19th International Computer Science and Engineering Conference (ICSEC)
Chiang Mai, Thailand, 23-26 November, 2015

Privacy-Preserving Access Control Model for Big Data Cloud

Somchart Fugkeaw
Department of Electrical Engineering and Information Systems
University of Tokyo, Japan
somchart@satolab.itc.u-tokyo.ac.jp

Hiroyuki Sato
Department of Electrical Engineering and Information Systems
University of Tokyo, Japan
schuko@satolab.itc.u-tokyo.ac.jp

***Abstract*—Due to the proliferation of advanced analytic applications built on a massive scale of data from several data sources, big data technology has emerged to shift the paradigm of data management. Big data management is usually taken into data outsourcing environment such as cloud computing. According to the outsourcing environment, security and privacy management becomes one of the critical issues for business decision. Typically, cryptographic-based access control is employed to support privacy-preserving authentication and authorization for data outsourcing scenario. In this paper, we propose a novel access control model combining Role-based Access Control (RBAC) model, symmetric encryption, and ciphertext attribute-based encryption (CP-ABE) to support fine-grained access control for big data outsourced in cloud storage systems. We also demonstrate the efficiency and performance of our proposed scheme through the implementation.**

Keywords: Big Data, Access Control, RBAC, CP-ABE, Encryption, Cloud Computing

I. INTRODUCTION

Big data (BD) is generally characterized to have a large amount of volume with a mixture of various data formats, and high velocity. This is beyond the capability of commonly used database systems. In addition, the computational resources equipped with high-end analytical software tools must be available to support the effective and efficient result analysis and presentation to support the decision making. Therefore, outsourcing service for managing big data in a cloud service is most convincing to achieve both costs reduction and high operational performance. Even though, cloud computing is promising, privacy and security of the data to be outsourced is paramount importance. Existing research works and cloud applications generally deploy encryption techniques and access control model to satisfy the security requirement.

Access control is among the most effective solutions for full-fledged network security control. Conventional access control models such as role-based access control (RBAC), mandatory access control (MAC), and discretionary access control (DAC) are not applicable to support the access control in untrusted domain like outsourcing environment. Generally, attribute-based access control (ABAC) is considered to provide flexibility and lightweight key management in data outsourcing environment. This is because the access control is based on the qualified set of attributes hold by users and it offers the encryption feature in the model itself.

However, data access control for big data should not only support the security but it should also provide a reasonable performance for the retrieval and processing of encrypted data. This is because a vast amount of new data volume may be loaded and queried anytime. Therefore, general attribute-based access control might not satisfy the performance requirement especially when there are a lot of attributes contained in the policy for encryption.

Attribute-based encryption (ABE) is considered as a powerful encryption scheme for outsourced data access control. This is because it formulates a light-weight encryption supporting both access policy enforcement and encryption feature. Goyal et al. proposed key-policy attribute-based encryption (KP-ABE) [7] to serve a more general and richer encrypted access control. In this scheme, the ciphertext is associated with a set of attributes for each of which a public key component is defined. User secret key is constructed to associate with the access structure. Nevertheless, the KP-ABE schemes [2,5] lacks the authority control over the policy for enforcement.

In 2007, Bethencourt et al. [8] proposed a Ciphertext Policy Attribute Based Encryption (CP-ABE) to address this limitation. In CP-ABE, each user is assigned a set of attributes which are embedded into the user's secret key and a public key is defined for each user attribute. The ciphertext is associated with the access policy structure in which the encryptor can define the access policy by her own control. Users are able to decrypt a ciphertext if their attributes satisfy the ciphertext access structure.

To date, access control solutions for cloud computing [3-6,12] have adopted CP-ABE to enforce the fine-grained access control with the focuses on minimizing key management cost, reducing computing cost of interaction between data owner and outsourced data storage, improving scalability and efficient revocation. However, the encryption performance of

CP-ABE is based on the number of attributes contained in the access policy and size of data. Thus, applying the CP-ABE for encrypting big data is not advisable and it leads to the overhead for data encryption and decryption.

Therefore, in addition to the fine-grained access control and privacy-preserving supporting big data in the cloud, performance-aware requirement is essential for practical deployment. In this paper, we employ symmetric encryption to be integrated into the CP-ABE for achieving both fine-grained access control and efficient performance for data encryption and decryption.

II. RELATED WORK

Tools and techniques for protecting big data must guarantee that processes and data are enforced by access control policies in a cost-effective and timely manner [19]. Traditional security perimeter such as firewall is not sufficient to protect sensitive big data clusters since the data clusters may be come from multiple sources and they may be distributed to several servers. Also, this also cannot prevent malicious insider. Instead, the authorization enforcement such as attribute-based authorization according to the access control polity is an effective way to control the proper access. By utilizing cloud infrastructure, sensitive data needs to be encrypted for the privacy and security.

Most existing research works have applied attribute-based access control (ABE) [3,5-8,12] for the access control in cloud computing since it is flexible and scalable to enforce the policies to a large number of unknown users as well as to provide the confidentiality over the data based on the attribute encryption.

Recently, Kan Yang et al [4] proposed DAC-MACS (Data Access Control for Multi-Authority Cloud Storage model. The authors apply CP-ABE technique to construct an access control model where there are several multi-authority issuing the attributes. The proposed scheme improves the decryption process and solves revocation problem in ABE by designing the decryption token and key update and ciphertext update algorithms. For the immediate revocation, their scheme reduces the cost for data re-encryption since only the ciphertext getting an effect is updated. However, this approach does not support write access.

In [13-14], the authors proposed a role-based encryption (RBE) scheme for cloud storage systems. The proposed RBE uses ABE for cryptographic access control and use identity broadcast encryption for key distribution. For the access control, the role-based access control (RBAC) policy is enforced through a public parameter of role and a group public to encrypt the data. Also, their scheme eliminates the cost of file re-encryption if there is any user revocation operation. However, the computation cost for storing tuples of ciphertext and cost for computing symmetric key K for every request is impractical for a large number of users having frequent data access over the remote cloud.

Recently, we have proposed a Collaborative- Cihpertext Policy- Attribute Role based Encryption (C-CP-ARBE) [15] scheme to support collaborative access control in multi-authority cloud systems. Our approach provides fine-grained access control and efficient revocation in collaborative data federations. However, the approach only supports the general data outsourcing setting and not consider in applying for big data.

In [16], the authors proposed access control model for big data based on Content-based Access Control (CBAC) model. Users can access the data based on the content rule defined in the CBAC policy. Also, content similarity is also used to compare with a preset threshold, and the user is granted access to all the records that pass the similarity check. However, this approach does not assume the data to be outsourced in other domains. Security features are not discussed.

Crucially, access control solutions for big data in cloud environment are now very demanding and they require more substantial investigations for usable security. Our research entails the practical solution for big data access control in cloud environment. We strategically extends our C-CP-ARBE scheme [15] to support privacy-preserving access control model for big data outsourced in the cloud by achieving both security and performance of operational use over big data. In this paper, we focus on the security and privacy-preserving control over the analytically integrated files (considered as big data) stored in the cloud storage.

III. BACKGROUND

Ciphertext Policy Attribute-based Encryption (CP-ABE)

Bethencourt et al. [8] propose an attribute-based encryption (ABE) primitive called a ciphertext policy attribute-based encryption (CP-ABE). Basically, the concept of cryptographic construction of CP- ABE is based on the bilinear maps.

The concept of bilinear maps is as follow.

Bilinear Maps

Let G_1 and G_2 be two multiplicative cyclic groups of prime order p and e be a bilinear map,

$e : G_1 \times G_1 \rightarrow G_2$. Let g be a generator of G_1 . Let H: $\{0,1\}^* \rightarrow G_1$ be a hash function that the security model is in random oracle.

The bilinear map e has the following properties:

1. Bilinearity: for all $u, v \in G_1$ and a, b $\in Z_p$, $e(u^a, v^b) = e(u, v)^{ab}$
2. Non-degeneracy: $e(g, g) \neq 1$.

Definition 1: (Access Structure [8]). Let $\{P_1,P_2,\ldots,P_n\}$ be a set of parties. A collection $A \subseteq 2^{\{P1,P2,\ldots,Pn\}}$ is monotone if $\forall B, C$: if $B \in A$ and $B \subseteq C$ then $C \in A$. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection) A of non-empty subsets of $\{P_1,P_2,\ldots,P_n\}$, i.e., $A \subseteq 2^{\{P1,P2,\ldots,Pn\}} \setminus \{\emptyset\}$. The sets in A are called the authorized sets, and the sets not in A are called the unauthorized sets.

In the context of CP-ABE, a set of parties is taken by the attributes. Thus the access structure A consists of the authorized sets of attributes.

IV. Our Proposed Big Data Cloud Access Control

4.1 System Model

Figure 1 shows a big data cloud access control architecture which portrays the process data from several sources to acquire the analytical data files which are stored in the cloud server.

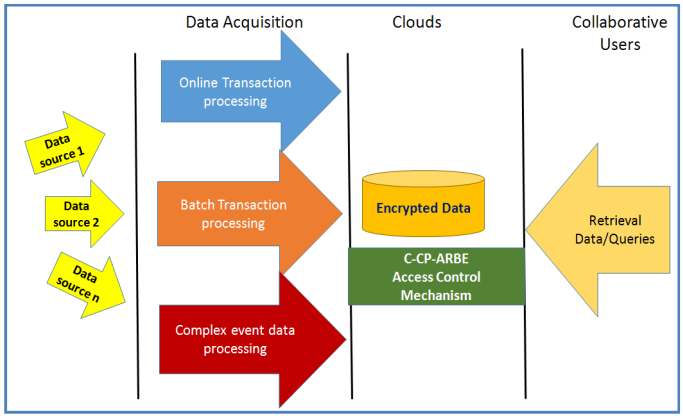

Fig. 1. Big Data Access Control Architecture in Cloud Computing

Basically, big data used for predictive analytics and data mining queries are obtained from the integration, normalization, and processing from multiple sources. Data generated from several sources usually contain various formats such as text, image, XML, etc. They are exposed to the big data processing software with very high velocity and huge volume. Due to the requirement of high efficiency in managing data and delivering service to users, cloud computing is generally employed to house the big data software and data storage. Security and privacy management thus comes to play as one of crucial requirements for cloud adoption. Also, the performance of data processing and data retrieval is also very important to be considered for the design of security solution.

The scope of this paper focuses on securing processed data files stored in the cloud. We assume that data providers upload their transaction or processed data into the cloud. They also specify the access control policy to regulate how the users gain access the particular resource and what privilege they have over the resource.

We propose the cryptographic access control mechanism to preserve the privacy of analytical data files that will be accessed by the users. In our proposed architecture, we extend our access control system called a Collaborative Ciphertext Policy Attribute Role-based Encryption (C-CP-ARBE) [15] to provide an expressive, flexible, and fine-grained access control for big data outsourced in the cloud. Our C-CP-ARBE is based on the integration of RBAC into CP-ABE and it consists of a collection of algorithms supporting authentication, authorization, and policy management functions for multi-authority cloud system. In terms of the performance of cryptographic operations, we embed symmetric encryption to encrypt compressed big data files and this makes big data encryption and decryption process more practical.

4.2 C-CP-ARBE Model

In this section, we describe basic definitions of our C-CP-ARBE model.

Definition 2 :User (U), Role (R), Attributes (attr), and Permission (P)

- User (U) is a subject who requests to request to access (read or write) the data outsourced by the data owner in the cloud. Each user is assigned the set of attributes with respect to his/her role by the attribute authority.
- Attributes(Attr) are a set of attributes used to characterize the user and associated to the particular attribute "role". A set of attributes is issued by attribute authority(AA).
- Role (R) is a super set of attribute where users and respective attributes are assigned to.
- Permission(P) is an action or privilege having value read (r) and write(w).

Definition 3: Access Control Policy (ACP)
ACP is a tree-based structure. Let ACP T is a tree represent the access structure in C-CP-ARBE. Each non-leaf node of the ACP tree represents the Role node and threshold gate where the Role node is a parent of threshold gate node. The threshold gate rule is the same as access tree of CP-ABE. We denote the parent of the children node x in the tree by parent(x). Thus, the parent of leaf node x is the pair of {Role node, threshold gate}. The function attr(x) is defined only x in a leaf node of the tree.

To provide a fine-grained access control, we introduce a special attribute "privilege" as an extended leaf (EL) node of the ACP T in order to identify the read or write privilege of the role. However, the EL is not used as a part of encryption by ACP while the parent of EL is considered a real leaf node.

As seen from Fig.2, our policy accommodates the expressive structure of policy specification that contains roles (medical doctor, nurse), attributes, and privileges of each role. The policy also allows users from other domains such as Hospital B's doctors who are a co-worker with Hospital A to access treatment records. The policy can be expressed by boolean operator AND, OR, and K-out of-N.

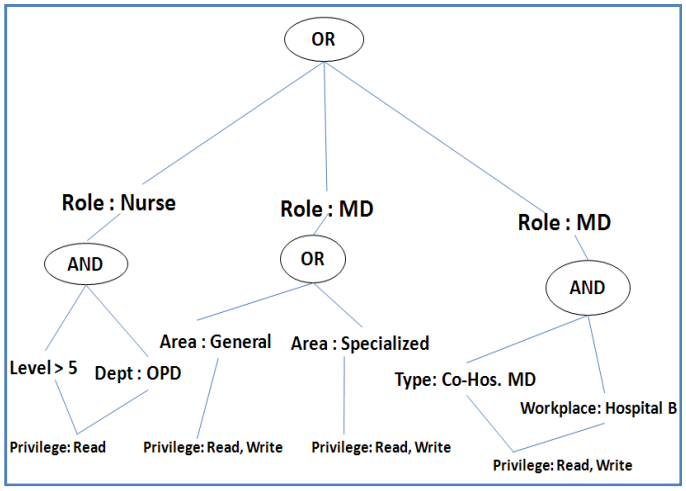

Fig.2. Access Control Policy of patient treatment record.

According to our access policy structure, data owners or administrators are able to manage the policy rules easily and conveniently. Besides, the policy can dynamically enforce the specific privilege access to users based on the role.

4.3 Key Management

In our system, there are three major groups of user key: PKI key pair, user decryption keys, and symmetric key. First, PKI key pair is a pair of public key and private key issued by the certification authority (CA). AAs, data owners, and users need to register for X.509 certificate from the CA for authentication, key encryption, and digital signing. Second key type is user decryption key (UDK) which is generated by attribute authorities (as of CP-ABE scheme) and it is given the user upon each service access. UDK is constructed from the user attributes and used to decrypt the ciphertext. Another key type is symmetric encryption key. We have two symmetric keys including SymKey used for data encryption in the inner layer and a secret seal (SS) used for outer layer encryption. SS is used to encrypt the ciphertext as another additional layer of encryption. For the SymKey, it is a session key generated per data encryption time. It is used to encrypt the data files and it will be concatenated with the ciphertext to be encrypted by the ACP. SymKey enables the encryption of big data to be done faster rather than using the CP-ABE directly encrypts the data. Our design is to reduce key management and provide single key binding for encryption. Only users having user decryption key (UDK) can decrypt the ciphertext encrypted by CP-ABE policy and the SymKey will be obtained and will be used to decrypt the encrypted data at the final stage.

Basically, each UDK, and SS are encrypted by user's public key. Then, the encrypted UDK (EDK), and encrypted secret seal (ESS) are obtained and stored in the cloud. We use a graph structure called user decryption key graph (UDKG) to structurally store EKD and ESS which are mapped to the individual user. Hence, users only keep their private key used to decrypt these keys and do not need to hold multiple UDKs if they have several ones.

4.4 Extended C-CP-ARBE Construct

The extended C-CP-ARBE consists of the following algorithms.

1. **Create Attribute Authority**(AA_{id})→PK_{aid}, SK_{aid}, $PK_{x.aid}$, The algorithm takes the attribute authority ID(AA_{id}) as input. It outputs the authority public key (public parameter) PK_{aid}, SecretKey SK_{aid}, and public attribute keys $PK_{x.aid}$ for all attributes issued by the AA_{aid}. AA has also a key pairs ($PubK_{aid}$, $PrivK_{aid}$) generated by CA.
2. **UserRegister**(U_{id}, $Cert_{uid.caSignature}$) → UL',. The userRegister algorithm takes input as userID and user's certificate issued by a trusted CA. If the user is authorized, the user list associated to particular role is updated.
3. **CreatRole**(SK_{aid},R_{ID}, *Set of* U_{id} →MK_R, ,UL. The CreateRole algorithm takes as inputs attribute authority's secret key SK_{aid} , RoleID R_{ID}, and set of users U_{id} who belong to the role. It returns master key of role MK_R, and user list *UL*.
4. **Create GroupRole parameter** GRP ($PrivK_{aaid}$,set of $R_1(U_1,U_2,..U_n)$, $R_2(U_1,U_2,..U_n)$,…, $R_n(U_1,U_2,..U_n)$ → GRP) The Create GroupRole parameter algorithm takes as input user id of each role R_{id} and concatenate the value sets and returns the GRP. Then, the GRP is signed (encrypted) by AA's private key and it will be updated when there is any adding or revoking of user to/from any role. GRP is encrypted with user private key and will be stored in UDKG.
5. **UserKeyGen**($S_{uid,aid}$, SK_{aid}, $Cert_{uid}$)→ $EDK_{uid,aid}$, RDK_{aid}. The KeyGen algorithm takes continuous two steps: (1) takes input as set of attributes $S_{uid,aid}$, attribute authority's secret key SK_{aid}, and public key certificate of users $Cert_{uid}$, then it returns the set of user decryption keys UDK (2) a UDK is encrypted with the global public key of the user and outputs the set of encrypted decryption keys $EDK_{uid,aid}$.
6. **UpdateUDKGraph** (U_{id}, $EDK_{uid,aid}$) →UDKG' . This algorithm takes U_{id} and encrypted decryption key (EDK) to update the UDKG.
7. **Encrypt**($PK_{aid,}$ M, GRP, ACP, SymKey) →SCT. Before encrypting, the file is compressed. Then, the encryption algorithm performs three continuous steps as followings:
   (1) Inner layer: the algorithm takes as inputs SymKey, and data M. Then it returns a ciphertext CT.
   (2) Middle layer, the algorithm takes ACP to encrypt the CT and SymKey. It returns encrypted ciphertext ECT.

(3) Outer layer, the algorithm takes GRP to encrypt ECT and returns sealed CT (SCT).

8. **Decrypt**(PK_{aid}, SCT, GSK_{uid}, $EDK_{uid,}$) → M. The algorithm takes user's global secret key GSK_{uid} to decrypt session key (generated from GRP), and $EDK_{uid,}$. Then the algorithm returns ECT, and UDK_{id}. Then, the algorithm takes UDK_{id} to decrypt ECT and returns SymKey and CT. Finally a CT is decrypted by the SymKey and the compressed M is obtained and it will be then uncompressed to return original M.

9. **RevokeUser**($U_{id,aid}$, SK_{aid} , UL_{Rid})→$UL_{Rid'}$, $GRP_{aid'}$. The RevokeUser algorithm takes $User_{uid,aid}$ and AA's secret key, and user list UL of the role having user revoked as inputs. The secret key of attribute authority is used to sign the revoked request and the revoked user is removed from the UL. Then, it returns updated UL. Finally, GRP is updated and the revoked user is deleted from UDKG.

V. ANALYSIS OF OUR PROPOSED SCHEME

1. Optimization of Key Management

Even though we introduce one more symmetric encryption layer (first layer), it does not provide a significant cost in overall encryption process. Big data size is huge; we avoid using CP-ABE to encrypt it directly. We consider compressing it first and then using symmetric encryption for fast encryption. Then, the ciphertext is managed flexibly with the fast symmetric encryption and then the CP-ABE is applied. Since the symmetric key size is small, the encryption process is instantly performed upon a single key source.

Our scheme also provides zero cost for key distributions. Keys necessary for decryption (SymKey, UDK, and GRP) are all encrypted and stored in the cloud server. Therefore, there are no decryption keys distributed to users. This process is transparent to users. Upon the requests for data access, users can get all keys for decryption automatically. Our scheme is very suitable for collaborative scenarios in which users may hold several decryption keys to access files shared by multiple data owners.

2. Efficient User Revocation Management

Our scheme does not require file re-encryption and user decryption key update when there is revocation of any users. Our algorithm only needs to update the global role parameters (GRP) and apply a new GRP to re-encrypt the ECT. Accordingly, revoked users cannot use their existing GRP to decrypt the FCT. In addition, once the user is revoked, their certificate is updated in the certificate revocation list (CRL) and they cannot use their certificate for further authentication. As a consequence, our scheme provides practical solution for user revocation management for big data outsourcing scenario.

3. Security Analysis

Collusion Resistance

In our scheme, it should be not possible for different users to combine their attribute sets and in order to gain access the resources. Even though the adversaries use an attribute having the identical name and value (position: doctor) from different attribute authorities, they cannot collude with the legal user in the authority shared the target resource. This is because each attribute has its own public key, unique id together the tag of attribute authority id. Each attribute is also signed by its attribute authority. In addition, a random number *r* constituted in the UserKeyGen algorithm enables the key constructed from a set of attributes is technically distinguishable and non-substitutable. This prevents the collusion attack from the adversaries.

Our proposed model is proven to be secure under the general random security model [15].

Forward and Backward security

Since our scheme integrates RBAC into the CP-ABE model, users and attributes are thus flexibly managed For newly joined users, they can access any encrypted files as long as they are enrolled by the trusted authorities and have access right (qualified set of attributes) specified in the policy. They are also granted decryption key based on their roles and attributes while the secret seal used for outer layer decryption will be encrypted by user's public key and sent by their data owners. Hence, new users are able to access files even the file is created before they join. This satisfies forward security.

For the backward security, our revocation algorithm support user revocation and guarantees that the revoked users cannot use their decryption key to access the content of any files even they have ever accessed.

VI. EXPERIMENT

We conduct the performance evaluation to demonstrate that efficiency of the proposed three-layer encryption as an enhancement of our C-CP-ARBE scheme. We compare encryption time and decryption time taken by our original C-CP-ARBE and the improved C-CP-ARBE (supporting big data version).

In our experiments, our system service is developed by PHP and Java language run on the Apache Sever. For the key management server, we use Open SSL as a core PKI service to generate key pairs to users and system entities. The service is run on Intel Xeon Processor X5650, 2.66GHz with Ubuntu Linux.

The test case is simulated by using a policy containing 10 leaf-nodes for data encryption. We use the policy to encrypt 10 MB, 50 MB, 500 MB, and 1 GB data files and measure the total time used for encryption.

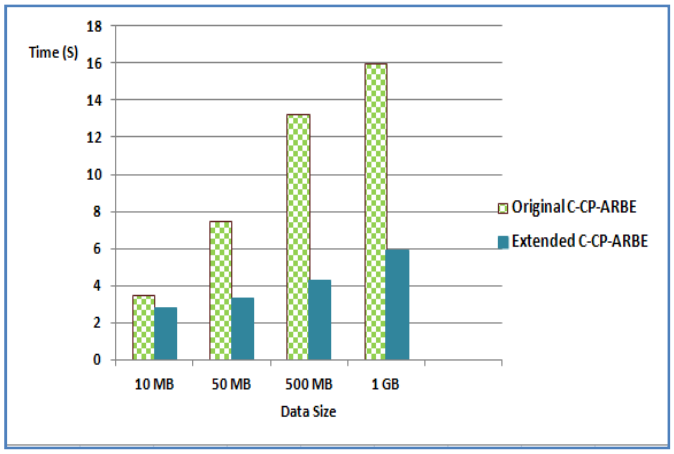

Fig. 3. Encryption Time

For the decryption time, we vary number of attributes contained in the user decryption key for decrypting 10MB file. In CP-ABE, decryption cost is generally subject to the number of attributes and size of the file for decryption.

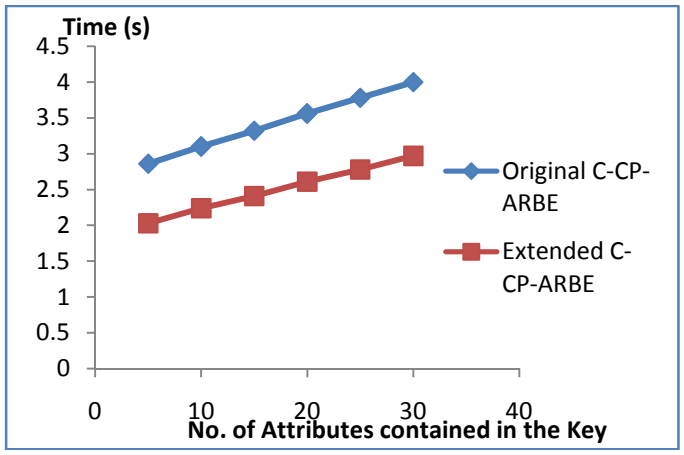

Fig. 4. Decryption Time

As seen from Fig. 3 and Fig. 4, the results show that the extended C-CP-ARBE takes less time than the original C-CP-ARBE for big data encryption and decryption. This is because the symmetric encryption significantly reduces encryption and decryption cost. However, this version is only suitable for big data processing. For general data files, our original version possesses a more agile and flexible key management as it contains only two encryption steps. When the data size increases, the extended version tends to give small change while the original version is more susceptible to the size of data file.

VI. CONCLUSION AND FUTURE WORK

We have proposed a privacy-preserving access control model for big data cloud scenario. Our access control model: C-CP-ARBE is extended to incorporate another encryption layer to provide efficient encryption for big data. Finally, we conduct the experiments to evaluate the performance of our proposed scheme. The results reveal that our extended version provides more efficient and practical deployment in supporting access control for big data outsourced in the cloud. For future works, we will work on a large scale of experiment for evaluating both bigger data size and performance of the concurrent access.

ACKNOWLEDGMENT

This research has been supported in part by Otsuka Scholarship.

REFERENCES

[1] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "Encryption Policies for Regulating Access to Outsourced Data", in ACM Transactions on Database Systems (TODS), April, 2010

[2] Shucheng Yu, Cong Wang, Kui Ren, and Wenjing Lou, "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing," IEEE INFOCOM 2010, San Diego, CA, March, 2010

[3] Zhiguo Wan, Jun-e Liu, Robert H. Deng: HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing. IEEE Transactions on Information Forensics and Security 7(2): 743-754 (2012)

[4] Kan Yang, Xiaohua Jia, Kui Ren, Bo Zhang, Ruitao Xie: Expressive, Efficient, and Revocable Data Access Control for Multi-Authority Cloud Storage. IEEE Trans. Parallel Distrib. Syst. 25(7): 1735-1744 (2014).

[5] Ming Li, Shucheng Yu, Yao Zheng, Kui Ren, and Wenjing Lou, "Scalable and Secure Sharing of Personal Health Records in Cloud Computing using Attribute-based Encryption," IEEE Transactions on Parallel and Distributed Systems, 2012

[6] Kan Yang, Xiaohua Jia, Kui Ren, Bo Zhang, Ruitao Xie: Expressive, Efficient, and Revocable Data Access Control for Multi-Authority Cloud Storage. IEEE Trans. Parallel Distrib. Syst. 25(7): 1735-1744 (2014)

[7] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-Based Encryption for Fine-grained Access Control of Encrypted Data. In Proc. of CCS'06, Alexandria, Virginia, USA, 2006.

[8] Bethencourt, J., Sahai, A. And Waters, B., Ciphertext-policy Attribute-based Encryption, IEEE Symposium of Security and privacy, Oakland, CA, USA, May 20-23, Los Alamitos, 2007.

[9] L.Cheung and C. Newport: Provably secure ciphertext policy ABE. ACM Conference on Computer and Communications Security 2007, USA 2007

[10] A.Sahai and B. Waters, Fuzzy Identity Based Encryption, In Advances in Cryptology – Eurocrypt, volume3494 of LNCS, pages 457-473. Springer, 2005.

[11] Katz, J., Sahai, A., Waters, B., Predicate encryption supporting disjunctions, polynomial equations, and inner products. Eurocrypt 2008. LNCS, Vol 4965.pp 146-162, Springer, heidelberg 2008.

[12] S.Fugkeaw, Achieving privacy and security in multi-owner data outsourcing, IEEE International Conference on Digital and Information Management (ICDIM 2012), Macau, August 2012

[13] L. Zhou, V. Varadharajan, and M. Hitchens, Enforcing Role-based Access Control for Secure Data Storage in the Cloud, The Computer Journal, Vol. 54 No.10, 2011.

[14] Lan Zhou, Vijay Varadharajan, Michael Hitchens: Achieving Secure Role-Based Access Control on Encrypted Data in Cloud Storage. IEEE Transactions on Information Forensics and Security 8(12): 1947-1960, 2013.

[15] Somchart Fugkeaw and Hiroyuki Sato, An extended CP-ABE based Access control model for data outsourced in the cloud, IEEE International Workshop on Middleware for Cyber Security, Cloud Computing and Internetworking (MidCCI 2015), Taichung, Taiwan, July 1-5,2015.

[16] Wenrong Zeng, Yuhao Yang, Bo Luo, Access Control for Big Data using Data Content, IEEE International Conference on Big Data, 2013