www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 4 April 2015, Page No. 11602-11608

Secured Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage

N.R RejinPaul[1] ,Alagu Lakshmi M.S[2] , Sai Rasmitha D[3]
nrrejinpaul@gmail.com[1], banulakshmi23@gmail.com[2], dsrasmitha@gmail.com[3]
Assistant Professor[1], UG student [2&3]
Computer science and engineering
Velammal Institute of Technology

***Abstract*——** **Cloud Computing has been envisioned as the next generation architecture of IT enterprise, due to its long list of unprecedented advantages in the IT history. Data sharing is an important functionality in cloud storage. In this paper, we show how to securely, efficiently, and flexibl yshare data with others in cloud storage. public-key cryptosystems that produce constant-size ciphertexts such that efficient delegation of decryption rights for any set of ciphertexts are possible .secret key holder can release a constant-size aggregate key for flexible choices of file formats in cloud storage, but the other encrypted files outside the set remain confidential. This compact aggregate key can be conveniently sent to others with very limited secure storage.**

***Index Terms*— Cloud storage, data sharing, key-aggregate encryption.**

Introduction

Cloud storage is gaining popularity recently. In enterprise settings, 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 morethan 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 virtual

machines (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 schemes which go as far as allowing a third-party auditor to check secret keys involved. Naturally, there are two extreme ways

I. User 1 encrypts all files with a single encryption keyand gives user2 the corresponding secret key directly.. User1 encrypts files with distinct keys and sends user 2 the corresponding secret keys.Obviously, the first method is inadequate since all unchosendata may be also leaked to user2 . For the second method,there are practical concerns on efficiency. The number ofsuch keys is as many as the number of the shared photos,

II. say, a hundred. Transferring these secret keys inherentlyrequires a secure channel, and storing these keys requiresrather expensive secure storage. The costs and complexitiesinvolved generally increase with the number of the decryption keys to be shared. In short, it is very heavyand costly to do that.Encryption keys also come with two flavors—symmetrickey or asymmetric (public) key. Using symmetric encryption,when Alice wants the data to be

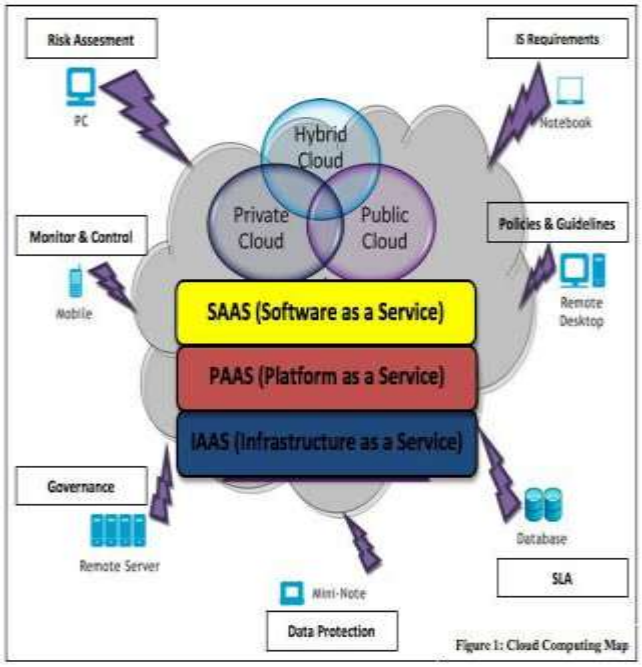


Figure 1: Cloud Computing Map

originated from athird party, she has to give the encryptor her secret key;obviously, this is not always desirable. By contrast, theencryption key and decryption key are different in publickey encryption. The use of public-key encryption givesmore flexibility for our applications. For example, inenterprise settings, every employee can upload encrypteddata on the cloud storage server without the knowledge ofthe company's master-secret key.Therefore, the best solution for the above problem is thatAlice encrypts files with distinct public-keys, but only sendsBob a single (constant-size) decryption key. Since thedecryption key should be sent via a secure channel andkept secret, small key size is always desirable. For example,we cannot expect large storage for decryption keys in theresource-constraint devices like smart phones, smart cards,or wireless sensor nodes. Especially, these secret keys areusually stored in the tamper-proof memory, which isrelatively expensive. The present research efforts mainlycommunication requirements(such as bandwidth, rounds of communication) likeaggregate signature However, not much has been doneabout the key itself .

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 pair via 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 delegateessecurely (via secure e-mails or secure devices) Finally, anyuser with an aggregate key can decrypt any ciphertextprovided that the ciphertext's class is contained in theaggregate key via Decrypt.. Setup executed by the data owner to setup anaccount on an untrusted server. On input a securitylevel parameter 1_ and the number of ciphertextclasses n (i.e., class index should be an integerbounded by 1 and n), it outputs the public systemparameter param, which is omitted from the inputof the other algorithms for brevity.. KeyGen: executed by the data owner to randomlygenerate a public/master-secret key pair . Encrypt; i executed by anyone who wants toencrypt data. On input a public-key pk, an index idenoting the ciphertext class, and a message m, itoutputs a ciphertext C.. Extract executed by the data owner fordelegating the decrypting power for a certain set ofciphertext classes to a delegatee. On input themaster-secret key msk and a set S of indicescorresponding to

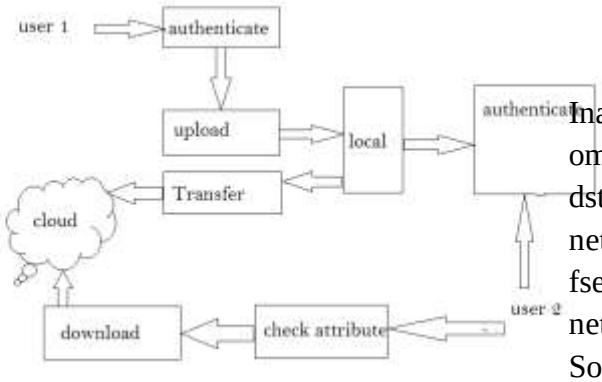
different classes, it outputs theaggregate key for set S denoted by KS.. Decrypt they are executed by a delegatee whoreceived an aggregate key KS generated by Extract.On input KS, the set S, an index i denoting the ciphertext class the ciphertext C belongs to, and C, itoutputs the decrypted result .

Sharing Encrypted Data

A canonical application of KAC is data sharing. The keyaggregation property is especially useful when we expectthe delegation to be efficient and flexible. The schemesenable a content provider to share her data in a confidentialand selective way, with a fixed and small ciphertextexpansion, by distributing to each authorized user a singleand small aggregate key.Here, we describe the main idea of data sharing in cloudstorage using KAC, illustrated in Suppose Alicewants to share her data m1;m2; on the server. Shefirst performs Setup nÞ to get param and execute KeyGento get the public/master-secret key pair Thesystem parameter param and public-key pk can be madepublic and master-secret key msk should be kept secret byAlice. Anyone (including user 1 herself) can then encrypteach mi by Encrypt;. The encrypted data areuploaded to the server.With param and pk, people who cooperate with Alicecan update Alice's data on the server. Once Alice is willingto share a set S of her data with a friend Bob, she cancompute the aggregate key KS for user 2 by performing Extract; Since KS is just a constant-size key, it iseasy to be sent to user 2 via a secure e-mail.After obtaining the aggregate key, user 2 can download thedata he is authorized to access. That is, for each i, user 2 downloads Ci (and some needed values in param) from theserver. With the aggregate key KS, user 2 can decrypt each Ciby Decrypt for each i.

Compact Symmetric-Key Encryption

Motivated by the same problem of supporting flexiblehierarchy in decryption power delegation (but in symmetric-key setting), enaloh et al. [8] presented an encryptionscheme which is riginally proposed for conciselytransmitting large number of keys nbroadcast scenario.The construction is simple and we briefly review itskey derivation process here for a concrete description ofwhat are the desirable properties we want to achieve. Thederivation of the key for a set of classes (which is a subset ofall possible ciphertext classes) is a follows: A composite modulus N ¼ p _ q is chosen where p and q are two largerandom primes. A master-secret key Y is chosen at randomfrom ZZ_N. Each class is associated with a distinct prime.All these prime numbers can be put in the public systemparameter.5 A constant-size key for set S0 can be generateds a concrete example, a key for classes represented bycan be generated as, from which each ofcan easily be derived (whileproviding no information about keys for any other class,say, e4). This approach achieves similar properties andperformances as our schemes. However, it is designed forthe symmetric-key setting instead. The encryptor needs toget the corresponding secret keys to encrypt data, which isnot suitable for many applications. Since their method isused to generate a secret value rather than a pair of public/secret keys, it is unclear how to apply this idea for publickeyencryption scheme.Finally, we note that there are schemes which try to reducethe key size for achieving authentication in symmetric-key.



User 1 will be authenticated and files will be uploaded to the local server and from local server files will be transferred to the cloud storage.Receivers i.e.,user 2 will be authenticated through local server,their attributes are checked and files are downloaded from the cloud storages.

SECURITY ISSUES

Whilecostandeaseof usearetwo greatbenefitsof cloudcomputing,therearesignificantsecurityconcern that must be taken in to consideration while movingdataacrossthenetwork.Cloudcomputingutiliz esthreedelivery models SaaS,PaaS andIaaS whichprovideinfrastructureresources,applicationpla tform andsoftwareasservicestotheconsumer. IaaS isthe foundationofallcloudservices,withPaaSbuilt uponitandSaaSinturnbuiltuponit.Justascapabilitiesa re inherited, so arethe information securityissues andrisks.Therearesignificanttradeoffstoeachmodeli nthetermsofintegratedfeatures,complexityvs.extensi bilityandsecurity.Ifthecloudserviceprovidertakescar eofonly thesecurityat thelowerpartofthesecurity architecture,theconsumersbecomemoreresponsiblef orimplementingandmanagingthesecuritycapabilitie sSecurity issues in different cloud service models exists.

Thefollowingkey ecurityelementsshouldbecarefully considered as anintegralpartoftheSaaS applicationdevelop- ment anddeployment process:

- Datasecurity
- Networksecurity
- Datalocality
- Backup
- DataBreaches
- Identitymanagementandsign-onprocess.

Networksecurity

InaSaaSdeploymentmodel,sensitivedataisobtainedfr omtheenterprises,processedbytheSaaSapplicationan dstored attheSaaSvendorend.Alldataflowoverthe networkneedstobesecuredinordertopreventleakageo fsensitive nformation.Thisinvolvestheuseofstrong networktraffic encryptiontechniquessuch as Secure SocketLayer(SSL)andtheTransportLayerSecurity(T LS)forsecurity.However,malicioususerscanexploitw eaknessesinnetworksecurityConfigurationtosniffnet workpackets.Thefollowing assessmentstestandvalidatethenetworksecurity oftheSaaS vendor:

- Networkpenetrationandpacket analysis
- Sessionmanagementweaknesses
- InsecureSSLtrust configuration.

Datalocality

InaSaaSmodelofacloudenvironment,theconsumersu setheapplicationsprovidedbytheSaaSandprocessthei r business data. But in thisscenario, the customer doesnotknowwherethedataisgettingstored.Inmanyac ases,thiscanbeanissue.Duecocomplianceanddatapriv acy lawsin variouscountries,locality ofdatais ofutmostimportanceinmanyenterprisearchitecture[1 0].Forexample,inmanyEU and South America countries, certain typesofdata cannot leavethecountrybecauseofpotentiallysensitiveinfor mation.Inadditiontotheissueoflocallaws,there"salsot hequestion of whose jurisdiction the data falls under, when an investigationoccurs.AsecureSaaS modelmustbecapableofprovidingreliabilitytothecust omeronthelocationofthedata oftheconsumer.

Backup

TheSaaSvendorneedstonsurethatallsensitiveenterpri se ataisregularlybackeduptofacilitatequickrecovery incase ofdisasters. Alsotheuseofstrongencryption schemestoprotectthebackupdataisrecommendedtopr eventaccidentalleakageofsensitiveinformation.Inthe caseofcloudvendors suchasAmazon,the dataat restinS3isnotencryptedbydefault.Theusersneedtose

parately encrypttheirdataand backupsso thatitcannotbeaccessedortamperedwithbyunauthorizedparties.

DataBreaches

Sincedatafromvarioususersandbusinessorganizations togetherinacloudenvironment,breachinginto thecloud environmentwillpotentially attackthedataof alltheusers. Thus the cloud becomes a high value target [11]. In the VerizonBusinessbreach reportblogit hasbeenstatedthat externalcriminalspose thegreatestthreat(73%),butachievetheleastimpact(30,000 compromisedrecords), resultingina PsuedoRiskScoreof67,500.Insidersposethe leastthreat(18%),andachievethegreatestimpact(375,000compromisedrecords),resultinginaPseudoRiskScoreof67,500.Partnersaremiddleinboth(73.39%and187,500)resulting inaPseudo RiskScoreof 73,125.ThoughSaaS advocatesclaimthatSaaS providerscanprovidebettersecuritytocustomers"datathanbyconventionalmeans,Insidersstillhaveaccesstothe data but itisjustthattheyareaccessing itinadifferentway.Insidersdonothavedirectaccesstodatabases,butitdoesnotreducetheriskofinsiderbreacheswhichcanbeamassive impactonthesecurity.The SaaSproviders"employeeshave accesstoalotmore informationandasingleincidentcould expose information from many customers. SaaS providers mustbecompliantwithPCIDSS(PaymentCardIndustry—DataSecurityStandards)[12]inorderto hostmerchantsthat must complywithPCIDSS.

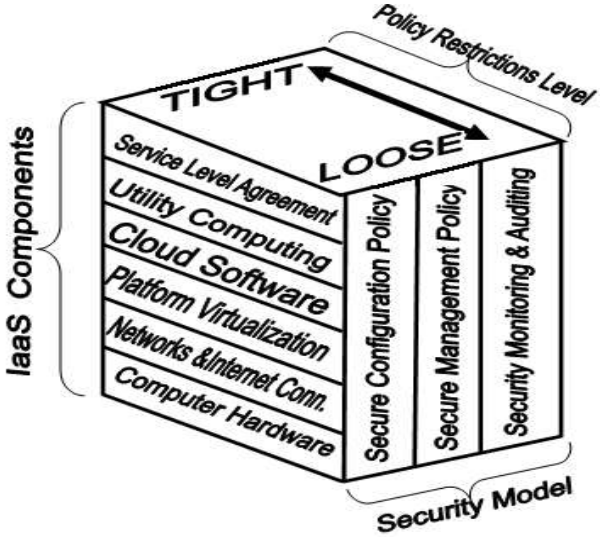


Fig 3:security model in IAAS

CONCLUSION

How to protect users' data privacy is a central question ofcloud storage. With more mathematical tools, cryptographicschemes are getting more versatile and often involve multiple keys 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.

FUTURE ENHANCEMENT

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. Otherwise, we need to expand the public-key as we described.Although the parameter can be downloaded withciphertexts, it would be better if its size is independent of the 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 resilientcryptosystemyet allows

efficient and flexible key delegation is also an interesting direction

REFERENCES

1. S.S.M. Chow, Y.J. He, L.C.K. Hui, and S.-M. Yiu, "SPICE – Simple Privacy-Preserving Identity-Management for Cloud Environment," Proc. 10th Int'l Conf. Applied Cryptography and Network Security (ACNS), vol. 7341, pp. 526-543, 2012.
2. L. Hardesty, Secure Computers Aren't so Secure.MITpress,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 on the 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," Cryptography and Security, pp. 442-464, Springer, 2012.
6. D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," Proc. 22nd Int'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, "Dynamic and Efficient Key Management for Access Hierarchies," ACM Trans. Information and System Security, vol. 12, no. 3, pp. 18:1-18:43, 2009.
8. J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records," 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-Key Decryption without Random Oracles," Proc. Information Security and Cryptology (Inscrypt '07), vol. 4990, pp. 384-398, 2007.
10. V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption 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 of Access 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 Master Keys," Proc. Advances in Cryptology (CRYPTO '89), vol. 435, pp. 316-322, 1989.
13. W.-G. Tzeng, "A Time-Bound Cryptographic Key Assignment Scheme for Access Control in a Hierarchy,"IEEE Trans. Knowledge and 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 Hierarchy for 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 in Secure Group Communications," Proc. IEEE INFOCOM '04, 2004.
17. Q. Zhang and Y. Wang, "A Centralized Key Management Scheme for Hierarchical Access Control," Proc. IEEE Global Telecomm. Conf. (GLOBECOM '04), pp. 2067-2071, 2004.
18. J. Benaloh, "Key Compression and Its Application to Digital Fingerprinting," technical report, Microsoft Research, 2009.

19. B. Alomair and R. Poovendran, "Information Theoretically Secure Encryption with Almost Free authentication," J. Universal Computer Science, vol. 15, no. 15, pp. 2937-2956, 2009.
20. D. Boneh and M.K. Franklin, "Identity-Based Encryption from the Weil 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, "Practical Leakage-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 to Decrypt 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 in Multi-Authority Attribute-Based Encryption,"Proc. ACM Conf. Computer and Comm. Security, pp. 121-130. 2009,
25. T. Okamoto and K. Takashima, "Achieving Short Ciphertexts or Short Secret-Keys for Adaptively Secure General Inner-Product Encryption," Proc. 10th Int'l Conf. Cryptology and Network Security (CANS '11),pp. 138-159, 2011
26. R. Canetti and S. Hohenberger, "Chosen-Ciphertext Secure Proxy Re-Encryption," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 185-194, 2007.
27. C.-K. Chu and W.-G. Tzeng, "Identity-Based Proxy Re-encryption without Random Oracles," Proc. Information Security Conf. (ISC '07), vol. 4779, pp. 189-202, 2007.
28. C.-K. Chu, J. Weng, S.S.M. Chow, J. Zhou, and R.H. Deng, "Conditional Proxy Broadcast Re-Encryption,"Proc. 14th Australasian Conf. Information Security and Privacy (ACISP '09), vol. 5594, pp. 327-342, 2009.
29. S.S.M. Chow, J. Weng, Y. Yang, and R.H. Deng, "Efficient Unidirectional Proxy Re-Encryption," Proc. Progress in Cryptology (AFRICACRYPT '10), vol. 6055, pp. 316-332, 2010.
30. G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage," ACM Trans. Information and System Security, vol. 9, no. 1, pp. 1-30, 2006.
31. D. Boneh, C. Gentry, and B. Waters, "Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys," Proc. Advances in Cryptology Conf. (CRYPTO '05), vol. 3621, pp. 258-275, 2005.
32. L.B. Oliveira, D. Aranha, E. Morais, F. Daguano, J. Lopez, and R. Dahab, "TinyTate: Computing the Tate Pairing in Resource-Constrained Sensor Nodes," Proc. IEEE Sixth Int'l Symp. Network Computing and Applications (NCA '07), pp. 318-323, 2007.
33. D. Naor, M. Naor, and J. Lotspiech, "Revocation and Tracing Schemes for Stateless Receivers," Proc. Advances in Cryptology Conf. (CRYPTO '01), pp. 41-62,2001.
34. T.H. Yuen, S.S.M. Chow, Y. Zhang, and S.M. Yiu, "Identity-Based Encryption Resilient to Continual Auxiliary Leakage," Proc. Advances in Cryptology Conf.