Secure Cloud Data Deduplication with Efficient Re-Encryption

ABSTRACT

Data deduplication technique has been widely adopted by commercial cloud storage providers, which is both important and necessary in coping with the explosive growth of data. To further protect the security of users’ sensitive data in the outsourced storage mode, many secure data deduplication schemes have been designed and applied in various scenarios. Among these schemes, secure and efficient re-encryption for encrypted data deduplication attracted the attention of many scholars, and many solutions have been designed to support dynamic ownership management. In this paper, we focus on the re-encryption deduplication storage system and show that the recently designed lightweight rekeying-aware encrypted deduplication scheme (REED) is vulnerable to an attack which we call it stub-reserved attack. Furthermore, we propose a secure data deduplication scheme with efficient re-encryption based on the convergent all-or-nothing transform (CAONT) and randomly sampled bits from the Bloom filter. Due to the intrinsic property of one-way hash function, our scheme can resist the stub-reserved attack and guarantee the data privacy of data owners’ sensitive data. Moreover, instead of re-encrypting the entire package, data owners are only required to re-encrypt a small part of it through the CAONT, thereby effectively reducing the computation overhead of the system. Finally, security analysis and experimental results show that our scheme is secure and efficient in re-encryption.

**EXISTING SYSTEM**

In an existing system,the CE scheme is inherently vulnerable to the brute-force dictionary attacks. In order to solve this problem, Bellare *et al*. [35] proposed the DupLESS scheme, in which a user obtains the key from a dedicated keyserver via an oblivious PRF (OPRF) protocol. The OPRF mechanism is used to “blind” the fingerprint. Based on the RSA mechanism, the key server is configured with a system-wide public/private key pair. This enables the key server to return the MLE key without knowing the original fingerprint. The rate-limits are used in the key generation requests and can be efficient against the brute-force attacks. If the key-server is secure, the encryption key appears to be derived from a random space. Shin *et al*. [39] extended the predicate encryption scheme in data deduplication. However,

this scheme only supports the requirement of singleuser data deduplication. By introducing the additional tag checking mechanism, Bellare *et al*. proposed the randomized convergent encryption (RCE) scheme [38]. After decrypting the ciphertext, the user uses the plaintext to generate the tag and compare it with the corresponding tag. If tags are consistency, the user accepts the ciphertext; else, rejects it. Thus, the RCE scheme guarantees the integrity of users’ data. However, these schemes suffer from security flaws with respect to user revocation. If the revoked users keep the convergent key, they can access the plaintext without permission. Thus, the confidentiality of users’ sensitive data cannot be guaranteed.

To deal with the problem of efficient and reliable key management, Li *et al*. [40] proposed a secure data deduplication scheme by employing a security Ramp secret sharing scheme [41]. To realize dynamic updates in the deduplication, Wen *et al*. [36] proposed a session-key-based convergent key management scheme and convergent key sharing scheme. To deal with the problem of dynamic ownership changes of outsourced data, Hur *et al*. [21] used the group key to re-encrypt the ciphertext, which allows only the authorized cloud user to access the shared data. Chen *et al*. [42] proposed a block-level message-locked encryption (BL-MLE) scheme, which achieves file-level and block-level deduplication. To flexibly support data access control and revocation, Yan *et al*. [43] proposed a scheme to deduplicate encrypted data stored in the cloud. Based on static or dynamic decision trees, Jiang *et al*. introduced a new primitive called R-MLE2 and proposed a cloud data deduplication with randomized tag [23]. Based on the PAKE protocol, Liu *et al*. [44] proposed a secure data deduplication scheme, which supports the client-side encryption without an additional independent server. To address the problem of authorized data deduplication, Li *et al*. proposed several data deduplication schemes supporting authorized duplicate checking in a hybrid cloud architecture [45]. However, these schemes mainly use the method of re-encryption to solve the problem of user revocation. The traditional re-encryption scheme inevitably brings abundant computation overhead.

Recently, efficient re-encryption techniques are attracting widespread attention in the scientific community. To achieve efficient re-encryption and lightweight rekeying in data deduplication, Li *et al*. [25] proposed a rekeying-aware encrypted deduplication storage system. In this scheme, a data owner does not need to re-encrypt the entire package but only a small part of it, saving excessive computation overhead. In addition, the authors extended REED with ciphertext-policy attribute-based encryption [30] to control the access privileges to different data. However, a security weakness is found in the REED scheme.

**Disadvantages**

* An existing methodology doesn’t implement Bloom Filter-Based Location Selection Method.
* The system not implemented Ciphertext-Policy Attribute-Based Encryption Method.

Proposed System

In this paper, we further study the above problems of secure and efficient re-encryption for deduplication storage. Our contributions are three folds:

*•* We point out a security weakness of the enhanced encryption of REED scheme [25], [26]. That is, this scheme is vulnerable to the so-called stub-reserved attack proposed in this paper.

*•* We propose a location selection method based on Bloom filter and a secure data deduplication scheme with efficient re-encryption. By using the symmetric encryption and this new location selection method, the revoked cloud user cannot obtain the sensitive data from the data owner. Thus data privacy is ensured. Moreover, instead of re-encrypting the entire package, data owners are only required to re-encrypt a small part of it through the CAONT, thereby effectively reducing the computation overhead of the scheme.

*•* We provide security analysis and performance evaluation of our scheme, and the results show that our scheme is secure and efficient.

**Advantages**

* The proposed system proposes a secure cloud data deduplication scheme with efficient re-encryption. Our scheme is designed for enterprise or user groups in which multiple users want to outsource the data to a remote cloud service provider.
* The cloud service provider can conduct deduplication on ciphertexts and save abundant storage overhead. The system of our scheme contains three entities: cloud user, key server and cloud service provider (CSP).

**SYSTEM REQUIREMENTS**

➢ **H/W System Configuration:-**

➢ Processor - Pentium –IV

➢ RAM - 4 GB (min)

➢ Hard Disk - 20 GB

➢ Key Board - Standard Windows Keyboard

➢ Mouse - Two or Three Button Mouse

➢ Monitor - SVGA

**Software Requirements:**

* Operating System - Windows XP
* Coding Language - Java/J2EE(JSP,Servlet)
* Front End - J2EE
* Back End - MySQL