Research Article

Self-Healing Group Key Distribution Facilitating Source Authentication Using Block Codes

Hung-Yu Chien

Department of Information Management, National Chi Nan University, Taiwan, China

Correspondence should be addressed to Hung-Yu Chien; hychien@ncnu.edu.tw

Received 28 December 2019; Revised 16 May 2021; Accepted 11 June 2021; Published 30 June 2021

Academic Editor: Petros Nicopolitidis

Copyright © 2021 Hung-Yu Chien. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Secure group key distribution is essential for many group-oriented applications such as sensor networks, multimedia broadcast services, and Internet of Things (IoT) scenarios. There are several challenges and requirements in designing secure group key distribution. Among them, computational efficiency, communication efficiency, adaptability to dynamic group membership change, robustness to various security threats, self-healing capacities, and source authentication are desirable. It is very challenging to design an efficient group distribution that satisfies all the requirements and challenges. Based on block codes, we propose an efficient self-healing group key distribution that facilitates both message source authentication and secure group key distribution, where the source identification and authentication can facilitate intrusion detection and identification. Both the privacy of the group key and the authentication of message sources are computationally secure. To the best of our knowledge, it is the first codes-based scheme that satisfies all the above requirements and facilitates message source authentication. The merits of the proposed scheme include the following: (1) it is highly efficient in terms of computation and communication, (2) it provides self-healing capacities for unstable environments, (3) it is very robust to various security threats and attacks, (4) it facilitates both message source authentication and secure group key distribution, and (5) it greatly improves the communication performance, compared to the state-of-the-art schemes. The security properties are analyzed, and the performance evaluations confirm its efficiency and practicality.

1. Introduction

Various group-oriented services such as audio/video broadcasting, conferencing, collective working, Internet-of-Things (IoT) scenarios, military operations, and rescue missions have been popular, and many new group-oriented services such as location-based social services are booming. To protect the confidentiality of the group-oriented services, it is common to encrypt the contents using the group keys, and only the authorized entities can access the contents using the keys. However, it is very challenging to design and deploy suitable group key distribution/agreement schemes to satisfy all the requirements of these services.

To facilitate group key sharing in group-oriented services, one approach is group key distribution and the other is group key agreement. In a group key agreement scheme, a set of entities co-operatively generate the group key. In a

group key distribution scheme, a trusted third party called Group Manager (GM) determines the group key and securely distributes the key to those legitimate entities. For those scenarios where either infrastructure or semi-infrastructure is available, group key distribution is preferred owing to the efficiency and easy deployment/maintenance. From the perspectives of how the membership structure is formed, group key distribution/agreement schemes can be divided into two categories. One category is for structured membership where the members of a group are predetermined or quite stably maintained. Examples are such as the broadcasting in a hierarchical tree [1] or some grouping of devices in an IoT scenario [2]. The other category is for those scenarios where the membership of a group dynamically changes. This paper focuses on group key distribution for the dynamic membership. Several metrics are commonly adopted to evaluate these group key distribution schemes.

Computational complexity and communication complexity are two important metrics as they affect the cost of the implementations, the energy consumption, the efficiency, the lifespan of batteries, etc.

For those services with dynamic membership, easy and efficient updating of the group keys for each session is necessary. In those scenarios with dynamic membership and unreliable connections, the function of self-healing allows the authorized entities to easily recover the previous group keys when they reconnect the networks later. Some wireless networks such as mobile ad-hoc networks and some IoT scenarios are likely to be unreliable and highly dynamic in the sense that nodes might move in and out of the range, switch to the sleep mode to extend the battery lifetime, or being disconnected from the networks due to attackers' disruption. In these scenarios, the system divides the lifetime of the group key distribution into successive time windows called epochs, where each epoch owns a different group key and those legitimate nodes are allowed to derive the group keys for the authorized epochs. A group key distribution scheme with the self-healing property allows a legitimate user (or node) to recover the lost group keys of the previous authorized epochs, even if some previous key distribution packets are lost. When it later reconnects the network, it can recover the lost group keys without requesting additional information from the GM.

The robustness against various security threats and attacks (such as collusion attacks, replay attacks, and impersonation attacks) should be ensured. All the above requirements have been studied and many group key distribution schemes have been designed accordingly.

However, the key function of source identification/authentication in a group-oriented service has been commonly neglected. In conventional group key distribution schemes, any legitimate entity can use the group key to encrypt the messages such that only authorized entities can decrypt and access the contents; however, the receivers cannot authenticate the source (the origin) of the messages since all legitimate entities access the same group key. The function of source identification and authentication is crucial for many group-oriented services where compromised entities need to be revoked. For example, many IoT devices are resource-limited and are vulnerable to being captured and being compromised; in such environments, it is very important that the intrusion detection system should continuously monitor the traffics to detect and identify the possible compromised devices. One another example is the group communications in military operations, where those captured entities need to be quickly detected and revoked [3]. With source authentication, it greatly facilitates the detection and identification of the compromised devices/entities. Similar scenarios and requirements exist in other group communications. The group communication with message source identification and authentication greatly improves the transmission efficiency, protects the privacy, and facilitates intrusion detection and identification.

Message Queue Telemetry Transport (MQTT) is one of the most popular IoT communication protocols. Chien et al. [4] proposed a secure MQTT group communication, but the

scheme does not provide source authentication. In Wireless Sensor Networks (WSNs), TESLA also needs the source authentication in the multicast scenarios [5, 6]. Kim et al. addressed the source authentication issue in Cyber-Physical Systems (CPS) such as smart grids [7]. Chan et al. tackled the challenge of source authentication in the group-based 3D streaming service [8]. Unfortunately, the conventional self-healing group key distribution study neglects this crucial function. This makes these conventional schemes functionally incomplete or security-weak for scenarios where authentication is crucial or desirable.

This paper, based on block codes [9], proposes a new self-healing group key distribution scheme with the source authentication function. The merits of the scheme include the following: (1) it exhibits excellent performance in terms of computation and communication, (2) it easily supports dynamic membership, (3) it provides self-healing capacities to cope with unreliable connections, (4) it facilitates both group key distribution as well as source message authentication, and (5) it greatly improves communication performance, compared to the state-of-the-art schemes. The rest of this paper is organized as follows. Section 2 discusses the related work. Section 3 introduces the preliminaries of Chien's block codes-based secret sharing. Section 4 presents the proposed scheme. Section 5 analyzes the security. Section 6 evaluates the performance. Finally, the conclusions are given in Section 7.

2. Related Work

Group key distribution schemes [1, 2] are designed for fixed membership scenarios and do not support the dynamic membership.

In 2002, Staddon et al. [10] first introduced the self-healing group key distribution with the dynamic membership for the unreliable connection scenarios; they defined the schemes, gave the lower bounds on the implementations, and proposed some constructions. Liu et al. [11] later generalized the definitions and gave some efficient constructions. Blundo et al. [12] slightly modified the definitions, designed some more efficient schemes, explored the lower bounds, and pointed out the weaknesses in some schemes in [10]. Blundo et al. [13] further modified the definitions and gave the new lower bounds on the implementations. Sáez [14] generalized the schemes in two aspects: (1) a general access structure instead of threshold-based ones; (2) a coalition of users can sponsor a user outside the group for one session. Daza et al. [15] noticed that the parameter settings of Dutta–Mulhoadhyay's works [16, 17] contradicted the lower bounds reported by [12, 13] and found the security flaws. The works in [10–14] mainly focused on the unconditional secure schemes while those in [10, 12] also gave some computational secure schemes. The schemes in [3, 18–26] mainly focused on designing computationally secure self-healing group key distributions with more flexible access structure or with improved efficiency.

Tian et al. [24], based on the mathematic structures used, classified the self-healing group key schemes [3, 14, 15, 18–28] into several categories: polynomial-based schemes,

exponential arithmetic-based schemes, vector space-based schemes, and pairing-based schemes. Among these approaches, pairing-based schemes demand higher computational cost. Rams and Pacyna did a survey of some earlier self-healing group key distribution schemes [28]. Some of them are restricted by the limitation of access structure expression; for example, these schemes only support the threshold-based access structures. Additionally, some are vulnerable to the collusion attacks. Vadlamudi and Vadlamudi [27] recently proposed a block codes-based scheme that facilitates the communication of any sets of users; the number and the set of revoked users are not constrained; a revoked user is allowed to rejoin the group in any of the later sessions, resist collusion attacks, and provide forward-and-backward secrecy. However, the scheme needs to transmit the new generator matrices in each session, which demands lots of communication overheads. Additionally, the scheme does not provide message source authentication.

Vijayakumar et al. [29] proposed an efficient key distribution scheme for the Internet pay-TV systems, using only simple computations; the Internet pay-TV key distribution schemes are similar to our scenarios where the service providers need to refresh the group keys for a set of dynamic groups of legitimate users, but there are some key differences between the two scenarios: (1) there is only one message source (the service provider) in the Internet pay-TV key distribution schemes; (2) due to the nature of the grouping of the services, the systems usually adopt some kinds of key hierarchy or tree structures to tackle the key distribution. Vijayakumar et al. [30] proposed a Chinese Remainder Theory-based distributed group key management scheme which supports the dynamic membership change for the unstructured peer-to-peer networks; the scheme reduces the computational complexity in each user side by slightly increasing the storage space of the peer users. The authors in [31] proposed a new Greatest Common Divisor-based key distribution protocol which aims at reducing the computational complexity and the amount of information stored in the Group Center and the group members; the self-healing function and the message source authentication function were not considered in the requirements. Wang et al. [32] proposed a mutual-healing group key distribution scheme in which the nodes can recover the lost group keys both securely and timely in the Unmanned Aerial Vehicles Ad-Hoc Network (UAANET); the mutual-healing function is efficiently achieved by adopting a private blockchain.

Based on block codes, Chien proposed a multisecret sharing scheme [33] and a dynamic-weight multisecret sharing scheme [34]. A unified approach, based on block codes, for both threshold-based secret sharing and general-access-structure secret sharing was proposed in [35]. Inspired by Chien's block-codes-based secret sharing and Vadlamudi-Vadlamudi's work [27], we will propose a new self-healing group key distribution scheme with the source authentication function. To the best of our knowledge, this is the first codes-based self-healing group distribution scheme that provides the flexible authorized group and also facilitates message source authentication.

3. Preliminaries

The notations used in this paper are summarized in Table 1.

Secure keyed one-way hash functions [27] are defined as follows: (1) given w and r , it is easy to compute $h(w, r)$; (2) given (possibly many) pair of r and $h(w, r)$, it is hard to compute w ; (3) without the knowledge of w , given (possibly many) pairs of r_i and $h(w, r_i)$, it is hard to compute $h(w, r')$, for $r \neq r_i$; (4) given w , it is hard to find two values r_1 and r_2 such that $h(w, r_1) = h(w, r_2)$, but $r_1 \neq r_2$. We also assume that $|h(w, r)| \leq m$, where $| \cdot |$ denotes the bit length so that $h(w, r)$ can be considered as an element in $GF(2^m)$.

Next, we briefly review the systematic block codes [38], and then, the technique based on the systematic block codes is described. A (n, k) (with $n > k$) linear block code over $GF(2^m)$ is defined by a $n \times k$ generator matrix $\mathbf{M}$ with symbols in $GF(2^m)$ and $k < 2^m$. In this paper, we denote the generator matrix as $\mathbf{G}(n, k)$, where n is the length and k is the dimension of the linear block codes. Denote $\mathbf{d} = (d_1, \dots, d_k)^t$ as a vector of k information symbols where d_i s are in $GF(2^m)$ and superscript t means vector transpose. Then, $\mathbf{V} = \mathbf{M} \mathbf{d} = (v_1, \dots, v_n)$ is the corresponding code word with v_i s in $GF(2^m)$.

A systematic block code is a type of linear block codes where the first k elements in a code word are identical to the information symbols $(d_1, \dots, d_k)$, and the last $n-k$ elements in the code word are denoted as $(c_1, c_2, \dots, c_{n-k})$ and called the parity symbols. In 1990, Ayanoglu et al. [38] designed a special type of systematic block code generator matrix $\mathbf{M}(n, k) = \begin{bmatrix} \mathbf{I}_{k \times k} \\ \mathbf{P}_{(n-k) \times k} \end{bmatrix}$, where $\mathbf{I}_{k \times k}$ is the $k \times k$ identity matrix and $\mathbf{P}_{(n-k) \times k}$ is a $(n-k) \times k$ matrix $[g^{(i-1)(j-1)}]$ with g being a primitive element in $GF(2^m)$ and $k < 2^m$. $\mathbf{I}_{k \times k}$ and $\mathbf{P}_{(n-k) \times k}$ can be represented as follows:

$$\mathbf{I}_{k \times k} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}, \quad \mathbf{P}_{(n-k) \times k} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & g & g^2 & \dots & g^{k-1} \\ 1 & g^2 & g^4 & \dots & g^{2(k-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & g^{n-k-1} & g^{(n-k-1)2} & \dots & g^{(n-k-1)(k-1)} \end{bmatrix}. \quad (1)$$

Since $\mathbf{V} = \mathbf{M}(n, k)\mathbf{d}$, then we have

$$c_i = \sum_{j=1}^k g^{(i-1)(j-1)} d_j, \quad (2)$$

where $1 \leq i \leq n-k$.

We require that $k < 2^m$ to satisfy the nonsingular requirement of matrix $\mathbf{P}$ [38]. The $(n-k)$ equations in (2) can, therefore, be viewed as the linear-independent equations of indeterminants $(d_1, d_2, \dots, d_k)$. If these $(n-k)$ equations

TABLE 1: The notations.

$U = \{U_1, U_2, \dots, U_N\}$	The set of users
$G_j; n_j = G_j $	$G_j \subset U$, the set of authorized users in session j ; $n_j = G_j $ denotes the number of users in G_j
$\mathbf{M}, \mathbf{M}_j(n_j, k_j)$, and $\mathbf{M}(n, k)$	Generator matrix for block codes, where n denotes the block length and k denotes the information length; $\mathbf{M}_j(n_j, k_j)$ denotes the generator matrix for session j , which is a submatrix of $\mathbf{M}(n, k)$
$k_i; gk_j; r_j$	k_i : secret key of user U_i ; gk_j : group key for session j ; r_j : a random nonce
$h_1(\cdot); h_2$	Secure keyed one-way hash functions $h_1(\cdot, \cdot): \{0, 1\}^{l_1} \times \{0, 1\}^{l_2} \rightarrow \{0, 1\}^{l_3}$, where l_1/l_2 , respectively, denotes the bit length of k_i /random nonce; one-way hash function $h_2: \{0, 1\}^{l_3} \rightarrow GF(2^m)$
$\text{seed}_{j,i}^0 = h_1(k_i, r_j)$	The seed value for user U_i 's hash chains at session j
$\text{seed}_{j,i}^s$	$\text{seed}_{j,i}^s = h_2^s(h_1(k_i, r_j)) = h_2^s(\text{seed}_{j,i}^0)$, where h_2^s denotes applying $h_2(\cdot)$ s times; s is the maximum number of encryptions each member can send in each session
$d = (d_1, d_2, \dots, d_k)^t$	Information symbols
$\mathbf{d}_j = (d_1, d_2, \dots, d_{n_j+1}) = (gk_j, \text{seed}_{j,1}^s, \dots, \text{seed}_{j,n_j}^s)$	Information symbols for session j ; it consists of the group key and the authentication tokens for the participants; s denotes the maximum number of messages in session j
$(c_1, c_2, \dots, c_{N-K}); \mathbf{c}_j$	Parity symbols
$T = \{t_1, t_2, \dots\}$	Successive time epochs
AD	Adversaries
AT_j	The set of authentication tokens for session j ; AT_j could be used to authenticate the message sources in session j
BC_j	$BC_j = \{G_j, r_j, c_1, \dots, c_{n_j}\}$; the broadcast data BC_j could be used to derive gk_j and AT_j
$\text{MAC}()$	Message authentication code function $\text{MAC}(): GF(2^m) \times \{0, 1\}^* \rightarrow \{0, 1\}^l$; it could be implemented, using HMAC [36]
$\text{ENC}/\text{Enc}_k(\text{msg})$	Encryption/encryption using the key k

were presented with $(n - k) < k$, then we would not be able to uniquely determine the values for these $(d_1, d_2, \dots, d_k)$. However, the remaining symbols can be recovered if some of these symbols d_i s are available such that the number of those missing symbols is smaller or equal to the number of equations. Based on this technique and hash functions, Chien et al. had proposed several secret-sharing schemes [33–35], and we shall propose our self-healing group key distribution scheme with source authentication.

4. Block Code-Based Group Key Distribution with Source Authentication

4.1. The System Model. There exist three kinds of entities. A Group Manager (GM), a set of users $U = \{U_1, U_2, \dots, U_N\}$, where $N \leq k - 1$, and adversaries AD s. The system consists of successive time epochs $T = \{t_1, t_2, \dots\}$ called sessions. The GM would like to securely distribute a group key gk_j and a set of authentication tokens to a set of legitimate users in each session $t_j \in T$. G_j denotes the set of the legitimate users for session j and satisfies $G_j \subset U$. Our definition of self-healing group key distribution with privacy is more flexible and practical in terms of the specification of the legitimate users and revoked users than most of the previous works, and it is the same as Vadlamudi–Vadlamudi's work [27]. We further extend the functions to include the source authentication. We list the features as follows.

- (1) There are no predetermined constraints on the number of legitimate users and on the number of revoked users.
- (2) The sets G_j s could change in each session without any restriction, that is, any user could be in the

legitimate user set or be out of the set at any session; legitimate users could be revoked at any session, and GM could let a revoked user rejoin the legitimate set at any later sessions.

- (3) The AD s in each session j could consist of any outsiders and any illegitimate users $U - G_j$ at that session; here, we define outsiders as those entities who have never been registered users of the system, but we define illegitimate users as those registered users who are illegitimate at some specific sessions. The goals of AD s are to access the contents of the group communications by deriving the group keys gk_j or to compromise the authentication in the group communications.
- (4) Only the legitimate users in the authorized set can access the communication contents and authenticate the source of the messages for that session.

Definition 1. D is a self-healing group key distribution with privacy and message source authentication if

- (1) G_j is the set of the authorized users for session j ; for any $U_i \in G_j$, the group key gk_j and the authentication tokens AT_j can be efficiently determined using BC_j and k_i .
- (2) gk_j and AT_j cannot be learned from the broadcasts $\{BC_1, \dots\}$ or the personal keys $\{k_1, \dots, k_N\}$ alone.
- (3) For any revoked user $U_i \in R = U - G_j$ and any adversary AD , it is computationally infeasible to compute gk_j and AT_j , using the broadcast BC_j and its secret keys.

- (4) For any coalition of the revoked users $U_i \in R = U - G_j$ and any adversary AD , it is computationally infeasible to compute gk_j and AT_j , using the broadcast BC_j and their secret keys.
- (5) Let C be a coalition of any users $U_i \in R = U - G_j$, and let $gk_C - AT_C$ be the set of all the group keys and authentication tokens they are authorized to access. Then, C cannot derive gk_j and AT_j corresponding to G_j , using the broadcasts, their secret keys, and $gk_C - AT_C$.
- (6) D is self-healing if a user $U_i \in G_j$ can derive gk_j and AT_j for any $j < j'$, using k_i and $BC_{j'}$.
- (7) D provides message source authentication, if the legitimate users $U_i \in G_j$ can use the tokens in AT_j to authenticate the message source of an encryption $Enc_{gk_j}(msg)$.

Please note that the above property 5 implies and extends the forward secrecy and the backward secrecy in the previous works in the sense that we allow any coalition of illegitimate users $U - G_j \cup A D$, and there are no constraints on the users joining the group or being revoked from the legitimate group.

4.2. The Proposed Scheme. The scheme consists of four phases: initialization phase, group key distribution phase, group key recovery phase, and group communication with authentication phase.

4.2.1. Initialization Phase. GM sets up a systematic block code with a generator matrix $\mathbf{M}(n, k)$ as specified in Section 2, a keyed one-way hash function $h_1(\cdot)$, and a one-way hash function h_2 . GM publishes the parameters $\mathbf{M}(n, k)$, $h_1(\cdot)$, and h_2 . For each user $U_i \in U$, GM selects a secret key k_i and securely distributes k_i to U_i .

4.2.2. Group Key Distribution Phase. Let $G_j \subset U$ be the authorized user set for a session $j \in T$, where $n_j = |G_j|$ denotes the number of users in G_j . To simplify the following presentation, let us suppose $G_j = \{U_1, U_2, \dots, U_{n_j}\}$ be the set of legitimate users for session j . GM selects a random group key gk_j and a nonce r_j and computes $seed_{j,i}^0 = h_1(k_i, r_j)$ for each user $U_i \in G_j$. It computes $seed_{j,i}^s = h_2^s(h_1(k_i, r_j))$ where s is the maximum number of messages transmitted by each user in one session and h_2^s denotes applying $h_2(\cdot)$ s times. gk_j is used as the group key, while $seed_{j,i}^s$ being the authentication token for $U_i \in G_j$.

GM prepares $\mathbf{d}_j = (d_1, d_2, \dots, d_{n_j+1}) = (gk_j, seed_{j,1}^s, \dots, seed_{j,n_j}^s)$ as the information to be broadcast and prepares the generator matrix $\mathbf{M}(2n_j + 1, n_j + 1)$ as specified in the preliminaries. GM generates $\mathbf{C}_j = (c_1, \dots, c_{n_j}) = \mathbf{P}_{n_j \times (n_j+1)} \mathbf{d}_j$, where the parities are specified as follows:

$$c_l = \sum_{u=1}^{n_j+1} g^{(t-1)(u-1)} d_u, \quad (3)$$

where $1 \leq l \leq n_j$.

Finally, GM broadcasts BC_j defined by equation (4). For self-healing, GM broadcasts the data defined by equation (5), where a window of p previous sessions is included; as our system does not limit the number of sessions it supports and can continuously run without a specified end, it is reasonable that the messages for long time ago are not interesting to the current entities; the selection of the window size p should depend on the scenarios and the application context. Please note that Vadlamudi-Vadlamudi's scheme [27] needs to distribute the new generator matrix and the new hash function in each session, but our scheme does not need to redistribute the generator matrix and the one-way hash because each generator matrix in one session is just a subpart of the generator matrix specified in equation (1). In each session, the GM just uses one subpart of the matrix in equation (1). The GM and the authorized users can easily determine the subpart, based on $n_j = |G_j|$:

$$BC_j = \{G_j, r_j, c_1, \dots, c_{n_j}\}, \quad (4)$$

$$\{BC_{j-p}, \dots, BC_j\} \text{ for a window of } (p+1) \text{ sessions.} \quad (5)$$

4.2.3. Group Key Recovery Phase. When a user $U_i \in G_j$ receives the broadcast BC_j , he prepares $seed_{j,i}^0 = h_1(k_i, r_j)$ and $seed_{j,i}^s$. With the recovered value of $seed_{j,i}^s$, now the set of n_j linear-independent equations in (2) has only n_j unknown variables, and they can be solved by U_i , that is, U_i can recover all the secrets $\mathbf{d}_j = (d_1, d_2, \dots, d_{n_j+1}) = (gk_j, seed_{j,1}^s, \dots, seed_{j,n_j}^s)$. This can be easily implemented, using the matrix multiplications.

4.2.4. Group Communication with Authentication Phase. Any legitimate user $U_i \in G_j$ can use the group key gk_j to encrypt its message msg and successively use $seed_{j,i}^{s-l}$, where $l = 1, \dots, s$ is a MAC (Message Authentication Code) key to generate the MAC.

The encryption ENC and the message authentication code MAC are specified as follows:

$$\begin{aligned} mac &= \text{MAC}(seed_{j,i}^{s-l}, msg), \\ \text{ENC} &= \text{Enc}_{gk_j}(U_i, msg, mac, seed_{j,i}^{s-l}), \end{aligned} \quad (6)$$

where U_i broadcasts $\{ENC\}$ in its l th encryption in session j .

Any legitimate user $U_i \in G_j$ can use the group key to decrypt the encryption and verify the source by checking whether both the verification equations in (7) hold. If the verifications succeed, then U_i accepts the message msg and is convinced of the source of the message. It stores the authentication token $seed_{j,i}^{s-l}$ for verifying the message in the next encryption:

$$\begin{aligned} seed_{j,i}^{s-l+1} &= h_2(seed_{j,i}^{s-l}), \\ mac &= \text{MAC}(seed_{j,i}^{s-l}, msg). \end{aligned} \quad (7)$$

5. Security Analysis and Verification

In this section, we prove that the proposed scheme satisfies the group key distribution privacy of Definition 1 in Theorem 1 and the message source authentication of Definition 1 in Theorem 2.

Theorem 1. *The proposed scheme satisfies the group key privacy of Definition 1, if the one-way hash and the keyed one-way hash function are secure.*

Proof. Since any single illegitimate user (including a revoked user) for session j is not stronger than the coalition of all illegitimate users in breaking the privacy, we directly prove the security against the coalition.

Let C be a coalition of any users $U_i \in R = U - G_j$, and let $gk_C - AT_C$ be the set of all the group keys and authentication tokens they are authorized to access.

Now we assume that the coalition can derive the group key gk_j , using their secret keys, the broadcasts, and $gk_C - AT_C$. Then, the set of n_j equations in (2) will have only n_j unknown variables $\{\text{seed}_{j,1}^s, \dots, \text{seed}_{j,n_j}^s\}$. As these equations are constructed as independent equations, the coalition can uniquely determine the values of these variables, that is, they can derive $\{\text{seed}_{j,1}^s, \dots, \text{seed}_{j,n_j}^s\}$, where $\text{seed}_{j,i}^s = h_2^s(h_1(k_i, r_j))$ and $U_i \in G_j$. It violates the security property of the one-way hash function and the keyed one-way hash function: (a) without the secret key k_i , one cannot compute $h_1(k_i, r_j)$ no matter how many pairs $\{h_1(k_i, r_{j'}), r_{j'}\}$ with $r_{j'} \neq r_j$ he gets; (b) without the secret $h_1(k_i, r_j)$, one cannot compute $h_2^s(h_1(k_i, r_j))$ for $s \geq 1$. So the coalition cannot derive the group key. $\square$

Theorem 2. *The proposed scheme provides the message source authentication if the MAC function and the one-way hash $h_2()$ are secure.*

Proof. For each encryption in session j , the sender needs to prepare $\text{mac} = \text{MAC}(\text{seed}_{j,i}^{s-l}, \text{msg})$ and $\text{ENC} = \text{Enc}_{gk_j}(U_i, \tilde{m}, \text{mac}, \text{seed}_{j,i}^{s-l})$ and broadcasts ENC. Any legitimate user $U_i \in G_j$ can use the group key to derive the message, mac and $\text{seed}_{j,i}^{s-l}$. Then, U_i verifies whether the two equations, $\text{seed}_{j,i}^{s-l+1} = h_2(\text{seed}_{j,i}^{s-l})$ and $\text{mac} = \text{MAC}(\text{seed}_{j,i}^{s-l}, \text{msg})$, hold. As long as the MAC function and the one-way hash $h_2()$ are secure, it ensures the authentication of the message source. $\square$

After the theorems, the security properties of our scheme can be analyzed from the following different perspectives.

- (i) Only the authorized user can derive the group key and the authentication tokens.

Given the public values in equations (2)–(4), we can see that (n_j+1) , the number of unknown symbols, is larger than the number of independent equations in (2). So, an adversary has no way to derive the secrets. On the contrary, any legitimate user $U_i \in G_j$ can calculate one value of the information $\text{seed}_{j,i} = h_1(k_i, r_j)$ and $\text{seed}_{j,i}^s$. With this value

$\text{seed}_{j,i}^s$, now the number of independent equations equals the number of unknown values, and U_i can solve the equations and derive all the values $\mathbf{d}_j = (d_1, d_2, \dots, d_{n_j+1}) = (gk_j, \text{seed}_{j,1}^s, \dots, \text{seed}_{j,n_j}^s)$.

- (ii) Only the authorized user can decrypt the encryptions and validate the source of the messages.

Since a legitimate user $U_i \in G_j$ can acquire the group key and tokens, $\mathbf{d}_j = (d_1, d_2, \dots, d_{n_j+1}) = (gk_j, \text{seed}_{j,1}^s, \dots, \text{seed}_{j,n_j}^s)$, from the broadcast, he can decrypt the encryptions using the group key gk_j . From the content $\{U_i, \text{msg}, \text{mac}, \text{seed}_{j,i}^{s-l}\}$, U_i derives the message msg, the authentication token $\text{seed}_{j,i}^{s-l}$, and mac. Using the token and mac, U_i can validate, using the verification equations in (7), the source of the message, as h_2 is a one-way hash, and HMAC is applied on the new seed and the message content to have the MAC.

- (iii) The self-healing property.

From the broadcast in equation (5), any legitimate user can derive the group keys and the authentication tokens corresponding to those sessions he is authorized, even if he loses the connections in those sessions. In our scheme, a window of p previous sessions is included; the selection of the window size p should depend on the scenarios and the application context.

- (iv) Anywise forward/backward secrecy.

To derive the group key and authentication codes in a session, one should prepare at least one seed $\in \{\text{seed}_{j,1}^s, \dots, \text{seed}_{j,n_j}^s\}$. Assume a user $U_i \in G_j$ is authorized in session j but not authorized in another session j' ($j \neq j'$); of course, U_i can prepare the $\text{seed}_{j,i} = h_1(k_i, r_j)$ and $\text{seed}_{j,i}^s$ in the authorized session j to solve equation (2). But, for session j' where U_i is not belonging to $G_{j'}$, U_i has no way to get any value $\in (\text{seed}_{j',1}^s, \dots, \text{seed}_{j',n_{j'}}^s)$ and cannot solve equation (2) for session j' . This limits U_i 's capacities of recovering the group keys and tokens in only those authorized sessions.

- (v) Anywise collusion resistance.

In our scheme, the only way to derive the group keys and tokens for session j is to prepare at least one seed $\in \{\text{seed}_{j,1}^s, \dots, \text{seed}_{j,n_j}^s\}$. Fortunately, any collusion without the help of at least one authorized user in session j cannot derive any seed and the group key from the broadcast. This ensures the collusion resistance.

6. Performance Evaluations

We first summarize the functions of several related schemes in Table 2. From the summarized table, we find that, even though all these schemes tackle the group key distribution challenges, the supported functions and the application

TABLE 2: Comparison of the functionalities of the related schemes.

Scheme	No. of revoked users	Authorized group	Revoked users rejoining a later session	Source authentication	Self-healing	Other key features
[12, 13]	Predetermined before deployment	Not flexible	Not allowed	Not provided	Yes	US
[14]	Predetermined before deployment	Not flexible	Not allowed	Not provided	Yes	US
[3, 19]	Predetermined before deployment	Not flexible	Not allowed	Not provided	Yes	CS
[39]	Predetermined before deployment	Not flexible	Yes but with different identity	Not provided	Yes	CS
[27]	Not predetermined	Flexible	Allowed	Not provided	Yes	CS; GKUPMC
[29]	Not predetermined	Flexible	Allowed	Not required ¹	Not consider	HKS; GKUMCSC.
[30]	Not predetermined but the number of group members is predetermined	Flexible	Allowed	Not provided	Not consider	PDGKM; GKUMC; the size of the encryption expands linearly with the number of the members
[32]	Not predetermined	Flexible	Allowed	Not provided	Mutual healing ²	GKUMC
[31]	Not predetermined	Flexible	Allowed	Not provided	Not consider	GKUMC
[26]	Not predetermined	Flexible	Allowed	Not provided	Yes	GKUMC
[21]	Predetermined	Not flexible	Not allowed	Not provided	Yes	GKUMC
[22]	Predetermined	Not flexible	Not allowed	Not provided	Yes	GKUPMC
Ours	Not predetermined	Flexible	Allowed	Provided	Yes	GKUPMC

In a pay-TV system, there is only one message source. A node can recover the lost keys from the ground station and the neighbors. CS: computationally secure; US: unconditionally secure. GKUMC: Group Key Updating triggered by Membership Change; GKUPMC: Group Key Updating Periodically and triggered by Membership Change; GKUMCSC: Group Key Updating triggered by Membership Change and Subscription Change; HKS: Hierarchical Key Structure; PDGKM: Partially Distributed Group Key Management.

scenarios of them are quite different; the differences in the application scenarios affect the design, the security properties, and the models very much. For example, the scheme [29] is designed for the pay-TV system where the group key updating is triggered by both its membership change and its subscription change, while the key updating of most of other schemes being triggered periodically or being triggered by membership change; the scenario also motivates its design of the hierarchical key structure; it does not need to consider the source authentication since there is only one message source in the pay-TV systems. The scheme [30] is designed for the P2P networks, where the group key management is partially distributed; even though the scheme might seem computationally efficient, we notice that the number of group member is predetermined, the number affects the initial public parameters, and the size of one single encryption expands linearly as the number increases. It is interesting that there are a group of schemes [32] that aim at providing the mutual-healing function which facilitates a node recover the lost keys from the group manager or from its neighbors. Therefore, to have a fair comparison among group key distribution schemes, we have to examine the application scenarios, the supported functions, and the performance.

In this paper, we focus on those systems which support flexible authorized group expression and flexible member-revoking condition, and a revoked-member could rejoin the system at a later session because such kinds of systems have wider application scenarios and have greater practical

application potentials. The schemes [26, 27, 29, 31, 32] and our scheme have these flexible functions. Among the schemes, only our scheme provides the source authentication, which is very important in many applications.

Now we evaluate the computational complexity and the communication complexity. A comparison table among several related schemes is given in Table 3.

6.1. The Communication Complexity. The communication overheads are listed in the 2nd column and the 3rd column in Table 3, where the 2nd column lists the overheads for one group key updating session and the 3rd column lists the costs for a w -session recovery. For some schemes which do not provide the key updating for one single session, we merge the 2nd column and the 3rd column into one column in the table. The 6th column lists the secret storage space in a node. To simplify the comparison, we let Len denote the bit length for one key or one identity for all the schemes, despite the variations of the bit lengths in the corresponding schemes and scenarios. We use the symbol “ $\sim$ ” to denote the asymptotic complexity which could further simplify the overhead estimation to capture the asymptotic complexity.

From the table, we can see that the schemes [26, 31] require the less communication overheads for one single session; however, we should notice that it does not have significant value to examine each individual performance metric (such as Column 2, Column 3, and Column 6) because the schemes aim at quite different application

TABLE 3: The comparisons of the computation/communication overhead of related schemes.

Scheme	Communication overhead of one single session	Communication overhead of self-healing of w -sessions	GM's computations in group key distribution	User's computations in group key recovery	Storage of private data in a node
[22]	$(\max((t+2), v, \bar{n} + 3v) + j + 1) \text{Len} \sim (nm + m + 1) \text{Len}$		For j th session: $(j+2j)T_h + (t+j+2)T_{\text{mul}} + IT_{\text{enc}} + (j+1)T_{\text{exp}} \sim 3mT_h + (n+m+2)T_{\text{mul}} + mT_{\text{enc}} + 2mT_{\text{exp}}$	$7T_{\text{mul}} + 4T_{\text{exp}} + 1T_{\text{inv}} + 3T_h + T_{\text{dec}}$	2 Len
[21]	$[(t+1)v + 1] \text{Len} + (v+j) \text{Len} \sim ((n+1)m + 2m + 1) \text{Len}$		$jT_h + 2jT_{\text{dec}} + (2nj + j + 2)T_{\text{mul}}$	$+(j+1)T_{\text{exp}} + T_{\text{inv}} + T_{\text{dec}} + 6T_{\text{mul}} + (j+3)T_{\text{exp}} + T_{\text{inv}}$	Len
[26]	$3 \log n \text{Len}$		$\log n(T_h + T_{\text{dec}})$	$2^n(1 + (1/2^{n-(n/2)} \log n - 1)T_h)$	$\log n \text{Len}$
[31]	Does not provide self-healing		$3 + 1T_{GC/D}$	$1 + T_{\text{inv}} + T_{\text{div}}$	3 Len
[30]	$O(n')$ broadcast $\sim O(n') \text{Len}$	$O(n')$ broadcast for one session of a n' -member batch join $\sim O(n') \text{Len}$	Not applicable	$O(n') T_{\text{add}}$ for a n' -member leave; $O(n') (T_{\text{mul}} + T_{\text{add}})$ for a n' -member join	2 Len
[27]	$(3\bar{n} + 1) \text{Len}$ and one generator $\text{matrix} \sim O(\bar{n}^2) \text{Len}$	$\sum_{i \in \text{leaving}} [(3\bar{n} + 1) \text{Len}] \sim O(w\bar{n}^2) \text{Len}$	$T_{\text{matrix_gen}} + T_{\text{matrix_mul}} + \bar{n}T_h \sim O(\bar{n}^{3.3737}) T_{\text{mul}}$	$T_{\text{matrix_gen}} + T_{\text{matrix_mul}} + 1T_h \sim O(\bar{n}^{3.3737}) T_{\text{mul}}$	1 Len
Ours	$(2\bar{n} + 1) \text{Len}$	$\sum_{i \in \text{leaving}} (2 G_j + 1) \text{Len} \sim O(w\bar{n}) \text{Len}$	$T_{\text{matrix_gen}} + \bar{n}T_h \sim O(\bar{n}^{2.3737}) T_{\text{mul}}$	$T_{\text{matrix_gen}} + (s+1)T_h \sim O(\bar{n}^{2.3737}) T_{\text{mul}}$	1 Len

(1) n' denotes the number of leaving members in a batch leave; n denotes the number of members in one session; m denotes the maximum number of sessions; Len denotes the bit length for one scalar value such as one key and one identity; w denotes the number of sessions to recover the keys. (2) t denotes the maximum number of revoked users; v denotes the number of sessions in which there are users joining the group ($v < j$), where j is the index of the current session; $|G_j|$ denotes the number of members in the j th session. (3) j denotes the j th session (could be replaced with $O(m)$); J denotes the number of sessions with joining users (could be replaced with $O(m)$), where m is the maximum number of sessions; Q denotes the number of neighbors around the request node. (4) T_h denotes the computational cost for one hashing; T_{mul} denotes the cost for one modular multiplication; T_{add} denotes the cost for one modular addition/subtraction; $T_{\text{matrix_gen}}$ denotes the computational cost for one matrix multiplication, which is $O(n^{2.3737})T_{\text{mul}}$; $T_{\text{matrix_mul}}$ denotes the cost for one modular addition/subtraction; T_{inv} denotes the cost for one modular inversion operation, which is $O(n^{3.3737})$; T_{exp} denotes the cost for one exponentiation operation; T_{div} denotes the cost for one modulo division operation; T_{sig} denotes the cost for one signature generation; T_{ver} denotes the cost for one signature verification; T_e/T_d denotes the cost for one asymmetric encryption/decryption; T_{enc} denotes the cost for one symmetric encryption/decryption; T_{dec} denotes the cost for one symmetric decryption; $T_{\text{GC/D}}$ denotes the cost for one Greatest Common Divisor operation.

scenarios and provide quite different functions. The schemes [27] and our scheme require moderate communication overheads, as their overheads are linear to the size of the number of the legitimate members in that session.

Among the schemes, we have to emphasize the comparison between our scheme and Vadlamudi–Vadlamudi’s scheme [27] because the two schemes have the same application scenarios and they all adopt the coding-based mechanisms. In this regard, we note that Vadlamudi–Vadlamudi’s scheme [27] needs to distribute the generator matrices for each session; but our scheme does not need to renew the generator matrices because each generator matrix in one session is just a subpart of the generator matrix specified in equation (1); the GM and the authorized users can easily determine the subparts, based on the number n_j .

The communication overhead of the GM’s broadcast of our scheme for session j consists of G_j , one random nonce, and n_j parities, where n_j is the number of authorized users in session j . It sums up to $2n_j + 1$ numbers in equation (4). For the self-healing property in equation (5), it totally sums up to $\sum_{t \in w\text{-session}} (2n_t + 1)$ numbers, which is of order $O(w\bar{n})$, where w denotes the number of sessions to recover and $\bar{n}$ denotes the number of members in a session. The communication overhead of one encryption in the group communication with the authentication phase consists of one encryption of four numbers in equation (6). Vadlamudi–Vadlamudi’s scheme [27] for session j only is $(2n_j^2 + 5n_j + 2)$ numbers; for the self-healing broadcasting of Vadlamudi–Vadlamudi’s scheme, it is $\sum_{t=1 \sim j} (2n_t^2 + 5n_t + 2)$ numbers, which is of order $O(w\bar{n}^2)$.

6.1.1. The Computational Complexity of Group Key Distribution/Recovery. The Columns 4 and 5, respectively, list the computational overheads of the GM and one member for recovering the group keys. In the table, the schemes [26, 30, 31] require the less computational overheads for recovering the keys; however, we note two points: (1) there exist many implicit communication overheads for securely transmitting membership authorization/authentication messages in [30]; (2) it does not have significant value to only examine these performance metrics individually (like Column 4 and Column 5), as the scenarios and the supported functions of these schemes vary a lot. Among the related schemes, our scheme and Vadlamudi–Vadlamudi’s scheme [27] have the same scenarios and adopt the same coding mechanisms. We, therefore, emphasize on the two schemes in the following comparison.

We first evaluate that of our scheme. Both the operations in performing the encryptions/decryptions of group keys and tokens in equation (2) and in the group key recovery phase involve the computations of $\text{seed}_{j,i}^s$ and the matrix multiplications. To prepare n_j $\text{seed}_{j,i}^s$, it requires $n_j(s + 1)$ hashes: $O(n_j s)$ hashes, where n_j is the number of authorized users in that session and s is the specified maximum number of encryptions in one session. To calculate the parities in equation (2), the complexity of matrix multiplication is

$O(\bar{n}^{2.3737})$ [38]. To derive the group key and tokens in one session, U_i performs $(s + 1)$ hashes and one matrix multiplication. For each encryption/decryption and the validation of the source of a message in the group communication with the authentication phase, one performs one encryption/decryption and one hash operation.

Vadlamudi–Vadlamudi’s work [27] specifies two possible implementations for the matrix preparation and the transmission, where one implementation reduces the cost of the GM, at the additional cost of the receivers. Since both the two implementations demand higher cost than our scheme, here we only consider the case of which the GM prepares the whole matrix by itself; and, we focus on estimating the asymptotic notation of the complexities without being caught in lengthy and tedious computations. This highlights the comparison of the asymptotic complexities. For Vadlamudi–Vadlamudi’s scheme, the GM additionally needs the matrix preparation in each session, which roughly equals n times the cost of the corresponding matrix multiplication. So, it totally needs $O(\bar{n}^{3.3737})$ operations. In a short summary, our scheme greatly improves both the communication overheads and the computational complexity, compared to its counterpart [27].

7. Conclusions

In this paper, we have proposed a new self-healing group key distribution which facilitates message source authentication, based on the systematic block codes. The contributions of this submission are listed as follows. (1) To the best of our knowledge, this is the first codes-based self-healing group key distribution scheme that provides message source authentication. (2) The determination of the authorized-user set is flexible; that is, any user is allowed to join the authorized set at any sessions and is allowed to leave the set at any sessions. (3) The communication overhead is greatly reduced, by eliminating the transmission of the generator matrix. (4) The computation is highly efficient, as it only involves one matrix multiplication and several hashing operations. (5) The scheme provides forward/backward secrecy. These merits make the proposed scheme much attractive and practical than its counterparts in many application scenarios.

Data Availability

No data were used to support this study.

Conflicts of Interest

The author declares that there are no conflicts of interest.

Acknowledgments

This project was partially supported by the Ministry of Science and Technology, Taiwan, R.O.C., under Grant no. MOST 108-2221-E-260-009-MY3.

References

- [1] W. G. Wen-Guey Tzeng, "A time-bound cryptographic key assignment scheme for access control in a hierarchy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 1, pp. 182–188, 2002.
- [2] H.-Y. Chien, "Group-oriented range-bound key agreement for internet of Things scenarios," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1890–1903, 2018.
- [3] R. Dutta, S. Mukhopadhyay, and M. Collier, "Computationally secure self-healing key distribution with revocation in wireless ad hoc networks," *Ad Hoc Networks*, vol. 8, no. 6, pp. 597–613, 2010.
- [4] H. Y. Chien, X. A. Kou, M. L. Chiang, and C. H. Su, "Secure and efficient MQTT group communication design," in *Proceedings of the 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2019)*, Toyama, Japan, July, 2019.
- [5] X. Xin Zhao and A. Prakash, "Source authentication in group communication systems," in *Proceedings of the 14th International Workshop on Database and Expert Systems Applications*, pp. 455–459, Prague, Czech Republic, September 2003.
- [6] Y. Yang Li, M. Miao Zhang, Y. H. Yanhui Guo, and G. Guoai Xu, "Optimized source authentication scheme for multicast based on merkle tree and TESLA," in *Proceedings of the IEEE International Conference on Information Theory and Information Security*, pp. 195–198, Beijing, China, December 2010.
- [7] Y. Kim, V. Kolesnikov, and M. Thottan, "Resilient end-to-end message protection for cyber-physical system communications," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 2478–2487, 2018.
- [8] M.-C. Chan, J.-R. Jiang, C.-W. Hung, and W. T. Ooi, "Group-based peer-to-peer 3D streaming authentication," in *Proceedings of the 2009 15th International Conference on Parallel and Distributed Systems*, pp. 685–691, Shenzhen, China, December 2009.
- [9] S. Lin and D. J. Costello Jr., *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1983.
- [10] J. Staddon, S. Miner, M. Franklin, D. Balfanz, M. Malkin, and D. Dean, "Self-Healing Key Distribution with Revocation," in *Proceedings of the IEEE Symposium on Security and Privacy*, Berkeley, CA, USA, May 2002.
- [11] D. Liu, P. Ning, and K. Sun, "Efficient self-healing key distribution with revocation capability," in *Proceedings of the 10-th ACM Conference on Computer and Communications Security*, Washington, DC, October 2003.
- [12] C. Blundo, P. D'Arco, A. De Santis, and M. Listo, "Design of self-healing key distribution schemes," *Designs, Codes and Cryptography*, vol. 32, no. 1-3, pp. 15–44, 2004.
- [13] C. Blundo, P. D'Arco, and A. De Santis, "Definitions and bounds for self-healing key distribution schemes," in *Automata, Languages and Programming. ICALP 2004*, J. Díaz, J. Karhumäki, A. Lepistö, and D. Sannella, Eds., Vol. 3142, Springer, Berlin, Germany, 2004.
- [14] G. Sáez, "Self-healing key distribution schemes with sponsorship," in *Communications and Multimedia Security. CMS 2005*, J. Dittmann, S. Katzenbeisser, and A. Uhl, Eds., Vol. 3677, Springer, Berlin, Germany, 2005.
- [15] V. Daza, J. Herranz, and G. Sáez, "Flaws in some self-healing key distribution schemes with revocation," *Information Processing Letters*, vol. 109, no. 11, pp. 523–526, 2009, <https://www.iacr.org/cryptodb/data/paper.php?pubkey=18028>.
- [16] R. Dutta and S. Mukhopadhyay, "Improved self-healing key distribution with revocation in wireless sensor network," in *Proceedings of the Wireless Communications and Networking Conference, (WCNC)*, pp. 2963–2968, Hong Kong, China, March 2007.
- [17] R. Dutta and S. Mukhopadhyay, "Designing scalable self-healing key distribution schemes with revocation capability," *Parallel and Distributed Processing and Applications*, Springer, Berlin, Germany, pp. 419–430, 2007.
- [18] T. J. Tsitaitse, Y. Cai, and A. Ditta, "Secure self-healing group key distribution scheme with constant storage for SCADA systems in smart grid," *Wireless Personal Communications*, vol. 101, no. 3, pp. 1749–1763, 2018.
- [19] R. Dutta, S. Mukhopadhyay, A. Das, and S. Emmanuel, "Generalized self-healing key distribution using vector space access structure," in *Networking 2008 Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet. Networking 2008. LNCS4982*, A. Das, H. K. Pung, F. B. S. Lee, and L. W. C. Wong, Eds., Springer, Berlin, Germany, 2008.
- [20] H. Guo, Y. Zheng, B. Wang, and Z. Li, "A note on an improved self-healing group key distribution scheme," *Sensors*, vol. 15, no. 10, Article ID 25033, 2015.
- [21] H. Guo, Y. Zheng, X. Zhang, and Z. Li, "Exponential arithmetic based self-healing group key distribution scheme with backward secrecy under the resource-constrained wireless networks," *Sensors*, vol. 16, no. 5, p. 609, 2016.
- [22] H. Guo, Y. Zheng, X. Li, Z. Li, and C. Xia, "Self-healing group key distribution protocol in wireless sensor networks for secure IoT communications," *Future Generation Computer Systems*, vol. 89, pp. 713–721, 2018.
- [23] J. Shen, S. Chang, Q. Liu, J. Shen, and Y. Ren, "Implicit authentication protocol and self-healing key management for WBANs," *Multimedia Tools and Applications*, vol. 77, no. 9, Article ID 11381, 2018.
- [24] B. Tian, S. Han, T. S. Dillon, and S. Das, "A self-healing key distribution scheme based on vector space secret sharing and one-way hash chains," in *Proceedings of the 9th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks Workshop on Security, Privacy and Authentication in Wireless Networks*, pp. 1–6, IEEE, Newport Beach, CA, USA, June 2008.
- [25] S. Agrawal and M. L. Das, "Mutual healing enabled group-key distribution protocol in wireless sensor networks," *Computer Communications*, vol. 112, pp. 131–140, 2017.
- [26] J. Hur and Y. Lee, "A reliable group key management scheme for broadcast encryption," *Journal of Communications and Networks*, vol. 18, no. 2, pp. 246–260, 2016.
- [27] C. V. Vadlamudi and S. P. D. Vadlamudi, "A novel self-healing key distribution scheme based on vector space access structure and MDS codes," *International Journal of Communication Systems*, vol. 32, no. 16, 2019.
- [28] T. Rams and P. Pacyna, "A survey of group key distribution schemes with self-healing property," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 820–842, 2013.
- [29] P. Vijayakumar, R. Naresh, S. H. Islam, and L. J. Deborah, "An effective key distribution for secure internet pay-TV using access key hierarchies," *Security and Communication Networks*, vol. 9, no. 18, pp. 5085–5097, 2016.
- [30] P. Vijayakumar, R. Naresh, L. Jegatha Deborah, and S. Hafizul Islam, "An efficient group key agreement protocol for secure P2P communication," *Security and Communication Networks*, vol. 9, no. 17, pp. 3952–3965, 2016.

- [31] P. Vijayakumar, S. Bose, and A. Kannan, "Centralized key distribution protocol using the greatest common divisor method," *Computers & Mathematics with Applications*, vol. 65, no. 9, pp. 1360–1368, 2013.
- [32] X. Li, Y. Wang, P. Vijayakumar, D. He, N. Kumar, and J. Ma, "Blockchain-based mutual-healing group key distribution scheme in unmanned aerial Vehicles ad-hoc network," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, Article ID 11309, 2019.
- [33] H. Y. Chien, J. K. Jan, and Y. M. Tseng, "A practical (t,n)-Multi-Secret sharing scheme," *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences, Japan*, vol. E83-A, no. 12, pp. 2762–2765, 2000.
- [34] H. Y. Chien, J. K. Jan, and Y. M. Tseng, "A new generalized group-oriented cryptosystem based on the systematic block codes," in *Proceedings of the 2000 International Computer Symposium Workshop on Cryptology and Information Security*, National Chung Cheng University, Chiayi, Taiwan, China, 2000.
- [35] H. Y. Chien, J. K. Jan, and Y. M. Tseng, "A unified approach to secret sharing schemes with low distribution cost," *Journal of the Chinese Institute of Engineers*, vol. 25, no. 6, pp. 723–733, 2002.
- [36] M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication," in *Proceedings of the Advances in Cryptology — CRYPTO '96, Lecture Notes in Computer Science*, pp. 1–15, Santa Barbara, CA, USA, August 1996.
- [37] L. Gong, "New protocols for third-party-based authentication and secure broadcast," in *Proceedings of the 2nd ACM Conference on Computer and Communications Security*, vol. 176–183, Fairfax, VA, USA, November 1994.
- [38] E. Ayanoglu, C. L. I, R. D. Citlin, and J. E. Mazo, "Diversity coding: using error control for self-healing in communication networks," in *Proceedings of Ninth Annual Joint Conference of the IEEE Computer and Communication Societies*, pp. 95–104, San Francisco, CA, USA, June 1990.
- [39] R. Dutta and S. Sanyal, "Collusion resistant self-healing key distribution in mobile wireless networks," *International Journal of Wireless and Mobile Computing*, vol. 5, no. 3, pp. 228–243, 2012.