ENCRYPTED DATA MANAGEMENT WITH DEDUPLICATION IN CLOUD COMPUTING

TRUPTI RONGARE

Abstract: Cloud Computing offers many services resources over the Internet and providing them to users on demand. It main service is data storage, processing, and management in the Internet of Thing(IoT). Various cloud service providers (CSPs) offer huge volumes of storage to maintain and manage Internet data, which can include videos, photos, and personal records.

To preserve cloud data confidentiality and user privacy, cloud data are often stored in an encrypted form. But duplicated data that are encrypted under different encryption schemes could be stored in the cloud, which greatly decreases the utilization rate of storage resources, especially for big data. Several data deduplication schemes have recently been proposed. But most of them suffer from security weakness and lack of flexibility to support secure data access control. This paper proposes a scheme based on attribute-based encryption (ABE) to deduplicate encrypted data stored in the cloud while also supporting secure data access control. In this paper the survey based on analysis and implementation, results show the efficiency, effectiveness, and scalability of the survey for potential practical deployment.

Keywords: Cloud Computing, Encryption, Access control, Memory, Servers, Internet of Thing (IoT) , attribute-based encryption, data storage, deduplication management.

Introduction: The most important cloud service is data storage service. The cloud users upload personal data to a cloud service provider (CSP) and allow it to maintain these data. In existing research proposed to only out source encrypted data to the cloud in order to ensure data privacy. But in this the duplicated data in an encrypted form are stored by the same or different users, especially for shared data. This duplicated data wastes networking resources. For this solution of deduplication had proved to achieve high space and cost saving. But however, the existing solutions for deduplication suffers from brute-force attacks. They cannot flexibly support data access control and revocation. Most of the present solutions cannot ensure reliability, security and privacy with performance.Few existing schemes for clouddata access control support data deduplication simultaneously, and few can ensure flexibility andsecurity with sound performance for cloud data deduplication that data owners control directly.

Study propose a scheme based on attribute-based encryption (ABE) to deduplicate encrypted data stored in the cloud and support secure data access control at the same time and efficient

System Design : This scheme system having three types of entities: see fig f

1. CSP that offers a storage service and performs honestly on data storage and management to gain commercial profit but can't be fully trusted since it's curious about the contents of stored data;
2. Data owner that stores its data at the CSP (we assume there's only one data owner for one data*M*);
3. data holders (*ui*, *i* = 1, . . . , *n*) that are eligible data users and could save the same data as the data owner at the CSP.

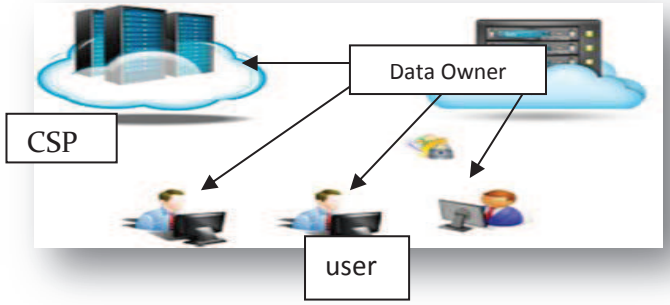

Figure 1. Three types of entities.

Assume that the system uses a data owner or holder device (such as a smart mobile device) to collect and pre-process data collected by IoT devices if they aren't capable of cryptographic operations or networking.

Assumptions also include the data's hash code $M(H(M))$ being applied as its indicator, which is used to check the duplication of data during data storage. It is assume the data holder signs the right hash code honestly for ownership verification at the CSP. This hash code is protected and can't be obtained by attackers. Second assume that the data owner has the highest priority for data storage management. A data holder should provide valid proof of ownership to request special treatment. The CSP, data owners, and data holders communicate with each other through asecure channel.

System Setup and Required Keys: In this scheme, there is management of keys of different authorized data users for deduplication by working with their identities (IDs) as valid attributes for accessing encrypted data stored at the CSP.

During system setup, every data owner or holder, CSP user *u*, maintains a public key *PKu*, which other users

employ to generate personalized secret attribute keys, and a secret key *SKu*, which is used in the decry-ption operation related to *PKu*. The data owner or holder uses the *SKu* to issue secret attribute keys to other users and to generate its own public key *PKIDu* of identity attribute *ID*. CSP user *u* generates *PKIDu* to encrypt a symmetric key *DEKu*, randomly selected for encrypting the dataof *u*, aiming to control data access and deduplication. The corresponding secret attribute keys for decrypting the cipher key encrypted by *PKIDu* are personalized for eligible data holders and issued by data owner *u*. To prevent collusion, every holder gets a different secret attribute key that only it can use. A secret attribute key of the attribute *ID*, issued for an eligible data holder *u′*by *u* is denoted as *SKID*(*u*,*u′*).Meanwhile, user *u* also generates a key pair *pku* and *sku* for Public-Key Cryptosystem(PKC), for example, signature generation and verification. The keys (*PKu*, *SKu*), (*pku*, *sku*) are bound to the unique identity of *u*, which can be a unique anonymous identifier. This binding is crucial for user identity verification. At system setup, *PKu* and*pku* are certified by an authorized third party as
Cert(*PKu*), *Cert*(*pku*) which the CSP and any CS Pusers can verify.

Algorithms: This scheme consists of several fundamental algorithms. This scheme adopt either cipher text policy ABE(CP-ABE) or key policy ABE (KP-ABE) to implement related algorithms.

1. **Initiate Node:** The Initiate Node algorithm takes asinput the node identifier *u*. It outputs several user credentials including (*PKu*, *SKu*) and (*pku*, *sku*). This process is conducted at user *u*.
2. **Create IDPK:** The Create IDPK algorithm is executed by *u* whenever *u* wants to control its data storage and access in the cloud. The algorithm checks the ID-related policies. If the result is positive, the algorithm outputs a public attribute key about the ID for user *u*, denoted *PKIDu*; otherwise, it outputs NULL.
3. **Issue IDSK:** The Issue IDSK algorithm checks whether*u′* with public key *PKu′*is eligible to hold the data. If it is, Issue IDSK outputs a secret attribute key *SKID*(*u*,*u′*) for user *u′*. Otherwise, it outputs NULL. This process is executed by *u* based on identity verification by checking that *Cert*(*PKu′*) is a valid certificate and the owner of *PKu′*is an eligible data holder. Data owner *u* sends *SKID*(*u*,*u′*) to *u′*through a secure channel or using the PKC. For example, data owner *u* would like a data holder with *ID = PKu′*to share its data storage. It encrypts its data encryption key *DEKu* with an access policy *AP*: *ID = PKu′*. The access policy can also contain multiple IDs or other control attributes when the data owner wants fine-grained control over the stored data in the CSP, for example, *AP*:*Vi*=1*n IDi*.
4. **Encrypt Key:** The Encrypt Key algorithm takes as input symmetric key *DEKu*, access policy *AP*, and public key *PKIDu* corresponding to the identity attribute occurring in *AP*. The algorithm encrypts *DEKu* with the policy *AP* and outputs the cipher key *CKu*. This process is conducted at *u* to support deduplication of data storage at the CSP.
5. **Decrypt Key:** The Decrypt Key algorithm takes as input cipher key *CKu* produced by the Encrypt Key algorithm; the access policy, *AP*, under which cipher key *CKu* was encrypted; *SKu′*; and *SKID*(*u*,*u′*). It decrypts *CKu* and outputs the corresponding plain key *DEKu* if the attributes are sufficient to satisfy *AP*; otherwise, it outputs NULL. This process is executed at *u′*if duplicated storage occur. It first checks the access policy *AP*, then conducts decryption to get *DEKu*.
6. **Encrypt:** The algorithm encrypts data *M* with *DEKu* and outputs cipher text *CTu*. This process is conducted at user *u* to protect its data with *DEKu*.
7. **Decrypt:** This algorithm is conducted at user *u′*or *u*to decrypt *CTu* with *DEKu* and output plain data *M*.

Scheme: Designing a deduplication scheme with ABE has several advantages. This scheme support digital rights management based on the data owner's expectations. Second, the scheme saves CSP storage space since it only stores one copy of the same data. Storing de duplication functional records could occupy some storage memory, but this cost is minimal compared to the cost of storing a large volume of duplicated data. Third, the proposed scheme can support many duplication instances and a huge volume of duplicated data. In this case, data holder *u′*only sends data package {*H*(*M*), *Sign*(*H*(*M*), *sku′*), *Cert*(*PKu′*),*Cert*(*pku′*)} without *CTu′* and *CKu′*for a duplication check before actually uploading the data. If duplication occurs, data holder *u′*can get *SKID*(*u*,*u′*) from
the data owner if it's eligible.

Data Deduplication: Figure 2. illustrates the data deduplication process at a CSP with the data owner's instruction and control. It is assume that user *u* is a data owner who saves data *M* at the CSP, using *DEKu* to protect the data, while user *u′*is a data holder who tries to save the same data at that CSP. First, each CSP user generates personal credentials and two key

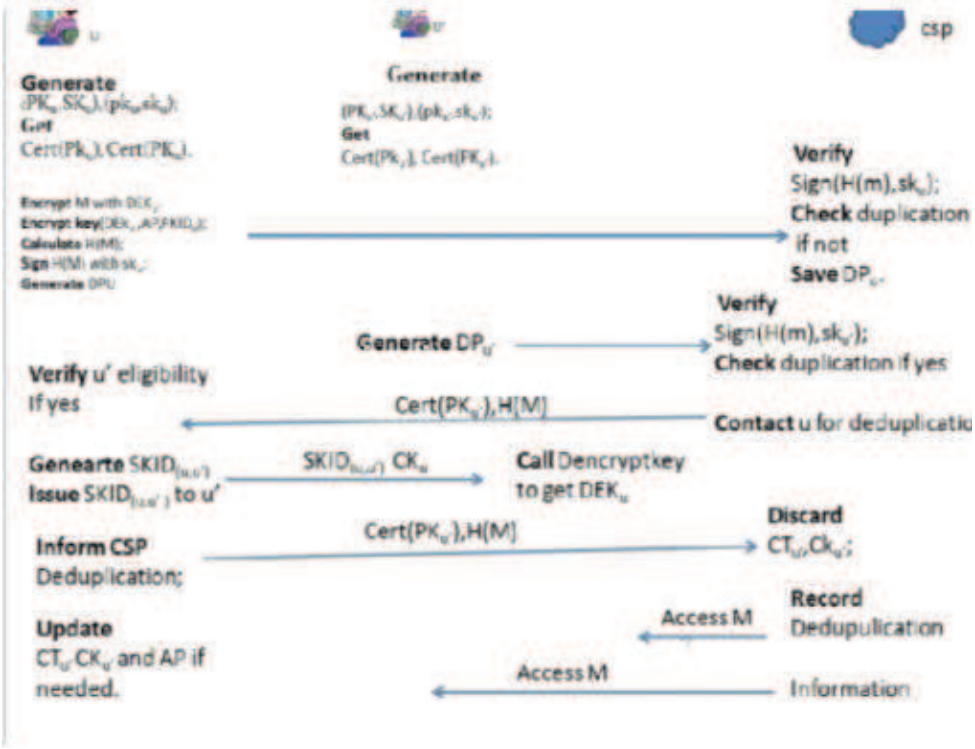

Fig 2 Data Deduplication Process

pairs (*PKu, SKu*) and (*pku, sku*) and gets the certificates of its public keys *Cert*(*PKu*) and *Cert*(*pku*).User *u* saves data *M* at the CSP. The user generates *CTu* = *Encrypt*(*DEKu*, *M*) and encrypts *DEKu* with *PKIDu* by calling *Encrypt Key*(*DEKu, AP, PKIDu*)to get *CKu*. It generates *PKIDu* according to *u*'sdata storage and access policy. User *u* calculates *H*(*M*) and signs it with *sku* as *Sign*(*H*(*M*), *sku*). Next, *u* sends the data package, *DPu* = {*CTu*, *CKu*, *H*(*M*),*Sign*(*H*(*M*), *sku*), *Cert*(*PKu*), *Cert*(*pku*)}, to the CSP. After receiving *DPu*, the CSP verifies *Cert*(*PKu*)and *Cert*(*pku*). If the verification is positive, the CS Puses verify *Sign*(*H*(*M*), *sku*) to check if duplicate data is saved by finding whether the same *H*(*M*) isin its storage. If the check is negative, the CSP saves *DPu*. If the check is positive and the pre stored data is from the same user, the CSP notifies that user. Ifa different user is storing the same data, the CSP performs deduplication. If data holder *u'*uploads the same data to the CSP, the CSP contacts data owner *u* by sending (*M*) and *Cert*(*PKu'*) for deduplication. User *u* verifies the eligibility of *u'*. If verification is positive, user *u* calls *Issue IDSK*(*ID*, *SKu*, *PKu'*) to generate *SKID*(*u,u'*) and issues *SKID*(*u,u'*) to *u'*to allow it to access *M*. Next, user *u* reports the successful data deduplication to the CSP. After receiving this notification, the CSP discards *CTu'* and *CKu'*and recodes the corresponding deduplication information in the system. In this step, the data owner can also update *DEKu*, upload new *CTu* and *CKu* to the CSP ,and send the newly encrypted *DEKu* (with ABE) to eligible data holders through the CSP or directly. At this moment, both u and u'can access data *M* stored at the CSP. User *u* uses *DEKu* directly, where as *u'*gets *DEKu* by calling *Decrypt Key*(*CKu*, *AP*, *SKu'*, *SKID*(*u,u'*)).

Data Deletion at the CSP: Data holder *u'*sends a deletion request to the CSP by providing *Cert*(*pku'*) and *H*(*M*). The CSP checks the request's validity, then removes the deduplication record of *u'*, and blocks its later access to *M*. The CSP further checks if the deduplication record is empty. If it is, the CSP deletes encrypted data *CTu*
and all related records. When data owner *u* sends a deletion request to the CSP with *Cert*(*pku*) and *H*(*M*), the CSP checks the request's validity, then removes the deduplication record of *u* and blocks its later access to data *M*. The CSP further checks whether the deduplication record is empty. If it is, it deletes encrypted data *CTu* and all related records. Otherwise, the CSP asks *u* to continuously manage deduplication. If *u* agrees, no action is performed at the CSP. If *u* doesn't agree, the CSP contacts data holder *u'*with the earliest data storage or data holder *u*ndesignated by *u* for later deduplication support. In this case, *u'*en crypts *DEKu*by calling *CKu'*= *Encrypt Key*(*DEKu*, *AP*, *PKIDu'*) and then performs as a delegate of the data owner to support deduplication by sending *CKu'*to CSP.

Data Owner Management: During data deletion, the CSP also asks the data owner to allow it to decide if the raw data needs to bereencrypted. For security purposes, the data owner could select a new *DEKu*,

reencrypt data *M*, update *CTu* and *CKu* at the CSP, and issue a new *CKu* to
existing eligible users for managing data deletion. Ift he real data owner uploads the data after the data holder, the CSP can save the data encrypted by the real data owner at the cloud and allow the data owner to manage and issue corresponding access keys toothier data holders.

Conclusion: Managing encrypted data with deduplication is significant in study for running a secure, dependable, and green cloud storage service, especially for big data processes. Future work includes efficient data ownership verification, scheme optimization with hardware acceleration at IoT devices for practical deployment, and development ofa flexible solution to support deduplication and data access controlled by either the data owner or its representative agent.

References

1. Z.C. Wen et al., "A Verifiable Data Deduplication Scheme in Cloud Computing Proc. Int'l Conf. Intelligent Networking and Collaborative Systems,2014, pp. 85–90.
2. J. Li et al., "A Hybrid Cloud Approach for Secure Authorized Deduplication," IEEE Trans. Parallel Distributed Systems, vol. 26, no. 5, 2015, pp. 1206–1216.
3. Suja A, Sivakumar S, Ramkumar P.S, Modified interleaved Buck Converter Implementation for Higher Step-Down Conversion Ratio; Engineering Sciences international Research Journal: ISSN 2320-4338 Volume 3 Issue 1 (2015), Pg 88-93
4. P. Meye et al., "A Secure Two-Phase Data Deduplication Scheme," Proc. IEEE 6th Int'l Symp. Cyberspace Safety and Security, IEEE 11th Int'l Conf. Embedded Software and Systems (HPCC, CSS,ICESS), 2014, pp. 802–809.
5. P. Puzio et al., "Clou Dedup: Secure Deduplication with Encrypted Data for Cloud Storage," Proc. IEEE 5th Int'l Conf. Cloud Computing Technology and Science, 2013, pp. 363–370.
6. Z. Yan, W. Ding, and H. Zhu, "Manage Encrypted Data Storage with Deduplication in Cloud," Proc. Int'l Conf. Algorithms and Architectures for Parallel Processing (ICA3PP), 2015, pp. 547–561.
7. Lalrinthara Pachuau, Zaithanzauva Pachuau, Comparison of Health Symptoms Faced By inhabitants Exposed to High and Low Mobile Phone tower Radiation; Engineering Sciences international Research Journal: ISSN 2320-4338 Volume 3 Issue 1 (2015), Pg 94-99

Trupti Rongare
Assoc. Professor, IT Department, KBP degree college Thane.