ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Scalable Data Sharing in Cloud Storage through Key-Aggregate Crypto System

Chekuri Amulya
PG Scholar,
Department of CSE,
Sri Chundi Ranganayakulu Engineering College,
Chilakaluripet, Guntur, AP, India.

E. Sambasiva Rao
Assistant Professor,
Department of CSE,
Sri Chundi Ranganayakulu Engineering College,
Chilakaluripet, Guntur, AP, India.

ABSTRACT:

Data sharing is an important functionality in cloud storage. In this paper, we show how to securely, efficiently, and flexiblyshare data with others in cloud storage. We describe new public-key cryptosystems that produce constant-size ciphertexts such thatefficient delegation of decryption rights for any set of ciphertexts are possible. The novelty is that one can aggregate any set of secretkeys and make them as compact as a single key, but encompassing the power of all the keys being aggregated. In other words, thesecret key holder can release a constant-size aggregate key for flexible choices of ciphertext set in cloud storage, but the otherencrypted files outside the set remain confidential. This compact aggregate key can be conveniently sent to others or be stored in asmart card with very limited secure storage. We provide formal security analysis of our schemes in the standard model. We alsodescribe other application of our schemes. In particular, our schemes give the first public-key patient-controlled encryption for flexiblehierarchy, which was yet to be known..

INTRODUCTION:

CLOUD storage is gaining popularity recently. In enterprisesettings, we see the rise in demand for dataoutsourcing, which assists in the strategic management ofcorporate data. It is also used as a core technology behindmany online services for personal applications. Nowadays,it is easy to apply for free accounts for email, photo album,file sharing and/or remote access, with storage size more, than 25 GB (or a few dollars for more than 1 TB). Togetherwith the current wireless technology, users can accessalmost all of their files and emails by a mobile phone in anycorner of the world.Considering data privacy, a traditional way to ensure it isto rely on the server to enforce the access control afterauthentication (e.g., [1]), which means any unexpectedprivilege escalation will expose all data.

In a shared-tenancycloud computing environment, things become even worse.Data from different clients can be hosted on separate virtualmachines (VMs) but reside on a single physical machine.Data in a target VM could be stolen by instantiating anotherVM coresident with the target one [2]. Regarding availabilityof files, there are a series of cryptographic schemeswhich go as far as allowing a third-party auditor to checkm the availability of files on behalf of the data owner withoutleaking anything about the data [3], or without compromisingthe data owners anonymity [4]. Likewise, cloud usersprobably will not hold the strong belief that the cloud serveris doing a good job in terms of confidentiality. A cryptographicsolution, for example, [5], with proven securityrelied on number-theoretic assumptions is more desirable,whenever the user is not perfectly happy with trusting thesecurity of the VM or the honesty of the technical staff.These users are motivated to encrypt their data with theirown keys before uploading them to the server.Data sharing is an important functionality in cloudstorage. For example, bloggers can let their friends view asubset of their private pictures; an enterprise may grant heremployees access to a portion of sensitive data. Thechallenging problem is how to effectively share encrypteddata. Of course users can download the encrypted datafrom the storage, decrypt them, then send them to others forsharing, but it loses the value of cloud storage. Users shouldbe able to delegate the access rights of the sharing data toothers so that they can access these data from the serverdirectly. However, finding an efficient and secure way toshare partial data in cloud storage is not trivial. Below wewill take Dropbox1 as an example for illustration.Our ContributionsIn modern cryptography, a fundamental problem we oftenstudy is about leveraging the secrecy of a small piece ofknowledge into the ability to perform cryptographicfunctions (e.g., encryption, authentication) multiple times.In this paper, we study how to make a decryption key morepowerful in the sense that it allows decryption of multipleciphertexts, without increasing its size. Specifically, ourproblem statement is

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

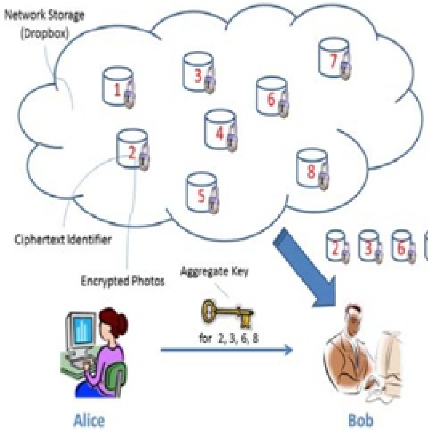


Fig. 1. Alice shares files with identifiers 2, 3, 6, and 8 with Bob bysending him a single aggregate key.

ciphertext called class. That means the ciphertexts arefurther categorized into different classes. The key ownerholds a master-secret called master-secret key, which can beused to extract secret keys for different classes. Moreimportantly, the extracted key have can be an aggregate keywhich is as compact as a secret key for a single class, butaggregates the power of many such keys, i.e., the decryptionpower for any subset of ciphertext classes.With our solution, Alice can simply send Bob a singleaggregate key via a secure e-mail.

Bob can downloadthe encrypted photos from Alice's Dropbox space and thenuse this aggregate key to decrypt these encrypted photos.The scenario is depicted in Fig. 1.The sizes of ciphertext, public-key, master-secret key,and aggregate key in our KAC schemes are all of constantsize. The public system parameter has size linear in thenumber of ciphertext classes, but only a small part of it isneeded each time and it can be fetched on demand fromlarge (but nonconfidential) cloud storage.

Previous results may achieve a similar property featuringa constant-size decryption key, but the classes need toconform to some predefined hierarchical relationship. Ourwork is flexible in the sense that this constraint is eliminated, that is, no special relation is required betweenthe classes. The detail and other related works can be foundin Section 3.We propose several concrete KAC schemes with differentsecurity levels and extensions in this paper. All constructionscan be proven secure in the standard model. To thebest of our knowledge, our aggregation mechanism2 in KAChas not been investigated.

2 KEY-AGGREGATE ENCRYPTION:

We first give the framework and definition for keyaggregateencryption. Then we describe how to use KACin a scenario of its application in cloud storage.

Framework:

A key-aggregate encryption scheme consists of fivepolynomial-time algorithms as follows.The data owner establishes the public system parametervia Setup and generates a public/master-secret3 key pairvia KeyGen. Messages can be encrypted via Encrypt byanyone who also decides what ciphertext class is associatedwith the plaintext message to be encrypted. Thedata owner can use the master-secret to generate anaggregate decryption key for a set of ciphertext classes viaExtract. The generated keys can be passed to delegates securely (via secure e-mails or secure devices) Finally, anyuser with an aggregate key can decrypt any ciphertext provided that the ciphertext's class is contained in theaggregate key via Decrypt.

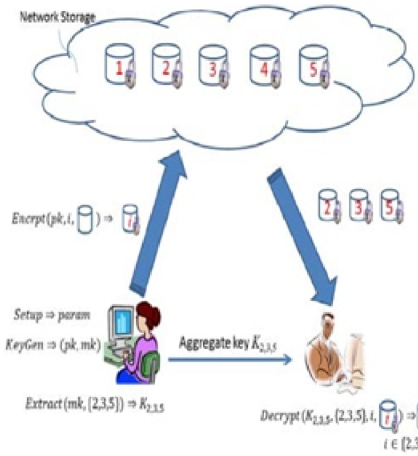
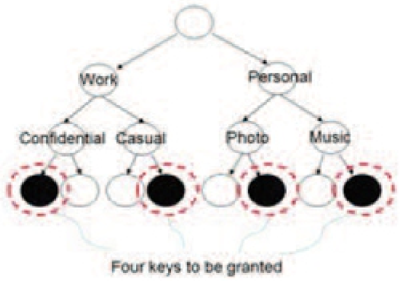


Fig. 2. Using KAC for data sharing in cloud storage.

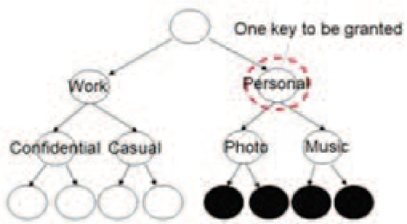
We take the tree structure as an example. Alice can firstclassify the ciphertext classes according to their subjects likeFig. 3. Each node in the tree represents a secret key, whilethe leaf nodes represents the keys for individual ciphertextclasses. Filled circles represent the keys for the classes to bedelegated and circles circumvented by dotted lines representthe keys to be granted. Note that every key of thenonleaf node can derive the keys of its descendant nodes.In Fig. 3a, if Alice wants to share all the files in the"personal" category, she only needs to grant the key for thenode "personal," which automatically grants the delegate the keys of all the descendant nodes ("photo," "music").

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Thisis the ideal case, where most classes to be shared belong tothe same branch and thus a parent key of them is sufficient.However, it is still difficult for general cases. As shown inFig. 3b, if Alice shares her demo music at work(“work”! “casual”! “demo” and “work”! “confidential”! “demo”) with a colleague who also has the rights to seesome of her personal data, what she can do is to give morekeys, which leads to an increase in the total key size. One canm see that this approach is not flexible when the classificationsare more complex and she wants to share different sets of filesto different people. For this delegatee in our example,



(a)



(b)

Fig. 3. Compact key is not always possible for a fixed hierarchy.

Compact Key in Identity-Based Encryption (IBE)IBE (e.g., [20], [21], [22]) is a type of public-key encryption inwhich the public-key of a user can be set as an identitystringof the user (e.g., an email address). There is a trustedparty called private key generator in IBE which holds amaster-secret key and issues a secret key to each user withrespect to the user identity. The encryptor can take thepublic parameter and a user identity to encrypt a message. The recipient can decrypt this ciphertext by his secret key.Guo et al. [23], [9] tried to build IBE with key aggregation.One of their schemes [23] assumes random oracles butanother [9] does not. . In their schemes, key aggregation isconstrained in the sense that all keys to be aggregated mustcome from different “identity divisions.” While there are anexponential number of identities and thus secret keys, only apolynomial number of them can be aggregated. Mostimportantly, their key-aggregation [23], [9] comes at theexpense of OðnÞ sizes for both ciphertexts and the publicparameter, where n is the number of secret keys which can beaggregated into a constant size one. This greatly increases thecosts of storing and transmitting ciphertexts, which isimpractical in many situations such as shared cloud storage.As we mentioned, our schemes feature constant ciphertextsize, and their security holds in the standard model.In fuzzy IBE [21], one single compact secret key candecrypt ciphertexts encrypted under many identities whichare close in a certain metric space, but not for an arbitraryset of identities and, therefore, it does not match with ouridea of key aggregation.

3.4 OTHER ENCRYPTION SCHEMES:

Attribute-based encryption (ABE) [10], [24] allows eachciphertext to be associated with an attribute, and themaster-secret key holder can extract a secret key for apolicy of these attributes so that a ciphertext can bedecrypted by this key if its associated attribute conformsto the policy. For example, with the secret key for the policyð2 _ 3 _ 6 _ 8Þ, one can decrypt ciphertext tagged with class2, 3, 6, or 8. However, the major concern in ABE is collusionresistancebut not the compactness of secret keys. Indeed,the size of the key often increases linearly with the numberof attributes it encompasses, or the ciphertext-size is notconstant (e.g.,[25]).

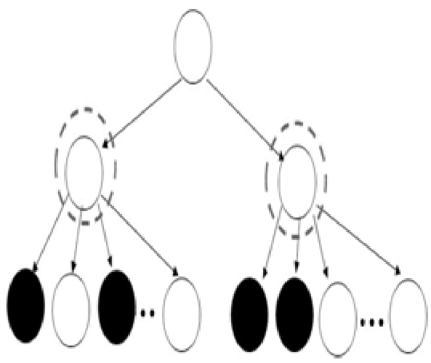


Fig. 4. Key assignment in our approach.

PERFORMANCE:

For encryption, the value ^eðg1; gnÞ can be precomputed andput in the system parameter. On the other hand, we can seethat decryption only takes two pairings while only one ofthem involves the aggregate key. That means we only needone pairing computation within the security chip storingthe (secret) aggregate key.

ISSN No: 2348-4845
International Journal & Magazine of Engineering,
Technology, Management and Research
A Peer Reviewed Open Access International Journal

It is fast to compute a pairingnowadays, even in resource-constrained devices. Efficientsoftware implementations exist even for sensor nodesDiscussionsThe "magic" of getting constant-size aggregate key andconstant-size ciphertext simultaneously comes from thelinear-size system parameter. Our motivation is to reducethe secure storage and this is a tradeoff between two kinds ofstorage. The parameter can be placed in nonconfidentiallocal storage or in a cache provided by the service company.

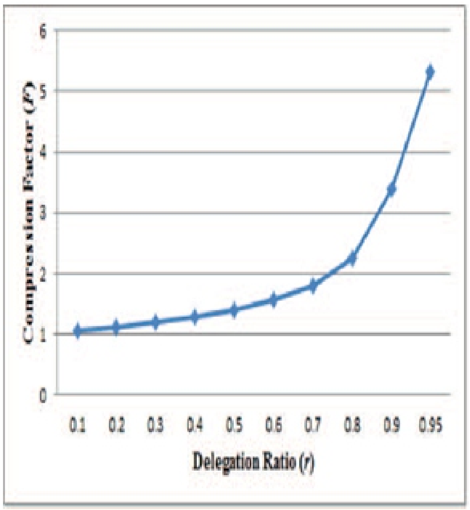
They can also be fetched on demand, as not all of them arerequired in all occasions.The system parameter can also be generated by a trustedparty, shared between all users and even hard-coded to theuser program (and can be updated via "patches"). In thiscase, while the users need to trust the parameter-generatorfor securely erasing any ephemeral values used, the accesscontrol is still ensured by a cryptographic mean instead ofrelying on some server to restrict the accesses honestly.

PERFORMANCE ANALYSIS

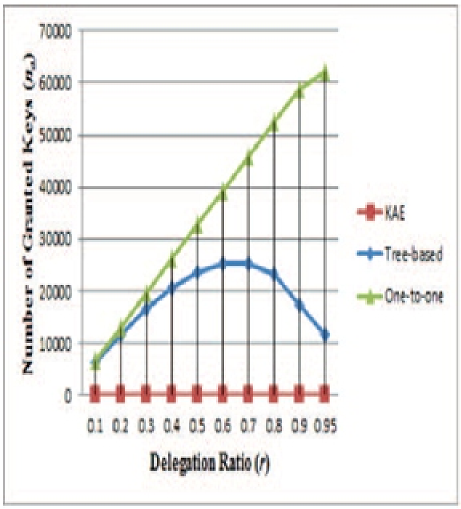
5.1 Compression Factors:

For a concrete comparison, we investigate the spacerequirements of the tree-based key assignment approachwe described in Section 3.1. This is used in the complete-subtree scheme, which is a representative solution to the-broadcast encryption problem following the well-known-subset-cover framework [33]. It employs a static logical keyhierarchy, which is materialized with a full binary key treeof height h (equals to 3 in Fig. 3), and thus can support up to2h ciphertext classes, a selected part of which is intended foran authorized delegatee.

In an ideal case as depicted in Fig. 3a, the delegatee canbe granted the access to 2hs classes with the possession ofonly one key, where hs is the height of a certain subtree(e.g., hs ¼ 2 in Fig. 3a). On the other hand, to decryptciphertexts of a set of classes, sometimes the delegatee mayhave to hold a large number of keys, as depicted in Fig. 3b.Therefore, we are interested in na, the number of symmetrickeysto be assigned in this hierarchical key approach, in anaverage sense.



(a)



(b)

Fig. 5. (a) Compression achieved by the tree-based approach fordelegating different ratio of the classes. (b) Number of granted keys (na)required for different approaches in the case of 65,536 classes of data.

CONCLUSION AND FUTURE WORK:

How to protect users' data privacy is a central question ofcloud storage. With more mathematical tools, cryptographicschemes are getting more versatile and often involve multiplekeys for a single application. In this paper, we considerhow to "compress" secret keys in public-key cryptosystemswhich support delegation of secret keys for differentciphertext classes in cloud storage.

No matter which oneamong the power set of classes, the delegatee can always getan aggregate key of constant size. Our approach is moreflexible than hierarchical key assignment which can onlysave spaces if all key-holders share a similar set of privileges.A limitation in our work is the predefined bound of thenumber of maximum ciphertext classes. In cloud storage,the number of ciphertexts usually grows rapidly. So wehave to reserve enough ciphertext classes for the futureextension. Otherwise, we need to expand the public-key aswe described in Section 4.2.Although the parameter can be downloaded withciphertexts, it would be better if its size is independent ofthe maximum number of ciphertext classes. On the otherhand, when one carries the delegated keys around in amobile device without using special trusted hardware, thekey is prompt to leakage, designing a leakage-resilientcryptosystem [22], [34] yet allows efficient and flexible keydelegation is also an interesting direction.

REFERENCES:

[1] S.S.M. Chow, Y.J. He, L.C.K. Hui, and S.-M. Yiu, “SPICE – Simplem Privacy-Preserving Identity-Management for Cloud Environment,”Proc. 10th Int’l Conf. Applied Cryptography and NetworkSecurity (ACNS), vol. 7341, pp. 526 543, 2012.

[2] L. Hardesty, Secure Computers Aren’t so Secure. MIT press, http://www.physorg.com/news176107396.html, 2009.

[3] C. Wang, S.S.M. Chow, Q. Wang, K. Ren, and W. Lou, “Privacy-Preserving Public Auditing for Secure Cloud Storage,” IEEE Trans.Computers, vol. 62, no. 2, pp. 362-375, Feb. 2013.

[4] B. Wang, S.S.M. Chow, M. Li, and H. Li, “Storing Shared Data onthe Cloud via Security-Mediator,” Proc. IEEE 33rd Int’l Conf.Distributed Computing Systems (ICDCS), 2013.

[5] S.S.M. Chow, C.-K. Chu, X. Huang, J. Zhou, and R.H. Deng,“Dynamic Secure Cloud Storage with Provenance,” Cryptographyand Security, pp. 442-464, Springer, 2012.

[6] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, “Aggregate andVerifiably Encrypted Signatures from Bilinear Maps,” Proc. 22ndInt’l Conf. Theory and Applications of Cryptographic Techniques(EUROCRYPT ’03), pp. 416-432, 2003.

[7] M.J. Atallah, M. Blanton, N. Fazio, and K.B. Frikken, “Dynamicand Efficient Key Management for Access Hierarchies,” ACMTrans. Information and System Security, vol. 12, no. 3, pp. 18:1-18:43,2009.

[8] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, “PatientControlled Encryption: Ensuring Privacy of Electronic MedicalRecords,” Proc. ACM Workshop Cloud Computing Security(CCSW ’09), pp. 103-114, 2009.

[9] F. Guo, Y. Mu, Z. Chen, and L. Xu, “Multi-Identity Single-KeymDecryption without Random Oracles,” Proc. Information Security mand Cryptology (Inscrypt ’07), vol. 4990, pp. 384-398, 2007.

[10] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-BasedEncryption for Fine-Grained Access Control of Encrypted Data,”Proc. 13th ACM Conf. Computer and Comm. Security (CCS ’06),pp. 89-98, 2006.

[11] S.G. Akl and P.D. Taylor, “Cryptographic Solution to a Problem ofAccess Control in a Hierarchy,” ACM Trans. Computer Systems,vol. 1, no. 3, pp. 239-248, 1983.

[12] G.C. Chick and S.E. Tavares, “Flexible Access Control with MasterKeys,” Proc. Advances in Cryptology (CRYPTO ’89), vol. 435,pp. 316-322, 1989.