Efficient Group Key Transfer Protocol for WSNs

Ching-Fang Hsu, Lein Harn, Tingting He, and Maoyuan Zhang

***Abstract*— Special designs are needed for cryptographic schemes in wireless sensor networks (WSNs). This is because sensor nodes are limited in memory storage and computational power. The existing group key transfer protocols for WSNs using classical secret sharing require that a t-degree interpolating polynomial be computed in order to encrypt and decrypt the secret group key. This approach is too computationally intensive. In this paper, we propose a new group key transfer protocol using a linear secret sharing scheme and factoring assumption. The proposed protocol can resist potential attacks and also significantly reduce the computation complexity of the system while maintaining low communication cost. Such a scheme is desirable for secure group communications in WSNs, where portable devices or sensors need to reduce their computation as much as possible due to battery power limitations.**

***Index Terms*— Group key transfer protocol, secret sharing, LSSS based on Vandermonde matrix, wireless sensor networks.**

I. Introduction

WIRELESS sensor networks (WSNs) have been developed in wide range of data acquisitions in battle fields, human body [1], [2], hazardous environments, etc. Most sensor nodes are small, low-cost, and low-power devices [3]. Sensors are randomly deployed without knowing their locations in prior of the deployment. Since sensors are low-cost, limited in both memory storage and computational power, it is a challenging research problem to develop cryptographic schemes suitable for WSNs.

Most research papers in WSNs propose schemes to establish pairwise keys for sensors. We can classify key establishment schemes in WSNs into two types, the *probabilistic schemes* and the *deterministic schemes*. The probabilistic scheme do not guarantee connectivity in WSNs. Eschenauer and Gligor proposed [4] the first *Random Key Pre-distribution scheme*. In their scheme, each sensor is pre-loaded with a key ring of k keys randomly selected from a large pool S of keys. After the deployment, if two neighbors share at least one key, they can establish a secure link in which the encryption key is one of the common keys. Otherwise, they should determine a secure path which is composed by successive secure links. The values of the key ring size k and the key pool size $|S|$ are chosen in such a way that the intersection of two key rings is not empty with a high probability. However, if the sensors are progressively corrupted, the attacker may discover a large part or the global key pool. Hence, a great number of links will be compromised. Chan et al. [5] proposed a protocol called *Q-composite scheme* that enhances the resilience of the random key scheme. In this solution, two neighboring nodes can establish a secure link only if they share at least Q keys. This approach enhances the resilience against node capture attacks because the attacker needs more overlap keys to break a secure link. However, this approach degrades the network secure connectivity coverage because neighboring nodes must have at least Q common keys to establish a secure link. Chan et al. [5] proposed a pairwise key pre-distribution scheme to protect the resilience against node capture and each captured node does not reveal any information about external links. The main drawback of their scheme is the non-scalability because the number of the stored keys depends linearly on the network size. This property will cause implementation issue of the scheme if the number of sensors in network is very large. Du et al. [6] proposed an enhanced random scheme with the node deployment knowledge. However, the application of this scheme is restrictive if the deployment knowledge is not possible. Rasheed and Mahapatra [7] proposes two key pre-distribution schemes based on the polynomial pool-based key pre-distribution scheme, the probabilistic generation key pre-distribution scheme, and the Q-composite scheme. Their schemes perform better in terms of network resilience to node capture than existing schemes if used in WSNs with mobile sinks. In 2013, Ruj et al. [8] proposed the first triple key establishment in WSNs. Three sensors can establish unique triple keys among them. Recently, Li and Xiong [9] proposed a heterogeneous online and offline signcryption scheme to secure communication between a sensor node and an Internet host. Their scheme is based on the bilinear pairing Which is a public-key-based approach.

The deterministic schemes do guarantee the connectivity in WSNs. Most deterministic schemes are based on threshold cryptography. Blom [10] proposed the first pairwise key establishment scheme based on threshold cryptography and Blundo et al. [11] further investigated the key establishment using polynomials. In Blum's scheme, every sensor node is preloaded with coefficients of a symmetric bivariate polynomial which is evaluated at one of its variables using its identification. The symmetry property of a polynomial allows every node to establish a pairwise key with every neighbor node. For an adversary to compromise a communication

Manuscript received January 11, 2016; revised March 2, 2016; accepted March 2, 2016. Date of publication March 4, 2016; date of current version April 26, 2016. This work was supported in part by the Research Funds of Central China Normal University within the Ministry of Education under Grant CCNU15ZD003 and Grant CCNU15A02018 and in part by the Major Project of National Social Science Fund under Grant 12&2D223. The associate editor coordinating the review of this paper and approving it for publication was Prof. Subhas C. Mukhopadhyay. *(Corresponding author: Ching-Fang Hsu.)*

C.-F. Hsu, T. He, and M. Zhang are with the Computer School, Central China Normal University, Wuhan 430079, China (e-mail: cherryjingfang@gmail.com; tthe@mail.ccnu.edu.cn; zhangmyccnu@126.com).

L. Harn is with the Department of Computer Science Electrical Engineering, University of Missouri–Kansas City, Kansas City, MO 64110 USA (e-mail: harnl@umkc.edu).

Digital Object Identifier 10.1109/JSEN.2016.2538292

link between two non-compromised nodes, it must capture at least a certain number of sensors (i.e., the threshold) to reconstruct the bivariate polynomial from its shares stored in the nodes and then break the system. For a polynomial of degree t, the scheme provides unconditionally secure if no more than $t-1$ sensors collude. Liu et al. [12] developed a general framework for pairwise key establishment based on the polynomial-based key pre-distribution protocol [11] and the probabilistic key distribution in [4] and [5]. Their scheme provided a higher probability for non-compromised sensors to establish secure communication links than that demonstrated in previous schemes. Khan et al. [13] proposed a pre-distribution scheme using a symmetric matrix and a generator matrix of maximum rank distance to establish pairwise keys for sensor nodes.

Secret sharing which was first introduced by both Blakley [15] and Shamir [27] independently in 1979 has been used to design group key transfer protocols for WSNs. There are two different approaches using secret sharing: one assumes a trusted offline server which is active only at initialization [16], [17], [20], [28] and the other assumes an online trusted server, called the key generation center, which is always active. The first type of approach is also called a key pre-distribution scheme. The main disadvantage of this approach is to require every user to store a large number of secrets. The second type of approach requires an online server to be active [26], in which the trusted KGC broadcasts group key information to all group members at once. This approach is similar to the model used in the IEEE 802.11i standard [23]. In 1989, Laih et al. [26] proposed the first algorithm based on this approach using any (t, n) secret sharing scheme to distribute a group key to a group consisting of $(t-1)$ members. Later, there are some papers [18], [25], [28] following the same concept to propose ways to distribute group messages to multiple users. Until [21] proposed a novel group key transfer protocol using (t, n) secret sharing that provided confidentiality and authentication, where KGC and each group member need to compute a t-degree interpolating polynomial to encrypt and decrypt the secret group key.

Linear secret sharing schemes (LSSSs) can be seen as a natural and useful generalization of threshold secret sharing schemes (TSSSs) and have been received considerable attention [14], [22], [24], [29], [31]. In this paper, we extend group key transfer protocols using secret sharing from TSSS to LSSS and propose an efficient protocol using LSSS that can resist potential attacks and also provide lower computational complexity while maintaining low communication complexity for secure group communications. The similar idea of using LSSS based on Vandermonde Matrix to achieve privacy was employed in [29] and [30]. The major difference between their scheme and ours is that our scheme allows lower computational complexity with factoring assumption, whereas the other schemes needs additional computational complexity by adding DH key agreement or ElGamal encryption algorithm. Hence, the proposed scheme is desirable for wireless sensor networks (WSNs), where portable devices or sensors need to reduce their computations as much as possible due to battery power limitations.

The rest of this paper is organized as follows: In the next section, we provide some preliminaries. In Section 3, we propose our group key transfer protocol. In Section 4, we prove the LSSS used in the proposed protocol is perfect and ideal. We analyze the security of our proposed protocol in Section 5. Performance evaluation of the proposed scheme is discussed in Section 6. We conclude in Section 7.

II. Preliminaries

In this section we review some basic definitions concerning factoring problem and linear secret sharing schemes.

A. Factoring Problem

Definition 1 (Factoring Problem): Let us choose two large safe primes p and q (i.e., primes such that $p' = \frac{p-1}{2}$ and $q' = \frac{q-1}{2}$ are also primes) and compute $m = pq$. m is made publicly known. Factoring problem is defined to compute factors p and q such that $m = pq$.

Definition 2 (Factoring Assumption): It is computationally intractable to solve the Factoring Problem.

B. Linear Secret Sharing Schemes

In a secret sharing scheme, a secret s is divided into n *shares* and shared among a set of n shareholders by a mutually trusted *dealer* in such a way that authorized subsets of shareholders can reconstruct the secret but unauthorized subsets of shareholders cannot determine the secret. If any unauthorized subset of shareholders can not obtain any information about the secret, then the scheme is called perfect. The set of authorized subsets of shareholders is called access structure and the set of unauthorized subsets of shareholders is called prohibited structure.

Karchmer and Wigderson [24] introduced monotone span programs (MSP) as linear models computing monotone Boolean functions. Beimel [14] proved that devising a linear secret sharing scheme (LSSS) for an access structure Γ is equivalent to constructing an MSP computing the monotone Boolean function f_Γ which satisfies $f_\Gamma(\vec{\delta}_A) = 1$ if and only if $A \in \Gamma$.

LSSS based on Vandermonde Matrix is introduced by Hsu et al. in [22]. Suppose that $\bar{V} = K^{n+1}$ is the $(n+1)$ dimensional linear space over a finite field K. The characteristic $char(K) = p$ and p is a safe large prime. Given a basis $\{e_1, \ldots, e_{n+1}\}$ of $\bar{V}$ with $\vec{e}_i = (0, \ldots, 0, \overset{i}{1}, 0, \ldots, 0) \in K^{n+1}$ for $1 \le i \le n+1$, the mapping $\mathbf{v} : K \to \bar{V}$ defined by $\mathbf{v}(x) = \sum_{i=1}^{n+1} x^{i-1} e_i = (1, x, x^2, \ldots, x^n)$ is determined. For $x_i \in K (i \in \{1, \ldots, n+1\})$, the Vandermonde Matrix V_{n+1} can be represented as follows

$$V_{n+1} = \begin{pmatrix} \mathbf{v}(x_1) \\ \mathbf{v}(x_2) \\ \cdots \\ \mathbf{v}(x_{n+1}) \end{pmatrix} = \begin{pmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^n \\ 1 & x_2 & x_2^2 & \ldots & x_1^n \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & x_{n+1} & x_{n+1}^2 & \ldots & x_{n+1}^n \end{pmatrix}$$

In LSSS based on Vandermonde Matrix, there are $(n+1)$ shareholders $P = \{P_0, P_1, \ldots, P_n\}$ and a mutually trusted dealer D, and the scheme consists of two algorithms:

1. *Share generation algorithm* the dealer D first picks a Vandermonde Matrix V_{n+1} and a random vector $\mathbf{r} = (r_0, r_1, r_2, \ldots, r_n) \in \bar{V}$ and let $\mathbf{r}$ be public, in which the secret $S = s_0 + s_1 + \cdots + s_n$ and all computations are performed in the finite field K, and D computes:

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^n \\ 1 & x_2 & x_2^2 & \ldots & x_1^n \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & x_{n+1} & x_{n+1}^2 & \ldots & x_{n+1}^n \end{pmatrix} \begin{pmatrix} r_0 \\ r_1 \\ \ldots \\ r_n \end{pmatrix} = \begin{pmatrix} s_0 \\ s_1 \\ \ldots \\ s_n \end{pmatrix}.$$

Then, the algorithm outputs a list of $(n + 1)$ shares $(x_0, x_1, \ldots, x_n)$ and distributes each share x_i to corresponding shareholder P_i secretly.

2. *Secret reconstruction algorithm* this algorithm takes all $(n+1)$ shares $(x_0, x_1, \ldots, x_n)$ and the public vector $\mathbf{r}$ as inputs, and outputs the secret $S = s_0 + s_1 + \cdots + s_n$ by computing each inner product $(\mathbf{v}(x_i), \mathbf{r}) = s_i$.

We note that because every set of at most $(t + 1)$ vectors of the form $\mathbf{v}(x)$ is linearly independent, the above scheme satisfies the basic requirements of secret sharing scheme as follows: (1) With knowledge of all $(n + 1)$ shares, it can reconstruct the secret S easily; (2) With knowledge of fewer than $(n + 1)$ shares, it cannot get *any* information about the secret S. LSSS based on Vandermonde Matrix is *information-theoretically secure* since the scheme satisfies these two requirements without making any computational assumption. For more information on this scheme, readers can refer to the original paper [22].

III. Design Principles

In this section, we describe the model of our group key distribution protocol and the security goals for this protocol.

A. Model

In our design, KGC firstly need to share a secret with each member of the group secretly. Then, these shared secrets determine a group of linearly independent vectors, where the number of these vectors is equal to the number of group members. Further, KGC can select a session key and separately compute the inner products of these vectors and a random vector determined by all group members. Afterwards, KGC publishes each value of the session key minus each inner product, where the number of those public values is equal to the number of group members. On the other hand, each group member is able to use his/her secret and the related public value to reconstruct the session key. Finally, all group members share a common session key for group communications.

B. Security Goals

The main security goals for our group key distribution protocol are: 1) key freshness; 2) key confidentiality; and 3) key authentication.

Key freshness is to ensure that a group key has never been used before. Thus, a compromised group key cannot cause any further damage of group communication. Key confidentiality is to protect the group key such that it can only be recovered by authorized group members; but not by any un-authorized user. Key authentication is to provide assurance to authorized group members that the group key is distributed by KGC; but not by an attacker.

IV. The Proposed Protocol

The proposed group key transfer protocol consists of three processes: initialization of KGC, user registration, and group key generation and distribution. Suppose that a set of users is $\{1, \ldots, n\}$ and the group is $\{1, \ldots, t\}$, where $n \geq t$. The detailed description is as follows:

A. Initialization of KGC

The KGC randomly chooses two safe large primes p and q (i.e., primes such that $p' = \frac{p-1}{2}$ and $q' = \frac{q-1}{2}$ are also primes) and computes $m = pq$, where $m \gg n$. m is made publicly known.

B. User Registration

Each user is required to register at KGC for subscribing the key distribution service. The KGC keeps tracking all registered users and removing any unsubscribed users. During registration, KGC shares a secret, $x_i \in K$, with each user i, where K is a finite field with the characteristic $char(K) = m$ and $x_i \neq x_j$ for any $i \neq j$, $i, j \in \{1, \ldots, n\}$. Thus, a secure channel is needed initially to share this secret with each user. From the shared secret x_i, n values $(x_i)^k$ for $k = 1, \ldots, n$ can be computed and saved by KGC and each user i, which will be used to access $\mathbf{v}(x_i)$ later. Then, KGC will transport the group key and interact with all group members in a broadcast channel.

C. Group Key Generation and Distribution

Suppose that $\bar{V} = K^{t+1}$ is the $(t + 1)$ dimensional linear space over K. Given a basis $\{e_1, \ldots, e_{t+1}\}$ of $\bar{V}$ with $\vec{e}_i = (0, \ldots, 0, \overset{i}{1}, 0, \ldots, 0) \in K^{t+1}$ for $1 \leq i \leq t + 1$, the mapping $\mathbf{v} : K \to \bar{V}$ defined by $\mathbf{v}(x) = \sum_{i=1}^{t+1} x^{i-1} e_i = (1, x, x^2, \ldots, x^t)$ is determined. Upon receiving a group key generation request from any user, KGC needs to access all vectors $\mathbf{v}(x_i)$ for $1 \leq i \leq t$ and randomly select a group key. KGC will distribute this group key to all group members in a secure and authenticated manner. All communications between KGC and group members are in a broadcast channel. For example, we assume that a group consists of t members, $\{1, \ldots, t\}$, and shared secrets are $x_1, x_2, \ldots, x_t$, and the corresponding vectors are $\mathbf{v}(x_1), \mathbf{v}(x_2), \ldots, \mathbf{v}(x_t)$. The key generation and distribution process contains five steps.

- Step 1. The initiator sends a key generation request to KGC with a list of group members as $\{1, \ldots, t\}$.
- Step 2. KGC broadcasts the list of all participating members, $\{1, \ldots, t\}$, as a response.
- Step 3. Each participating group member needs to send a random challenge, $R_i \in K$, to KGC.
- Step 4. KGC randomly selects a group key $K_G \in K$ and a random value $R_0 \in K$. KGC also computes t additional

values, $U_i = (K_G - K_i) \bmod m$ for $i = 1, \ldots, t$, and $Auth = h(K_G, 1, \ldots, t, R_0, R_1, \ldots, R_t, U_1, \ldots, U_t)$, where the vector $\vec{r} = (R_0, R_i, \ldots, R_t) \in K^{t+1}$, the inner product $(\mathbf{v}(x_i), \vec{r}) = K_i$ and h is a one-way hash function. KGC broadcasts $\{Auth, R_0, U_i\}$, for $i = 1, \ldots, t$, to all group members. All computations are performed in Z_m^*.

- Step 5. For each group member, i, knowing the public value, U_i, is able to compute the inner product $(\mathbf{v}(x_i), \vec{r}) = K_i$ and recover the group key $K_G = (U_i + K_i) \bmod m$. Then, i computes $h(K_G, 1, \ldots, t, R_0, R_1, \ldots, R_t, U_1, \ldots, U_t)$ and checks whether this hash value is identical to $Auth$. If these two values are identical, i authenticates the group key is sent from KGC.

V. Security Analysis

Adversaries can be categorized into two types. The first type of adversaries is outsiders of a particular group. The outside attacker can try to recover the secret group key belonging to a group that the outsider is unauthorized to know. This attack is related to the confidentiality of group key. In our proposed protocol, anyone can send a request to KGC for requesting a group key service. The outside attacker may also impersonate a group user to request a group key service. In security analysis, we will show that the outside attacker gains nothing from this attack since the attacker cannot recover the group key. The second type of adversaries is insiders of a group who are authorized to know the secret group key; but inside attacker attempts to recover other member's secret shared with KGC. Since any insider of a group is able to recover the same group key, we need to prevent inside attacker knowing other member's secret shared with KGC.

Theorem 1: The proposed protocol achieves the following security goals: 1) key freshness, 2) key confidentiality, and 3) key authentication.

Proof: We assume that a group consists of t members, $\{1, 2, \ldots, t\}$, and shared secrets are $x_1, x_2, \ldots, x_t$. The proposed protocol achieves the following security goals:

1) Key freshness is ensured by KGC since a random group key is selected by KGC for each service request. In addition, the equation $K_G = (U_i + K_i) \bmod m$ used to recover the group key is a function of random challenge selected by each group member and random value $R_0 \in K$ selected by KGC.
2) Key confidentiality is provided due to the security features of the proposed LSSS. KGC randomly selects a group key K_G and makes t values, $U_i = (K_G - K_i) \bmod m$ for $i = 1, \ldots, t$, publicly known. For each authorized group member, including the secret shared with KGC, he/she knows the inner product $(\mathbf{v}(x_i), \vec{r}) = K_i$. Thus, any authorized group member is able to recover the secret group key $K_G = (U_i + K_i) \bmod m$. However, for any unauthorized member (or outsider), there are only t values $U_i = (K_G - K_i) \bmod m$ for $i = 1, \ldots, t$ available and he obtains no information on K_i and $\sum_{1 \le i \le t} K_i$. Thus, unauthorized member knows nothing about the group key. This property is information theoretically secure since there has no other computational assumption based upon.
3) Key authentication is provided through the value $Auth$ in step 4. $Auth$ is a one-way hash output with the secret group key and all members' random challenges as input. Since the group key is known only to authorized group members and KGC, unauthorized members cannot forge this value. Any insider also cannot forge a group key without being detected since the group key is a function of the secret shared between each group member and KGC. In addition, any replay of $\{Auth, R_0, U_i\}$, for $i = 1, \ldots, t$, of KGC in step 4 can be detected since the group key is a function of each group member's random challenge.

Theorem 2 (Outsider Attack): Assume that an attacker who impersonates a group member for requesting a group key service, then the attacker can neither obtain the group key nor share a group key with any group member.

Proof: Although any attacker can impersonate a group member to issue a service request to KGC without being detected and KGC will respond by sending group key information accordingly; however, the group key can only be recovered by any group member who shares a secret with KGC. This security feature is information theoretically secure, which is ensured by the proposed LSSS.

If the attacker tries to reuse a compromised group key by replaying previously recorded key information from KGC, this attack cannot succeed in sharing this compromised group key with any group member since the group key is a function of each member's random challenge and the secret shared between group member and KGC. A compromised group key cannot be reused if each member selects a random challenge for every conference.

Theorem 3 (Insider Attack): Assume that the protocol runs successfully v times and the applied factoring instances are intractable, then the secret $x_i \in K$ of each group member shared with KGC remains unknown to all other group members (and outsiders).

Proof: For a group key service request, KGC randomly selects a group key K_G and makes t values, $U_i = (K_G - K_i) \bmod m$ for $i = 1, \ldots, t$, publicly known. For each authorized group member, with knowledge of the secret shared with KGC and t public information, he/she knows U_i and is able to compute the inner product $(\mathbf{v}(x_i), \vec{r}) = K_i$. Thus, any authorized group member is able to reconstruct the group key $K_G = (U_i + K_i) \bmod m$, where the vector $\vec{r} = (R_0, R_i, \ldots, R_t) \in K^{t+1}$. However, the secret $x_i \in K$ of each group member shared with KGC remains unknown to outsiders.

In our proposed protocol, group key service requests from group members are not authenticated. An adversary (insider) can make several service requests to KGC and forge challenges of the target group member. For example, the adversary makes two service requests for a group containing the adversary and the target group member. The adversary also forges the challenges of the target group member for these two services. The KGC generates the group keys K_{G1} and K_{G2} respectively. Thus, the adversary can obtain the inner products

$(\mathbf{v}(x_{t\arg et}), \vec{r}_1) = K_{t\arg et1} (\text{mod } m)$ and $(\mathbf{v}(x_{t\arg et}), \vec{r}_2) = K_{t\arg et2} (\text{mod } m)$. By subtracting these two inner products, the adversary obtains a tth degree equation as

$$f(x_{t\arg et}) = (\mathbf{v}(x_{t\arg et}), \vec{r}_1 - \vec{r}_2) = K_{t\arg et1} - K_{t\arg et2} (\text{mod } m),$$

where $\mathbf{v}(x_{t\arg et}) = \sum_{i=1}^{t+1} (x_{t\arg et})^{i-1} e_i$. It is commonly believed that the adversary needs to first solve two separate equations in $f(x_{t\arg et}) = K_{t\arg et1} - K_{t\arg et2} (\text{mod } p)$ and $f(x_{t\arg et}) = K_{t\arg et1} - K_{t\arg et2} (\text{mod } q)$, respectively, in order to solve the secret $x_{t\arg et}$. This is an intractable problem due to factoring assumption. Some well-known modern cryptosystems are also based on the same assumption. For example, the security of Rabin's cryptosystem and the security of RSA cryptosystem.

VI. Performance Evaluation

In this section, we will firstly compare our scheme with public-key-based key distribution protocols. Then, we compare ours with a threshold secret-sharing-based key distribution protocol [21] proposed recently, in terms of computational and communication costs.

A. Comparison 1

In comparing with the public-key-based key distribution protocols, our scheme has the following advantages:

(a) Instead of using public-key encryptions in which the security is based on some computation assumptions, we use the secret sharing as the tool of broadcast encryption in which the security is unconditionally secure. In addition, instead of performing encryption one at a time, our scheme can perform encryption all at once to reduce computational complexity.

(b) The public-key-based key distribution protocol requires larger rekeying overheads when membership of any user has changed since broadcasting keys need to be updated. But our scheme uses secret sharing and KGC can manage any membership change efficiently. There is no rekeying issue.

B. Comparison 2

1) Time Complexities: Let TM, TI and TH be execution time for performing a modular multiplication, a modular inverse and the one-way hash function H, respectively. As compared to TM or TI, the time for performing modular addition or subtraction required in the proposed scheme can be ignored.

a) The proposed scheme: In the initial phase of user registration, the time complexity for computing $(x_i)^k$ for $k = 1, \ldots, n$ by each user i is $(n-1) \times \text{TM}$, and the time complexity for computing $(x_i)^k$ for $k = 1, \ldots, n$ and $i = 1, \ldots, n$ by KGC is $n \times (n-1) \times \text{TM}$. For each group key transfer, when a group key $K_G \in K$ and a random value $R_0 \in K$ are selected, the time complexity for distributing the group key by KGC is $t \times (t+1) \times \text{TM} + \text{TH}$. Then, the time complexity for recovering the group key by each group member is $(t+1) \times \text{TM} + \text{TH}$.

TABLE I

Computational Comparison of the Proposed Scheme and Harn's Scheme in Each Group Key Distribution

Scheme	Distributing the group key	Recovering the group key
The proposed scheme	$t\times(t+1)\times$TM+TH	$(t+1)\times$TM+TH
Harn's scheme	$t\times(t+1)\times t\times$(TM+TI)+TH	$(t+1)\times t\times$(TM+TI)+TH

TABLE II

Comparison of Communication Costs Between the Proposed Scheme and Harn's Scheme

Scheme	User registration	Group key generation and distribution	Total
The proposed scheme	$n \mid pq \mid$	$t \mid pq \mid +(t+1) \mid pq \mid + \mid H \mid$	$(n+2t+1) \mid pq \mid + \mid H \mid$
Harn's scheme	$2n \mid pq \mid$	$t \mid pq \mid +2t \mid pq \mid + \mid H \mid$	$(2n+3t) \mid pq \mid + \mid H \mid$

b) Harn's scheme: For each group key transfer, when a group key k is selected, the time complexity for distributing the group key by KGC is $t \times (t+1) \times t \times (\text{TM} + \text{TI}) + \text{TH}$. Then, the time complexity for recovering the group key by each group member is $(t+1) \times t \times (\text{TM} + \text{TI}) + \text{TH}$.

Computational comparison of the proposed scheme and Harn's scheme [21] is shown in Table 1. From Table 1, as compared with Harn's scheme, in the initial phase of user registration, the proposed scheme requires $(n-1)$ additional modular multiplication operations for each user, and $n \times (n-1)$ additional modular multiplication operations for KGC. However, in each group key transfer, the proposed scheme using LSSS causes a significant decrease of the computational complexity of system.

2) Communication Costs: The communication costs required in the proposed scheme are measured by the total volume of data transmission during user registration and group key generation and distribution, respectively. From $m = pq$, $|pq|$ is the size of the adopted finite field Z_m^* and $|H|$ is the output size of one-way hash function H. Obviously, the communication cost for user registration is $n|pq|$. The communication cost for group key generation and distribution is $t|pq| + (t+1)|pq| + |H|$.

In Harn's scheme, for the same number n of users, the communication costs required in user registration is $2n|pq|$ and the communication cost for group key generation and distribution is $t|pq| + 2t|pq| + |H|$.

Comparison of communication costs between the proposed scheme and Harn's scheme [21] is shown in Table 2. It can be seen that in the proposed scheme using LSSS, the communication costs required can be reduced by $n|pq| + (t-1)|pq|$.

VII. Conclusions

We have proposed an efficient group key transfer protocol based on a linear secret sharing for wireless sensor networks (WSNs). We provide group key authentication.

Security analysis for possible attacks is included. As a result, this protocol can resist potential attacks and also significantly reduce the overhead of system implementation. This protocol is suitable for mobile communications.

REFERENCES

[1] T. Gao, D. Greenspan, M. Welsh, R. R. Juang, and A. Alm, "Vital signs monitoring and patient tracking over a wireless network," in *Proc. IEEE 27th Annu. Int. Conf. Eng. Med. Biol. Soc. (IEEE-EMBS)*, Jan. 2006, pp. 102–105.

[2] L. Gu *et al.*, "Lightweight detection and classification for wireless sensor networks in realistic environments," in *Proc. 3rd ACM Conf. Embedded Netw. Sensor Syst.*, Nov. 2005, pp. 205–217.

[3] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, vol. 43, no. 5, pp. 51–58, 2000.

[4] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proc. 9th ACM Conf. CCS*, 2002, pp. 41–47.

[5] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proc. IEEE Symp. SP*, May 2003, pp. 197–213.

[6] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in *Proc. IEEE INFOCOM*, Mar. 2004, pp. 586–597.

[7] A. Rasheed and R. Mahapatra, "Key predistribution schemes for establishing pairwise keys with a mobile sink in sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 1, pp. 176–184, Jan. 2011.

[8] S, Ruj, A, Nayak, and I. Stojmenovic, "Pairwise and triple key distribution in wireless sensor networks with applications," *IEEE Trans. Comput.*, vol. 62, no. 11, pp. 2224–2237, Nov. 2013.

[9] F. Li and P. Xiong, "Practical secure communication for integrating wireless sensor networks into the Internet of Things," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3677–3684, Oct. 2013.

[10] R. Blom, "Non-public key distribution," in *Advances in Cryptology*, D. Chaum, R. L. Rivest, and A. T. Sherman, Eds. New York, NY, USA: Plenum, 1982, pp. 231–236.

[11] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly secure key distribution for dynamic conferences," *Inf. Comput.*, vol. 146, no. 1, pp. 1–23, 1998.

[12] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in *Proc. 10th ACM Conf. Comput. Commun. Secur. (CCS)*, Oct. 2003, pp. 52–61.

[13] E. Khan, E. Gabidulin, B. Honary, and H. Ahmed, "Matrix-based memory efficient symmetric key generation and pre-distribution scheme for wireless sensor networks," *IET Wireless Sensor Syst.*, vol. 2, no. 2, pp. 108–114, Jun. 2012.

[14] A. Beimel, "Secure schemes for secret sharing and key distribution," M.S. thesis, Faculty Comput. Sci., Technion-Israel Inst. Technol., Haifa, Israel, 1996.

[15] G. R. Blakley, "Safeguarding cryptographic keys," in *Proc. Nat. Comput. Conf. Amer. Fed. Inf. Process. Soc. (AFIPS)*, vol. 48. 1979, pp. 313–317.

[16] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly secure key distribution for dynamic conferences," *Inf. Comput.*, vol. 146, no. 1, pp. 1–23, Oct. 1998.

[17] R. Blom, "An optimal class of symmetric key generation systems," in *Proc. EUROCRYPT Workshop Adv. Cryptol.*, 1984, pp. 335–338.

[18] S. Berkovits, "How to broadcast a secret," in *Proc. EUROCRYPT Workshop Adv. Cryptol.*, 1991, pp. 536–541.

[19] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.

[20] A. Fiat and M. Naor, "Broadcast encryption," in *Proc. 13th Annu. Int. Cryptol. Conf. Adv. Cryptol. (CRYPTO)*, 1994, pp. 480–491.

[21] L. Harn and C. Lin, "Authenticated group key transfer protocol based on secret sharing," *IEEE Trans. Comput.*, vol. 59, no. 6, pp. 842–846, Jun. 2010.

[22] C.-F. Hsu, Q. Cheng, X. Tang, and B. Zeng, "An ideal multi-secret sharing scheme based on MSP," *Inf. Sci.*, vol. 181, no. 7, pp. 1403–1409, 2011.

[23] *IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Medium Access Control (MAC) Security Enhancements*. IEEE Standard 802.11i-2004, IEEE Computer Society, 2004.

[24] M. Karchmer and A. Wigderson, "On span programs," in *Proc. 8th Annu. Conf. Struct. Complex.*, San Diego, CA, USA, May 1993, pp. 102–111.

[25] C.-H. Li and J. Pieprzyk, "Conference key agreement from secret sharing," in *Proc. 4th Australasian Conf. Inf. Secur. Privacy (ACISP)*, 1999, pp. 64–76.

[26] C. S. Laih and J. Y. Lee, "A new threshold scheme and its application in designing the conference key distribution cryptosystem," *Inf. Process. Lett.*, vol. 32, no. 3, pp. 95–99, 1989.

[27] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[28] G. Sáez, "Generation of key predistribution schemes using secret sharing schemes," *Discrete Appl. Math.*, vol. 128, no. 1, pp. 239–249, 2003.

[29] C. Hsu, B. Zeng, G. Cui, and L. Chen, "A new secure authenticated group key transfer protocol," *Wireless Pers. Commun.*, vol. 74, no. 2, pp. 457–467, 2014.

[30] C. Hsu, B. Zeng, and M. Zhang, "A novel group key transfer for big data security," *Appl. Math. Comput.*, vol. 249, pp. 436–443, Dec. 2014.

Ching-Fang Hsu was born in Hubei, China, in 1978. She received the M.Eng. and Ph.D. degrees in information security from the Huazhong University of Science and Technology, Wuhan, China, in 2006 and 2010, respectively. From 2010 to 2013, she was a Research Fellow with the Huazhong University of Science and Technology. She is currently an Assistant Professor with Central China Normal University, Wuhan, China. Her research interests are in cryptography and network security, especially in secret sharing and its applications.

Lein Harn received the B.Sc. degree in electrical engineering from National Taiwan University in 1977, the M.S. degree in electrical engineering from the State University of New York, Stony Brook, in 1980, and the Ph.D. degree in electrical engineering from the University of Minnesota in 1984. He joined the Department of Electrical and Computer Engineering, University of Missouri-Columbia, as an Assistant Professor in 1984. In 1986, he moved to the Computer Science and Telecommunication Program at the University of Missouri–Kansas City (UMKC). While at UMKC, he went on development leave to work with the Racal Data Group, FL, for a year. His research interests are cryptography, network security, and wireless communication security. He has authored number of papers in digital signature design and applications, and wireless and network security. He has written two books on security. He is investigating new ways of using digital signature in various applications. In 2015, he was appointed as a Chu-Tian Researcher with the School of Computer Science and Technology, Hubei University of Technology, China.

Tingting He received the Ph.D. degree from Central China Normal University, in 2003. She is currently a Professor with Central China Normal University, Wuhan, China. Her research interests are in natural language processing and network security.

Maoyuan Zhang received the Ph.D. degree in computer science from the Huazhong University of Science and Technology, Wuhan, China. He is currently a Professor with Central China Normal University, Wuhan. His research interests are in computer networks and network security.