산업기술연구(강원대학교 산업기술연구소 논문집), 제24권 B호, 2004.
Journal of Industrial Technology, Kangwon Natl. Univ., Koea, No. 24 B, 2004.

보안성과 유연성을 갖춘 데이터 공유 방안

A Data Sharing Scheme with Security and Flexibility

이 구 연* 김 화 종* 정 충 교*
Lee, Goo-Yeon Kim, Hwa-Jong Jeong, Choong-Kyo

Abstract

We propose and analyse a flexible secure file sharing scheme which can be used for data sharing among members in P2P environment. When a member wants to share data, notification messages are sent to the members with whom the member wants to share data. Each notification message includes one-time password encrypted with the receiver's public key. A member who received the notification message can download the data by using the one-time password. The proposed scheme provides selective sharing, download confirmation and efficient memory management. In terms of security, the proposed scheme supports authentication, entity privacy, replay attack protection and disguise prevention.

Keywords : *P2P security, data sharing, file sharing one-time password, symmetry key, public key*

1. Introduction

Peer-to-Peer (P2P) system has been widely used for data sharing among open users (peers). A pure P2P system does not use a centralized sever, while a hybrid P2P system requires an index server to locate the data. Napster is a typical example of the hybrid type. Gnutella and Freenet are well known systems for the pure types [1].

Researches on the P2P system include routing algorithm to find wanted data [2,3], scalability problem [2, 3, 4, 5, 6], trust of users [4] and data security [4, 5, 6, 7]. However, data sharing among specific members has not attracted much attention in the P2P researches because a P2P system is basically used for data sharing among open users. However some data need to be shared with specific members (a group of friends or some members of a community) whereas others may be shared among open users. For example, one may want to share a music or video file among intimate friends. A plain P2P system does not support authentication or privacy for such an application.

For secure data sharing among specific users, additional functionality such as authentication or encryption should be implemented over the plain P2P system. Requirements of a secure data sharing system are:

- Authentication of proper members

* 강원대학교 전기전자정보통신공학부

- Confidentiality of shared data (entity privacy)
- Confirmation of data download
- Prevention of retransmission (replay) attack
- Disguise prevention
- Avoidance of memory waste due to long term archiving
- Support for sharing of large data such as music or video files

Historically, many file sharing algorithms have been studied in the field of shared storage systems. However the file sharing algorithms can not be used directly in the P2P system because the requirements of the two systems differ. In file sharing algorithms, the provider of shared data needs receiver authentication and safe access control method, while the receiver needs authenticity of the provider and the data. In [7], a distributed polling algorithm was introduced which can be used to investigate the authenticity of the shared data before download to avoid download from a malicious node. However this algorithm does not provide access control to limit the access right only to the specific qualified users.

Similarly, a data encryption and key management algorithm in a distributed storage system was introduced in [8]. However, this scheme does not include user authentication function. In [9], a mechanism which enables only eligible users to access encrypted shared data was introduced for secure distributed storage systems.

In this paper, we propose and analyse a flexible secure file sharing scheme which can be used for data sharing among specific members in a P2P environment. In the proposed scheme, the sender notifies by email or messenger to the members that shared data has been posted. This notification message includes information needed to download the data, e.g., URL and one-time password to access the data. For authentication of each member, public key cryptography is used. The members may share public keys each other in advance or a key server may be used for key generation and management.

2. Data Sharing Scheme with Security and Flexibility

The proposed scheme can be implemented atop a plain P2P system. For secure data sharing, each member should know IDs and public keys of other members. When a member wants to share a data with some other members, he (she) generates a one-time password for each member and put this one-time password into the notification message.

Each member has a Peer_List which contains IDs and public keys of the members whom he (she) wants to communicate with. Fig. 1 shows the sequence of data generation, posting, notification, and download. Fig. 2 shows the Peer_List and Password_Table. The Password_Table is generated for each shared data. The Password_Table contains members' IDs, one-time passwords (or states) and life time of the data.

Let us assume that member A has data to share with others. Each step in Fig. 1 is explained in the following:

① Member A chooses a symmetric key, Ks which is used to encrypt the shared data.
② Member A selects members to share the data with in the Peer_List and makes a Password_Table which includes one-time passwords of the members. Member A also sets a life time, Tè, of the shared data.
③ The shared data (encrypted with Ks) and Password_Table are moved to the Shared_Folder. The Password_Table should be hidden from other members.
④ Member A notifies the posting of the data to the selected members through email or messenger. The notification message, for example to the member B, is as follows:

A→B : {A, B, {A, B, shared-data-URL, Ks, Tè, Password(B)}Private_Key(A)}Public_Key(B)

where {W}K denotes that message W is encrypted by a key K. shared-data-URL represents the location of the data to be shared, and Ks is the symmetric encryption key. After life time Te, the shared data will be removed from the Shared_Folder to avoid memory waste. Password(B) is the one-time password which will be used by member B to access (i.e., to

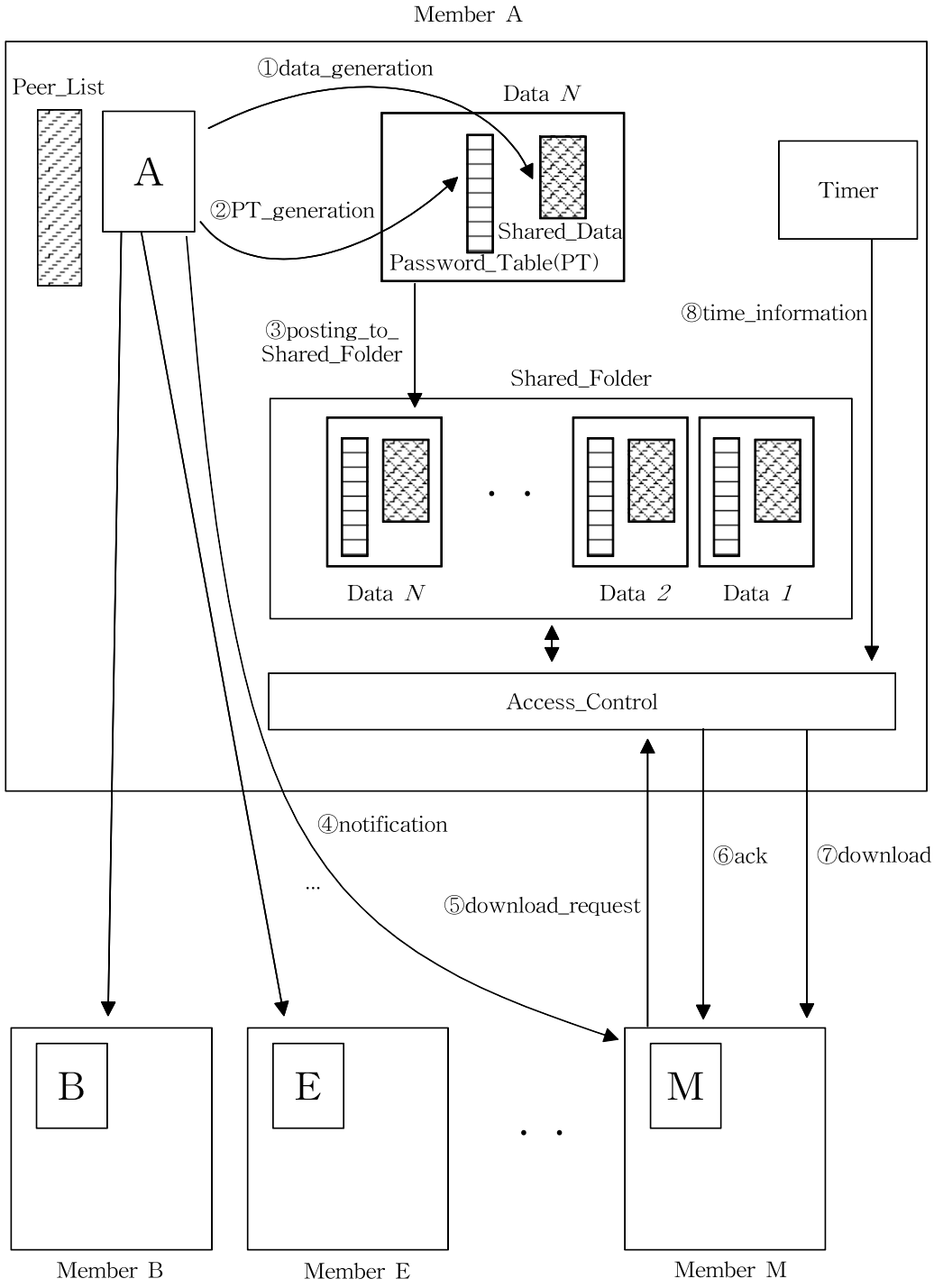

Fig. 1 The sequence of data generation, posting, notification and download

download) the data. The notification message is encrypted first with A's private key, then with B's public key. This makes only B be able to decrypt the notification message and guarantees that A has sent the message. The notification message is sent to each member in the Password_Table. It is noted that notification message (which is small in size) is encrypted by the public key of each member, whereas the shared data (which is normally large) is encrypted by the symmetric key for efficient processing.

⑤ Each notified member sends a download request to A. For the member M, the download request message is :

$M \rightarrow A$: {M, A, {M, A, shared-data-URL, Password(M)}Private_Key(M)}Public_Key(A)

The download request message contains one-time password Password(M) for authentication. Part of the download request message is encrypted by member M's private key for digital signature, and the whole message is encrypted by member A's public key for data secrecy.

⑥ Member A sends back ack message to member M if Password(M) is in the Password_Table. However, member A sends back failure message to M if the Password_Table does not contain Password(M).

⑦ After receiving ack message, member M can download the data and decrypt it with Ks. After the data has been successfully downloaded to member M, member A sets the state of member M in the Password_Table to DOWNLOADED. This prevents another access to the data with the same Password(M). In this way, Password(M) is used as an one-time password. When all the states of the Password_Table are set to DOWNLOADED, the Access_Control module deletes the shared data and moves the Password_Table to the Archive_Folder.

⑧ Access_Control module monitors the time elapsed from the generation of each data and compares it with the life time Te set for the data. When Te has expired, the data is removed, and the Password_Table is moved to Archive_Folder. The data is no longer accessible to any members, whereas the sender knows who has downloaded the data by checking the Password_Table in Archive_Folder.

ID	Public Key
B	Public_Key(B)
C	Public_Key(C)
D	Public_Key(D)
E	Public_Key(E)
...	...
M	Public_Key(M)
...	...

(a) Peer_List of Member A

Lifetime : T_e	
ID	Password/State
B	Password(B)
E	Password(E)
...	...
M	Password(M)
...	

(b) Password_Table of Data N

Fig. 2 Peer_List of Member A and Password_Table of Data N in Fig. 1.

3. Analysis of the Proposed Scheme

In this section, we analyse the functions of the proposed scheme and compare its characteristics, especially in terms of secure data sharing, with a plain P2P system, data sharing via e-mail attachment, multicast using a group key and shared storage systems. And we explore the security aspects of the proposed scheme.

3.1 Features of the Scheme

Selective Sharing : In the plain P2P system, a peer shares data with all other peers and there is no selective sharing function. The proposed scheme provides selective sharing functionality. Each individual data can be shared by different group of members. This selective sharing function can be considered to be an advanced function of the plain P2P system.

Download Confirmation : In the proposed scheme, the sender can confirm that the data has been downloaded by a member. This can be done by checking the state field in the Password_Table in Shared_Folder or Archive_Folder.

Efficient Memory Management: If the shared data reside forever in the Shared_Folder, data that are no longer to be downloaded will use up all the memory especially if the individual data tend to be very large as in the case of multimedia files. For efficient memory management, the data are removed when all the members have downloaded the data or when the life time has expired.

Integration with existing P2P systems: The proposed scheme can be easily implemented over a plain P2P system. To implement the proposed scheme, the Access_Control module and the Peer_List management software need to be installed over existing pure or hybrid P2P systems.

3.2 Comparison with other schemes

Comparison with multicast using a group key : Multicast can be used for data sharing among a group of members, in which a group key is used for the security [11]. However multicast is different from a P2P system where data retrieval occurs when a peer wants to access the data. In a multicast system, all members of a multicast group receive shared data all the same time. Furthermore the proposed scheme needs not to handle group join, leave or updating group key, because the proposed scheme uses pubic keys of members.

Comparison with Email: Email attachment can be used conveniently for data distribution among a group of members. However even a member who does not want the data can not but receive the data, which will waste the memory and annoy the receiver. This makes a serious shortcoming especially when the data are very large. Furthermore some email systems do not transfer large files e.g., more than 10 MBytes.

Comparison with Shared Storage System: A shared storage system is the traditional means for data sharing among specific members. However, a shared storage system requires membership enrollment and does not provide selective sharing, download confirmation or life time of shared data.

3.3 Security Analysis

Authentication : The proposed scheme provides authentication of the receiver. Only a proper receiver can use the one-time password because the one-time password is encrypted by the receiver's public key.

Entity Privacy : Shared data is encrypted by a symmetric key which is sent to specific members encapsulated in the notification message. Others who do not have the symmetric key cannot decrypt the data.

Prevention of Retransmission : When a member has downloaded the data with proper one-time password the member's state in the Password_Table is changed to DOWNLOADED. Second access to the data with the same one-time password will be rejected by Access_Control module. This scheme prevents the replay attack.

Disguise Prevention : As shown in steps ④ and ⑤ of Fig.1, the notification message is encrypted with the sender's private key. A receiver knows that this message is from the proper sender if the message is decrypted with the sender's public key.

4. Conclusion

In this paper we proposed a secure data sharing algorithm for extended P2P systems. For member authentication, the public key cryptography is used. When a member wants to

share data with others, the member sends notification messages via e-mail or messenger to the members with whom he (she) wants to share data. Members who received the notification message can download the data by using one-time password and URL of the data, which are extracted from the notification message.

Unlike conventional P2P, e-mail, group multicast using a group key or shared storage systems, the proposed scheme provides selective sharing, download confirmation and efficient memory management. In terms of security, the proposed scheme handles authentication, entity privacy, replay attack protection and disguise prevention. The proposed scheme can be implemented over a plain P2P, thereby can be used to extend a plain P2P to satisfy more complicated data sharing applications.

References

[1] A. Oram, Peer-To-Peer, O'reilly, 2001.
[2] S. Ratnassmy, P. Francis, M. Handley and R. Karp, "A scalable Content-Addressable Network", *Proceeding of ACM SIGCOMM'01*, San Diego, USA, 2001
[3] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek and H. Balakrishnan, "Chord A scalable peer-to-peer lookup service for Internet applications", *Proceeding of ACM SIGCOMM'01*, San Diego, USA, 2001
[4] M. Gastro, P. Druschel, A. Ganesh, A Rowstron and D. S. Wallach, "Secure routing for structure peer-to-peer overlay networks", *Proceeding of OSDI 2002*, Boston, USA, 2002
[5] E. Sit and R. Morris, "Security considerations for peer-to-peer distributed hash tables.", *Proceedings of the 1st International workshop on peer-to-peer Systems(IPTPS'02)*, Cambridge, USA, 2002.
[6] A. Rowstron and P. Druschel "Pastry : Scalable, decentralized object location and routing for large-scale peer-to-peer systems", *Proceeding of the 18th IFIP/ACM international Conference on Distributed Systems Platforms. Heidelberg*, Germany, Nov. 2001.
[7] E. Damiani, D. C. Vimercati and S. Paraboschi, "A Reputation-Based Approach for Choosing Reliable Resources in Peer-to-peer Networks", *Proceedings of ACM CCS'02*, Washington D.C, USA, Nov. 2002.
[8] Y. Kim, F. Maino, M. Narasimha and G. Tsudik, "Secure Group Services for Storage Area Networks.", SISW 2002, Dec. 2002.
[9] E. L Miller, W. E. Freeman, D. Long and B. Reed, "Strong Security for Network-Attached Storage", *Proceedings of the Fast 2002 Conference on File and Storage Technologies by USENIX*, Monterey, USA. Jan. 2002.
[10] L. Zhou and Z. Haas, "Securing Ad Hoc Networks", *IEEE Network Magazine*, Vol 13. No. 6, Nov. 1999.
[11] S. M. Iolus, "A Framework for Scalable Secure Multicasting", *Proceedings of the ACM SIGCOMM '97*, September 1997.