A Novel Authenticated Multi-party Key Agreement for Private Cloud

Tuo He[1], Maode Ma[2], Wenping Ma[1] and Bingsheng He[3]
[1]State Key Laboratory of Integrated Service Networks
Xidian University, Xi'an, China
[2]School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore
[3]School of Computer Engineering
Nanyang Technological University, Singapore

Abstract— **Cloud computing technology is an emerging technology for various types of vast information to be processed in the data centers to overcome the serious shortage of resource such as space, power and cost, etc. Along with the development of cloud computing, security of cloud computing is becoming more and more critical. In this paper, we aim to propose a set of secure and efficient authenticated multi-party key agreement protocols based on the hierarchical identity-based cryptography (HIBC) for private cloud. For different scenarios, we design loop-level and cascade-level authentication protocols among users in a private cloud of hierarchical structure, respectively. The new multi-party protocols are supposed to be more efficient and secure than the other existing solutions.**

Keywords—Hierarchical identity-based Cryptography (HIBC); Key agreement protocol; Authentication for cloud computing.

I. INTRODUCTION

Cloud computing is a large-scale distributed computing paradigm [1]. Since it can reduce the burdens and costs of small and medium-sized enterprises on data storage and maintenance for the information processing, this technology has made rapid development. According to different deployments of the infrastructures, the cloud computing can be roughly divided into public cloud and private cloud. The former one offers its computing resources as a service to general public, the latter one, which is designed for exclusive use by a single organization, offers higher degree of control over performance, reliability and security. There are many security concerns in both public cloud and private cloud, in this paper authentication process in private cloud, which needs to be disposed. In a private cloud, along with the development of central resource, each user will not need strong storage and computation power but central server, it will require more and more efficient communication for information sharing. The hierarchical structure system, where each entity will have its distinct position at different layers, can be appropriate to be deployed in a private cloud such as an enterprise. Consider that several entities may compose a temporary group for efficient communication, the multi-party authentication key agreement protocol is needed. However, few of existing work have addressed the issue of secured authenticated key agreement for the communication among the entities at hierarchical structure.

To achieve an authenticated multi-party key agreement in classical wireless networks, an identity based hierarchical circle model has been proposed by Eskeland et al [2]. By this scheme, there are several hierarchical security circle classes and arbitrary amount of users, who locate in the same class, are allowed to establish a common secret class key. This scheme can provide an efficient authentication and achieve a multiple party key agreement. But the distribution of user's private key is executed by a trusted third party (TTP), which is an additional trusted party that may be infeasible for some scenarios in practice. The proposed protocol cannot support key agreement among users at different layers, and thus is not suitable for our private cloud of hierarchical structure scenario.

An authenticated key agreement scheme has been proposed in [3], which introduces a tree based structure for a multiple-level cloud computing. By this scheme, a trusted authenticator (TA) locating at the upper level of the tree structure, takes the charge of distributing public/private key pairs for the users at the lower layers. By using the identity-based authentication key agreement framework presented in [4], this scheme can successfully achieve mutual authentications among users in cloud computing. But it has been designed only as a mutual key agreement protocol, which can not be directly extended to a multi-party scenario. Furthermore, the users are assumed to be totally trusted, which is not feasible. It is also possible for any user in the same trusted authentication domain to launch a man-in-the-middle attack between other two users so that this scheme is not secure enough for our private cloud scenario.

The identity-based hierarchical model for authenticated key agreement has been first proposed for cloud scenario in [5], where the proposed protocol skillfully devises a private key distribution process and successfully proposes a practical hierarchical model by HIBC. But the key reconstruction is so complex that lots of nodes are involved. By changing the key distribution scheme, a private key in [6] has been designed to only employ the nodes in the immediate upper level, which can largely reduce the number of the nodes required and the number of parameters for key reconstruction. However, both of the two schemes have just focused on the authentication between users with the cloud service provider or a delegated virtual machine, while key agreement process among users has not been addressed.

978-1-4799-2003-7/14/$31.00 ©2014 IEEE

In this paper, evolved from the hierarchical identity-based models in [5-6], we design a hierarchical identity based cryptography (HIBC) system for private cloud and propose loop-level and cascade-level authenticated multi-party key agreement protocols for different scenarios as our major contributions. Compared to the existing multi-party key agreement schemes, our scheme cannot only securely and efficiently achieve an authenticated key agreement but also can provide a limited key delegation and a damage control, which will enhance the practicality of protocols in private cloud.

The rest of the paper is organized as follows. In Section II, we review the preliminary knowledge and present a practical hierarchical key distribution structure for private cloud. In Section III, based on the presented hierarchical structure, we design two multi-party key agreement authentication protocols respectively for loop-level and cascade-level scenarios. In Section IV, we give security evaluation and efficiency analysis on the proposed protocols. At last, in Section V, we conclude the paper with a summary.

II. PRELIMINARIES

In this section, we first state the security goals of our designed scheme and then we present a hierarchical model, which will pave the way of the development of our protocols.

A. Security Goals

To be suitable for cloud computing, our proposed scheme needs to be designed to resist any known malicious attacks. We assume that an adversary has been a former participant and may hold a former session key. The security properties must be satisfied in the presence of the passive adversaries with such capabilities. As an active adversary, he may modify any session key message or attempt to impersonate any legitimate user by replacing the session key message which was known before.

We list some desirable security properties that need to be identified. Typically, the importance of these properties will depend on the particular application.

- Known keys secure. A protocol can still achieve its goal in the face of an adversary who has learned some previous session keys.
- (Perfect) forward secrecy. If long-term secrets of one or more entities are compromised, the secrecy of previous session keys would be not affected.
- Unknown key-share secure. Entity i cannot be coerced into sharing a key with another entity without i's knowledge, i.e., when i (correctly) believes the key is shared with entity j, but j believes the same key is shared with some entity $k \neq i$.
- Key-compromise impersonation. Suppose i's secret value is disclosed. Clearly an adversary that knows this value can now impersonate i because by this value, i can be precisely identified. However, it may be desirable that this loss cannot enable an adversary to impersonate other entities communicate to i.
- No key control. Neither entity can predetermine any portion of the shared secret key being established.

B. Hierarchical Identity Based Cryptography

Recently, a new technology named identity based cryptography (IBC) [7] has been developed quickly, for the direct derivation of public keys from a known identity of the users (e.g. email address). This cryptography technology can eliminate the need for certificates compared to the traditional certificate based authentication technology. Based on IBC primitive, Gentry [8] has proposed a hierarchical identity based cryptography (HIBC) with a hierarchical structure to alleviate the workload of the root node. By Gentry's HIBC, a root private key generator (PKG) is able to distribute the workload by delegating private key generation and identity authentication to low-level PKGs, who in turn generate private keys for user in their domains in the next layer. Another advantage is to hold a damage control, which implies that the disclosure of a secret key of a domain PKG cannot compromise the secret key of higher-level PKGs. Due to these features, HIBC has been widely applied to various network scenarios.

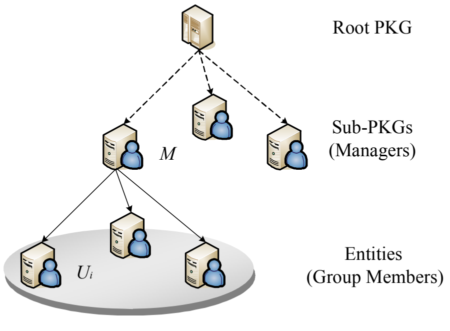

Fig. 1 Hierarchical Structure Model for Private Cloud

C. Hierarchical Structure Model for Private Cloud

In the private cloud scenario under the consideration, there is a hierarchical structure model for the entities at different layers. The entity at the upper layer can manage and escrow a common session key of the selected entities at lower layer to form a temporary group. The HIBC can be adopted to support the hierarchical architecture. The hierarchical structure in [5-6] will be selected for cloud computing, as shown in Fig. 1. There are L levels in the hierarchical structure, the node at the top level is the root PKG, while nodes at the middle levels are sub-PKG with the entities located at the bottom. The upper level's sub-PKG node, who can be viewed as manager M, has direct control power to its adjacent child node, e.g., U_i who can be viewed as an employee of manager M.

III. AUTHENTICATED KEY AGREEEMTN SCHEME

In this section, we describe our proposed scheme, which consists of the following two phases, key agreement preparation phase and authenticated key agreement phase.

As assumed in the hierarchical model, the root PKG node locates at top layer, the sub-PKG node $U_{t-1,M}$ locates at one middle layer $t-1$. The entities $\{U_{t,1}, U_{t,2}, ..., U_{t,N}\}$, as the nodes $\{ID_{t,1}, ..., ID_{t,N-1}, ID_{t,N}\}$, locate at adjacent lower layer *t*. A temporary team μ consists of the several group members in the considered hierarchical structure. The notions used in the scheme are defined in Table I.

TABLE I NOTATIONS

Notation	Description
N	Number of group member
$U_{t,i}$	Entity U_i in *t* level
$r_{t,i}$	Fresh random number of $U_{t,i}$
$P_{t,i}$	Public key of $U_{t,i}$
$S_{t,i}$	Private key of $U_{t,i}$
$x_{t,i}$	Temporary public key of $U_{t,i}$
K_μ	Agreement session key

A. Key Agreement Preparation

This phase is described as a preparation for authenticated key agreement. As one of the team members, $U_{t,i}$ acquires public/private key pair from $U_{t-1,M}$, where $i \in (1, ..., N)$. Then, the hierarchical system will be initialized as follows.

(1) *Root PKG setup*: Root PKG runs parameter generator g to generate groups G_1 and G_2 of some prime order q and an admissible pairing $\hat{e}: G_1 \times G_1 \rightarrow G_2$. Then it chooses cryptography hash function: $H: \{0,1\}^* \rightarrow Z_q^*$. Root PKG sets $P_0 = H(ID_0)P$, $P_0 \in G_1$, which can be viewed as a generator of G_1. Then it picks a random $s_0 \in Z_q^*$ and sets $S_0 = s_0 P_0$, $Q_0 = s_0 P$. The root PKG's master key is S_0 and the system parameters are $< G_1, G_2, \hat{e}, H, Q_0, P >$.

(2) *Lower-level setup*: The upper node $U_{t-1,M}$ (sub-PKG), who locates at middle level $t-1$, randomly picks a secret value $s_{t,M} \in Z_q^*$, which is secretly preserved by $U_{t-1,M}$ itself.

(3) *Extract key pair*: Each node in the cryptosystem has its owned identifier, which is cascade of each node at upper layer, e.g., identifier of $U_{t,i}$ is $\{ID_0, ..., ID_{t-1,M}, ID_{t,i}\}$. To generate public/private key, node $U_{t-1,M}$ computes: $P_{t,i} = H(ID_0 \| ... \| ID_{t-1,M} \| ID_{t,i})P$, and by using $P_{t,i}$ sets $S_{t,i} = S_{t-1,M} + s_{t,M} P_{t,i}$, $Q_{t-1,M} = s_{t,M} P$, where $S_{t-1,M}$ is $U_{t-1,M}$'s private key generated by parent node $U_{t-2,L}$. Node $U_{t-1,M}$ sends generated public/private key pair $< P_{t,i}, S_{t,i} >$ to entity $U_{t,i}$, where $Q_{t,M}$ is public.

In the same process, all entities nodes in the hierarchical cryptosystem get their public/private key pair, e.g., $U_{t,i}$ gets its key pair $< P_{t,i}, S_{t,i} >$ from upper node. After this key agreement preparation phase, we have built a hierarchical identity based cryptosystem, based on which the key agreement scheme can work to get agreements among the entities nodes.

B. Authenticated key Agreement

In this phase, a novel authenticated multi-party key agreement scheme is presented for the group users who locate in the hierarchical structure. According to loop-level and cascade-level scenarios, the scheme consists of two different key agreement protocols.

1) Loop-Level Key Agreement Protocol (protocol I)

In the considered scenario, the entities at the same layer can be selectively invited to join in a temporary team in a hierarchical system. The participants firstly need to be authenticated by their identity and then generate a common session key for interim communications. Base on the structure of HIBC, we have a new idea to design a key agreement authentication scheme, which will be able to provide efficiency and security to generate a session key for group session among the group members. The progress of multi-party key agreement for group member $U_{t,1}, U_{t,2}, ..., U_{t,N}$ acts as following:

Step1. Each participant $U_{t,i}$ selects a random $r_{t,i} \in Z_q^*$, generates the temporary public key $x_{t,i} = r_{t,i} P$, computes verification information $w_{t,i} = S_{t,i} + r_{t,i} P_{t,i}$, $v_{i,j} = r_{t,i} P_{t,j}$, keeps $v_{i,i} = r_{t,i} P_{t,i}$. Participant $U_{t,i}$ sends the key agreement information $< ID_{t,i}, x_{t,i}, w_{t,i}, v_{i,j} >$ to another member $U_{t,j}$. By similar information generation method, $U_{t,i}$ sends the key agreement information messages to other group members.

Step2. (Authentication) Another participants, $U_{t,j}$ who locates in the same level and owns public value $Q_{t,M}$ can verify the identity of key agreement information's sender $U_{t,i}$, if $\hat{e}(P, w_{t,i}) = \hat{e}(x_{t,i}, P_{t,i}) \hat{e}(P, S_{t-1,M}) \hat{e}(Q_{t,M}, P_{t,i})$, where $U_{t,j}$ can compute $\hat{e}(P, S_{t-1,M}) = \hat{e}(P, S_{t,j}) / \hat{e}(Q_{t,M}, P_{t,j})$.

Step3. (Session Key) After authenticating identity of each participant, the group member $U_{t,j}$ can compute $v_{t,i} = \hat{e}(x_{t,i}, S_{t,j}) / \hat{e}(Q_{t,M}, v_{i,j}) = \hat{e}(x_{t,i}, S_{t-1,M})$, and for briefness, the private key $S_{t-1,M}$ of $U_{t-1,M}$ can be described as:

$$\begin{aligned} S_{t-1,M} &= S_{t-2,L} + s_{t-1,L} P_{t-1,M} = s_0 P_0 + \cdots + s_{t-1,L} P_{t-1,M} \\ &= [s_0 H(ID_0) + \cdots + s_{t-1,L} H(ID_0 \| \cdots \| ID_{t-1,M})]P \\ &= [s_0 H_0 + \cdots + s_{t-1,L} H_{t-1,M}]P = \Delta_{t-1,M} P \end{aligned},$$

where $H_{t,i} = H(ID_0 \| ... \| ID_{t-1,M} \| ID_{t,i})$, $\Delta_{t-1,M} \in Z_q^*$, thus $v_{t,i} = \hat{e}(x_{t,i}, S_{t-1,M}) = \hat{e}(P,P)^{r_{t,i}\Delta_{t-1,M}}$. Group member $U_{t,j}$ can obtain the session key: $K_{t,\mu} = \prod_{\alpha=1}^{N} v_{t,\alpha} = \hat{e}(P,P)^{\Delta_{t-1,M} \sum_{\alpha=1}^{N} r_\alpha}$.

After the above three steps, by using the same method, all members in the temporary team can share a common session key $K_{t,\mu}$, the identity of each participant has also been verified.

2) Cascade-level Key Agreement Protocol (protocol II)

Considering other scenarios, the participators of temporary session group may locate at different layers but same trusted domain, the upper node play the role of parent node for adjacent lower node. In our second scenario, the entities in the cascade levels (under same ancestor node) compose a temporary group for a session of communication. The cascade-level key agreement protocol for group member $U_{l,i}, U_{l+1,i}, ..., U_{l+N,i}$ is shown as follows.

Step1. Each participant $U_{t,i}$ in cascade-level temporary group, where $t \in \{l, ..., l+N\}$, selects a random $r_{t,i} \in Z_q^*$ and generates temporary public key $x_{t,i} = r_{t,i}P$, computes verification information $w_{t,i} = S_{t,i} + r_{t,i}P_{t,i}$, then sends key agreement information $< ID_{t,i}, x_{t,i}, w_{t,i} >$ to ancestor node $U_{l,i}$. Other participants send the information by similar method, and ancestor node $U_{l,i}$ keeps its own selected random $r_{l,i}$.

Step2. (Authentication) The group member $U_{l,i}$, who locates at the highest rank *l* of the linear level, has the ability to verify identity of each descendant node by its private key $S_{l,i}$ and verification information $w_{t,i}$, if $\hat{e}(P, w_{t,i}) = \hat{e}(x_{t,i}, P_{t,i})\hat{e}(P, S_{t,i})$, where $\hat{e}(P, S_{t,i}) = \hat{e}(P, S_{l,i}) \prod_{\beta=l+1}^{t} \hat{e}(Q_{\beta,i}, P_{\beta,i})$.

Step3. (Session Key) After authenticating identity of each entity, ancestor node $U_{l,i}$ can compute partial session key with the key agreement information for each group member:

$$K_{t,i} = \prod_{\beta=l,\beta\neq t}^{l+N} \hat{e}(S_{l,i}, x_{\beta,i}) = \prod_{\beta=l,\beta\neq t}^{l+N} \hat{e}(\Delta_{l,i}P, r_{\beta,i}P) = \hat{e}(P,P)^{\Delta_{l,i} \sum_{\beta=l,\beta\neq t}^{l+N} r_{\beta,i}}.$$

Then the node $U_{l,i}$ returns partial session key $K_{t,i}$ to each cascade node $U_{t,i}$ by the secure channel, which is used to distribute public/private key pair. Each participant $U_{t,i}$ can get the session key by its private key $S_{t,i}$ and random $r_{t,i}$:

$$K_{\mu,i} = K_{t,i}\hat{e}(S_{l,i}, x_{t,i}) = \prod_{\beta=l}^{l+N} \hat{e}(S_{l,i}, x_{\beta,i}) = \hat{e}(P,P)^{\Delta_{l,i} \sum_{\beta=l}^{l+N} r_{\beta,i}},$$

where $\hat{e}(S_{l,i}, x_{t,i}) = \hat{e}(S_{t,i}, x_{t,i}) / \prod_{\beta=l+1}^{t} \hat{e}(Q_{\beta,i}, P_{\beta,i}r_{t,i})$.

In both scenarios, to ensure the timeliness of verification information, the key agreement information is required to be secretly transmitted with timestamp. Based on the designed multi-party key agreement scheme, the protocols cannot only allow that each participant acquires the session key for group communication, but also can disperse the workload from root manager to all group participants.

IV. Evaluation of Authentication Scheme

In this section, we give a detailed security analysis of our protocols and present an efficiency comparison with other existing schemes.

A. Security Analysis

In this sub-section, we first discuss the designed goal of the protocols obtained, and give a detailed security analysis to the proposed protocols respectively under the passive and active attack models.

Theorem 1 *Assume proposed protocols are run by multi-party users, the protocols can provide key secrecy and user authentication.*

Proof: 1) Key Secrecy: In the loop-level scenario, lack of the uniform management of an upper node, each participant $U_{t,\alpha}$ has to firstly compute intermediate value from other participants, e.g. to get the value $v_{t,i}$ for $U_{t,i}$, $U_{t,j}$ computes $v_{t,i} = \hat{e}(x_{t,i}, S_{t,j}) / \hat{e}(Q_{t,M}, v_{i,j}) = \hat{e}(P,P)^{r_{t,i}\Delta_{t-1,M}}$, and it is infeasible to find out $v_{t,i}$ without $U_{t,j}$'s private key $S_{t,j}$ and transmitted message $v_{i,j}$ from $U_{t,i}$. The participant $U_{t,j}$ can get the session key $K_{t,\mu}$ by the product of $v_{t,\alpha}$, where $\alpha \in ¥$. In the cascade-level scenario, each participant $U_{t,j}$ gets partial session key $K_{t,j}$ from highest rank node $U_{l,i}$. It is computational infeasible for an adversary to get $K_{t,j}$ without knowing private key $S_{l,i}$. After receiving partial key $K_{t,j}$, the participant $U_{t,j}$ can get the session key $K_{\mu,i} = K_{t,i}\hat{e}(S_{l,i}, x_{t,i})$, to get $\hat{e}(S_{l,i}, x_{t,i})$, it needs participant $U_{t,i}$'s private key $S_{t,i}$, fresh random $r_{t,i}$ and public values $Q_{l+1,i}, ..., Q_{t,i}$. Thus in the generation process of session key K_μ, the privacy has been well kept. *2) User Authentication*: Both two key agreement protocols use the same identity authentication approach, which is analogous to the way in [9-10]. For first scenario, all the participants locate at same level, the verification message of each participant is constituted by $w_{t,i} = S_{t,i} + r_{t,i}P_{t,i}$, where $r_{t,i}$ is a fresh random that selected by $U_{t,i}$, and $S_{t,i}$ is private key of group member $U_{t,i}$. Other participants are able to compute $\hat{e}(P, w_{t,i}) = \hat{e}(x_{t,i}, P_{t,i})\hat{e}(P, S_{t-1,M})\hat{e}(Q_{t,M}, P_{t,i})$, where the value $\hat{e}(P, S_{t-1,M})$ can be further computed by their own private keys and public value $Q_{t,M}$; for second scenario, the ancestor node can verify the signature message by computing $\hat{e}(P, w_{t,i}) = \hat{e}(x_{t,i}, P_{t,i})\hat{e}(P, S_{t,i})$, and $\hat{e}(P, S_{t,i})$ can be computed by private key $S_{l,i}$, and public values $Q_{l+1,i}, \cdots, Q_{t,i}$. According to property of bilinear map, the verification process will be effective and security.

Assumption 1 *Bilinear Diffie-Hellman (BDH)* problem: $\forall a,b,c \in_R Z_q^*$, given aP, bP and cP, it is difficult to compute $\hat{e}(P,P)^{abc}$.

Then we will simply analysis the resistance of proposed protocols to passive adversary and active adversary by demonstrating equivalence to variations of the *BDH* problem. We assume that *BDH* problem is computationally infeasible. Also, we assume that the hash function *h* behaves like a random oracle, in the sense its output cannot be distinguished from random output.

Definition 1 *Passive attack model*: A passive adversary whose aim is key recovery for a given session involves eavesdropping on messages passed between participants for session. The attack is successful if the session key can be recovered with a probabilistic polynomial time algorithm given as input, the eavesdropped message passes as well as any other publicly available parameters.

Theorem 2 *Protocols are secure against passive attack and provide security attributes of forward secrecy and known session key security unless the BDH problem can be solved.*

Proof: 1) Passive Adversary: We assume that a passive adversary cloud eavesdrop all traffic. Due to the session key is generated by each participant's private key and chosen fresh random, such as $S_{t,i}$ and $r_{t,i}$, the adversary has no information about them but may have the ability to intercept and capture the transmitted messages in the public network. The passive off-line adversary can get the key agreement information $x_{t,i}$, $v_{i,j}$ for first scenario and $x_{t,i}$ for second scenario, where $x_{t,i} = r_{t,i}P$, $v_{i,j} = r_{t,i}P_{t,j}$ and $Q_{t,M} = s_{t,M}P$, thus the cryptanalysis of the session key $K_{t,\mu}$ or $K_{\mu,i}$ can be reduced to break the variations of *BDH* problem. Thus both protocols resist passive attacks. *2) Forward Security*: Suppose that both participant $U_{t,i}$'s long-term private key $S_{t,i}$ and its transmitted key agreement messages are compromised, however the adversary still cannot obtain the previously established session key $K_{t,\mu}$ or $K_{\mu,i}$, because temporary random that decide the value of session key are varied in each session, no useful or relevant information can be derived from the presented key agreement messages. Thus the leakage of participants' long-term private keys will not impact the security of the previously established session keys. *3) Known Session Key Security*: For both protocols, the session keys are consists of fresh random and private key of upper node. Suppose the adversary may not only own the transmitted key agreement message but also get previous session key, which is the product of $\hat{e}(S_{t-1,M}, x_{t,\alpha})$ or $\hat{e}(S_{l,i}, x_{\beta,i})$, where $\alpha, \beta \in Z_N$. Due to the varied temporary random, it is no use to get information from presented transmitted messages, even if the transmitted messages relevant to compromised session key are exposed, such as $x_{t,\alpha}$ and $x_{\beta,i}$, the adversary may have no ability to compute private key $S_{t-1,M}$ or $S_{l,i}$ based on the security of variations of *BDH* problem. As on the other hand, the fresh random ensure the difference of session keys. Thus leaking of previous session key will not affect the use of the new session key.

Definition 2 *Active attack model*: An active adversary involves an attacker who is given access to all publicly available information and attempts to successfully complete a protocol with other participant by impersonating one member. Review the key agreement protocols are successful if each of the participants accepts the information of others, and terminate with the same session key.

Theorem 3 *Protocols are secure against active attack and provides security properties of impersonation attack, Key-compromise impersonation and no key control.*

Proof: *1) Active Adversary*: An active adversary may be any former group member who has leaved the group. In both scenarios, before generating the session key, the participant's identity $ID_{t,i}$ needs to be verified by other participants or upper node. Without being accepted as a session participant, the adversary cannot participate in the calculation of common session key. As we discussed above, although former group member owns previous session keys and transmitted key agreement information, it is computational infeasible to recover any useful information, such as upper node $U_{t-1,M}$'s private key $S_{t-1,M}$. Thus both protocols can resist active imitative and forge attack. *2) Key-compromise impersonation*: In both scenarios under consideration, if an adversary obtains $U_{t,i}$'s public key $P_{t,i}$, previous session keys, even used random $r_{t,i}$, the adversary cannot be authenticated as the identity $ID_{t,i}$ without its private key $S_{t,i}$. If $S_{t,i}$ has been compromised, the adversary can only impersonate $U_{t,i}$, but has no ability to impersonate other participants to $U_{t,i}$, because the generation of each participant's verification information needs their respective private keys. And although adversary owns the verifiable transmitted identity information from other participants, but the transmitted verification information is not fresh, the adversary cannot disguise as the senders to $U_{t,i}$ successfully. *3) No key control*: The session key is generated by the product of all members' private keys and the fresh random numbers in both two scenarios. Thus none of the participant $U_{t,i}$ can predetermine any portion of the shared session key. Although the parent node knows children nodes' private keys and can directly delegate them, the random numbers selected by the session participants ensure the freshness of the session key.

B. *Efficiency Analysis*

We provide comparisons between our proposal and several existing schemes on the properties and communication cost. By the detailed comparison, we further show the effectiveness of the proposed authentication scheme.

1) Advantage of Proposed Scheme

The outstanding advantages of our protocols are two features including the hierarchical structure and the limited key delegation. By a construction of hierarchical key distribution, the protocols can successively distribute the workload from an upper parent node to its lower child nodes,

TABLE II
ADVANTAGE OF SCHEME

Scheme	Authen tication	Multi-party Key agreement	Hierarchical structure	Key delegation
[3-4]	YES	NO	NO	NO
[5-6]	YES	NO	YES	NO
[9]	YES	YES	NO	NO
[6, 12]	YES	YES	YES	NO
Protocol I	YES	YES	YES	YES
Protocol II	YES	YES	YES	YES

TABLE III
COMPARISON OF COMPUTATION COST

Scheme	Member Verification	Key Generation	Transmitted Message Cost
[9]	$2 \cdot C_{BM} + C_{cmp}$	C_h	C_h
[10]	$2 \cdot C_{BM} + C_{cmp}$	$(N-1)C_{BM}$	C_h
Protocol I	$3 \cdot C_{BM} + C_{cmp}$	$2N \cdot C_{BM}$	*Null*
Protocol II	$3 \cdot C_{BM} + C_{cmp}$	$N \cdot C_{BM}$	$(N-1)C_{BM}$

The computational cost composes three sources, which are C_{BM}, the cost for computing a bilinear map, C_{cmp}, the cost for comparing two content texts, and C_h, the cost for a hash function.

which can largely save the required storage and computation power for users. Both protocols can support key delegation because the private key derivation is carried out from a parent node to its child nodes. Since the manager node owns the public/private key pairs of its adjacent child nodes and can totally substitute them, it is natural for the manager node to be authorized and authenticated and to generate the group session key K_μ for its child nodes, when they are not free or off-line.

The comparison with the existing schemes in Table II, we clearly show the advantages of our proposed protocols for cloud computing. Compared with the schemes in [3-4], our solution supports efficient key agreement protocols among multiple users in a hierarchy structure. To compare with the schemes in [5-6, 9-12], our protocols acquire limited key delegation that can be useful in practical applications.

2) Cost Comparison

In order to show the efficiency of our protocols, we present a cost comparison between the proposed protocols with some existing key agreement schemes. We select schemes in [9, 10] for comparison because the two key agreement schemes are designed for the multi–party network scenarios and have efficient computing performance. By choosing arbitrary participants, we present the member of verifications and the key generations and the cost of message transmission in the process of consultation on session key to analyze the protocol's computation and communication cost.

The comparison with the schemes in Table III, we can show that our protocols have a similar cost that they need only an additional C_{BM} operation compared with the schemes in [9-10] on the member verifications. From the aspects of key generation and message transmission, Protocol I and protocol II achieve the mutual conversion of the cost. Compared to existing schemes, they respectively need additional N and $N-1$ operations of bilinear pairing, but it is not important for cloud computing.

V. CONCLUSION

With the development of cloud computing, user does not need strong storage and computation power, achieving secure and efficient communication among users recently becomes research hotspot. To resolve this problem, we have derived the authenticated multi-party key agreement scheme based on the hierarchical identity-based cryptography system for private cloud. The proposed authenticated key agreement scheme is proved to be semantic secure and more efficient than the existing key agreement schemes such as the ones in [3-12]. In this paper, we have focused ourselves on the discussion on the multi-party key agreement for private cloud users in the same trusted domain, while it can be further extended to the multi-party key agreement among cloud entities in different trusted domains. We will study this issue in our further work.

REFERENCES

[1] M. Armbrust, A. Fox and R. Griffith, "A View of Cloud Computing," Communications of the ACM 2010, Vol. 53, No. 4, April 2010, pp. 50~58.

[2] S. Eskeland, V. Oleshchuk, "Hierarchical Multi-Party Key Agreement for Wireless Networks," Third International Symposium on Information Assurance and Security, 2007.pp. 29~31.

[3] L. Kang and X. Zhang, "Identity-based Authentication in Cloud Storage Sharing," Proceedings of International Conference on Multimedia Information Networking and Security 2010, MINES2010, 2010, pp. 851~855.

[4] N. P. Smart, "Identity-based Authenticated Key Agreement Protocol Based on Weil Pairing," Electronics Letters, Vol. 38, No. 13, June 2002, pp. 630~632.

[5] H. Li, Y. Dai, L. Tian and H. Yang, "Identity-based Authentication for Cloud Computing," Cloud Computing, Vol. 5931, 2009, pp.157~166.

[6] J. Huang, I. Liao, and C. -K. Chiang, "Efficient Identity-based Key Management for Configurable Hierarchical Cloud Computing Environment," Proceedings of IEEE 17th International Conference on Parallel and Distributed Systems, ICPADS 2011, pp. 883~887.

[7] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes," Advance in Cryptology, Vol. 196, 1985, pp. 47~53.

[8] C. Gentry and A. Silverberg, "Hierarchical ID-based Cryptography," Advance in Cryptology – Aisacrypt 2002, Vol. 2501, 2002, pp. 149~155

[9] X. Liu, Q. Zhang, C. Wang, "New scheme of ID-based authenticated multi-party key agreement," Journal of Computer Applications, 2011, pp. 1302~1304.

[10] S.J. Wang, "Hierarchical Key Agreement Protocols in Group-User Systems," Fifth International Conference on Information Assurance and Security, IAS '09, 2009, pp. 259 ~262.

[11] G. Rosario et al., "Strongly-Resilient and Non-interactive Hierarchical Key-Agreement in MANETs," 13th European Symposium on Research in Computer Security (ESORICS), 2008, pp. 49~65.

[12] S.Y. Lim, M.H. Lim, "Energy-Efficient and Scalable Group Key Management for Hierarchical Sensor Network," Journal of Ubiquitous Systems & Pervasive Networks, Vol. 2, No. 1, 2011, pp. 39~47.

[13] C. Schridde and T. Dornemann, "An Identity-based Security Infrastructure for Cloud Environments," Proceedings of IEEE International Conference on Wireless communications, Networking and Information Security 2010, WCNIS 2010, pp. 644~649.

[14] E. Klaoudatou, et al., "A Survey on Cluster-Based Group Key Agreement Protocols for WSNs," IEEE Communications Surveys & Tutorials, Vol. 13, Issue 3, 2011, pp. 429~442.