_{i=1}^{t} y_{Di} \prod_{\substack{j=1 \\ j \neq i}}^{t} \frac{x_{Dj}}{x_{Dj} - x_{Di}} \bmod q_D \tag{13}$$

Scheme 10: An Authentication and Key Establishment Scheme for the IP-Based Wireless Sensor Networks
The proposed scheme has suggested using a public key for every network and a private key generated from the public key for the communication channel between each two sensor nodes in the same network [11]. The generation of the public key depends on two generated codes: node share and network share as in equations (14-15) where (PSN, GSN) and (PN, GN) are the pair of the prime number and group generator of the network entity and Network Security Manger respectively, c_f is the point on the elliptic curve, p_f is the prime field generator, and $IP_{Network}$ is the Internet Protocol of the network. The private number is generated from the public key as in (4).

$$\text{Public Key} = S_f \oplus T \bmod p_f \tag{14}$$

Where:

- S_f= Node Share= $IP_{NETWORK} \cdot c_f \cdot G_{SN} \bmod p_{SN}$
- T= Network Share= $S_f \cdot G_N \bmod p_N$

$$\text{Private Key} = (\text{Public Key} \oplus \text{Random Number})^{-1} \bmod p_{SN} \tag{15}$$

When the sensor node moves from the network x_a to the network y_a, it requires to get the public key of the network y_a. So, it sends its node share to the Network Security Manager of the network y_a, without encryption, which generates the public key as in equation (14) and sends back to the sensor node.

Scheme 11: Authenticated Nodes in Wireless Sensor Network
The BS preloads two keys in each sensor nodes, n_s and K_s, which both vary from one sensor node to another [12]. Both keys are generated in the BS based on stored secrete key, P_v. The sensor node starts its secure communication with the BS by attaching n_s with its encrypted message using K_s to transmit them to the BS. The BS generates K_s from n_s to decrypt the encrypted message and K_s is updated with each packet during the encryption process and also during the decryption process. The keys updating processes depend on two different S-boxes [12]. The contents of these S-boxes are secret and stored in both BS and sensor nodes. The BS uses the updated K_s to reply to the sensor node and attaches new keys at the end of this message. These keys are the new n_s and K_s to use them in the next authentication process and also a key, K_p, for the communication between the sensor nodes and their CH. The CH and the sensor nodes can update their key, K_p, with the first transmission between them.

3. Results and Discussion

The discussed key management schemes showed different approaches to secure the communication channels. These different approaches can be classified the schemes into three groups. Schemes 5, 6, and 9 suggested to preload initial key(s). Schemes 1, 2, 3, 4, 7, 10, and 11 are based on public key(s) and/or secret key(s). Scheme 8 depends on the absent of attackers during the initial communication between the sensor nodes. The keys of these schemes are updated periodically depending on the previous keys to secure the communication channel or with each packet and with each message as shown in scheme 11. Through the discussion of these schemes, we note some drawbacks. These drawbacks enable attackers to crack most of these schemes. These drawbacks are shown as follows:

Scheme 1: Diffie Hillman Encryption Algorithm
There are a mistake in Diffie Hillman algorithm in his calculations. This mistake occurred in many key management schemes. This mistake is $(K_{BS})^{y_{CH}} = (g_{DH}^{x_{DH}} \bmod p_{DH})^{y_{CH}} = g_{DH}^{x_{DH} y_{DH}} \bmod p_{DH}$. The sensor node received K_{BS} and did not receive $g_{DH}^{x_{DH}}$ and p_{DH} separately. So, when the sensor node wants to encrypt K_{BS}, he will encrypt K_{BS} and will not encrypt $g_{DH}^{x_{DH}}$ alone

and both solutions are not equal; the result of $(g_{DH}^{x_{DH}y_{DH}} \bmod p_{DH})$ is always less than p_{DH} while $(g_{DH}^{x_{DH}} \bmod p_{DH})^{y_{CH}}$ is not always less than p_{DH}. And also, at BS, $(K_{SN})^{x_{CH}} \neq g_{DH}^{x_{DH}y_{DH}} \bmod p_{DH}$. The next numerical example will show the ability of BS and sensor node to generate the same shared key:

1. Let g_{DH}=5, p_{DH}=11, x_{DH}=4, and y_{DH}=7.
2. So, $(K_{BS})^{y_{CH}} = (g_{DH}^{x_{DH}} \bmod p_{DH})^{y_{CH}} = (5^4 \bmod 11)^7 = (625 \bmod 11)^7 = (9)^7 = 4782969$.
3. And also, $(K_{SN})^{x_{CH}} = (g_{DH}^{y_{DH}} \bmod p_{DH})^{x_{CH}} = (5^7 \bmod 11)^4 = (78125 \bmod 11)^4 = (3)^4 = 81$ which is not equal to $(K_{BS})^{y_{CH}} = 4782969$.
4. And also, even if the modular is applied to the results (if it is used in both parties), they are not equal because 4782969 mod 11=4 and 2187 mod 11 =9.

So, as a final result, Diffie Hillman algorithm has a mistake in his calculations and can't generate a shared key in both sides and this assumption, $(g_{DH}^{x_{DH}} \bmod p_{DH})^{y_{CH}} = g_{DH}^{x_{DH}y_{DH}} \bmod p_{DH}$, must be removed from the key management schemes because as we showed later: "the result of $(g_{DH}^{x_{DH}Y_{DH}} \bmod p_{DH})$ is always less than p_{DH} while $(g_{DH}^{x_{DH}} \bmod p_{DH})^{y_{CH}}$ is not always less than p_{DH}". $(g_{DH}^{x_{DH}} \bmod p_{DH})^{y_{CH}} = g_{DH}^{x_{DH}y_{DH}} \bmod p_{DH}$ is correct only when $g_{DH}^{x_{DH}y_{DH}} \leq p_{DH}$ and if $g_{DH}^{x_{DH}y_{DH}} > p_{DH}$ then, $(g_{DH}^{x_{DH}} \bmod p_{DH})^{y_{CH}} \neq g_{DH}^{x_{DH}y_{DH}} \bmod p_{DH}$.

Scheme 2: Rivest, Shamir and Adleman (RSA) Encryption Algorithm

- BS announces e_{RSA} and n_{RSA} as public keys. The sensor node uses these public key to encrypt his message, but RSA does not consider the inverse direction because both the sensor node and attackers know e_{RSA} and n_{RSA} to decrypt any encrypted message from BS to the sensor node. So, attacker can crack the RSA easily from the inverse direction.
- Equation (6) can be written as $P_{RSA} = \sqrt[e_{RSA}]{(m \cdot n_{RSA} + C_{RSA})}$ where: m=0, 1, 2, …. Both e_{RSA} and n_{RSA} are public keys and known. And also, C_{RSA} is known which is the received encrypted message. P_{RSA} is an integer value because each byte value in [0,255] range. So, P_{RSA} will be limited to some values because $\sqrt[e_{RSA}]{(m \cdot n_{RSA} + C_{RSA})}$ must be integer which increases the probability to crack the system to high level. So, the attackers can crack RSA from multiple encrypted packets from the sensor node to the BS.

Scheme 3: ElGAMAL Encryption Algorithm

The BS uses g_{EEA} as a public key and sends h_{EEA} to the sensor node. Both h_{EEA} and g_{EEA} are broadcasted to the sensor nodes. So, the hackers can get x_{EEA} using the relation $x_{EEA}=\ln(h_{EEA})/\ln(g_{EEA})$. And also, $y_{EEA}=\ln(C_{1EEA})/\ln(g_{EEA})$.

Scheme 4: Elliptic Curve Encryption Algorithm

- The Elliptic curve is not suitable for the encryption or for the authentication in both directions. For example: if BS tends to encrypt a message, it will use its secret key, d_E, and send C_{E1} attached with C_{E2} where: $C_{E1}=d_E.p_E$ and $C_{E2}=M_E+d_E.Q_E$. The sensor node will use his secret key, K_E, to recover M_E by the same way but, $C_{E2}-K_E.C_{E1} \neq M_E$ because $C_{E2}-K_E.C_{E1}=(M_E+d_E.Q_E)-K_E.(d_E.p_E)=(M_E+d_E.d_E.p_E)-K_E.(d_E.p_E) \neq M_E$.
- Each sensor node can get the secret key, d_E, of the BS easily as it identifies the two parameters of the relation $Q_E=d_E.p_E$ which are used to encrypt its message. In addition, the BS can get the secret key, K_E, of the sensor node from the received C_{E1} because p_E is known.

Scheme 5: An Efficient Key Management Scheme for Data-Centric Storage Wireless Sensor Networks

Although the author refers to the use of a unique master key for future secure communications with the BS, a respective master key for each node for key updating is used. The BS will face difficulties to deal with the unique master key and also the respective master key for each sensor node.

In the case of the unique master key, the author considered this solution is a drawback as he showed in his comments on the related work because the security defense might be thoroughly destroyed while attackers crack any node in the network to get the unique master key. Attackers do not need to monitor each cluster individually; they just need to monitor the reception of the BS from CHs to get all the sensed data. In the case of the respective master key, the BS can't store in its database a unique key for each sensor to update their keys and this idea is not acceptable as is clear from the author's comments on the pairwise key scheme.

It is suggested to use a common key for all sensor nodes and this key will be updated with each CHs-election process because the sensor nodes do not use this key except with CHs-election process only. And also, BS broadcasts a common key for all CHs after the CHs-election process and this key will be updated with each message.

Scheme 6: An Efficient Key Distribution Scheme to Secure Data-Centric Routing Protocols in Hierarchical Wireless Sensor Network

The sensor nodes are exposed to the maintenance for long time or reduction in their energy level. The two expected problems can prevent the senor node to share in the key-updating time. The no sharing in the key updating can cause a problem with the network to transmit a sensed data. This problem will appear in a huge number of sensor nodes and this matter is expected in the near time with the first days of operating the sensor nodes in WSN.

Scheme 7: A Secret Sharing-Based Key Management in Hierarchical Wireless Sensor Network

To recover $f_{cin}(ID_{CHi})$, $(y_i)^{xo}$ should equal to $(y_o)^{xi}$. There are two cases to calculate $(y_i)^{xo}$ and $(y_o)^{xi}$. The two cases will be shown by examples.

Case 1: let N=21, x_i=4, x_o=2, and g=6. The results of $(y_i)^{xo}$ and $(y_o)^{xi}$ will be $(y_i)^{xo} = (g^{xi} \bmod N)^{xo} = (6^4 \bmod 21)^2 = (1296 \bmod 21)^2 = (15)^2 = 225$ and $(y_o)^{xi} = (g^{xo} \bmod N)^{xi} = (6^2 \bmod 21)^4 = (36 \bmod 21)^4 = (15)^4 = 50625$. So, $(y_i)^{xo}$ and $(y_o)^{xi}$ are not equal. Note that: $(y_i)^{xo} = (g^{xi} \bmod N)^{xo}$ does not equal to $(y_i)^{xo} = g^{xi*xo} \bmod N$ because $(y_i)^{xo} = g^{xi*xo} \bmod N = 6^{4*2} \bmod 21 = 6^8 \bmod 21 = 1679616 \bmod 21 = 15$.

Case 2: Consider $y_i=(g \bmod N)^{xi}$ and do not consider $y_i=(g^{xi} \bmod N)$. Also, consider $y_o=(g \bmod N)^{xo}$ and do not consider $y_o=(g^{xo} \bmod N)$. Let N=21, x_i=4, x_o=2, and g=6. Then, $(y_i)^{xo}=((g \bmod N)^{xi})^{xo} =(g \bmod N)^{xi*xo}$ and $(y_o)^{xi}=((g \bmod N)^{xo})^{xi} =(g \bmod N)^{xo*xi}$. So, $(y_i)^{xo}$ and $(y_o)^{xi}$ are equal to each other but, if we consider case 2, then , attackers can get xi and xo from the communication channels easily because $x_i=(\ln y_i) / (\ln (g \bmod N))$ and $x_o=(\ln y_o) / (\ln (g \bmod N))$. From x_i and y_i, the attacker can get the session key easily.

As a result, the scheme can not be applied if we consider case 1 and the scheme can be cracked easily if we consider case 2. The same comment can be made to the cluster key management process.

Scheme 8: An Energy-Efficient Key Pre-distribution Scheme for Secure Wireless Sensor Networks using Eigenvector

The generation of M_s and M_d depending on the eigenvectors and the Eigenvalues is a good approach but the results of the two sensor nodes, M_s and M_d, are exchanged in no secure communication. Also, the key-updating depends on the previous session key which can be obtained from the previous process. Even if the two sensor nodes use a common channel to exchange M_s and M_d, the problem of the common key will remain there.

Scheme 9: A Dynamic Key Management Scheme Based on Secret Sharing for Hierarchical Wireless Sensor Networks

- The hash function is not suitable to generate a key for encryption because it generates a signature code with few bits.
- The attaching of $h(r_{D1})$ to $E_K(r_{D1})$ increases the possibility that attackers can get r_{D1}.
- The validation of both the received $h(r_{D1})$ and the stored one, $h'(r_{D1})$, is not right process because the receiver does not know the generated random number before the reception.
- All CHs exchange their IDs in no secure channel to generate BK so, attackers can use these IDs to crack and analyze the traffic and also these IDs can be used to discover the keys.
- The key updating process considers the sensor node active always and the updating of r_{D1} and r_{D2} periodically does not consider the maintenance of sensor nodes or the reduction of their energy level to a low value at some time which adds an obstacle to the WSN. The no sharing in key updating time means the sensor node becomes out of the communication with the sensor nodes. This problem can affect on many sensor nodes at some time and can cause a great problem and adds great efforts to the operator to deal with this problem.

Scheme 10: An Authentication and Key Establishment Scheme for the IP-Based Wireless Sensor Networks

Equation (14) shows that S_f-length is less than p_{SN} length. So, T-length must be the same length of S_f which leads us to generate a public key of a length less than or equal S_f-length as in equation (14) or P_{SN}-length. Thus it could be concluded that there is no need to use P_{SN} in equation (4) which can be rewritten as in equation(16).

$$\text{Private Key} = (\text{Public Key} \oplus \text{Random Number})^{-1} \quad (16)$$

The motion of the sensor node to a new network requires to apply these steps on a secure channel with the network security manager of the new network to get a private key to communicate with the neighboring network in the new network but, the image cannot specify the destination node ID to request the communication with this network because the communication channel is secure. Even if he knows his neighbor ID, he does not need to apply these generation steps to get a private key, instead the network security manager of the new network can generate a private key directly for both nodes (if they know each

other). So, it should use a common key for all the networks or the sensor node should request the common key of the new network before leaving his network because the sensor node does not know any secure communication with the new network.

Scheme 11: Authenticated Nodes in Wireless Sensor Network

The use of n_s and K_s enhances the security level more than the others because of:

- The dependence on an updated key, K_s, to encrypt the message.
- The keys is updated with each packet, message, and CH election process.
- The dependence on using a secret channel for each node separately without storing keys in the memory of the other sensor nodes which solves the problem of memory overload as in the pairwise key management scheme.
- No negative effectiveness on the share of the sensor nodes with the BS after the maintenance or the battery charge.
- No dependence on public key(s) to encrypt the data.
- No use of the same key to encrypt two successive packets.
- The use of updated keys with the CH rather than the updated keys used with the BS.

It is suggested to embed n_s key within the bits of the encrypted message rather than the attaching before to enhance the security level.

4. Conclusions

The data security in the WSN is based on the key management schemes used authenticate the sensor nodes to each others and also with BS. These schemes secure the communication channels in the WSN and exchange the encryption keys used to encrypt the exchanged data in the communication channels. Many different schemes are discussed in the paper. The concepts of these schemes are based on two aspects. These aspects are the preloading initial key(s) in each sensor node and the dependence on the use of public and secret key(s). The discussion of these schemes showed the drawbacks of these schemes. These drawbacks give the ability to the hackers to crack the network. The error in the use of modular operation to get the shared key and the dependence on a public key to encrypt/decrypt the data are two drawbacks discussed in this paper. The two drawbacks are famous in most key management schemes. The schemes are analyzed to show the mistakes in the use of these different approaches to avoid to use them in the future. The key-updating periodically is one of the suggestion used in the modern schemes. Most of the schemes depend on the key-updating through BS and one of the discussed schemes updates its keys with each packet independent on BS. The key-updating with each packet enhances the security level without memory overload and cares the drawbacks of the other schemes. The results show the preference of scheme 11 which based on key-updating with each packet to the others.

References

[1] V. Rathod and M. Mehta, "Security in Wireless Sensor Network: A survey", Ganpat University Journal of Engineering & Technology, vol. 1, no. 1, Jan. 2011.

[2] A. Kishk, N. Messiha, N. Ayad, N. El-Fishawy, and F. Abdel-Samie, "Enhancement in the Identities-Exchange Process during the Authentication Process", (IJCNS) International Journal of Computer and Network Security, vol. 1, no. 3, Dec. 2009.

[3] F. Moghaddam, M. Alrashdan, and O. Karimi, "A Hybrid Encryption Algorithm Based on RSA Small-e and Efficient-RSA for Cloud Computing Environments", Journal of Advances in Computer Network, vol. 1, no. 3, Sep. 2013.

[4] R. Kayalvizhi, M. Vijayalakshmi, and V. Vaidehi, "Energy Analysis of RSA and ELGAMAL Algorithms for wireless Sensor Networks", Recent Trends in Network Security and Applications Communications in Computer and Information Science, vol. 89, pp. 172-180, 2010

[5] A. Kaur, "Energy Analysis of Wireless Sensor Networks using RSA and ECC Encryption Method", International Journal of Scientific & Engineering Research, vol. 4, no. 5, May 2013.

[6] J. Huang, S. Yang, and C. Dai, "An Efficient Key Management Scheme for Data-Centric Storage Wireless Sensor Networks", 2013 International Conference on Electronic Engineering and Computer Science (ELSEVIER), IERI Procedia 4, pp. 25-31, 2013.

[7] A. Guermazi and M. Abid, "An Efficient Key Distribution Scheme to Secure Data-Centric Routing Protocols in Hierarchical Wireless Sensor Networks", The 2nd International Conference on Ambient Systems, Networks and Technologies (ELSEVIER), Procedia Computer Science 5, pp. 208-215, 2011.

[8] Z. Yiying, W. Chunying, C. Jinping, and L. Xiangzhen, "A Secret Sharing-Based Key Management in Hierarchical Wireless Sensor Network", International Journal of Distributed Sensor Networks (Hindawi Publishing Coporation), Article ID 406061, vol. 2013.

[9] S. Choi, K. Kim, and H. Youn, "An Energy-Efficient Key Predistribution Scheme for Secure Wireless Sensor Networks Using Eigenvector", International Journal of Distributed Sensor Networks (Hindawi Publishing Coporation), Article ID 216754, vol. 2013.

[10] B. Enjian and J. Xueqin, "A Dynamic Key Management Scheme Based on Secret Sharing for Hierarchical Wireless Sensor Network", TELKOMNIKA, vol. 11, no. 3, pp. 1514-1523, March 2013.

[11] S. Khan, C. Pastrone, L. Lavagno, and M. Spirito, "An Authentication and Key Establishment Scheme for the IP-Based Wireless Sensor Network", The 7th International Symposium on Intelligent Systems Techniques for Ad Hoc and Wireless Sensor Networks (ELSEVIR), Procedia Computer Science 10, pp. 1039-1045, 2012.

[12] A. Kishk, N. Messiha, N. El-Fishawy, A. Alkafs, and A. Madian, "Proposed Hierarchical Routing Protocol with Simple Nodes Locating Algorithm for Authenticated Nodes in Wireless Sensor Network", International Conference on Engineering and Technology (ICET), 2014.