European Journal of Scientific Research
ISSN 1450-216X / 1450-202X Vol. 133 No 4 July, 2015, pp.369-385
http://www.europeanjournalofscientificresearch.com

An Efficient Secure Key Management Scheme based on Secret Sharing for Hierarchical Wireless Sensor Networks

Tamer Mohamed Barakat
Department of Electronics and Communications
Faculty of Engineering, Fayoum University, Fayoum, Egypt
E-mail:tmb00@fayoum.edu.eg

Abstract

Wireless sensor networks (WSNs) have acquired a lot of interest due to huge number of applications. If WSNs are deployed in inimical ambience, the nature of sensor nodes must be considered such as the limitation of memory resources, low computation ability, short communication range and energy constraints. So that it is necessary to use an efficient and secure key management scheme to avoid the mentioned limitation issues as well as to reduce the security risk.

In this paper, we propose an Efficient Secure Key Management scheme (ESKM) based on secret sharing scheme which address the above problems. The proposed scheme generates three security keys; master, cluster, and sensor keys to provide secure data communication for whole nodes in the hierarchical structure of WSNs. Compared to other key management schemes; ESKM scheme has strong security and resistance against captured and forward secrecy attacks. Finally, the simulation results show that, ESKM scheme has low energy consumption, less key storage and low communication overhead compared to the existing key management schemes.

Keywords: Key management, wireless sensor networks, ESKM scheme, secret sharing.

1. Introduction

Wireless sensor networks (WSNs) are employed in a wide range of applications including disaster relief operations, forest-fire detection, battlefield surveillance, pollution measurement, healthcare applications, and full gas monitoring [1-3]. However, WSNs have such characteristics as the large scale of deployment, power limitation, limited computing ability and memory capability, the limitation of communication bandwidth and the dynamic characteristic. Due to these limitations, WSNs are more vulnerable to security threats than traditional wireless networks; that is, the WSNs are often suffer from variety of attacks, such as passive eavesdropping attack , impersonation attack, message replay attack, packet distortion attack, and flood attack and so on [4]. Furthermore, Carman et al [5] descries the security constraint issues of distributed WSNs especially if they are deployed in inimical ambience. These issues give the adversary a great opportunity to compromise any snsor node during basic sensor operations such as node addition and deletion or node replacement process which applied on failure nodes without physical contact.

Therefore, it is important to develop security schemes suitable for sensor networks which provide data confidentiality, integrity, freshness, availability, and authentication. A key management strategy is a suitable cryptographic method to protect the communication in WSNs which it has been intensively studied in the some literatures [11-17] that based on the hierarchical architecture of WSNs.

Y. Zhang et al [12] provide the secret sharing mechanism to distribute keys into nodes. However, This scheme was sufferd from two important problems; security problem and low performance problem. For the security problem, if the long-term private keys of one or more entities are compromised, the secrecy of previous session keys, which was established by honest entities is affected, i.e, this scheme doesn't prvide perfect forword secrecy [18]. On the other hand, this scheme consume more energy since it exchanges several messages to establish key system.

Therefore, In this paper, a new key management scheme is presented which named Efficient Secure Key Management (ESKM) shceme; that is, Efficient phrase is refers to that this scheme is a good computation and communication overhead whereas Secure phrase is refers to that our scheme has strong securtity agains catured node attaks.

Since ESKM scheme is depends on threshold secret sharing technique to generate security keys, it can able to solve the mentioned issues for Y. Zhang et al's work, that is, it provides three types of system keys: master key which protects the communication channel between each Cluster Head (CH) an the Base Station (BS), cluster key which protects the communication channel between each sensor node (SN) and its corresponding CH, and sensor key which secure communication among sensor nodes in the same CH. Then, each key will be divided to different sub keys where each sensor node only stores its sub keys. By applying the ESKM scheme, the attacker cannot be able to reconstruct the security key (master, cluster, or sensor keys) since the ESKM is based on hierarchical structure and it depends on secret sharing protocol during key update mechanism. Furthermore, the security and performance analysis demonstrate that our scheme is secured against node captured attack and forward secrecy attack. Compared with existing schemes, the ESKM has lower computation and storage overhead.

The reminder of this paper is organized as follows. Section 2 describes the related work in the field. Our motivations and contributions are given in Section 3. Section 4 presents the system model and background of secret sharing scheme. The preliminaries and assumptions are given in section 5. In Section 6 we present the proposed key management (ESKM) scheme in details. We evaluate ESKM using security and performance analysis in section 7. Finally, the conclusion and future work are presented in Section 8.

2. Related Work

We first review work in establishing shared keys in sensor network, then review several works in key management scheme in sensor network.

Eschenauer et al [6] proposed a distributed key establishment mechanism that relies on probabilistic key sharing among the nodes of a random graph and uses a shared-key discovery protocol for key establishment. Du, Deng, Han, and Varshney [7] proposed a new key predistribution scheme, which substantially improves the resilience of the network compared to the existing schemes. This scheme exhibits a nice threshold property: when the number of compromised nodes is less than the threshold, the probability that any nodes other than these compromised nodes are affected is close to zero. This desirable property lowers the initial payoff of smaller scale network breaches to an adversary, and makes it necessary for the adversary to attack a significant proportion of the network.

Blundo et al. [8] proposed several schemes which allow any group of t parties to compute a common key while being secure against collusion between some of them. These schemes focus on saving communication costs while memory constraints are not placed on group members.

A survey on key distribution and authentication for resource-starved devices in mobile environments is given in [9].

In [10], the authors provide a secure communication through exchanging secret keys between neighbor nodes without any cryptography methods. Unfortunately, in this research, the scheme is consume more energy as well as the authentication among sensors requires large amount of messages to establish a secure communication among nodes which it negatively affects on the overall network performance.

In [11], authors presented some key management schemes that based on secret sharing. In these schemes, sensor nodes split messages into subshares and forward them among several disjoint paths to defend DoS attack. Since these schemes need data aggregation using secret sharing to establish communication among sensor nodes; the attacker can easily defeat the security key as well as the energy consumption is very high.

Y. Zhang et al [12] proposed a secret-based key management scheme to enhance network security and survivability. They supposed that this scheme can't only reduce the energy consumption but also enhance the security level.

3. Motivations and Contributions

3.1 Motivations

Key Management is a major challenge to achieve security in wireless sensor networks. Sensor networks are typically wireless deployments, sometimes in hostile environments, and are subjected to greater security risks. Establishing secure communications involving the setup and distribution of secret keys is an open problem for sensor network researchers. So that, the main motivation of key managementis to increase network’s resilience against node capturewithout using more memory. A powerful adversary may defeat secret key and encryption randomness for the intercepted communication. Therefore, key management scheme in WSNs is a security mechanism that provides data confidentiality, integrity, freshness, availability, and authentication.

3.2 Contributions

The main contribution of this paper is to introduce a hierarchical key management scheme which solves the mentioned issues for Y. Zhang et al's work to efficiently enhance the security and survivability for the clustered WSNs. Our proposed scheme enjoys the following properties:

i. ESKM is a dynamically clustering key management scheme based on secret sharing for WSNs.
ii. ESKM is not only substantially improves network resilience against node capture over existing schemes, but also meets the demands of the scalability.
iii. ESKM provides an authentication mechanism to authenticate a new user as well as to isolate the compromised node.
iv. ESKM can solve the security issue in the previous work which provides perfect forward secrecy.
v. Moreover, compared with existing key management schemes, ESKM can achieve better network performance in terms of network size, key storage overhead and communication overhead.
vi. ESKM can adjust the TTL to control and limit the cluster size which balances the computation and storage overhead.

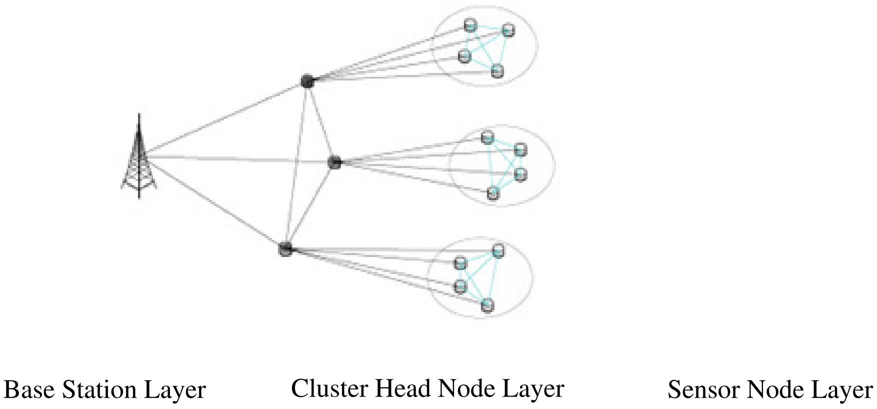
4. System Model

4.1 Network Model

In ESKM scheme, the network model is consists of three different layers which illustrated in **Fig. 1**:

BS: Base station (BS) is responsible to handle communication for all WSNs in both internal and external networks to connect the WSNs with external network. Meanwhile, any sensor node needs to communicate with another node in different networks; the BS is responsible to securely route traffic between them. Since the BS plays as a control center, BS has unlimited computational and communication power, memory storage capacity, and very large radio transmission range to reach all sensors in the network.

Figure 1: Hierarchical wireless sensor network model



CH: Cluster head (CH) is responsible to secure communication between each CH and all members in the same cluster. Also, it is responsible to route data from sensor nodes in the same cluster to the BS.

SN: Sensor node (SN) is responsible for: gathering data and check the validity of them and then transmit these data to the corresponding cluster head.

4.2 Background of Secret Sharing Scheme

Threshold secret sharing schemes were first introduced in 1979 by Shamir [13] and Blakley [14] . In Shamir's scheme the domain of secrets and shares is the elements of a finite field $GF(\varphi)$ for some prime-power $\varphi > n$. Let $x_1, x_2, .., x_n \in GF(\varphi)$, where $(x_i \neq 0, \varphi \text{ is a pig prime})$. To share a secret $S \in GF(\varphi)$, the dealer chooses $t - 1$random elements $a_1, a_2, \ldots.. a_{t-1}$. For $1 \leq i \leq$ and computes

$$f(x_i) = S + a_1 x_i + a_2 x_i^2 + \ldots. + a_{t-1} x_i^{t-1} (\text{mod } \varphi)$$

Then it distributes them to the participants in security with x_i . Hence, the t participants of n can reconstruct the secret S using the Lagrange interpolation polynomial, such as:

$$S = f(0) = \sum_{i=1}^{t} y_i \prod_{\substack{j=1 \\ j \neq i}}^{t} \frac{-x_j}{x_i - x_j} \quad \text{mod } \varphi \qquad (1)$$

Based on the idea of threshold secret sharing scheme in hierarchical WSN, our proposed scheme is provides a high security defense against the node capture attack which is based on secret sharing scheme, that is, it divides the security key in hierarchical wireless sensor network model into n sub keys to guarantee the security of network.

5. Preliminaries

Before we present the proposed scheme, the following assumptions and notations must be considered.

5.1 Assumptions

In the ESKM scheme, the following assumptions must be considered:

1. Each sensor has a unique identification (ID) assigned by the BS.

2. The BS is located in a secured location and it has unlimited computational and memory storage.
3. The BS has suitable intrusion detection system to detect out whether the node is being compromised or not.
4. Authentication mechanism must be considered between the BS and all CHs, each CH and its member nodes and among all sensor nodes in the same cluster in the network.
5. Capabilities in energy, computation, and radio range are equally likely for all sensors.
6. The BS can communicate directly with each sensor if the corresponding CH is compromised.

5.2 Notations the Notations that used in this Paper are Summarized in Table 1

Table 1: Notations

Notation	Description
BS	Base station
TTL	Time to Live
n_i	The ith nonce in the set of nonces N
K^{ℓ}	The initial key shared by all nodes
C_i	Cluster head of i cluster
ID_{C_i}	The identity of C_i
S_{ij}, S_{im}	The jth &mth sensor nodes of the ith cluster respectively.
$ID_{S_{ij}}, ID_{S_{im}}$	The identity of S_{ij} and S_{im} respectively.
$Z^{\ell}_{BS-C_i}$	The master key during session ℓ
$Z^{\ell}_{C_i-S_{ij}}$	The cluster key during session ℓ
$Z^{\ell}_{S_{ij}-S_{im}}$	The sensor key during session ℓ
y^{ℓ}_i	The sub key of C_i during session ℓ
x^{ℓ}_i	The session key of C_i during session ℓ
r^{ℓ}_1	Random number generated in the master key phase during session ℓ
$N^{\ell}_{C_i}$	Nonce of C_i during session ℓ
$L^{\ell}_{C_i}$	Location of C_i during session ℓ
$R^{\ell}_{C_i}$	Random number generated by C_i during session ℓ
N^{ℓ}_{BS}	Nonce of BS during session ℓ
y^{ℓ}_{ij}	The sub key of S_{ij} during session ℓ
x^{ℓ}_{ij}	The session key of S_{ij} during session ℓ
r^{ℓ}_2	Random number generated in the cluster key phase during session ℓ
$N^{\ell}_{S_{ij}}, N^{\ell}_{S_{im}}$	Nonce of S_{ij} and S_{im} respectively during session ℓ
$L^{\ell}_{S_{ij}}, L^{\ell}_{S_{im}}$	Location of $\mathrm{S_{ij}}$ and $\mathrm{S_{im}}$ respectively during session ℓ
$R^{\ell}_{S_{ij}}, R^{\ell}_{S_{im}}$	Random numbers generated by $\mathrm{S_{ij}}$ and $\mathrm{S_{im}}$ respectively during session ℓ
r^{ℓ}_3	Random number generated in the sensor key phase during session ℓ
$E_X(M)$	The symmetric encryption of M message using key X
$h(\ )$	One-way hash function
ℓ	Session period
Msg_D	Message of network pre-distribution phase
Msg_I	Message of network initialization
Msg_C	Message of clustering
Msg_E	Message of key establishment
Msg_MK	Message of master key reconstruction
Msg_CK	Message of cluster key reconstruction
Msg_SK	Message of sensor key reconstruction
Msg_UK	Message of key updating establishment

6. The Proposed Scheme

ESKM scheme consists of four main phases: pre-distribution phase, network initialization phase, key-establishment phase, and key-updating phase.

6.1 Pre-distribution Phase

Prior deployment, each sensor node is initially pre-loaded with: a nonce as a random number, its identity, and the initial key K^{ℓ} shared with the BS during the session period ℓ.

In ESKM, since the CH is responsible to manage all tasks in the cluster such as sending and receiving data from its members, manage the authentication process, and handle the network scalability, it is strongly recommended use Low-Energy Adaptive Clustering Hierarchy (LEACH) [14] to randomly chosen CHs and periodically replaced them to provides since the CH has a large energy consumption and it must be replaced periodically to provides best equilibrium of computation overhead, communication overhead, and energy consumption. Besides to ensure that compromised any cluster head is as much hard as possible by an adversary.

Therefore, a node uses the cluster head election algorithm [15, 16] to decide whether it becomes a CH or not. After the node decides to be a CH, it will announce the candidate information to other nodes.

Each node will choose its neighbor CH based on the cluster size to join it depending on the broadcast message sent by the CH.

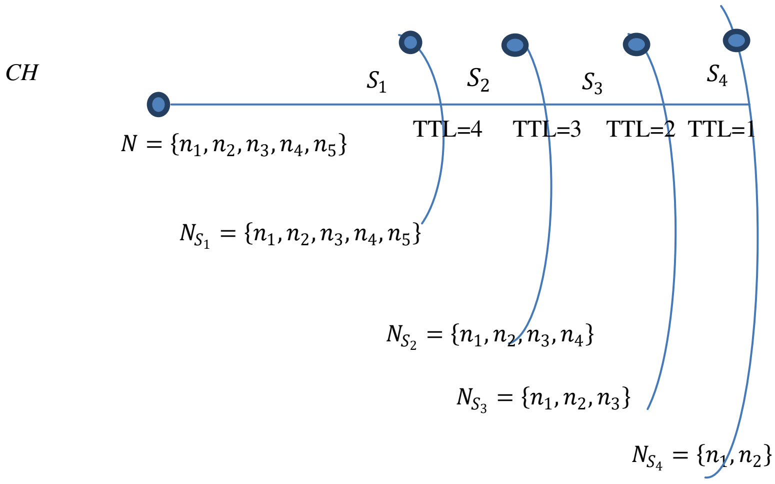
This message contains the unique identification of the CH and the transmission range which is limited by(TTL), that is, calculated by hop count to provide the connectivity; the CH generates a set of nonce named

N which are random numbers. These nonces are based on the TTL, that is, if $\mathrm{TTL} = 4$, then the sensor will get five$(TTL + 1)$nonces such that$\mathrm{N} = \{n_1, n_2, n_3, n_4, n_5\}$.

Obviously, the more nonce are generated, the great connectivity is provided.

Therefore, depending on the cluster size(TTL), other nodes can receive different sets of messages from different CHs as (2) in different distance (hop range) as shown in **Fig. 2**.

Figure 2: The deployment of nonces from the cluster head (TTL=4)


CH
S_1
S_2
S_3
S_4
TTL=4
TTL=3
TTL=2
TTL=1
$N = \{n_1, n_2, n_3, n_4, n_5\}$
$N_{S_1} = \{n_1, n_2, n_3, n_4, n_5\}$
$N_{S_2} = \{n_1, n_2, n_3, n_4\}$
$N_{S_3} = \{n_1, n_2, n_3\}$
$N_{S_4} = \{n_1, n_2\}$

$$N = \{n_i | \, n_1, \dots, n_{TTL+1}\}$$
$$C_i \rightarrow *: E_{K^{\ell}}\left(ID_{C_i} \parallel N \parallel Msg\_D\right) \quad (2)$$

6.2 Network Initialization Phase

After deployment, the BS starts to form the initialization phase as follows.

Step 1: C_i sends an encrypted message to BS informing that it becomes as cluster head.

$C_i \rightarrow BS: E_{K^\ell}\left(ID_{C_i} \parallel Msg\_I\right)$

Step 2: in the same time, all sensor nodes send their identifications to the BS

$S_{ij} \rightarrow BS: E_{K^\ell}\left(ID_{S_{ij}} \parallel Msg\_I\right)$

Step 3: BS will send only the identity list of all neighbor sensor nodes to the corresponding CH based on the cluster size.

$BS \rightarrow C_i\text{: } E_{K^\ell}\left(ID_{S_{ij}} \parallel Msg\_I\right)$

Step 4: $\mathrm{C_i}$ then received the message and decrypt it using the initial key and send it to its members informing them to join the cluster accordingly.

$C_i \rightarrow S_{ij}\text{: } E_{K^\ell}(Msg\_C)$

6.3 Key-establishment Phase

In this phase, we describe how to secure communication between: each sensor node and the corresponding cluster head, between each cluster head and the base station, and among sensor nodes in the same cluster.

Therefore, this phase can be classified into three sub-phases: master-key establishment, cluster-key establishment, and sensor-key establishment.

Firstly, the BS randomly chooses two primes p and . Then, calculate the modulus φ such that $\varphi = 3pq + 2$ and φ is a prime. Hence, the BS broadcast this modulus to all sensors in the network.

$BS \rightarrow *\text{: } E_{K^\ell}(\varphi \parallel Msg\_E)$

6.3.1 Master-key Establishment

In this sub-phase, the ESKM establishes a session key between BS and CHs to protect their communication. The procedure steps of this phase are illustrated in figure 2 and summarized as follows.

Step 1: assume that during each session period $\{\ell\}_{\ell=1,\ldots.,M}$, BS randomly and uniformly chooses m plynomial $f(x)$ and $t-1$ integers $a_1, a_2, \ldots.. a_{t-1}$, where $m - 1 \geq t$. Then, it constructs a $(\mathrm{t}-1)th$ degree polynomial $f(x)$ as follows:

$$f^{\ell}_{BS-C_i}(x) = Z^{\ell}_{BS-C_i} + a^{\ell}_{1,BS-C_i}x + a^{\ell}_{2,BS-C_i}x^2 + \cdots + a^{\ell}_{t-1,BS-C_i}x^{t-1} \qquad mod\ \varphi$$

Where $Z^{\ell}_{BS-C_i}$ is the master key which used to establish a secure communication between BS and all cluster heads.

Step 2: in this step, the authentication process will proceed between the BS and each CH as follows.

BS broadcasts Hello message encrypted by K^ℓ to each CH which contains $E_{K^\ell}\left(ID^{\ell}_{C_i} \parallel N^{\ell}_{BS} \parallel Msg\_MK\ \right), R^{\ell}_{BS}$.

CH chooses r^{ℓ}_1 as a random number and calculates the required verification key $K^{\ell}_{V_1}$ where $K^{\ell}_{V_1} = \left(ID^{\ell}_{C_i} \oplus K^\ell\right) \oplus K^\ell$

When CH receives the Hello message, it decrypts this message using the initial key K^ℓ to get the nessary information of BS and then calculates the following messages and sends them to the BS:

$$C_i \rightarrow BS\text{:} \left\{ \begin{array}{c} E_{K^{\ell}_{V_1}}\left(r^{\ell}_1\right) \\ E_{K^\ell}\left(ID^{\ell}_{C_i} \parallel N^{\ell}_{C_i} \parallel L^{\ell}_{C_i}\right), R^{\ell}_{C_i} \\ H_{C_i} = h^{\ell}\left(R^{\ell}_{C_i} \parallel R^{\ell}_{BS} r^{\ell}_1 \parallel N^{\ell}_{C_i} N^{\ell}_{BS}\right) \end{array} \right\}$$

BS calculates the verification key $K_{V_1}^{\ell} = \left(ID_{C_i}^{\ell} \oplus K^{\ell}\right) \oplus K^{\ell}$, decrypts the received messages and calculates the corresponding hash function:

$$H_{BS} = h^{\ell}\left(R_{C_i}^{\ell} \parallel R_{BS}^{\ell} r_1^{\ell} \parallel N_{C_i}^{\ell} N_{BS}^{\ell}\right)$$

BS then verifies that $H_{BS} = H_{C_i}$. If the verification is achieved, the BS will send an acknowlodgment message to CH indicating that the authentication process is successfully completed.

Step 3: BS computes $y_i^{\ell} = f_{BS-C_i}^{\ell}\left(r_1 + ID_{C_i}^{\ell}\right)$ as a sub key of C_i and sends it to C_i

$BS \to C_i\colon E_{K^{\ell}}\left(y_i^{\ell}\right)$

Step 4: when received message, C_i generates x_i as its session key with other CHs and then deletes the initial key

$x_i^{\ell} = h^{\ell}\left(r_1 + ID_{C_i}^{\ell}\right)$

Step 5: C_i begins to reconstruct the master key using secret sharing scheme as follows:

1) BS chooses randomly$t - 1$ CHs which named as C_j , encrypts the reconstruction message by the initial key and then sends this message to C_j.

$BS \to C_j\colon E_{K^{\ell}}\left(ID_{C_i}^{\ell} \parallel Msg\_R\right), j > i$

2) C_j uses its subkey x_j to encrypt the subkey y_j^{ℓ} and transmit it to C_i concatenated with its identity ID_j^{ℓ}
3) C_i can deduce x_j^{ℓ}, decrypt the message and get y_j^{ℓ}
4) Then, C_j can reconstruct the master key $Z_{BS-C_i}^{\ell}$ according to Lagrange interpolation polynomial, where:

$$Z_{BS-C_i}^{\ell} = \sum_{i=1}^{t} y_i^{\ell} \prod_{\substack{j=1 \\ j \neq i}}^{t} \left(\frac{ID_{C_j}^{\ell}}{ID_{C_j}^{\ell} - ID_{C_i}^{\ell}}\right) mod\ \varphi$$

5) Hence, C_i deletes the initial key and using $Z_{BS-C_i}^{\ell}$ as a master key to establish secure communication with the BS.

6.3.2 Cluster-key Establishment

In this sub-phase, the protocol establishes the cluster-key between CH and its members.

Similar to the master key sub-phase, the cluster key can be generated as follows.

Step 1: cluster head C_i randomly and uniformaly chooses m plynomial $f(x)$ and$t - 1$ integers $a_1, a_2, \ldots\ldots a_{t-1}$, where $m - 1 \geq t$. Then, it constructs a $(t-1)th$ degree polynomial $f(x)$ during each session period $\{\ell\}_{\ell=1,\ldots.,M}$ as follows:

$f_{C_i-S_{ij}}^{L}(x) = Z_{C_i-S_{ij}}^{L} + a_{1,C_i-S_{ij}}^{L} x + a_{2,BC_i-S_{ij}}^{L} x^2 + \cdots + a_{t-1,C_i-S_{ij}}^{L} x^{t-1} mod\varphi$

Where $Z_{C_i-S_{ij}}^{\ell}$is the cluster key which used to secure the communication data between each cluster head and its member nodes.

Step 2: in this step, the authentication process will proceed between C_i and each sensor node in i^{th} cluster as follows.

1) CH broadcast Hello message encrypted by K^{ℓ} to each SN

   $C_i \to S_{ij}\colon E_{K^{\ell}}\left(ID_{C_i}^{\ell} \parallel N_{C_i}^{\ell} \parallel L_{C_i}^{\ell} \parallel Msg\_CK\right), R_{C_i}^{\ell}$.
2) When SN receives the Hello message, it decrypts this message using the initial key K^{ℓ} to get the necessary information of its cluster head.
3) Each SN chooses r_2^{ℓ} as a random number and calculates the required verification ke$K_{V_2}^{\ell}$y where $K_{V_2}^{\ell} = \left(ID_{S_{ij}}^{\ell} ID_{C_i}^{\ell} \oplus K^{\ell}\right) \oplus K^{\ell}$

4) S_{ij}sends the following messages to the corresponding CH:

$$S_{ij} \rightarrow C_i \colon \left\{ \begin{array}{c} E_{K_{V_2}^{\ell}}\left(r_2^{\ell}\right) \\ E_{K^{\ell}}\left(ID_{S_{ij}}^{\ell} \parallel N_{S_{ij}}^{\ell} \parallel L_{S_{ij}}^{\ell}\right), R_{S_{ij}}^{\ell} \\ H_{S_{ij}} = h^{\ell}\left(R_{S_{ij}}^{\ell} \parallel R_{C_i}^{\ell} r_2^{\ell} \parallel N_{S_{ij}}^{\ell} N_{C_i}^{\ell}\right) \end{array} \right\}$$

5) C_i calculates the verification key $K_{V_2}^{\ell} = \left(ID_{S_{ij}}^{\ell} ID_{C_i}^{\ell} \oplus K^{\ell}\right) \oplus K^{\ell}$, decrypts the received messages and calculates the corresponding hash function:

$$H_{C_i} = h^{\ell}\left(R_{S_{ij}}^{\ell} \parallel R_{C_i}^{\ell} r_2^{\ell} \parallel N_{S_{ij}}^{\ell} N_{C_i}^{\ell}\right)$$

6) C_i then verifies that $H_{C_i} = H_{S_{ij}}$. If the verification is achieved, the CH will send an acknowledgment message to S_{ij} indicating that the authentication process is successfully completed.

Step 4: C_i computes $y_{ij}^{\ell} = f_{C_i - S_{ij}}^{\ell}\left(r_2^{\ell} + ID_{S_{ij}}^{\ell}\right)$ as a sub key of S_{ij} and and sends it to S_{ij}

$C_i \rightarrow S_{ij} \colon E_{K^{\ell}}\left(y_{ij}^{\ell}\right)$

Step 5: when received message, S_{ij} generate x_{ij}^{ℓ} as its session key with other SNs and then deletes the initial key

$x_{ij}^{\ell} = h^{\ell}\left(r_2^{\ell} + ID_{S_{ij}}^{\ell}\right)$

Step 6: S_{ij} begins to reconstruct the cluster key using secret sharing scheme as follows:

1) C_i chooses randomly $t - 1$ SNs and sends to them the reconstruction message encrypted by the initial key.

$C_i \rightarrow S_{i,j+1} \colon E_{K^{\ell}}\left(ID_{S_{ij}}^{\ell} \parallel Msg\_R\right)$

2) When $S_{i,j+1}$ received the encrypted message, it uses its subkey $x_{i,j+1}^{\ell}$ to encrypt the subkey $y_{i,j+1}^{\ell}$ and transmit it to S_{ij} together with its identity $ID_{i,j+1}^{\ell}$.

$S_{i,j+1} \rightarrow S_{ij} \colon ID_{i,j+1}^{\ell} \parallel E_{x_{i,j+1}^{\ell}}\left(y_{i,j+1}^{\ell} \parallel Msg\_R\right)$

3) S_{ij} decrypts the receiving message to get $\mathrm{y}_{\mathrm{i,j+1}}^{\ell}$ and then deduce the value of $x_{i,j+1}^{\ell}$.

4) Then, S_{ij} can reconstruct the cluster key $Z_{C_i - S_{ij}}^{\ell}$ according to Lagrange interpolhation polynomial, where:

$$Z_{C_i - S_{ij}}^{\ell} = \sum_{i=1}^{t} y_{ij}^{\ell} \prod_{\substack{m=1 \\ m \neq j}}^{t} \left(\frac{ID_{S_{im}}^{\ell}}{ID_{S_{im}}^{\ell} - ID_{S_{ij}}^{\ell}}\right) mod\varphi$$

5) Hence, S_{ij} deletes the initial key and using $Z_{C_i - S_{ij}}^{\ell}$as a cluster key to establish secure communication with the corresponding C_i.

6.3.3 Sensor-key Establishment

In this sub-phase, if the sensor node S_{ij} wants to exchange information with $S_{i,j+1}$, it must establish secure sensor key between them. The generation of this key is described as follows.

Step 1: S_{ij} sends a Hello message to $S_{i,m}$ which contains its information:

$S_{ij} \rightarrow S_{i,m} \colon E_{K^{\ell}}\left(ID_{S_{ij}}^{\ell} \parallel N_{S_{ij}}^{\ell} \parallel L_{S_{ij}}^{\ell} \parallel Msg\_SK\right), R_{S_{ij}}^{\ell}$

Step 2: $S_{i,m}$ decrypt the received message, chooses random number r_3^{ℓ} and then calculates the verification key $K_{V_3}^{\ell}$

$K_{V_3}^{\ell} = \left(ID_{S_{ij}}^{\ell} ID_{im}^{\ell} \oplus K^{\ell}\right) \oplus K^{\ell}$

Step 3: $S_{i,m}$ sends the following messages to S_{ij}:

$$S_{im} \to S_{ij}: \left\{ \begin{array}{c} E_{K_{V_3}^{\ell}}\left(r_3^{\ell}\right) \\ E_{K^{\ell}}\left(ID_{S_{im}}^{\ell} \parallel N_{S_{im}}^{\ell} \parallel L_{S_{im}}^{\ell}\right), R_{S_{im}}^{\ell} \\ H_{S_{im}} = h^{\ell}\left(R_{S_{im}}^{\ell} \parallel R_{S_{ij}}^{\ell} r_3^{\ell} \parallel N_{S_{im}}^{\ell} N_{S_{ij}}^{\ell}\right) \end{array} \right\}$$

Step 4: node S_{ij} decrypts the received messages, calculates the verification key $K_{V_3}^{\ell} = \left(ID_{S_{ij}}^{\ell} ID_{im}^{\ell} \oplus K^{\ell}\right) \oplus K^{\ell}$, computes $H_{S_{ij}} = h^{\ell}\left(R_{S_{im}}^{\ell} \parallel R_{S_{ij}}^{\ell} r_3^{\ell} \parallel N_{S_{im}}^{\ell} N_{S_{ij}}^{\ell}\right)$, and verifies that $H_{S_{ij}} = H_{S_{im}}$

Step 5: if the verification is achieved, S_{ij} sends an acknowledment message to S_{im} indicating that the authentication process is successfully completed.

Step 6: then, the sensor key between $S_{ij} and\ S_{i,m}$ can be generated as follows:

$Z_{S_{ij},S_{im}}^{\ell} = h^{\ell}\left(r_3^{\ell} + K_{V_3}^{\ell} \oplus K^{\ell}\right)$

So that S_{ij} can securly exchange information with S_{im} using $Z_{S_{ij},S_{im}}^{\ell}$ as follows:

$S_{ij} \to S_{im}: E_{Z_{S_{ij},S_{im}}^{\ell}}[I]$

Where I is the information to be exchanged.

6.4 Scalability and Key Updating Phase

In our solution, we consider the scalability of network as well as key updating mechanism. Therefore, this phase is categorized to dynamic key updating, node addition, node isolation, and CH replacement.

6.4.1 Dynamic Key Updating

When a new member wants to join the network during session period ℓ, the key updating mechanism is establishing to secure communication among all sensor nodes which prevent any adversary to reveal any data.

In ESKM scheme, the updating mechanism is based on changing only the random numbers $r_1^{\ell}, r_2^{\ell}, and\ r_3^{\ell}$ in the corresponding key establishment phases and all keys stored in all nodes can be updated accordingly. The master key update mechanism can be summarized as follows:

Step 1: CH chooses r_{11}^{ℓ} as a new random number and uses the previous master key as the current initial key to calculate the required new verification key$K_{V_{11}}^{\ell}$ using the where $K_{V_{11}}^{\ell} = \left(ID_{C_i}^{\ell} \oplus Z_{BS-C_i}^{\ell}\right) \oplus Z_{BS-C_i}^{\ell}$

Step 2: in this step, the authentication process will proceed between the BS and each CH as follows.

1) CH calculates the following messages and send them to the BS:

$$C_i \to BS: \left\{ \begin{array}{c} E_{K_{V_{11}}^{\ell}}\left(r_{11}^{\ell}\right) \\ E_{Z_{BS-C_i}^{\ell}}\left(ID_{C_i}^{\ell} \parallel N_{C_i}^{\ell} \parallel L_{C_i}^{\ell} \parallel Msg\_UK\right), R_{C_i}^{\ell} \\ H_{C_i} = h^{\ell}\left(R_{C_i}^{\ell} \parallel R_{BS}^{\ell} r_{11}^{\ell} \parallel N_{C_i}^{\ell} N_{BS}^{\ell}\right) \end{array} \right\}$$

2) BS calculates the verification key$K_{V_{11}}^{\ell} = \left(ID_{C_i}^{\ell} \oplus Z_{BS-C_i}^{\ell}\right) \oplus Z_{BS-C_i}^{\ell}$, decrypts the received messages and calculates the corresponding hash function:

$H_{BS} = h^{\ell}\left(R_{C_i}^{\ell} \parallel R_{BS}^{\ell} r_{11}^{\ell} \parallel N_{C_i}^{\ell} N_{BS}^{\ell}\right)$

3) BS then verifies that $H_{BS} = H_{C_i}$. If the verification is achieved, the BS will send an acknowlodgment message to CH indicating that the authentication process is successfully completed.

Step 3: BS computes $y'^{\ell}_{i} = f_{BS-C_i}^{\ell}\left(r_{11}^{\ell} + ID_{C_i}^{\ell}\right)$ as a sub key of C_i and encrypt it using the initial key$Z_{BS-C_i}^{\ell}$and sends it to C_i

$BS \rightarrow C_i: E_{Z^{\ell}_{BS-C_i}}(y'^{\ell}_i)$

Step 4: when received message, C_i generate x'^{ℓ}_i as its session key with other CHs and then deletes the initial key

$x'^{\ell}_i = h^{\ell}(r^{\ell}_{11} + ID^{\ell}_{C_i})$

Step 5:C_i begins to reconstruct the cluster key using secret sharing scheme as follows:

$$Z'^{\ell}_{BS-C_i} = \sum_{i=1}^{t} y'^{\ell}_i \prod_{\substack{j=1 \\ j\neq i}}^{t} \left(\frac{ID^{\ell}_{C_j}}{ID^{\ell}_{C_j} - ID^{\ell}_{C_i}} \right) mod\varphi$$

4) Step 6: Hence, C_i deletes the initial key and using $Z'^{\ell}_{BS-C_i}$ as a master key to establish secure communication with the BS.

By the same manner, we can get the new cluster key $Z'^{\ell}_{C_i-S_{ij}}$ and the new sensor key $Z'^{\ell}_{S_{ij},S_{im}}$by choosing new random numbers r^{ℓ}_{22} and r^{ℓ}_{33} respectively.

6.4.2 Node Addition

When a new member SN_{new} needs to join a certain C_i during session period ℓ, it is initioally preloaded with the initial key K^{ℓ} from the BS and it chooses randomly an ID_{new} and begins the authentication process with the target $C_i$according to cluster-key establishment phase. Then, $SN_{new}$is acceptable and join the network and C_i will update the BS with the new information.

6.4.3 Node Isolation

When a compromised sensor node is detected, the BS will inform the corresponding CH about this captured node. Then, the responsible CH will add the identity of this node into the revoked list. Simultaneously, the CH will inform all its members that the captured SN is invalid. Since the secure communication for other nodes is not affected by the compromised node, the key updating mechanism is not necessary.

6.4.1 CH Replacement

If a certain CH is compromised and detected by the BS, the BS will proceed the following steps:

1) It will inform all sensor nodes including other cluster heads for this failure CH.
2) Add the corresponding identity of the captured cluster into the revoked list.
3) Using LEACH algorithm to select new CH.
4) Trigger the key updating mechanism immediately.

7. Security Analysis and Performance Evaluation

In this section, we prove that the ESKM scheme has strong network resiliency against captured attack as well as it provides strong security against forward secrecy problem.

Then, we compare the performance of our scheme with a low energy key management protocol (LEKM) in [17] and a secret sharing based key management (SSKM) in [12].

7.1 Security Analysis

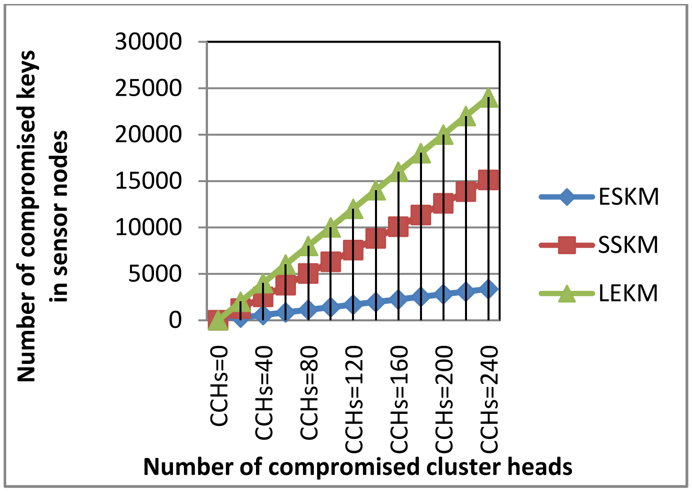
ESKM scheme provides different types of security levels: (1) the BS encrypts the identifications of all sensor nodes with the initial keyK^{ℓ}, only authorized nodes can decrypt this message. (2) The authentication process must be taken, before sending any data, between each BS and all CHs, between each CH and its members and among all SNs in the same cluster to confirming the identity. (3) ESKM scheme allows data to be sent only after encrypt it using a suitable security key which the generation of this key is based on threshold secret sharing scheme. (4) ESKM scheme a secure communication link since it provides a strong one-way hash function which can prevent the attacker from analyzing the information because that, for any given values h it is computationally infeasable to find x such that $H(x) = h$, freshness mechanism using time stamps (TTLs), and nonce. (5) Since ESKM allows each CH and each SN to store only sub keys rather than master or cluster key, the attacker may success to only get the sub keys but he can't be able to reconstruct the master or cluster key as long as he can capture fewer than t CHs. (6) Besides, our scheme has a nice feature which provides the BS to urge the key updating mechanism regularly which prohibit the attacker to reconstruct the master or the cluster keys.

ESKM provides secure cluster formation process and prevents the malicious nodes to join the network.

Fig. 3 illustrates the comparison between our scheme and different schemes (LEKM and SSKM) with respect to the resilience against sensor node capture attack.

From this comparison we can observe that, in LEKM the network resilience against cluster compromising is very poor because that compromising one CH will allow the attacker to penetration of other CHs since in LEKM scheme, all secret keys are pre-loaded in all CHS prior deployment. Once the CH is compromised, the attacker can play a game that, he will a man-in-the middle attacker and he begins to masquerade himself to appear as the existing compromised CH for both the BS and the corresponding sensor nodes. Then, he can able to monitor, alter or inject messages into a communication channel, and then he can easily break the entire network accordingly.

Figure 3: Number of compromised sensor nodes vs. number of the compromised cluster heads in key establishment phase



In SSKM, the network resiliency against node captured still poor since the identity of CH is sends to the BS in clear text during network management phase. Also the BS sends the value of secret parameter g to all sensors in the network. Any attacker can capture the identity of the CH and the value of g; he can get the private key x and easily calculates the session key between the CH and the BS.

Meanwhile, SSKM still suffers from the forward secrecy problem since with the knowledge of the private key ; the attacker can compromise all nodes in the network accordingly.

In ESKM, each CH stores only sub keys rather than the master key. So if any CH is compromised, it won't affect on the security of other CHs because the attacker can't reconstruct that master key.

Clearly, our scheme has not only the best network resilience against node capture attack but also secure against forward secrecy problem.

7.2 Dynamic Analysis

ESKM scheme supports the scalability of network, that is, a new node can be joined to any cluster after generating the shared key with the corresponding CH by itself. Therefore, updating the old keys is not required in this case since the security of that new SN will not affect on the security of other nodes in the same cluster.

On the other hand, when a node is deleted out of the cluster, there is no need to update the key information since each node has a unique key shared with the CH. Just, the CH will inform the BS to remove the identity of the deleted node and then broadcast to other nodes.

Also, our scheme can dynamically update the all secured keys (master, cluster, and sensor keys) by just changing the random number $r_1, r_2, andr_3$, so that the BS hs no use for reselection of all keys and polynomial.

Therefore, our scheme can meet the dynamic requirements of WSNs.

7.3 Performance Evaluation

We are using the network simulator NS-2 [18] to evaluate the performance of the ESKM scheme.
Then, this simulation we consider the following assumptions:

i. a network is consists of 250 sensor nodes
ii. The BS is fixed and located near the sensing and it has unlimited energy,
iii. All the IDs are 16-bits and the shared keys for the ordinary (normal) nodes are 128-bits whereas for cluster head nodes are 48 byte.
iv. The number of chosen CH is 100, each cluster has 100 members inside and the average degree of sensor node is 60.
v. The running time of this Simulation is 10 minutes for every simulation and using 512 bytes for the packet size.

In order to evaluate the performance of the our scheme, we consider three metrics: maximum supported network size, key storage overhead, communication overhead, and energy consumption.

7.3.1 Maximum Supported Network Size

Since WSNs are usually composed of a large number of sensors, when the network size linearly increased, the number of keys stored in each sensor node also linearly increased. But due to memory storage limitation, the maximum supported network size is limited for several key management schemes that based on flat network model. Our proposed ESKM scheme is based on the hierarchical network model which has a better scalability than flat network.

Thus, our scheme can be suitable for any size of WSNs.

7.3.2 Key Storage Overhead

In ESKM scheme, each sensor only needs to store one key (initial key) in its memory regardless the total number of sensor nodes in the network. This technique will increase the memory efficient especially for the large-scale WSNs.

Calculation of the memory overhead is based on the type of node; if the node is normal nodes the storage overhead is the size of only one key $Z^{\ell}_{S_{ij},S_{im}}$. On the other hand, if the node is CH, it needs to store only two keys $Z^{\ell}_{BS-C_i}$, $Z^{\ell}_{C_i-S_{ij}}$, and $Z^{\ell}_{S_{ij},S_{im}}$. Thus, the storage overhead of CH is expressed as:

$$\left\{sizeofZ^{\ell}_{S_{ij},S_{im}} + sizeofZ^{\ell}_{BS-C_i} + sizeofZ^{\ell}_{C_i-S_{ij}}\right\}$$

In practices, we consider that the default key size is 128 bits. Subsequently, the storage overhead for the normal node is 128 bits whereas for a cluster head it 48 bytes which is less than 1 KB.

Table 2 represents a general formula to calculate the storage overhead for all network phases. Where, $e$is the required steps to complete each phase individually,d is the average degree of sensor node, U is the number of the key shares, V is the number of other keys that used during the setup of any phase, and $K_{MP}, K_{CP}, andK_{SP}$ are the size of shared keys (in KB) for master, cluster, and sensor key phases respectively.

Fig. 4 shows the total number of key storage overhead which equal to the summation of storage overhead for all network phases according to a general form which represented in **Table 2**.

We are concluded from this figure that ESKM the storage overhead for the ESKM is very low compared with other previous schemes.

Figure 4: Number of compromised sensor nodes vs. number of the compromised cluster heads in key establishment phase

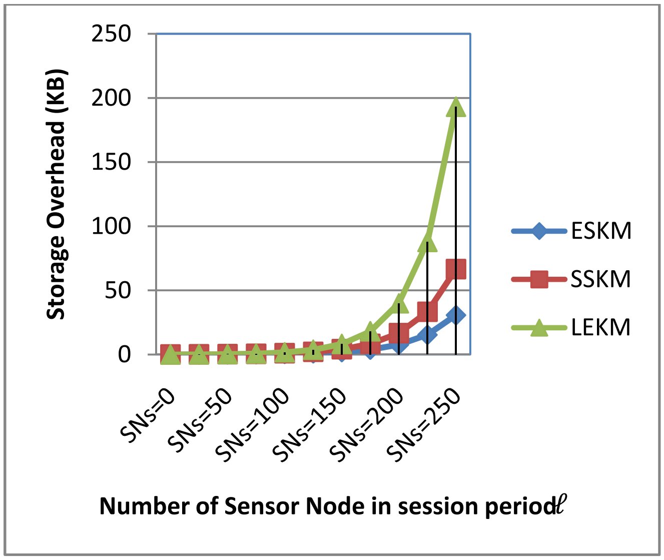


Table 2: General Form of Storage Overhead

Phase type	Formula of storage overhead
Initialization Phase (IP)	$e\left(\frac{d}{N}\right) * C_i * S_{ij}$
Master Key phase (MP)	$K_{MP} * e * C_i * (U * ID_i + V)$
Cluster Key phase (CP)	$K_{CP} * e\left(\frac{d}{N}\right) * S_{ij} * (1 + U * ID_{ij} + V)$
Sensor Key phase (SP)	$K_{SP} * e\left(\frac{d}{N}\right) * S_{ij} * (1 + U * V)$

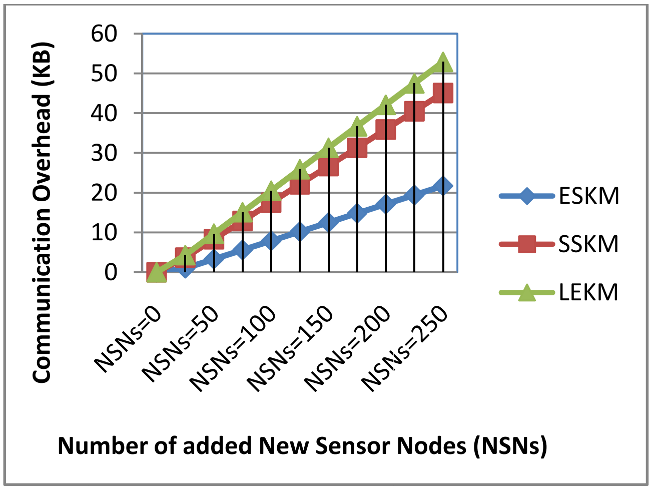
7.3.3 Communication Overhead

Adding new sensor nodes may be required for many applications to replace any failure node. So in ESKM scheme, each sensor only stores one key during the initialization phase which means that the handshake message is much shorter than those previous schemes. Applied to the practice, we can choose the value of t according to the size of WSN. However, In LEKM and SSKM, adding new sensor nodes requires sharing all new key information for those nodes among all existing sensor nodes during the initialization phase, this procedure is consumes a lot of communication overheads, especially for a large-scale network.

Obviously, ESKM has the lowest communication overhead compared with other schemes which is illustrated in **Fig.5.**

Hence, ESKM scheme is the efficient scheme which it reduces the required storage overhead.

Figure 5: Communication overhead vs. Sensor node addition

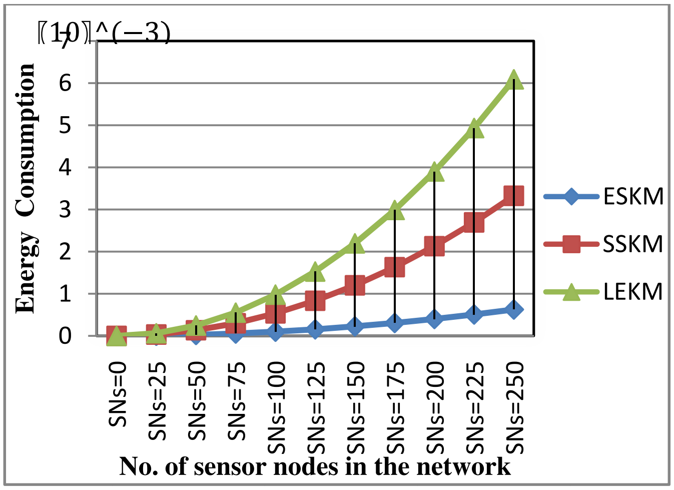


7.3.4 Energy Consumption

We compare the ESKM scheme with LEKM and SSKM with respect to the energy consumption.

In **Fig. 6** we compare the average energyconsumption for those three schemes with respect to different network size.It is observed that, with the increasing of number nodes from 50 nodes to 250 nodes, the energy consumption of both LEKM and SSKM schemes is significantly higher by 1.03% and 1.88% than the proposed scheme respectively because of the communication overhead.

Figure 6: Energy consumption vs. number of nodes



8. Conclusion and Future Work

In this paper, we have pointed out the security leaks of Y. Zhang et al's scheme. Then, we propose a novel secure key management scheme which is depends on the concept of threshold secret sharing scheme to enhance network security and survivability.

Although we employ the hierarchical structure, ESKM scheme provides the cluster size limitation by tuning the TTL to balance the overall energy consumption of the network and enhancing the connectivity of nodes.

In contrast to other clustered architecture security mechanisms, we calculate three security keys (master, cluster and sensor keys) and each of which contains its own sub-keys which derived by Lagrange interpolation formula and then transmitting these keys to each node accordingly.

Also, a periodically key updating mechanism with low energy consumption is presented.

Therefore, the ESKM scheme enjoyed several advantages which are summarized as follows:

1. It provides an authentication mechanism which proves the identity of each new node as well as to segregate any unauthorized node.
2. It provides powerful security resistant against captured attack and forward secrecy attack.
3. It provides scalability on demand.
4. Since each node can generate its shared key by pre-loaded information and it is different from each other, compromising of any node won't reveal the key information of other nodes.

Simulation and security analysis has shown that, ESKM achieved better network resiliency against node capture attack compared with previous key management schemes such as LEKM and SSKM, also it is reduced the energy consumption effectively in terms of computation and communication overheads .

In the future, we will study all possible attacks that may defeat our proposed scheme and show how to enhance security of this scheme to overcome these attacks and then simulate the results.

Acknowledgments

The author would like to thank the anonymous reviewers of European Journal of Scientific Research for their valuable comments.

References

[1] I. F. Akyildiz,W. Su, Y. Sankarasubramaniam, and E. Cayirci*, "A survey on sensor networks,"* IEEE Communications Magazine, vol. 40, no. 8, pp. 102–114, 2002.

[2] Marcos AS, Simpllcio J, Paulo SLM. *"A Survey on Key Management Mechanisms for Distributed Wireless Sensor Networks"*. Computer Networks. 2010; 54: 2591-2612.

[3] Zhang JQ, Varadharajan V. *"Wireless Sensor Network Key Management Survey and Taxonomy"*. Journal of Network and Computer Applications. 2010; 33: 63-75

[4] Y. Zhou, Y. Fang, and Y. Zhang, *"Securing wireless sensor networks: a survey,"* IEEE Communications Surveys and Tutorials, vol. 10, no. 3, pp. 6–28, 2008.

[5] D. W. Carman, P. S. Kruus and B. J. Matt,*"Constraints and Approaches for Distributed Sensor Network Security,"* dated September 1, 2000. NAI Labs Technical Report #00-010, available at http://download.nai.com /products/media/nai/zip/nailabs-report-00-010-_nal.zip

[6] L. Eschenauer and V. Gligor*, "A Key Management Scheme for Distributed Sensor Networks,"* Proceedings of the 9th ACM Conference on Computing and Communication Security, Nov 2002.

[7] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, *"A pairwise key predistribution scheme for wireless sensor networks,"* in Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS), Washington, DC, USA, October 27-31 2003, pp. 42–51.

[8] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, *"Perfectly-secure key distribution for dynamic conferences,"* Lecture Notes in Computer Science, vol. 740, pp. 471–486, 1993.

[9] C. Boyd and A. Mathuria*, "Key establishment protocols for secure mobile communications: A selective survey,"* Lecture Notes in Computer Science, vol. 1438, pp. 344–355, 1998.

[10] M. Bertier, A. Mostefaoui, and G. Tr´edan, *"Low-cost secretsharing in sensor networks,"* in Proceedings of the IEEE 12th International Symposium on High Assurance Systems Engineering (HASE '10), pp. 1–9,November 2010.

[11] T. Claveirole, M. Dias De Amorim, M. Abdalla, and Y. Viniotis, *“Securing wireless sensor networks against aggregator compromises,”* IEEE CommunicationsMagazine, vol. 46, no. 4, pp. 134–141, 2008.

[12] Y. Zhang, C. Wu, J. Cao, and X. Li, *"A Secret Sharing-Based Key Management in Hierarchical Wireless Sensor Network,"* the international Journal of Distributed Sensor Networks, vol. 2013, Article ID 406061, 7 pages, 2013

[13] A. Shamir. *“How to share a secret”,* Communications of the ACM, 1979, 22(11): 612-613.

[14] Heinzelman WR, Chandrakasan A, Balakrishnan H. “*Energy-efficient Communication Protocol for Wireless Microsensor Networks*”. Proceedings of the 33rd Annual Hawaii International Conference on System Sciences. Hawaii. 2000; 2: 33-43.

[15] K. Ramesh and K. Somasundaram, *“A comparative study of clusterhead selection algorithms in wireless sensor networks,”* International Journal of Computer Science & Engineering Survey, vol. 2, no. 4, pp. 153–164, 2011.

[16] J. S. Chen, Z. W. Hong, N. C. Wang, and S. H. Jhuang, *“Efficient cluster head selection methods for wireless sensor networks,”* Journal of Networks, vol. 5, no. 8, pp. 964–970, 2010.

[17] G. Jolly, M. C. Kuscu, P. Kokate, M. Younis, *“A low energy management protocol for wireless sensor networks,”* In Proceeding of the Eighth IEEE International Symposium on Computers and Communication (ISCC’03), KEMER - ANTALYA, TURKEY. June 30 - July 3 2003.

[18] NS-2 web site. [Online] Available: http://www.isi.edu/nsnam/ns.