IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2016

Dynamic Hierarchical Attribute Set-Based ECC Encryption for Secure Cloud Storage

Sadia Syed[1], Dr. M. Ussenaiah[2]

Scholar, Dept of Computer Science, Vikrama Simhapuri University, Nellore, A.P[1]

Assistant Professor, Dept of Computer Science, Vikrama Simhapuri University, Nellore, A.P[2]

Abstract: Accessing Data and control it is an efficient way to provide data security in the cloud. But in untrusted cloud servers, the data storage and retrival control becomes a challenging in cloud storage systems. Existing access control schemes are not satisfactory in cloud storage systems, because they either produce multiple encrypted copies of the same data or require a fully trusted cloud server. Since this new computing technology requires users to entrust their precious data to cloud providers, in that sense the security and privacy concerns on outsourced data is increasing. Attribute-based encryption (ABE) provides a mechanism for typical access control over encrypted data. But in most ABE systems, the cipher text size and the decryption overhead, which causes for the complexity of the access policy, and also one single Trusted Authority (TA) and Cipher text Policy (CP-ABE) are unable to manage multiple group owners for encryption process and access policy. For achieving scalability, flexibility, and fine-grained access control of data in cloud, we propose Hierarchical Attribute-Set-Based Encryption (HASBE). This is an extension of cipher text-policy Attribute-Set-Based Encryption (ASBE) with a hierarchical structure of users with compound attributes. Through which we will achieve Sophisticated, flexible and scalable data access control with the help of compound attributes of HASBE.In addition to that we are using ECC Elliptic Curve Cryptography Instead of RSA in ABE. This scheme achieves scalability due to its hierarchical structure, but also achieve flexibility and fine-grained access control. In addition, HASBE with ECC employs idol access time to deal with user revocation more efficiently than existing schemes. We formally proved the security of HASBE based on security of the cipher text-policy attribute-based encryption (CP-ABE) scheme by analyzing its performance and computational complexity. We have implemented our scheme and proved that it is both efficient and flexible in dealing with access control for outsourced data in cloud with comprehensive experiments.

Keywords: Cloud Computing, Cloud Data Storage Security, ABE, HASBE, Access Policies, ECC.

1. INTRODUCTION

CLOUD computing is a new computing paradigm is built on virtualization, utility computing, parallel and distributed computing, service-oriented architecture. In the recent years, cloud computing has became as one of the most effective influential paradigms in the computer and computing industry, and has grabbed extensive attention from both educational bodies and industry. Access of customer information to high-level executives of the company only. In such situations access control of sensitive data is either required by based on company regulations. In this, we propose a hierarchical attribute-set-based Encryption (HASBE) scheme for access data from cloud. HASBE extends the cipher text-policy attribute-set-based encryption (CP-ASBE, or ASBE for short) scheme with a hierarchical structure of users, to achieve secure, scalable, flexible and fine-grained access control. First, we have shown HASBE extends the ASBE algorithm with a hierarchical structure to improve security, scalability and flexibility at the same time inherits the feature of fine access control of ASBE. Second, we demonstrate how to implement a full-fledged access control scheme for cloud computing based on HASBE. The scheme provides full support for hierarchical user grant, file creation, file deletion, and user revocation in cloud computing. Third, we formally shown the security of the proposed scheme based on the security of the CP-ABE scheme to analyze its performance in terms of computational overhead. Lastly, we implement HASBE and conduct comprehensive experiments for performance evaluation, and our experiments demonstrate that HASBE has satisfactory performance.

2. RELATED WORK

We have studied attribute-based encryption (ABE), and provide a brief overview of the ASBE scheme. After that, we examine existing access control schemes based on ABE.

Attribute Based Encryption: Attribute-based encryption (ABE) is a recent approach which reconsiders concept of public-key cryptography. Usually In traditional public-key cryptography, a message is encrypted by the receiver’s public-key. Identity-based cryptography, identity-based encryption (IBE) changed the traditional understanding of public-key cryptography making public-key to be an arbitrary string, say suppose the email address of the receiver. ABE defines the identity not atomic but as a set

DOI 10.17148/IJARCCE.2016.5923

of attributes,Ex: roles, files and messages can be encrypted with respect to subsets of attributes (key-policy ABE - KP-ABE) or policies defined over a set of attributes (ciphertext-policy ABE - CP-ABE). The key element is, it should only be able to decrypt a ciphertext if the person having a key for "matching attributes" where user keys are issued by trusted party.

In Attribute-based Encryption (ABE) scheme, attributes are meant for important role. Attributes are to be differentiated to generate a public key for encrypting data and it's been used as an access policy to control users' accessing. The access policy is cut into two as key policy & cipher text policy. The key-policy attribute are used for defining encrypting data and policy implemented in user`s key, and the cipher text policy is actual access structure on the cipher text. This access structure can also be in either monotonic or non-monotonic formats.

Setup, KeyGen, Encrypt, and Decrypt:
Setup (d). In this, d is the depth of key structure. It take input as depth parameter d. it will generate public key PK and master secret key MK.
Key Gen (MK, u, A): Accepts input the master secret key MK, the recognize user u, and key structure A. It produces secret key SKu for user u.
Encrypt (PK, M, T): Accepts the input public key PK, a message M, and an access tree T. It generates a cipher text CT.
Decrypt (CT, SKu): accepts input as cipher text CT and a secret key SKu for user u. It generate a message m. If the key structure A associated with the secret key SKu satisfies the access tree T, along with the cipher text CT, then m is the actual correct message M. Or else, m going to be null.
The specified algorithms are equal to those of CP-ABE, apart from to support recursive key structure. Public key and the master key of ASBE are extended from CPABE to support recursive key structure.
Access Tree
Let's think that the cloud server provider is untrusted in that senses it may collude with malicious users to steal file contents stored in the cloud for its own benefit...

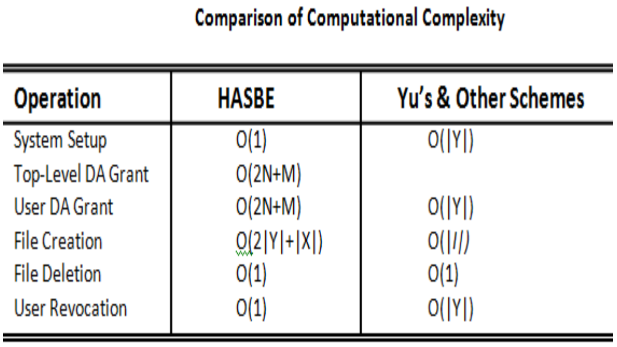
OUR CONSTRUCTION:
Now we present our HASBE scheme, which is an extension of the ABE algorithm with a hierarchical user structure. We after show how HASBE is applied for hierarchical user grant; data file creation, file access, user revocation, and file deletion.

Our existing solution applies cryptographic methods by disclosing data decryption keys only to authorize users.

- ✓ These solutions will introduce a heavy computation overhead on the data owner for key distribution and data management when data access control is required, and thus do not scale well.
- ✓ On the one hand, the workloads often contain very sensitive information, such as the business financial records, proprietary research data, or personally identifiable finance, health information etc.
- ✓ On the other hand, the operational details within the cloud are not transparent to customers.

Access data control and encrypting the data as two different components, which may cause redundant operations with interaction between them. Therefore, in this, we propose a Hierarchical and Elliptic Curve Cryptography scheme to encrypt Cloud data for user access control. This scheme with the features of hierarchy and high security is preferred for access control in Cloud. The hybrid feature denotes Elliptic Curve Cryptography (ECC) for a fine grained control of user access. The plain data in Cloud are encrypted by ECC which is a public-key encryption algorithm with high performance in security. Our access control scheme is implemented as part of cloud storage service. It allows authorized users can access the corresponding data in Cloud. The simulation experiments show that in comparison with other schemes, Our ECC-based scheme can achieve better performance in both security and efficiency of Cloud data access.

Elliptical curve cryptography (ECC):
Elliptical curve cryptography (ECC) is a public key encryption technique based on elliptic curve theory that can be used to create smaller, faster and more efficient cryptographic keys. ECC generates keys through the properties of the elliptic curve equation instead of the usual traditional method of generation as the product output of very large prime numbers. The technology can be used in combination with most public key encryption methods, such as RSA, and Diffie-Hellman. According to some researchers, ECC can achieve a level of security with a 164-bit key that other systems require a 1,024-bit key to achieve. Because ECC helps to establish equivalent security with lower computing power and battery resource usage.



Comparison of Computational Complexity

Operation	HASBE	Yu's & Other Schemes
System Setup	O(1)	O(|Y|)
Top-Level DA Grant	O(2N+M)	
User DA Grant	O(2N+M)	O(|Y|)
File Creation	O(2|Y|+|X|)	O(|I|)
File Deletion	O(1)	O(1)
User Revocation	O(1)	O(|Y|)

Performance Analysis:
System Setup:
The total computational complexity of System Setup is O (1). Where the computational complexity of Top-Level Authority Domain Grant operation is O (2N+M). The computational complexity is O(2N + M). New File Creation: The computation complexity of New File Creation is O (2|Y| + |X|).

DOI 10.17148/IJARCCE.2016.5923

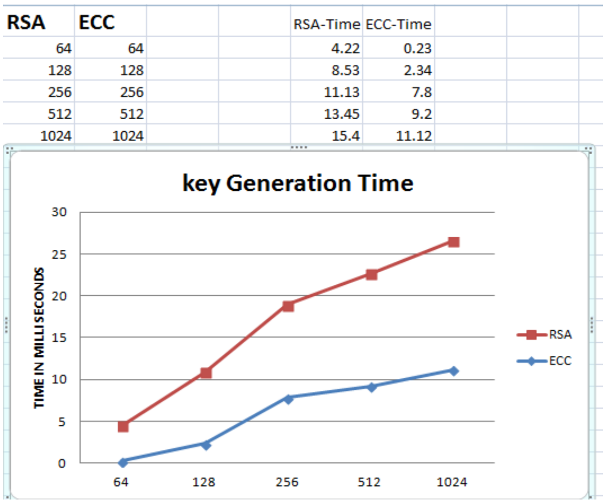
3. IMPLEMENTATION

Hasbe-setup: here it will Generates a public key PK and a master key MKo.
Hasbe-keygen: Given PK and MKo, produces a private key for a key structure. The key structure with depth 1 or 2 is supported.
Hasbe-keydel: Given PK and MKi of DAi, delegates some areas of DAi's private keys to its new user or DAi+1 in its domain. The key is equivalent to producing private keys by the root authority.
Hasbe-keyup: Given PK, the private key, the new attribute and the subset, produces a brand new private key which contains the new attribute.
Hasbe-enc: Given PK, encrypts a file by using an access tree policy described in a policy language.
Hasbe-rec: Given PK, a private key along with encrypted file, re-encrypt the file.consider that the private key could be able to decrypt the encrypted file.

Implementation:
We have implemented a Dynamic and multilevel HASBE toolkit using cpabe toolkit (http://acsc.csl.sri.com/cpabe/) developed for CP-ABE which uses the Pairing-Based Cryptography library (http://crypto.stanford.edu/pbc/).like the cpape toolkit, our toolkit also provides a number of command line tools as follows:
Hasbe-setup: will Generates a public key PK and a master key MK0.

- **Hasbe-keygen:** provide PK and MK0, will produce a private key for a key structure. The key structure with depth 1 or 2 will supported.
- **Hasbe-keydel:** provide PK and MKi of DA, delegates some parts of DA 's private keys to the new user or DA to its domain. The delegated key is equal to producing private keys by the root Authority.
- **Hasbe-keyup:** provide PK, which is the private key, the new attribute and the subset, produces a new private key which contains the new attribute.
- **Hasbe-enc:** provide PK, encrypts a file under an access tree policy specified in a policy language.
- **Hasbe-dec:** provide a private key, which decrypts a file.
- **Hasbe-rec:** provide PK, the private key and with encrypted file, re-encrypt that file. consider that the private key could be able to decrypt the encrypted file.

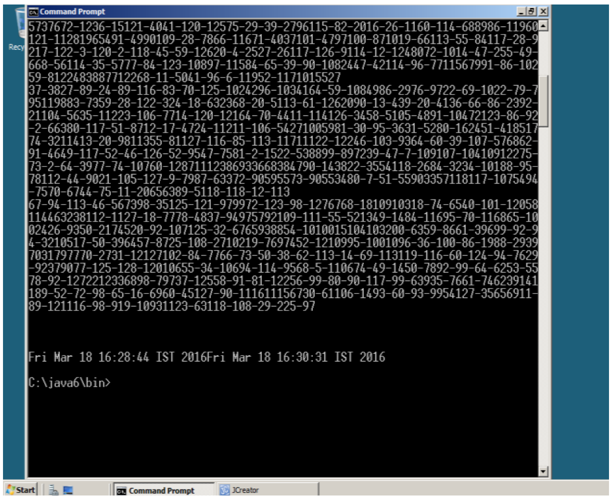
SECURITY PROOF OF SCHEME:
The security view of HSBE-ECC-based access control scheme depends on few core encryption components: Attribute Set, AES, B+Tree and ECC. Which will clarify the security bottlenecks; we analysis each component and produced security proof of the scheme. Which shows the security components model of HSBE-ECC. There are two storage facilities in the model, the trust center and data center. The trust center carries the AES encrypted private key, level keys and the relationship graph consists of KUID. The data center going to stores ECC encrypted data associated with data relevant tags which corresponding to the KUTDs of owners in relationship graph.



RSA	ECC	RSA-Time	ECC-Time
64	64	4.22	0.23
128	128	8.53	2.34
256	256	11.13	7.8
512	512	13.45	9.2
1024	1024	15.4	11.12

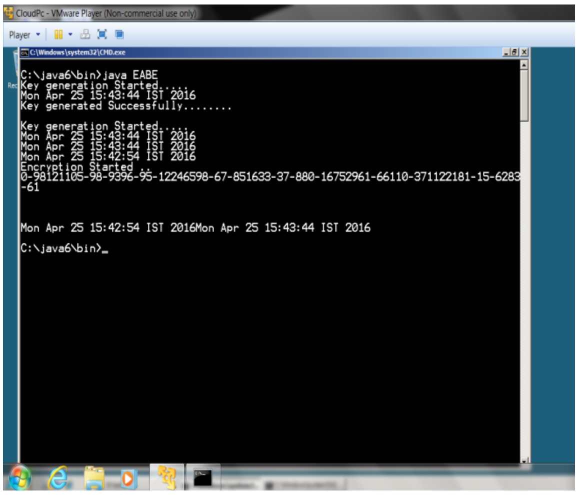
IMPLEMENTATION RESULTS:

Normal ABE Encryption:



```
5737672-1236-15121-4041-120-12575-29-39-2796115-82-2016-26-1160-114-688986-11960
121-11281965491-4990109-28-7866-11671-4037101-4797100-871019-66113-55-84117-28-9
217-122-3-120-2-118-45-59-12620-4-2527-26117-126-9114-12-1248072-1014-47-255-49-
668-56114-35-5777-84-123-10897-11584-65-39-90-1082447-42114-96-7711567991-86-102
59-8122483887712268-11-5041-96-6-11952-1171015527
37-3827-89-24-89-116-83-70-125-1024296-1034164-59-1084986-2976-9722-69-1022-79-7
95119883-7359-28-122-324-18-632368-20-5113-61-1262090-13-439-20-4136-66-86-2392-
21104-5635-11223-106-7714-120-12164-70-4411-114126-3458-5105-4891-10472123-86-92
-2-66380-117-51-8712-17-4724-11211-106-54271005981-30-95-3631-5280-162451-418517
74-3211413-20-9811355-81127-116-85-113-11711122-12246-103-9364-60-39-107-576862-
91-4649-117-52-46-126-52-9547-7581-2-1522-538899-897239-47-7-109107-10410912275-
73-2-64-3977-74-10760-12871112386933668384790-143822-3554118-2684-3234-10188-95-
78112-44-9021-105-127-9-7987-63372-90595573-90553480-7-51-55903357118117-1075494
-7570-6744-75-11-20656389-5118-118-12-113
67-94-113-46-567398-35125-121-979972-123-98-1276768-1810910318-74-6540-101-12058
11446323811 2-1127-18-7778-4837-94975792109-111-55-521349-1484-11695-70-116865-10
02426-9350-2174520-92-107125-32-6765938854-1010015104103200-6359-8661-39699-92-9
4-3210517-50-396457-8725-108-2710219-7697452-1210995-1001096-36-100-86-1988-2939
7031797770-2731-12127102-84-7766-73-50-38-62-113-14-69-113119-116-60-124-94-7629
-92379077-125-128-12010655-34-10694-114-9568-5-110674-49-1450-7892-99-64-6253-55
78-92-1272212336898-79737-12558-91-81-12256-99-80-90-117-99-63935-7661-746239141
189-52-72-98-65-16-6960-45127-90-111611156730-61106-1493-60-93-9954127-35656911-
89-121116-98-919-10931123-63118-108-29-225-97



Fri Mar 18 16:28:44 IST 2016Fri Mar 18 16:30:31 IST 2016

C:\java6\bin>
```

OUR HSBE-ECC SCHEME:

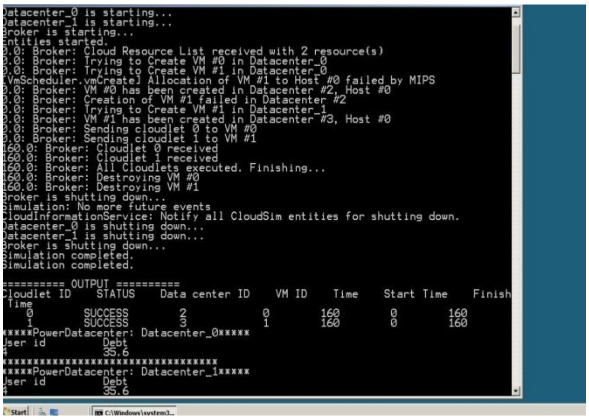


CloudPc - VMware Player (Non-commercial use only)
Player
C:\Windows\system32\CMD.exe

```
C:\java6\bin>java EABE
Key generation Started.....
Mon Apr 25 15:43:44 IST 2016
Key generated Successfully........

Key generation Started.....
Mon Apr 25 15:43:44 IST 2016
Mon Apr 25 15:43:44 IST 2016
Mon Apr 25 15:42:54 IST 2016
Encryption Started ..
0-98121105-98-9396-95-12246598-67-851633-37-880-16752961-66110-371122181-15-6283
-61



Mon Apr 25 15:42:54 IST 2016Mon Apr 25 15:43:44 IST 2016

C:\java6\bin>_
```

DOI 10.17148/IJARCCE.2016.5923

SIMULATION RESULTS



```
Datacenter_0 is starting...
Datacenter_1 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 2 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
[VmScheduler.vmCreate] Allocation of VM #1 to Host #0 failed by MIPS
0.0: Broker: VM #0 has been created in Datacenter #2, Host #0
0.0: Broker: Creation of VM #1 failed in Datacenter #2
0.0: Broker: Trying to Create VM #1 in Datacenter_1
0.0: Broker: VM #1 has been created in Datacenter #3, Host #0
0.0: Broker: Sending cloudlet 0 to VM #0
0.0: Broker: Sending cloudlet 1 to VM #1
160.0: Broker: Cloudlet 0 received
160.0: Broker: Cloudlet 1 received
160.0: Broker: All Cloudlets executed. Finishing...
160.0: Broker: Destroying VM #0
160.0: Broker: Destroying VM #1
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Datacenter_1 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID    Time    Start Time    Finish
 Time
    0        SUCCESS        2             0        160        0          160
    1        SUCCESS        3             1        160        0          160
*****PowerDatacenter: Datacenter_0*****
User id          Debt
4                35.6
**********************************
*****PowerDatacenter: Datacenter_1*****
User id          Debt
4                35.6
```

4. CONCLUSION

In this, we have introduced the HASBE scheme for realizing flexible, scalable and fine-grained accessing control in area of cloud computing. The HASBE scheme will incorporates a hierarchical structure and because of its Dynamic nature users can be gone through a delegation algorithm to ASBE. HASBE will not only supports compound attributes due to dynamic attribute set combinations, and also achieves efficient user revocation because of multiple value assignments of attributes. We formally proved the security of HASBE based on the security of CP-ABE. We implemented the proposed scheme, and conducted comprehensive performance analysis, evolution, that shows its efficiency and advantages over the existing schemes.

We also propose an effective hierarchical access control scheme based on HSBE-ECC algorithm, which integrates ECC, Attribute Set to authorize hierarchical users to accession to encrypted Cloud Data. AES will encrypt private key of ECC by using the key produced by user and user hierarchy. Through HSBE-ECC users can share their data with the authorized users. The simulated experiments show that this scheme achieves a good performance on the efficiency and security of Cloud data access.

REFERENCES

[1] K. J. Biba, Integrity Consideration for Secure Computer Systems The MITRE Corporation, Tech. Rep. 1977

[2] H. Harney, A Colgorve, and P D McDaniel, "Principles of policy in secure groups." In Proc. NDSS, San Diego, CA, 2001.

[3] P. D. McDaniel and A. Prakash, "Methods and limitations of security policy reconcilation," in Proc. IEEE Symp. Security and Priacy. Berkely, CA. 2002.

[4] R. Buyya, C. Shin Yeo, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and relatity for delivering computing as 5th utility." Future Generation Compu. Syst. Vol. 25. Pp 599-616, 2009.

[5] B. Barbara, "Salesforce.com Raising the level of networking," Inf. Today, vol 27. Pp 45-45, 2010..

[6] A. Ross, "Technical perspective: A chilly sense of security," Commun. ACM, vol. 52, pp. 90–90, 2009.

[7] D. E. Bell and L. J. LaPadula, Secure Computer Systems: Unified Exposition and Multics Interpretation The MITRE Corporation, Tech. Rep., 1976. [7] K. J. Biba,

DOI 10.17148/IJARCCE.2016.5923