**Military Technical College
Kobry El-Kobbah,
Cairo, Egypt**

**8th International Conference
on Electrical Engineering
ICEENG 2012**

An Efficient Location based Hybrid and Dynamic Key Management Scheme for SurvSec Security Architecture

By

Mohamed Helmy Megahed* Prof. Dimitrios Makrakis**

Abstract:

Wireless Sensor Networks usually consists of a large number of tiny sensors with limited computation capability, memory space, power resource and communications capabilities. WSN's are extremely vulnerable to numerous attacks; due to several factors such as unattended deployment and lack of tamper resistant packages. Secure communications in wireless sensor networks is critical. Key management is the fundamental security mechanism in WSN. The objective of key management is to establish and maintain secure channels among communicating nodes. Location based key management protocols are very efficient methods in terms of key connectivity, storage overhead, improving the security and scalability and localizing attacks. Also, dynamic key management assumes long lived networks with more frequent addition of new nodes thus requiring network rekeying for sustained security and survivability. In this paper, we proposed a new location based hybrid and dynamic key management scheme that establishes secret keys between sensor nodes. The hybrid scheme reduces the high cost public key operations at the sensor side and replaces them with efficient symmetric key based operations. The proposed system can efficiently resist cloning attack and sybil attack where the attacker can copy the certificate of a node, public key, private key and the node ID to launch these attacks.

Keywords:

Dynamic, Hybrid, Key Management, Cloning attack, Sybil attack, ECC, Location based, Certificate Authority, Symmetric Key, and Public Key.

* Egyptian Armed Forces, mmega080@uottawa.ca
** School of Information Technology and Engineering, University of Ottawa, Ottawa, Canada, dimitris@site.uottawa.ca

1. Introduction:

Wireless sensor networks (WSNs) are the wireless networks that comprise a large number of spatially distributed small autonomous devices cooperatively monitoring environmental conditions and sending the collected data to a command center using wireless channels. This small device, called sensor node, consists of sensor, wireless communication device, small microcontroller and energy source. Wireless sensor network has some unique characteristics such as large scale of deployment, mobility of nodes, node failures, communication failures and dynamic network topology. In addition, each sensor node has constraints on resource such as energy, memory, computation speed and bandwidth because of the constraints on size and cost.

Wireless sensor networks have many applications in both military and civilian such as battlefield surveillance, habitat monitoring, healthcare and traffic control. Many applications of the WSN require secure communications. Wireless sensor networks are prone to different types of malicious attacks, such as impersonating, masquerading, cloning, interception for misleading due to the wireless connectivity, the absence of the physical protection and the unattended deployment. Therefore, the security in sensor network is extremely important and the resource conscious security protocols and management techniques become necessity.

Key management protocols are the core of the secure communications. The goal of the key management is to establish secure links between neighbouring sensors at network formation phase. In wireless sensor network key management, we need to take into account for two phases: initialization and network formation. The network formation has two steps: shared key discovery and key establishment. In shared key discovery, two sensors try to find a common key. The key establishment means two sensors establish common key to secure communications between them. Recently, many key management schemes for the wireless sensor network have been proposed. Some researchers have investigated the wireless sensor networks key management schemes and divided them into three different categories based on the encryption technique, dynamicity and location. The category based on encryption has three classes which are symmetric key based key management; asymmetric key based key management and hybrid key management. The category based on location produced location based key management. The category based on dynamicity has two classes which are static key management and dynamic key management.

Recently, researchers have suggested utilizing the location of sensor nodes after node deployment to improve the security and scalability of key management schemes. Location based key management protocols are very efficient methods in terms of key connectivity and storage overhead. While static key management schemes primarily assume that administrative keys outlive the network and emphasize pairwise communication keys, dynamic key management schemes advocate rekeying and compromised nodes revocation to achieve resilience to attacks in long lived networks.

Motivated by the fact of insufficient hardware resources, a great deal of research has focused on the symmetric cryptography based solutions [1-8] for light-weight computation. These symmetric-key schemes, however, require complicated key management that may cause large memory and communication overhead. This drawback has not yet been investigated by experimental work. Recent progress in implementation of elliptic curve cryptograph on sensors [9-11] proves public key cryptography is now feasible for resource constrained sensors.

In this paper, we proposed a new location based, hybrid and dynamic key management that uses a hybrid key management scheme in order to establish secret keys between nodes and the scheme is based on the nodes location to avoid cloning attack and sybil attack.

1.1 Motivations:

The published symmetric key based key management protocols and public key based key management protocols are vulnerable to sybil attack and cloning attack. In these attacks, the attacker can stole the identity of the sensor then launch impersonation attack to use it elsewhere in the network. Also, the attacker can copy the certificate of the node beside the public and private keys for cloning attack to join the network with legitimate credentials.

Our proposed key management scheme is designed to efficiently resist cloning attack and sybil attack through utilizing sensor nodes locations in location based key management protocol. The proposed scheme is a hybrid key management scheme to incorporate the advantages of both the symmetric and asymmetric key management schemes. Also, the proposed scheme is a dynamic key management which will provide the network with rekeying and revocation of compromised sensor nodes.

2.1 Outline of the Paper:

Section 2 presents the related work. Section 3 describes the network assumptions and threat model. Section 4 describes the proposed location based, hybrid and dynamic key management scheme. Section 5 presents the security analysis of the proposed scheme. Section 6 presents the simulation results. Section 7 presents the performance analysis. Section 8 presents the comparison with other key management protocols. Finally, Section 9 concludes the paper.

3.1 Contributions:

1- We proposed a simple hybrid key management protocol.

2- We designed the protocol such that it is location based key management protocol to localize the attacks and to resist collusion attack, sybil attack and cloning attack.

3- We designed the protocol such that it is dynamic key management protocol to provide rekeying, revocation of compromised sensor nodes and addition of new nodes.

2. Related Work:

In this section, we present the related work to our proposed scheme.

2.1 SurvSec Security Architecture

Surveillance Security (SurvSec) is a new designed security architecture for reliable network recovery from single BS failure of surveillance WSN with single BS [12]. SurvSec relies on a set of sensor nodes serve as Security Managers for management and storage of the security related data of all sensor nodes. SurvSec security architecture provides methodologies for choosing and changing the security managers of the surveillance WSN. SurvSec has three components: (1) Sensor nodes serve as Security Managers, (2) Data Storage System, (3) Data Recovery System.

SurvSec is used for securing the surveillance WSN during the time between the BS failure and the new mobile BS deployment which is the perfect time for attackers to compromise many legitimate nodes then destroy the security of the whole network. Also, SurvSec describes how the new BS will verify the trustworthiness of the deployed WSN otherwise a new WSN must be deployed. Therefore, for mission critical applications such as surveillance WSN, if the BS fails, this problem is addressed through employing the security architecture of SurvSec to detect the BS failure, monitor the network sensitive security issues to store security data in multiple replica, and send the stored data to the new BS after it is authenticated.

Security Managers Network Setup and the Methodology to choose the Security Managers:

1- The base station divides the network into divisions of three layers as shown in Fig. 1.

2- The BS assigns the first layer of security managers as the sensor nodes cluster heads of the first layer sensor nodes as shown in Fig. 1.

3- The BS assigns the next security managers after three layers of the cluster heads and so on.

4- The BS sends to each sensor node its security manager and the BS sends to each security manager its responsibilities of sensor nodes.

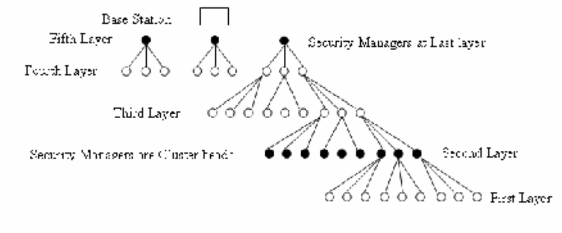

Fig. 1 Security Managers Network Setup

We will employ our designed location based, hybrid and dynamic key management for SurvSec security architecture.

2.2 Static versus Dynamic Key Management

The success of a key management scheme is determined in part by its ability to efficiently survive attacks on highly vulnerable and resource challenged sensor networks. Key management schemes in sensor networks can be classified broadly into dynamic or static solutions based on whether rekeying (update) of administrative keys is enabled post network deployment.

2.2.1 Static Key Management Scheme

These schemes assume that once administrative keys are predeployed in the nodes, they will not be changed. Administrative keys are generated prior to deployment, assigned to nodes either randomly or based on some deployment information, and then distributed to nodes. Most static schemes use the overlapping of administrative keys to determine the eligibility of neighbouring nodes to generate a direct pair-wise communication key.

The basic key predistribution scheme was first proposed by Eschenauer and Gligor [13]. It assumes homogeneous nodes that are loaded with keying material and perform the same key management functions. In this scheme *k* keys are randomly selected by each node out of a large pool of *P* keys. A major advantage of such scheme is the exclusion of the base station in key management. Another advantage is incurring no post-deployment communication overhead on sensor nodes. However, successive node captures enable the attacker to reveal keys stored in captured nodes and use them to attack other nodes.

An enhancement of the basic scheme was proposed in [14], in which two nodes can establish a link only if they share *q* keys. Liu and Ning [15, 16] provided further enhancements by using *t*-degree bivariate key polynomials. Instead of selecting *k* keys out of a pool of *P* simple keys for each node as in the basic Eschenauer and Gligor scheme [13], a key server first randomly generates a pool of *P* bivariate *t*-degree polynomials, each of which is uniquely identified by a polynomial ID. The server then chooses a random subset of polynomials and distributes the polynomial shares and polynomial IDs to the sensor nodes. Two nodes can directly communicate only if they can identify at least one polynomial in common by exchanging their polynomial IDs, and using the polynomial-based scheme to compute the pair-wise communication key.

In [16], the authors assume that nodes are deployed in groups; each group might represent a deployment event to a certain location in the deployment field. Individual nodes are assumed to be aware of their group prior to deployment.

2.2.2 Dynamic Key Management Scheme

Basically, dynamic key management schemes change administrative keys periodically, or on demand or on detection of node capture. The major advantage of dynamic keying is enhanced network survivability, since any captured key(s) is replaced in a timely manner in a process known as rekeying.

Another advantage of dynamic keying is providing better support for network expansion; upon adding new nodes, unlike static keying, which uses a fixed pool of keys, the probability of network capture does not necessarily increase. Both homogeneous and heterogeneous dynamic key management schemes have been proposed in the literature.

The major challenge in dynamic keying is to design a secure yet efficient rekeying mechanism. A proposed solution to this problem is using exclusion-based systems (EBSs); a combinatorial formulation of the group key management problem developed in [17].

Rekeying takes place either periodically or when one or more nodes are captured (or suspected of being captured). A drawback of the basic EBS-based solution is that a small number of nodes may collude and collectively reveal all the network keys.

The application of EBS was first proposed for key management in sensor networks in [18]. In this scheme, nodes were assumed anonymous (with no preloaded node ID). The sensor network establishes a coordinate system (or virtual infrastructure) around the base station.

An example of non-EBS dynamic keying schemes is due to Jolly et al. [19] who proposed a key management scheme based on identity based symmetric keying. The network model involves a base station and several clusters of sensor nodes, each led by a (better equipped) cluster gateway. Rekeying involves reestablishment of clusters and redistribution of keys.

Although the storage requirement is very affordable, the rekeying procedure is inefficient due to the large number of messages exchanged for key renewals. In addition, they require a centralized key server to play a major role in key management. Since the network model involves three types of nodes: sensor nodes, cluster gateways, and base station with different keying functionalities, this scheme is classified as heterogeneous where no node location or other deployment information is used in key assignment.

In order to address the collusion problem in EBS, Younis et al. proposed SHELL [20]; an EBS-based scheme that performs location based key assignment to minimize the number of keys revealed by capturing collusion nodes.

2.3 Key Management based on Encryption Key

2.3.1 Symmetric key based key management scheme

Symmetric-key based schemes are widely used because these schemes consume less computation time than other schemes, which are suitable for the limited resource characteristics of the wireless sensor network. However, the shortages of the symmetric key schemes are also obvious. Different schemes may have different weakness such as security strength (resilience), scalability and connection probability (connectivity). Based on the key distribution, key discovery and key establishment in the schemes, we can divided these schemes into eight categories: entity based key management schemes [21], pairwise key pre-distribution schemes [2], pure probabilistic-based schemes [13], polynomial-based key pre-distribution schemes [4], matrix- based key pre-distribution schemes [5], tree-based key pre-distribution schemes [6], combinatorial design-based key pre-distribution schemes [7] and exclusion basis systems EBS-based key pre-distribution schemes [8, 17].

The symmetric-key based key management schemes, however, require complicated key management that may result in large memory and communication overhead. Also, symmetric key based key management schemes are susceptible to man-in-the middle attack, collusion attack, cloning attack and sybil attack.

2.3.2 Asymmetric key based key management scheme

The public key based key management schemes have many advantages such as low communications overhead, low storage overhead, high scalability. It can provide simpler solution with much stronger security strength. Public key solutions were thought to be computationally expensive for wireless sensor network. However, some researchers [22] show that public key schemes are viable on sensor node.

Public key based schemes have been categorized into three types: RSA-based asymmetric encryption system, ECC-based asymmetric encryption system and ID-based key agreement schemes. In general, public key schemes have better security strength, scalability and connectivity. The shortage is the computation overhead.

RSA and elliptic curve cryptography (ECC) are two major public key techniques. Public key technology is widely used in the realm of Internet. On the other hand, some researchers believe that these techniques are too heavyweight for sensor network because of its constrains. However, several research groups (Gura et al. [9]; Watro et al. [23]; Karlofand et al. [24]; Gaubatz et al. [25]) have successfully implemented public-key cryptography in wireless sensor networks. Gura et al. [9] compared the ECC and RSA on small devices. They show that both RSA and elliptic curve cryptography public key cryptography is viable on 8-bit CPU. The relative performance advantage of ECC point multiplication over RSA modular exponentiation increases with the decrease in processor word size and the increase in key size. They also demonstrate that ECC-160 point multiplication outperforms the RSA-1024 private-key operation by an order of magnitude and is within a factor of 2 of the RSA-1024 public-key operation. The asymmetric key based key management protocols require higher computations than symmetric key based key management schemes. Also, asymmetric key based key management schemes are susceptible to cloning attack and sybil attack.

2.3.3 Hybrid schemes

Several research groups (Huang et al. [11]; Zhang and Varadharajan [26]) proposed the hybrid key establishment schemes for wireless sensor networks. The motivation is to exploit the difference among the base station, the cluster heads and the sensor, and place the cryptographic burden on the base station or the sensors where the resources are less constrained. Sensors are more computational power and energy resources limited. On the other hand, the base station has much more computational power and other resources. The hybrid key establishment schemes reduce the high computational cost on the sensors by placing them on the base station side. Huang et al. [11] proposed a hybrid authenticated key establishment scheme, which is based on a combination of elliptic curve cryptography (ECC) and symmetric-key operations. The hybrid key establishment protocol reduces the high cost elliptic curve random point scalar multiplications at the sensor side and replaces them with low cost and efficient symmetric-key based operations. Moreover, it authenticates the two identities based on elliptic curve implicit certificates to avoid the typical key management problem in pure symmetric-key based protocols.

Hybrid schemes are suitable for the larger hierarchical wireless sensor network. Hybrid schemes may have advantages of both asymmetric key and symmetric schemes for larger sensor network. The public key based key management schemes will make strong security and become a reality with more research work in the future. The ongoing direction is how to secure the wireless sensor network by combining the cryptographic techniques to provide the best solution for the different environmental.

2.4 Key Management based on Location

Liu et al. propose in [27] LBKs (location-based keys) that relies on location information to achieve key management. The keys are established according to the geographical location of sensor nodes. However, knowing the geographical location of nodes is not guaranteed with random deployment.

Recently researchers have suggested utilizing the location of sensor nodes [28- 32] after node deployment to improve the security and scalability of key management schemes. Location based key management protocols are very efficient methods in terms of key connectivity and storage overhead. Location-aware key management is resilient against node capture attacks in large-scale sensor networks.

3. Network Assumptions and Threat Model:

In this section, we formulate the network assumptions and the attack model.

3.1 Network Assumptions

We consider a wireless sensor network consisting of a base station, many cluster heads and numerous sensor nodes which are grouped into clusters, and each node has a unique ID. Each node has a unique location. Each cluster is controlled by a cluster head, which can broadcast messages to all sensors in the cluster. The network architecture is depicted in Fig. 1

The assumptions of this model are as follows:

1- We assume that sensors are static, so once they are deployed they do not leave their locations.

2- Some nodes continuously store the detected security threats and all other security data related to sensor nodes where these nodes are named security managers. The security managers store the nodes ID and location underneath.

3- We assume that the goal of the adversary is to uncover the keys used in the system in order to compromise the network.

4- We assume that our key management scheme is supported by secure routing protocol such as SAODV [40] runs with the key management process.

3.2 Attack Model

In this paper, we mainly consider an adversary that tries to uncover the keys of the network and manipulate the system through capturing and compromising some network nodes. No trust assumptions are made on the sensors. When sensors are captured; their memory can be read and erased or tampered with. The cluster heads are not assumed to be tamper proof either. Cluster heads compromise attack includes the uncovering of its keys through collude. Also, the attacker can launch collusion attack, cloning attack and sybil attack.

4. Proposed Scheme:

The proposed scheme has five phases which are key predistribution phase, key establishment phase, key revocation phase, rekeying phase and add new node phase. We cannot implement group key management protocol in our security architecture because our proposed security architecture is designed for hostile environment and every node must have its own key.

4.1 Key Predistribution Phase:

The key predistribution phase consists of acquiring the sensors certificate from the certificate authority CA and determining the sensors locations.

4.1.1 Acquiring the certificate:

The proposed scheme has two types of sensor nodes: security managers and sensors where we put the cryptographic burden where the resources are less constrained. The hybrid scheme reduces the high cost of public-key operations at the sensor side and replaces them with efficient symmetric-key based operations. Meanwhile, the scheme authenticates the two sensors based on public-key certificates to avoid the typical key management problem in pure symmetric-key based protocols and maintains a good amount of scalability. ECC is used in this protocol to perform security functions on sensors with limited computing resources. Compared with other public key crypto algorithms, much smaller key lengths are required with ECC to provide a desired level of security, which means faster processing speed, smaller communication complexity, and smaller key storage requirements.

To prevent the impersonation attack, this protocol uses certificates in the key-establishment protocol, which provide a mechanism to check cryptographically to whom the public key belongs and if the sensor is a legitimate member of a particular network. The use of a trusted interface to pre-establish a certificate and root key in a sensor thwarts both active and passive attacks in subsequent key establishment protocols. The certificates are acquired before each sensor joins the network.

The protocol uses the elliptic curve implicit certificate scheme [33], because of the resulting low communication complexity, which is a dominant factor for low bit transmission channels in sensor networks.

The certificate generation processes for sensor *U* and security manager *V* are performed offline before they join the network.

1- First, an elliptic curve E defined over *GF(p)* (where *p* is the characteristic of the base field) with suitable coefficients and a base point *P* of large order *n* is selected and made public to all users.
2- CA selects a random integer q_{CA} as its static private key, and computes the static public key $Q_{CA}= q_{CA} X P$.
3- To obtain a certificate and the static private-public key pair, the sensor U randomly selects a temporary key pair (g_U , G_U) and sends G_U to CA via a secure out-of-band interface.
4- CA verifies U's identity and the authenticity of the request received from U. CA also selects a temporary key pair (g_{CA} , G_{CA}) and computes the elliptic curve point $B_U = G_U +G_{CA}$.
5- The implicit certificate IC_U for U is constructed as the concatenation of CA's static public key Q_{CA}, the device identity ID_U, the elliptic curve point B_U and the certification expiration date t_U , i.e., the certificate is the following (Q_{CA}, ID_U, B_U, t_U).
6- CA then applies a one-way hash function *H* on IC_U and derives an integer e_U from the $H(IC_U)$.
7- Finally, CA computes U's private-key reconstruction data $s_U = g_{CA}\, e_U + q_{CA}$ (mod *n*), then CA computes U's public key $Q_U = e_U\, B_U + Q_{CA}$, and sends s_U and IC_U back to U.
8- After U receives the certificate from CA, it computes the hash value $H(IC_U)$ and derives an integer e_U from $H(IC_U)$. U also computes its static private key $q_U = s_U + g_U\, e_U$ (mod *n*) and its public key $Q_U = q_U$ X *P*.
9- U then reconstructs the public key $\hat{O}_U = e_U\, B_U + Q_{CA}$. If $\hat{O}_U = Q_U$, U accepts the certificate and outputs the static key pair (q_U , Q_U); otherwise it rejects the certificate.
10- By repeating the same process, the security manager V acquires its certificate IC_V and static key pair which is (q_V , Q_V).

4.1.2 Determining the sensors locations:

Nodal location information has played an important role in many sensor network applications, including target tracking, geographic routing, and location based key management.

Previous works of location based key management is used with symmetric key based key management schemes where the deployment field is divided into cells and each cell is assigned a key pool to localize the area of attacks.

A number of localization algorithms have been reported. Different researchers have different strategies to categorize them with various criteria. In general, those strategies can be divided into direct and indirect localization, centralized localization methods and distributed localization methods, range-based localization methods and range-free localization methods, absolute localization methods and relative localization methods.

We propose to get the location information from the followings approach:

The indirect approaches of localization were introduced to overcome some of the drawbacks of the GPS-based direct localization techniques while retaining some of its advantages, like accuracy of localization. In this approach, a small subset of nodes in the network, called the beacon nodes [34], are either equipped with GPS receivers to compute their location or are manually configured with their location. These beacon nodes then send beams of signals providing their location to all sensor nodes in their vicinity that don't have a GPS receiver. Using the transmitted signal containing the location information, sensor nodes compute their location. This approach effectively reduces the overhead introduced by the GPS-based method.

At the end of this phase each node will have its location in two parts *x* position and *y* position.

4.2 Key Establishment Phase:

When sensor node and security manager (SM) first communicate to each other, they execute our hybrid key establishment protocol as follows: At the beginning of this phase, the security manager will send to the base station the IDs and locations of its sensor nodes underneath to prevent collusion attack, cloning attack and sybil attack. If the sensor node ID is replicated, it is revoked. Also, if the sensor node location is replicated, it is revoked. This ensures low communication overhead before the key establishment process starts for cloned nodes.

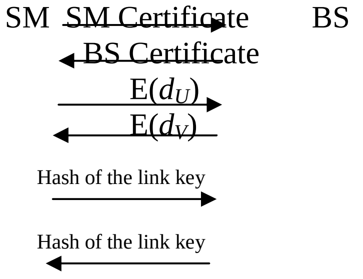

Fig. 2 Communications between Security Manager and BS

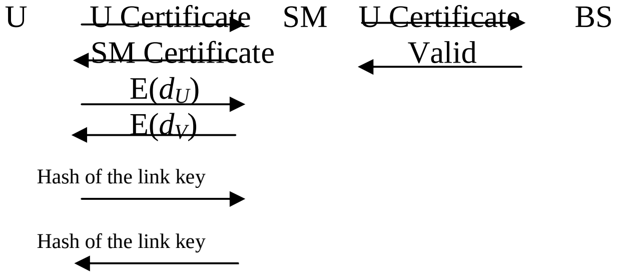

Fig. 3 Communications between Security Manager and Sensor

1. Each security managers establish a link key with the base station in six messages as shown in Fig. 2 where all nodes have the public key of the base station.
2. After the security managers establish link keys with the base station, they communicate with their nodes underneath to share link keys with them. Security manager needs eight messages to share key with a node as shown in Fig. 3.
3. The sensor node U and the security manager V send to each other their implicit certificates. IC_U

and IC_V where certificates are simply the certificate authority public key Q_{CA} together with the device ID and certification expiration date signed by certificate authority CA. U certificate is the following (Q_{CA}, ID_U, B_U, t_U). V certificate is the following (Q_{CA}, ID_V, B_V, t_V). The content of the certificate is verified at the other side, including the device identity, the validity period and the sensor location. If any check fails, the protocol is terminated.

4. V computes the hash value H (IC_U) and derives an integer e_U from H (IC_U). V then obtains U's public key $Q_U = e_U\ B_U + Q_{CA}$. After performing the certificate processing operation, V can conclude that Q_U is genuine, provided that U later evidences knowledge of the corresponding private key q_U.
5. U computes the hash value H (IC_V) and derives an integer e_V from H (IC_V). U then obtains V's public key $Q_V = e_V\ B_V + Q_{CA}$. After performing the certificate processing operation, U can conclude that Q_V is genuine, provided that V later evidences knowledge of the corresponding private key q_V.
6. U selects a *k*-bit random number c_U of 160 bits to produce its link key contribution then U gets L which is the square root of its square of x position added to square of y position. U calculates the value of $d_U = H(c_U \| L \| ID_U)$ where H is a cryptographic hash function to map a binary string to a random integer.
7. U encrypts d_U with V public key Q_V where $Q_V = P$ X q_V. To encrypt and send a message d_U to V, U chooses a random positive integer x and produces the ciphertext Cm consisting of the pair of points [35]:
   $Cm = (x\,P,\ d_U + x\,Q_V)$.
8. V decrypts the received message and obtains d_U. To decrypt the ciphertext, V multiplies the first point in the pair by V's private key and subtracts the result from the second point:
   $d_U + x\,Q_V - q_V\,(x\,P) = d_U + x\,(q_V\,P) - q_V\,(x\,P) = d_U$.
9. V selects a *k*-bit random number c_V of 160 bits to produce its link key contribution then U gets L which is the square root of its square of x position added to square of y position. V calculates the value of $d_V = H(c_V \| L \| ID_V)$ where H is a cryptographic hash function to map a binary string to a random integer.
10. V encrypts d_V using symmetric key encryption under key d_U, generating the value $y = E_{du}$ ($ID_v \|$ d_V). V sends y to U.
11. V generates the link key with U by calculating K = H($d_u \| ID_U \| d_V \| ID_V$).
12. U decrypts the received message y using symmetric key encryption under key d_U to obtain the value d_V.
13. U generates the link key with V by calculating K = H($d_u \| ID_U \| d_V \| ID_V$).
14. V calculates z = H(K) and sends z to U.
15. U verifies z by calculating z' = H(K) and checks if $z = z'$. If yes, the link key is established correctly. Otherwise, the protocol is terminated. Note that proper encryption mode needs to be used, such as the Cipher Block Chaining (CBC) mode, where the results of encrypting previous blocks affect the encryption of the current block. This ensures high security level.

4.3 Key Revocation Phase:

The first component of our dynamic based key management scheme is the keys revocation of the compromised sensor nodes. We assume that SurvSec security architecture has a compromised nodes detection algorithm to be able to detect compromised nodes.

When a sensor node is compromised by an adversary, all the session keys used by this sensor node will be revoked. Assume that the compromised node is detected by some scheme and it is reported to the security manager. The security manager will broadcast a revocation message containing the identification and location of the compromised node to all the nodes underneath. This message is sent also to the Base station. A digital signature (denoted as sign) is computed over the message by utilizing the ECDSA at [36] and security manager's private key. Once receiving a revocation message, a sensor node checks whether it communicates with the compromised node or not. If so, the sensor node revokes the session keys shared between them. Since each sensor node knows security manager's public key, when a sensor node receives the revocation message, it can check the integrity of the message by verifying the digital signature. This prevents an adversary from sending a fake revocation message.

When a node is failed, it must be excluded from the network. If it is an ordinary node, the only thing to be done is that the security manager revokes its keys and sends to the base station the identity and location of the failed node. If the failed node is a security manager or if the compromised node is a security manager, the network needs reclustering to choose another security manager. Also, if the compromised node or the fault node is a cluster head, the network needs reclustering to choose another cluster head. The new security manager needs to exchange keys with the sensor nodes underneath through rekeying process. The new security manager will send its public key to other sensor nodes underneath for signature verification and key exchange.

4.4 Rekeying Phase:

The second component of our dynamic based key management scheme is rekeying after compromised nodes detection or fault nodes detection or rekeying can be done periodically. Rekeying is used when the security manager is compromised where we apply reclustering to choose another security manager then the base station informs the new security manager with its nodes underneath. This new security manager share a key with the base station then the new security manager sends its certificate to its nodes underneath to share link keys with them.

There can be two kinds of re-keying, scheduled and unscheduled. Unscheduled re-keying is required whenever the node is compromised and needs to be revoked. Scheduled re-keying might be performed at regular intervals. It is usually required to enhance the resilience of the network by keeping the keys fresh.

First of all, the security manager sends to BS the IDs and location of its downstream sensor nodes to discover collusion attack, cloning attack and sybil attack.

4.5 Add New Node Phase:

When a new node is to join the network, it needs to report to the base station its physical location and its identity through the nearest security manager to prevent the collusion attack, cloning attack and sybil attack. The new sensor node tries to find its nearest security manager by broadcasting a Hello message contains the new node certificate.

To support the addition of new nodes, the security manager sends to the base station the new node ID and location and if it is genuine the key establishment process continues. If the ID is faked or it is copied, the base station sends to the security manager to revoke this new node. Also, if the node location is replicated, the base station sends to the security manager to revoke this new node.

After the base station approves the joining of the new node, the new node starts to communicate with the security manager to establish a link key as shown in the key establishment phase.

5. Security Analysis:

The security analysis of our proposed protocol focuses on the resilience to node compromising attack.

5.1 Compromised Node Attack

Node captures in hostile environments is inevitable. An effective key management scheme should be able to recover from such attacks to be effective. We describe some of the inherent security advantages of utilizing our proposed key management scheme. Then, using the threats identified in section 3, we analyze how well our proposed scheme recovers from those attacks. A clustered and hierarchical framework for WSN with security managers applying distributed security provides many beneficial security properties. Isolation is the primary benefit of clustered key management scheme. Security managers are responsible for distributing and establishing link keys. Therefore, an attack such as compromised node attack that reveals keys of sensor nodes within one cluster will not impact any other cluster in the network. We assume that SurvSec security architecture has compromised node detection algorithm to detect compromised nodes. We assume that security managers are reported with the compromised sensor nodes underneath.

Node compromising attack refers to the capability of an attacker to inject cloned nodes or false IDs in the network using the key materials it gets from the compromised nodes.

5.2 Collusion Attack

Two nodes can collude when they share their keys with each other. In other words, colluding nodes would grow their knowledge about the network security measures. Our designed protocol is resistant to collusion attack. Since each sensor node communicates with a security manager therefore; compromised nodes need to discover themselves to launch the collusion attack. Each compromised sensor node will only reveal its link key with the security manager plus its public and private key. Therefore, it is conceivable that when the compromised sensor nodes collude they will only reveal their keys but this collusion attack will not result in capturing the network. If the compromised sensor node changes its location for launching collusion attack, it will be discovered and revoked. From such a scenario, the adversary is incapable of revealing all encrypted communications in the network. The main idea of our proposed scheme is the location based key management where every node report its ID and location before it join the network to prevent collusion attack, cloning attack and sybil attack.

5.3 Sybil Attack

In the Sybil attack, a malicious node behaves as if it were a larger number of nodes, for example by impersonating other nodes or simply by claiming false identities.

In our protocol, it is clear that if an attacker injects false IDs for new sensor nodes, the new nodes with false IDs will be discovered as they do not have a certificate to join the network. Also, if the new nodes have a copied certificate, they will be discovered when the new nodes register their ID and location at the base station during key establishment phase. In our protocol, it is clear that if an attacker compromises number of nodes and copy their memories for sybil attack, the new nodes with the compromised sensor nodes IDs and certificates will be discovered when they are injected in the network. This is done because the new nodes register their ID and location at the base station during key establishment phase.

In addition, an attacker cannot use the compromised keys to discover the communication of other sensor nodes.

As a conclusion, our protocol is resilient to the injection of false nodes with non-existing identities or copied identities in the network.

5.4 Cloning Attack

One important physical attack is the introduction of cloned nodes into the network. It is easy for an adversary to capture legitimate nodes, make clones by copying them, and integrate these clones back into the network. The clones may even be selectively reprogrammed to subvert the network.

In our protocol, if an attacker deploys cloned nodes, and tries to convince his neighbours of the validity of the clones, shortly the cloned nodes will be discovered. When the cloned nodes join the network, they will communicate with the security manager, cloned nodes can present a valid certificate to the security manager. When the security manager registers the new cloned nodes at the base station, the base station will discover that these cloned nodes have a copied ID and replicated location. Therefore, the base station will revoke the keys of these cloned nodes.

As a consequence, an attacker compromising sensor nodes cannot use them to launch cloned nodes attack.

In addition, the mechanism of key establishment described in section 4 guarantees that all nodes of the network will be registered at the base station to prevent cloning attack and to discover all cloned nodes.

As a conclusion, our protocol is resilient to the injection of cloned nodes in the network.

6. Simulation Results:

6.1 Simulation Environment:

We built a model for the proposed design and we implemented a simulator in MATLAB that can scale to thousand of nodes. In this simulator, sensors can send and receive data from each other's. The simulation verifies the correctness and the feasibility of our security architecture. It is our future work to implement SurvSec in some sensor network testbeds with all its ingredients. Our simulation scenarios include *N* nodes distributed randomly. We choose *N* as 6561 sensor nodes.

The followings are the built models for simulation:

1- Network setup model for the security managers.
2- Attackers' model.
3- Key establishment five phases.

In the simulations, these parameters are given as follows:

1- The number of sensor nodes *N* is varied from 3^5= 243 to 3^8 =6561 sensor nodes where each cluster is three nodes and 3^5 represent five layers network architecture and 3^8 represent eight layers network architecture.
2- The simulation is done for security managers every two layers and then for security managers every three layers.

6.2 Simulation Results:

In this section, we evaluate the communication overhead and the storage overhead under different *N* for security managers every two layers or every three layers.

The communication overhead for security manager to exchange a key with the base station is six. The communication overhead for a sensor node to establish a key with the security manager is eight.

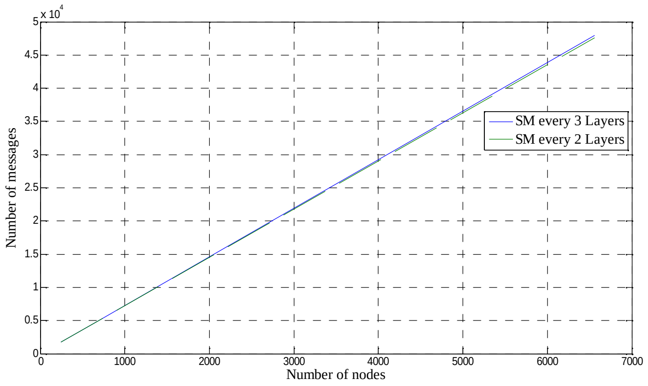

Fig. 4 Communication Overhead

Total communication overhead Com = 8 (N-N_{SEC}) + 6 N_{SEC}, Com = 8N-2N_{SEC} where *N* is number of nodes and N_{SEC} is number of security managers. Fig. 4 shows that communication overhead for security managers every three layers is slightly higher than communication overhead for security managers every two layers.

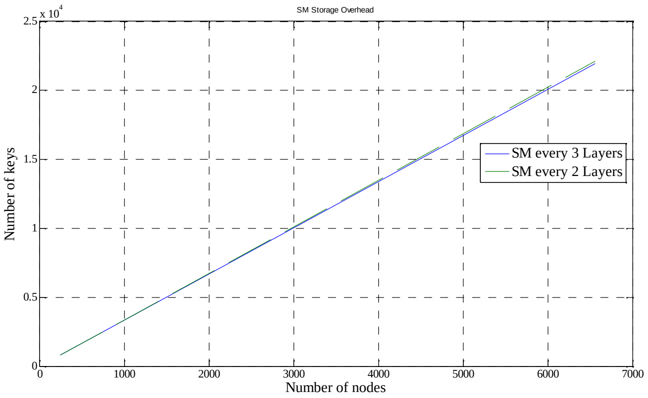

Fig. 5 Security Managers Storage Overhead

Each security manager stores all the public keys of its nodes underneath, links keys with its nodes underneath, its public key, its private key and link key with the base station.

Security managers storage overhead SM_S = (2N_S+3) N_{SEC}.

Where N_S is the number of nodes under each security manager and N_{SEC} is the number of security managers. Fig. 5 shows that storage overhead of the security managers for security managers every two layers is slightly higher than storage overhead of the security managers for security managers every three layers. A security manager every three layers is responsible for twelve nodes and it stores twenty seven keys. A security manager every two layers is responsible for three nodes and it stores nine keys.

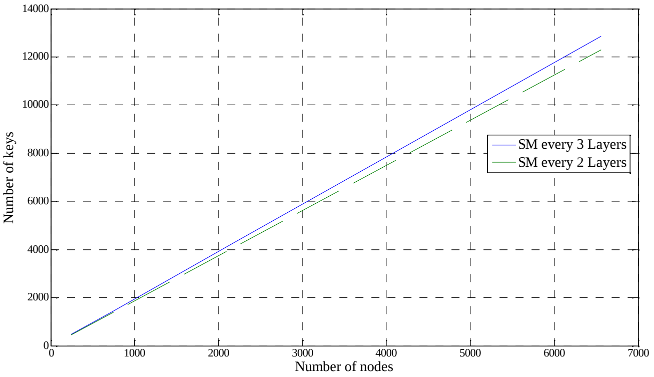

Fig. 6 Sensors Storage Overhead

Each sensor node stores its public key, its private key and link key with its security manager.

Sensor nodes storage overhead S_S = 3(N - N_{SEC}).

Where N is the number of nodes in the network and N_{SEC} is the number of security managers. Fig. 6 shows that storage overhead of sensor nodes for security managers every three layers is slightly higher than storage overhead of sensor nodes for security managers every two layers.

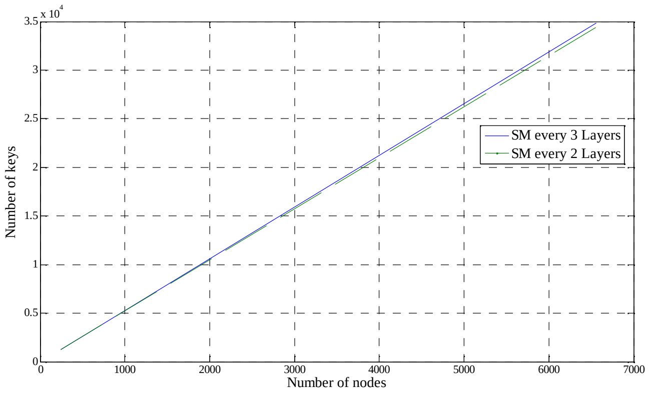

Fig. 7 Total Storage Overhead

Fig. 7 shows that storage overhead for security managers every 3 layers is slightly higher than storage overhead for security managers every 2 layers.

We found that it is better to divide the network in to two layers where security managers will exist every two layers to have better communication overhead and storage overhead.

7. Performance Analysis:

The performance analysis is measured in six folds which are the followings:
Computation complexity, communication complexity and storage complexity are used to measure the efficiency of security scheme. Also, scalability, resiliency and connectivity can be used for measuring the performance of the security schemes.

7.1 Computation Complexity

Computation complexity is the number of unit functions executed. Each sensor node computes the public key of the security manager. The security manager computes the public key of each sensor node underneath. Each sensor node performs 4 times hash to generate the link key with the security manager. Each security manager performs 4 times hash to generate the link key with one sensor node. The sensor node encrypts its part of the link key with the security manager’s public key and its private key using ECC with 160 bit EC scalar multiplication and addition. Also, the security manager decrypts the received message from the sensor node with its private key and the node public key using ECC.
The security manager encrypts its part of the link key using symmetric key under the key from the sensor node. The sensor node decrypts the message from the security manager and uses the other part of the key to generate the link key with the security manager. Also, rekeying after compromised security manager will incur computation overhead as described above. Furthermore, compromised node revocation incurs a computation complexity to sign the revocation message at security manager and to verify the message at the sensor nodes.

7.2 Communication Complexity

Communication complexity is the number and size of packets sent and received by a sensor node. We need to determine the number of sent and received messages to establish the link key between the sensor node and the security manager. In our protocol, the number of messages sent and received to establish a key between one sensor node and a security manager is 8 messages. Also, the number of messages sent and received to establish a key between a security manager and the base station is 6 messages. Device ID is 64 bits, expiration time is 64 bits, random number is 160 bits and L the sensor location is 64 bits. The certificate is 76 bytes from 20 bytes CA public key, 8 bytes node ID, 40 bytes node public key reconstruction data and 8 bytes validity time certificate.

Also, rekeying after compromised security manager will incur communication overhead as described above. Furthermore, compromised node revocation incurs a communication overhead.

7.3 Storage Complexity

Storage complexity is the amount of memory units required to store security credentials.

Each sensor node stores its public key and private key and the link key shared with the security manager. The security manager stores all of the public keys of its sensor nodes underneath plus the shared keys with each sensor node underneath plus its public, private key and link key with the base station. The security manager will use the idea of secret sharing and distributed storage with its neighbour nodes to store the public keys of its sensor nodes underneath. Each node stores 3 keys which are public key, private key and link key with its security manger.

7.4 Scalability

Scalability means whether a scheme support sensor node revocation/addition for large wireless sensor network. Our proposed scheme has high scalability for large scale network using certificate for each new sensor node joining the network.

7.5 Resiliency

Resilience means the probability that a link is compromised when an adversary captures a node or the number of sensors required for adversary to compromise the whole wireless sensor network.

We assume that an adversary can mount a physical attack on a sensor node after it is deployed and read secret information from its memory. We need to find how a successful attack on x sensor nodes by an adversary affects the rest of the network. In particular, we want to find the fraction of additional communication (i.e., communications among uncaptured nodes) that an adversary can compromise based on the information retrieved from the x captured nodes. In our proposed scheme, we found that the adversary cannot affect any other communications based on the compromised sensor nodes unless these nodes are security managers. Also, Rekeying improves network resiliency.

7.6 Connectivity

Connectivity means the connection probability for two nodes have the same predistributed key or establishing a key path between them. Our proposed scheme has high connectivity for large scale network where each node can connect to nearest security manager.

8. Comparison with others Works:

Now, we perform the performance analysis of our scheme and compare it with other key management schemes such as LEAP key management scheme [37], energy efficient hybrid scheme [38], and backbone hybrid assisted scheme [39]. We compare it in terms of storage cost and number of messages exchanged for key establishment.

Table 1, Performance Comparison

	Key Management		Storage
	Messages	**Complexity**	
LEAP [37]	N(1+5d)+ 2(d-1)^2/N+2N	O(d^2)	(3d+2+L) KL
Energy Efficient Hybrid Scheme [38]	$N+3m+N_S+N_{CH}$	O(log m)	(d+2) KL+ k KL
Backbone Hybrid Assisted Scheme [39]	3d + 2	O(log d)	4 KL
Our Proposed Scheme	$8(N-N_{SEC})$ + $6\ N_{SEC}$	O(N)	$3(N-N_{SEC})$ KL + $(2N_O+3)$ N_{SEC}KL

d is the average degree of the network, N is number of nodes in network, N_S is number of nodes in a subnetwork $N_S < N$, m is number of desired partial keys, KL is individual key length, k is dynamic subnetwork key length, L is length of key chain; N_{SEC} is the number of security managers, N_{CH} is a number of cluster heads and N_O is the number of nodes each security manager is responsible for. The degree of the network is defined as the average number of nodes that are within communication range of a given node. It is proportional to the density of nodes in the network and may be a value from 10 to 20 for a reasonably dense network.

As shown in Table I, Backbone assisted scheme only requires one broadcast message to initiate the key setup phase; it further requires two broadcasts to its first hop neighbours during rest of the process. It makes a total of three broadcast messages. It also requires two unicast messages to and from the gateway. Backbone hybrid assisted scheme is not valid for long lived sensor network in hostile environment. LEAP requires series of messages and key exchanges between nodes and network wide. Similarly, energy efficient hybrid scheme requires quite a few messages; network wide, with the cluster head and with nodes. Our proposed scheme requires high number of messages but less than LEAP and it achieves higher resiliency to network attacks in addition to high scalability. During the real-time execution of our protocol, a total of 8 massages are exchanged, two for mutual authentication and implicit certificates, two for node registration at base station, two for the afterwards link key generation process and another two for the explicit key confirmation.

Backbone assisted scheme has only 4 group keys which is low storage overhead but this scheme is not valid for long lived sensor network in hostile environment. Backbone assisted scheme is not scalable and it has low resiliency to node capture attack. LEAP has high storage overhead, good scalability and very good resiliency to node capture attack. Energy efficient hybrid scheme has medium storage overhead. Energy efficient hybrid scheme assumed that nodes are safe from capture and compromise for a period of time after deployment which is not valid for long lived sensor network in hostile environment therefore; it is not resilient to node capture attack. Energy efficient hybrid scheme is scalable. Our designed protocol has medium storage overhead where each node has public key, private key and link key, the security manager has keys double the number of nodes underneath where security manager stores all public keys of nodes underneath and all link keys of nodes underneath plus the security manager public key, private key and link key with the base station. Our protocol is scalable and it has high resiliency to node capture attack, collusion attack, cloning attack and sybil attack.

9. Conclusion:

In this paper, we propose a novel location based and hybrid key management scheme for Wireless Sensor Networks which utilizes Elliptic Curve Cryptography and the symmetric key cryptography. We propose a hybrid authenticated key-establishment protocol, in which we reduce the computation intensive elliptic curve scalar multiplication of a random point at the sensor side, and use symmetric key cryptographic operations instead. On the other hand, it authenticates the two identities based on elliptic curve implicit certificates, solves the key distribution and storage problems, which are typical bottlenecks in pure symmetric-key based protocols. The hybrid key establishment protocol has less sensor side computation complexity compared to other public-key based key establishment protocols.

In addition, we also design a dynamic key management based on rekeying, keys revocation and addition of new nodes which significantly increase the resiliency of the network to compromised node attack, collusion attack, cloning attack and sybil attack. The performance evaluation and security analysis show that our proposed key management scheme has good communication overhead, storage overhead, computations overhead and it provides perfect scalability and resiliency against node capture, cloning attack and sybil attack.

References:

[1] H. Chan and A. Perrig. “PIKE: Peer Intermediaries for Key Establishment in Sensor Networks”, *INFOCOM*, 2005.

[2] Chan H, and Perrig A, “Random key predistribution schemes for sensor networks”. In: Proceedings of the 2003 IEEE symposium on security and privacy, May 2003. pp. 197–213.

[3] DuW, Han YS, Chen S, and Varshney PK, “A key management scheme for wireless sensor networks using deployment knowledge”. In: Proceedings of IEEE INFOCOM04. IEEE 2004. pp. 586–97.

[4] Liu D, and Ning P. “Establishing pairwise keys in distributed sensor networks”. In: Proceedings of 10th ACM conference on computer and communications security (CCS03). 2003. pp. 41–7.

[5] Yu Z, and Guan Y. A “Robust group-based key management scheme for wireless sensor networks”. In: Proceedings of IEEE wireless communications and networking conference (WCNC 2005), New Orleans, LA USA. IEEE Press; 2005. pp. 13–7.

[6] Lee J, and Stinson DR. “Deterministic key predistribution schemes for distributed sensor networks”. In: Proceedings of ACM symposium on applied computing 2004, Lecture notes in computer science, vol. 3357, 2005, Waterloo, Canada, 2004. p. 294–307.

[7] Camtepe SA, and Yener B. “Combinatorial design of key distribution mechanisms for wireless sensor networks”. IEEE/ACM Transactions on Networking (TON) 2007;15(2):346–358.

[8] Eltoweissy M, Moharrum M, and Mukkamala R. “Dynamic key management in sensor networks”. IEEE Communications Magazine 2006;April: 122–30.

[9] N. Gura, A. Patel, A.Wander, H. Eberle, and S.C. Shantz. “Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs”. In CHES, Cambridge, MA, Aug 2004.

[10] Liu and Ning. http://discovery.csc.ncsu.edu/software/TinyECC/2005

[11] Qiang Huang, Johnas Cukier, Hisashi Kobayashi, Bede Liu and Jinyun Zhang, “Fast authenticated key establishment protocols for self-organizing sensor networks”, WSNA '03 Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications

[12] Mohamed Megahed, and Dimitrios Makrakis, “SurvSec: A New Security Architecture for Reliable Network Recovery from Base Station Failure of Surveillance WSN”, 2nd International Conference on Ambient Systems, Networks and Technologies, ANT 2011

[13] L. Eschenauer and V. Gligor, “A Key Management Scheme for Distributed Sensor Networks,” Proc. 9th ACM Conf. Comp. and Commun. Sec., Nov. 2002, pp. 41-47.

[14] H. Chan, A. Perrig, and D. Song, “Random Key Predistribution Schemes for Sensor Networks,” IEEE 2003, pp. 197–213.

[15] D. Liu and P. Ning, “Improving Key Pre-Distribution with Deployment Knowledge in Static Sensor Networks,” ACM Trans. Sensor Networks, 2005, pp 204–39.

[16] D. Liu, P. Ning, and W. Du, “Group-Based Key Pre-Distribution in Wireless Sensor Networks,” Proc. 2005 ACM Wksp. Wireless Security (WiSec 2005), Sept. 2005, pp.11–20.

[17] M. Eltoweissy et al., “Combinatorial Optimization of Key Management in Group Communications,” J. Network and Sys. Mgmt., Special Issue on Network Security, Mar. 2004, p. 332b.

[18] M. Eltoweissy et al., “Group Key Management Scheme for Large-Scale Wireless Sensor Network” Ad Hoc Networks, 2005, pp.796-802.

[19] G. Jolly et al., “A Low-Energy Key Management Protocol for Wireless Sensor Networks,” IEEE 2003, p. 335.

[20] M. Younis, K. Ghumman, and M. Eltoweissy, “Location aware Combinatorial Key Management Scheme for Clustered Sensor Networks,” to appear, IEEE Trans. Parallel and Distrib. Sys., 2006.

[21] Perrig A, Szewczyk R, Wen V, Cullar D, and Tygar JD. “SPINS: security protocols for sensor networks”. In: Proceedings of the 7th annual ACM/IEEE international conference on mobile computing and networking, July 2001. p. 189–99.

[22] DuW, Wang R, and Ning P. “An efficient scheme for authenticating public keys in sensor networks”. MobiHoc, 2005. pp. 58–67.

[23] Watro R, Kong D, Cuti S, Gardiner C, Lynn C, and Kruus P. “Tinypk: securing sensor networks with public key technology”. In: Proceedings of the 2nd ACM workshop on security of ad hoc and sensor networks (SASN 04). New York, NY, USA: ACM Press; 2004. p. 59–64.

[24] Karlofand C, Sastry N, and Wagner D. “Tinysec: a link layer security architecture for wireless sensor networks”. In: Second ACM conference on embedded networked sensor systems (SensSys 2004), pp. 162–75.

[25] Gaubatz G, Kaps J-P, and Sunar B. “Public key cryptography in sensor networks”. In: 1st European workshop on security in ad-hoc and sensor networks (ESAS 2004), 2004.

[26] Zhang J, and Varadharajan V. “Group-based Wireless Sensor Network Security Scheme”. In: The fourth international conference on wireless and mobile communica- tions (ICWMC 2008), July 2008.

[27] D. Liu and P. Ning, “Location-based pairwise key establishments for static sensor networks,” in Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks, pp. 72–82, October 2003.

[28] Cungang Yang, Celia Li, and Jie Xiao, “Location-based design for secure and efficient wireless sensor networks”, Elsevier 2008, pp. 3119–3129.

[29] Katerina Simonova, Alan C. H., Ling, X., and Sean Wang, “Location-aware Key Predistribution Scheme for Wide Area Wireless Sensor Networks”, SASN’06, ACM 2006.

[30] Yanchao Zhang, Wei Liu, Wenjing Lou and Yuguang Fang, “Securing Sensor Networks with Location-Based Keys, IEEE 2005.

[31] Mohamed F. Younis, and Mohamed Eltoweissy, “Location-Aware Combinatorial Key Management Scheme for Clustered Sensor Networks”, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 17, NO. 8, IEEE 2006.

[32] Chunguang Ma, Guining Geng, Huiqiang Wang, and Guang Yang, “Location-aware and secret share based dynamic key management scheme for WSN”, Networks Security, Wireless Communications and Trusted Computing Conference, IEEE 2009.

[33] Rene Struik and Gregg Rasor, “Mandatory ECC Security Algorithm Suite”, IEEE 2002, Wireless Personal Area Networks, March 2002.

[34] C. Savarese, J. Rabay and K. Langendoen. “Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks”. USENIX Technical Annual Conference, Monterey, CA, June 2002.

[35] E. Fujisaki, T. Kobayashi, H. Morita, H. Oguro, T. Okamoto, S. Okazaki, and D. Pointcheval. “PSEC: Provably secure elliptic curve encryption scheme”. IEEE 2000.

[36] M. Aydos, T. Yan and C. K. Koc. “A High-speed ECC-based Wireless Authentication Protocol on an ARM Microprocessor”. 16th Annual Computer Security Applications Conference (ACSAC'00), 2000.

[37] S. Zhu, S. Setia and S. Jajodia, “LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks,” Proc. of the 10th ACM Conference on Computer and Communications Security (CCS ’03), Washington D.C., pp. 62-72, October, 2003.

[38] Tim Landstra, Maciej Zawodniok, and S. Jagannathan, "Energy-Efficient Hybrid Key Management Protocol for Wireless Sensor Networks," pp.1009-1016, (LCN 2007), 2007.

[39] Tufail, A. and Ki-Hyung Kim; “A backbone assisted hybrid key management scheme for WSN”, International Conference on Information Society (i-Society) 2011, pp. 86-91, IEEE 2011.

[40] Mohd Anuar Jaafar, and Zuriati Ahmad Zukarnain, “Performance Comparisons of AODV, Secure AODV and Adaptive Secure AODV Routing Protocols in Free Attack Simulation Environment”, European Journal of Scientific Research, ISSN 1450-216X Vol.32 No.3 (2009), pp.430-443.