Journal of The Korea Society of Computer and Information
Vol. 18, No. 9, September 2013
www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2013.18.9.079

검색가능 암호시스템을 위한 데이터 암호기법의 문제점 분석

손 정 갑*, 양 유 진**, 오 희 국*, 김 상 진**

Analysis of Data Encryption Mechanisms for Searchable Encryption

Junggab Son*, Yu-Jin Yang**, Heekuck Oh*, Sangjin Kim**

요 약

클라우드 컴퓨팅의 보급으로 최근 데이터 아웃소싱에 대한 요구가 매우 높아지고 있다. 하지만 클라우드 컴퓨팅의 근본적인 걱정인 외부 서버 신뢰 문제에 대한 만족할만한 수준의 해결책이 아직 제시되고 있지 못하다. 이 때문에 검색가능 암호화에 대한 연구가 최근에 다시 활발해지고 있다. 하지만 검색 기능에 대한 연구에만 집중되어 중요한 요소 중 하나인 데이터 암호메커니즘에 대한 연구는 상대적으로 소홀히 되고 있다. 적절한 암호메커니즘의 적용 없이는 검색가능 암호화를 실제 서버에 적용하는 것이 불가능하다. 이 논문에서는 다중 사용자가 이용하는 검색가능 암호시스템에서 지금까지 제안된 데이터 암호메커니즘과 사용 가능한 메커니즘들을 분석하여 그들의 장단점을 논한다. 분석 결과 논문에서 고려한 브로드캐스트 암호 기법, 속성기반 암호 기법, 프록시 재암호화 기법은 모두 적절한 해결책이 되지 못한다. 현존하는 기법들의 가장 큰 문제는 별도의 완전히 신뢰할 수 있는 서버가 필요하다는 것과 외부 사용자와 완전히 신뢰하지 못하는 서버 간 공모 공격을 방지할 수 없다는 것이다.

▸ Keywords : 검색가능 암호시스템, 브로드캐스트 암호시스템, 속성기반 암호시스템, 프록시 재암호화 시스템

Abstract

Recently, the need for outsourcing sensitive data has grown due to the wide spreading of cost-effective and flexible cloud service. However, there is a fundamental concern in using such

•제1저자 : 손정갑 •교신저자 : 김상진
•투고일 : 2013. 7. 9, 심사일 : 2013. 7. 29, 게재확정일 : 2013. 8. 19.
* 한양대학교 컴퓨터공학과(Dept. of Computer Science and Engineering, Hanyang University)
** 한국기술교육대학교 컴퓨터공학부(School of Computer Science and Engineering, Korea University of Technology and Education)
※ 이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2012-R1A2A2A 01046986).
※ 이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단 기초연구사업의 지원을 받아 수행된 연구임 (No. 2012- R1A1A2009152).
※ 이 논문은 2012년도 한국기술교육대학교 교수교육연구진흥비 지원에 의하여 연구되었음.

service since users have to trust external servers. Therefore, searchable encryption can be a very valuable tool to meet the security requirements of data outsourcing. However, most of work on searchable encryption focus only on privacy preserving search function and relatively lacks research on encryption mechanism used to actually encrypt data. Without a suitable latter mechanism, searchable encryption cannot be deployed in real world cloud services. In this paper, we analyze previously used and possible data encryption mechanisms for multi-user searchable encryption system and discuss their pros and cons. Our results show that readily available tools such as broadcast encryption, attribute-based encryption, and proxy re-encryption do not provide suitable solutions. The main problem with existing tools is that they may require separate fully trusted servers and the difficulty in preventing collusion attacks between outsiders and semi-trusted servers.

▸ Keywords : searchable encryption, broadcast encryption, attribute-based encryption, proxy re-encryption

I. 서 론

To provide secrecy or privacy, data can be maintained in an encrypted form. This is especially needed if data is outsourced and users do not want to reveal their data to external servers. However, this makes it difficult for the readers to selectively retrieve a specific data of their wanting since servers cannot access the inner contents. This becomes more difficult if multiple users should be allow to store and retrieve the same data since it additionally requires group management that must include user revocation. Moreover, to enhance privacy, users may also want to hide their access and search patterns.

To this end, SE (Searchable Encryption) can be used which was first proposed by Song et al. [1]. In 2004, Boneh et al. [2] proposed public key based SE which started this area of research in earnest. Basic usage scenario of searchable encryption is as follows.

- Step 1. (Storing Phase): A user, refer to as a writer, encrypts data and some keywords related to that data and send them to external storage.
- Step 2. (Querying Phase): A user, refer to as a reader, encrypts search keywords which are called trapdoors and send them to the external server.
- Step 3. (Search Phase): The server uses the trapdoors with the stored encrypted keywords to find a relevant data and sends the resulting data to the requested user.

In addition to above phases, in a multi-user setting, additional steps may be required to allow users to join or leave.

In SE, the following three basic cryptographic keys are used.

- DEK (Data Encryption Key): A cryptographic key used to encrypt the data by the writer. The legitimate readers must also be able to decrypt the data encrypted with DEK.
- KEK (Keyword Encryption Key): A cryptographic key used to encrypt the keyword by the writer.
- TGK (Trapdoor Generation Key): A cryptographic key used to generate a trapdoor by a reader. Only authorized readers should be able to generate a valid trapdoor.

Since the server uses the trapdoor against encrypted keywords, KEK and TGK normally have

some kind of relationship with each other. However, DEK does not have to be related to KEK or TGK.

Since Boneh et al.'s proposal [2], there have been many papers on SE, but to our knowledge, we have not found a paper that specifically focuses on DEK mechanism for SE. Even papers, that provide a solution for multi-user setting [3-4], do not provide an efficient DEK mechanism. In most schemes, the result of step 1 produces a ciphertext that is proportional to the number of readers. Moreover, they normally do not consider user revocation.

In this paper, we focus on DEK mechanism for SE in pursuit of finding efficient solutions for various settings. Especially, we categorize searchable encryption with respect to reader-writer model of searchable encryption. We investigate previous approaches and also consider tools such as BE (Broadcast Encryption) [5], ABE (Attribute-Based Encryption) [6], and ProxyRE (Proxy Re-Encryption) [7] and their combinations. Our research shows that existing technology all have deficits and cannot be directly applied to SE as suggested by some. It is even more difficult if we consider the possibility of collusion between the external server and outsiders. We would like to note that we do not describe the mathematical details of mechanisms proposed or considered for two reasons: mathematical details of protocols are not essential to understanding our arguments and for space reasons.

The remainder of this paper is organized as follows. In section 2, we review SE system focusing on date encryption mechanism for supporting multiple users. In section 3, we give a brief summary of previous works on data encryption mechanism for multi-user settings. In section 4, we consider various mechanisms for encrypting data in SE and give analysis of these mechanisms in section 5. Finally, we conclude and give future directions in section 6.

II. Searchable Encryption Model

1. Participants

There are following four types of participants in a SE system.

- Writer: A user that has permission to store data in the external server.
- Search server: An external server that maintains outsourced data and performs search requests on behalf of users.
- Reader: A user that has permission to retrieve data from the external server by sending trapdoors.
- Additional server: In some systems, another server is deployed to supplement search server in providing sophisticated requirements.

Most peculiar and unusual setting compared to other applications is that SE uses curious-but-honest model for search server. That is, search server will conform to every rule except that this server is anxious to see the content of encrypted data. More concrete definition of curious-but-honest model will be given in section 2.5. Due to this model, it may be very awkward to use additional servers. For example, in multi-user setting, we may need an additional server to deal with group management. However, conventional approach of using centralized group key system arise a contradicting situation of fully trusting one server and semi-trusting another one. One could think of using threshold-based secret sharing schemes to distribute rights to remove a single but powerful server. However, it would be difficult for users to find several trusted servers to support them when using public cloud services.

We would like to note that large companies or institutes should run their own servers for managing

groups or keys when outsourcing their sensitive data. This would lower the overall data management cost while not minimizing trust put on the external server. However, small companies and normal users would find it very difficult to use such scenario.

2. Types of Searchable Encryption Systems

SE systems can be divided into following types depending on the number of writers and readers and their relationship.

- SWSR (Single-Writer-Single-Reader): Only a single user can store and retrieve data.
- SWMR (Single-Writer-Multi-Reader): Only a single user can store, but multiple users can retrieve data.
- MWSR (Multi-Writer-Single-Reader): Multiple users can store, but only a single designated user can retrieve data.
- MWMR (Multi-Writer-Single-Reader): Multiple users can store and retrieve data.

In multi-user setting, group concept is needed and systems should provide a way to allow users to join and leave. We could characterize such systems with respect to who controls the group management. In SWMR, the single writer should manage the group whereas in MWMR separate server may be needed to manage the group. In MWSR, group management may not be required at all. In other words, we could have a model where there is no restriction on writers of the system. Email system can be viewed as MWSR model with no restriction on senders. MWMR is the most complicated setting where the group of writers and readers can even be different groups. Several models can be simultaneously provided. For example, some data may be restricted to SWSR whereas other data may use MWMR.

3. Usage Model of Searchable Encryption System

Normal usage model of SE is already given in section 1 which includes 3 steps: storing, querying, and searching. Although there are approaches to enrich the search flexibility and the quality of the search result, many systems only provide boolean keyword search and assumes that only a small number of keywords are associated with each data. Moreover, these keywords must also be selected by the writer of the data. However, this paper does not focus on this aspect of SE.

In querying phase, most previous works do not explicitly describe the mechanism used to authenticate readers. Although only those users with correct TGK can generate trapdoors, trapdoor itself cannot be used to authenticate and deny illegal users since trapdoors are normally reusable (trapdoor for same keyword generated by a specific user does not change) and they are not verifiable by search servers.

The 3 steps of conventional SE are normally described as an algorithm which means the main participant performs the given algorithm by himself and sends the result to the opposite party. However, in [8], storing and querying phase requires several interactions between parties involved. Obviously this increase communication cost but can be considered to solve those problems which were difficult to overcome in the conventional setting. We called this approach interactive SE system.

4. Data Encryption Mechanism

One of the fundamental reasons for using SE is to maintain data in an encrypted form to protect them from unnecessary disclosure. As a result, readers of the system also receive them in the same form. Therefore, legal receivers must further decrypt them to obtain the actual data. Let's review issues regarding DEK in each four types of SE system given in section 2.2.

In SWSR, since the reader and the writer are identical entity, a symmetric key can be used for DEK. In MWSR, since there is only a single reader, the public key of the reader can be used for DEK.

Therefore, deciding the data encryption mechanism for SWSR and MWSR is very trivial. On the other hand, since we need to consider user revocation, it is not trivial to determine a data encryption mechanism for SWMR and MWMR.

Another important aspect is the need for updating previously encrypted data. One could think that we could trust the server to deny request from revoked users. However, since we use curious-but-honest model, it is not clear whether we can trust the server on this matter. Even if we can trust the server, the system may still be vulnerable if the actual data is not double encrypted using a session key. For example, revoked users who have the previous DEK can still obtain data by eavesdropping the outgoing channel of the server.

Obvious solution to above problem is to re-encrypt all the data. However, if the maintained data are very large, it would be very impractical if the system has to re-encrypt all the previous data. Moreover, the server who maintains the data cannot perform the re-encryption because users do not want to reveal the content of their data. To summarize, the followings are functional requirements of DEK for SE.

- DEK should support multi-user setting that should include a way to handle group management.
- DEK should provide some kinds of means to efficiently update encrypted data maintained in the external storage.

5. Adversary Model

This adversary model is concerned only with the security of data encryption mechanism of SE. The goal of the adversary in this model is to obtain some kind of information about the data stored in the server. Any type of participants can be an adversary of the system, but we will only concentrate on outsiders and collusion between the outsider and the search server. The revoked mem- bers of a group are also considered as an outsider.

We assume that outsiders can obtain all the data that are exchanged between the server and legal users of the system. A system may use additional secure channel to exchange values but these values can always be obtained by the adversaries if they can collude with the server. Previous papers do not consider collusion between a server and users even if they assume curious-but-honest model. However, since they are curious, we believe they will be tempted to collude with users to relieve their curiosity.

6. Access Control Model

In a multi-user setting, depending on the customer need, various types of access control model may be used. However, in this paper, we consider following three types.

- ACT1. Access control on a data depends on each user separately.
- ACT2. Access control on a data depends on the group the user is affiliated with.
- ACT3. Access control on a data depends on user's rank.

In ACT2, each user may belong to several groups. In ACT3, users holding higher rank can see all the data that can be accessed by lower ranked users.

III. Related Works

Many previous papers on SE do not concentrate on data encryption mechanism. However, data encryption mechanism is an essential component which makes the entire system obsolete if it does not provide the necessary security and efficiency. Especially, in a multi-user setting, several users must be able to decrypt the given ciphertext and it should consider a way to revoke users who no longer have the privilege to access the data. To current,

previous approaches do not consider existence of multiple groups and their effect on the server load. In this section, we preview data encryption mechanisms used by previous proposals that consider multi-user settings.

Curtmola et al. [9] proposed a SWMR model based SE system that use only symmetric encryption. In this system, the reader group and the single writer share a common symmetric group key which is used to encrypt the trapdoor. The single writer has to manage the addition and revocation of users using centralized group key mechanism. The search server is trusted to deny revoked users by examining the trapdoor given by users. In other words, they do consider the possibility of collusion between the server and past group members. Moreover, they do not discuss which key is used for DEK.

Hwang and Lee [4] proposed public key based SWMR system. In this system, the single writer must have all the public keys of multiple readers to store a data and resulting ciphertext is proportional to number of readers. However, this system does not provide any kind of group management. In other words, a new group is established each time a writer stores a data. It is also unclear how the server maintains such data in its storage. In a naive approach, such data may be replicated in each reader's allocated storage.

Bao et al. [8] proposed a MWMR system. All members of a group share a common key which is used as DEK. However, when a user is revoked, this key is not updated. Therefore, revoked users can still obtain data by eavesdropping the outgoing channel of the server. Moreover, the responsibility of denying requests from revoked user is given to the search server. Therefore, this approach is vulnerable to collusion attack by the server and a revoked user.

Shao et al. [10] proposed a MWMR system that uses ProxyRE for DEK. However, they used a very weak proxy re-encryption technique that enables proxy to obtain the private key of a user by colluding with other users. They also do not consider user revocation and do not analyze the burden of maintaining re-encryption keys by the server.

Dong et al. [11] also proposed a MWMR system that uses ProxyRE for DEK. They used RSA-based ProxyRE which can reduce the number of re-encryption key maintained by the proxy compared to other re-encryption techniques. However, unlike others, this scheme requires a separate server to issue public key pairs for users using its master key. Therefore, this server can obtain all the encrypted data which may not suit well with SE environment. Moreover, the used ProxyRE technique is also vulnerable to collusion attacks.

IV. Possible Data Encryption Key Mechanism for Searchable Encryption

In this section, we examine possible DEK solutions for SE systems. We only consider SWMR and MWMR models because solutions for other two models are very straightforward.

1. Previous Approach of Using Public Key Encryption

In [4], the ciphertext includes a separate value which is needed by each receiver to decrypt it. Therefore, a writer must obtain all the public keys of receivers before encrypting a data and the operation cost and the size of resulting ciphertext is proportional to the number of receivers. Therefore, this approach is not scalable and do not provide ways to dynamically manage group membership.

2. Broadcast Encryption

It is very intuitive to consider BE for DEK in multi-user settings for SE since it can effectively solve user revocation problem. However, there is a subtle difference between SE and BE. In BE, it is about broadcasting a new data to current members and does not consider past encrypted data. However,

in SE, although writers can encrypt the data according to the current group, the encrypted data are maintained and serviced again and again to a group that may change after data has been stored.

Although there are distributed group key mechanisms, it is not practical since it requires too many interactions between members to establish and update group keys. Therefore, a centralized group key mechanism looks more suitable for SE. Another consideration using group key mechanism is that one could think stateless group key mechanism [12] may be needed in SE environment. However, we could store the required update messages and forward them to readers when they access the system. In other words, SE is different to real-time video broadcasting environment.

Using this approach, we could easily design a mechanism to provide ACT2 and ACT3. However, it is not easy to provide ACT1 because a new group may need to be dynamically constructed for each new data. Therefore, stateless BE may be considered for ACT1. However, in this case, entire user set may have to be fixed in advance.

For ACT2, each group is assigned a group key using a centralized group key mechanism. For a each data, which should be accessible to multiple groups, we encrypt the data using each group's group key. Therefore, this approach is very inefficient since we require operations and storage proportional to number of groups that can access the given data.

For ACT3, for each rank group, a separate logical key hierarchy is used and group key (the key assigned to the root of the tree) of each group is linked using a hash chain. In this case, a key allocated to a node in the logical key tree must be independent of each other like LKH [5]. Therefore, members who belong to upper rank group can use their group key to compute group keys of lower rank group. When there is a change to one of the group, corresponding group and lower rank groups require group key update. However, lower rank groups only require a single broadcast message that includes the new group key encrypted using the old one.

Although centralized group key mechanism can be used to provide DEK in SE, it still has the following problems.

- Since the group manager generates and distributes the group key, the manager can access all the encrypted data.
- Only the members of a specific group can generate data for that group. Public BE system [13] may be used to overcome this problem.
- Although such mechanism provides scalable and efficient way to revoke users, such mechanism do not provide a way to update previously encrypted but maintained data.
- Although stateful mechanisms can be used, it would be preferable to use stateless mechanisms. However, there exists no widely accepted stateless mechanism that supports dynamic group.

3. Attribute-based Encryption

Since ABE can provide fine-grained access control to encrypted data and assumes semi-trusted storage server, it is another technology that can be intuitively considered for DEK. ABE also has some group notion in that users can encrypt data so that multiple users holding some attributes can all decrypt it. However, there is a subtle difference between ABE and SE. ABE setting includes a key server and storage, whereas SE only includes external storage.

Most suitable usage of ABE is for SWMR model and it can provide all three ACTs due to the ample access control expressiveness of ABE. In this case, the single writer can also play the role of the key server thus, removing the need for introducing another trusted server. However, for MWMR, it is inevitable to adopt a separate trusted server to issue keys to members of groups.

Despite the advantages of ABE, there are

following problems of using ABE as DEK in SE.

- Since the current ABEs are based on identity-based systems, keys in ABE are issued by a trusted server which results in a similar problem of requiring a fully trusted server.
- Attribute revocation and user revocation in ABE does not yet have a satisfiable solution. Most interesting solution to revocation problem in ABE is using NOT operation [14]. However, the size of the ciphertext increases as users are revoked from that ciphertext. Moreover, this would require updating ciphertexts maintained at the server.
- Recently, group key technology was integrated with ABE to solve the revocation problem [15] but these kinds of solutions do not go well with SE since it only increase the required number of fully trusted servers.

4. Proxy Re-encryption

We now look at ProxyRE as a solution for DEK since it has already been considered in SE context before [10-11]. Usage scenario is as follows.

- The writer encrypts the data using its public key.
- The writer generates re-encryption keys for users that have read access to the given data.
- The encrypted data and re-encryption keys are sent to the search server.

The most interesting feature of ProxyRE is that it does not require additional servers. Users can generate re-encryption keys without interacting with any other parties if they have the public key of targeted users.

However, there are the following issues about using ProxyRE for DEK.

- The additional storage cost of maintaining re-encryption keys may be high. In general, if there are n users of system, the server may have to maintain $n \times n - 1$re-encryption keys.
- Conventional ProxyRE can only provide coarse-grained access control. All data encrypted with A's public key can be transformed for B if the proxy has re-encryption key. Conditional ProxyRE [16] has been introduced to alleviate this problem. Furthermore, recently ABE has been integrated with ProxyRE [17]. However, advantages of using ABE to control delegation of decryption rights compared to just using ABE is insignificant and it only increases one additional level. Therefore, from DEK point of view, it is more preferable to use just ABE.
- ProxyRE also does not normally consider user revocation. We could revoke a user by removing the stored re-encryption key, but this would require honesty from semi-trusted search server.
- It is difficult to provide ACT2 and ACT3 since the nature of ProxyRE does not have group concept (i.e. in ProxyRE, each user originally encrypts the data using their own public key) or considers scalability of services.

For ACT1, instead of using his public key, the writer can generate a random public key pair and used it to encrypt data. We could also use conditional ProxyRE for ACT1. However, the number re-encryption keys required in both cases do not change.

In MWMR model, a member of a group should be able to construct a ciphertext that can be decrypted by other members of that group. Naively, we could think of following method: a TTP generates a group public key pair and re-encryption keys for all the members of the group and members of the group use the group public key to encrypt data and the search server can re-encrypt the data when users access a data. However, it would be better to just use group key mechanism instead.

We could also consider a solution using conditional ProxyRE by associating a condition with each group. Although this looks like a feasible approach, the fundamental nature of ProxyRE

(encrypting data with their own public key) does not go well with group encryption/decryption. Moreover, the same condition must be shared between members of group, which may require group key mechanism.

V. Analysis

From our observations, all the techniques considered in this paper do not provide efficient solutions for DEK. The one of the main drawbacks of these approaches is that separate fully trusted server is needed which hinders them as a general acceptable solution. Although, in SWMR model, the single writer can control the group management for users, the need of separate server is inevitable in MWMR model. Another unsolved problem is that we have not found an efficient way to update encrypted data to deny revoked users from acquiring them even if they collude with the search server.

It is clear that it is infeasible to remove decryption rights that has already been allocated without re-encrypting the data. Since it is difficult to delegate the re-encryption of stored data to the search server, only viable solution would be to download all the data and decrypt and re-encrypt with a new key, which is clearly inefficient and would not be acceptable to many users.

Most feasible solution considered in this paper for DEK is as follows.

- SWMR-ACT1: stateless BE
  - Ciphertext size: $O(\log n)$, where n is the number of users considered.
  - Problems: updating previous encrypted data, addition and removal of users.
- SWMR-ACT2, ACT3: ABE
  - Ciphertext size: proportional to the attributes assigned to the cipertext.
  - Problems: updating previous encrypted data, user revocation.
- MWMR-ACT3: BE with hash-chain
  - Ciphertext size: $O(1)$
  - Problems: updating previous encrypted data, requires separate trusted server

VI. Conclusion

Searchable encryption may be a valuable tool to be considered as a solution for security concerns in cloud computing service. However, as shown in this paper, the current state lacks practical solutions. Especially, we consider adversary model that includes possibility of collusion between users and the semi-trusted search server. Papers dealing with searchable encryption should clearly state that although it is possible for search to be done without revealing any information to the server performing the service, it is difficult to provide an efficient data encryption mechanisms for multiuser settings. In the future, we should try to find a mechanism that can delegate the search server to update the stored encrypted data to satisfy security requirements of dynamic change to group members.

참고문헌

[1] D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," Proc. IEEE Symp. on Security and Privacy, pp. 41-55, May 2000.

[2] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," Advances in Cryptology, Eurocrypt 2004, LNCS 3027, pp. 506-522, Springer, May 2004.

[3] J. Baek, R. Safav-Naini, and W. Susilo, "On the integration of public key data encryption and public key encryption with keyword search," Proc. Information Security Conference (ISC 2006), LNCS 4176, pp. 217-232, Springer, September 2006.

[4] Y.H. Hwang and P.J. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," Proc. Pairing

2007, LNCS 4575, pp. 2-22, Springer, July 2007.
〔5〕 C.K. Wong, M. Goulda, and S.S. Lam, "Secure group communications using key graphs," IEEE/ACM Trans. on Networking, Vol. 8, No. 1, pp. 16-30, Feb. 2000.
〔6〕 J. Bethencourt, A. Sahai, and B. Waters, "Ciphertextpolicy attribute-based encryption," Proc. of the IEEE Symp. on Security and Privacy, pp. 321-334, May 2007.
〔7〕 G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," ACM Trans. on Information and System Security, Vol. 9, No. 1, pp. 1-30, Feburary 2006.
〔8〕 F. Bao, R.H. Deng, X. Ding, and Y. Yang, "Private query on encrypted data in multi-user setting," Proc. Information Security Practice and Experience (ISPEC 2008), LNCS 4991, pp. 71-85, Springer, April 2008.
〔9〕 R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions," Proc. ACM Conf. on Computer and Communications Security (CCS '06), pp. 79-88, Oct. 2006.
〔10〕 J. Shao, Z. Cao, X. Liang, and H. Lin, "Proxy re-encryption with keyword search," Information Sciences, Vol. 180, No. 13, pp. 2566-2587, July 2010.
〔11〕 C. Dong, G. Russello, and N. Dulay, "Shared and searchable encrypted data for untrusted servers," Proc. Conf. Data and Applications Security (DAS 2008), LNCS 5094, pp. 127-143, Springer, July 2008.
〔12〕 D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracing schemes for stateless receivers," Advances in Cryptology, Crypto 2001, LNCS 2139, pp. 41-62, Springer, August 2001.
〔13〕 J.H. Park, H.J. Kim, M.H. Sung, and D.H. Lee, "Public key broadcast encryption scheme with shorter transmissions," IEEE Trans. on Broadcasting, Vol. 54, No. 3, pp. 401-411, September 2008.
〔14〕 A. Lewko, A. Sahai, and B. Walters, "Revocation systems with very small private keys," Proc. IEEE Symp. Security and Privacy, pp. 273-285, May 2010.
〔15〕 J. Hur and D. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," IEEE Trans. on Parallel and Distributed Systems, Vol. 22, No. 7, pp. 1214-1221, July 2011.
〔16〕 J. Weng, Y. Yang, Q. Tang, R.H. Deng, and F. Bao, "Efficient conditional proxy re-encryption with chosen-ciphertext security," Proc. Information Security Conf. (ISC 2009), LNCS 5735, pp. 151-166, Springer, September 2009.
〔17〕 J. Zhao, D. Feng, and Z. Zhang, "Attribute-based conditional proxy re-encryption with chosen-ciphertext security," Proc. Global Telecommunication Conf. (GLOBECOM 2010), pp. 1-6, December 2010.

저 자 소 개

손 정 갑

2009: 한양대학교
컴퓨터공학과 공학사,
2011: 한양대학교
컴퓨터공학과 공학석사,
현 재: 한양대학교
컴퓨터공학과 박사과정.
관심분야: 클라우드컴퓨팅보안,
암호기술응용
Email : jgson@infosec.hanyang.ac.kr

양 유 진

2011: 한국기술교육대학교
컴퓨터공학부 공학사,
2013: 한국기술교육대학교
컴퓨터공학과 공학석사.
관심분야: 암호기술응용
Email : sunyujin@koreatech.ac.kr

오 희 국

1983: 한양대학교
전자공학과 공학사,
1989: 아이오와주립대학
전자계산학과 공학석사,
1992: 아이오와주립대학
전자계산학과 공학박사,
1994: 한국전자통신연구원 선임연구원,
현 재: 한양대학교
컴퓨터공학과 교수.
관심분야: 암호프로토콜, 네트워크보안
Email : hkoh@hanyang.ac.kr

김 상 진

1995: 한양대학교
전자계산학과 공학사,
1997: 한양대학교
전자계산학과 공학석사,
2002: 한양대학교
전자계산학과 공학박사,
현 재: 한국기술교육대학교
컴퓨터공학부 부교수.
관심분야: 암호기술응용
Email : sangjin@koreatech.ac.kr