IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 1, Ver. I (Jan – Feb. 2015), PP 53-59
www.iosrjournals.org

Secure Data Sharing Using Compact Summation key in Hybrid Cloud Storage

Pratap M Mohite, DipaDharmadhikari,RavindraBankar
(Computer Science &Engineering,Marathwada Institute of Technology, India)
(Computer Science &Engineering,Marathwada Institute of Technology, India)
(Master of Computer Applications,Dr.BAMU, India)

Abstract: *Data security is crucial aspect in cloud storage. Providing security to a single file or to set of files is another important factor. In cloud, security is applied to a set of files i.e. to central location. In file sharing approach a single key can be used to gain access to storage. Applying security to each file in cloud is not possible with cloud environment. To solve this type ofproblems, new improved key assignment scheme have to be used. New improved algorithm that satisfies these conditions is summation key encryption. Summation key is of power more than two keys. It takes one public key, known to both parties & another key is generated using keygen algorithm. Data owner retrieves the summation key from the alternatives made to share data. This key can be dispatch to second user using secure channel for example email or smart cards. Summation key is compact in size & also decrease the size of files after encryption. Time duration required to generate summation key is very short as compare to Identity based or Attribute based encryption scheme.Practically how this algorithm is beneficial to provide security to individual files in cloud repositoryis described here.*

Keywords: *Cloud Server,Data Sharing, Data Security, Hybrid Cloud, Summation key.*

I. Introduction

Cloud system provides data sharing capabilities;this can provide an abundant of benefits to the users. There is currently a push for information technology industries to increase their data sharing efforts. In information technology industry there is tremendous increase in data outsourcing. Data can be outsourced or infrastructure can be shared or software can be used. Cloud services divide into three categories infrastructure as a service, platform as a service & software as a service. Infrastructure as a service is provision model in which on organization outsourced the equipment used to support operations including storage, hardware, servers & networking components. Platform as a service is a way of rent hardware, operating system storage & network capacity over the internet. Software as a service is a software distribution model in which applications are hosted by a vendors or service provider & made available to customers over network typically the internet.

Data privacy isa traditional way to shield access control mechanism after conventional authentication. Any unauthorized person can procure access & reveal all data. Sharing data over cloud encounters some security related problems. Files shared from different clients can be hosted on different virtual machines for example virtualization in Linux, but whole data is stored on single machine. Data in target virtual machine could be stolen by instantiation another virtual machine co-resistance with victim machine. Multiple cryptographic schemes are used to store & check availability of files on behalf of data owner but can't leak anything from that. But data owner can't depend totally on cloud server due to lack of confidentiality.

For this purpose data owner has to use some security policies to prevent unauthorized person to take data without permission of data owner. This will lead to manipulate some cryptographic methods while uploading data on cloud server. Principle term related to cloud is data sharing, but appraise the cloud computing environment propound without exposing mission critical applications and data to third party vulnerabilities.

Data owner can use website or any application to share his private data with his friends. Consider D as a set of whole data & $d_1, d_2, \mathrm{d}_3, ..., d_n$ are data elements i.e. $D = \{d_1, d_2, d_3, ..., \mathrm{d}_n\}$

Data owner desire to share only some data for ex: d_1, d_2 with his friend .This sort of action cannot possible in cloud computing environment. Reason behind this, data owner has not specified d_1, d_2 data separately; data is bunch of files. Using only one key data owner can gain access to data & he has to share that key also. His friend can see the whole dataset D.

From Fig 1, Assume that Peter puts his personal data on cloud storage & he does not desire that his distinctive data will not be available to everyone. Due to much security violation possibilities he cannot totally relied on cloud storage. To prevent this, whole data D is stored on to cloud. But after some days his friend Anna Want some photos from dataset D. At this situation Peter has to give his secret key to decrypt D set. By using only one single key Anna can gain access to whole set of data D& she would be take another files from set.
To prevent such situation Peter has to possible ways,

1) Either apply a single key to whole dataset D , k(D) Or
2) Apply keys to each elements in dataset for ex, $d_1, d_2, \mathrm{d}_3, ..., d_n$ i.e. $k_1(d_1), k_2(d_2), k_3(\mathrm{d}_3)...k_n(d_n)$

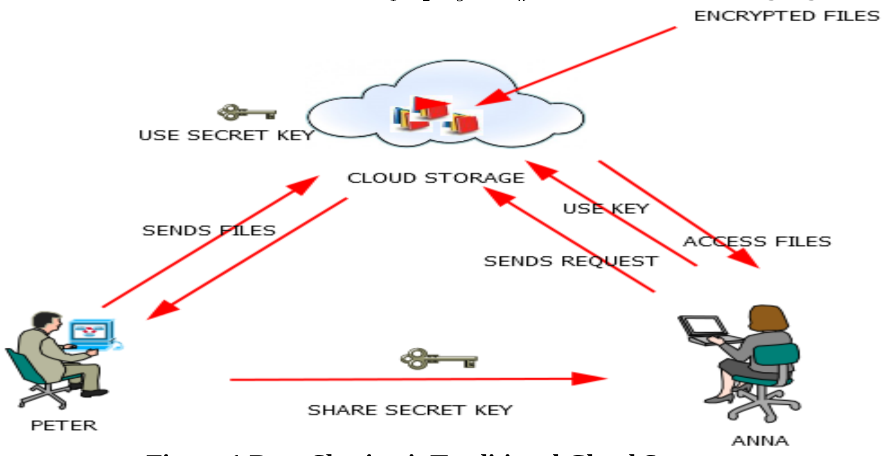


Figure 1 Data Sharing inTraditional Cloud System

First method is adverse because by using one single key Anna can obtain all data stored in cloud storage. In second method, applying each file a key can unbearable because of number of keys are as many as the number of files, say thousands. Transferring such keys requires a reliable storage the values & complication involved generally grow with the number of decryption keys to be shared. Encryption keys also come with two flavors symmetric key encryption& asymmetric key encryption.In symmetric key encryption scheme, Anna &Peter share a single common key to perform encryption & decryption operations. In asymmetric key scheme, Anna & Peter have different encryption and decryption keys. Flexibility is main application of public key encryption. For example in organization, every member can upload encrypted data on the cloud server without the knowledge of master secret key.

Therefore the best solution for above problem is that Peter encrypts all files with unique public keys, but only sends Anna a single decryption key which is constant in size that is called as summation key Fig 2. This key can be sent via a secure channel called email or a smart card or wireless sensor network. Basic aim of current system is to minimize hardware and communication cost as well as time duration of key generation.

There are various cryptographic schemes; those can be used to apply security mechanism over cloud storage but some lack in size & some are in integrity. Those are demonstrated in next section.

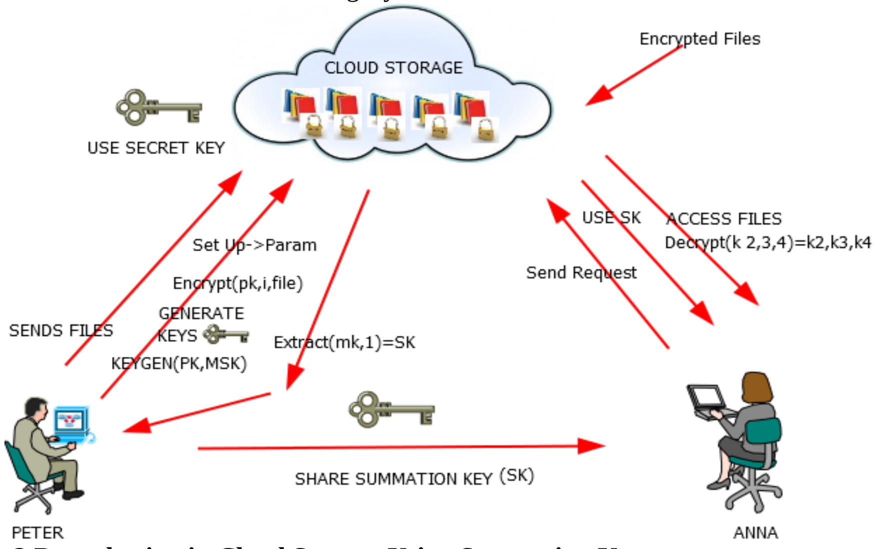


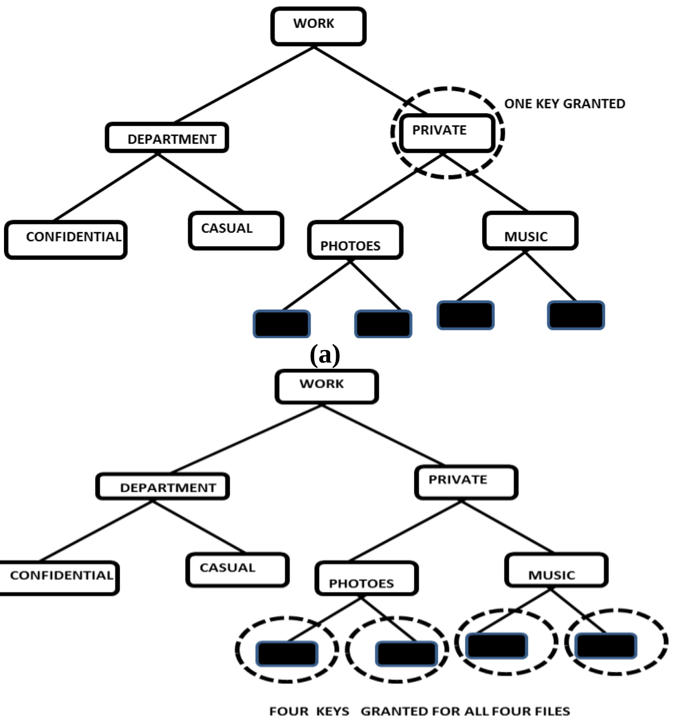
Figure 2 Data sharing in Cloud Storage Using Summation Key.

II. Related work

1. Types of Cryptography

1.1 Predefined Hierarchical Scheme

Predefined Hierarchical Schemes [1] aim to minimize the cost in storing and managing secret keys for general cryptographic use. Employing a tree structure a secrete key for a given arm can be used to procured the secret keys of its child nodes.Sandhu [2] proposed a technique to initiate a tree hierarchy of symmetric keys by using reiterated evaluations of pseudo random functions block cipher on stable secret. This notion can be verbalized from a tree to a graph.Another more advanced cryptographic key assignment schemes bears access policy that can be modelled by an acyclic graph [3].Most of these schemes construct keys for symmetric key cryptosystems, even though the key derivation may require modular arithmetic as used in public key cryptosystems, which are more expensive than" symmetric key operation" such as pseudorandom function[4]. Taking tree structure as an example, Peter can first classify the ciphertext classes according to their subjects like Fig 3.



(b)

Figure 3 a) One Key for Hierarchy, b) Four Keys for Hierarchy

Each node in the tree represents a secret key, while the leaf nodes represent the keys for individual ciphertext classes. Filled rectangles 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 canderive the keys of its descendant nodes. In Fig 3(a), if Peter wants to share all the files in the "personal" category, he only needs to grant the keyfor the node "personal", which automatically grants thedelegate 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 parent key of them is sufficient. However, it is still difficult for general cases. As shown in Fig 3(b), if Peter shares his 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 classificationshare more complex and she wants to share different setsof files to different people [5].

For this delegate in ourexample, the number of granted secret keys becomesthe same as the number of classes.In general, hierarchical approaches can solve the problem partially if one intends to share all files under a certain branch in the hierarchy. On average, the number ofkeys increases with the number of branches. It is unlikelyto come up with a hierarchy that can save the numberof total keys to be granted for all individuals (which canaccess a different set of leaf-nodes) simultaneously [6].

1.2 Attribute-based encryption

Attribute is analogues with cipher text. Data owner with master secret key can obtain a secrete key for the policy of attributes so that a ciphertext can be decrypted by this key if its associated attributes conforms to policy. Each attribute is associated with data this leads to increase in size of keys. For example with the secret key for the policy (2V3V6V8), one can decrypt ciphertext tagged with class 2, 3, 6 or 8 [7].The measure perturbed in attribute based encryption is collusion-resistance but not the compactness of secret keys. Actually, the size of the key often increases linearly with the number of attributes it encompasses, or the ciphertext-size is not immutable. To delegate the decryption power of some ciphertexts without sending the secret key to the delegate, a useful primitive is proxy re-encryption (PRE) [8]

A PRE permits peter to delegate tothe server (proxy) the ability to transform the ciphertexts encrypted under her public-key into ones for. PRE is well known to have innumerable applications including cryptographic file system. Nevertheless, Anna has to trustworthy the proxy that it only turns ciphertexts according to her instruction, which is what user wants to avoid at the first place [9]. Even worse, if the proxy colludes with, some form of Anna's secret key can be recovered which can decrypt Anna's (convertible) ciphertexts without Peter's further help. That also means that the transformation key of proxy should be well protected Using PRE just moves the secure key storage requirement from the delegate to the proxy. It is thus undesirable to let the proxy reside in the storage server. That will also be inconvenient since every decryption requires separate interaction with the proxy.

1.3 Identity Based Encryption

Identity-based encryption (IBE) is a type of public-key encryption in which the public-key of a user can be assign as an identity-string of the user (e.g. an email address). There is a trusted party called private key generator (PKG) in IBE which holds a master-secret key and issues a confidential key to each user concerning to the user identity. The encrypt or can take the public parameter and a user singularity to encrypt a message. The recipient can decrypt this ciphertext by his secret key [10].

Guoetet al, tried to build IBE with key aggregation. One of their schemes assumes random oracles but another does not. In their schemes, key aggregation is constrained in the sense that all keys to be summarize must come from different "identity divisions" [11]. While there are an exponential number of identities and thus secret keys, only a polynomial number of them can be summarize.

Most importantly, their key-aggregation [12], comes at the expense of $O(n)$ sizes for both ciphertexts and the public parameter, where nis the number of secret keys which can be summation into a constant size one. This substantiallygrows the costs of storing and transmitting ciphertexts, which is inappropriate in many cases such as shared cloud storage. As mentioned, in this scheme, feature constant ciphertext size, and their security holds in the standard model. In fuzzy IBE [13], one single compact secret key can decrypt ciphertexts encrypted under many identities which are close in a certain metric space, but not for an arbitrary set of identities and therefore it does not match with our idea of key aggregation.

Table 1: key Assignment Schemes

Scheme Name	Decryption Key	Encryption Key
Predefined key hierarchy	Not Constant	Constant
Compact Key	Constant	Constant
Attribute Based Encryption	Non Constant	Constant
Identity Based Encryption	Constant	Non constant
Compact Summation Key Encryption	Constant	Constant

In Table 1, shows comparisons between all the available schemes with proposed compact summation key encryption. Predefined key hierarchy has different encryption & decryption keys. Compact key have both encryption & decryption keys constant but those are single key no combination. Attribute based encryption have both different keys. Identity based encryption have one key constant and another non constant key. In compact summation key encryption both keys are constant less size.

III. Compact Summation Key Encryption

As there are various methods available for cloud security but they are lagging in storing keys & managing files. These schemes cannot provide a much security as required by market. In this paper a new approach called compact summation key (CSK) Encryption for storing and managing files is used. These schemes overcome all the drawbacks of IBE, ABE, HKM, and CK.

In traditional cryptography, messages were kept secret, but that approach can't be applied to modern cryptographic systems. Main aim of this paper is to keep data more secure and enhance power of encryption & decryption keys without increasing size of them.The design of basic scheme is inspired from the collusion-resistant broadcast encryption scheme proposed by Bonehet al. [14]. Although their scheme supports constant-size secret keys, every key only has the power for decrypting ciphertexts associated to a particular index. From above requirements there is need to devise a new extract algorithm and the corresponding encrypt algorithm.

This algorithm is divided into five polynomial time algorithms. Those are combined together to enhance security capabilities & to provide high level of security. Those steps are as follows,

1) **Step 1:**

$SETUP(1^{\lambda}, \mathrm{n})$: In $SETUP$, itrandomly pics a bilinear group G of prime order p where p lies between $2^{\lambda} \leq p \leq 2^{\lambda+1}$. A generator $g \in G$ and $\alpha \in Z_p$ .Where Z_p is another bilinear group to compute $g_i = g^{\alpha} \in G$ for $i = 1, 2, 3.., n, n+2, 2n$ .Output of the parameter as $param = \{g, g_1, ..., g_n, g_{n+2}, .., g_{2n}\}$ where param is group of small bilinear classes. Note that each ciphertext class is represented by an index in integer set $\{1, 2, ..., n\}$,where n the maximum number of ciphertext classes & 1^{λ} is security level parameter and the number of ciphertext classes n (i.e.class index should be an integer bounded by $1 \,\&\, i$.

2) **Step 2**

$KeyGen(pk, msk)$: Picks $\gamma \in_R Z_p$ output the public and master secret key pair $(pk = v = g^r, \mathrm{msk} = \gamma)$ where pk public key is and msk is the master secret key& γ is generated secret key v is vertices of tree generated.

3) **Step 3**

$Encrypt(pk, i, m)$: For a message $m \in G_T$ and index $i \in \{1, 2, 3 ... \mathrm{n}\}$, randomly picks $t \in_R Z_p$ and compute the ciphertext as 2^h where T is index for G bilinear group of prime numbers & 2^h is generated Keys height as in hierarchical key assignment scheme where h can be incremented as 16, 18, and 20.

4) **Step 4**

$Extract(msk = \gamma, s)$: For the set S of indices $j's$ the summation key is computed as

$$k_s = \prod_{j \in s} g_{n+1} - j, c_2$$

SinceS does not include 0 , can always be retrieved from param where K_s is extracted key which is summation key generated from cipher index classes & from group of bilinear group G .

5) **Step 5**

$Decrypt(k_s, s, i, C = \{c_1, c_2, c_3\})$: If $i \notin s \,.\, i = \{1, 2, 3 ... \mathrm{n}\}$ Otherwise, return the message:

$$m = c_3 \cdot \hat{e}(k_s \cdot \prod_{j \in S, j \neq i} g_{n+1-j+i}, c_1) / \hat{e}(k_s \cdot \prod_{j \in S} g_{n+1-j+i}, c_2)$$

Forthe data owner, with the knowledge of γ ,term $\hat{e}(\mathrm{g}_1, \mathrm{g}_n)^t$ can be easily recovered by $\hat{e}(\mathrm{c}_1, \mathrm{g}_n)^{\gamma} = \hat{e}(\mathrm{g}^t, \mathrm{g}_n)^{\gamma} = \hat{e}(\mathrm{g}_{1,} \mathrm{g}_n)^t$, where m is message which is converted to original form K_s i.e. summation key can be removed from message.

IV. Experimental Results

CASE I: File Compression Ratios

When files encrypted,file size can be changed dramatically, to check this BSdiff, Xdelta tools are used. These are the file compression tools that can be used to compress files. These results of file size can be compared with CSK scheme. As one of the feature of CSK scheme is file compression.Xdelta is a command line program for delta encoding, which generates two file differences. This is similar to diff and patch, but it is targeted for binary files and does not generate human readable output. The differences are recorded in discrete files called "deltas" or "diffs". In situations where differences are small for ex, the change of a few words in a large document or the change of a few records in a large table delta encoding greatly reduces data redundancy. Collections of unique deltas are substantially more space-efficient than their non-encoded equivalents.

In table 2 comparison of compression ratios of files are described with CSK scheme. In which summation key algorithm compress file size at large amount as compared with XDelta & BSDiff file compression tools. These tools specifically used for storing data on server. So CSK takes small space so data owner can put more files on cloud storage. This is shown graphically in Fig 4 CSK is at very high level in compression.

Table 2: File Comparison Ratios

File Name	File Size in KB(original)	XDelta	BSDiff	CSK
File 1	1197	616	464	57
File 2	399	240	214	46
File 3	362	211	197	42

File 4	296	176	173	39
File 5	293	174	171	38
File 6	260	164	160	36
File 7	91	76	71	17
File 8	15	10	9	8

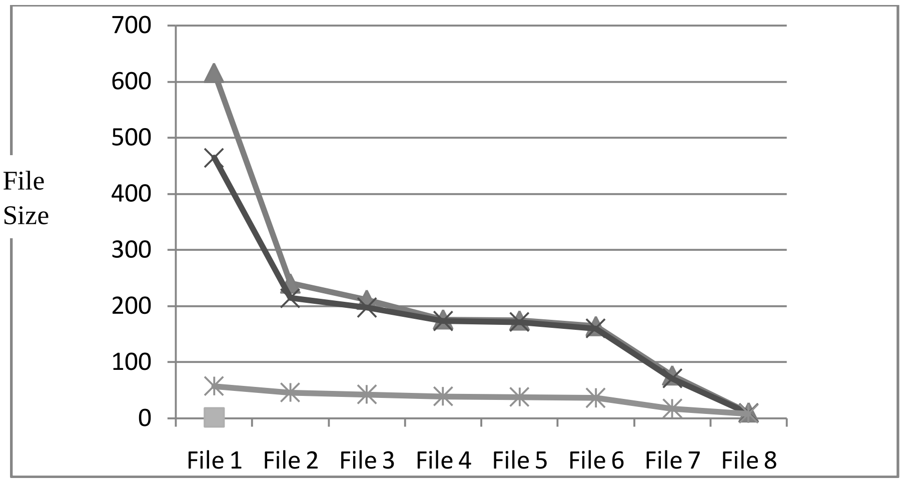


Figure 4File Comparison Ratios

CASE II: Comparative Execution Time

Table 3 shows comparative execution time of various popular schemes with CSK scheme in electronic code book mode. In this table compact summation key required very less execution time to performing its encryption task.

Fig5shows how CSK encrypt data in a very short duration as compared with hierarchical, attribute based and compact key. As data size increases, execution time also increases.

Table 3: Execution Time in Different Schemes

Size in Bytes	**Cryptographic Methods**			
	Hierarchical	**Attribute Based**	**Identity Based**	**Summation Key**
20527	2	7	4	2
36002	4	13	6	3
45911	5	17	8	4
59852	7	23	11	6
69545	9	26	13	7
137325	17	51	26	13
158959	20	60	30	16
166364	21	62	31	17
191383	24	72	36	19
232398	30	87	44	24

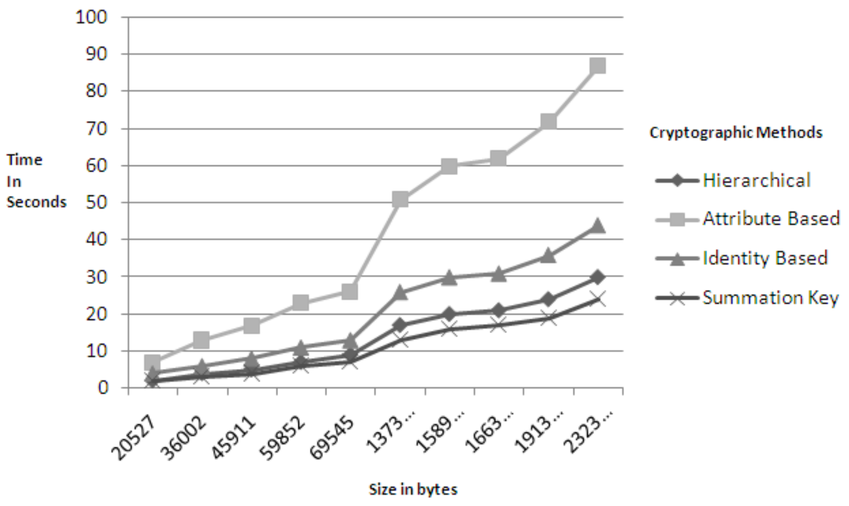


Figure 5Execution Time in Different Cryptographic Methods

Size in bytes

V. Conclusion

Protectuser's data is a prime question in cloud storage. Cryptographic schemes are used frequently and involve multiple keys for a single application this will lead to increase the size of keys. With compact summation key encryption algorithmdata can be kept secure with compact sized summation key on cloud server. This paper describes "summarization" of secret keys in public-key cryptosystems. This supports delegation of secret keys for different ciphertext classes in cloud storage. No matter which one among the power set of classes, the delegate can always get a summation key of constant size which is 16 bit in size. This scheme is more flexible than hierarchical key assignment, IBE& ABE. CSK scheme not only saves spaces but also time, required to generate keys. This is possible if & only if all key holders share a similar set of privileges.

Limitation in this work is the predefined bound of the number of maximum ciphertext classes. In cloud storage, the number of cipher texts usually grows rapidly.So data owner has to reserve enough ciphertext classes for the future extension. Otherwise, itneeds to expand the public-key as described in previous section. Although the parameter can be downloaded with ciphertexts, it would be better if its size is independent of the maximum number of ciphertext classes. On the other hand, when one carries the delegated keys in a mobile device without using special trusted hardware, the key is prompt to leakage, designing a leakageresilient cryptosystem, yet allows efficient andflexible key delegation is also an interesting direction.

References

[1]. S. G. Akl and P. D. Taylor, "Cryptographic Solution to a Problem of Access Control in a Hierarchy," ACM Transactions on ComputerSystems (TOCS), vol. 1, no. 3, pp. 239–248, 1983.

[2]. G. C. Chick and S. E. Tavares, "Flexible Access Control with Master Keys," in Proceedings of Advances in Cryptology – CRYPTO'89, ser. LNCS, vol. 435. Springer, 1989, pp. 316–322.

[3]. W.-G. Tzeng, "A Time-Bound Cryptographic Key Assignment Scheme for Access Control in a Hierarchy," IEEE Transactions on Knowledge and Data Engineering (TKDE), vol. 14, no. 1, pp. 182–188, 2002.

[4]. 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.

[5]. R. S. Sandhu, "Cryptographic Implementation of a Tree Hierarchy for Access Control," Information Processing Letters, vol. 27, no. 2, pp. 95–98, 1988

[6]. C.-K. Chu and W.-G.Tzeng, "Identity-Based Proxy Re-encryption Without Random Oracles," in Information Security Conference (ISC '07), ser. LNCS, vol. 4779. Springer, 2007, pp. 189–202.

[7]. C.-K. Chu, J. Weng, S. S. M. Chow, J. Zhou, and R. H. Deng, "Conditional Proxy Broadcast Re-Encryption," in AustralasianConference on Information Security and Privacy (ACISP '09), ser.LNCS, vol. 5594. Springer, 2009, pp. 327–342.

[8]. S. S. M. Chow, J. Weng, Y. Yang, and R. H. Deng, "Efficient Unidirectional Proxy Re-Encryption," in Progress in Cryptology - AFRICACRYPT 2010, ser. LNCS, vol. 6055.Springer, 2010, pp.316–332.

[9]. G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage," ACM Transactions on Information and System Security (TISSEC), vol. 9, no. 1, pp. 1–30, 2006.

[10]. D. Boneh, C. Gentry, and B. Waters, "Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys," inProceedings of Advances in Cryptology - CRYPTO '05, ser. LNCS, vol. 3621. Springer, 2005, pp. 258–275.

[11]. L. B. Oliveira, D. Aranha, E. Morais, F. Daguano, J. Lopez, and R. Dahab, "TinyTate: Computing the Tate Pairing in ResourceConstrained Sensor Nodes," in Proceedings of 6th IEEE International Symposium on Network Computing and Applications (NCA '07).IEEE, 2007, pp. 318–323.

[12]. D. Naor, M. Naor, and J. Lotspiech, "Revocation and Tracing Schemes for Stateless Receivers," in Proceedings of Advances in Cryptology - CRYPTO '01, ser. LNCS. Springer, 2001, pp. 41–62.

[13]. T. H. Yuen, S. S. M. Chow, Y. Zhang, and S. M. Yiu, "IdentityBased Encryption Resilient to Continual Auxiliary Leakage," in Proceedings of Advances in Cryptology - EUROCRYPT '12, ser. LNCS, vol. 7237, 2012, pp. 117–134.

[14]. D. Boneh, X. Boyen, and E.-J. Goh, "Hierarchical Identity Based Encryption with Constant Size Ciphertext," in Proceedings of Advances in Cryptology - EUROCRYPT '05, ser. LNCS, vol. 3494.Springer, 2005, pp. 440–456.

[15]. D. Boneh, R. Canetti, S. Halevi, and J. Katz, "Chosen-Ciphertext Security from Identity-Based Encryption," SIAM Journal on Computing (SIAMCOMP), vol. 36, no. 5, pp. 1301–1328, 2007.