also done by this method which includes role modifications, file updates and permission. Whereas in this method the data owner will have to perform all operations which leads to unrealistic and unsuitable in the public cloud.

3. Enhanced RBAC Model

The main role of RBAC is to simplify the management in the user permission. In case of the self-data protection the RBAC method describes the access, control policies which supports arbitrary constraints and specific data. The access policies are not specified by the data owners for their data at role level but can defines other constraints. All the requirement is done by the E_RBAC method. It supports inheritance, role assignment and constraints. In E_RBAC each data object is protected and the security requirement is needed to each object.

3.1. *Construction of E_RBAC*

In case of self-contained protection method for protecting the enterprise data in the public cloud the data has own access control policy that has flexible, fine grained and flexible role-based access control. In this method the user cannot authorize all by themselves and it still require policy verification. It builds the mechanism so the third party sever can be eliminated. Self-protection of the data are achieved by the encryption. The integrity of access control and encryption is provided by CP-ABE.

3.2. Expressing E_RBAC with ECP-ABE

To construct the E_RBAC two problems are solved. In the first case the support of ECP-ABE role assignment since it includes a role inheritance and expressed as extended attribute. The negative assignment is expressed as NOT operator is not suitable for extended node that is express as positive assignment. Expressing an E_RBAC access policy using extended tree in ECP-ABE because the policy model of ECP-ABE and E_RBAC is different. To solve the above problem, we define a threshold value in ECP-ABE and supports role assignment. A policy mapping is presented which will transform E_RBAC model to extended access tree.

3.2.1. Support role assignment

To implement the support role assignment in E_RBAC there is an improvement in policy expression of ECP-ABE. The threshold k in the node is extended and redefined for various requirements. So ECP-ABE is extended by assigning k different value.

3.2.2. Map E_RBAC with ECP-ABE access control tree

A mapping model is presented to transform E_RBAC access policy to ECP-ABE access tree. When the value of $k>0$ it indicates the threshold value of leaf node or internal node. If $k<0$ it indicates the operators.

The rules for mapping are as follows

(1) The environment constraint, user constraint and role assignment of E_RBAC corresponds to leaf nodes in ECP-ABE access tree and is shown in figure 1. The role inheritance is involved by role assignment and expressed in the form of extended leaf node. If the complex operator is included in environment and user constraint then it is expressed in the form of extended leaf node else it is expressed in leaf node.
(2) The AND, OR and threshold are the logical combination of E_RBAC role assignments, environment and user constraint which is in internal nodes of access tree.

The access policy of E_RBAC is explained as follows:

Policy(d_x) = ((role = sales – employee OR
(role != product – employee AND
security – level ≤ 5)) AND
10:00 ≥ time ≥ 12:00

In the internal nodes the AND and OR symbols are expressed. The constraints role = sales – employee OR role != product – employee is the role assignment and considered as the extended leaf node. The constraints security – level ≤ 5 and 10:00 ≥ time ≥ 12:00 indicates comparison operator and also considered as the extended leaf node. The E_RBAC maps the access tree is shown in the figure 1 and the symbols Λ and V represents AND and OR respectively.

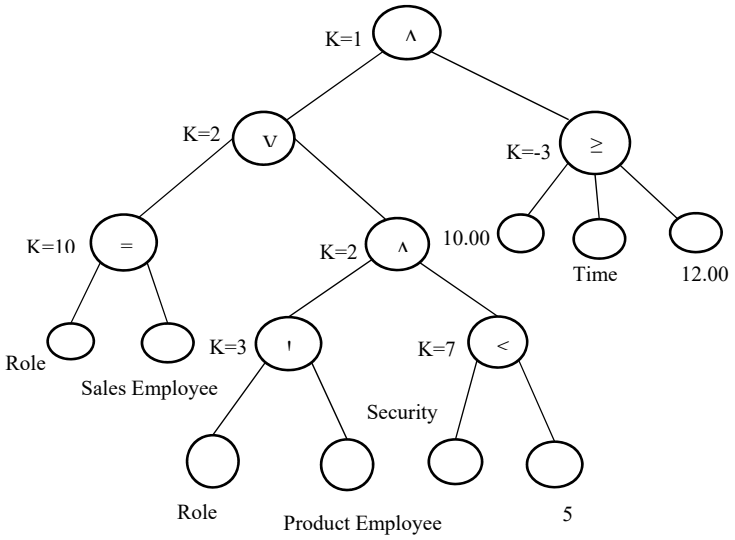


Figure 1. E_RBAC access control tree

3.3. E_RBAC Verification

ECP-ABE cannot have the ability for handling role inheritance. To overcome this limitation the attribute verification is modified during the key generation. The following algorithm explains the concept of verification. The attribute tree is added with the user information and the SKG maintains the copy of the attribute tree. In each role some operations are performed. Operator "=" checks the role of the user is equal to r in the attribute tree and if it is true it returns the attribute. If the operator "!=" then the SKG checks the role of the user is not equal to r. If the condition does not satisfy it returns the value null. Finally, PKG retrieves the attributes that is used for the generation of the secret key.

Input : Attribute Name, Value and Operator

Output: Group of String

Step 1: Expression expr(AN,AO,AV) retrieved from extended leaf node, the value AN, AO, AV denotes the name, operator and value of the attribute.

Step 2: The attribute set W is traverse and the produces the name of the attribute AN and value V'.

Step 3: Let AVsize and AOsize denotes the size of array AV and AO respectively.

Step 4: if AN== "role" then AOsize = 1 and AVsize = 1

```
if AO[1] == "=" && V' ≥ V[1] then
   convert expr(AN.AO.AV) into string AS == AV[1]
return AS;
else if AO[1] == "!=" &&V' ≤ V[1] then
    convert expr(AN.AO.AV) into string AS != AV[1]
return AS;
else null;
end if
else if AOsize ==1 && AVsize == 1 then
    Evaluate expr(AV'.AO[1].AV[1]);
 if expr(AV'.AO[1].AV[1]) is TRUE then
     convert expr(AN.AO.AV) into string AS = "AN. AO[1].AV[1]";
return AS;
else null;
end if
else if AOsize ==2 && AVsize == 2 then
    Evaluate expr(AV'.AO[1].AV[1]);
```

if expr(AV'.AO[1].AV[1] .AO[2].AV[2]) is TRUE then
Convert expr(AN.AO.AV) into string AS = "AV'.AO[1].AV[1] .AO[2].AV[2]";
return AS;
else null;
end if
else null;
end if

3.4. E_RBAC Scheme Model

The E_RBAC is created by integrating the RBAC and ECP-ABE. To protect the data the E_RBAC access policies are enforced to the data and the encryption is done by ECP-ABE which is explained in figure 2. It contains the following phases.

Setup: The master key mk and public parameter pk is generated.

Policy Specify: The access policies are specified by the owner of the data which will enforces the ERBAC rules to the data and it is converted to the access tree T*.

Encryption: The encryption transforms the access tree T* to the normal tree T and encryption is done by using T and cipher text C_{T*} is produced which contains T*.

Key Request: The user sends a SKG to extracted parts and T* is analyzed and the access tree is extracted which used to decrypt C_{T*}.

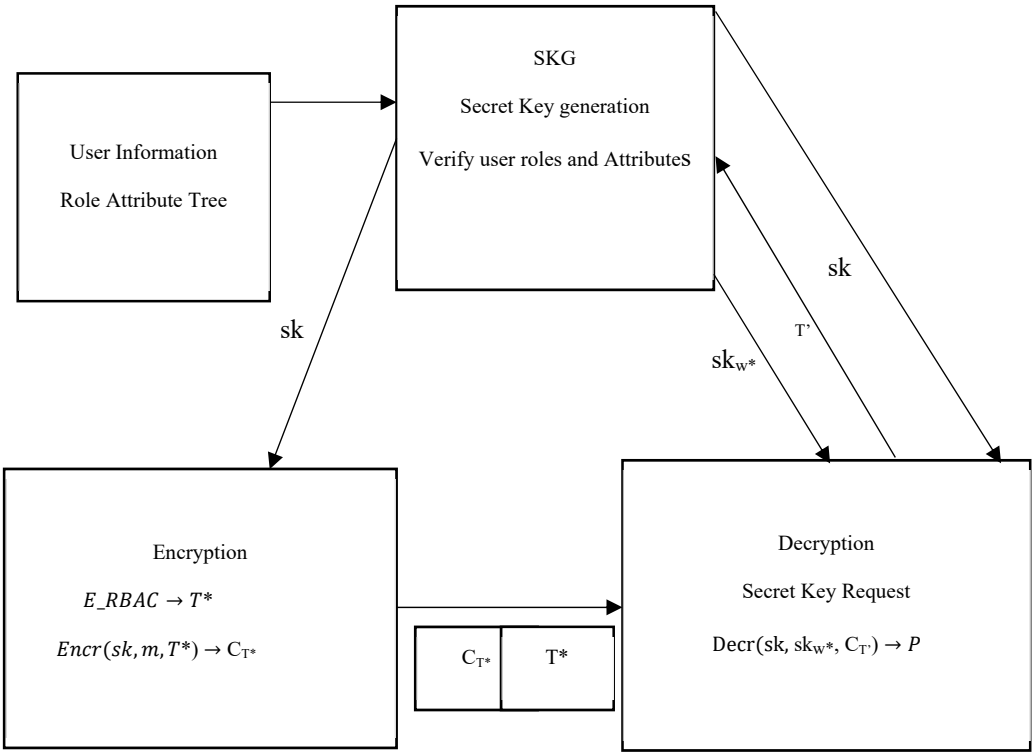


Figure 2. RBAC-CPABE model

Decryption: If the E_RBAC access policies of w* are satisfied then the plaintext p is returned.

Key Generation: The user attributes are verified by SKG by using verification algorithm and finally PKG generates secret key sk_{w*}, attribute sets w* which corresponds to w*.

In E_RBAC method the RBAC access policy is used by the user. Depending on the mapping model the E_RBAC is transform to ECP-ABE access tree. Data access contains two process which includes decryption and secret key request. The both processes are performed in the decryption module. To get the secret key the decryption sends all the leaf nodes in the access tree to the SKG and it keeps the user information and inheritance tree. By using the attribute verification method, the SKG verifies the roles and user attribute with extended attributes which generates the secret key and sends to the user. If the role of the user, environment information and instric attributes satisfies the E_RBAC policy. Then the cipher text is successfully decrypt by the user.

4. Implementation Details

The implementation framework is based on the E_RBAC model having three phases: SKG, encryption and decryption. In case of the enterprise data it has large amount of data so leads to computational burden. So, the efficiency will be reduced. To avoid this situation the User Authority (UA) is introduced and it did the work of SKG. In this model the sender signs a message and the signature is verified by the receiver for the respond to the user. To verify and sign the identity the IBE scheme is used. The frame work has three phases namely cloud server which stores the enterprise data, authorized users in the group and trusted servers which manages user attributes and secret key generation.

4.1. *Encryption*

The access policies are defined by the data owners and the data will be encrypted before publishing the data into the cloud server. The data is used by the data owner and the E_RBAC access control policy is given as input. The access control policy method is mapped with the access tree. Then by using the encryption and signature module the user data is encrypted by hybrid encryption and IBE secret key. AES encryption and E_RBAC method is used to encrypt the user data and the AES secret key respectively using access tree. The AES and E_RBAC cipher text, signature and access tree is outsourced to the public cloud.

4.2. Decryption

Access control is achieved by using decryption and it contains two steps namely data decryption and secret key application. By using E_RBAC secret key application leaf nodes in the access tree are extracted. The extracted information is sent to UA with the user identity which forms a request for the E_RBAC secret key. After the message is received from UA the signature is verified with authentication module by the receiver and extracts the private key. If the user attribute satisfies access control policy then the E_RBAC cipher text is decrypted and get the private key and the data is decrypted.

4.3. User Attribute

The UA will authenticate user attribute by creating SKG and generate secret keys. When the UA receives the data it first checks the authentication of the user by using authentication module. If the message is valid the request type is analyzed by the UA ie., E_RBAC secret key or IBE secret key request. If it is for IBE secret key the user identities are extracted by UA and sends to SKG through IBE secret key module. If it is for E_RBAC secret key the user identity is extracted through management module and E_RBAC secret key application verifies the attribute set with inheritance tree and user information which generates attribute set and send to SKG and the user secret key is generated.

4.4. Secret Key Generation

SKG generates secret key which is used for encryption and decryption. By using the management module and authentication module SKG verifies whether the data is receiver from the authorized user or not. If it is a valid user the secret key is generated by SKG using IBE secret key generation or E_RBAC secret key generation. In E_RBAC the signed secret key to the UA which verifies the SKG validity and returns to the user.

5. Performance Evaluation

ECP-ABE has proven the security against the CPA. In ECP-ABE the E_RBAC introduce the security problem. First attribute a_i in ECP-ABE will apply for private key and not included in access tree. In E_RBAC method the adversary A challenges DC-RBAC policy S instead extended tree T*. Ensuring E_RBAC security it should modify the restriction. Second, before encryption the DC-RBAC method is mapped with extended tree and we have to check whether any security risk is introduced in mapping process.

The security is analyzed by the following ways.

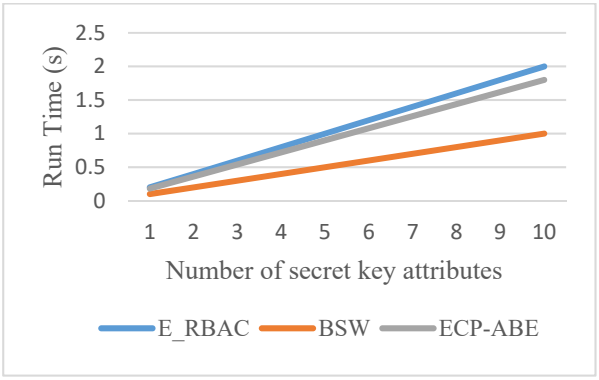
(1) In first problem the access policy S in DC-RBAC is challenges with A, A cannot able to decrypt the cipher text since the attributes w is used to get private key in the restriction a_i Є S the constraint and role assignment which includes complex operators are not satisfied. So, the attributes in the extended leaf node in T* are not satisfied. Therefore, the variation in the access control policy will not affect security.
(2) The access control policy S maps with the extended leaf T* which is visible with A. So, the attack probability of A will not increase when compare with ECP-ABE method.

The efficiencies in CP-ABE and ECP-ABE are nearly same and the encryption and decryption used in ECP-ABE and E_RBAC are same. The main difference between ECP-ABE and E_RBAC and the additional overhead. In the encryption the DC-RBAC access control policy are mapped to the ECP-ABE extended tree. In the key generation the extended attribute generation and verification are associated with the role inheritance.

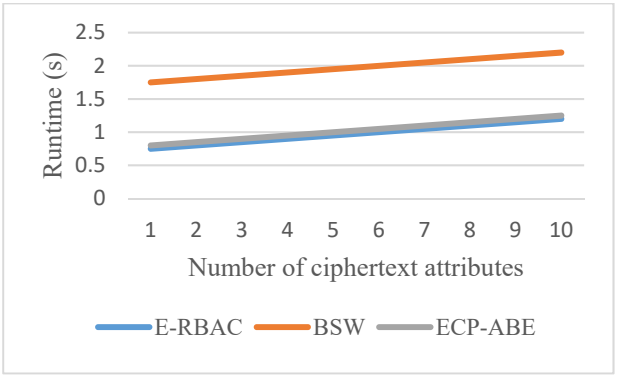
In this experiment the efficiency of classical BSW, ECP-ABE and E_RBAC are compared. Three policy group files are taken as test samples. Ten test policy files are available in each group and there are 1 to 10 attribute nodes. Group A contains the policies which has standard attributes and used in BSW scheme. Group B contains policies which has environment constraints and user attributes which is used in ECP-ABE method. Group C

contains policies which has environment constraint, user attributes and role assignment which is used in RBAC-CPABE method.

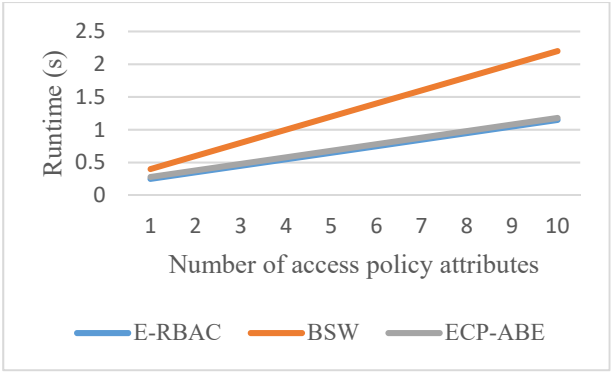
The comparison of efficiency is shown in the figure 3. There are similar results for ECP-ABE and E_RBAC in private key, encryption and decryption in time consumption and lower than BSW. In encryption the required time to map DC-RBAC access control policy and ECP-ABE access tree structure are small so it will be ignored and the verification time needed for extended attributes with the role inheritance does not need extra overhead. So the E_RBAC efficiency is good than the ECP-ABE model. The E_RBAC has enhanced access control, high flexibility and better hierarchy capability.



a. Encryption



b. Secret key



c. Decryption

Figure 3. Comparing efficiencies of RBAC-CPABE and BSW models.

E_RBAC model was created based on ECP-ABE method, ITHJ09 [25]. ECP-ABE use any ABE scheme having tree structure. Similarly, E_RBAC model are also based on any ABE scheme that have tree structure and we can extend the tree. So, the security is improved with the existing method and more over the efficiency property is improved and there is an improvement in the access control methodology.

6. Conclusion

The problem while storing the enterprise data is addressed by E_RBAC model. Based on RBAC model we propose extended access control method so that the owner can specify their own RBAC policies to their data object. In the case of the role constraints E_RBAC contains environmental and attribute constraint which contains the information about the environment and the user information. So, E_RBAC achieves fine grained and flexible access control. To construct self-data protection E_RBAC is merged with ECP-ABE which forms a policy mapping method. The experimental analysis and the security analysis define E_RBAC is not having the security risk and computational overhead when compared with CP-ABE model and the access control is also improved. Hence to achieve an efficient outsourced enterprise data protection in the public cloud is done by E_RBAC model.

References

[1] C. S. Alliance. (2011). Security Guidance for Critical Areas Focus in Cloud Computing V3.0. [Online]. Available:https://downloads.cloudsecurityalliance.org/initiatives/guidance/csaguide.v3.0.pdf
[2] D. Boneh and M. Franklin, ``Identity-based encryption from the Wpairing," in Proc. CRYPTO, Santa Barbara, CA, USA, Aug. 2001, pp. 213229.
[3] Y. Zhu, G.-J. Ahn, H. Hu, and H. Wang, ``Cryptographic role-based security mechanisms based on role-key hierarchy," in Proc. 5th ACM Symp.Inf., Comput. Commun. Secur., Beijing, China, Apr. 2010, pp. 314319.
[4] Y. Zhu, H.-X. Hu, G.-J. Ahn, H.-X. Wang, and S.-B. Wang, ``Provably secure role-based encryption with revocation mechanism," J. Comput. Sci. Technol., vol. 26, no. 4, pp. 697-710, Jul. 2011.
[5] Y. Zhu, G.-J. Ahn, H. Hu, D. Ma, and S. Wang, ``Role-based cryptosystem:A new cryptographic RBAC system based on role-key hierarchy," IEEE Trans. Inf. Forensics Security, vol. 8, no. 12, pp. 2138-2153, Dec. 2013.
[6] A. Sahai and B. Waters, ``Fuzzy identity-based encryption," in Advances Cryptology-EUROCRYPT (Lecture Notes in Computer Science), Berlin, Germany: Springer, May 2005, pp. 457-473.
[7] V. Goyal, O. Pandey, A. Sahai, and B. Waters, ``Attribute-based encryption for fine-grained access control of encrypted data," in Proc. 13th ACM Conf Comput. Commun. Secur., Oct. 2006, pp. 89-98.
[8] J. Bethencourt, A. Sahai, and B. Waters, ``Ciphertext-policy attribute based encryption," in Proc. IEEE Symp. Secur. Privacy (SP), May 2007, pp. 321-334.
[9] Y. Zhu, D. Huang, C. J. Hu, and X. Wang, ``From RBAC to ABA Constructing flexible data access control for cloud storage services," IEEE Trans. Services Computing, vol. 8, no. 4, pp. 601-616, Jul. 2015.
[10] L. Cheung and C. Newport, ``Provably secure ciphertext policy ABE," in Proc. 14th ACM Conf. Comput. Commun. Secur., Oct. 2007, pp. 456-465.
[11] T. Nishide, K. Yoneyama, and K. Ohta, ``Attribute-based encryption with partially hidden encryptor-specied access structures," in Applied Cryptography and Network Security. New York, NY, USA: Springer, Jun. 2008, pp. 111-129.
[12] K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi, ``A ciphertext policy attribute-based encryption scheme with constant ciphertext length," in Information Security Practice and Experience. Xi'an, China: Springer, Apr. 2009, pp. 13-23.
[13] F. Guo, Y. Mu, W. Susilo, D. S. Wong, and V. Varadharajan, ``CP-ABE with constant-size keys for lightweight devices," IEEE Trans. Inf. Forensics Security, vol. 9, no. 5, pp. 763-771, May 2014.
[14] P. Junod and A. Karlov, ``An efcient public-key attribute-based broadcast encryption scheme allowing arbitrary access policies," in Proc. 10th Annu. ACM Workshop Digit. Rights Manage. Chicago, IL, USA, Oct. 2010, pp. 13-24.
[15] R. Ostrovsky, A. Sahai, and B. Waters, ``Attribute-based encryption with non-monotonic access structures," in Proc. 14th ACM Conf. Comput. Commun. Secur.. Alexandria, VA, USA, Oct./Nov. 2007, pp. 195-203.
[16] Y. Zhu, H. Hu, G.-J. Ahn, M. Yu, and H. Zhao, ``Comparison-based encryption for fine-grained access control in clouds," in Proc. 2nd ACM Conf. Data Appl. Secur. Privacy. San Antonio, TX, USA, Feb. 2012, pp. 105-116.
[17] B. Lang, R. Xu, and Y. Duan, ``Extending the ciphertext-policy attribute based encryption scheme for supporting flexible access control," in Proc. 10th Int. Conf. Secur. Cryptogr.. Reykjavik, Iceland, Jul. 2013, pp. 1-11.
[18] B. Lang, R. Xu, and Y. Duan, ``Self-contained data protection scheme based on CP-ABE," in E-Business Telecommunications, vol. 456. Berlin, Germany: Springer, 2014, pp. 306-321.
[19] B. Waters, ``Functional encryption for regular languages," in Advances in Cryptology-CRYPTO. Santa Barbara, CA, USA: Springer, Aug. 2012, pp. 218-235.
[20] D. Ferraiolo and R. Kuhn, ``Role-based access control," in Proc. 15th Nat. Comput. Secur. Conf., Oct. 1992, pp. 554-563.
[21] J. Crampton, ``Cryptographic enforcement of role-based access control," in Formal Aspects of Security and Trust. Pisa, Italy: Springer, Sep. 2011, pp. 191-205.
[22] L. Zhou, V. Varadharajan, and M. Hitchens, ``Enforcing role-based access control for secure data storage in the cloud," Comput. J., vol. 54, no. 10, pp. 1675-1687, 2011.
[23] C. Hong, Z. Lv, M. Zhang, and D. Feng, ``A secure and efficient role based access policy towards cryptographic cloud storage," in Proc. 12th Int. Conf. WebAge Inf. Manage. (Lecture Notes in Computer Science), Wuhan, China, Sep. 2011, pp. 264-276.
[24] J.L. Joneston Dhas, S. Maria Celestin Vigila and C. Ezhil Star, "A Framework on Security and Privacy-Preserving for Storage of Health Information Using Big Data," International Journal of Control Theory and Application. Vol. 10, No.03, pp. 91-100, 2017.
[25] L. Ibraimi, Q. Tang, P. Hartel, and W. Jonker, ``Efcient and provable secure ciphertext-policy attribute-based encryption schemes," in Information Security Practice and Experience. Xi'an, China: Springer, Apr. 2009, pp. 1-12.z