Hindawi
Security and Communication Networks
Volume 2024, Article ID 9874958, 1 page
https://doi.org/10.1155/2024/9874958

Retraction

Retracted: A Novel Hierarchical Key Assignment Scheme for Data Access Control in IoT

Security and Communication Networks

Received 8 January 2024; Accepted 8 January 2024; Published 9 January 2024

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope

(2) Discrepancies in the description of the research reported

(3) Discrepancies between the availability of data and the research described

(4) Inappropriate citations

(5) Incoherent, meaningless and/or irrelevant content included in the article

(6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] X. Li, M. Ye, J. Chen, J. Chen, and Y.-C. Chen, "A Novel Hierarchical Key Assignment Scheme for Data Access Control in IoT," *Security and Communication Networks*, vol. 2021, Article ID 6174506, 12 pages, 2021.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 6174506, 12 pages
https://doi.org/10.1155/2021/6174506

Research Article

A Novel Hierarchical Key Assignment Scheme for Data Access Control in IoT

Xiaoyu Li,[1] **Min Ye**,[2] **Jiahui Chen**,[3] **Jianhui Chen**,[1] **and Yeh-Cheng Chen**[4]

[1]*School of Intelligence Engineering, Zhengzhou University of Aeronautics, Zhengzhou 450 046, China*
[2]*GuangDong Overseas Chinese Vocational School, and Guangdong Communication Polytechnic, Guangzhou 510 520, China*
[3]*School of Computers, Guangdong University of Technology, Guangzhou 510 006, China*
[4]*Department of Computer Science, University of California, Davis, CA 95616, USA*

Correspondence should be addressed to Jiahui Chen; csjhchen@gmail.com

Received 19 August 2021; Revised 10 November 2021; Accepted 23 November 2021; Published 6 December 2021

Academic Editor: Xingsi Xue

Hierarchical key assignment scheme is an efficient cryptographic method for hierarchical access control, in which the encryption keys of lower classes can be derived by the higher classes. Such a property is an effective way to ensure the access control security of Internet of Things data markets. However, many researchers on this field cannot avoid potential single point of failure in key distribution, and some key assignment schemes are insecure against collusive attack or sibling attack or collaborative attack. In this paper, we propose a hierarchical key assignment scheme based on multilinear map to solve the multigroup access control in Internet of Things data markets. Compared with previous hierarchical key assignment schemes, our scheme can avoid potential single point of failure in key distribution. Also the central authority of our scheme (corresponding to the data owner in IoT data markets) does not need to assign the corresponding encryption keys to each user directly, and users in each class can obtain the encryption key via only a one-round key agreement protocol. We then show that our scheme satisfies the security of key indistinguishability under decisional multilinear Diffie-Hellman assumption. Finally, comparisons show the efficiency of our scheme and indicates that our proposed scheme can not only resist the potential attacks, but also guarantee the forward and backward security.

1. Introduction

Internet of Things (IoT) is the internetworking of smart sensing devices with network connectivity which enable these devices to collect and exchange data. To a certain extent, IoT can be viewed as a physical and logical extension of the current Internet. In the coming years, it is expected that the IoT can bridge many diverse technologies to enable new application services by connecting sensing devices together in support of intelligent decision making [1].

Since sensor data has the huge potential value, many IoT commercial corporations, called IoT data owners, provide pay-on-demand access services on original IoT data. That is, IoT data are made available to users as they pay for what they need. Thus, data confidentiality is at the top of the list of concerns for IoT data owners. Although encryption can provide data confidentiality, classic encryption methods cannot meet the requirement of flexible and fine-grained access control for IoT data markets. This is because the users' access rights in real applications are often organized in a hierarchy. Take the vehicle-to-everything (V2X) network as an example; it is based on lots of sensing devices that create and transmit data from these surroundings through various links, such as vehicle-to-person (V2P), vehicle-to-infrastructure (V2I), vehicle-to-vehicle (V2V), vehicle-to-building (V2B), and so on. As shown in Figure 1, the access rights of these three subscribed users have a hierarchical structure on V2X data. The automaker has the supreme seniority and can access all V2X data, while the logistics company only accesses V2V data and V2I data. The self-driving service company can access more data than the logistics company, but less data than the automaker.

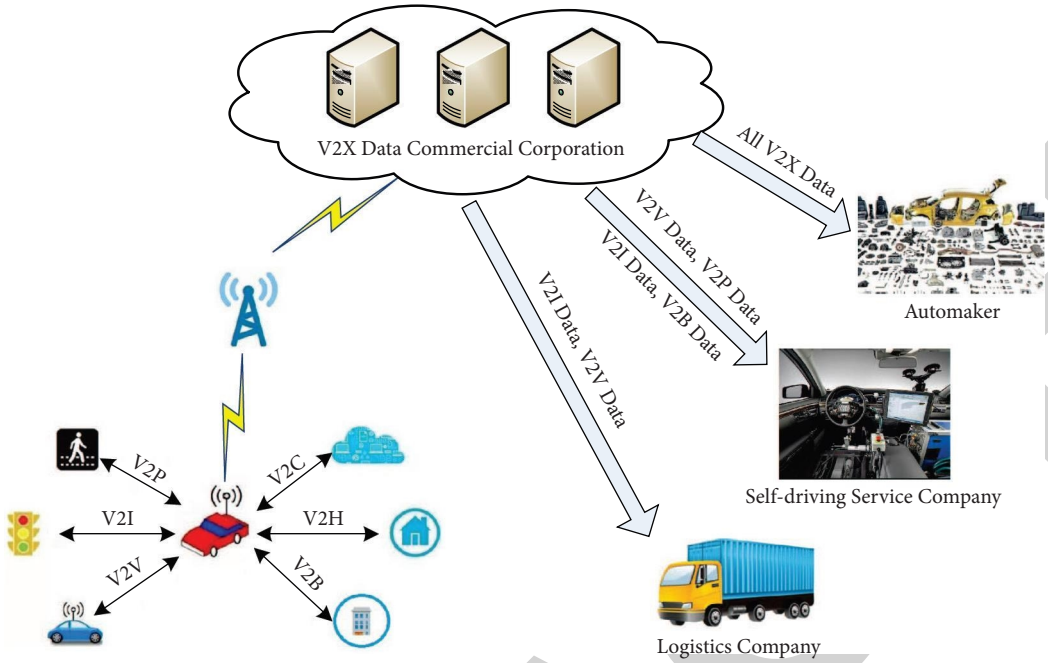

Figure 1: Hierarchical data access in V2X scenario.

From the perspective of function realization, access control is an alternative form of data sharing. And there is an extensive research carried out in proposing the ciphertext-policy attribute-based encryption (CP-ABE) [2] in the fields of secure and flexible IoT data sharing [3, 4]. However, existing CP-ABE schemes have a high overhead since the implementation of access structure is complicated. Moreover, attribute revocation is also an intractable problem in CP-ABE and requires extra computation and communication costs to deal with. With this in mind, many researchers study the issue of data sharing in IoT with a different primitive: group key management [5, 6]. However, traditional group key management shows poor flexibility and scalability for multigroup access control.

Our contributions: in this paper, we propose a novel hierarchical key assignment scheme (HKAS) for secure and flexible access control in IoT data markets. Some significant features of our proposed scheme are as follows:

(i) The proposed scheme can avoid potential single point of failure of IoT data owner in key distribution. In our scheme, IoT data owner only focuses on the maintenance of the hierarchical structure, and users obtain the encryption keys via a one-round key agreement protocol.

(ii) Different from many dependent key schemes, the encryption key and private information of each class in our proposed HKAS are independent. This protects the encryption key being derived from the private information and improves the security of IoT data service system.

(iii) Our proposed scheme supplies efficient dynamical updates. When the hierarchical structure or user dynamically changes, IoT data owner updates the public information by using only one broadcast message.

(iv) We prove that our scheme can reach the security of key indistinguishability under decisional multilinear Diffie-Hellman assumption. Furthermore, our proposed scheme can avoid potential attacks such as collusive attack, sibling attack, and collaborative attack.

1.1. Related Works. In the IoT systems, a large number of sensor data is generated and transmitted. Without any doubt, data is an extremely important asset for all organizations. Thus, secure access control (or data sharing) which refers to the access rights of sensor data, is a paramount concern in IoT [7]. As we recalled above, many studies make use of CP-ABE to achieve the fine-grained access control in various IoT applications [8, 9]. However, CP-ABE is a cumbersome cryptographic mechanism, which is not suitable for resource-constrained IoT networks. In [10], Seo et al. proposed a certificateless-effective key management protocol for secure data access control in dynamic wireless sensor networks. All of the above solutions aim at establishing the secure system deployment for IoT. In terms of business operation, the IoT applications and services have the requirement of data sharing on sensor data. A key management scheme for publish-subscribe system that is compliant with the data access control requirements of smart grid and IoT protocols is proposed in [11]. As we know, the key management schemes based on the preshared key framework and key pool framework are not scalable for large numbers of entities and dynamic changes in relationships [12].

Firstly proposed by Akl and Taylor [13], HKAS is an efficient cryptographic method for solving the hierarchical

multigroup access control problem by allowing authorized users to have different access privileges. Since then, Hassen et al. [14] classified HKAS into two major approaches: dependent key approach and independent key approach. In the dependent key approach, users are organized in a hierarchy and allocated with a certain amount of security classes, where a security class can represent an individual user or a group of users. In the hierarchical structure, a central authority (CA) is used to assign an encryption key and some private information to each security class. More precisely, the encryption key is used to protect the data by constructing a symmetric cryptosystem, while the private information is devoted to deriving the encryption keys of all classes in the lower-down hierarchy. On the other side, there are also two ways in deriving the encryption keys: direct one and indirect one. The direct key derivation does not need to compute all the intermediate keys on the path from the higher class to the lower class, while the indirect key derivation needs to do so. Contrary to the dependent key approach, HKAS based on independent key approach considers the hierarchical relations between user groups and resource groups and each user needs to maintain the encryption keys of all resources which he/she can access. However, the composition of the user group in an independent key scheme is a little different from that in the dependent key scheme. More precisely, the number of user groups in the independent key scheme is usually larger than that in the dependent key scheme for solving the same hierarchical access control problem.

Thereafter, many researchers proposed numerous dependent key schemes [15–19]. One of the main approaches to construct a dependent key scheme is to use a prime number's fundamental properties [13, 20, 21], which brings in some additional drawbacks, such as large public information, vulnerable to GCD operation and collusive attack [22, 23]. Thanks to the lower computation overhead and smaller key size, elliptic curve cryptosystem (ECC) was devoted to constructing the dependent key schemes in [24, 25]. However, these two schemes cannot resist the collaborative attack [26] and sibling attack [27], respectively. In addition to the security issues, another drawback of HKASs based on ECC is the huge amounts of public information [28], which leads to collusive attack [22]. A more general scenario has been considered for HKAS, where the access control is not only hierarchical, but also shared with other classes [29].

The abovementioned dependent HKASs solve only the hierarchical access control problems for classes rather than users. This means that the CA needs to assign the corresponding encryption key (or private information) to each user in a point to point manner. Thus if there are many users in the classes, the efficiency will decrease dramatically. Moreover, the rekeying mechanism also needs to be guaranteed due to the confidentiality requirement in dynamic key management.

On the other side, the independent key approach which uses key trees and graphs techniques is user-oriented [30], such as the integrated multigroup key management scheme for the contributory environment proposed in [31]. Although independent key scheme is quite simple to deploy, it does not offer efficient support for the hierarchy changes, especially in dynamic access situation. The reason is that such a scheme needs to update lots of keys. Based on multilinear map, Zhou et al. proposed a decentralized multigroup key management scheme for hierarchical access control in [32]. In their independent key scheme, the rekeying mechanism is to negotiate among the involved user groups. Specifically, each involved user group's server reselects a new public parameter and carries out one-round group key agreement protocol based on multilinear map. However, the number of involved user groups will be large in the case of massive user groups. And the parameter size of multilinear map is linear in the number of involved user groups. Due to the implementation of multilinear map, Zhou et al.'s scheme is inefficient when the number of user groups is very large.

1.2. Organization. The rest of this paper is arranged as follows. Definitions and background information are given in Section 2. In Section 3, we propose our HKAS based on multilinear map and discuss the dynamic key management. The security and performance analysis of our proposed scheme are presented in Section 4 and Section 5, respectively. Finally, we conclude this paper in Section 6.

2. Preliminaries

This section gives some background knowledge that will be used in this paper. Firstly, we give a brief description of hierarchical key assignment. Then, we present the security model of HKAS. Finally, we introduce the definition of multilinear map and two intractable problems on multilinear map.

2.1. Hierarchical Key Assignment. The hierarchical structure of a system is represented by a partially ordered set as poset. It is defined as a set of classes $V = \{V_1, V_2, \ldots, V_n\}$ with respect to a binary relation "$\leq$". The notation $V_i \leq V_j$ means that the users in class V_j can access the data of users in class V_i; i.e., the access right of class V_j is higher than or equal to that of class V_i. If $V_i \leq V_j$ and there is no class $V_k \in V$ such that $V_i \leq V_k \leq V_j$, we say that class V_j is an immediate predecessor of class V_i, which is denoted by $V_i \leq V_j$. Here, class V_i is also considered as the immediate successor of class V_j.

Formally, the above mentioned poset $(V, \leq)$ can be represented as a directed graph $G = (V, E)$, and we say that the vertices V in G coincide with the classes and an edge $(V_i, V_j) \in E$ if and only if $V_j \leq V_i$. Without loss of generation, we set G as a directed acyclic graph. In $G = (V, E)$, we define two associated sets for each class: $Anc(V_i, G)$ and $Des(V_i, G)$. If there is a path from class V_i to class V_j in G, we denote $V_i \in Anc(V_j, G)$ and $V_j \in Des(V_i, G)$. The immediate predecessors and immediate successors of class V_i in G are denoted by $Pre\,d(V_i, G)$ and $Succ(V_i, G)$, respectively.

Let Γ be a set of access graphs corresponding to partially ordered hierarchies. An HKAS [13, 33] for Γ is defined as follows.

Definition 1. An HKAS for Γ is defined as a pair of algorithms (Gen, Der) which satisfy the following conditions:

(i) $Gen(1^\kappa, G)$ is a probabilistic polynomial-time algorithm that takes a security parameter 1^κ and a graph $G = (V, E) \in \Gamma$ as input. And it outputs:

(a) A piece of private information sk_i and an encryption key k_i for class $V_i \in V$;
(b) A piece of public information pub.

(ii) $Der(V_i, V_j, sk_i, pub)$ is a deterministic polynomial-time algorithm that inputs the public information pub, two classes $V_i, V_j \in V$, and V_i's private information sk_i. If $V_i \in Anc(V_j, G)$, it outputs the encryption key k_j which will be assigned to class V_j. Otherwise, it outputs a rejection symbol $\perp$.

We use (SK, K, pub) to denote the output of $Gen(1^\kappa, G)$, where SK and K are considered as the sets of private information and encryption keys of classes, respectively.

2.2. Security Model of HKAS. The security model of HKAS was formally provided in [33]. Atallah et al. proposed two different notions of security: security against key recovery (KR-security) and security for key indistinguishability (KI-security). The KR-security means that an adversary is not able to compute an encryption key which cannot be derived from the corrupted users, whereas the KI-security requires that an adversary is not able to distinguish the encryption key from a random string of the same length. Thus the KI-security implies the KR-security. Recently, Freire et al. [34] proposed the notion of security for strong key indistinguishability (S-KI-security) and argued that their new notion is strictly stronger than KI-security. Such a problem has been recently addressed in [35], which shows that S-KI-security is not stronger than KI-security, and claimed the equivalence between these two security notions. A similar result has been also shown in the unconditionally secure setting by [36].

Thus based on the above conclusion, in this paper, we mainly concentrate on the KI-security and we only consider the security model for a static adversary. Formally, let there be an access graph $G = (V, E)$; we define a static adversary $\mathcal{A}_{stat}$ that firstly chooses a class $V_* \in V$ and an algorithm $Corrupt$ which can provide public information pub and some private information sk_j to the adversary by using Gen algorithm on the access graph G. Let $corr$ denote the output of $Corrupt$. On receiving a private information sk_j, the adversary can compute an encryption key k_j of class V_j. Then let there be another encryption key k_* not derived from all the private information sk_j and encryption keys k_j. We finally define a challenge phase that gives either the encryption key k_* or a random string of the same length; the adversary's goal is to distinguish these two cases. The definition of KI-security is given as follows.

Definition 2. Let there be a set of access graphs Γ corresponding to partially ordered hierarchies, and lett (Gen, Der) be an HKAS for Γ. We consider the following two experiments:

(i) Experiment $Exp^{KI-1}_{\mathcal{A},V_*}(1^\kappa, G)$
$(SK, K, pub) \leftarrow Gen(1^\kappa, G)$
$corr \leftarrow Corrupt_{V_*}(SK)$
$d \leftarrow \mathcal{A}(1^\kappa, G, pub, corr, k_*)$
return d

(ii) Experiment $Exp^{KI-0}_{\mathcal{A},V_*}(1^\kappa, G)$
$(SK, K, pub) \leftarrow Gen(1^\kappa, G)$
$corr \leftarrow Corrupt_{V_*}(SK)$
$\rho \leftarrow \{0,1\}^{length(k_*)}$
$d \leftarrow \mathcal{A}(1^\kappa, G, pub, corr, \rho)$
return d

For any $G = (V, E) \in \Gamma$ and $V_* \in V$, the advantage of $\mathcal{A}_{stat}$ is defined as $Adv^{KI}_{\mathcal{A}}(1^\kappa, G) = |\Pr[Exp^{KI-1}_{\mathcal{A},G}(1^\kappa, G) = 1] - \Pr[Exp^{KI-0}_{\mathcal{A},G}(1^\kappa, G) = 1]|$. An HKAS is said to be secure in the sense of key indistinguishability with respect to each static adversary, if $Adv^{KI}_{\mathcal{A}}(1^\kappa, G)$ is negligible for each graph $G \in \Gamma$ and each class $V_* \in V$.

Then, some underlying attacks, such as the contrary attack, sibling attack, and collaborative attack [16, 27, 37], are investigated in the security assessment. Besides, based on the requirement of practical application, HKAS should also consider the forward and backward security as is stated in [38]. The forward security means that a user cannot access the future data of the class V_i when revoking this user from class V_i, while the backward security implies that a user cannot access the previous data of the class V_j when adding this user into the class V_j. We will consider all these security features in the next part of our paper.

2.3. Multilinear Map and Complexity Assumptions. The multilinear map is a novel primitive and has many cryptographic applications, such as the multipartite key exchange protocol [39–42] and revocation system [43, 44].

Remark: in this paper, we mainly focus on how to construct an HKAS using the property of multilinear maps. Attacks against an instance of multilinear map can translate to attacks against our proposed scheme, if our scheme is based on this instance. Although various instances of multilinear maps are proved to be insecure, the work on multilinear maps is being continued and new candidates of multilinear maps are proposed. Due to it, some candidates of multilinear maps are proposed in [45, 46]. Thus our proposed scheme can be immediately instantiated with these candidates of multilinear maps.

Let p be a prime number, and let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two multiplicative cyclic groups of order p. A map $e_n: \mathbb{G}_1^n \longrightarrow \mathbb{G}_2$ is said to be an $n-$ multilinear map [39] if it satisfies the following properties:

(1) If $a_1, \ldots, a_n \in \mathbb{Z}_p$ and $g_1, \ldots, g_n \in \mathbb{G}_1$, then $e_n(g_1^{a_1}, \ldots, g_n^{a_n}) = e_n(g_1, \ldots, g_n)^{a_1 \cdots a_n}$.

(2) The map e_n is called nondegenerate once it satisfies the following condition: if g is a generator of $\mathbb{G}_1$, then $e(g, \ldots, g)$ is a generator of $\mathbb{G}_2$.

Similar to the bilinear case, the computational multilinear Diffie-Hellman (CMDH) and decisional multilinear Diffie-Hellman (DMDH) problem are described as follows.

Definition 3. Let g be a generator of $\mathbb{G}_1$ and e_n: $\mathbb{G}_1^n \longrightarrow \mathbb{G}_2$ be an $n-$ multilinear map. Given $g^{a_1}, \ldots, g^{a_{n+1}} \in \mathbb{G}_1$, where $a_1, \ldots, a_{n+1} \in \mathbb{Z}_p$, the CMDH problem is to compute $e_n(g, \ldots, g)^{a_1 \cdots a_{n+1}}$ in $\mathbb{G}_2$, and the DMDH problem is to distinguish T between $e_n(g, \ldots, g)^{a_1 \cdots a_{n+1}}$ and a random $\mathbb{G}_2-$ element.

CMDH assumption: this assumption says that it is hard to solve the CMDH problem. More precisely, the advantage for any probability polynomial-time algorithm $\mathcal{A}$ to solve the CMDH problem is negligible.

DMDH assumption: it supposes that any probability polynomial-time algorithm $\mathcal{A}$ has a negligible advantage in solving the DMDH problem.

3. Our Proposed Scheme

We now propose our HKAS based on multilinear map. Then, we give the processes of rekeying in dynamic environments, including inserting a new class, removing an existing class, adding user, and revoking user.

3.1. System Model. The important features of our proposed scheme are the centralized control policy for hierarchy and the distributed key agreement policy for the encryption key in each class. Figure 2 shows the system overview of our scheme. It is important to point out that each IoT data owner needs to play the CA's role of HKAS in IoT data markets. The arrowhead with a solid line in Figure 2 represents the hierarchy between two classes. For example, there is an arrowhead from class V_i to class V_l; it means $V_l \leq V_i$. It should be noted that the hierarchical structure of classes is considered to be public.

In this system, the CA computes the encryption keys of classes in a top-down manner. That is, the encryption keys of those being the root node in G are firstly computed. Then, the encryption keys of their immediate successors are derived by the CA. This process repeats until the encryption keys of all the classes are computed. Finally, the CA broadcasts the public information of each class. Once receiving this public information, users in each class can obtain the corresponding encryption key via a one-round key agreement protocol. For the private information of each class, it needs to be sent to each user in a unicast channel to accomplish the key derivation. This can be done at the time of registration. Using the encryption key and private information of a class, any of the users in that class can derive the encryption keys in the lower classes. For dynamic key management, it can be solved without the point to point communication between the CA and each involved user.

3.2. Key Generation and Derivation. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two multiplicative cyclic groups of the same prime order p, and let g be a generator of $\mathbb{G}_1$. H_1: $\{0,1\}^* \longrightarrow \mathbb{Z}_p$ and H_2: $\{0,1\}^* \longrightarrow \mathbb{G}_1$ are two one-way hash functions. e_n: $\mathbb{G}_1^n \longrightarrow \mathbb{G}_2$ is an $n-$ multilinear map. The notation U_{ij} denotes user U_j in the class V_i, and let the identity of U_j be ID_{ij}.

Key generation: for $G = (V, E)$, the CA chooses $s, x \in \mathbb{Z}_p^*$ as the master keys and computes g^s as the public information of this system. If class V_i is a root node in G, the CA sets $d_i = 1$ and the private information of class V_i as $s_i = H_1(x \| \{1 \cdots 1\}^k)$, where $k \in \mathbb{N}$ is a public parameter. Otherwise, there exists a maximum path from a root to class V_i in G, and the CA sets d_i as the number of classes in this path. The private information of class V_i is $s_i = \underbrace{H_1(H_1(\ldots H_1(x \| \{1 \ldots 1\}^k) \ldots \| \{1 \ldots 1\}^k) \| \{1 \ldots 1\}^k)}_{di}$.

Take Figure 2 as an example; the private information of class V_i and class V_j is all $H_1(x \| \{1 \cdots 1\}^k)$. Then, the private information of class V_l is $H_1(H_1(H_1(x \| \{1 \cdots 1\}^k) \| \{1 \cdots 1\}^k) \| \{1 \cdots 1\}^k)$. To be specific, such a setting has two reasons. On the one hand, a class V_i may be located in different paths in $G = (V, E)$. An intractable problem is how to ensure the consistency of computation on the encryption keys of lower classes for higher classes on different path. If d_i is set as the above setting, this problem will have gone with the wind. On the other hand, such a setting is conducive to reflecting the hierarchy of classes.

If a user U_j wants to join into the class V_i, U_j should register with the CA and obtain the private information s_i. The public key and private key of U_{ij} are $pk_{ij} = H_2(ID_{ij})^{k_{ij}}$ and $sk_{ij} = H_2(ID_{ij})^{k_{ij}s}$, respectively. Of course, k_{ij} can also be chosen by user U_i.

For class V_i in the hierarchy, the CA uses $d_i \| g^{r_i} \| \{pk_{i1}, \ldots, pk_{it}\}$ as the public information of class V_i, if $U_{i1}, \cdots U_{it}$ are the group users in class V_i. The g^{r_i} can be computed by s_i and the encryption key of class V_i with hash function H_1 as follows.

Let $V_{i_1}, \ldots, V_{i_{n_i}}$ be the immediate predecessors of class V_i in G, i.e., $Pre(V_i, G) = \{V_{i_1}, \ldots, V_{i_{n_i}}\}$. Once we obtain the public information of these classes, user U_{ij} can compute the encryption key of class V_i by

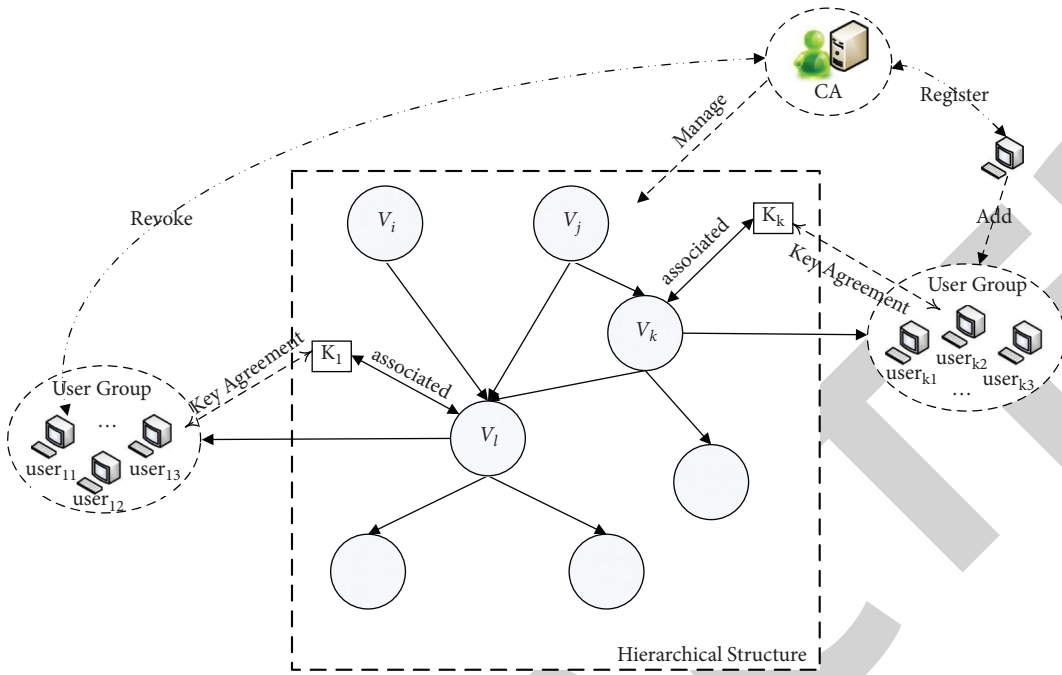

Figure 2: System model of our scheme.

$$k_i = e_n\left(pk_{i1}, \ldots, pk_{ij-1}, sk_{ij}, pk_{ij+1}, \ldots, pk_{it}, g^{r_{i_1}}, \ldots, g^{r_{i_{n_i}}}, g^s, \ldots, g^s\right),$$
$$= e_n\left(H_2(ID_{i1}), \ldots, H_2(ID_{it}), g, \ldots, g\right)^{k_{i1}\cdots k_{it} r_{i_1}\cdots r_{i_{n_i}} s^{n-n_i-t+1}}. \quad (1)$$

Of course, the CA also can compute the encryption key of class V_i by using the master key:

$$e_n\left(pk_{i1}, \ldots, pk_{it}, g^{r_{i_1}}, \ldots, g^{r_{i_{n_i}}}, g^s, \ldots, g^s\right)^s$$
$$= e_n\left(pk_{i1}, \ldots, pk_{it}, g, \ldots, g\right)^{k_{i1}\cdots k_{it} r_{i_1}\cdots r_{i_{n_i}} s^{n-n_i-t+1}} = k_i. \quad (2)$$

Finally, the public information of class V_i is computed by $g^{r_i} = g^{(s_i \oplus H_1(k_i))}$.

The initial encryption key of class V_i is $e(g, \ldots, g)^{r_{i_1}\cdots r_{i_{n_i}} s^{n-n_i+1}}$, if $Pre\,d(V_i, G) = \left\{V_{i_1}, \ldots, V_{i_{n_i}}\right\}$ and there are no users in this class. This can be seen as a preset way for hierarchy. It can make the data access control more granular and scalable.

Key derivation: assume that $V_i \in Anc(V_j, G)$. The path from V_i to V_j in G is $V_i = V_{j_t}, V_{j_{t-1}}, \ldots, V_{j_1}, V_{j_0} = V_j$. Each user in class V_i can derive the encryption key of class V_j as shown in Figure 3.

Form the way of key derivation, users in a class derive the encryption key of the lower classes with the need of iterative computation. To avoid this drawback, we can modify the formula of the encryption key as

$$k_i = e_n\Bigg(pk_{i1}, \ldots, sk_{ij}, \cdots pk_{it}, \underbrace{g^{r_{i_1}}, \ldots, g^{r_{i_{n_i}}}}_{\text{the public information of } V_i^c\text{s ancestor classes}}, g^s, \ldots, g^s\Bigg). \quad (3)$$

For a class in the hierarchy, the number of its higher class will be much larger than that of its immediate predecessors. This requires that the parameter n should be chosen larger for the direct key derivation. As it is widely known, the multilinear map will be more and more costly with the growth of parameter n. Therefore, we only discuss our proposed indirect key scheme in this paper. Furthermore, the number of users in a class is also an important factor for the size of the parameter n. We can integrate users into a virtual user with the help of group key agreement protocol. The private key of this virtual user is $g^{sH_1(k')}$, where k' is the negotiated group key of these users.

3.3. Dynamic Key Management. Data access control should consider the dynamic management at the level of individual users, while hierarchical access control also needs to consider the dynamic management at the level of user groups.

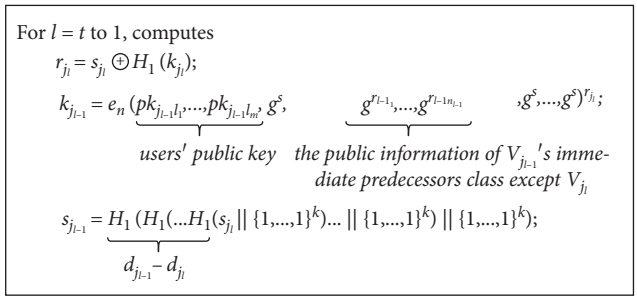

Figure 3: The process of key derivation.

Therefore, we consider the following four situations in our HKAS, which are corresponding to four scenarios in the dynamic hierarchical access control in IoT: user groups' addition and revocation and individual user's subscription and unsubscription.

Inserting a new class: let class $V_i, V_j \in V$ satisfy the relation $V_j \le V_i$ in the hierarchy. Now, consider that a new class V_t needs to be inserted into the hierarchy such that $V_j \le V_t \le V_i$. If there are no users in class V_t, the CA needs the following steps to manage a new hierarchical structure.

(1) Compute class V_t's private information $s_t = H_1(s_i \| \{1 \cdots 1\}^k)$. Then, compute the encryption key of class V_t: $k_t = e_n(g^{r_i}, g^s, \cdots . g^s)^s$. The public information of class V_t is V_t: $(d_t = d_i + 1) \| g^{r_t} = g^{s_t \oplus H_2(k_t)}$.

(2) For each class $V_l \in Des(V_t, G)$, compute the new d_l and g^{r_l} as described in key generation.

(3) Update the corresponding public information of these classes and broadcast a message with the form of "add $(V_t: d_t \| g^{r_t})$ into the hierarchy $\{(V_l, d_l, g^{r_l}) | V_l \in Des(V_t, G)\}$".

After receiving this message, users who are in the affected classes compute the new encryption key of the corresponding class as described in the key generation. Meanwhile, the private information of some affected classes must be updated by the new $\{d_l | V_l \in Des(V_t, G)\}$.

If the new class V_t has some users in the initial status, the process is similar to the above.

Removing an existing class: assume that an existing class V_i is to be removed from the hierarchy. The CA performs the following steps to maintain the new hierarchical structure:

(1) Remove the public information of class V_i.

(2) For each class $V_j \in Des(V_i, G)$, compute the new g^{r_j} for updating the public information of this class.

(3) Update the public information of the affected classes and broadcast a rekeying message with the form of $\{$Remove class V_i from the hierarchy $\{(V_j, g^{r_j}) | V_j \in Des(V_i, G)\}\}$.

After receiving this message, users in the affected classes compute the new encryption key as the key generation.

What calls for special attention is that the private information of the affected classes will not be updated. This is because the private information of class is computed by a one-way hash function H_1 and obtaining its preimage is intractable for the involved users.

The creation of a new relation into the hierarchy or the revocation of an existing relation from the hierarchy can be easily solved by invoking the above two processes.

Adding user: when a new user requests to join a class in the hierarchy, this user should register with the CA, thus obtaining his/her private key and the private information of the joined class by a secure channel. Let ID_{it+1} denote the identity of this user. This implies that user U_{t+1} wants to join into the class V_i. After the registration, the CA firstly appends the public key $pk_{it+1} = H_2(ID_{it+1})^{k_{it+1}}$ into the public information of class V_i. Then, the CA computes the new g^{r_m} for class V_m, where $V_m \in (Des(V_i, G) \cup \{V_i\})$. When obtaining these results, the CA updates the new g^{r_m} of these affected classes. Finally, the CA broadcasts a rekeying message with form of "Adding a new user into $\{V_i \| pk_{it+1} \| \{(V_i, g^{r_i})\} \cup \{(V_m, g^{r_m}) | V_m \in Des(V_i, G)\}\}$". Once receiving this message, user U_{ij} updates the encryption key of class V_i by

$$\begin{aligned} k_i' &= e_n\left(pk_{i1}, \ldots, sk_{ij}, \ldots, pk_{it}, H_2(ID_{it+1})^{k_{it+1}}, g^{r_{i_1}}, \ldots, g^{r_{i_{n_i}}}, g^s, \cdots . g^s\right) \\ &= e_n\left(H_2(ID_{i1}), \ldots, H_2(ID_{it}), H_2(ID_{it+1}), g, \cdots . g\right)^{k_{i1} \cdots k_{it} k_{it+1} r_{i_1} \cdots r_{i_{n_i}} s^{n-n_i-t}}. \end{aligned} \tag{4}$$

Users in class V_m, where $V_m \in Des(V_i, G)$, can derive the new encryption key by the updated public information. The computational method is the same as the key generation.

Revoking user: when the CA wants to revoke a user with identity ID_{ij}, the CA firstly deletes this user's public information from the public information of class V_i. Secondly, the CA computes and updates the new g^{r_m} for class V_m, where $V_m \in (Des(V_i, G) \cup \{V_i\})$. At last, the CA broadcasts a rekeying message with form of $\{$Re*voking auser fromV*$_i \| pk_{ij} = H_2(ID_{ij})^{k_{ij}} \| \{(V_i, g^{r_i})\} \cup \{(V_m, g^{r_m}) | V_m \in Des(V_i, G)\}\}$. Once receiving this message, the users in class V_i compute the new encryption key of class V_i by

$$k_i' = e_n\left(pk_{i1}, \ldots, sk_{it}, \ldots, pk_{ij-1}, g^s, pk_{ij+1}, \ldots, pk_{it}, g^{r_{i_1}}, \ldots, g^{r_{i_{n_i}}}, g^s, \cdots . g^s\right),$$
$$= e_n\left(H_2(ID_{i1}), \ldots, H_2\left(ID_{ij-1}\right), g^s, H_2\left(ID_{ij+1}\right), \ldots, H_{ID_{it}}, g, \ldots, g\right)^{k_{i1}\cdots k_{ij-1}k_{ij+1}\cdots k_{it}r_{i_1}\cdots r_{i_{n_i}}s^{n-n_i-t+2}}. \quad (5)$$

Users in class V_m, where $V_m \in Des(V_i, G)$, also update the corresponding encryption key as described for key generation.

Note that the rekeying message broadcasted by the CA has no authentication. This drawback exists widely in all constructions of HKAS. It will suffer from "Man-in-the-Middle" attack, where an attacker can masquerade as the CA to send the rekeying message. For what concerns this security weakness, we can solve it by using a signature scheme.

4. Security Analysis

In this section, we show that the proposed scheme can resist various attacks through formal and informal security analysis. Then, we discuss the performance of our proposed scheme.

From the construction of our proposed scheme, we can see that such a scheme belongs to the HKAS and is based on the dependent key approach which refers to the users in each class. Thus the users should be considered in the security model. For this purpose, we modify Definition 2 by allowing the adversary $\mathcal{A}$ to corrupt some users. All the corrupted users are only in the classes whose access right is lower than that of the attacked class. In our proposed scheme, we assume that the encryption key of a class cannot be deduced from the private information of that class. We require that *Corrupt* can provide the encryption keys of some classes to the adversary, besides the public information *pub* and some private information sk_j. Finally, all the private information and encryption keys provided to the adversary are assigned to the classes whose access right is lower than that of the attacked class.

Theorem 1. *Our proposed scheme satisfies the KI-security, assuming that the DMDH problem is hard to be solved.*

Proof. In the proof, we need to show how to turn a static adversary $\mathcal{A}$ that can break our proposed scheme into a challenger $\mathcal{S}$ that can break the DMDH problem. Assume that the static adversary $\mathcal{A}$ chooses class V_m.

Once obtaining the parameters of DMDH problem: $g^{a_1}, \ldots, g^{a_{n+1}}$ and T, the challenger $\mathcal{S}$ sets $g^{a_1}, \ldots, g^{a_n}$ as the public key of the users in the class V_m and the public information of V_m's immediate predecessors. The public information of the system is denoted by $g^{a_{n+1}}$. Moreover, the challenger $\mathcal{S}$ randomly chooses $x' \in \mathbb{Z}_p$ to generate the private information of each class.

Observed from Definition 2, the only difference between $Exp^{KI-1}_{\mathcal{A},V_*}(1^\kappa, G)$ and $\mathrm{Exp}^{KI-0}_{\mathcal{A},V_*}(1^\kappa, G)$ is the input of $\mathcal{A}$, which corresponds to the real encryption key k_* and a random value $\rho \in \{0,1\}^{length(k_*)}$. The encryption key of class V_m is set as $k_m = T$. If $T = e_n(g, \ldots, g)^{a_1 \cdots a_{n+1}}$, then k_m is the real encryption key of class V_m. Otherwise, k_m is a random value in $\mathbb{G}_2$. The public information of class V_m is computed by $g^{(s_m' \oplus H_1(k_m))}$, where s_m' is computed by $x' \| \{1 \cdots 1\}^k$ along with hash function H_1.

For each class $V_i \in Anc(V_m, G)$, the challenger $\mathcal{S}$ randomly chooses $c_j \in \mathbb{Z}_p$ for user U_{ij}. The public key and private key of user U_{ij} are g^{c_j} and $(g^{a_{n+1}})^{c_j}$, respectively. Thus, the challenger $\mathcal{S}$ can compute the encryption key and public information of class V_l, where $V_l \in Anc(V_m, G)$ and $V_l \notin \mathrm{Pre}\, d(V_m, G)$.

Since class V_i and users in class V_i, where $V_i \in Anc(V_m, G)$, cannot be corrupted by the adversary $\mathcal{A}$ in the attack game, such modifications can be regarded as independent on the public and private information of classes in the adversary's view.

For each user in class V_t, where $V_t \notin (Anc(V_m, G) \cup \{V_m\})$, the challenger $\mathcal{S}$ randomly chooses $c \in \mathbb{Z}_p$ and sets g^c as the public key of this user. If $\mathcal{A}$ wants to corrupt this user, the challenger $\mathcal{S}$ returns $g^{a_{n+1}c}$ to the adversary $\mathcal{A}$, as the private key of this user. Then, the private information and the encryption key of class V_t are all allowed to be corrupted by the adversary $\mathcal{A}$, due to the fact that the challenger $\mathcal{S}$ has x' and the private key of the user in class V_t. Furthermore, the distribution of the encryption key k_t is the same as the one described in the key generation. Moreover, the public information of class V_t which can be computed by the challenger $\mathcal{S}$ is also provided to adversary $\mathcal{A}$.

Finally, $\mathcal{A}$ outputs a bit d as the response to whether the given value from $\mathcal{S}$ is the real encryption key of class V_m. And this output is also the answer of challenger $\mathcal{S}$ for the DMDH problem.

Thus, we have $Adv^{KI}_{\mathcal{A}}(1^\kappa, G) \le Adv^{DMDH}_{\mathcal{S},\mathcal{G}}(\lambda)$.

By DMDH assumption, we know that $Adv^{DMDH}_{\mathcal{S},\mathcal{G}}(\lambda)$ is negligible. Thus, we complete the proof of this theorem.

Collusive attack: let $V_i \le V_j$. Collusive attack means that an insider attacker in class V_i attempts to derive the encryption key of class V_j. The insider attacker only uses the public parameters, k_i, s_i, and his/her private key. The encryption key k_j in our proposed scheme is hidden in the value of the discrete logarithm with the first treatment of hash function. Due to the discrete logarithm and one-way hash function properties, it is computationally hard for this attacker. That is, our proposed scheme can resist the contrary attack.

Sibling attack: assuming that $V_i \le V_k$ and $V_y \le V_k$ are satisfied, neither the relation $V_i \le V_j$ nor $V_j \le V_i$ exists. The sibling attack considers whether a malicious user in class V_i can derive the encryption key of class V_j. This malicious user has to encounter the difficulty of computing the preimage of H_1 even if $d_i \le d_j$. Similarly, the attacker needs to solve the discrete logarithm problem and the CMDH problem if

Table 1: Security comparison among some existing HKASs.

Scheme	Security assumption	Secure against collusive attack	Secure against sibling attack	Secure against collaborative attack	Forward security	Backward security
Hwang-Yang scheme [19]	Integers factorization	×	√	√	√	√
Lo et al.'s scheme [21]	PKE with CPA-secure and hash function	√	√	√	×	√
Jeng and Wang scheme [24]	ECC	√	√	×	√	√
Chung et al.'s scheme [25]	ECC and hash function	√	×	√	√	√
Our scheme	CMDH and DMDH	√	√	√	√	√

starting from the public information g^{r_j} and the generation of k_j, respectively. It is an intractable problem for the malicious user. Hence, the proposed scheme is secure against this attack.

Collaborative attack: collaborative attack is the case when several users in the set of classes $CS = \{V_i | V_i \notin Anc(V_j, G)\}$ collaborate to derive the encryption key k_j. To launch such an attack, these users need to derive the master key from g^s or solve the CMDH problem. At least, these users must derive the preimage k_j from the one-way hash function H_1, if $V_i \not= V_j$ and $d_i \leq d_j$ exist. It is computationally infeasible to do these tasks. Thus, the proposed scheme can resist a collaborative attack.

Forward security: from the processes of removing an existing class or revoking user, we know that the encryption keys of the corresponding classes will be updated by the computation of the multilinear map. Consider, for example, the user revocation; the new encryption key of the corresponding class is obtained by substituting the public information g^s for the public key of the revoked user in the multilinear map. Since the private keys of other users in the corresponding class and the master key are unknown, the revoked user should solve the CMDH problem if he/she wants to obtain the new encryption key. It is impossible for the revoked user because of the CMDH assumption. If the revoked user wants to derive the new encryption key from the public information of the corresponding class, he/she has to deal with two intractable problems: solving the discrete logarithm and obtaining the preimage of a one-way hash function. Therefore, our proposed scheme can guarantee forward security.

Backward security: As previously stated, the involved users use the public information of the new inserted class or the new public information of the immediate predecessors to compute the new encryption key of the corresponding class when inserting a new class into the hierarchy. If adding a new user into a class, users in that class obtain the new encryption key by substituting the public key of this new user for anyone of public information g^s in the multilinear map. The encryption keys of these classes lower down in the hierarchy are all updated by the new public information. For the previous encryption keys of the affected classes, it is an instance of the CMDH problem to this new user. So, backward security is retained in our proposed scheme.

We compare our proposed scheme among some existing HKASs in terms of security. Table 1 gives the comparison results. □

5. Performance Analysis

It is known that computation, storage, and communication costs are the three main factors in the performance evaluation. For ease of exposition, let $n_i^1 = |Anc(V_i, G)|$, $n_i^2 = |Dec(V_i, G)|$ and $n_i^3 = |\mathrm{Prec}(V_i, G)|$. It is clear that $n_i^3 \leq n_i^1$.

In our proposed scheme, the storage overheads of each user are the size of his/her private key and private information of the corresponding class. To obtain the encryption key k_i, each user in the class V_i needs to compute one time of the multilinear map. Let $V_j \leq V_i$; users in class V_i need to compute $l - 1$ times of multilinear map and XOR operator, along with multiple times of hash function for deriving the encryption key of V_j, where l is the number of classes in the path from class V_i to class V_j. The times of hash function are certainly no more than $2|V|$.

The rekeying computation costs for the CA are $n_i^2 + 1$ times of the multilinear map, modular exponentiation, and XOR operator, along with multiple times of hash function when inserting or removing a class in the hierarchy. The communication cost for rekeying is one broadcast. The involved users need one computation of a multilinear map for obtaining the new encryption key of the corresponding class. Besides, the update of the private information for each affected class needs no more than one time of the hash function.

Although our proposed scheme belongs to a dependent key scheme, in the construction of encryption keys, it also focuses on each user. Thus the system public information for the dependent key scheme should also contain the public key of each user. The average number of users in each class is denoted by m. Besides these, we also set the average number of affected user groups in the independent key scheme as $n_i^1 + n_i^2$.

We compare our proposed scheme with Zhou et al.'s scheme [32] in terms of performance. The results are given in Table 2, where L_1 denotes the size of user's private key or the security parameter of a public key encryption scheme, L_2 represents the size of the ciphertext for a public key encryption scheme, and L_3 denotes the security parameter for multilinear map. Our proposed scheme may have some

Table 2: Performance comparison between our scheme and Zhou et al.'s scheme.

Scheme	Private information of each user (besides k_i)	Public information of system	Key derivation in dependent key scheme or obtaining encryption key in independent key scheme	Maximal computation and communication costs for rekeying
Zhou et al.'s scheme [32]	L_1	$(\|V\| + \|N\|)L_1$	$\approx (\lfloor \log_2 m \rfloor + 1)C_{PE}$	$\approx C_{MM}(n_i^1 + n_i^2) + C_H$ $+m(n_i^1 + n_i^2)C_{PE}$ $\approx m(n_i^1 + n_i^2)L_2 + (n_i^1 + n_i^2)L_1$
Our scheme	$\approx L_3 + \log_2\|V\|$	$(\|V\| + \|N\|)L_3 + \|V\|\log_2\|V\|$	$\leq (l-1)(C_{MM}(n_i^3 + m) + C_{XOR})$ $+(2\|V\| + l - 1)C_H$	$\leq (n_i^2 + 1)(C_{MM}(n_i^3 + m) + C_{ME} + C_{XOR}) + 2\|V\|C_H$, $\approx n_i^2(L_3 + \log_2\|V\|)$

Notations: $|V|$: number of classes in the access graph G. $|N|$: number of users in the access graph G. C_{PE}: computation cost for decryption of a public key encryption scheme. $C_{MM}(n)$: computation cost of n-multilinear map. C_{ME}: computation cost of modular exponent. C_{XOR}: computation cost of XOR operator. C_H: computation cost of hash function.

advantages over communication costs for rekeying, while the computation cost is a disadvantage for each user in the system. The parameter for the multilinear map used in our proposed scheme is less than that in Zhou et al.'s scheme with high probability. More importantly, our proposed scheme does not limit the number of user groups or access resources if the total number between users in a class and immediate predecessors of that class is a feasible value for a multilinear map. Once we employ the technology of virtual users, the computation cost of our proposed scheme is certainly less than that of Zhou et al.'s scheme.

6. Conclusion

In this paper, we propose an HKAS by using the building block of multilinear map for secure and flexible access control in IoT data markets. In our proposed scheme, the CA only updates the public information of each class for maintaining the hierarchical structure, and users in each class almost independently manage the corresponding encryption key via a one-round key agreement protocol. Moreover, the public information of the higher classes does not need to do any operation in dynamic environments. We show that the proposed scheme ensures KI-security based on the DMDH assumption. A shortcoming of our proposed scheme is that it only applies to the case of a very small amount of users, since the computation and storage costs for implementing the multilinear map are all expensive. To construct a simple and practical dependent key scheme without using a key assignment, novel ideas are expected, and we leave it as our future work.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant no. 61 902 079, the Scientific and Technological Key Project of Henan Province under Grant no. 192 102 210 283 and no. 202 102 210 399, the Key Scientific Research Project of Colleges and Universities in Henan Province under Grant no. 20A520040 and no. 21A520047, and the Open Project Foundation of Information Technology Research Base of Civil Aviation Administration of China under Grant no. CAAC-ITRB-201 707.

References

[1] A. I. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: a survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[2] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the 2007 IEEE Symposium on Security and Privacy (S&P'2007)*, pp. 321–334, IEEE, Oakland, CA, USA, May 2007.

[3] J. Hao, C. Huang, J. Ni, H. Rong, M. Xian, and X. S. Shen, "Fine-grained data access control with attribute-hiding policy for cloud-based IoT," *Computer Networks*, vol. 153, pp. 1–10, 2019.

[4] T. A. A. Victoire, "Secure sharing of IOT data in cloud environment using attribute-based encryption," *Journal of Circuits, Systems, and Computers*, vol. 30, no. 6, Article ID 2150102, 2021.

[5] M. Knapp, T. Greiner, and X. Yang, "Pay-per-use sensor data exchange between IoT devices by blockchain and smart contract based data and encryption key management," in *Proceedings of the 2020 International Conference on Omni-layer Intelligent Systems(COINS'2020)*, pp. 1–5, IEEE, Barcelona, Spain, September 2020.

[6] C. Chen, X. Deng, W. Gan, J. Chen, and S. H. Islam, "A secure blockchain-based group key agreement protocol for IoT," *The Journal of Supercomputing*, vol. 77, no. 8, pp. 9046–9068, 2021.

[7] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: the road ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015.

[8] S. Banerjee, S. Roy, V. Odelu et al., "Multi-authority CP-ABE-based user access control scheme with constant-size key and ciphertext for IoT deployment," *Journal of Information Security and Applications*, vol. 53, Article ID 102503, 2020.

[9] X. Lu, S. Fu, C. Jiang, and P. Lió, "A fine-grained IoT data access control scheme combining attribute-based encryption and blockchain," *Security and Communication Networks*, vol. 2021, Article ID 5308206, 13 pages, 2021.

[10] S. Seo, J. Won, S. Sultana, and E. Bertino, "Effective key management in dynamic wireless sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 371–383, 2015.

[11] P. Anantharaman, K. Palani, and S. W. Smith, "Scalable identity and key management for publish-subscribe protocols in the Internet-of-Things," in *Proceedings of the 9th International Conference on the Internet of Things (IoT'2019)*, p. 12, October 2019.

[12] M. Ma, G. Shi, and F. Li, "Privacy-oriented blockchain-based distributed key management architecture for hierarchical access control in the IoT scenario," *IEEE Access*, vol. 7, pp. 34045–34059, 2019.

[13] S. G. Akl and P. D. Taylor, "Cryptographic solution to a problem of access control in a hierarchy," *ACM Transactions on Computer Systems*, vol. 1, no. 3, pp. 239–248, 1983.

[14] H. R. Hassen, A. Bouabdallah, H. Bettahar, and Y. Challal, "Key management for content access control in a hierarchy," *Computer Networks*, vol. 51, no. 11, pp. 3197–3219, 2007.

[15] A. De Santis, A. Ferrara, and B. Masucci, "Efficient provably-secure hierarchical key assignment schemes," *Theoretical Computer Science*, vol. 412, no. 41, pp. 5684–5699, 2011.

[16] V. Odelu, A. K. Das, and A. Goswami, "A secure effective key management scheme for dynamic access control in a large leaf class hierarchy," *Information Sciences*, vol. 269, pp. 270–285, 2014.

[17] I. Lin, M. Hwang, and C. Chang, "A new key assignment scheme for enforcing complicated access control policies in hierarchy," *Future Generation Computer Systems*, vol. 19, no. 4, pp. 457–462, 2003.

[18] A. De Santis, A. L. Ferrara, and B. Masucci, "Cryptographic key assignment schemes for any access control policy," *Information Processing Letters*, vol. 92, no. 4, pp. 199–205, 2004.

[19] M. Hwang and W. Yang, "Controlling access in large partially ordered hierarchies using cryptographic keys," *Journal of Systems and Software*, vol. 67, no. 2, pp. 99–107, 2003.

[20] P. D'Arco, A. De Santis, A. L. Ferrara, and B. Masucci, "Variations on a theme by Akl and taylor: security and tradeoffs," *Theoretical Computer Science*, vol. 411, no. 1, pp. 213–227, 2010.

[21] J. Lo, M. Hwang, and C. Liu, "An efficient key assignment scheme for access control in a large leaf class hierarchy," *Information Sciences*, vol. 181, no. 4, pp. 917–925, 2011.

[22] C. Hsu and T. Wu, "Cryptanalyses and improvements of two cryptographic key assignment schemes for dynamic access control in a user hierarchy," *Computers & Security*, vol. 22, no. 5, pp. 453–456, 2003.

[23] S. Wang and C. Laih, "Cryptanalysis of Hwang–Yang scheme for controlling access in large partially ordered hierarchies," *Journal of Systems and Software*, vol. 75, no. 1, pp. 189–192, 2005.

[24] F. Jeng and C. Wang, "An efficient key-management scheme for hierarchical access control based on elliptic curve cryptosystem," *Journal of Systems and Software*, vol. 79, no. 8, pp. 1161–1167, 2006.

[25] Y. F. Chung, H. H. Lee, F. Lai, and T. S. Chen, "Access control in user hierarchy based on elliptic curve cryptosystem," *Information Sciences*, vol. 178, no. 1, pp. 230–243, 2008.

[26] Y. Lin and C. Hsu, "Secure key management scheme for dynamic hierarchical access control based on ECC," *Journal of Systems and Software*, vol. 84, no. 4, pp. 679–685, 2011.

[27] A. K. Das, N. R. Paul, and L. Tripathy, "Cryptanalysis and improvement of an access control in user hierarchy based on elliptic curve cryptosystem," *Information Sciences*, vol. 209, pp. 80–92, 2012.

[28] V. Odelu, A. K. Das, and A. Goswami, "An effective and secure key-management scheme for hierarchical access control in e-medicine system," *Journal of Medical Systems*, vol. 37, no. 2, pp. 1–18, 2013.

[29] A. Castiglione, A. De Santis, and B. Masucci, "Hierarchical and shared key assignment," in *Proceedings of the 17th International Conference on Network-Based Information Systems(NBiS'2014)*, pp. 263–270, IEEE, Salerno, Italy, September 2014.

[30] Y. Sun and K. J. R. Liu, "Hierarchical group access control for secure multicast communications," *IEEE/ACM Transactions on Networking*, vol. 15, no. 6, pp. 1514–1526, 2007.

[31] X. Gu, Y. Zhao, and J. Yang, "Reducing rekeying time using an integrated group key agreement scheme," *Journal of Communications and Networks*, vol. 14, no. 4, pp. 418–428, 2012.

[32] W. Zhou, Y. Xu, and G. Wang, "Decentralized group key management for hierarchical access control using multilinear forms," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 3, pp. 631–645, 2016.

[33] M. J. Atallah, M. Blanton, N. Fazio, and K. B. Frikken, "Dynamic and efficient key management for access hierarchies," *ACM Transactions on Information and System Security*, vol. 12, no. 3, p. 18, 2009.

[34] E. S. V. Freire, K. G. Paterson, and B. Poettering, "Simple, efficient and strongly KI-secure hierarchical key assignment schemes," in *Proceedings of the Cryptographers' Track at the RSA Conference 2013(CT-RSA'2013), Vol. 7779 of Lecture Notes in Computer Science*, pp. 101–114, Springer-Verlag, Berlin, San Francisco, USA, 2013.

[35] A. Castiglione, A. De Santis, and B. Masucci, "Key indistinguishability versus strong key indistinguishability for hierarchical key assignment schemes," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 4, pp. 451–460, 2016.

[36] M. Cafaro, R. Civino, and B. Masucci, "On the equivalence of two security notions for hierarchical key assignment schemes in the unconditional setting," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 485–490, 2015.

[37] T. Chen and J. Huang, "A novel key management scheme for dynamic access control in a user hierarchy," *Applied Mathematics and Computation*, vol. 162, no. 1, pp. 339–351, 2005.

[38] Y. Kim, A. Perrig, and G. Tsudik, "Simple and fault-tolerant key agreement for dynamic collaborative groups," in *Proceedings of the 7th ACM Conference on Computer and Communications Security(CCS'2000)*, pp. 235–244, ACM, Athens, Greece, November 2000.

[39] D. Boneh and A. Silverberg, "Applications of multilinear forms to cryptography," *Contemporary Mathematics*, vol. 324, no. 1, pp. 71–90, 2003.

[40] H. M. Lee, K. J. Ha, and K. M. Ku, "ID-based multi-party authenticated key agreement protocols from multilinear forms," in *Proceedings of the 8th International Information Security Conference(ISC'2005), Vol. 3650 of Lecture Notes in Computer Science*, pp. 104–117, Springer-Verlag, Berlin, Singapore, Singapore, 2005.

[41] H. Zhong and C. Xu, "ID-based multi-party authenticated key agreement protocols using multilinear forms," *Acta Electronica Sinica*, vol. 36, no. 10, pp. 1869–1872, 1890.

[42] H. Jia, Y. Hu, X. A. Wang, Z. Liu, and W. Xiong, "Extensional schemes of multipartite non-interactive key exchange from multilinear maps," in *Proceedings of the 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing(3PGCIC'2015)*, pp. 771–774, IEEE, Krakow, Poland, November 2015.

[43] H. Wang, D. He, J. Shen, Z. Zheng, X. Yang, and M. H. Au, "Fuzzy matching and direct revocation: a new CP-ABE scheme from multilinear maps," *Soft Computing*, vol. 22, no. 7, pp. 2267–2274, 2018.

[44] J. Zhao, B. Wei, and Y. Su, "Communication-efficient revocable identity-based signature from multilinear maps," *J. Ambient Intelligence and Humanized Computing*, vol. 10, no. 1, pp. 187–198, 2019.

[45] F. Ma and M. Zhandry, "The mmap strikes back: obfuscation and new multilinear maps immune to CLT13 zeroizing attacks," in *Proceedings of the 16th International Conference on Theory of Cryptography(TCC'2018), Vol. 11240 of Lecture Notes in Computer Science*, pp. 513–543, Springer-Verlag, Berlin, Panaji, India, November 2018.

[46] P. Austrin, P. Kaski, and K. Kubjas, "Tensor network complexity of multilinear maps," *Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik* in *Proceedings of the 10th Innovations in Theoretical Computer Science Conference(ITCS'2019), Vol. 124 of Leibniz International Proceedings in Informatics*, vol. 7, pp. 1–21, San Diego, USA, December 2019.