International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

Key-Aggregate Cryptosystem for Confidential Data Sharing in Public Cloud

Ms. Vemula Chaitanya
(M.Tech student) Deportment of Computer science and Engineering Aurora's Technological and Research Institute
Email id:vemulachaitanya7@gmail.com
Mrs. M. Nirmala
(Associate Professor) Deportment of Computer Science and Engineering and IT Aurora's Technological and Research Institute
Email id:madhavapeddynirmala@gmail.com

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

Index Terms—Data sharing; Cloud storage; key-aggregate encryption; patient-controlled encryption

1INTRODUCTION

We see the rise in demand for data outsourcing, which assists in the strategic management of corporate data in Cloud storage, and is also used as acore technology behindmany online services for personalapplications.
Nowadays,itiseasytoapplyforfreeaccountsforemail, photo album,file sharingand/or remoteaccess, with storagesizemore than 25GB.Alongwiththecurrentwirelesstechnology, users canaccessalmost alloftheir databya mobile phonefrom any place.

In Cloud Storage, data from differentclients can be hostedon separatevirtualmachines(VMs) but reside onasingle physicalmachine.Datainadestined VMcould bestolen by instantiatinganotherVMco-residentwith the target one.To know about the data availability,there areaseries ofcryptographicschemes which goasfaras allowingathird-partyauditorto check the availability of files on behalf of the data owner without leaking anythingabout the data.Likewise, cloud users probablywill not hold the strongbelief that the cloud server isdoing agood jobinterms ofconfidentiality.A cryptographicsolution, with provensecurityre- liedonnumber-theoreticassumptionsismore desirable, wheneverthe user isnot perfectly happywith trusting the securityofthe VMor the honestyofthe technical staff. These users are motivatedto encrypttheir data with theirownkeysbeforeuploadingthem totheserver.

Data sharingis an important functionalityin cloud storage.Consider anexample,bloggerscanlettheir

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

friendsview a subset of their private pictures;an enterprisemay grant her employees access to a portionof sensitive data. The challenging problem is how to effectively share encrypteddata. Of course users can download theencrypteddata from thestorage, decryptthem, then send them toothers forsharing,butitlosesthevalue of cloudstorage.Usersshouldbeabletodelegatetheacces s rights ofthesharingdata toothers sothattheycanaccess these data from theserver directly.However,findingan efficient and secure way to share partial data in cloud storageisnottrivial. Here we take Dropbox as for this situation.

We considerthat Alice keepsall her private photos on Dropbox,and shedoesnotwant toexpose herphotosto everyone.Due tovariousdata leakage possibilityAlice cannot feelrelievedbyjustrelying ontheprivacyprotection mechanismsprovidedbyDropbox,sosheencrypts allthephotosusing herownkeysbeforeuploading.One day, Alice's friend, Bob, asks her to share the photos taken over allthese years which Bobappearedin.Alice can then use the share functionof Dropbox, but the problemhereis how to delegatethe decryption rights for these photosto Bob. Anoptiont h a t Alice hasistosecurelysend Bobthesecret keysinvolved. Generally,there aretwo ways forherunderthe traditionalencryptionparadigm

- Aliceencryptsallfileswith asingle encryptionkey and givesBobthecorrespondingsecretkeydirectly.
- Aliceencryptsfileswith distinctkeysandsends Bob the correspondingsecret keys.

Clearly,the first methodisinadequatesince all un-chosen data may bealso leaked toBob.Forother one,there are practicalconcernson efficiency. We should have separate key for every photo,say,athousand.Transferringthese secret keys inherentlyrequiresa secure channel,and storing these keysrequiresratherexpensive secure storage.The costsand complexities for these generallyincrease with thenumberofthedecryptionkeystobeshared.In other words, it is very expensive.

Encryptionkeys also come with two flavors — sym- metric keyorasymmetric(public) key.Using symmetric encryption,when Alicewantsthe data tobeoriginated from a third party, she has to give the encryptorher secret key; obviously,this is not alwaysdesirable.By contrast,theencryptionkeyand decryption keyare differentinpublic-keyencryption.Theuseofpublic-key encryptiongivesmoreflexibilityforourapplications.F or example,inenterprisesettings, every employeecanup- load encrypteddata onthecloud storageserver without the knowledgeofthe company'smaster-secretkey.

The best available solutionforthisproblemis that Alice encryptsfileswith distinctpublic-keys,but only sends Bobasingle (constant-size) decryptionkey. Since the decryptionkey shouldbe sent via a secure channeland kept secret, small key size is alwaysdesirable. Example,we can not expect large storage for decryptionkeys in the resource-constraint devices likesmart phones, smart cards orwirelesssensor nodes. Especially,these secret keys are usuallystoredin the tamper-proofmemory,which isrelativelyexpensive.The present research efforts mainly focus on minimizing the communicationrequirements.

2KEY-AGGREGATEENCRYPTION

A) FRAMEWORK

Mainly, a key-aggregate encryption scheme has of fivepolynomial-time algorithms as follows.The data owner establishes the public system parametervia Setup and generates a public/master-secret3 keypair via KeyGen. Messages can be encrypted via Encryptby anyone who also decides what ciphertext class isassociated with the plaintext message to be encrypted.Master secrete can be used by the data owner to generatean aggregate decryption key for a set of ciphertextclasses. The generated keys can be passed todelegatees securely (via secure e-mails or secure devices)Finally, any user with an aggregate key can decryptany ciphertext provided that the ciphertext's class iscontained in the aggregate key via Decrypt mechanism.

- Setup(1^{λ},n):executedbythedata ownertosetup an accountonanuntrusted server. Oninput

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

asecurity level parameter 1^{λ} and the numberof ciphertext classes n (i.e., class index should be an integer boundedby1and n),itoutputsthe public system parameterparam, which isomittedfrom the input ofthe other algorithmsforbrevity.

- KeyGen: executedby the data ownerto randomly generateapublic/master-secretkeypair (pk,msk).
- Encrypt(pk,i, m):executedbyanyonewho wantsto encryptdata. On inputapublic-keypk,an index i denotingthe ciphertextclass, and amessagem, it outputsaciphertextC.
- Extract(msk,S):executedbythedata owner fordel- egating thedecrypting powerforacertain setofci-phertextclassestoadelegatee. Oninputthemaster- secret keymskand asetSofindices corresponding todifferentclasses, itoutputstheaggregatekeyfor setSdenotedby K_S .
- Decrypt(K_S,S,i,C): executed by a delegateewho receivedanaggregatekey K_S generatedbyExtract. On input K_S,the set S,an index i denotingtheciphertextclass the ciphertextCbelongs to,and C, itoutputsthe decryptedresult mifi

B) SHARING ENCRYPTED DATA

Anapplication of KACis data sharing.The schemesenable acontentprovidertoshare her data in aconfidentialand selective way, with afixedand small ciphertextexpansion,bydistributing toeachauthorized user asingle and small aggregatekey.Here we describethe main idea of data sharingin cloudstorageusing KAC,illustratedinFigure2.Suppose Alice wantsto share her data $m_1,m_2,...,m_\nu$ on theserver. Shefirst performsSetup(1^{λ},n) toget paramandexecute KeyGentogetthe public/master-secretkeypair (pk,msk). The systemparameterparamand public-key pkcanbemade public and master-secretkeymskshould bekept secretbyAlice.Anyone (includingAliceherself) canthen encrypteach m_i by C_i=Encrypt(pk,i,m_i).Finally, the encrypted data are sent to server.

With paramand pk,peoplewho cooperatewith Alice can updateAlice's data on the server. Once Alice is willing toshare aset Sofher data with afriend Bob, she can compute the aggregate key K_S for Bob byperformingExtract(msk,S).Since K_S isjust aconstant sizekey,itiseasy tobesent toBobviaasecure e-mail.

After getting the aggregatekey,Bobcandownload thedata that he has access with.Thatis,foreachi∈S, Bobdownloads C_i (and some neededvalues inparam) from the server. With the aggregatekey K_S,Bob candecrypteach C_i by Decrypt(K_S,S,i,C_i)foreach i∈S.

3 RELATED WORK

Here we compare our basic KAC scheme with other possible solutions on sharing in secure cloud storage.We summarize our comparisons in Table 1.

3.1 Cryptographic Keys for a Predefined Hierarchy

Here we discuss the most relevant study in thecryptography/security. Cryptographic keyassignment schemes aim toreduce the expense in storing and managing secretkeys for general cryptographic use. Utilizing a tree structure,a key for a given branch can be used to derivethe keys of its descendant nodes. Just granting the parent key implicitly grants allthe keys of its descendant nodes.. The concept canbe generalized from a tree to a graph. More advancedcryptographic key assignment schemes support accesspolicy that can be modeled by an acyclic graph or acyclic graph. Most of these schemes producekeys for symmetric-key cryptosystems, even thoughthe key derivations may require modular arithmetic asused in public-key cryptosystems, which are generallymore expensive than "symmetric-key operations" suchas pseudorandom function.

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

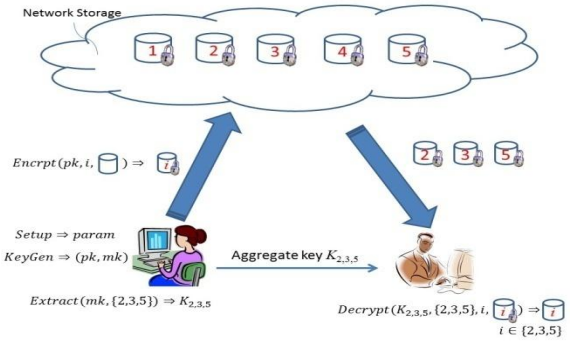


Figure 2

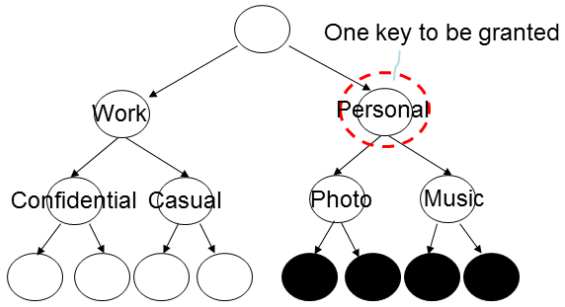


Fig. 3a

Consider the below figure3, Each node in the tree representsa secret key, while the leaf nodes represents the keysfor individual ciphertext classes. Filled circles representthe keys for the classes to be delegated and circlescircumvented by dotted lines represent the keys to begranted. Note that every key of the non-leaf node canderive the keys of its descendant nodes.

	Decryptionkeysize	Ciphertextsize	Encryptiontype
Keyassignmentschemes	most likely non-constant	constant	symmetricorpublic-key
Symmetric-keyencryptionwith CompactKey(e.g.,[8])	Constant	constant	symmetric-key
IBEwith CompactKey(e.g.,[9])	Constant	non-constant	public-key
Attribute-BasedEncryption(e.g.,[10])	non-constant	constant	public-key
KAC	Constant	constant	public-key

Table I : Comparisonsbetweenourbasic KACschemeandotherrelatedschemes

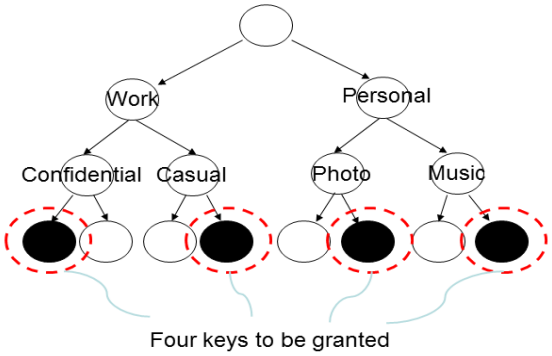


Fig 3b

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

In Figure 3(a), if Alice wants to share all the files in the "personal" category, she only needs to grant the keyfor the node "personal", which automatically grants thedelegatee the keys of all the descendant nodes ("photo","music"). This is the ideal case, where most classes tobe shared belong to the same branch and thus a parentkey of them is sufficient

But in general cases, it is difficult. Asshown in Figure 3(b), if Alice shares her demomusic at work ("work"!"casual"!"demo" and"work"!"confidential"!"demo") with a colleaguewho also has the rights to see some of her personaldata, what she can do is to give more keys, which leadsto an increase in the total key size. One can see thatthis approach is not flexible when the classificationsare more complex and she wants to share different setsof files to different people. For this delegatee, the number of granted secret keys becomesthe same as the number of classes.

3.2 Compact Key in Symmetric-Key Encryption

Benaloh etal.presented an encryptionscheme which isoriginallyproposedforconcisely transmittinglarge numberof keys in broadcast scenario [18].Theconstructionissimple and webriefly review its key derivationprocess here for a concrete description ofwhat arethedesirable propertieswewant toachieve. Thederivation ofthekeyforasetofclasses (which isa subset ofall possible

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

ciphertextclasses) is as follows. A compositemodulusN =p·qischosen wherepand qaretwo large randomprimes.Amastersecret key Y ischosen at randomfrom Z^*.Each classisassociatedwith a distinctprimee$_i$.All these prime numberscan be put in the public systemparameter A constant-sizekey for set $S^{()}$can be generated. Asaconcrete example,akeyforclasses representedbye1,e2,e3 canbe generated as $Y^{(1/e1.e2.e3)}$ from which, $Y^{1/e1}$, $Y^{1/e2}$, $Y^{1/e3}$ can be easily derived.

This approachachievessimilar properties andperformancesas our schemes. However,itisdesignedforthesymmetric-keysetting instead.Theencryptorneeds to get the correspondingsecret keys to encryptdata, which isnot suitableformany applications.Sincetheir methodisused togenerateasecret value ratherthan a pairofpublic/secret keys,itisunclearhowtoapplythis idea forpublic-keyencryptionscheme.

4 CONCRETECONSTRUCTIONSOFKAC

Let **G** and $\mathbf{G_T}$ be two Cyclic groups of prime order **p** and **e:G xG** → $\mathbf{G_T}$be a map with the following properties.

- Bilinear: $\forall g_1, g_2 \in \mathbb{G}$, $a, b \in \mathbb{Z}$, $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$.
- Non-degenerate: for some $g \in \mathbb{G}$, $e(g, g) \neq 1$.

G is a bilinear group if all the options involved abouve are efficiently computable.. many classes of elliptic curves feature billinear groups**.**

4.1 A Basic Construction

Although some schemesupports constant-size secret keys, every key only has the power for decrypting ciphertexts associated to aparticular index. We thus need to devise a new Extractalgorithm and the corresponding Decrypt algorithm.

- Setup(1^λ, n): Randomly pick a bilinear group G of prime order p where $2^\lambda \leq p \leq 2^\lambda + 1$, a generator $g \in \mathbb{G}$ and $\alpha \in_R \mathbb{Z}_p$. Compute $g_i = g^{\alpha^i} \in \mathbb{G}$ for $i = 1, \ldots, n, n+2, \ldots, 2n$. Output the system parameter as param = (g, $g_1, \ldots, g_n; g_{n+2}, \ldots, g_{2n}$) (**a** can be safely deleted after Setup).
- KeyGen(): Pick $\gamma \in_R \mathbb{Z}_p$, output the public and aster-secret key pair: $(\mathtt{pk} = v = g^\gamma, \mathtt{msk} = \gamma)$.
- Encrypt(pk; i;m): For a message $m \in \mathbb{G}_T$ and an index $i \in \{1, 2, \cdots, n\}$, randomly pick $t \in_R \mathbb{Z}_p$ and compute the cipher text $\mathfrak{C} = \langle g^t, (vg_i)^t, m \cdot \hat{e}(g_1, g_n)^t \rangle$.
- $\mathsf{Extract}(\mathtt{msk} = \gamma, \mathcal{S})$: for the set **S** of indicesj's, the aggregate key is computed as $K_{\mathcal{S}} = \prod_{j \in \mathcal{S}} g_{n+1-j}^{\gamma}$. since **S** does not include $\hat{e}(g_1, g_n)^t \rangle$. can always be retrieved from param.
- Decrypt($K_{\mathcal{S}}, \mathcal{S}, i, \mathfrak{C} = \langle c_1, c_2, c_3 \rangle$): If $i \notin \mathcal{S}$, output $\perp$. Otherwise, return the message: $m = c_3 \cdot \hat{e}(K_{\mathcal{S}} \cdot \prod_{j \in \mathcal{S}, j \neq i} g_{n+1-j+i}, c_1) / \hat{e}(\prod_{j \in \mathcal{S}} g_{n+1-j}, c_2)$.

  For the data owner, with the knowledge of γ, the term $\hat{e}(g_1, g_n)^t$ can be easily recovered by $\hat{e}(c_1, g_n)^\gamma = \hat{e}(g^t, g_n)^\gamma = \hat{e}(g_1, g_n)^t$.

For correctness, we can see that

$$
\begin{aligned}
& c_3 \cdot \hat{e}(K_{\mathcal{S}} \cdot \prod_{j \in \mathcal{S}, j \neq i} g_{n+1-j+i}, c_1) / \hat{e}(\prod_{j \in \mathcal{S}} g_{n+1-j}, c_2) \\
= \; & c_3 \cdot \frac{\hat{e}(\prod_{j \in \mathcal{S}} g_{n+1-j}^{\gamma} \cdot \prod_{j \in \mathcal{S}, j \neq i} g_{n+1-j+i}, g^t)}{\hat{e}(\prod_{j \in \mathcal{S}} g_{n+1-j}, (vg_i)^t)} \\
= \; & c_3 \cdot \hat{e}(\prod_{j \in \mathcal{S}, j \neq i} g_{n+1-j+i}, g^t) / \hat{e}(\prod_{j \in \mathcal{S}} g_{n+1-j}, g_i^t) \\
= \; & c_3 \cdot \frac{\hat{e}(\prod_{j \in \mathcal{S}} g_{n+1-j+i}, g^t) / \hat{e}(g_{n+1}, g^t)}{\hat{e}(\prod_{j \in \mathcal{S}} g_{n+1-j+i}, g^t)} \\
= \; & m \cdot \hat{e}(g_1, g_n)^t / \hat{e}(g_{n+1}, g^t) = m.
\end{aligned}
$$

4.2 Public-Key Extension

If a user needs to classify his ciphertexts into morethan n classes, he can register for additional key pairs(pk2; msk2), . . , (pk_l`; $msk_l$`). Each class now is indexed by a 2-level index in $\{(i, j) | 1 \leq i \leq \ell, 1 \leq j \leq n\}$ and the number of classes is increased by **n** for each added key.

Figure 4. Figure 4 shows the flexibility of our approach.We achieve "local aggregation", which means the secretkeys under the same branch can always be aggregated.We use a quaternary tree for the last level just for betterillustration of our distinctive feature. Our advantage isstill preserved when compared with quaternary trees inhierarchical approach, in which the latter either delegatesthe decryption power for all 4 classes (if the key for theirparent class is delegated) or the number of keys will bethe same as the number of classes. For our approach, atmost 2 aggregate keys are needed in our example.

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

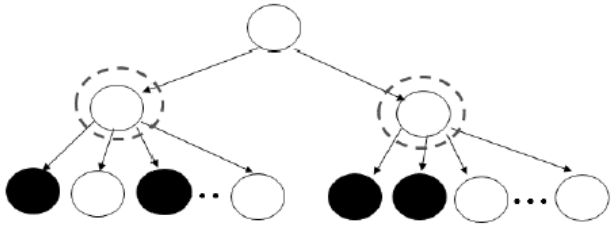


Fig. 4. Key assignment in our approach

- Setup and KeyGen: Same as the basic construction
- Extend(pkl; mskl): ExecuteKeyGen() to get $(v_{l+1}, \gamma_{l+1}) \in \mathbb{G} \times \mathbb{Z}_p$, output the public and extended master-secrete keys as $\mathtt{pk}_{l+1} = (\mathtt{pk}_l, v_{l+1}), \mathtt{msk}_{l+1} = (\mathtt{msk}_l, \gamma_{l+1})$.
- Encrypt$(\mathtt{pk}_l, (a,b), m)$: Let $\mathtt{pk}_l = \{v_1, \cdots, v_l\}$. For an index $(a,b), 1 \le a \le l, 1 \le b \le n$, pick $t \in_R \mathbb{Z}_p$, output the ciphertext as $\mathfrak{C} = \langle g^t, (v_a g_b)^t, m \cdot \hat{e}(g_1, g_n)^t \rangle$.
- Extract$(\mathtt{msk}_l, \mathcal{S}_l)$: Let $\mathtt{msk}_l = \{\gamma_1, \gamma_2, \cdots, \gamma_l\}$. For a set $\mathcal{S}_l$ of indices $(i,j), 1 \le i \le l, 1 \le j \le n$, get $g_{n+1-j} = g^{\alpha^{n+1-j}}$ from param, output:

$$K_{\mathcal{S}_l} = \Big(\prod_{(1,j)\in\mathcal{S}_l} g_{n+1-j}^{\gamma_1}, \prod_{(2,j)\in\mathcal{S}_l} g_{n+1-j}^{\gamma_2}, \cdots, \prod_{(l,j)\in\mathcal{S}_l} g_{n+1-j}^{\gamma_l} \Big).$$

- Decrypt$(K_{\mathcal{S}_l}, \mathcal{S}_l, (a,b), \mathfrak{C})$: If $(a,b) \notin \mathcal{S}_l$, output $\perp$. Otherwise, let $K_{\mathcal{S}_l} = (d_1, \cdots, d_l)$ and $\mathfrak{C} = \langle c_1, c_2, c_3 \rangle$. Output the message:

$$m = \frac{c_3 \cdot \hat{e}(d_a \cdot \prod_{(a,j)\in\mathcal{S}_l, j\neq b} g_{n+1-j+b}, c_1)}{\hat{e}(\prod_{(a,j)\in\mathcal{S}_l} g_{n+1-j}, c_2)}.$$

Just like the basic construction, the decryption canbe done more efficiently with the knowledge of i's.
Correctness is not much more difficult to see:

$$\begin{aligned} & c_3 \cdot \hat{e}(d_a \cdot \prod_{(a,j)\in\mathcal{S}_l, j\neq b} g_{n+1-j+b}, c_1) \\ & / \hat{e}(\prod_{(a,j)\in\mathcal{S}_l} g_{n+1-j}, c_2) \\ = \; & c_3 \cdot \hat{e}(\prod_{(a,j)\in\mathcal{S}_l} g_{n+1-j}^{\gamma_a} \cdot \prod_{(a,j)\in\mathcal{S}_l, j\neq b} g_{n+1-j+b}, g^t) \\ & / \hat{e}(\prod_{(a,j)\in\mathcal{S}_l} g_{n+1-j}, (v_a g_b)^t) \\ = \; & c_3 \cdot \hat{e}(\prod_{(a,j)\in\mathcal{S}_l, j\neq b} g_{n+1-j+b}, g^t) / \hat{e}(\prod_{(a,j)\in\mathcal{S}_l} g_{n+1-j}, g_b^t) \\ = \; & m \cdot \hat{e}(g_1, g_n)^t / \hat{e}(g_{n+1}, g^t) = m. \end{aligned}$$

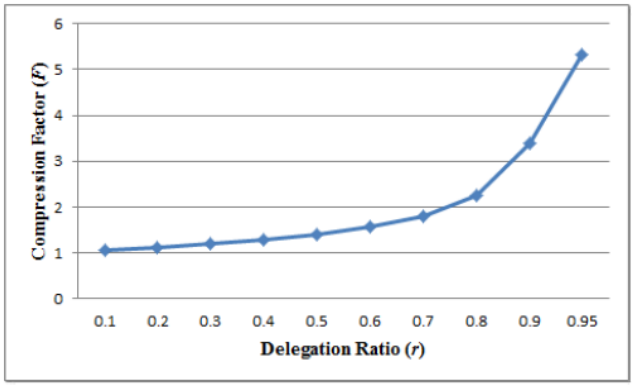
5 PERFORMANCE ANALYSIS

5.1 Compression Factors

We consider, that there are exactly 2h ciphertext classes,and the delegatee of concern is entitled to a portion**r** of them. That is, **r** is the delegation ratio, the ratioof the delegated ciphertext classes to the total classes.Obviously, if **r** = 0, $\mathbf{n_a}$ should also be 0, which meansno access to any of the classes; if r = 100%, $\mathbf{n_a}$ shouldbe as low as 1, which means that the possession of onlythe root key in the hierarchy can grant the access to allthe 2h classes. Consequently, one may expect that namay first increase with r, and may decrease later. Weset r = 10%; 20%, . . . ,90%, and choose the portion in arandom manner to model an arbitrary "delegation pattern"for different delegatees. For each combination of rand h, we randomly generate 104 different combinationsof classes to be delegated, and the output key set sizena is the average over random delegations.

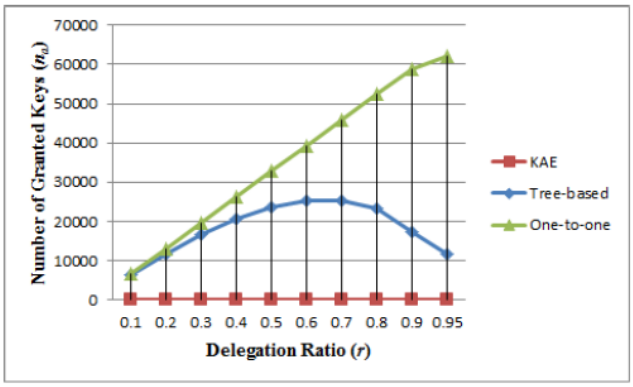
We tabulate the results in Table 2, where h = 16; 18; 20respectively. For a given h, $\mathbf{n_a}$ increases with the delegation ratio **r** until r reaches ~ 70%. An amazing fact isthat, the ratio of $\mathbf{n_a}$ to N(= 2^{h+1}- 1), the total number ofkeys in the hierarchy (e.g., N = 15 in Figure 3), appearsto be only determined by r but irrelevant of h. Thisis because when the number of ciphertext classes (2h)is large and the delegation ratio (r) is fixed, this kindof random delegation achieves roughly the same key assignment ratios (n_a=N). Thus, for the same r, na grows exponentially with h. We can easily estimate how manykeys we need to assign when we are given r and h.

The average number of delegatedclasses that each granted key can decrypt. Specifically, itis the ratio of the total number of delegated classes (r2h)to the number of granted keys required (na). Certainly,higher compression factor is preferable because it meanseach granted key can decrypt more ciphertexts. Figure5(a) illustrates the relationship between the compressionfactor and the delegation ratio. Somewhat surprisingly,we found that F = 3:2 even for delegation ratio of r =0:9, and F < 6 for r = 0:95, which deviates from theintuition that only a small number of "powerful" keysare needed for delegating most of the classes. We canonly get a high (but still small) compression factor whenthe delegation ratio is close to 1 and a comparison of the number of granted keys betweenthree methods is depicted in Figure 5(b).



(a)

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org



(b)

Fig. 5. (a) Compression achieved by the tree-based approach for delegating different ratio of the classes (b) Number of granted keys (n_a) required for different approaches in the case of 65536 classes of data

6 NEW PATIENT-CONTROLLED ENCRYPTION

Motivated by the nationwide effort to computerizeAmerica's medical records, the concept of patientcontrolledencryption (PCE) has been studied. InPCE, the health record is decomposed into a hierarchicalrepresentation based on the use of different ontologies,and patients are the parties who generate and store secretkeys. When there is a need for a healthcare personnel toaccess part of the record, a patient will release the secretkey for the concerned part of the record. In the workof 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-keyPCE for fixed hierarchy (the IBE analog of the folkloremethod, as mentioned in Section 3.1), and RSA-basedsymmetric-key PCE for "flexible hierarchy" (which is the"set membership" access policy as we explained).

Our work provides a candidate solution for the missingpiece, public-key PCE for flexible hierarchy, whichthe existence of an efficient construction was an openquestion. Any patient can either define her own hierarchyaccording to her need, or follow the set of categoriessuggested by the electronic medical record system sheis using, such as "clinic visits", "x-rays", "allergies","medications" and so on. When the patient wishes togive access rights to her doctor, she can choose anysubset of these categories and issue a single key,

h	r	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.95
16	n_a	6224.8	11772.5	16579.3	20545.8	23520.7	25263.8	25400.1	23252.6	17334.6	11670.2
	$\frac{n_a}{N}$	4.75%	8.98%	12.65%	15.68%	17.94%	19.27%	19.38%	17.74%	13.23%	8.90%
18	n_a	24895.8	47076.1	66312.4	82187.1	94078.8	101052.4	101594.8	93025.4	69337.4	46678.8
	$\frac{n_a}{N}$	4.75%	8.98%	12.65%	15.68%	17.94%	19.27%	19.38%	17.74%	13.23%	8.90%
20	n_a	99590.5	188322.0	265254.1	328749.5	376317.4	404205.0	406385.1	372085.2	277343.1	186725.4
	$\frac{n_a}{N}$	4.75%	8.98%	12.65%	15.68%	17.94%	19.27%	19.38%	17.74%	13.22%	8.90%

TABLE 2
Compression ratios for different delegation ratios and tree heights

fromwhich keys for all these categories can be computed.Thus, we can essentially use any hierarchy we choose,which is especially useful when the hierarchy can becomplex. Finally, one healthcare personnel deals withmany patients and the patient record is possible storedin cloud storage due to its huge size (e.g., high resolutionmedical imaging employing x-ray), compact key size andeasy key management are of paramount importance.

7 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 article, 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.

REFERENCES

[1] S.S.M.Chow, Y.J.He, L.C.K.Hui, and S.-M.Yiu,"SPICE - Simple Privacy-Preserving Identity-ManagementforCloud Envi- ronment,"in AppliedCryptographyandNetworkSecurity-ACNS

2012,ser.LNCS,vol.7341. Springer,2012,pp. 526–543.

[2] L.Hardesty, "Securecomputersaren'tsosecure," MITpress,2009, http://www.physorg.com/news176107396.html.

[3] C.Wang, S. S. M.Chow, Q.Wang, K.Ren,and W.Lou,"Privacy-PreservingPublicAuditingforSecureCloud Storage," IEEETrans. Computers,vol.62,no.2,pp. 362–375,2013.

[4] B.Wang, S.S.M.Chow, M.Li,and H.Li,"StoringShared Data on the Cloud via Security-Mediator,"in International Conference onDistributedComputingSystems-ICDCS2013. IEEE,2013.

[5] S.S.M.Chow, C.-K.Chu, X.Huang,J.Zhou, and R.H. Deng, "DynamicSecure Cloud Storage with Provenance,"in Cryptog-raphyandSecurity:FromTheorytoApplications-EssaysDedicated toJean-JacquesQuisquaterontheOccasionofHis65thBirthday,ser. LNCS,vol.6805. Springer,2012,pp. 442–464.

[6] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate andVerifiably EncryptedSignatures fromBilinearMaps," in ProceedingsofAdvancesinCryptology-EUROCRYPT'03,ser.LNCS, vol.2656. Springer,2003,pp. 416–432.

[7] M.J.Atallah, M.Blanton, N.Fazio,and K.B.Frikken, "Dynamic and Efficient Key Managementfor Access Hierarchies," ACM TransactionsonInformationandSystemSecurity(TISSEC),vol. 12, no.3,2009.

[8] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient ControlledEncryption:EnsuringPrivacy of Electronic Medical Records," in ProceedingsofACM WorkshoponCloudComputing Security(CCSW'09). ACM,2009,pp. 103–114.

[9] F.Guo, Y.Mu, Z. Chen, and L.Xu,"Multi-Identity Single-Key DecryptionwithoutRandomOracles," inProceedingsof Informa- tion Security and Cryptology(Inscrypt '07),ser. LNCS, vol. 4990. Springer,2007,pp. 384–398.

[10] V.Goyal, O.Pandey,A.Sahai, and B.Waters, "Attribute-Based EncryptionforFine-GrainedAccess ControlofEncrypteddata," inProceedingsofthe13thACMConferenceonComputerandCom- municationsSecurity(CCS'06). ACM,2006,pp. 89–98.

[11] S.G.Akland P.D.Taylor, "CryptographicSolutiontoaProblem ofAccessControl inaHierarchy,"ACMTransactionsonComputer Systems(TOCS),vol.1,no.3,pp. 239–248,1983.

[12] G. C. Chick and S.E. Tavares, "Flexible Access Controlwith Master Keys," inProceedingsofAdvancesinCryptology-CRYPTO '89,ser.LNCS,vol.435. Springer,1989,pp. 316–322.