Information Sciences 258 (2014) 355–370

Contents lists available at SciVerse ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Time-based proxy re-encryption scheme for secure data sharing in a cloud environment

Qin Liu [a,b], Guojun Wang [a,*], Jie Wu [b]

[a] School of Information Science and Engineering, Central South University, Changsha, Hunan Province 410083, PR China
[b] Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, USA

ARTICLE INFO

Article history:
Available online 26 October 2012

Keywords:
Cloud computing
Time
Proxy re-encryption
Attribute-based encryption

ABSTRACT

A fundamental approach for secure data sharing in a cloud environment is to let the data owner encrypt data before outsouring. To simultaneously achieve *fine-grained access control on encrypted data* and *scalable user revocation*, existing work combines attribute-based encryption (ABE) and proxy re-encryption (PRE) to delegate the cloud service provider (CSP) to execute re-encryption. However, the data owner should be online in order to send the PRE keys to the CSP in a timely fashion, to prevent the revoked user from accessing the future data. The delay of issuing the PRE keys may cause potential security risks. In this paper, we propose a time-based proxy re-encryption (TimePRE) scheme to allow a user's access right to expire automatically after a predetermined period of time. In this case, the data owner can be offline in the process of user revocations. The basic idea is to incorporate the concept of time into the combination of ABE and PRE. Specifically, each data is associated with an *attribute-based access structure* and an *access time*, and each user is identified by a set of *attributes* and a set of *eligible time periods* which denote the period of validity of the user's access right. Then, the data owner and the CSP are required to share a *root secret key* in advance, with which CSP can automatically update the access time of the data with the time that it receives a data access request. Therefore, given the re-encrypted ciphertext, only the users whose attributes satisfy the access structure and whose access rights are effective in the access time can recover corresponding data.

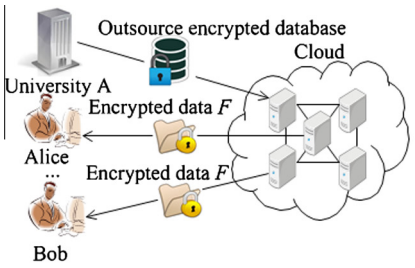
1. Introduction

Cloud computing has increasingly become a commercial trend due to its desirable properties, such as scalability, elasticity, fault-tolerance, and pay-per-use [1]. Small and medium-sized organizations, in particular, can achieve great flexibility at a low price by outsourcing their data and query services to the cloud. The cloud infrastructures are more powerful and reliable than personal computing devices, but they are still susceptible to internal threats (e.g., via virtual machines) and external threats (e.g., via system vulnerabilities) that may leak user sensitive data [7,25]. Therefore, many organizations still hesitate to adopt cloud services [24].

To prevent unsolicited disclosure of sensitive information, *data owners* may have to encrypt their data before outsourcing [6,15,18]. In this way, only the authorized *users* with the decryption keys can recover the data, and other unsolicited accessors without the decryption keys, e.g., the cloud service provider (CSP), cannot execute decryption, even if they successfully

* Corresponding author.
E-mail address: csgjwang@mail.csu.edu.cn (G. Wang).
URL: http://trust.csu.edu.cn/faculty/~csgjwang (G. Wang).

0020-0255/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2012.09.034



Data F's access structure:
"Staff" OR
"Student AND CIS"

User information

User ID	Attributes	Effective time
Alice	Staff, CIS	01/01/2012-12/31/2012
Bob	Student, CIS	05/01/2012-06/30/2012
...	...	...

Fig. 1. University A outsources an encrypted database to the cloud.

obtain the ciphertexts stored in the cloud. However, new problems, such as *fine-grained access control on the encrypted data* and *scalable user revocation*, emerge for this solution.[1]

To illustrate, let us consider the following application scenario, as shown in Fig. 1. Suppose that University A outsources the electronic library database to a cloud for easy access by its staff and students. For the protection of copyright, each piece of data is encrypted before outsourcing. In this application, the staff and students are users, and University A is the data owner who will specify the access structure for each data, and will distribute decryption keys to users. Once joining University A, each user will first be assigned an access right with certain validity for accessing the outsourced database. Once the period of validity passes, this user should request an extension for his access right from University A.

In Fig. 1, data F's access structure stipulates that only the staff or the students in computer and information sciences (CIS) department have the right to access it. In this access structure, the data owner does not know the exact identities of the authorized users, but rather he only has a way to describe them using certain descriptive attributes, such as `Staff`, `Student`, and `CIS`. Therefore, the adopted encryption system should have the ability to efficiently implement a fine-grained access control over attributes.

Ciphertex-policy attribute-based encryption (CP-ABE) [3,19] as a promising branch of ABE [23] has such a property. In CP-ABE, users are identified by a set of *attributes* rather than an exact identity. For each eligible attribute, the user will be issued a *user attribute secret key* (UAK). Each data is encrypted with an *attribute-based access structure*, such that only the users whose attributes satisfy the access structure can decrypt the ciphertext using their UAKs. For example, for data which is encrypted with the access structure $\{(\texttt{Student} \wedge \texttt{CIS}) \vee \texttt{Staff}\}$, either users with attributes `Student` and `CIS`, or users with attribute `Staff`, can recover data.

Furthermore, from the above application scenario, we observe that each user's access right is only effective in a predetermined period of time. For example, the effective time of Alice's access right is from 01/01/2012 to 12/31/2012 and she can access the database in year 2012, but the effective time of Bob's access right is from 05/01/2012 to 06/30/2012 and thus he cannot access the database after June.

A naïve way is to let University A expose the effective time of each user's access right to the CSP, with which the CSP can execute user revocation correctly. The main drawback of this approach is that the CSP will know the effective time of each user's access right, which may cause potential leakage of sensitive information. For example, the CSP may guess the user whose access right has longer effective time is in a more important position than that with shorter-term access right. Due to the same reason, the ticket-based access control [21], which requires the users to expose their tickets to the CSP, may also expose the effective time of each ticket.

Another approach is to require the data owner to personally execute user revocation. A revoked user still retains the keys issued earlier, and he can recover data if he obtains corresponding ciphertexts. To prevent the revoked user from accessing further data, the data owner needs to immediately re-encrypt the ciphertexts. Furthermore, the data owner needs to distribute new keys to the remaining authorized users for them to access the database. When there are frequent user revocations, this solution will result in a heavy workload on the data owners.

A better solution should take full advantage of abundant resources in a cloud by delegating the CSP to execute computationally intensive tasks in user revocations, while *leaking the least information* to the CSP. Existing work [29,28] proposed the idea of applying the combination of proxy re-encryption (PRE) [4,13] and ABE to a cloud environment. This approach requires that once a user is revoked from a system, the data owner should send the *PRE keys* to the CSP, with which the CSP can be delegated to re-encrypt corresponding ciphertexts. The original schemes in [29,28] also allow the CSP to be delegated to distribute the *update keys* to remaining authorized users for them to generate new UAKs. However, the CSP should first know the identities of these authorized users, and it will finally know the effective time of each user. Therefore, to avoid leaking additional information, the data owner should distribute the update keys on his own. Due to the security of ABE and PRE, the CSP cannot know neither underlying data nor UAKs. The main problem of this approach is that the data owner

[1] There are certainly more than two problems existing in such an environment, e.g., impersonation of user identity and compromising data integrity. We do not solve all the potential security threats, but only address those directly related to our work.