Wireless Sensor Network, 2010, 2, 689-697
doi:10.4236/wsn.2010.29083 Published Online September 2010 (http://www.SciRP.org/journal/wsn)

An Adaptive Key Management Framework for the Wireless Mesh and Sensor Networks

Mi Wen[1], Zhi Yin[1], Yu Long[2], Yong Wang[1]
[1]*department of Computer Science & Engineering;* [2]*department of Computer Science & Engineering;*
[1]*Shanghai University of Electric Power;* [2]*Shanghai Jiaotong University,*
[1]*Shanghai, China*
E-mail: *superwm_9@yahoo.com*
Received June 30, 2010; *revised July* 31, 2010; *accepted August* 30, 2010

Abstract

Wireless sensor networks (WSNs) and wireless mesh networks (WMNs) are popular research subjects. The interconnection of both network types enables next-generation applications and creates new optimization opportunities. Currently, plenty of protocols are available on the security of either wireless sensor networks or wireless mesh networks, an investigation in peer work underpins the fact that neither of these protocols is adapt to the interconnection of these network types. The internal cause relies on the fact that they differ in terms of complexity, scalability and network abstraction level. Therefore, in this article, we propose a unified security framework with three key management protocols, MPKM, MGKM, and TKM which are able to provide basic functionalities on the simplest devices and advanced functionalities on high performance nodes. We perform a detailed performance evaluation on our protocols against some important metrics such as scalability, key connectivity and compromise resilience, and we also compare our solution to the current keying protocols for WSNs and WMNs.

Keywords: Wireless Mesh Sensor Network, Key Management, Adaptive Security, Group Key

1. Introduction

The success of wireless technologies today caused the international wireless network research community to have high hopes for the future. Wireless mesh sensor network (WMSN) is a new architecture that merges advantages of wireless mesh networks (WMN) and wireless sensor networks (WSN), especially on scalability, robustness and balanced energy dissipation [1]. Wireless sensor networks and wireless mesh networks are popular research subjects. The interconnection of both network types enables next-generation applications and creates new optimization opportunities [2]. Many application scenarios could benefit from a successful and optimal interconnection between WSNs and WMNs. For example, a wireless mesh network can be used as a backbone for collecting sensor data from remote sensor clusters, or, resource intensive calculations with sensor data may be performed on a mesh router instead on a sensor node. Although plenty of research is available on all aspects of either wireless sensor networks or wireless mesh networks, little information is available on the interconnection of these network types. Their difference between WMNs and WSNs in wireless technologies, addressing protocols, routing strategies and security mechanisms make an effective interconnection be challenging.

Especially, WMSNs are always deployed in hostile environments to track target, monitor battlefield, detect intruder or do some scientific explorations and the openness of the wireless environment makes security in WMSNs a critical concern in the deployment of such group applications. In wireless communication environments an adversary not only can eavesdrop the radio traffic in a network, but also can intercept the exchanged data. To prevent the malicious node impersonating good nodes for spreading misleading data intentionally, secret keys should be used to achieve data confidentiality, integrity and authentication between communicating parties [3]. But in WMN networks, security and trust is most often guaranteed using either pre-shared keys, or by relying on certificate based encryption techniques [4]. Because of the limited capacities of sensor nodes, the security approaches used in WMNs are not suitable for WSNs [5]. Some sensor nodes might be unable to im-

plement any certificate based security mechanism at all. Therefore, the development of adaptive key management protocols is a promising approach to enable low end devices to participate in heterogeneous network architectures securely.

Adaptive key management protocol is an effective approach to provide efficient and secure interconnection, while respecting the individual characteristics of each network type. The main difficulty with adaptive key management protocols is the creation of a basic key management protocol version that can be deployed on a very basic network node. Current popular key management protocols in WMNs such as the [6-8] are relatively complex. Even though the performance of sensor nodes will increase over time, there will always remain a class of devices that is unable to run these complex protocols. Therefore, there is a need for novel, simple techniques that are able to provide basic functionalities on the simplest devices and at the same time they can be extended to support advanced functionalities on high performance nodes. Thus, an adaptive and modular key management approach is needed.

In this paper we present a unified security framework that embodies three key management protocols which can provide adaptive security for WMSNs for the need of the applications. The framework includes three key management protocols: (a) the Matrix based Pairwise Key Management (MPKM) protocol for sensor nodes with limited resources. (b) the Matrix based Group Key Management (MGKM) protocol for the network with sinks or cluster heads except the sensor nodes; (c) the Threshold Key Management (TKM) protocol for the network with mesh nodes in addition. All of these three protocols are interrelated elements: MGKM can be extended from MPKM, and TKM can be extended from MGKM. Therefore, the accession of MGKM and TKM in the WMSNs will not increase the storage or communication overhead of sensor nodes.

2. Related Works

Key Management is one of the main challenges in securing wireless networks, and has been addressed by many authors. In this section, we present an overview of some approaches and protocols for keying management in both WSNs and WMNs respectively.

2.1. Key Management in WSNs

To date, the key management protocols in sensor networks can be mainly classified into two types: pairwise key management protocols and group key management protocols. In pairwise key management protocols [9-14], each pair of communication nodes should establish a shared key. One attractive idea in the pairwise key management is key pre-distribution, *i.e.*, pre-installing a limited number of secrets in sensor nodes prior to actual deployment; after the deployment, if two neighboring nodes have some common keys, they can setup a secure link by the shared keys. While in the group key management protocols [15-17], the key idea is to broadcast information that is useful only for trusted nodes. Combined with its pre-distributed secrets, this broadcast information enables a trusted sensor node to reconstruct a group key. Most pairwise key and group key management protocols in WSNs are based on symmetric key cryptography, such as Du's [12] key Matrix based, Camtepe's [9] Combinatorial Design based, Liu's [13] polynomial based protocols. These solutions are designed to sustain severe computation power, storage, mobility, and energy constraints, and as a result have limited scalability and robustness. Although some research [18] shows that the right selection of algorithms and associated parameters along with code optimization can make public key cryptography feasible for sensor networks. For example, the ECC and RSA based key management protocols. The major shortcomings of them are the associated expensive computation and the high probability of likely penetration by malicious agents. Also all current asymmetric key related studies only support their feasibility for WSN's. Unfortunately, as we know, none of current works propose complete key management infrastructure compatible public and private key cryptography.

2.2. Key Management in WMNs

Secure group communication is a mature research area and has a large body of research literature. The main objective of a secure group communication protocol is to ensure the data confidentiality against outsiders such that only legitimate group members can recover the group data. Existing solutions for wired networks [19-21] are not well suited for WMNs as they fail to take into consideration the multi-hop communication paradigm featured by WMNs, as well as the communication security among mesh clients within the coverage of a mesh router. These protocols also do not exploit unique features of WMNs, such as the broadcast nature of wireless communication. ARSA [7] proposes attack-resilient security architecture for WMNs, which uses ID-based cryptography (IBC). SeGrOM [8] propose a new protocol framework for secure group overlay multicast in WMNs. LSSS [6] presents an ideal linear multi-secret sharing protocol, by using monotone span programs. Though, they achieve efficient and secure group communication in WMNs. They can not be employed in the WSNs due to their ex-

pensive energy consumption and also they can not offer modular security for WMSNs.

In general, none of the existing protocols considered the unique features of WMSNs, such as coexistence of resources constrained sensor nodes and powerful mesh nodes, increasing scalability when remote cluster sensors get interconnected thanks to the presence of a WMN, all of which can be leveraged for designing more optimized protocols. Our work tries to fill this gap by designing such a complete key management infrastructure specially for WMSNs based on our previously key management protocols [14,17].We will take into account the diversity of nodes' ability and propose a unified key management framework, which includes simple techniques that are able to provide basic functionalities on the simplest sensor devices and at the same time they can be extended to support advanced functionalities on high performance mesh nodes.

3. System Model and Assumptions

3.1. Network Model

Define our target network environment is the interconnection of WSNs and WMNs, called WMSNs. The WMNs include a set of static wireless routers, called mesh nodes (MN), organized in a backbone network and communicating through multi-hop wireless links. Mobile clients (MC) connect to the wireless mesh through a local access router, called access point (AP), and communicate with each other through the wireless mesh. While the WSN has the hierarchical architecture consisting of numerous sensor nodes (SN) grouped in clusters and each cluster has a cluster head (CH), which is responsible collecting and merging local data from sensor nodes and send it to mesh nodes. Clusters of sensors can be formed based on various criteria such as location, communication range, resource and energy capabilities, *etc*. (See **Figure 1**). Resource intensive calculations with sensed data may be performed on a MN. MN here can be considered as an actuator node in WSNs and can take immediate response when monitoring some abnormal phenomena in WSNs. Many application scenarios could benefit from successful and optimal WMSNs. For example, the WMNs can be used as a backbone for collecting sensor data from remote sensor clusters. For clarity, we describe the terms in the scope of this article is specified as follows:

Sensor nodes are network nodes with limited capabilities in terms of processing power, memory capacity and bandwidth, equipped with a sensor and/or actuator chip. As such, a sensor node can be a source of data in a network, but could as well be used as intermediate node to forward data from one sensor device to another, or to a data collection device, called a cluster head (see **Figure 1 (ii)**). Sensor nodes are small sized and limited in cost. With WSN, all forms of wireless networks between sensor devices are indicated [1]. These sensor networks are self-forming, and are used to gather data in places where the use of cabled sensors is hard, costly or undesired. No restriction is made based on network size or topology: both single hop networks between SNs and a CH, and complex multi-hop networks with meshed topologies are considered.

Mesh nodes are relatively powerful networked nodes, equipped with relatively powerful wireless interfaces and thus are able to transmit and receive at higher bandwidths than sensor nodes. With WMN or wireless mesh networks, all forms of wireless networks between mesh nodes are indicated. Again, there are no restrictions on the topology. Mesh networks are often used as a wireless backbone for the interconnection of end user devices. WMNs might also offer additional functionality to the client networks; for example, provide an uplink to the Internet (see **Figure 1(i)**). Mesh networks are self-forming and self-healing, and are therefore an ideal solution to provide connectivity in places where cabled networks cannot easily be installed. Furthermore, because of their self-organizing character, mesh networks can be rolled out fast, making them ideal candidates to be used as emergency network infrastructure.

3.2. Definitions and Notations

In the following section, we give some definitions and notations used in this paper unavoidable.

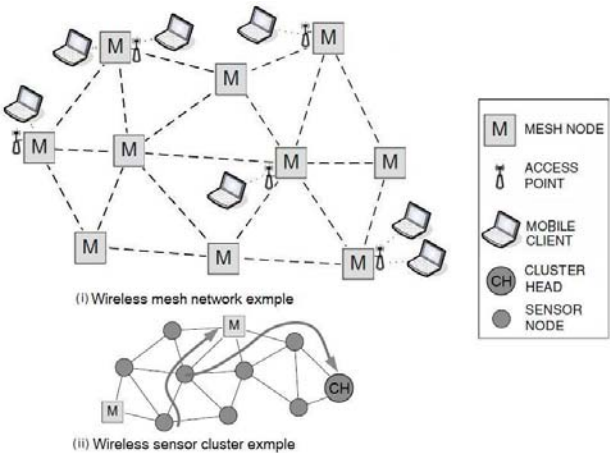

Figure 1. Wireless mesh and sensor network architecture example.

4. Proposed Framework

In this section, we first describe MPKM, our basic key management protocol handling the pairwise key establishment for the resource limited sensor nodes. We then describe two additional protocols, (a) MGKM, a key management protocol handling the group key of the WSNs and (b) TKM, a key management protocol handling the key sharing in WMN using the asymmetric cryptography. All these protocols together can manage the key establishments in WMSNs. Since MPKM and MGKM are previously proposed in [14] and [17]. Here we include them only to form the unified key managemen framework. Thus, we will focus on the TKM protocol in the latter of this paper. Due to the node resource limitation, MPKM and MGKM are based on symmetric key cryptography while TKM is based on asymmetric key cryptography. All of these three protocols are interrelated elements: MGKM can be extended from MPKM and TKM can be extended from MGKM. (See **Figure 2**). Therefore, the accession of MGKM and TKM in the WMSNs will not increase the storage or communication overhead of sensor nodes. This is one of the advantages of our framework and it just satisfies the requirement of key management technique in WMSNs, scalable and lightweight.

4.1. The Matrix Based Pairwise Key Management (MPKM) Protocol

In MPKM, each sensor node is programmed according to the application requirements before network deployment. As we all know, Blom proposed a key distribution approach [12], which allows any pair of nodes in a network to be able to find a pairwise secret key. As long as no more than nodes are compromised, the network is perfectly secure. For the sake of key updating, we modify the Blom's symmetric matrix construction in [14]. We briefly describe how to use our modified version of Blom's key distribution approach to establish paiwise key as follows. Some used notations in this paper are given in **Table 1**.

The base station (BS) in WSNs (acting as a trusted server) first computes a $n*n$ matrix B over a finite field $GF(q)$, B is considered as the public information, q is a prime, and $q < n$. One example of such a matrix is a Vandermonde matrix whose element $b_{ij} = (g^j)^i \bmod q$, where g is the primitive nonzero element of $GF(q)$ and g^j is the jth column seed. That means:

$$B = \begin{bmatrix} 1 & 1 & 1 & & 1 \\ g & g^2 & g^3 & & g^n \\ & & & & \vdots \\ g^{n-1} & (g^2)^{n-1} & (g^3)^{n-1} & \cdots & (g^n)^{n-1} \end{bmatrix}$$

Table 1. Table type styles.

N_i	A sensor node i ($i=1,...,n$), where N_n is the trusted dealer and it has more power than normal sensor node, i.e. the cluster head in a cluster.
s_i	Row seed of the matrix D; the row seed used in each row i of matrix D should not bigger than s_i.
g^j	Column seed of matrix B
K_{ij}	The pairwise key between node N_i and N_j
K_G	The group key of the initial set N={ $N_1, N_2, ..., N_n$ }
GID_i	The group or cluster identity of cluster i
ID_i	The identity of sensor node i
K_{Gi}	The group key of the sensor cluster i
sk	The secret key to be shared by the mesh nodes
sk_i	The secret key share of mesh node i

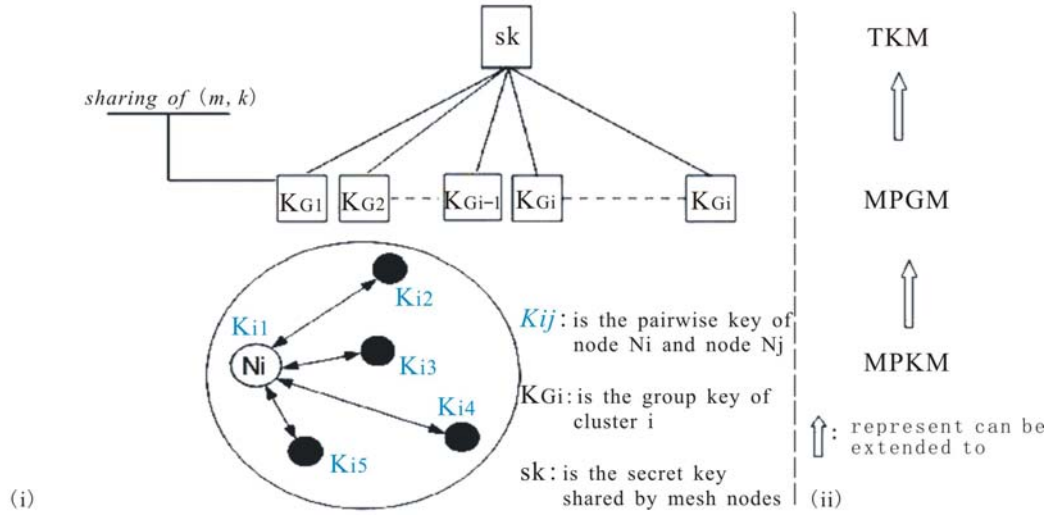

Figure 2. Overview of our key management framework.

This construction requires that $n^2 < \varphi(q)$ *i.e.*, $n^2 < q-1$. Since *B* is a Vandermonde matrix, it can be proved that the n columns are linearly independent when $g, g^2, g^3 \cdots, g^n$ are all distinct.

Next, the BS generates n row seeds $s_1, \cdots s_n$, where $s_i (i = 1, ..., n)$ is the random prime number of *GF*(*q*) and it is only known to the powerful node (N_n) in the network, *e.g.*, the cluster head of WSNs. And then BS creates a random $n*n$ symmetric matrix *D* over *GF*(*q*). Each row of the *D* is composed of hash values of the row seeds. Differing from the construction of matrix *B*, the elements in symmetric matrix *D* are generated as follows:

$$for(i = 1; i \le n; i++)$$
$$for(j = 1; j \le n; j++)$$
$$\{if(i > j)\ d_{ij} = H^i(s_j); else\ d_{ij} = H^j(s_i);\}$$

Where d_{ij} is the element in matrix *D*. An example of matrix *D* with size $3*3$ is shown as follows:

$$\begin{bmatrix} H^1(s_1) & H^2(s_1) & H^3(s_1) \\ H^2(s_1) & H^2(s_2) & H^3(s_2) \\ H^3(s_1) & H^3(s_2) & H^3(s_3) \end{bmatrix}$$

At last, the BS computes a $n*n$ matrix $A = (DB)^T$, where *T* indicates a transposition of the matrix. The elements in matrix *A* denote as a_{ij}, where $a_{ij} = \sum_{\beta=1}^{n} d_{j\beta} b_{\beta i}$. The matrix *B* is public while the matrix *D* is kept secret by the base station. Since *D* is symmetric, the key matrix $K = AB$ can be written as:

$$K = (DB)^T B = B^T D^T B = B^T DB = (AB)^T = K^T$$

Thus *K* is also a symmetric matrix and $K_{ij} = K_{ji}$, where K_{ij} is the element of *K* at *i*th row and *j*th column. We take K_{ij} (or K_{ji}) as the pairwise key between node N_i and node N_j. To carry out the above computation, nodes N_i and N_j should be able to compute K_{ij} and K_{ji} respectively. This can be easily achieved using the following key pre-distribution procedure, for node N_i:

(1) Store the *i*th row of matrix *A* at node N_i, denoted as $r_i(A)$, *i.e.*, $\mathrm{r_i(A)} = [\mathrm{a_{ij}}]$, $(j = 1, \cdots, n)$.

(2) Store the *i*th column seed g^i of matrix *B* at N_i.

After deployment, each node has a piece of secret information as described above. When nodes N_i and N_j need to find the pairwise key between them, they first exchange their column seeds of matrix *B* (since *B* is the public information, it can be sent in plaintext). Then, by using the preloaded secrets, they can compute K_{ij} (or K_{ji}) respectively as: $K_{ij} = \sum_{\beta=1}^{n} a_{i\beta} b_{\beta j}$. It can be proved that the above protocol is *n*-secure because all the rows in *D* are linearly independent. And this property guarantees the uniqueness of the pairwise keys in the cluster.

4.2. The Matrix Based Group Key Management (MGKM) Protocol

When nodes with better resources, named as CHs, are deployed in the network, they can be used to collect and merge local data from sensor nodes and send it to mesh nodes. Mesh nodes then distribute the required information to the end user clients. Now, we assumed that after deploying the new nodes into the operating WSNs, the clusters can be formed based on various criteria such as capabilities, location, and communication range *etc*. [17].

Now, without loss of generality, let $N = \{N_1, N_2, ..., N_n\}$ be the initial set of participants in each cluster group that want to generate a group key. Assume that there are *n-1* sensor nodes and a powerful node N_n in a cluster (N_n may be a CH). The detailed steps of the group distribution are presented as follows.

Step 1: Initially, each node $N_i (1 \le i \le n-1)$ is pre-loaded a row $r_i(A)$ from matrix *A* and column seed g^i as described in Subsection 4.1. Then, after deployment, each node pre-computes K_{in}, K_{ii} (*i.e*, $K_{ii} = \sum_{\beta=1}^{n} a_{i\beta} b_{\beta i}$) an K_{ii}^{-1}. N_i sends the enciphered message (N_i, $C_i = E_{K_{in}}(K_{ii})$) to node N_n and keeps K_{ii}^{-1} in its local memory. K_{in} is the pairwise key between node N_i and N_n. || stands for message concatenation.

Step 2: Node N_n computes K_{nn} as above. Upon receiving each $(\mathrm{N_i}, C_i)(1 \le i \le n-1)$, node N_n deciphers them and computes $x_i = K_{nn} K_{ii}$. Next, node N_n computes $K_G = K_{nn} \prod_{i=1}^{n-1} x_i$. Finally, the powerful node N_n broadcasts (N_n, $x_1 ... x_{n-1}$) to other nodes.

Step 3: On receiving the broadcast messages, each node $\mathrm{N_i} (1 \le i \le n-1)$ computes the common group key $K_G = x_j K_{jj}^{-1} \prod_{i=1}^{n-1} x_i$.

Note that the client N_i may pre-compute K_{ii}^{-1} to reduce the computational load. Until now, storage limitation is becoming less of a concerning issue as many add-on memory cards are widely available. And we can prove that the proposed protocol is a contributory group key agreement protocol. Reference [17] for detailed prove process.

4.3. The Threshold Key Management (TKM) Protocol

If a WMN is added to an already existing WSN to collect the sensed messages, should any adjustments to the WSN protocols be made? In general: the less adjustments are to be made on either WSN or WMN protocols, the faster an interconnection can be realized and the sooner an interconnection strategy might be adopted. Moreover, using either a single symmetric key or by relying on certificate based encryption techniques to achieve key management operations for the mesh nodes risks high

probability of key leakage or creates a vulnerable point in the network. Thus, based on the two former key management protocols, MPKM and MGKM, we design a threshold key management protocol for mesh network. In such a system, the group keys of the WSNs will be calculated as a secret key shared by n mesh nodes. And the secret key can be recovered by a coalition of t mesh nodes.

4.3.1. Preliminaries of Threshold Secret Sharing

The proposed protocol is based on the (m, k) threshold cryptography [22]. Generally speaking, threshold cryptography is used for distribution of a secret value S based on polynomial interpolation, and an (m, k) threshold protocol allows m parties to perform cryptographic operations, so that any k parties can jointly perform key discovery whereas (k-1) parties cannot derive any information even after collusion. The parameter k represents the threshold. A sample threshold cryptography protocol proposed by Shamir can be explained as follows:

Consider the secret S, we can store the secret about S into n shares ($s_1, \ldots, s_m$) via a randomly chosen k degree polynomial $f(x) = a_0+a_1x+\ldots+a_{k-1}x^{k-1}$ where $a_0 = S$. Secret shares are obtained by $s_i = f(i)$, $i = 1 \ldots m$. The m shares of secrets are simply $\{f(1), f(2), \ldots, f(m)\}$. Given k points from the above m shares, we can derive the coefficients of $f(x)$ by interpolation and hence calculate the secret $S = \sum_{i=1}^{k} f(i)L_i(0)$, where $L_i(0)$ is the Lagrange coefficient such as

$$L_i(x) = \frac{\prod_{j \neq i}(x - x_j)}{\prod_{j \neq i}(x_i - x_j)}$$

Therefore, the above protocol is a (n, k) threshold cryptography protocol.

4.3.2. The Proposed TKM Protocol

We assume that there is a Trusted Server (TS, the base station in WSN also can act as a TS) which can calculate the secret key and the key shares for bootstrapping the mesh nodes. And we also assume that when a WMN with m mesh nodes is added to an already existing WSN, each mesh node is innocent and cannot be compromised during the first several minutes after deployment since compromising a node takes some time. Based on this, the system initialization process is carried out in two phases: secret key calculation and mesh nodes bootstrapping. In the first phase, the TS will collect the group keys of the sensor nodes and calculate the secret key to be shared by the mesh nodes. Here, we assumed that the messages delivered among sensor nodes and mesh nodes are always encrypted by their group keys, the collaboration of t mesh nodes can decrypt them. In the second phase, the TS creates an (m, k) sharing ($sk_1, sk_2, \ldots, sk_m$) and privately distributes these shares to m mesh nodes where ($m < M$) and M is the network size.

Secret key calculation phase: We assume that before the WMN is added, the WSNs are classed by t clusters, and each cluster has a CH and several SNs. Also, the keys in WSNs are already established by using MPKM and MGKM protocols. Thus, in this phase, the TS

- Step 1: Broadcast a hello message $\{ID_{TS}, hello\}$ to the sensor nodes in WSNs.
- Step 2: On receiving the hello message, each CH or the SNs will reply a message which contains its group keys $\{GID_i, K_{Gi}, ID_{CH}\}$.
- Step 3: By distinguish the different group keys K_{Gi} ($i=1,\ldots, t$) from the WSNs. The TS calculate the secret key SK for mesh nodes as following:

$Sk = K_{G1} \oplus K_{G2} \oplus \ldots\ldots \oplus K_{Gt}$, where $\oplus$ represents XOR operation.

Mesh nodes bootstrapping phase: In this phase, the TS performs the following operations:

- Step1: Create a random polynomial of degree $k-1$: $f(x) = sk + a_1x + \cdots + a_{k-1}x^{k-1} (mod\ p)$ where p is a large prime number and sk is the shared private key of the mesh nodes.
- Step2: Calculate and send to each node i the corresponding share of (sk): $sk_i = f(i) (mod\ p)$. For simplicity i is assumed to be an integer and nodes to be initialized range from $1 \ldots m$.
- Step3: Calculate and store locally the decryption supplementary keys S_i for each sensor group as follows, and then deleted the K_{Gi} ($i=1,\ldots, t$) permanently.

$S_i = sk \oplus K_{Gi} (i = 1,\ldots, t)$

When a mesh node j receives messages from sensor nodes, it should broadcast a request to its neighbors. After collecting k-1 valid shares from its neighbors, it combines them with its share in order to issue the secret key sk. If the WSN only have one group, then $sk = K_{G1}$, the requester mesh node j can use sk to decrypt the received messages immediately. If the WSN have two or more groups, the requester node j should ask the TS for help. The TS replies with the group i's decryption supplementary keys S_i. Then, the mesh node j decrypt the received messages by calculate $sk \oplus S_i$, where $sk \oplus S_i = K_{G1}$.

5. Post Deployment Operations

Network post-deployment issues are critical factors in determining the efficiency of any key management protocol for WMSN specific environment. Each protocols working in correspondence to these issues is explained against the following matrices.

Scalability: Each of the three protocols supports node

additions after network deployment. In case of MPKM and MPGM, when a new node $ID_{(n+1)}$ wants to join the network, the BS will generate the key information for node n+1 and the Real-Time generation (RTG) program in [14] will be triggered to expand the key matrix. Thus, all nodes in its cluster will establish new pairwise key with this node. And periodically their group key also will be updated according to reference [17].

In TKM, when a new sensor cluster wants to join the existing WMSN, here we assume that the pairwise keys and group key in this cluster have been established by using the MPKM and MPGM protocols. The main issue in this procedure is the secret key updating and the key share information updating for existing mesh nodes. There are two types of techniques can address this problem.

(1) *Regeneration*: in this approach, the TS first recalculate the secret key by XOR *sk* and the new cluster's group key, here we denote it as K_{Gnew}. So, the new secret key for the mesh nodes in new mesh network is $sk' = sk \oplus K_{Gnew}$. Then, the TS recreates a random polynomial of degree $k-1$: $h(x) = sk' + b_1x + \cdots + b_{k-1}x^{k-1} \pmod p$ and resends the key share for mesh nodes. Finally, the mesh nodes can establish new secret key to guarantee the network security. The drawback of this approach is that it introduces substantive communication and computation overhead.

(2) *Real-Time updating*: This technique relies on the following Homomorphic property. If $(s_1, \ldots, s_n)$ is an (n, k) sharing of S and $(s'_1, \ldots, s'_n)$ is an (n, k) sharing of S', then$(s_1 \oplus s'_1, \ldots, s_n \oplus s'_n)$ is the an (n, k) sharing of $S \oplus S'$, if we set $S' = 0$, then we get a new (n, k) sharing of S.

Now, let $(sk_1,\ldots,sk_m)$ be the (m, k) sharing of sk and K_{Gnew} is the group key of the new cluster, the TS creates a random polynomial of degree $k-1$: $p(x) = K_{Gnew} + c_1x + \cdots + c_{k-1}x^{k-1} \pmod p$ where p is a large prime number. And then the TS calculates the corresponding share of (K_{Gnew}): $K_i = p(i) \pmod p$ and sends it to each node i. Thus, the mesh nodes can get a new sharing $(sk_1',\ldots,sk_m')$ of sk' where $sk_i' = sk_i \oplus K_i$.

Key connectivity: Key connectivity is described as the number of keys required to be stored on each node for specified level of required network connectivity.

MPKM establishes a pairwise key for each pair of nodes in the same cluster. Since the keying information of the nodes in one cluster is coming from the same symmetric key matrix, each pair of nodes in one cluster can establish a paiwise key. This provides 100% cluster wide connectivity.

MPGM provides good key connectivity on frequent broadcast basis. In a typical information gathering scenario where the primary purpose of the nodes is to gather data and forward it to BS or MN, nodes in one cluster communicate with each other more frequently. Such communication is ensured by a common cluster wide group key. Hence, broadcast based cluster-wide key connectivity is ensured.

In TKM, to encrypt the communication between the sensor nodes and the mesh node needs the sensor's group key, which is established in MPGM. And the decryption of the communication messages needs local collaboration of the neighboring mesh nodes by using the threshold cryptography. All these keys are established and secure. Therefore, complete network-wide key connectivity is ensured by using these three protocols.

6. Performance Analysis

6.1. Security Analysis

We have proposed an adaptive key management framework for WMSNs in this paper. From simple MPKM to complex TKM, they can provide modular security services from basic functionalities to advanced functionalities on kinds of nodes. For instance, they can establish both pairwise key and group key for sensor nodes with the same pre-distributed secrets and update their keys at the same time. Meanwhile, the secret key of the mesh nodes also can be established by extending the key management of the group keys. Furthermore, our framework satisfies the security requirements of both WSNs and WMNs. In this section, we analyze the security of our proposed key management framework.

Compromise resistance: In MPKM, since the paiwise key of each sensor node in our protocol is different from each other, the discovery of the captured keys cannot give out any knowledge of the keys in the innocent nodes. Moreover, any node's failure or compromise triggers a key updating process and, thus, the keys shared by the captured nodes and the innocent nodes are get rid of. That means MPKM provides sufficient security, no matter how many sensors in the same cluster are compromised MPKM can achieve perfect compromise resilience. In MPGM, the group key is established by the collaboration of all the nodes in the same cluster, no participant can predetermine the common key and each node has the same right to verify if its contribution is included in their currently used group key. Also, the group key can be updated periodically. Thus, MPGM has perfect group fairness and compromise resilience. In TKM, any k nodes can jointly perform key discovery whereas $(k$-1) nodes cannot derive any information even after collusion. Thus, it is k secure and has (k-1) resistance. Also, the secret key can be updated when the group keys changed or a new cluster wants to join the WMNs.

Group confidentiality: In MPGM, nodes that are not the part of the group should not have access to any key

that can decrypt any data broadcast to the group. For the message, only the powerful node (such as BS or TS) can decrypt the message to get the contributions of the client nodes by using its pairwise key with the client nodes. With the same reason, only the contributory nodes can recover the group key by using their own contribution. Therefore, the group key is group confidential.

6.2. Comparison with other Protocols

Now let us compare our protocols with the available key management techniques for wireless sensor networks and wireless mesh networks. Such as Du's [12] key Matrix, Camtepe's [9] Combinatorial Design, Liu's [13] polynomial based protocols for WSN, ARSA [7], SeGrOM [8] and LSSS [6] for secure WMN group communication.

For convenience, the following notations are used to analyze the various properties: Y: Yes or has; N: No or hasn't; Inc.: will increase; Nor.: remain normal; λ**:** the threshold, represents that if more than λ nodes be compromised, the protocol is not secure; little p: the protocol only provides p connectivity; big P: the protocol provides 100% or strong key connectivity. **Table 2** lists the comparisons among our protocols and protocols only for the WSN or WMNs.

We consider the performance comparisons in terms of the scalability，compromise resistance，key connectivity，nesh nodes security，modular security service. It is obvious that only our protocols can be scalable when the network size changes. If some nodes are compromised by the adversary, protocols for WSN and WMN only have a certain level of resistance, if the number of compromised nodes exceed the level, the network will not secure any more. While our protocols can maintain 2P+λ resistance, which mean MPKM and MPGM can maintain perfect compromise resistance, only TKM has a threshold λ**.** In terms of nodes security, only our protocols can provide security protection for both sensor nodes and mesh nodes and the storage for nodes will not increase no matter other two protocols being included or not. Thus, from the table we can see only the protocols proposed in our framework can be adaptive for WMSNs.

7. Conclusions

The interconnection of heterogeneous WSN and WMN networks is a pilot case which can be used to derive directions for the research on future heterogeneous network architectures. One of the major challenges for the development of future network architectures is the creation of adaptive key management protocols for the diversity of network nodes. In this paper we have presented an adaptive key management framework for WMSNs. It includes three possible keying implementations for different network nodes, *i.e.* pairwise key for sensor nodes, group key for high and low level sensor nodes and secret key shares through threshold cryptography for mesh nodes. The results clearly show that they can adapt to the different resource availability and achieve levels of security. In short, no matter the addition of low end nodes or high end devices to a secure environment might introduce security risks. The design of our key management framework can give simple but effective security solutions for each level of devices. And as we know, our framework is the first one which provides adaptive and modular security service for WMSNs. This is extremely important for the future generation of integrated networks.

Adaptive security is an interesting future research track. It will remain a continuous challenge to integrate low-end devices in future networking environments. In our future work, we would like to investigate new keying mechanism with better extension properties to provide perfect security and robust continuity for future generation of integrated networks.

8. Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant No.60903188, 60863001 and 60903189, the Innovation Program of Shanghai Municipal Education Commission under Grant No.10YZ157, the Shanghai university scientific selection

Table 2. Comparisons among our protocols and protocols only for the WSN or WMNs

	Du's	Liu's	Camtepe's	ARSA	SeGrOM	LSSS	Our's
Scalability	N	N	N	N	N	N	Y
Compromise resistance	λ	*p*	*p*	*p*	*p*	*p*	2P+λ
Key connectivity	*p*	*p*	*p*	P	P	P	P
Node's Storage overhead	In.	Inc.	Inc.	Nor.	Inc.	Inc.	Nor.
Sensor security	Y	Y	Y	N	N	N	Y
Mesh nodes security	N	N	N	Y	Y	Y	Y
Modular security service	N	N	N	N	N	N	Y

and cultivation for outstanding young teachers in special fund No.sdl09010.

9. References

[1] S. Bouckaert, E. D. Poorter, B. L. J. Hoebeke, I. Moerman and P. Demeester, "Strategies and Challenges for Interconnecting Wireless Mesh and Wireless Sensor Networks," *Wireless Personal Communications*, Vol. 53, No. 3, 2010, pp. 443-463.

[2] J. Ishmael and N. Race, "Wireless Mesh Networks (Handbook)," Chapter 7, Lancaster University, pp. 149-166.

[3] M. Wen, L. Dong, Y. F. Zheng and K. F. Chen, "Towards Provable Security for Data Transmission Protocols in Sensor Network," *Journal of Information Science and Engineering*, Vol. 25, No. 1, 2009, pp. 319-333.

[4] S. Glass, M. Portmann and V. Muthukkumarasamy, "Securing Wireless Mesh Networks," *IEEE Internet Computing*, Vol. 12, No. 4, 2008, pp. 30-36.

[5] S. Avancha, J. Undercoffer, A. Joshi and J. Pinkston, "Security for Wireless Sensor Networks," *Wireless Sensor Networks*, Kluwer Academic Publishers, Norwell, 2004, pp. 253-275.

[6] F. H. Ching, H. C. Guo, C. Qi and J. Chen, "A Novel Linear Multi-Secret Sharing Protocol for Group Communication in Wireless Mesh Networks," *Journal of Network and Computer Applications*, Vol. 10, No. 16, 2010.

[7] Y. C. Zhang and Y. G. Fang, "ARSA: An Attack-Resilient Security Architecture for Multi-Hop Wireless Mesh Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 24, No. 10, 2006, pp. 1916-1928.

[8] J. Dong, K. Ackermann and C. Nita-Rotaru, "Secure Group Communication in Wireless Mesh Networks," *Ad Hoc Networks*, Vol. 10, No. 16, 2009, pp. 1563-1576.

[9] S. A. Camtepe and B. Yene, "Key Distribution Mechanism for Wireless Sensor Networks," TR-05-07 Rensselaer Polytechnic Institute, Computer Science Department, March 2005.

[10] L. Eschenauer and V. D. Gligor, "A Key-Management Scheme for Distributed Sensor Networks," *Proceeding of the 9th ACM Conference on Computer and Communication Security*, Washington, DC, 2002, pp. 41-47.

[11] H. Chan, A. Perrig and D. Song, "Random Key Predistribution Schemes for Sensor Networks," *IEEE Symposium on Security and Privacy*, 2003, pp, 197-213.

[12] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz and A. Khalili, "A Pairwise Key Predistribution Scheme for Wireless Sensor Networks," *ACM Transactions on Information and System Security*, Vol. 8, No. 1, 2005, pp. 228-258.

[13] D. Liu and P. Ning, "Establishing Pairwise Keys in Distributed Sensor Networks," *ACM Transactions on Information and System Security*, Vol. 8, No. 1, 2005, pp. 41-77.

[14] M. Wen, K. F. Chen, Y. F. Zheng and H. Li, "A Reliable Pairwise Key-Updating Scheme for Sensor Networks," *Journal of Software*, Vol. 18, No. 5, 2007, pp. 1232-1245.

[15] D. Liu, P. Ning and K. Sun, "Efficient Self-Healing Group Key Distribution with Revocation Capability," *Proceedings of the 10th ACM Conference on Computer and Communications Security*, Washington, DC, 2003, pp. 231-240.

[16] W. Zhang and G. Cao, "Group Rekeying for Filtering False Data in Sensor Networks: A Predistribution and Local Collaboration-Based Approach," *Proceedings from the Conference of the IEEE Communications Society*, 2005, pp. 503-514.

[17] M. Wen, J. S. Lei, Z. Tang, X. X. Tian, K. F. Chen and W.D. Qiu, "A Verified Group Key Agreement Protocol for Resource-Constrained Sensor Networks," *Lecture Notes in Computer Science,* Vol. 5854, 2009, pp. 413-425.

[18] G. Gaubatz, JP. Kaps and B. Sunar, "Public Key Cryptography in Sensor Networks Revisited," *Proceedings of the 1st European Workshop on Security in Ad-Hoc and Sensor Networks*, Springer, 2004, pp. 2-18.

[19] S. Zhu, C. Yao, D. Liu, S. Setia and S. Jajodia, "Efficient Security Mechanisms for Overlay Multicast Based Content Delivery," *Computer Communications*, Volume 30, No. 4, February 2007, pp. 793-806.

[20] S. Zhu, S. Setia, S. Xu and S. Jajodia, "GKMPAN: An Efficient Group Rekeying Scheme for Secure Multicast in Ad-hoc Networks," *Mobiquitous*, Vol. 00, 2004, pp. 42-51.

[21] R. Balachandran, B. Ramamurthy, X. Zou and N. Vinodchandran, "CRTDH: An Efficient Key Agreement Scheme for Secure Group Communications in Wireless ad Hoc Networks," *Proceedings of IEEE International Conference on Communications*, Vol. 2, 2005, pp. 1123-1127.

[22] A. Shamir, "How to Share a Secret," *Communication ACM*, Vol. 22, No.11, 1979, pp. 612-613.

