Research Article

Lightweight Noninteractive Membership Authentication and Group Key Establishment for WSNs

Qi Cheng,1 Chingfang Hsu,2 and Lein Harn3

1Product Engineering Department, Wuhan Digital Engineering Institute, Wuhan 430074, China
2Computer School, Central China Normal University, Wuhan 430079, China
3Department of Computer Science Electrical Engineering, University of Missouri-Kansas City, Kansas City, 64110 MO, USA

Correspondence should be addressed to Chingfang Hsu; cherryjingfang@gmail.com

Received 8 January 2020; Revised 28 March 2020; Accepted 30 April 2020; Published 16 May 2020

Academic Editor: Salvatore Strano

Wireless sensor networks (WSNs) exhibit their potential capacity in the next generation of mobile communication networks and wireless systems (5G). Collected data in WSNs are different from most data transmitted in digital communication applications. Most collected data in WSNs contain only few bits of information. Conventional protocols are not suitable for WSNs since this environment needs more flexible and lightweight protocols for secure group communications. Hence, how to realize the mutual secure and lightweight communication is a big challenge for WSNs. User authentication and key establishment are two fundamental security services in secure communications for WSNs. In this paper, we propose a novel design which embeds the function of membership authentication and group key establishment in WSNs. By using an asymmetric bivariate polynomial, membership authentication and pairwise shared keys distribution are realized. Then, each member mixes his/her input with pairwise shared keys with other members and releases the encrypted value in a broadcast channel. After collecting all released values, each member can compute the group key efficiently. Our proposal is noninteractive and lightweight. As it enjoys low computation and communication costs compared with the state-of-the-art cryptographic solutions, this design is more suitable for efficient membership authentication and group key establishment in WSNs.

1. Introduction

Wireless Sensor Networks (WSNs) have been developed to collect data remotely for various applications [1, 2]. For example, data have been collected for traffic analysis, weather prediction, and medical analysis, etc. [3, 4]. For security reason, the collected data need to be protected from eavesdropping. Data encryption requires that both the source node and the receiver node share a pairwise shared key. The source node encrypts the collected data under the shared key, and the receiver node decrypts the ciphertext under the shared key to recover the data.

In general, security research studies in WSNs are focused on the development of key establishment and key management solutions. Random key predistribution schemes [5–7] have been developed to allow two sensors to establish a shared key. The random key distribution is a probabilistic scheme and does not guarantee connectivity in WSNs. Each sensor is preloaded with k keys randomly selected from a large pool of keys. Blom [8] proposed the first pairwise key establishment scheme based on threshold cryptography. This approach is a deterministic scheme which can guarantee connectivity in WSNs. Blundo et al. [9] have discussed the key establishment using polynomials. Khan et al. [10] proposed a predistribution scheme using a symmetric matrix and a generator matrix of maximum rank distance to establish pairwise keys for sensor nodes. Group key distribution based on bivariate polynomials [11–14] has also been developed to allow a group of sensors to establish a shared key deterministically. The design of WSNs has been classified into two types: flat and hierarchical. In flat WSNs, all sensors have the same capabilities to collect data and forward data to other sensors in the network. In hierarchical WSNs, devices are organized into a hierarchy based on their
Our proposed protocol is secure against inside attackers and outside attackers. Furthermore, confidentiality, authentication, freshness, forward secrecy, and backward secrecy of group key can be achieved.

One unique feature of our group key establishment is that the XOR operation is the main computation, so it is lightweight.

The organization of this paper is as follows. In Section 2, we provide some preliminaries about bivariate polynomials. In Section 3, we present the model of our protocols including the protocol description, types of adversaries, and security properties of our proposed protocol. Our proposed protocol including three parts (a) token generation, (b) membership authentication, and (c) group key establishment is given in Section 4. In Section 5, we analyze the security and performance of this protocol. The conclusion is given in Section 5.
2. Preliminaries

In Shamir’s $(t,n)$ SS [25], the dealer selects a univariate polynomial, $f(x)$, with degree $t−1$ and $f(0)=s$, where $s$ is the secret. The dealer generates shares, $f(x_i) \mod p$, $i = 1, 2, \ldots, n$, for shareholders, where $p$ is a prime with $p > s$ and $x_i$ is the public information associated with each shareholder, $U_i$. Each share, $f(x_i)$, is an integer in $\mathbb{GF}(p)$. Shamir’s $(t,n)$ SS satisfies security requirements of a $(t,n)$ SS, that is, (a) with $t$ or more than $t$ shares can reconstruct the secret and (b) with fewer than $t$ shares cannot obtain any information of the secret. Shamir’s SS is unconditionally secure.

In Shamir’s $(t,n)$ SS, shareholders cannot verify the validity of their shares obtained from the dealer. In 1985, Chor et al. [26] extended the notion of SS and proposed the first verifiable secret sharing (VSS). Verifiability is the property of a VSS which allows shareholders to verify their shares. Invalid shares may be caused either by the dealer during share generation or by channel noise during transmission. VSS is performed by shareholders after receiving their shares from the dealer and before using their shares to reconstruct the secret. If invalid shares have been detected, shareholders can request the dealer to regenerate new shares. There are many $(t,n)$ VSSs [27–32] using bivariate polynomials, denoted as BVSSs. A bivariate polynomial with degree $t−1$ can be represented as $F(x,y) = \sum_{i=0}^{t-1} \sum_{j=0}^{t-1} a_{i,j} x^i y^j \mod p$, where $a_{i,j} \in \mathbb{GF}(p)$. We can classify BVSSs into two types, the symmetric BVSSs, denoted as SBVSSs [28, 30, 32], and the asymmetric BVSSs, denoted as ABVSSs, [27, 29, 31]. If the coefficients satisfy $a_{i,j} = a_{j,i}, \forall i, j \in [0,t-1]$, it is a symmetric bivariate polynomial. Shares generated by a bivariate polynomial can be used to establish pairwise keys between any pair of shareholders. In all $(t,n)$ SBVSSs, the dealer selects a bivariate polynomial, $F(x,y)$, with degree $t−1$, and $F(0,0)=s$, where $s$ is the secret. The dealer generates shares, $F(x_i,y_j) \mod p$, $i = 1, 2, \ldots, n$, for shareholders, where $p$ is a prime with $p > s$ and $x_i$ is the public information associated with each shareholder, $U_i$. Each share, $F(x_i,y_j)$ is a univariate polynomial with degree $t−1$. Note that shares generated in a SBVSS satisfy $F(x_i,x_j) = F(x_j,x_i)$, $\forall i, j \in [0,t-1]$, and the pairwise key, $F(x_i,x_j) = F(x_j,x_i)$, can be established between the pair of shareholders, $U_i$ and $U_j$. In a similar way, in an ABVSS, the dealer generates a pair of shares, $F(x_i,y_j) \mod p$ and $F(x_j,x_i) \mod p$, $i = 1, 2, \ldots, n$, for each shareholder, and the pairwise secret key, $F(x_i,x_j)$ or $F(x_j,x_i)$, can also be established between the pair of shareholders, $U_i$ and $U_j$.

In this paper, we propose a novel design of efficient membership authentication and group key establishment for WSNs. Our design integrates solutions of membership authentication, pairwise shared key establishment, and group key establishment together. In other words, we propose to use a bivariate polynomial to generate tokens. Tokens of members obtained during registration can be used for (a) membership authentication; (b) pairwise shared keys distribution; and (c) group key establishment. However, most of the existing cryptographic solutions need additional membership authentication and shared keys distribution and also need interactive communications or complex computations for encryption and decryption [33–36].

3. Model of Our Proposed Protocol

In this section, we describe the model of our proposed membership authentication and group key agreement protocol for WSNs including the network model and security model, which provide the type of adversaries and security features.

3.1. Protocol Description for the Network Model. Without loss of generality, suppose that there has a mutually trusted membership registration center (MRC) and there are $n$ users, $\{U_1, U_2, \ldots, U_n\}$, involved in group communications. Each user is required to register at MRC, and MRC manages all registered users which includes removing any unsubscribed users or adding new users. In order to achieve secure communications, each group’s session key is needed to be securely distributed to all corresponding group members in prior of exchanging messages. Typically, if all participants are members and act honestly, the protocol is successful; i.e., only the members belonging to the same group can derive this group’s session key. Otherwise, it fails, i.e., group members obtain nothing. Thus, membership authentication before the group key establishment is necessary.

In our proposed protocol, each user needs to register at the MRC initially and obtain secret token. The MRC selects an asymmetric bivariate polynomial and generates tokens. Token of each user is two univariate polynomials: one is $t−1$ degree in $x$ and the other one is $h−1$ degree in $y$.

In order to establish a secure group communication involving $m$ (i.e., $2 \leq m < n$) members, it requires to execute a membership authentication first in which all participated users interact with each other to prove that they belong to the same group. In the membership authentication, each member needs to broadcast a random integer. After receiving all random integers, each member needs to use his secret tokens to compute pairwise shared keys and then compute a hash output as his authentication response. Members can use this authentication response to authenticate his membership. This membership authentication can also identify nonmembers. At the end of membership authentication, each member knows exactly the memberships of users participated in the secure group communication. Then, by using XOR operation function, each member mixes his/her input with pairwise shared keys and, after that, uses his/her pairwise shared keys to encrypt the computed value, and next, sends this value to other members. After collecting all values from other members, each member can compute the group key; that is, a secret group session key is obtained by each member individually. There is no interaction with other members to compute the group key. Thus, our proposed protocol is very efficient in both membership authentication and group key establishment since there is only broadcast transmission.
Furthermore, the computation of each member needs only polynomial evaluation, XOR computation, and hash function which are much faster than most public-key computations. We will give detail discussion for its performance evaluation in Section 5.

3.2. Security Model. Now, we introduce the security model which include the type of adversaries and the required security features for secure group communication. These security requirements will be analyzed in Section 5.

3.2.1. Type of Adversaries. We consider two types of attacks: inside and outside attacks. The inside attackers are legitimate members who have obtained valid tokens from MRC initially. From inside attack, colluded members try to recover the MRC’s secret polynomial used to generate tokens for members and then use these uncovered tokens to obtain group keys which they are not authorized to access. On the other hand, the outside attackers are illegitimate members who try to generate valid tokens of members and use them to impersonate members in a secure group communication or to recover secret group keys which they are not authorized to access. In Section 5, we will give the detailed security analysis about these two types of attackers.

3.2.2. Security Features. For secure group communication, the group key establishment protocol needs to have the following security features.

(a) Correctness: the protocol can successfully authenticate memberships of all participated users and then establish a secret group key among all members.

(b) Freshness of authentication response: the authentication responses generated by members in the membership authentication can only be used for one time. This feature can prevent replay attack in which attackers replay recorded authentication response to fail the membership authentication.

(c) Freshness of group keys: the secret group key generated by members in the key establishment can only be used for one time communication. This feature can prevent attackers to reuse previously compromised group keys to gain access to other secure communications.

(d) Freshness of the group key authentication: recording a previously used authentication cannot impersonate a member since this random integer is different in every session.

(e) Forward secrecy of group keys: the forward secrecy is ensured if a departing member cannot access the content of communications of any future group session.

(f) Backward secrecy of group keys: the backward secrecy is ensured if a new member cannot access the content of communications of any past session.

4. Our Proposed Protocol

In this section, we present a membership authentication and group key establishment protocol using an asymmetric bivariate polynomial and XOR operation function. The protocol is described in Algorithm 1. There are three phases in our protocol, i.e., token generation, membership authentication, group key establishment, and authentication. For every phase, we give the illustrative figure, respectively, in Figures 1–3.

5. Analysis

In this section, we address the security and performance of our proposed protocol.

5.1. Security Analysis. In this section, we discuss security features and possible attacks of our protocol as described in Section 3.2.

5.1.1. Security Features

(a) Correctness: In membership authentication, if all participated users are members as they claimed in Step 1 of membership authentication, each member, $U_i$, in Step 2 should be able to compute the pairwise shared key $k_{i,j}$. Thus, in Step 4, the authentication response, $\text{Auth}_{u_i} = h(k_{i,j} || r_i)$, can be used to verify $U_j$’s membership by $U_i$. Nonmembers cannot forge this authentication response since nonmembers do not know the secret tokens of member, $U_j$. In group key establishment, the correctness of this property comes from the rule of XOR operation and $q_v = s_i \oplus k_{i,1} \oplus k_{i,2} \oplus \cdots \oplus k_{i,m} \mod p$, where $j = 1, 2, \ldots, m, j \neq i$; thus, $q_{v_0} \oplus q_{v_1} \oplus \cdots \oplus q_{v_i}$ $\oplus \cdots \oplus q_{v_m} \mod p = s_1 \oplus s_2 \oplus \cdots \oplus s_i \oplus \cdots \oplus s_m \mod p = K_{i,j}$, $i = 1, 2, \ldots, m$.

(b) Freshness of authentication response: in Step 3 of membership authentication, the authentication response, $\text{Auth}_{u_i} = h(k_{i,j} || r_i)$, is a hash output of pairwise shared key and random integer selected by participated member initially. Recording a previously used authentication response cannot impersonate a member since this random integer is different in every session.

(c) Freshness of group keys: in the group key establishment, the group key, $K = s_1 \oplus s_2 \oplus \cdots \oplus s_i \oplus \cdots \oplus s_m \mod p$, is determined by $U_i$’s secret input $s_i$ initially. This group key is different in every session.

(d) Freshness of the group key authentication: in Step 6 of group key establishment, the authentication $H(K_i || L)$ is a one-way hash output with input $s$ group key determined by each member’s secret input $s_i$ and sum of random integers selected by the participated member initially. Recording a previously used authentication cannot impersonate a
Token generation
For n users \( U_1, U_2, \ldots, U_n \), the MRC selects a random asymmetric polynomial, \( F(x, y) = a_{0,0} + a_{1,0}x + a_{0,1}y + a_{1,1}xy + a_{2,0}x^2 + a_{0,2}y^2 + a_{1,2}xy^2 + a_{2,1}x^2y + a_{2,2}x^2y^2 + \cdots + a_{t-1,t-1}x^{t-1}y^{t-1} \mod p \), where \( F(x, y) \) is t degree in x and \( h-1 \) degree in \( y \) (i.e., \( h > 2t - 2 \)). We will prove this condition in Theorem 1 in \( y, a_{ij} \in GF(p), \forall i \in [0, t-1], \forall j \in [0, h-1] \), and \( p \) is a prime integer with \( p > n \). The MRC computes a pair of shares, \( s_i(y) = F(x_i, y) \) and \( s_j(x) = F(x_j, x) \), for each user, \( U_i, i = 1, 2, \ldots, n \), where \( x_i \notin \{0, 1\} \) is the public information associated with each user, \( U_i \). The MRC sends each pair of shares as \( U_i \)'s token, \((s_i(y), s_j(x))\), to user \( U_i \) through the secure channel.

Membership authentication
We assume that \( m \) (i.e., \( 2 \leq m < n \)) users, \( \{U_1, U_2, \ldots, U_m\} \), want to engage in a group key establishment in WSNs. 

**Step 1.** Each member \( U_i \) broadcasts a random integer, \( r_i \in GF(p) \) to all other members, where \( i = 1, 2, \ldots, m \).

**Step 2.** Assumption that the value \( F(x_i, x_j) \), with \( x_i < x_j \), is used as the pairwise shared key between the shareholders \( U_i \) and \( U_j \). Each member \( U_i \) uses one of the shares of his token, \( s_i(y) \) or \( s_j(x) \), to compute pairwise shared keys, \( k_{i,j} = s_i(x_j) = F(x_i, x_j) \), \( j \neq i \), between any other users, where \( k_{i,j} \) is the secret key shared between users, \( U_i \) and \( U_j \).

**Step 3.** Each member \( U_i \) computes authentication responses, \( Auth_i = h(k_{i,j}, r_j) \), \( j = 1, 2, \ldots, m, j \neq i \), where \( h(k_{i,j}, r_j) \) is a one-way hash output with \( k_{i,j} \) and \( r_j \), as inputs. Each \( Auth_i \) is sent to member \( U_i \) publicly for authentication.

**Step 4.** After receiving \( Auth_i = h(k_{i,j}, r_j) \), from member \( U_i \), the member \( U_j \) uses his computed pairwise shared key, \( k_{i,j} = s_i(x_j) = F(x_i, x_j) \), in Step 2 to compute \( h(k_{i,j}, r_j) \) and check whether \( Auth_i = h(k_{i,j}, r_j) \). If the checking is successful, member \( U_j \) has been authenticated; otherwise, member \( U_j \) has not been authenticated. Repeat this process for all other members \( U_{j,i} \), \( i = 1, 2, \ldots, m, i \neq j \).

**Group key establishment and authentication**
Let us assume that at the end of membership authentication, all \( m \) members, \( \{U_1, U_2, \ldots, U_m\} \), have been successfully authenticated. Then, members follow an XOR operation algorithm to complete the group key establishment process. However, all exchange information among members is encrypted under the pairwise shared keys, \( k_{i,j} \), \( j = 1, 2, \ldots, m, j \neq i \), in the Step 2 of membership authentication.

**Step 1.** Each member \( U_i \) needs to select a secret input \( s_i \in GF(p) \) and broadcasts a random integer, \( l_i \in GF(p) \) to all other members, where \( i = 1, 2, \ldots, m \).

**Step 2.** Each member \( U_i \) uses his pairwise shared keys with other members to compute \( q_{i,j} = s_i \oplus k_{i,j} \oplus k_{j,i} \oplus \cdots \oplus k_{j,m} \mod p, j = 1, 2, \ldots, m, j \neq i \), where \( \oplus \) is the XOR operation.

**Step 3.** Each member \( U_i \) uses his computed pairwise shared keys, \( k_{j,i} \), \( j = 1, 2, \ldots, m, j \neq i \), in the Step 2 of membership authentication to encrypt \( q_{i,j} \), as \( u_{ij} = E_{k_{j,i}}(q_{i,j}) \), \( j = 1, 2, \ldots, m, j \neq i \). Member \( U_i \) sends each \( u_{ij} \) to member \( U_j \).

**Step 4.** After receiving \( u_{ij} \), from another member, member \( U_j \) uses his computed pairwise shared key, \( k_{j,i} \), in the Step 2 of membership authentication to decrypt as \( q_{i,j} = E_{k_{j,i}}(u_{ij}) \). Repeat this process for all \( u_{ij} \), \( j = 1, 2, \ldots, m, j \neq i \).

**Step 5.** After obtaining \( q_{i,j}, j = 1, 2, \ldots, m, j \neq i \), from all other members, member \( U_i \) computes \( q_i \oplus q_1 \oplus \cdots \oplus q_{i-1} \oplus q_{i+1} \oplus \cdots \oplus q_m \mod p = s_i \oplus s_1 \oplus \cdots \oplus s_{i-1} \oplus s_{i+1} \oplus \cdots \oplus s_m \mod p = K_i, i = 1, 2, \ldots, m \), where \( \oplus \) is the XOR operation.

**Step 6.** Each member \( U_i \) computes and broadcasts \( H(K_i, L), i = 1, 2, \ldots, m \), and then checks if \( H(K_1, L) = H(K_2, L) = \cdots = H(K_m, L) \mod p \), where \( L = \sum_{j=1}^{m} l_j \) and \( H(K, L) \) is a one-way hash output with \( K_i \) and \( L \) as input s. If the checking is successful, the group key has been authenticated, \( K_i = K \) is the secret group communication key; otherwise, the group key has not been authenticated. Repeat this process for all group members \( U_{i,j} \), \( i = 1, 2, \ldots, m \).
member since this random integer is different in every session.

(e) Forward secrecy of group keys: if a member has departed from the group, the departed member cannot access the content of future communications since the any group key, $K$, can only be computed by members involved in the secure communication.

(f) Backward secrecy of group keys: if a member joins the group, the new member cannot access the content of any past communications since the any
5.1.2. Possible Attacks

**Theorem 1** (inside attack). In the proposed protocol, if \( h > 2t - 2 \), it needs at least \( t \) insider attackers to work together to reconstruct the tokens. The proposed protocol can resist up to \( t-1 \) colluded members to recover the secret polynomial \( F(x, y) \) of MRC.

**Proof.** Inside attackers are legitimate members who own valid tokens from the MRC during registration.

Since \( F(x, y) = a_0 + a_1 x + a_2 xy + a_3 y^2 + a_4 x^2 + a_5 xy^2 + a_6 x^2 y + a_7 x y^3 + \ldots + a_{16} x y^4 \mod p \) is an asymmetric polynomial of degree \( t-1 \) in \( x \) and \( h-1 \) degree in \( y \), which contains \( th \) different coefficients. In the proposed scheme, each token \( \{ s_i(y), s_i(x) \} \) contains two univariate polynomials with degree \( h-1 \) in \( y \) and \( t-1 \) degree in \( x \), respectively. In other words, each user can use his token to establish \( t+h \) linearly independent equations in terms of the coefficients of the asymmetric bivariate polynomial \( F(x, y) \). When there are \( t-1 \) colluded users with their tokens together, they can establish total \( (t+h)(t-1) \) equations. At the same time, for \( t-1 \) colluded users, there are \( 2C_{2t-1}^t \) pairwise keys. Hence, having \( t-1 \) colluded users’ shares, the total number of linearly independent equations available to the colluded users (i.e., \( th > (t+h)(t-1)-2C_{2t-1}^t \), they cannot recover the bivariate polynomial. Hence, they cannot learn any information of the secret. From \( th > (t+h)(t-1)-2C_{2t-1}^t \), we obtain \( h > 2t-2 \). Hence, if \( h > 2t-2 \), it assures that \( t-1 \) colluded inside adversaries cannot recover the secret polynomial \( F(x, y) \) selected by MRC initially. Thus, it needs at least \( t \) inside attackers to work together to reconstruct the tokens. The proposed protocol can resist up to \( t-1 \) colluded members to recover the secret polynomial \( F(x, y) \) of MRC. According to the security level requirement, the proper values of \( t \) and \( h \) can be selected. For example, when \( n = t-1 \), all member collisions cannot recover the secret polynomial \( F(x, y) \) of MRC.

**Theorem 2** (outside attack). In the proposed protocol, the outside attacker cannot obtain any secret information.

**Proof.** Outside attackers are illegitimate users who do not own any valid tokens from MRC. The outside attackers may try to impersonate members in the group key establishment to obtain the group key. However, since in the group key establishment, all exchange information of legitimate members are encrypted using pairwise shared keys and outside attackers do not own any valid token to recover any pairwise shared key, so the outside attacker cannot obtain any secret information.

5.2. Performance Evaluation. Most of the latest schemes can either provide user authentication or group key establishment separately [37–40]. They need additional membership authentication and shared keys distribution and also need interactive communications and complex computations for encryption and decryption. We first discuss performance features of our protocol as follows.

Compared with the existing schemes, our protocol can provide both membership authentication and group key establishment simultaneously. By using a bivariate polynomial, membership authentication and pairwise shared keys distribution are realized at the same time. Then, just by the XOR operation, each member mixes his/her input with pairwise shared keys with other members and releases the encrypted value in a broadcast channel. After collecting all released values, each member can compute the group key efficiently. In our protocol, tokens of members obtained during registration can be used for (a) membership authentication; (b) pairwise shared keys distribution; and (c) group key establishment. It is very efficient.

According to the definition in most communications, “Interactive communications” means acting one upon or with the other. In our group key establishment phase, each member computes his/her own values and releases the values to others without “waiting” for other members’ inputs. In other words, each member does not need waiting time in computing and releasing values to other members. We call this property “noninteractive,” which can speed up the communication process significantly.

Our proposal is noninteractive, computation-efficient, and lightweight, which has the advantages in storage, computation, and communication cost. Specific analysis is as follows.

5.2.1. Storage Cost. In our protocol, each member needs to store a token, \( \{ s_i(y), s_i(x) \} \), which consists of two univariate polynomials: one is \( t-1 \) degree in \( x \) and the other one is \( h-1 \) degree in \( y \). Thus, each shareholder needs to store \( t+h \) coefficients of a univariate polynomial. The storage requirement for each user is \( (t+h)\log_2p \) bits, where \( p \) is the modulus. This polynomial-based modulus is far less than public-key-based modulus.

5.2.2. Computation Cost. For our protocol, in Step 2 of membership authentication, when evaluating the polynomials, Horner’s rule [32] can be used to reduce the computational cost. Each shareholder needs to compute \( m-1 \) pairwise shared keys, \( k_{ij} = s_i(x_j) = F(x_i, x_j), j = 1, 2, \ldots, m, \) \( i \neq j \), by evaluating \( m-1 \) different polynomials. Using Horner’s rule, evaluating a polynomial of degree \( h-1 \) needs \( h-1 \) multiplications and \( h \) additions. In addition, each member needs to generate one authentication response and to verify \((m-1)\) authentication responses. Since each authentication response is a hash output, each member needs to compute \( m \) hash outputs. In steps of group key establishment, there are all XOR operations, symmetric encryption, and decryption operations which is very efficient.
in comparing with all existing protocols. Finally, there is only computing hash function to authenticate the group session key by each member. The computation load of our proposed protocol is much simpler than most public-key based schemes. For example, the RSA [41] public-key operation requires approximately $1.5 \log_2 N$ modulo multiplications (i.e., in RSA, $N$ is at least 1024 bits).

5.2.3. Communication Cost. The communication of membership authentication is performed completely in the broadcast channel. The total communication time is to transmit $m$ random integers, $[r_i, i = 1, 2, \ldots, m]$, and $m(m - 1)$ authentication responses for all participated group members. To establish the group key, the total communication time is to transmit $m$ random integers, $[l_i, i = 1, 2, \ldots, m]$, $m(m - 1)$-encrypted messages and $m$ hash outputs to authenticate the group session key for all participated group members. In our protocol, all transmitted data are computed on polynomial-based modulus. Furthermore, since our protocols are noninteractive, all released values can be broadcasted simultaneously and they are very efficient.

6. Conclusion

We have proposed a novel design of lightweight membership authentication and group key establishment protocol for WSNs. Our protocol provides both membership authentication and group key establishment simultaneously. However, all existing schemes can provide membership authentication and key establishment separately. We have included the security analysis and performance evaluation in the paper. Our protocol is very efficient in terms of computation and communication, so it is absolutely attractive for secure group communications in WSNs.

Data Availability

The data used to support the findings of this study are included within this article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors’ Contributions

Chingfang Hsu and Qi Cheng contributed equally to this work.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (Grants nos. 61772224 and 61872152), the Fundamental Research Funds for the Central Universities (No. CCNU19TS019), and the Research Planning Project of National Language Committee (No. YB135-40).

References

[1] A.-N. Shen, S. Guo, and H.-Y. Chien, “An efficient and scalable key distribution mechanism for hierarchical wireless sensor networks,” in Proceedings of the 2009 IEEE Sarnoff Symposium, pp. 1–5, Princeton, NJ, USA, March 2009.
[2] O. Cheikhrouhou, “Secure group communication in wireless sensor networks: a survey,” Journal of Network and Computer Applications, vol. 61, pp. 115–132, 2016.
[3] Q. Jiang, N. Zhang, J. Ni, J. Ma, X. Ma, and K.-K. Raymond Choo, “Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles,” IEEE Transactions on Vehicular Technology, 2020.
[4] Q. Jiang, Z. Chen, J. Ma, X. Ma, J. Shen, and D. Wu, “Optimized fuzzy commitment based key agreement protocol for wireless body area network,” IEEE Transactions on Emerging Topics in Computing, 2019.
[5] L. Eschenauer and V. D. Gligor, “A key management scheme for distributed sensor networks,” in Proceedings of the 9th ACM Conference on Computer and Communications Security—CCS ’02, pp. 41–47, Washington, DC, USA, November 2002.
[6] H. Chan, A. Perrig, and D. Song, “Random key predistribution schemes for sensor networks,” in Proceedings of the IEEE Symposium on Research in Security And Privacy, pp. 197–213, Berkeley, CA, USA, May 2003.
[7] 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 Proceedings of the INFOCOM 2004, vol. 1, p. 597, Hong Kong, China, March 2004.
[8] R. Blom, “Non-public key distribution,” in Advances in Cryptology: Proceedings of Crypto, D. Chaum, R. L. Rivest, and A. T. Sherman, Eds., Plenum, New York, NY, USA, pp. 231–236, 1982.
[9] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, “Perfectly-secure key distribution for dynamic conferences,” in Advances in Cryptology, E. F. Brickell, Ed., Springer-Verlag, Berlin, Germany, pp. 471–486, 1992.
[10] E. Khan, E. Gabidulin, B. Honary, and H. Ahmed, “Matrix-based memory efficient symmetric key generation and predistribution scheme for wireless sensor networks,” IET Wireless Sensor Systems, vol. 2, no. 2, pp. 108–114, 2012.
[11] A. Albakri and L. Harn, “Non-Interactive group key predistribution scheme (GKPS) for end-to-end routing in wireless sensor networks,” IEEE Access, vol. 7, pp. 31615–31623, 2019.
[12] A. Albakri, L. Harn, and S. Song, “Hierarchical key management scheme with probabilistic security in a wireless sensor network (WSN),” Security and Communication Networks, vol. 2019, Article ID 3950129, 11 pages, 2019.
[13] L. Harn, C.-F. Hsu, O. Ruan, and M.-Y. Zhang, “Novel design of secure end-to-end routing protocol in wireless sensor networks,” IEEE Sensors Journal, vol. 16, no. 6, pp. 1779–1785, 2016.
[14] L. Harn and C.-F. Hsu, “Predistribution scheme for establishing group keys in wireless sensor networks,” IEEE Sensors Journal, vol. 15, no. 9, pp. 5103–5108, 2015.
[15] N. Clifford, H. Sam, Y. Tom, and R. Kenneth, “The Kerberos network authentication service (V5),” 2016.
[16] IEEE 802 LAN/MAN Standards Committee, “IEEE 802.11, the working group setting the standards for wireless LANs,” 2019.
[17] W. Diffie and M. Hellman, “New directions in cryptography,” IEEE Transactions on Information Theory, vol. 22, no. 6, pp. 644–654, 1976.

[18] L. Harn and C. Lin, “Efficient group Diffie-Hellman key agreement protocols,” Computers & Electrical Engineering, vol. 40, no. 6, pp. 1972–1980, 2014.

[19] S. Lin, J. Kim, and G. Tsudik, “Flexible robust group key agreement,” IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 5, pp. 879–886, 2011.

[20] A. Joux, “A one round protocol for tripartite Diffie-Hellman,” in Algorithmic Number Theory: 4th International Symposium, ANTS-IV, Lecture Notes in Computer Science, vol. 1838, Springer-Verlag, Berlin, Germany, 2000.

[21] D. Abbasinezhad-Mood and M. Nikooghadam, “Efficient anonymous password-authenticated key exchange protocol to read isolated smart meters by utilization of extended Chebyshev chaotic maps,” IEEE Transactions on Industrial Informatics, vol. 14, no. 11, pp. 4815–4828, 2018.

[22] C. S. Laih, J. Y. Lee, and L. Harn, “A new threshold scheme and its application in designing the conference key distribution cryptosystem,” Information Processing Letters, vol. 32, no. 3, pp. 95–99, 1989.

[23] L. Harn and C. Lin, “Authenticated group key transfer protocol based on secret sharing,” IEEE Transactions on Computers, vol. 59, no. 6, pp. 842–846, 2010.

[24] R. Jiao, H. Ouyang, Y. Lin et al., “A computation-efficient group key distribution protocol based on a new secret sharing scheme,” Information, vol. 10, no. 5, p. 175, 2019.

[25] A. Shamir, “How to share a secret,” Communications of the ACM, vol. 22, no. 11, pp. 612–613, 1979.

[26] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch, “Verifiable secret sharing and achieving simultaneity in the presence of faults,” in Proceedings of the 26th IEEE Symposium on the Foundations of Computer Science, pp. 383–395, Portland, OR, USA, October 1985.

[27] R. Cramer, I. Damgard, S. Dziembowski, M. Hirt, and T. Rabin, “Efficient multiparty computations secure against an adaptive adversary,” LNCS, in Proceedings of 18th Annual IACR EUROCRYPT, Prague, Czech Republic, vol. 1592, Springer, Berlin, Germany, pp. 311–326, 1999.

[28] Y. Cheng and D. P. Agrawal, “A improved key distribution mechanism for large-scale hierarchical wireless sensor networks,” Journal of Ad Hoc Networks, vol. 5, no. 1, pp. 35–48, 2007.

[29] Y. Desmedt and Y. Frankel, “Shared generation of authenticators and signatures,” in Advances in Cryptology—Crypto ’89, pp. 457–569, Springer-Verlag, Berlin, Germany, 1991.

[30] J. Katz, C. Koo, and R. Kumaresan, “Improving the round complexity of VSS in point-to-point networks,” LNCS, in Proceedings of the ICALP ’08, Part II: Automata, Languages and Programming, vol. 5126, Springer, Berlin, Germany, pp. 499–510, 2008.

[31] R. Kumaresan, A. Patra, and C. P. Rangan, “The round complexity of verifiable secret sharing: the statistical case,” in Advances in Cryptology—ASIACRYPT 2010, LNCS, vol. 6477, pp. 431–447, Springer, Berlin, Germany, 2010.

[32] D. E. Knuth, The Art of Computer Programming, Semi-Numerical Algorithms, Vol. 2, Addison-Wesley, Boston, MA, USA, 1981.

[33] L. Harn and C.-F. Hsu, “A practical hybrid group key establishment for secure group communications,” The Computer Journal, vol. 60, no. 11, pp. 1582–1589, 2017.

[34] L. Harn and C.-F. Hsu, “A novel design of membership authentication and group key establishment protocol,” Security and Communication Networks, vol. 2017, Article ID 8547876, 7 pages, 2017.