Efficient method for re-encryption in unreliable clouds using r3 algorithm essay ...

ASSIGN BUSTER

Abstract— Aim of the project is to re-encrypt the data and revoke the access rights from the users. In this paper, we propose an efficient data retrieval scheme using attribute-based encryption. The proposed scheme is best suited for cloud storage systems with substantial amount of data. It provides rich expressiveness as regards access control and fast searches with simple comparisons of searching entities. The proposed scheme also guarantees data security end-user privacy during the data retrieval process. A key approach to secure cloud computing is for the data owner to store encrypted data in the cloud, and issue decryption keys to authorized users. The cloud storage based information retrieval service is a promising technology that will form a vital market in the near future. Although there have been copious studies proposed about secure data retrieval over encrypted data in cloud services, most of them focus on providing the strict security for the data stored in a third party domain. However, those approaches require astounding costs centralized on the cloud service provider, this could be a principal hindrance to achieve efficient data retrieval in cloud storage.

Keywords— Attribute-based encryption, cloud computing, Proxy re-encryption.

I. Introduction

Cloud computing, which gets its name as a representation for the Internet [1], is becoming a popular term and has been used by an increasing number of organizations. In cloud computing environment, services are not provided by a single server or a small group of servers; instead, various computing and storage services are provided by some collection of data centers owned and maintained by a third party.

Cloud infrastructures can be roughly categorized as either private or public. In a private cloud, the infrastructure is managed and owned by the customer and located on-premise (i. e., in the customers region of control) [2]. In particular, this means that access to customer data is under its control and is only granted to parties it trusts. In a public cloud the infrastructure is owned and managed by a cloud service provider. This means that customer data is outside its control and could potentially be granted to entrusted parties.

An alternative solution is to apply the proxy re-encryption (PRE) technique. . Proxy re-encryption allows a proxy to convert a cipher text computed under Alice’s public key into one that can be opened by Bob’s secret key. There are many useful applications of this primitive. For instance, Alice might wish to temporarily forward encrypted email to her colleague Bob, without giving him her secret key [3]. In this case, Alice the delegator could designate a proxy to re-encrypt her incoming mail into a format that Bob the delegate can decrypt using his own secret key. Clearly, Alice could provide her secret key to the proxy but this requires an impracticable level of trust in the proxy.

The primary advantage of PRE scheme is that they are unidirectional (i. e., Alice can delegate to Bob without Bob having to delegate to her) and do not require delegators to reveal all of their secret key to anyone – or even interact with the delegate – in order to allow a proxy to re-encrypt their cipher texts [4] . In this schemes, only a inadequate amount of trust is placed in the proxy.

For example, it is not able to decrypt the cipher texts it re-encrypts and we prove our schemes secure even when the proxy publishes all the re-encryption information it knows. This enables a number of applications that would not be sensible if the proxy needed to be fully trusted.

A better solution is to allow each cloud server to autonomously re-encrypt data without receiving any command from the data owner. In this paper, a time based re-encryption scheme is proposed, which allows each cloud server to automatically Re-encrypt data based on its internal time. In this scheme the data is associated with an access control and an access time. Each user is issued keys associated with attributes and attribute effective times [5].

The user can decrypt the data using the keys with attributes rewarding the access control, and access time. The data owner and the CSP share a secrete key with which each cloud server can re-encrypt data by updating the data access time. We propose a PRE scheme that achieves both the standard-model CCA security (i. e., without random oracles) and the proxy invisibility.

To satisfy two security conceptions simultaneously, our construction is based on the ideas of the first standard-model CCA-secure (non-type-based) PRE scheme and the fully secure anonymous identity-based encryption scheme. We use these techniques to achieve the standard-model CCA security. In, a private key generator of identity-based encryption combines the master key and an identity in the form of an inverse number for anonymity [6]. Similarly, to generate a re-encryption key, we combine the delegator’s private key and a type into an inverse form for proxy invisibility. The security proof of the proposed scheme is given in a formal security model.

II. Related Work

ABE is a new cryptographic technique that resourcefully supports fine grained access control. Fine-grained access control systems assist granting differential access rights to a set of users and allow flexibility in specifying the access rights of individual users. Hierarchical attribute-based encryption (HABE) model by combining a HIBE system and a CP-ABE system, to provide fine-grained access control and full delegation.

Our scheme relies on time to re-encrypt data. However, in a cloud, the internal clock of each cloud server may differ. There have been several solutions to this problem. For instance, proposed a probabilistic synchronization scheme, for reading remote clocks in networks subject to unbounded random message delays. The method can be used to improve the precision of both internal and external synchronization algorithms. To achieve loose synchronization in the cloud, and to determine the maximal time difference between the data owner and each cloud server by applying these techniques, our scheme can always achieve correct access control in unreliable clouds.

ABE is a new cryptographic technique that efficiently supports fine grained access control. The combination of PRE and ABE was first introduced by, and extended by. In a hierarchical attribute-based encryption (HABE) scheme is proposed to achieve high performance and full delegation. The main difference between prior work and ours is that we do not require the underlying cloud infrastructure to be reliable in order to ensure correctness. Our scheme relies on time to re-encrypt data. However, in a cloud, the internal clock of each cloud server may differ. There have been several solutions to this problem. For instance, proposed a probabilistic synchronization scheme, which exchanges messages to get remote servers' accurate clocks with high probability [7]. III. Preliminary

A. System Model

We assume that the system is composed of the following parties: the CSP, the trusted third party (TTP), enterprise users, end users, and internal trusted parties (ITPs). The first two parties: the CSP operates a large number of interconnected cloud servers with abundant storage capacity and computation power to provide high- quality services and the TTP is responsible for generating keys for the CSP and enterprise users [8]. B. Design model

The main design goal is to achieve scalable user revocation while protecting data security in cloud computing. Specifically, we categorize our goal into the following points: • Fine Grained Access control: The data owner can specify expressive access structure for each data. • Data consistency: This requires that all data users who request file F, should obtain the same content in the same time slice. • Data confidentiality: The CSP and malicious users cannot recover data without the data owner's permission. • Cost Efficiency: The re-encryption cost on the CSP is relatively low.

C. Adversary Model

There are two kinds of adversaries in the system: CSP, and malicious users.

The CSP will correctly execute the protocol defined previously, but may try to gain some additional information about the stored data. The malicious user wants to access the data, to which he is not eligible to access [9]. The communication channels are assumed to be secured under existing security protocols such as SSL to protect data security during information transferring. Note that both a CSP, and malicious users, can exist together. We assume that the CSP will not collude with any malicious user. However, malicious users may collude to obtain additional information. For example, if Alice is authorized to possess attributes a1. . . am from T S1to T Sn , she will be issued keys as is shown in Table I.

The security requirements of the R3 scheme are as follows:

1) Access control correctness: This requires that a data user with invalid keys cannot decrypt the file.

2) Data consistency: This requires that all data users who request file F, should obtain the same content in the same time slice.

3) Data confidentiality: The file content can only be known to data users with valid keys. The CSP is not considered a valid data user.

4) Efficiency: The cloud servers should not re-encrypt any file unnecessarily. This means that a file that has not been requested by any data user should not be re-encrypted.

IV. ALGORITHM ANALYSIS Algorithm: Extended R3 (a synchronized clock with delays)

While Receive write commands W (F, ti+1, seqnum) do

```
If Current time is earlier than ti+1 + α then
    Build Window i for file F
    Commit the write command in Window i at ti+1 + α
Else
    Reject the write command
    Inform the data owner to send write command earlier
While Receive a read request R (F, T Si) do
    If Current time is later than ti+1 + α then
        Re-encrypt the file in Window i with T Si
    Else
        Hold on the read command until ti+1 + α
```

V. IMPLEMENTATION

There are mainly logins for admin, data owner and user.

Project Software’s used for public cloud execution:

1) Data Storage: Iden Cloud

2) Code Deployment: CPanel

3) Domain: Presproject. org

1) Cloud Storage:

Amazon was one of the first companies to offer cloud services to the public, and they are very sophisticated. Amazon offers a number of cloud services, including • Elastic Compute Cloud (EC2): Offers virtual machines and extra CPU cycles for your organization.

• Simple Storage Service (S3): Allows you to store items up to 5GB in size in Amazon’s virtual storage service.

- Simple Queue Service (SQS): Allows your machines to talk to each other using this message-passing API.

- Simple DB: A web service for running queries on structured data in real time. This service works in close conjunction with Amazon Simple Storage Service (Amazon S3) and Amazon Elastic Compute Cloud (Amazon EC2), collectively providing the ability to store, process, and query data sets in the cloud [10].

Step 1: First Login as Data Owner by entering the username and password
Step 2: Now the Data owner will upload the file that he want to share with the users who are authenticated to him by securely encrypting the file with their respective login id’s.

Step 3: Now the Data Owner will give the access rights to the specific users and the server should login and activate the file before the users enter the key and get the rights to download the file.

Step 4: The server should activate the file to make the users to authenticate it. After that the user should login by entering the username and password.

Step 5: Now the user should select the specific server id and enter the decryption key and download the file. Once the user downloaded the file the file will be automatically re-encrypted by using R3 scheme.

VI. Conclusion

In this paper, we proposed the Time based scheme to achieve fine-grained access control and scalable user revocation in a cloud environment. Our scheme enables each user’s access right to be effective in a pre-determined period of time, and enable the CSP to re-encrypt cipher texts automatically, based on its own time. Thus, the data owner can be offline in the process of user revocations. The main problem with our scheme is that it requires the effective time periods to be the same for all attributes associated with a user. Although we provide a possible improvement, the users will be issued more UAKs. Our future work is to allow different effective time periods for different attributes associated with a user, without increasing the number of UAKs associated with each user.

References

[1]S. Kamara and K. Lauter, “ Cryptographic cloud storage,” Financial Cryptography and Data Security, 2010.

[2]M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, “ A view of cloud Computing,” Communications of the ACM, 2010.

[3]A. Sahai and B. Waters, “ Fuzzy identity-based encryption,” Advances in Cryptology–EUROCRYPT, 2005.

[4]V. Goyal, O. Pandey, A. Sahai, and B. Waters, “ Attribute-based encryption for fine-grained access control of encrypted data,” in Proc. of ACM 2006.

[5]J. Bethencourt, A. Sahai, and B. Waters, “ Cipher text-policy attribute-based encryption,” in Proc. of IEEE Symposium on S&P, 2007.

[6]M. Blaze, G. Bleumer, and M. Strauss, “ Divertible protocols and atomic proxy cryptography,” Advances in Cryptology–EUROCRYPT, 1998.

[7]A. Boldyreva, V. Goyal, and V. Kumar, “ Identity-based encryption with efficient revocation,” in Proc. of ACM CCS, 2008.

[8]G. Wang, Q. Liu, and J. Wu, “ Hierarchical attribute-based encryption for fine-grained access control in cloud storage services,” in Proc. of ACM CCS (Poster), 2010.

[9]S. Yu, C. Wang, K. Ren, and W. Lou, “ Achieving secure, scalable, and fine-grained data access control in cloud computing,” in Proc. of IEEE [10]S. Satyanarayana, T. Gopikiran, B. Rajkumar. “ Cloud Business Intelligence” international journal of advanced and innovative research (ijair) volume 1 issue 6, 2012.