**Secure Data Sharing in Cloud Computing Using Revocable-Storage Identity-Based Encryption**

**ABSTRACT:**

Cloud computing provides a flexible and convenient way for data sharing, which brings various benefits for both the society and individuals. But there exists a natural resistance for users to directly outsource the shared data to the cloud server since the data often contain valuable information. Thus, it is necessary to place cryptographically enhanced access control on the shared data. Identity-based encryption is a promising cryptographical primitive to build a practical data sharing system. However, access control is not static. That is, when some user’s authorization is expired, there should be a mechanism that can remove him/her from the system. Consequently, the revoked user cannot access both the previously and subsequently shared data. To this end, we propose a notion called revocable-storage identity-based encryption (RS-IBE), which can provide the forward/backward security of ciphertext by introducing the functionalities of user revocation and ciphertext update simultaneously. Furthermore, we present a concrete construction of RS-IBE, and prove its security in the defined security model. The performance comparisons indicate that the proposed RS-IBE scheme has advantages in terms of functionality and efficiency, and thus is feasible for a practical and cost-effective data-sharing system. Finally, we provide implementation results of the proposed scheme to demonstrate its practicability.

**EXISTING SYSTEM:**

* Boneh and Franklin first proposed a natural revocation way for IBE. They appended the current time period to the ciphertext, and non-revoked users periodically received private keys for each time period from the key authority.
* Boldyreva, Goyal and Kumar introduced a novel approach to achieve efficient revocation. They used a binary tree to manage identity such that their RIBE scheme reduces the complexity of key revocation to logarithmic (instead of linear) in the maximum number of system users.
* Subsequently, by using the aforementioned revocation technique, Libert and Vergnaud proposed an adaptively secure RIBE scheme based on a variant ofWater’s IBE scheme.
* Chen et al. constructed a RIBE scheme from lattices.

**DISADVANTAGES OF EXISTING SYSTEM:**

* Unfortunately, existing solution is not scalable, since it requires the key authority to perform linear work in the number of non-revoked users. In addition, a secure channel is essential for the key authority and non-revoked users to transmit new keys.
* However, existing scheme only achieves selective security.
* This kind of revocation method cannot resist the collusion of revoked users and malicious non-revoked users as malicious non-revoked users can share the update key with those revoked users.
* Furthermore, to update the ciphertext, the key authority in their scheme needs to maintain a table for each user to produce the re-encryption key for each time period, which significantly increases the key authority’s workload.

**PROPOSED SYSTEM:**

* It seems that the concept of revocable identity-based encryption (RIBE) might be a promising approach that fulfills the aforementioned security requirements for data sharing.
* RIBE features a mechanism that enables a sender to append the current time period to the ciphertext such that the receiver can decrypt the ciphertext only under the condition that he/she is not revoked at that time period.
* A RIBE-based data sharing system works as follows:
* Step 1: The data provider (e.g., David) first decides the users (e.g., Alice and Bob) who can share the data. Then, David encrypts the data under the identities Alice and Bob, and uploads the ciphertext of the shared data to the cloud server.
* Step 2: When either Alice or Bob wants to get the shared data, she or he can download and decrypt the corresponding ciphertext. However, for an unauthorized user and the cloud server, the plaintext of the shared data is not available.
* Step 3: In some cases, e.g., Alice’s authorization gets expired, David can download the ciphertext of the shared data, and then decrypt-then-re-encrypt the shared data such that Alice is prevented from accessing the plaintext of the shared data, and then upload the re-encrypted data to the cloud server again.

**ADVANTAGES OF PROPOSED SYSTEM:**

* We provide formal definitions for RS-IBE and its corresponding security model;
* We present a concrete construction of RS-IBE.
* The proposed scheme can provide confidentiality and backward/forward2 secrecy simultaneously
* We prove the security of the proposed scheme in the standard model, under the decisional ℓ-Bilinear Diffie-Hellman Exponent (ℓ-BDHE) assumption. In addition, the proposed scheme can withstand decryption key exposure
* The procedure of ciphertext update only needs *public information*. Note that no previous identity-based encryption schemes in the literature can provide this feature;
* The additional computation and storage complexity, which are brought in by the forward secrecy, is all upper bounded by O(log(T )2), where T is the total number of time periods.

**SYSTEM ARCHITECTURE:**



**MODULES:**

* System Construction Module
* Data Provider
* Cloud User
* Key Authority (Auditor)

**MODULES DESCSRIPTION:**

**System Construction Module**

In the first module, we develop the proposed system with the required entities for the evaluation of the proposed model. The data provider (e.g., David) first decides the users (e.g., Alice and Bob) who can share the data. Then, David encrypts the data under the identities Alice and Bob, and uploads the ciphertext of the shared data to the cloud server.

When either Alice or Bob wants to get the shared data, she or he can download and decrypt the corresponding ciphertext. However, for an unauthorized user and the cloud server, the plaintext of the shared data is not available.

**Data Provider**

In this module, we develop the Data Provider module. The data provider module is developed such that the new users will Signup initially and then Login for authentication. The data provider module provides the option of uploading the file to the Cloud Server. The process of File Uploading to the cloud Server is undergone with Identity-based encryption format. Data Provider will check the progress status of the file upload by him/her. Data Provider provided with the features of Revocation and Ciphertext update the file. Once after completion of the process, the Data Provider logouts the session.

**Cloud User**

In this module, we develop the Cloud User module. The Cloud user module is developed such that the new users will Signup initially and then Login for authentication. The Cloud user is provided with the option of file search. Then cloud user feature is added up for send the Request to Auditor for the File access. After getting decrypt key from the Auditor, he/she can access to the File. The cloud user is also enabled to download the File. After completion of the process, the user logout the session.

**Key Authority (Auditor)**

Auditor Will Login on the Auditor's page. He/she will check the pending requests of any of the above person. After accepting the request from the above person, he/she will generate master key for encrypt and Secret key for decrypt. After the complete process, the Auditor logout the session.

**SYSTEM REQUIREMENTS:**

**HARDWARE REQUIREMENTS:**

* System : Pentium Dual Core.
* Hard Disk : 120 GB.
* Monitor : 15’’ LED
* Input Devices : Keyboard, Mouse
* Ram : 1GB.

**SOFTWARE REQUIREMENTS:**

* Operating system : Windows 7.
* Coding Language : JAVA/J2EE
* Tool : Netbeans 7.2.1
* Database : MYSQL

**REFERENCE:**

Jianghong Wei, Wenfen Liu, Xuexian Hu, “Secure Data Sharing in Cloud Computing Using Revocable-Storage Identity-Based Encryption”, **IEEE Transactions on Cloud Computing 2016.**