情報保護學會論文誌
第15卷 第4號, 2005. 8

보안성과 유연성을 갖춘 Peer-to-Peer 데이터 공유 기법의 설계 및 구현*

이 구 연,[1†‡] 이 용,[2] 김 화 종,[1] 정 충 교,[1] 이 동 은[1]
[1]강원대학교, [2]코넬대학교

Design and Implementation of a Peer-to-Peer Data Sharing Scheme for Closed User Group with Security and Flexibility*

Goo-Yeon Lee,[1†‡] Yong Lee,[2] Hwa-Jong Kim,[1]
Choong-Kyo Jeong[1] and Dong-Eun Lee[1]
[1]Kangwon National University, [2]Cornell University

ABSTRACT

We propose and implement a flexible secure peer-to-peer(P2P) file sharing scheme which can be used for data sharing among closed user group (CUG) members. 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 storage management. In terms of security, the proposed scheme supports authentication, entity privacy, replay attack protection and disguise prevention. We also implement the proposed system and find that the system is very useful among P2P service of closed user groups.

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

Ⅰ. 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[2-6], trust of users[4] and data security[4-7]. However, data sharing among specific closed user group(CUG) 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

접수일 : 2005년 3월 17일 ; 채택일 : 2005년 8월 11일
* This work was supported by the Brain Korea 21 project of Kangwon National University in 2005 and was supported by Kangwon Institute of Telecommunications and Information(KITI).
† 주저자, ‡ 교신저자, leegyeon@kangwon.ac.kr

want to share a music or video file among intimate friends. A plain P2P system does not support authentication or privacy for such a CUG application.

For secure data sharing among specific CUG 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
- Confidentiality of shared data(entity privacy)
- Confirmation of data download
- Prevention of retransmission(replay) attack
- Disguise prevention
- Avoidance of storage 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 are different each other. 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 in order to avoid download from a malicious node. However this algorithm does not provide access control to limit the access 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. 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 implement a flexible secure peer-to-peer file sharing scheme which can be used for data sharing among specific CUG members. In the proposed scheme, the sender notifies by email or messenger to the members that a 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. From the implementation, we see that the proposed scheme is very useful among a closed user group for sharing valuable data or paid contents.

Ⅱ. Data Sharing Scheme with Security and Flexibility

At first, we will describe some definitions and notations of the data sharing scheme in this section.

□ **Peer_List** : The list of members to share a data by a specific member. When the specific member shares a data, selected members from its Peer_List (i.e. a subset of the Peer List) should be made up before sharing the data. Each member has its own Peer_List. Therefore, deletion, addition or change of members in the Peer_List is performed for each member independently. A new member may ask to others to include itself in their Peer_Lists via sending its ID and public key through an e-mail or messenger.

□ **Password_Table** : The table which includes selected members from Peer_list for a specific data and one time password for each selected member. The Password Table also includes the download information for each selected Peer_List member.

□ **Password(B)** : The one time password for member B to access a specific data. One time password is generated for each member when the specific data is created and destroyed after each member downloaded the data.

□ **Shared_Folder** : The folder where shared data is stored, which is accessible from outside the system.

□ **Access_Control** : The control module in a system which manages the access from other members to its Shared Folder.

□ **Archive_Folder** : A data is moved from the Shared_Folder to Archive_Folder when the data is shared by all the selected Peer_List members or life time of the data is expired.

□ $\mathbf{K_s}$: The symmetric key used to encrypt a shared data. Different $\mathbf{K_s}$ is generated for each shared data.

□ $\mathbf{T_e}$: The life time of a shared data. After $\mathbf{T_e}$, the data is moved from the Shared_Folder to Archive_Folder.

□ $\mathbf{\{W\}_K}$: denotes that message W is encrypted by the key K.

In the proposed scheme, for secure data sharing, each member should know IDs and public keys of other CUG 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 gen-

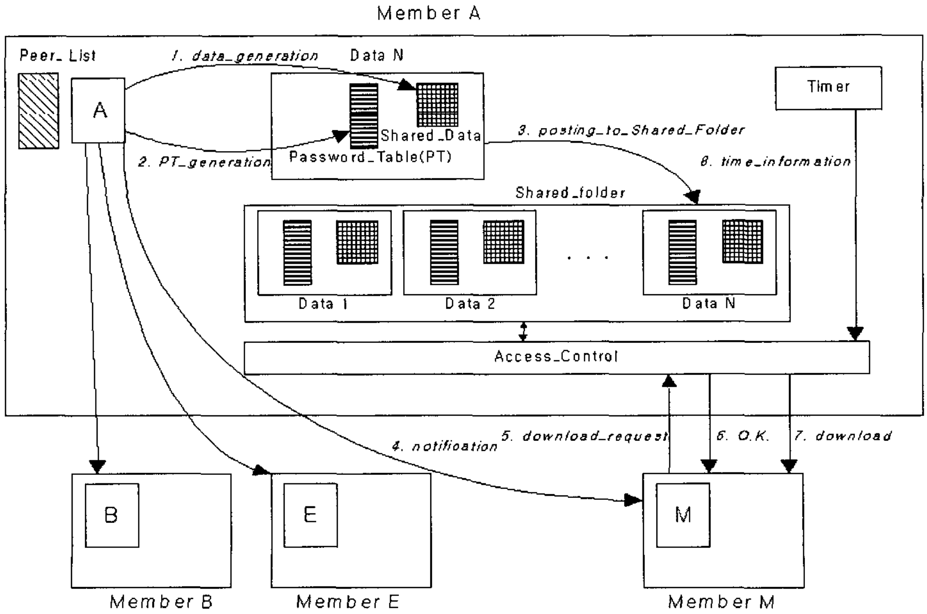


Fig. 1. The sequence of data generation, posting, notification and download

erated 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, K_s which is used to encrypt the shared data.

② Member A selects CUG 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, T_e, of the shared data.

③ The shared data (encrypted with K_s) 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:

$$A \rightarrow B : \{A,\ B,\ \{A,\ B,\ \text{shared-data-URL},\ K_s,\ T_e,\ \text{Password}(B)\}_{\text{Private\_Key}(A)}\}_{\text{Public\_Key}(B)}$$

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 K_s is the symmetric encryption key. After life time T_e, the shared data will be removed from the Shared_ Folder to avoid storage waste. Password(B) is the one-time password which will be used by member B to access(i.e., to 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.

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

Peer_List of Member A

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

Password_Table of Data N

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

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

$$M \rightarrow A : \{M,\ A,\ \{M,\ A,\ \text{shared-data-URL},\ \text{Password}(M)\}_{\text{Private\_Key}(M)}\}_{\text{Public\_Key}(A)}$$

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 O.K. 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 O.K. message, member M can download the data and decrypt it with K_s. 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). If the download is not completed due to various reasons including unstable network conditions, the state of member M in the Password_Table stays unchanged for later access. 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. It is noted that Password(M) is also used to show whether the member M downloaded the data or not, in addition to authenticate the proper user.

⑧ Access_Control module monitors the time elapsed from the generation of each data and compares it with the life time T_e set for the data. When T_e 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.

III. 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 Storage Management*** : If the shared data reside forever in the Shared_Folder, data that are no longer to be downloaded will use up all the storage especially if the individual data tend to be very large as in the case of multimedia files. For efficient storage 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. 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 storage 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.

Ⅳ. Implementation and Discussions

4.1 Test bed

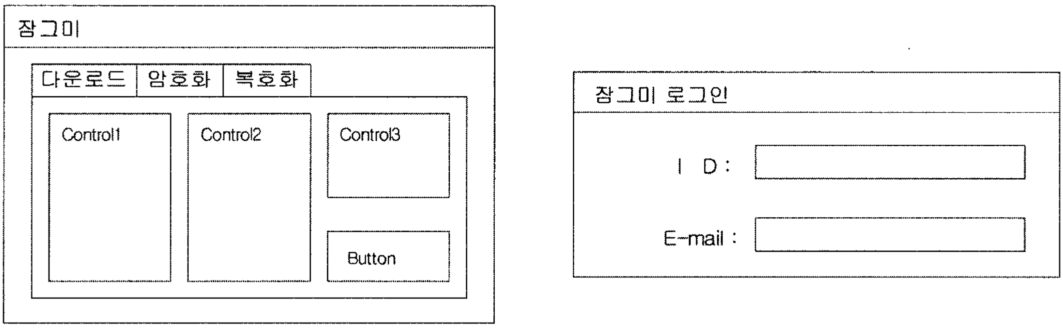
The prototype implementation was developed in the following environments.

Peer Desktop OS	Windows XP
Development Tool	Microsoft visual studio .NET
Development Language	C++
Server	Linux (Hancom Linux 3.1 pro)

In the basic operation of the proposed scheme, any server is not needed. However, a server is introduced for this prototype implementation to send notification emails. In practice, when the proposed scheme is exploited in the market, the SMTP function of the server can be replaced with the other existing email system.

4.2 Menu Configuration

Fig. 3 shows the block diagram of menu



(a) Main dialog (b) Login dialog

Fig 3. (a) Main dialog (b) Login dialog

configuration and Fig. 4 shows the menu screen shot of the implemented system.

□ ***Main dialog*** : Main dialog is composed of 3 sheets: download property sheet, encryption property sheet and decryption sheet. Each sheet performs the following functions.

- Download property sheet : download related functions are performed.

Control 1	It displays lists of users who are sharing data with this user.
Control 2	It displays the shared data of the other user selected at Control1
Button	Download start button.

- Encryption property sheet : encryption related functions are performed.

Control 1	It shows tree-structured files for this user to encrypt.
Control 2	It shows user list for this user to share the selected data with.
Control 3	It displays calendar for this user to select the expired time for sharing the selected data from.
Button 1	Encryption start button.
Button 2	It is used to add another user to share data with.

- Decryption property sheet : decryption related functions are performed.

Control 1	It shows encrypted files for this user to download.
Control 2	It shows tree-structured directories for this user to select the place for storing the decrypted data from.
Button	File decription start button.

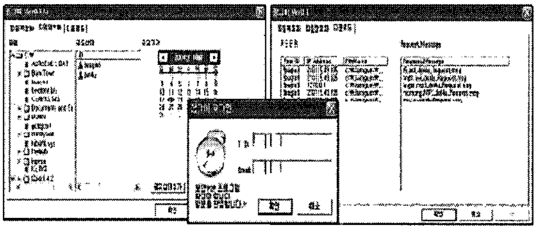


Fig 4. User interface

□ ***Login dialog*** : When the user runs the program first, the system asks him/her to input the ID and e-mail address to use thereafter. As the next step, the system generates a public key/private key pair for the user and the user's profile.

4.3 Implementation

The login window simply requires user ID and email address, and notifies the user that the login information will be stored in the system for later use. The next login process will automatically use the stored information. E-mail is used to get notified of the shared message or additional user information, which allows off-line use can access the system. The message and request messages are encrypted with RSA, which requires encryption with private key and decryption with public key with authentication. Data part is encrypted with 56bit key (symmetric) DES algorithm. We selected a short key size algorithm for fast processing. For more secure system, however, we can use long encryption keys. The user interface is shown in Fig. 4. The sharing duration of data is given by date rather than number of days for convenience. We did not restrict the number of Peer_List members, and allowed the number to be increased via "join" process. The size of a shared data is not also restricted, and we test the system by using a 800Mbyte video file.

4.4 Discussion

We implemented a prototype system to test the proposed scheme. We used Windows and tested text, audio, and video data. Data encryption and decryption are performed at the user's PC so there was no transmission delay. Unlike the traditional P2P system, the proposed system required authentication in the login process. However the encryption and decryption time was short enough because it uses a simple text for authentication. The system uses sharing folder scheme in order to share data, which may require much disc space for many sharing data. In order to alleviate this problem, the system removes old files by setting sharing duration. The proposed system can be used among a closed user group for sharing valuable data or paid contents. The system will protect authorized data from (traditional) unlimited P2P file sharing.

V. Conclusion

In this paper, we proposed and implemented a peer-to-peer secure data sharing scheme for CUG members. 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 storage management. In terms of security, the proposed scheme handles authentication, entity privacy, replay attack protection and disguise prevention.

We implemented the proposed scheme and see that it can be very conveniently used among a closed user group for sharing valuable data or paid contents and it will protect authorized data from (traditional) unlimited P2P file sharing. 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] Andy 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.

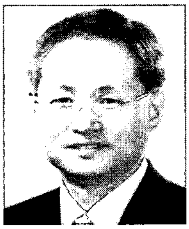
[8] Y. Kim, F. Maino, M. Narasimha and G. Tsudik, "Secure Group Services for Storage Area Networks.," SISW 2002, Dec. 2002.

[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.

〈著者紹介〉

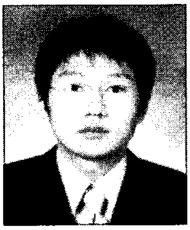
이 구 연 (Goo-Yeon Lee) 정회원
1986년 2월: 서울대학교 전자공학과(학사)
1988년 2월: KAIST 전기및전자공학과(석사)
1993년 2월: KAIST 전기및전자공학과(박사)
1993년~1996년: 디지콤정보통신연구소
1996년~1997년: 삼성전자
1997년~현재: 강원대학교 전기전자정보통신공학부 부교수
〈관심분야〉 이동통신, Ad-hoc 네트워크, 네트워크보안, 초고속 통신

이 용 (Yong Lee) 정회원
1997년: 연세대학교 컴퓨터과학과(석사)
2001년: 연세대학교 컴퓨터과학과(박사)
1993년~1994년: 디지콤 정보통신
2001년~2003년: 한국정보보호진흥원 전자서명인증관리센터
2003년~2004년: 이화여자대학교 계약교수
2004년~현재: 코넬대학교 방문연구원
〈관심분야〉 정보보호, 이동통신, Ad-hoc 네트워크 보안



김 화 종 (Hwa-Jong Kim) 정회원
1982년: 서울대학교(학사-전자공학)
1984년: KAIST 전기및전자공학과(석사)
1988년: KAIST 전기및전자공학과(박사)
1988년~현재: 강원대학교 전기전자정보통신공학부 교수
〈관심분야〉 데이터 통신, 컴퓨터네트워크, 네트워크 프로그래밍

정 충 교 (Choong-Kyo Jeong) 정회원
1982년: 서울대학교(학사-전기공학)
1984년: KAIST 전기및전자공학과(석사)
1989년: KAIST 전기및전자공학과(박사)
1989년~1995년: LG정보통신(주) 책임연구원
1995년~현재: 강원대학교 전기전자정보통신공학부 교수
〈관심분야〉 인터넷, 통신프로토콜, 통신망성능분석



이 동 은 (Dong-Eun Lee) 정회원
2005년 2월: 강원대학교 정보통신공학과 졸업
2005년 3월~현재: 강원대학교 컴퓨터정보통신공학과 석사과정
〈관심분야〉 네트워크, 정보보호, ad hoc 네트워크