TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 13/18 pp234-241
Volume 14, Number 2, April 2009

Efficient User Authentication and Key Management for Peer-to-Peer Live Streaming Systems*

LIU Xuening (刘雪宁), YIN Hao (尹 浩)**, LIN Chuang (林 闯), DU Changlai (杜长来)

Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

Abstract: Recent development of the peer-to-peer (P2P) live streaming technique has brought unprecedented new momentum to the Internet with the characters of effective, scalable, and low cost. However, before these applications can be successfully deployed as commercial applications, efficient access control mechanisms are needed. This work based on earlier research of the secure streaming architecture in TrustStream, analyzes how to ensure that only authorized users can access the original media in the P2P live streaming system by adopting a user authentication and key management scheme. The major features of this system include (1) the management server issues each authorized user a unique public key certificate, (2) the one-way hash chain extends the certificate's lifetime, (3) the original media is encrypted by the session key and delivered to the communication group, and (4) the session key is periodically updated and distributed with the media. Finally, analyses and test results show that scheme provides a secure, scalable, reliable, and efficient access control solution for P2P live streaming systems.

Key words: peer-to-peer (P2P) live streaming; user authentication; key management; hash chain; media-dependent

Introduction

Live streaming has always been envisioned as one of the potential killer applications for the Internet, yet is perhaps the greatest unfulfilled promise. Many challenges have been identified and analyzed in the design of video streaming systems such as the need for high bit rates, end-to-end delay requirements, packet losses, network congestion, service guarantees, and security. In the Internet research community, multicasting has been shown as a promising technique to significantly reduce duplication in data transmissions. Two types of systems have been proposed with multicasting in the IP layer and by pushing the multicast functionalities to the edge of the networks, for example, application layer multicasts or end system multicasts using Narada[1], CAN[2], or NICE[3]. At the same time, there have been tremendous efforts in industry, in particular content delivery network (CDN) providers like Akamai, to strategically place a large number of streaming servers around the Internet. These enable end users to obtain streaming media from a nearby server, thus reducing the end-to-end delay and overall network congestion.

There have also been developments in the peer-to-peer (P2P) technologies starting in the 1990s. This new paradigm has changed the Internet in a fundamental way, with the development of many P2P applications such as Napster, BitTorrent, and Skype. P2P traffic now accounts for more than 60% of the traffic in

Received: 2008-03-13; revised: 2008-12-01

* Supported by the National Natural Science Foundation of China (No. 60673184), the National High-Tech Research and Development (973) Program of China (No. 2007AA01Z419), the National Basic Research (863) Program of China (No. 2008CB317101), and Tsinghua – ChinaCache CDN Program

** To whom correspondence should be addressed.
E-mail: hyin@csnet1.cs.tsinghua.edu.cn; Tel: 86-10-62782606

the Internet[4]. One of the key features of P2P systems is that each peer contributes resources including bandwidth, computing power, and storage space; thus, the total system capacity actually increases as more clients join the system. Another key advantage of P2P techniques is the robustness in case of failure since each peer does not rely upon any centralized server for content retrieval. In recent years, P2P-based live streaming technologies have gained more and more attention from the research community and industry since the development of CoolStreaming[5]. Several systems have attracted large numbers of viewers recently, such as PPLive[6] and SopCast[7].

However, before these group-oriented media streaming applications can be successfully deployed commercially, efficient access control mechanisms are needed to ensure that only the authorized users can access the original media.

In P2P streaming applications, since each peer directly transmits media to other peers, one access control method is to ensure that each participant can verify the other peers' legitimate identity for accessing the media, i.e., user authentication. Several user authentication schemes have been proposed for P2P systems without central authorities, such as the reputation-based authentication, digital fingerprint authentication, and identifier-based authentication. However, authentication methods without central authorities cannot (almost impossible) provide good security[8]. Symmetric key cipher-based authentication methods such as Kerberos[9], and public key cipher-based authentication methods such as X.509 (PKI)[10], are server-based authentication solutions. In Kerberos, all users must be registered with the Kerberos server and the server must share a secret key with each of the participants. The secure channel between each pair of peers is established on the unique ticket generated by the Kerberos server, which requires about $O(n^2)$ overhead where n is the total number of members, so the server easily becomes the system bottleneck. Therefore, the Kerberos scheme is not a good authentication solution for P2P applications.

The other important method of access control in group communication systems is through encryption and selective distribution of the session key (SK) used to encrypt the communication information, which is usually called key management or key distribution[11]. Key management schemes for group communication can be divided into centralized, decentralized, and distributed approaches[12]. The one-to-many characteristics of video multicast applications suggest a centralized approach. In a centralized key management approach, a single entity such as the key distribution center (KDC) controls the entire group of members by initialing and updating the session keys and distributing the re-keying messages. The centralized approaches can be further classified into hierarchy tree schemes (HTS) and centralized flat schemes (CFS). However, in these schemes, the system key must be changed once a user leaves or joins the group which leads to an extremely high overhead for the distribution center to maintain a key tree as the group grows very large. The distribution center will quickly become a bottleneck which affects the entire group.

This paper extends the secure and scalable media streaming architecture of TrustStream[13], to P2P streaming security using user authentication and key management schemes to provide security guarantees for P2P streaming over the Internet in a scheme-named "sStream". TrustStream needs a key management scheme similar to NICE designed as a tree-based application layer multicast protocol. The distributed hash table (DHT) technique-based key distribution scheme has been used for P2P streaming systems[14]. However, these papers did not present good discussions of the user authentication. There are several user authentication studies for different P2P applications[15] but few for P2P live media streaming.

In this scheme, before each authorized user joins the system, the user first obtains a unique public-key certificate from the management server. As the clients depart, the challenge is to provide efficient certificate revocation so that the certificates become invalid after the clients leave the system[16]. The lifetime of the issued certificate is based upon the media content instead of the host's system time in sStream, with the one-way hash chain employed to extend the certificate lifetime to reduce the computational overhead such as for the digital signature when the certificate is re-generated. Furthermore, a short certificate revocation list (CRL) is periodically distributed via an overlay network so that the communication cost is very low. This user authentication scheme is named "UAS". In addition, the system uses a key distribution overlay

network and a periodic global re-keying mechanism, which is highly scalable, efficient, and robust against frequent arrivals and departures of members.

1 User Authentication Scheme

1.1 Scheme overview

In the present scheme, the management server or the authorization server (AS) issues each new user a unique public-key certificate if it is an authorized user. The certificate signature is generated from the private key of the AS and the corresponding public key is published to all the communication group members to verify the certificate presented by any other participant. The certificate lifetime is based on the media content, i.e., it is valid during certain frames. Typically, users need new certificates from the AS when the certificate is invalid, which may result in high computational overhead for the re-signature process. The present scheme employs a one-way hash chain to extend the certificate's lifetime[17,18], which significantly reduces the computational overhead for the certificate update. Otherwise, the AS periodically delivers a CRL to the participants via a P2P overlay network. Each user receives the CRL by push or pull transfers with the media distribution. The CRL identifies revoked certificates for users who have departed but their certificate has not yet expired. Since the lifetime is limited, the CRL needs fewer certificates which reduces the communication overhead. The following sections describe UAS in detail, including the certificate generation, certificate update, certificate revocation, and certificate verification. The notation summary is given in Table 1.

Table 1 Notation summary

AS	Authorization server
SN	Super node
$\mathrm{PriK_{AS}}$, $\mathrm{PubK_{AS}}$	Private and corresponding public keys from the AS
n	Total number of users
i, j	User indices, $0 < i, j \leqslant n$
U_i, U_j	i-th and j-th users
PriK_i, PubK_i	Private and corresponding public keys of U_i
CT_i	Certificate of U_i
$H^m(x)$	$H^m(x)=H(H^{m-1}(x))$, $m>1$, $H()$ is a one-way hash function, such as SHA[19]

1.2 Certificate generation

Before any user U_i joins the P2P live streaming system, it generates a pair of keys PriK_i and the corresponding PubK_i based on a public key cipher, and then generates a random value R_i and calculates $H^m(R_i)$. Afterwards, user U_i sends a login request to the AS through a secure channel established with a secure protocol, such as SSL. The AS verifies whether U_i is an authorized user based on the personal information in the received login request. When authorized, the AS creates a unique certificate CT_i with the signature PriK_i for U_i and issues it. The major components of CT_i include:

$$\{\mathrm{ID}_i|T_\mathrm{s}|T_\mathrm{e}|T|\mathrm{IP}_i|\mathrm{PubK}_i|H^m(R_i)|H^m(\mathrm{RS}_i)|\mathrm{SigN}_i\}.$$

Pubk_i and $H^m(R_i)$ are received from the login message and ID_i is the identifier of user U_i. The AS assigns a unique integer to every authorized user as its identifier. The integer is incremented by one for each new user and the AS does not assign revoked IDs to new users. T_s and T_e denote the certificate lifetime. T_s is the current media frame delivered by the source server. The lifetime interval is $T_\mathrm{e}-T_\mathrm{s}$, which is usually determined statistically in real applications. T_e is the certificate update interval for one certificate. IP_i is the network address of U_i, which is fixed during a group communication session. RS_i is a private number for U_i only known by the provider servers (the AS and other super nodes) for certificate generation. $H^m(\mathrm{RS}_i)$ is the m-times hash value of RS_i, which is the key value for the certificate update. Finally, SigN_i is the digital signature which is non-repudiation evidence that authenticates the integrity as well as the origin of the certificate, such as using digital signature algorithms RSA[20].

1.3 Certificate update

As illustrated in Fig. 1, each user certificate is only valid during its lifetime between media frames T_s and T_e. User U_i can use the following certificate update processes instead of receiving a new certificate to extend the validity of its certificate for accessing subsequent media content:

(1) Before user U_i accesses frame between frames $<T_\mathrm{e}+(t-1)\times T,\ T_\mathrm{e}+t\times T>$, where $0<t<m$, U_i sends a certificate update request $\{\mathrm{ID}_i|t|H^{m-t}(R_i)\}$ to the AS.

(2) After receiving the update request message, the AS first verifies whether the request message is really

from U_i by checking whether
$H(H^{m-t}(R_i))=H^{m-(t-1)}(R_i)$ or $H^t(H^{m-t}(R_i))=H^m(R_i)$.

(3) When the request is original from U_i, the AS sends the response message for the certificate update to U_i. The response message is $\{\mathrm{ID}_i|H^{m-t}(\mathrm{RS}_i)\}$, where $H^{m-t}(\mathrm{RS}_i)$ is the update key for certificate CT_i.

(4) After obtaining the response message, user U_i can continue accessing the media from frame $T_e+(t-1)\times T$ to $T_e+t\times T$ by the identification in CT_i and the update key $H^{m-t}(\mathrm{RS}_i)$.

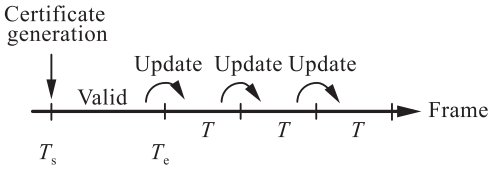


Fig. 1 Certificate update

1.4 Certificate revocation

As clients leave the system, the certificates should become invalid even though the certificate lifetime has not expired. The certificate revocation processes use a short CRL that is periodically generated by the AS and distributed to all the participants via an overlay network with pull or push transfers. The CRL distribution overlay is established on the media data transmission network or the P2P membership network.

The CRL is a bit string of size N, where N is the number of users that have subscribed to the system with unexpired certificates. Each certificate is mapped to a bit in the bit string in the CRL. The certificate's ID is the index of the bit in the string. ID is the number of ID's. A bit value of '1' indicates that a certificate is invalid and '0' indicates validity. When a user departs from the system, its corresponding bit in the string is set to '1' by the AS. T_L is the current media frame when the CRL is issued. SigN is the digital signature of the CRL used as non-repudiation evidence. Figure 2 presents the components of the CRL.

T_L	ID	N	Bit string	SigN

Fig. 2 Certificate revocation list

1.5 Certificate verification

The basic processes for one of the participants U_k to verify whether user U_i is authorized to access the media data in frame f includes the following processes:

(1) U_k first gets the certificate CT_i from U_i or any other member and checks the origin of CT_i with the verification function $V(\mathrm{CT}_i,\mathrm{SigN}_i)$, where SigN_i is the digital signature of CT_i.

(2) U_k then gets the IP address of U_i by receiving its messages and checking whether it is equal to the IP_i in CT_i. To receive the media information, a user cannot lie about its IP address; thus, this approach thwarts unauthorized nodes from using the certificate of an authorized user.

(3) U_i sends the message $\{M_i|E(M_i)\}$ to U_k, where M_i is a random number and $E(M_i)$ is the encrypted value of M_i encrypted using a certain public cipher algorithm with its private key PriK_i. After receiving this message, U_k uses the public key PubK_i in the CT_i to calculate $D(E(M_i))$ and checks whether $D(E(M_i))$ is equal to M_i.

(4) U_k sends the message $\{E(M_k)\}$ to U_i, where M_k is a random number generated by U_k and $E(M_k)$ is the encrypted value of M_k encrypted using a public cipher algorithm with U_i's public key PubK_i in the certificate CT_i. After receiving this message, only U_i can use its private key PriK_i to calculate $D(E(M_k))$ to get M_k, where M_k is equal to $D(E(M_k))$. The random number M_k is considered to be the symmetric secret key for establishing the secure channel from U_k to U_i.

(5) U_k verifies whether U_i's certificate CT_i is valid for accessing media frame f. First, if f is less than T_s, CT_i is invalid. Second, if f is between T_s and T_e, U_k checks its bit value in the newest CRL to ascertain the validity of CT_i. Third, when f is later than T_e and during the interval $<T_e+(t-1)\times T, T_e+t\times T>$, the validity of CT_i is determined by whether U_i owns the update key value of $H^{m-t}(\mathrm{RS}_i)$, because $H^t(V)$ is equal to $H^m(\mathrm{RS}_i)$ if and only if V is equal to $H^{m-t}(\mathrm{RS}_i)$.

Therefore, the participants can verify the validity of other participants by this process in a secure channel established for the session key distribution and encrypted media delivery.

2 Key Management Scheme

2.1 Scheme overview

To guarantee that only authorized users can access the original media even when eavesdropping occurs, the original content should be encrypted for delivery in the distributed system. The secure scheme must use encryption and selective distribution of the session

keys used to encrypt the content, i.e., key management. This section describes an efficient key management scheme based upon the user authentication scheme presented for the P2P live media streaming system in Section 1, which is referred to as "KMS".

In IP multicast communications, users have no responsibility to forward key materials to other group users. In contrast, for P2P communications, the users in the delivery tree also act as routers. As such, the KDC only has to deliver a new SK securely to a small number of users, which are its immediate neighbors who then forward the new SK securely to their own neighbors. In this way, a session key is propagated to all the online users in a hop-by-hop fashion. However, this scheme requires a secure channel between each pair of neighboring users. As mentioned in Section 1, the UAS can be adopted to establish a secure channel between peers.

2.2 User joins

When a user wants to join the P2P media streaming group, it should first contact the KDC to be authenticated. Then, it can find trusted neighbors in the group and get future re-keying messages from them. The detailed procedures for user joining are:

(1) Before user i joins the group, it sends a login request to the KDC via a secure channel. After being authenticated, user i becomes a new legitimate member and get its certificate from KDC.

(2) User i contacts other users in the group as its neighbors through the member management protocol of the P2P streaming application, such as the Gossip protocol or the DHT technique. They then exchange their certificates to verify each other's legal identity.

(3) Once a neighbor user j verifies that user i is a legal member of the current group session, user j adds user i to its local view of its legal neighbors. The HTS mechanism is used to maintain the local view for later key distribution. Figure 3 shows the logic key tree of user j before user i becomes its trusted neighbor. In Fig. 3, KEK is the key of the secure channel between user j and its trusted neighbors.

(4) User j sends the key materials $\{\mathrm{PubK}_i(K_8), K_8(K'_{78}), K'_{78}(K'_{58}), K'_{58}(\mathrm{KEK}')\}$ to user i, where $A(B)$ is the value of B encrypted by A. User j also sends other key materials to its old neighbors after user i joins as shown in Fig. 4, e.g., user j sends $\{K_7(K'_{78}), K'_{78}(K'_{58}), K'_{58}(\mathrm{KEK}')\}$ to U_7.

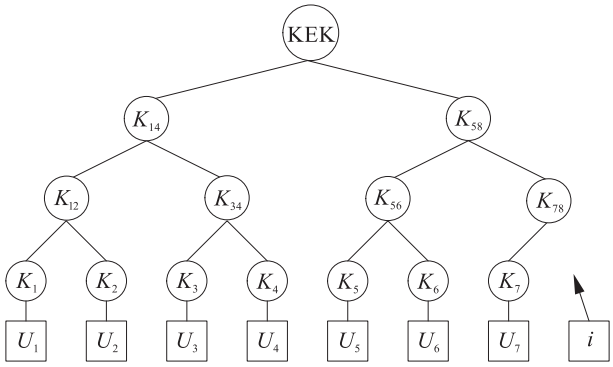


Fig. 3 Logic key tree of *j* before *i* joins

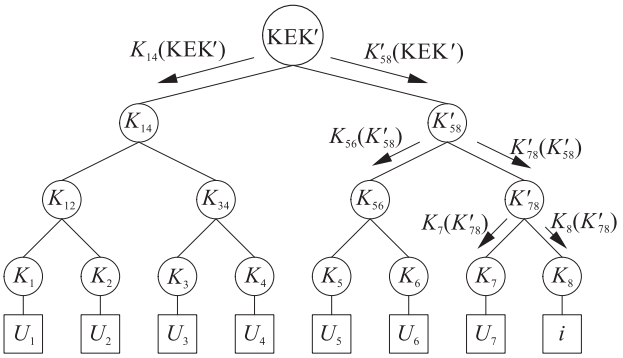


Fig. 4 Logic key tree of *j* after *i* joins

(5) After receiving the key materials respectively, each neighbor of user j receives the newest KEK′ shared by user j, including the new user i.

(6) User i establishes the trusted memberships with its new neighbors in the same way.

2.3 User leaves

When a user leaves the group, it first notifies all its neighbors and then contacts the KDC to logout.

(1) User i first sends a LEAVE message to all its neighbors.

(2) After receiving the LEAVE message from user i, neighbor j changes its local view of its legal neighbors. Figures 5 and 6 show the logic key trees of j as i is leaving.

(3) As illustrated in Fig. 6, user j changes some of the key values in its logic key tree and then sends some of the key materials to its neighbors. For example, it sends $\{K_4(K'_{34}), K'_{34}(K'_{14}), K'_{14}(KEK')\}$ to user U_4. Thus, its neighbors receive the newest KEK′ from j.

2.4 Key distribution overlay

Each user U_k shared a key KEK with its valid neighbor users through the secure channel. KEK is changed

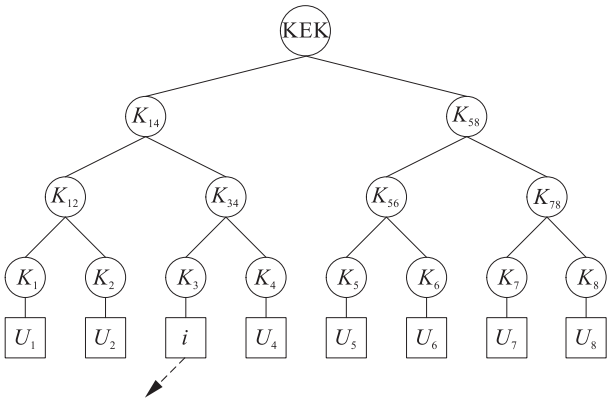


Fig. 5 Logic key tree of *j* before *i* leaves

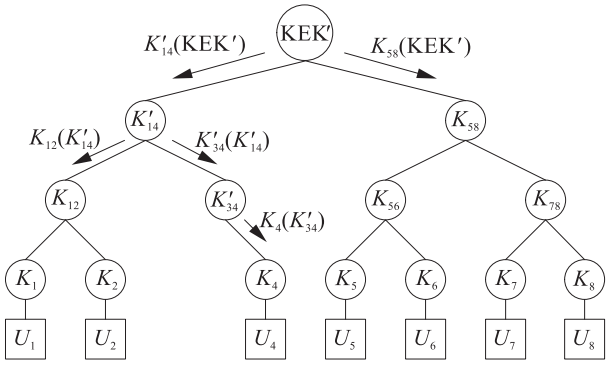
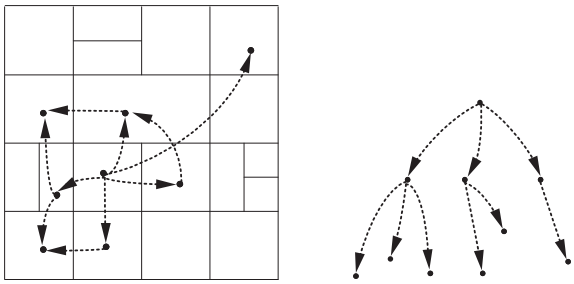


Fig. 6 Logic key tree of *j* after *i* leaves

once its neighbors change. U_k uses the KEK to encrypt the session key with the symmetric cipher and sends the encrypted re-keying to its neighbors. Only valid neighbors who share the KEK can obtain the correct SK. Figure 7 shows the two overlay network topologies for the key distribution.



(a) Mesh (b) Tree

Fig. 7 Overlay network for key distribution

As with TrustStream, the re-keying message can also be distributed in a media-dependent manner, as shown in Fig. 8. The key distribution overlay is attached to the data transmission network to make the transfer media-dependent. The various procedures for peers to receive the different media blocks for the session key distribution are shown in Fig. 9.

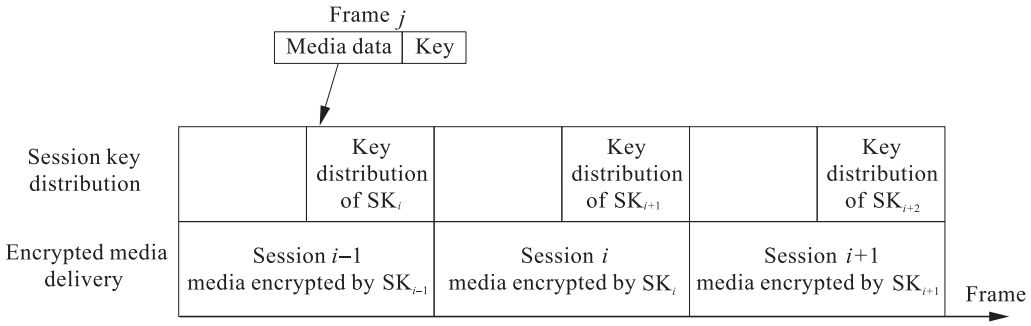


Fig. 8 Media-dependent key distribution

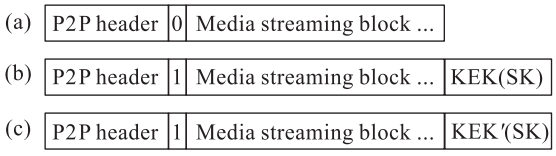


Fig. 9 Data blocks in P2P streaming system

(1) When user *i* receives a media block as shown in Fig. 9a, it checks whether the flag bit is zero and only transmits this block to its children neighbors that need the media block.

(2) When user *i* receives a media block as shown in Fig. 9b, it first calculates the new session key since it owns the KEK shared with its parent neighbor sending this media block. Then, it use its own shared KEK′ to encrypt the new session key. User *I* then replaces the re-keying materials with KEK′(SK) as shown in Fig. 9c and transmits this block to its children neighbors.

3 Security and Performance Analysis

3.1 Security analysis

These schemes provide confidentiality which is the main security requirement. First, the digital signature algorithm guarantees that the certificate is only generated by the AS based on its private key. Second, other

users cannot impersonate the certificate of U_i without knowing its private key. Third, the IP detection prevents imitations by other certificates even if the private key becomes known. Furthermore, since the RS_i is only known by the AS and the other servers and the hash function $H()$ is one-directional, user U_i cannot calculate $H^{m-t}(RS_i)$ with its information without the certificate update response from the server. Thus, only authorized users can get the correct certificate to become legal neighbors of other legal peers. The HTS mechanism for each member provides forward and backward secrecy which ensures that only legal users can share the KEK with its legal neighbors to continuously obtain the session key to access the media. Thus, all the users but only the users permitted by the service provider can access the media data.

3.2 Performance analysis

The message exchanges between the users and servers during the certificate update processes are not through a secure channel, which reduces the computational overhead through use of the hash function instead of a digital signature. Since the CRL in the UAS only maintains valid certificates, the CRL is shorter than the CRL in the previous scheme. Furthermore, the CRL and the re-key messages are periodically deliveried by hopping between users, each user only maintains a limited number of neighbors, so the communication costs at each user and at the server are very low, almost $O(k)$ which is independent of the group size and the number of clients joining or leaving, where k is the neighbor number of each peer. Therefore, the scheme is scalable for large-scale P2P applications. In addition, since each user obtains the key materials from multiple neighbors, the key distribution is robust when a user momentarily departs and packet transmission is lost. Finally, the media-dependent method ensures that each user obtains the appropriate session key for continuous reception as the re-keying messages are embedded into the data block at the beginning of the sessions. To improve the efficiency, the key materials could be embedded into multiple blocks during each session.

3.3 Simulations

P2P live media streaming was simulated with populations ranging from 1000 to 1 000 000 with the server randomly sending the CRL and re-keying to only 10 users. The simulations show the times for the message distribution for various mesh or tree arrangements. For the tree-based schemes, the time complexity of communication is $O(\log_2 N)$ and is $O(\log_4 N)$ for mesh scheme as shown in Fig. 10.

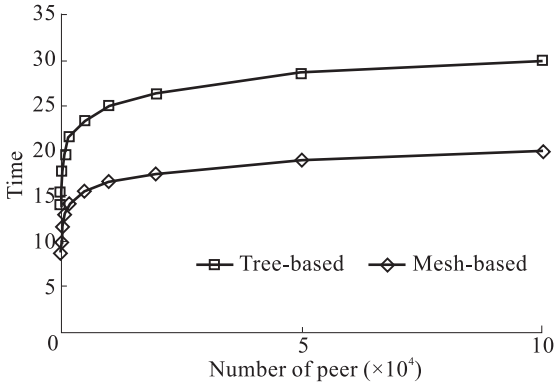


Fig. 10 Time of key distribution

4 Conclusions

This paper presents a user authentication and key management scheme for single P2P live streaming systems. Future work will analyze multiple stream systems. The current system uses the public-key certificate technique, one-way hash chain, and CRL delivery for the user authentication and builds a highly scalable, efficient key distribution overlay network for periodical re-keying, which is robust against dynamical member events. The system gives security guarantees for P2P live media streaming systems. Future work will also combine these schemes with existing P2P streaming protocols to examine its performance in real environments.

References

[1] Chu Y H, Rao S G, Seshan S, Zhang H. A case for end system multicast. In: Proceedings of the ACM SIGMETRICS. Santa Clara, USA, 2000.

[2] Ratnasamy S, Handley M, Karp R, Shenker S. Application-level multicast using content-addressable networks. In: Proceedings of the NGC. London, UK, 2001.

[3] Banerjee S, Bhattacharjee B, Kommareddy C. Scalable application layer multicast. In: Proceedings of the ACM SIGCOMM. Pittsburgh, USA, 2002.

[4] Li Bo, Yin Hao. The peer-to-peer live video streaming in the Internet: Issues, existing approaches and challenges. *IEEE Communications Magazine*, 2007, **45**(6): 94-99.

[5] Zhang X Y, Liu J C, Li B, Yum T S P. CoolStreaming/DONet: A data-driven overlay network for peer-to-peer live media streaming. In: Proceedings of the IEEE INFOCOM. Miami, USA, 2005.

[6] PPLive, http://www.pplive.com/, 2008.

[7] SopCast, http://www.sopcast.org/, 2008.

[8] Aslund J. Authentication in peer-to-peer systems. http://www.ep.liu.se/exjobb/isy/2002/3153/.

[9] Kohl J. The Kerberos network authentication service (V5). RFC 1510, Sep 1993.

[10] Adams C. Internet X.509 public key infrastructure – Certificate management protocols. RFC 2510, Mar 1999.

[11] Trappe W, Song J, Liu K J R. Key management and distribution for secure multimedia multicast. *IEEE Transactions on Multimedia*, 2003, **5**(4): 544-557.

[12] Rafaeli S, Hutchison D. A survey of key management for secure group communication. *ACM Computing Surveys*, 2003, **35**(3): 309-329.

[13] Yin Hao, Lin Chuang, Qiu Feng, Liu Xuening, Wu Dapeng. TrustStream: A novel secure and scalable media streaming architecture. In: Proceedings of the ACM MULTIMEDIA. Singapore, 2005.

[14] Qiu Feng, Lin Chuang, Yin Hao. EKM: An efficient key management scheme for large-scale peer-to-peer media streaming. In: Proceedings of the PCM. Hangzhou, China, 2006.

[15] Zhu Sencun, Yao Chao, Liu Donggang, Setia S, Jajodia S. Efficient security mechanisms for overlay multicast-based content distribution. In: Proceedings of the ACNS. NY, USA, 2005.

[16] Naor M, Nissim K. Certificate revocation and certificate update. In: Proceedings of the 7th USENIX security symposium. San Antonio, USA, 1998.

[17] Micali S. Efficient certificate revocation. Technical Report MIT/LCS/TM-542b. Massachusetts Institute of Technology, USA, 1996.

[18] Aiello W, Lodha S, Ostrovsky R. Fast digital identity revocation. In: Hugo Krawczyk, ed. Advances in Cryptology CRYPTO'98, International Association for Cryptologic Research. Berlin, Germany: Springer-Verlag, 1998.

[19] Eastlake D, Jones P. US Secure hash algorithm 1 (SHA1). RFC 3174, Sep 2001.

[20] Kaliski B. PKCS #1: RSA encryption version 1.5. RFC 2313, 1998.

Cooperation Agreement Between Tsinghua University and Chung-Ang University Renewed

Tsinghua University and Chung-Ang University renewed their Agreement of Cooperation and signed a Memorandum of Agreement for student exchange on November 23, 2008 when Chung-Ang University's President Park Bum Hoon and Vice President Hong Won Pyo visited the Tsinghua campus.

According to the Agreements, Tsinghua University and Chung-Ang University will further expand their academic cooperation in joint research, faculty and student exchanges, joint organization of seminars and academic meetings, exchange of materials in education, research, publications, and academic information.

(From http://news.tsinghua.edu.cn, 2008-12-24)