Key-Factor Cryptosystem for Data Sharing in Clouds

N.Chandana
M.Tech Student,
Department of CSE,
Sree Rama institute of Technology and Science,
Kuppenakuntla, Penuballi, Khammam,TS India.

N.Naveen
Assistant Professor,
Department of CSE,
Sree Rama institute of Technology and Science,
Kuppenakuntla, Penuballi, Khammam,TS India.

ABSTRACT:

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

Index Terms:

Cloud storage, data sharing, key-aggregate encryption, patient-controlled encryption.

INTRODUCTION:

CLOUD storage is gaining popularity recently. In enterprise settings, we see the rise in demand for data outsourcing, which assists in the strategic management of corporate data. It is also used as a core technology behind many 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). Together with the current wireless technology, users can access almost all of their files and emails by a mobile phone in any corner of the world.Considering data privacy, a traditional way to ensure it is to rely on the server to enforce the access control after authentication (e.g., [1]), which means any unexpected privilege escalation will expose all data. In a shared-tenancy cloud computing environment, things become even worse.

Data from different clients can be hosted on separate virtual machines (VMs) but reside on a single physical machine. Data in a target VM could be stolen by instantiating another VM coresident with the target one. Regarding availability of files, there are a series of cryptographic schemes which go as far as allowing a third-party auditor to check the availability of files on behalf of the data owner without leaking anything about the data [3], or without compromising the data owners anonymity .

Likewise, cloud users probably will not hold the strong belief that the cloud server is doing a good job in terms of confidentiality. A cryptographic solution, for example, [5], with proven security relied on number theoretic assumptions is more desirable, whenever the user is not perfectly happy with trusting the security of the VM or the honesty of the technical staff. These users are motivated to encrypt their data with their own keys before uploading them to the server.
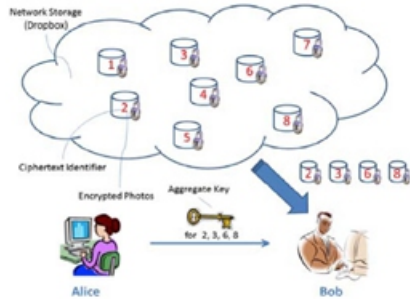
Volume No: 1 (2015), Issue No: 3 (August)
www. IJRACSE.com

August 2015
Page 41

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

Existing System:

We first give the framework and definition for key aggregate encryption. Then we describe how to use KAC in a scenario of its application in cloud storage. A key-aggregate encryption scheme consists of five polynomial-time algorithms as follows. The data owner establishes the public system parameter via Setup and generates a public/master-secret3 key pair via KeyGen. Messages can be encrypted via Encrypt by anyone who also decides what ciphertext class is associated with the plaintext message to be encrypted. The data owner can use the master-secret to generate an aggregate decryption key for a set of ciphertext classes via Extract. The generated keys can be passed to delegates securely (via secure e-mails or secure devices) Finally, any user with an aggregate key can decrypt any ciphertext provided that the ciphertext's class is contained in the aggregate key via Decrypt.

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

Proposed System:

This section we compare our basic KAC scheme with other possible solutions on sharing in secure cloud storage. We summarize our comparisons in Table We take the tree structure as an example. Alice can first classify the ciphertext classes according to their subjects like Fig. 3. Each node in the tree represents a secret key, while the leaf nodes represents the keys for individual ciphertext classes.

Filled circles represent the keys for the classes to be delegated and circles circumvented by dotted lines represent the keys to be granted. Note that every key of the non leaf 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 the node "personal," which automatically grants the delegate the keys of all the descendant nodes ("photo," "music"). This is the ideal case, where most classes to be shared belong to the same branch and thus a parent key of them is sufficient.
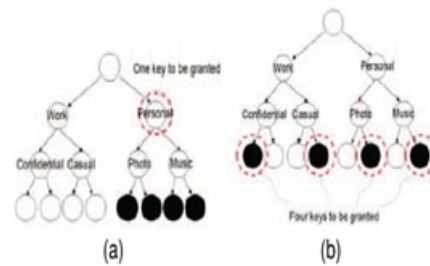

Fig. 3. Compact key is not always possible for a fixed hierarchy.

PERFORMANCE ANALYSIS:
Compression Factors:

For a concrete comparison, we investigate the space requirements of the tree-based key assignment approach we 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 key hierarchy, which is materialized with a full binary key tree of height h (equals to 3 in Fig. 3), and thus can support up to 2h ciphertext classes, a selected part of which is intended for an authorized delegatee.

In an ideal case as depicted in Fig. 3a, the delegatee can be granted the access to 2hs classes with the possession of only one key, where hs is the height of a certain subtree (e.g., hs ¼ 2 in Fig. 3a). On the other hand, to decrypt ciphertexts of a set of classes, sometimes the delegatee may have to hold a large number of keys, as depicted in Fig. 3b. Therefore, we are interested in na, the number of symmetrickeys to be assigned in this hierarchical key approach, in an average sense.
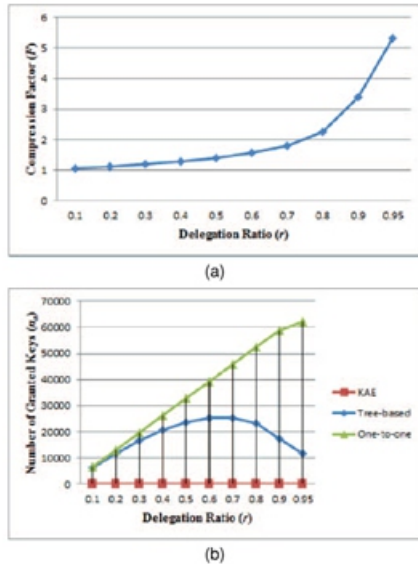
Fig. 5. (a) Compression achieved by the tree-based approach for delegating different ratio of the classes.(b) Number of granted keys (na) required for different approaches in the case of 65,536 classes of data.

NEW PATIENT-CONTROLLED ENCRYPTION (PCE):

Motivated by the nationwide effort to computerize America's medical records, the concept of patient-controlled encryption has been studied [8]. In PCE, the health record is decomposed into a hierarchical representation based on the use of different ontologies, and patients are the parties who generate and store secret keys. When there is a need for a healthcare personnel to access part of the record, a patient will release the secret key for the concerned part of the record. In the work of Benaloh et al. [8], three solutions have been provided, which are symmetric-key PCE for fixed hierarchy (the "folklore" tree-based method in Section 3.1), public-key PCE for fixed hierarchy (the IBE analog of the folklore method, as mentioned in Section 3.1), and RSAbased symmetric-key PCE for "flexible hierarchy" (which is the "set membership" access policy as we explained).

CONCLUSION :

How to protect users' data privacy is a central question of cloud storage. With more mathematical tools, cryptographic schemes are getting more versatile and often involve multiple keys for a single application. In this paper, we consider how to "compress" secret keys in public-key cryptosystems which support delegation of secret keys for different ciphertext classes in cloud storage.

No matter which one among the power set of classes, the delegatee can always get an aggregate key of constant size. Our approach is more flexible than hierarchical key assignment which can only save spaces if all key-holders share a similar set of privileges. A limitation in our work is the predefined bound of the number of maximum ciphertext classes. In cloud storage, the number of ciphertexts usually grows rapidly. So we have to reserve enough ciphertext classes for the future extension.

ACKNOWLEDGMENTS:

REFERENCES:

[1] S.S.M. Chow, Y.J. He, L.C.K. Hui, and S.-M. Yiu, "SPICE - SimplePrivacy-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.

Volume No: 1 (2015), Issue No: 3 (August)
www. IJRACSE.com

August 2015
Page 43

[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-KeyDecryption without Random Oracles," Proc. Information Securityand 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.

[13] W.-G. Tzeng, "A Time-Bound Cryptographic Key AssignmentScheme for Access Control in a Hierarchy," IEEE Trans. Knowledgeand Data Eng., vol. 14, no. 1, pp. 182-188, Jan./Feb. 2002.

[14] G. Ateniese, A.D. Santis, A.L. Ferrara, and B. Masucci, "Provably-Secure Time-Bound Hierarchical Key Assignment Schemes,"J. Cryptology, vol. 25, no. 2, pp. 243-270, 2012.

[15] R.S. Sandhu, "Cryptographic Implementation of a Tree Hierarchyfor Access Control," Information Processing Letters, vol. 27, no. 2,pp. 95-98, 1988.

[16] Y. Sun and K.J.R. Liu, "Scalable Hierarchical Access Control inSecure Group Communications," Proc. IEEE INFOCOM '04, 2004.

[17] Q. Zhang and Y. Wang, "A Centralized Key Management Schemefor Hierarchical Access Control," Proc. IEEE Global Telecomm. Conf.(GLOBECOM '04), pp. 2067-2071, 2004.

[18] J. Benaloh, "Key Compression and Its Application to DigitalFingerprinting," technical report, Microsoft Research, 2009.

[19] B. Alomair and R. Poovendran, "Information Theoretically SecureEncryption with Almost Free Authentication," J. UniversalComputer Science, vol. 15, no. 15, pp. 2937-2956, 2009.

[20] D. Boneh and M.K. Franklin, "Identity-Based Encryption from theWeil Pairing," Proc. Advances in Cryptology (CRYPTO '01),vol. 2139, pp. 213-229, 2001.

[21] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," Proc.22nd Int'l Conf. Theory and Applications of Cryptographic Techniques(EUROCRYPT '05), vol. 3494, pp. 457-473, 2005.

[22] S.S.M. Chow, Y. Dodis, Y. Rouselakis, and B. Waters, "PracticalLeakage-Resilient Identity-Based Encryption from Simple Assumptions,"Proc. ACM Conf. Computer and Comm. Security,pp. 152-161, 2010.

[23] F. Guo, Y. Mu, and Z. Chen, "Identity-Based Encryption: How toDecrypt Multiple Ciphertexts Using a Single Decryption Key,"Proc. Pairing-Based Cryptography Conf. (Pairing '07), vol. 4575,pp. 392-406, 2007.

[24] M. Chase and S.S.M. Chow, "Improving Privacy and Security inMulti-Authority Attribute-Based Encryption," Proc. ACM Conf.Computer and Comm. Security, pp. 121-130. 2009,

[25] T. Okamoto and K. Takashima, "Achieving Short Ciphertexts orShort Secret-Keys for Adaptively Secure General Inner-ProductEncryption," Proc. 10th Int'l Conf. Cryptology and Network Security(CANS '11), pp. 138-159, 2011.

Author's:

N.Chandana is a student of Sree Rama Institute of Technology & Science, Kuppenakuntla,Penuballi, Khammam, TS,India.Presently she is Pursuing her M.Tech (CSE) from this collegeHis area of interests includes Information Security, Cloud Computing, Data Communication & Networks.

Mr. N.Naveen is an efficient teacher, received M.Tech from JNTU Hyderabad is working as an Assistant Professor in Department of C.S.E, Sree Rama Institute of Technology & Science, Kuppenakuntla, Penuballi, Khammam, AP,India. He has published many papers in both National & International Journals. His area of Interest includes Data Communications & Networks, Database Management Systems, Computer Organization, C Programming and other advances in Computer Applications.