EAC: Encryption Access Control Scheme for Policy Revocation in Cloud Data

Phyo Wah Wah Myint, Swe Zin Hlaing, Ei Chaw Htoon
University of Computer Studies (Taungoo)
Taungoo, Myanmar
phyowahwahmyint@ucstaungoo.edu.mm, swezin@uit.edu.mm, eichawhtoon@uit.edu.mm

Abstract

Since a lot of information is outsourcing into cloud servers, data confidentiality becomes a higher risk to service providers. To assure data security, Ciphertext Policy Attributes-Based Encryption (CP-ABE) is observed for the cloud environment. Because ciphertexts and secret keys are relying on attributes, the revocation issue becomes a challenge for CP-ABE. This paper proposes an encryption access control (EAC) scheme to fulfill policy revocation which covers both attribute and user revocation. When one of the attributes in an access policy is changed by the data owner, the authorized users should be updated immediately because the revoked users who have gained previous access policy can observe the ciphertext. Especially for data owners, four types of updating policy levels are predefined. By classifying those levels, each secret token key is distinctly generated for each level. Consequently, a new secret key is produced by hashing the secret token key. This paper analyzes the execution times of key generation, encryption, and decryption times between non-revocation and policy revocation cases. Performance analysis for policy revocation is also presented in this paper.

Key Words- Ciphertext Policy Attributes-Based Encryption (CP-ABE), Access Policy, Policy Revocation, Updating Policy Level

1. Introduction

Currently, the use of cloud-based storage services is an enormous growth because they offer rentable frameworks based on pay per use pattern. During the use of cloud storage data, the lack of data control for the data owner can occur among the communications of the parties. Hence, data security for cloud storage is solved by studying the cryptographic techniques. Since the encryption technique is critical for the cloud data sharing system among the organizations, some of the existing encryption techniques can be applied for retrieving cloud data against unauthorized accesses. However, the traditional cryptographic techniques are not enough to be perfect security for cloud data because the data should be controlled by the data owner rather than the Cloud Service Providers (CSPs) [3]. Moreover, the access policies, which are controlled by the data owner, can be frequently updated according to several situations among the parties [10]. Mostly, access policies are specified as the values of the attributes of cloud users. Therefore, the researchers are interested in CP-ABE for retrieving the cloud storage data. Since the CP-ABE will grant the data owner the right to define access policy control, it is adopted by the cloud users as access control based on attributes. Despite the CP-ABE is popular in access control for the cloud environment, the attributes management is still at a critical point [1]. More policy changes can be more complex in key management [7]. Especially, the key management is essential for access policy changes when the data owner manages attributes for his authorized users. In this paper, a policy revocation is proposed for enhancing traditional CP-ABE because the traditional CP-ABE did not consider any revocation. In the rest of the paper, section 2 deals with the related works of CP-ABE. The research background is explained in section 3. The proposed EAC scheme is presented in section 4. The performance analysis and experimental results are shown in section 5. The conclusion and future work are included in section 6.

2. Related Work

Researchers are incidentally aiming to enhance traditional CP-ABE as the demand for revocation. Initially, Bethencourt et al. [2] proposed a traditional CP-ABE scheme with the use of an essential attribute structure for delegation. They had restrictions for verifying security using generic group heuristic. Any attributes revocation was not considered in their scheme. S. Jahid and N. Borisov [4] introduced a Proxy-based Immediate Revocation of ATTribute-based Encryption namely the PIRATTE scheme for efficient revocation in CP-ABE. They built an Online Social Networks (OSN) architecture that uses the PIRATTE to accomplish access control by encryption. They enforced their PIRATTE scheme to compare with the traditional CP-ABE scheme. According to their scheme, the Proxy Re-Encryption (PRE) should be considered for the scalability in terms of OSN computing and interaction. K. Yang et al. [11] introduced an effective way of attributes revocation to deal with dynamic changes in user access rights in large-scale systems. They designed a modern underlying CP-ABE scheme enabling revocation of the attribute, in which each attribute has been assigned a version number. When a user revokes an attribute, the authority produces a new version key for such a revoked attribute and an update key for doing that. The components related to the revoked attribute in the ciphertext could also be upgraded to the current version using the update key. They delegated the task of the ciphertext update to the server using the PRE process. However, the attributes' versions may always be several that can cause the overload records for the cloud server. C. Wang et al. [9] suggested a CP-ABE based cryptographic cloud storage data revocation scheme. The original data in their scheme is first divided into several slices, then submitted to the cloud storage. When a revocation happens, the data owner only must retrieve re-encrypt, and re-upload one slice instead of all the data. However, their scheme may be a more time-consuming operation because the system must divide the slice of the original data before uploading it to the server. J. Li et al. [5] addressed a flexible fine-grained cloud computing ABE scheme for revocation. A concept for group management of users was presented. The group, which has the same structure of attributes, was classified in the same group and the users were also associated with each group of attributes. For each group, a secret group key is created. The group manager updates the group key except for revoking users whenever a user is removed from the group. They also improved traditional CP-ABE by a group manager to control the group key for each user. However, their scheme takes a little time over the other schemes because they need to set a group manager. J. K. Liu et al. [6] nominated an effective revocable CP-ABE scheme by integrating the list of revocations in the ciphertext. Because the revocation list grew longer as time goes on, they also compromised this factor by introducing a secret key time validation procedure so that users will have their keys expired on a date. These keys could be removed after their expiry date from the revocation list to keep the revocation list short. They simulated their fundamental system as a cloud-free system. In consideration of their scheme, the expiration time should be extended to test for the cloud storage system.

According to the survey on the related works, this paper nominated policy revocation for enhancing the traditional CP-ABE. In this paper, the proposed EAC scheme is applied to support policy revocation by evaluating performance analysis and execution times for key generation, encryption, and decryption between non-revocation and policy revocation cases. A session key is also considered for each authorized user.

3. Research Background

The CP-ABE is a modified form of Attributes-Based Encryption (ABE). ABE is a public key based one-to-many encryption that enables users to encrypt and decrypt data based on the attributes of a user [2]. If and only if the set of attributes of the user key meets the attributes of the ciphertext, the decryption is possible for end-users. Before introducing the concept of CP-ABE, a basic idea for an access policy is initially described.

3.1. Access Policy

In CP-ABE, both encryption and decryption phases are dependent upon the user's secret key which is integrated with an access policy. An access policy consists of essential attributes of data users for determining authorized users. It is a very important concept to deny unauthorized access for providing fine-grained access control. It includes a threshold value and consists of its children [8]. Each non-leaf node stands for a threshold gate. If num_x is the number of children for a node x and k_x is its threshold value, then $0 < k_x \leq num_x$. For each access policy, leaf node x is identified with an attribute and a threshold value $k_x= 1$. Where x is OR compatible, then $k_x = 1$; if the node x relates to the gate AND, then $k_x = num_k$.

3.2. Traditional CP-ABE Scheme

Traditional CP-ABE is a basic idea of current enhanced CP-ABE schemes. It includes four basic functions such as Setup, KeyGen, Encrypt, and Decrypt [2]. The functions of traditional CP-ABE are depicted as follows:

- Setup (λ, U) $\longrightarrow$ {MK, PK}: The setup function takes the security parameter λ and the universe attributes U as an input. It takes outputs as master secret key MK and public parameter PK.
- KeyGen (A_{user}, MK, PK) $\longrightarrow$ {SK}: It is run by an authority. In the key generation function, the access policy of user A_{user}, MK, and PK are taken as input. A secret key SK is produced as an output.
- Encrypt (PK, M, A_{Owner}) $\longrightarrow$ {C}: It is run by the data owner. The encryption function inputs PK, plaintext M and access policy of data owner A_{Owner}. It outputs the ciphertext C associated with A_{Owner}.
- Decrypt (C, SK, PK) $\longrightarrow$ {M}: It is run by the data user. In the decryption function, it inputs C, SK, and PK. If A_{user} satisfies A_{Owner} taking part in C, it produces output M.

To explain the above functions, let there be two data users namely John and Molly. Assume that A_{Owner} is defined by a threshold or the essential attributes as {Role: 'Nurse', Field: 'Public Health', Hospital: 'Shwe La Min'}. Let A_{John} be an access policy of John which consists of {Role: 'Lab Member', Field: 'Allergy and Immunology', Hospital: 'Shwe La Min'}. Let A_{Molly} be a threshold of Molly's attributes which represents the access structure for {Role: 'Nurse', Field: 'Public Health', Hospital: 'Shwe La Min'}. Because A_{Molly} satisfies A_{Owner}, a secret key for Molly is generated by the authority for decrypting ciphertext associated with A_{Owner}. Molly can decrypt the ciphertext by using her secret key. Since A_{John} does not satisfy A_{Owner}, John is an unauthorized user.

3.3. Revocation Issue for CP-ABE Scheme

Most of the researchers are trying to improve the traditional CP-ABE by enhancing the revocation issue.

3.3.1. CP-ABE with Attributes Revocation. In the case of attribute revocation, for instance, Bob's original attributes are {Physics; Student; University A}. Then, he has modified to {Chemistry; Student; University A}. The original Physics attribute (but not Student or University A) should therefore be revoked. In CP-ABE, revocation of the attributes is a significant task because the characteristics of an entity also change over time, e.g., work status, wage, etc. Once any attribute of a user changes, it will get a new secret key with a new attribute. Otherwise, the old secret key still fits with an access policy, and the decryption process is worked out. Therefore, the problem of revocation in attributes level is a demand for researchers to enhance traditional CP-ABE.

3.3.2. CP-ABE with User Revocation. In the case of user revocation, for instance, the user has quit the company or organization. Therefore, it must revoke all attributes of that user. As compared with the previous case of attribute revocation, Bob has quit University A and should therefore be revoked. Since CP-ABE is one of the cryptographic mechanisms to prevent unauthorized users, researchers are also addressing the problem of user revocation in CP-ABE.

4. Proposed EAC Scheme

The proposed EAC scheme addressed to deal with the policy revocation which covers both attributes and user revocation in CP-ABE.

4.1. Overall System Architecture

There are four entities in the system structure for the proposed EAC scheme. They are cloud server, data owner, data user, and trusted authority.

Figure 1 shows an overall system architecture for the proposed EAC scheme by comparing it with the non-revocation case. In the figure, the data owner encrypts his plaintexts by defining his access policies for each plaintext. Then, he stores his ciphertexts to the cloud server. The trusted authority (TA) generates the corresponding SK and session key (𝔖) for the data user according to each request. The authorized user can use his 𝔖 and SK to decrypt the ciphertext.

When the data owner updates his access policy in the ciphertext, the policy revocation case takes place in the system. In this case, four types of updating policy level ($\text{Upd}_{\text{level}}$) are initially classified for the data owner.

To explain each $\text{Upd}_{\text{level}}$, let the number of essential attributes in access policy be n. These four types of updating policy levels are as followings:

- 'All-Attributes-Changes' level: 'All-Attributes-Changes' means that changing all attributes to update existing access policy. (No. of changes = n)
- 'Below-the-Half-Attributes-Changes' level: This level means that changing the number of attributes which is below half of the total attributes in the access policy. (No. of changes < n/2)
- 'Over-the-Half-Attributes-Changes' level: This level means that changing the number of attributes which is over half of the total attributes in the access policy. (No. of changes > n/2)
- 'By-Name-Changes' level: This level means that important username change is formed forcefully in the access policy. (No. of changes =1)

According to the above four levels, each updated secret key ($\text{Upd}_{\text{SK)}}$ can be separated relying on the $\text{Upd}_{\text{level}}$ that the data owner updates his access policy. Principally, the TA determines to calculate either SK or Upd_{SK} to encrypt the plaintext according to the behavior of the data owner. TA generates SK for the non-revocation case and Upd_{SK} for the policy revocation case. For the policy revocation case, the authorized user can decrypt the ciphertext by similarly using his 𝔖 and Upd_{SK} as in the non-revocation case.

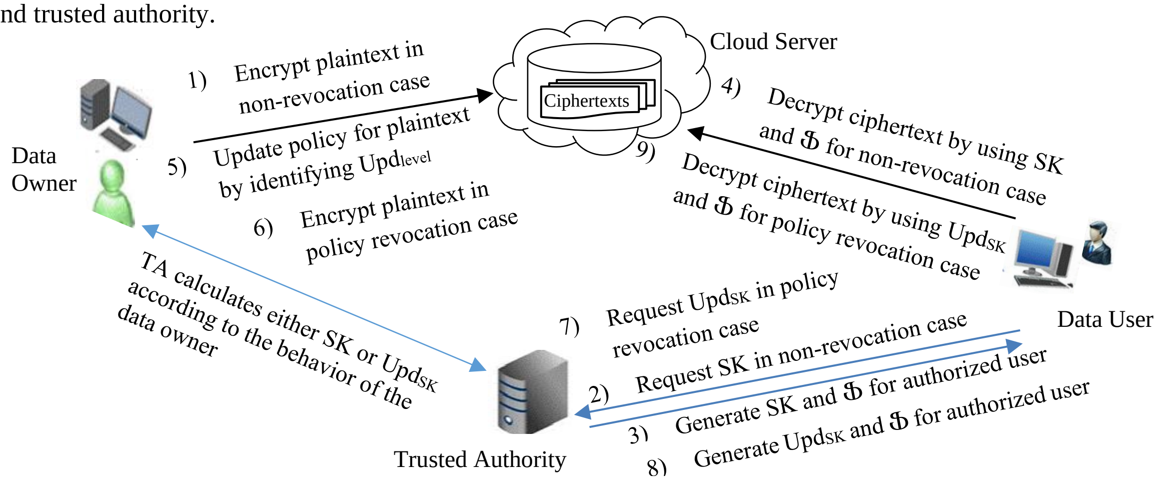
Cloud Server
Ciphertexts
Data Owner
Data User
Trusted Authority
1) Encrypt plaintext in non-revocation case
5) Update policy for plaintext by identifying Upd_level
6) Encrypt plaintext in policy revocation case
4) Decrypt ciphertext by using SK and 𝔖 for non-revocation case
9) Decrypt ciphertext by using Upd_SK and 𝔖 for policy revocation case
TA calculates either SK or Upd_SK according to the behavior of the data owner
7) Request Upd_SK in policy revocation case
2) Request SK in non-revocation case
3) Generate SK and 𝔖 for authorized user
8) Generate Upd_SK and 𝔖 for authorized user

Figure 1. Overall System Architecture for Proposed EAC Scheme

4.2. Proposed Algorithms for Policy Revocation

There are four main algorithms to deal with policy revocation in the proposed EAC scheme. They are updating policy algorithm, update key generation algorithm, re-encryption algorithm, and decryption algorithm, respectively.

Each access policy is defined by a unique identity that represents essential attributes for determining authorized users. In the updating policy algorithm, the current policy identity of the data owner (DO) is noted as $\text{Policy}_{\text{id}}$ and the new policy identity that the DO wants to change is noted as $\text{New\_Policy}_{\text{id}}$. For the individual updating policy levels, the four status numbers are identified. Let the status number be α. According to each $\text{Upd}_{\text{level}}$, α is defined and the identity of content for plaintext is captured in the system.

Updating Policy algorithm

```
Input: $\text{Policy}_{\text{id}}$, $\text{New\_Policy}_{\text{id}}$
Output: $\text{Upd}_{\text{level}}$
1.  DO chooses one of four policy updating levels
2.  Identify $\text{Upd}_{\text{level}}$ according to a choice in step1
3.  Get α by determining each $\text{Upd}_{\text{level}}$
4.  Set $\text{New\_Policy}_{\text{id}}$ as a current policy identity and revoke
    old $\text{Policy}_{\text{id}}$
5.  Return $\text{Upd}_{\text{level}}$
```

The re-encryption algorithm is used to re-encrypt the plaintext message (M) associated with $\text{New\_Policy}_{\text{id}}$. The DO uses the Upd_{SK} for an encryption key. It outputs the ciphertext C.

Re-Encryption Algorithm

```
Input: M, $\text{New\_Policy}_{\text{id}}$, $\text{Upd}_{\text{SK}}$
Output: C
1.  DO updates $\text{Policy}_{\text{id}}$ to $\text{New\_Policy}_{\text{id}}$ for M
2.  Get $\text{Upd}_{\text{level}}$ = UpdatePolicy($\text{Policy}_{\text{id}}$, $\text{New\_Policy}_{\text{id}}$)
3.  Get $\text{Upd}_{\text{SK}}$ from the TA.
4.  C = ReEncrypt (M, $\text{New\_Policy}_{\text{id}}$, $\text{Upd}_{\text{SK}}$)
5.  Return C
```

In the update key generation algorithm, TA creates a default key string (ʁ) and takes the unique identity of M which is to be accessed as β. TA initially identifies the status number (α) according to the $\text{Upd}_{\text{level}}$. Next, the TA converts numbers to strings for both β and α. Then, the TA concatenates the key ʁ with the strings of β and α. This concatenation is converted to the Base64String format as a secret token key (SK_{token}). Finally, the TA generates Upd_{SK} by getting a hash code of the MD5 function for SK_{token}. The final Upd_{SK} is also uniformly converted to the Base64String format output. The TA also generates a 𝔅 for each user according to the user's request identity (ɤ).

Update Key Generation Algorithm

```
Input: ʁ, β, ɤ, $\text{Upd}_{\text{level}}$, $\text{New\_Policy}_{\text{id}}$
Output: $\text{Upd}_{\text{SK}}$, 𝔅
1.  Initialize α according to $\text{Upd}_{\text{level}}$
2.  TA takes the ɤ and β from the users' requests list
3.  TA gets ʁ, and converts both β and α to strings
4.  $\text{SK}_{\text{token}}$ = Concatenate(ʁ, β, α)
5.  $\text{SK}_{\text{token}}$ = ToBase64String($\text{SK}_{\text{token}}$)
6.  $\text{Upd}_{\text{SK}}$ = GetHashCode($\text{SK}_{\text{token}}$)
7.  $\text{Upd}_{\text{SK}}$ = ToBase64String($\text{Upd}_{\text{SK}}$)
8.  TA generates 𝔅
9.  Records the $\text{New\_Policy}_{\text{id}}$ to filter unauthorized users
10. Return $\text{Upd}_{\text{SK}}$, 𝔅
```

In the decryption algorithm, the data user (DU) proves his policy identity ($\text{User\_Policy}_{\text{id}}$) to decrypt C. If his $\text{User\_Policy}_{\text{id}}$ satisfies the $\text{New\_Policy}_{\text{id}}$ of the DO, the TA will give the Upd_{SK} and 𝔅 to the DU. The DU decrypts the proxy ciphertext (C') from the cloud by using his 𝔅 to get the original ciphertext C. Finally, the DU can decrypt the C by using his Upd_{SK}.

Decryption Algorithm

```
Input: C, 𝔅, $\text{Upd}_{\text{SK}}$, $\text{User\_Policy}_{\text{id}}$
Output: M
1.  DU enters the system by proving his attributes.
2.  The system records his $\text{User\_Policy}_{\text{id}}$ according to his
    attributes.
3.  DU requests the $\text{Upd}_{\text{SK}}$ and 𝔅 to the TA.
4.  The system records his $\text{User\_Policy}_{\text{id}}$ according to his
    attributes.
5.  DU requests the $\text{Upd}_{\text{SK}}$ and 𝔅 to the TA.
6.  DU checks his mail inbox and gets the $\text{Upd}_{\text{SK}}$, 𝔅 from TA.
7.  DU takes the C' from the cloud.
8.  C = Decrypt (C', 𝔅, $\text{User\_Policy}_{\text{id}}$)
9.  M = Decrypt (C, $\text{Upd}_{\text{SK}}$, $\text{User\_Policy}_{\text{id}}$)
10. Return M
```

5. Performance Analysis and Experimental Results

All the experiments are performed on the Windows10 system including Intel Core i5 with 1.6GHz and 1.8GHz processors, 1TB of the hard disk drive, and 8GB of RAM. A software version of the Microsoft Visual Studio 2019 is used for implementation. In this experiment, the personal health records (PHRs) dataset is tested in the Microsoft Azure cloud server.

5.1. Performance Analysis

Policy management and the variation on threshold and $\text{Upd}_{\text{level}}$ are discussed about performance analysis for policy revocation of the proposed EAC scheme.

5.1.1. Policy Management. Since access policy is essential for determining authorized users to permit data retrieving from the cloud server, policy management is an important impact on the CP-ABE scheme. Therefore, the scalability for policy management is considered for growing up the numbers of access policies. Since three essential attributes in an access policy are tested in the proposed EAC scheme, let m be the total numbers of access policies. Assume that i, j, and k are the total numbers of essential attribute$_1$, essential attribute$_2$, and essential attribute$_3$ respectively. The total number of policies in the proposed EAC scheme is (i * j * k) or m. For example, the total number of policies will become 1210 policies or rules in the case of 11 essential attribute$_1$, 11 essential attribute$_2$, and 10 essential attribute$_3$. Similarly, the more increased numbers of usernames that are permitted for authorization can be growing the total number of policies for usernames forcefully.

5.1.2. Threshold and Upd$_{level}$ Variation. In the proposed EAC scheme, specifying the numbers of Upd$_{level}$ is corresponding to the variation of the numbers of threshold or essential attributes in an access policy. This means that let the number of thresholds in an access policy be n. If n is an even number, a new updating policy level will become namely 'Half-Attributes-Changes' as n/2 changes or 50% revocation in a threshold. Currently, the number of thresholds is three, therefore, n is 3 as an odd number. Therefore, the 'All-Attributes-Changes' level represents 'n' changes, 'Below-the-Half-Attributes-Changes' level represents 'changes < n/2' and the 'Over-the-Half-Attributes-Changes' level represents 'changes > n/2'. The fourth level known as the 'By-Name-Changes' level is evaluated on one essential attribute or threshold of username forcefully. According to the number of thresholds in an access policy, the Upd$_{level}$ can be growing one more level as future 'Half-Attributes-Changes' in an even number threshold.

5.2. Experimental Results

The execution times for key generation, encryption, and decryption are resulted by an average measure after ten-round testing for experiments. Results are measured for ten attributes from a universe of PHRs dataset. Three essential attributes are defined for access policy in each PHR data.

Figure 2 shows the comparisons for key generation, encryption and decryption times between non-revocation case and policy revocation case of the proposed EAC scheme. In the figure, the key generation time for the non-revocation case takes 558.9 milliseconds, and the policy revocation case takes 334.8 milliseconds, respectively. The policy revocation case takes faster key generation time than non-revocation case because of classifying revocation levels and using the Base64String format key. For encryption time, the non-revocation case takes 469.9 milliseconds, and the policy revocation case takes 421.3 milliseconds long as shown in the figure. For decryption time, the non-revocation case takes 587.3 milliseconds and the policy revocation case of the proposed EAC scheme takes 586 milliseconds, respectively. According to the feature of CP-ABE, both encryption and decryption times are not serious among different CP-ABE schemes.

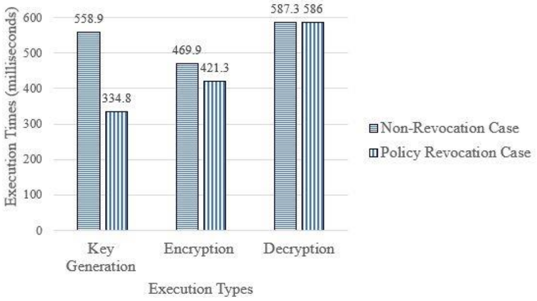

Figure 2. Key Generation, Encryption and Decryption Times between Non-Revocation case and Policy Revocation case

Figure 3 shows the measurements of the variation on encryption times and the number of attributes between non-revocation case and policy revocation case of the proposed EAC scheme.

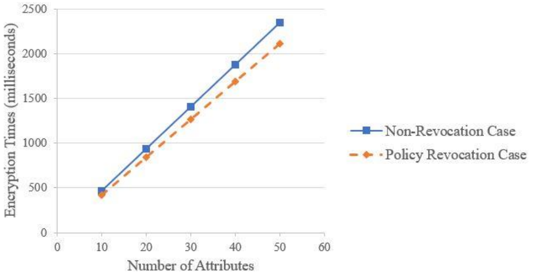

Figure 3. Encryption Times and Attributes Variation between Non-Revocation case and Policy Revocation case

The comparisons of the variation on decryption times and the number of attributes between non-revocation case and policy revocation case are also shown in Figure 4.

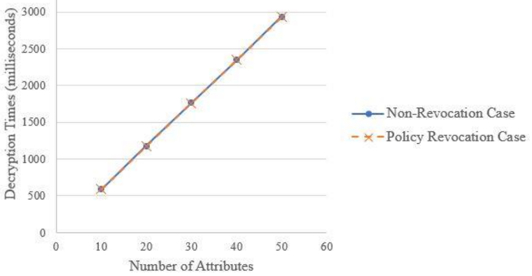

Figure 4. Decryption Times and Attributes Variation between Non-Revocation case and Policy Revocation case

Both encryption and decryption times are tested for 10 attributes from the whole attributes-universe. The more numbers of attributes take longer for both encryption and decryption times as shown in figure 3 and figure 4.

Table 1. Analytical Comparisons between Proposed EAC Scheme and Other Enhanced Scheme

Scheme	Updated Secret Key ($\mathrm{Upd_{SK}}$)	Number of $\mathrm{Upd_{SK}}$	Attributes Revocation	Policy Revocation	Threshold Variation	Group Manager set up	Session Expiration
Li2017's CP-ABE [5]	Yes	1	Yes	No	No	Yes	No
Proposed EAC Scheme	Yes	More than 1	Yes	Yes	Yes	No	Yes

Table 1 shows analytical comparisons between the proposed EAC scheme and other enhanced CP-ABE scheme namely Li2017's scheme [5]. Li2017's scheme [5] introduced the revoked user list (RUL) which is used for group user management by a group manager. As shown in Table 1, since the proposed EAC scheme does not need the group manager to set up the RUL, it takes faster key generation time as compared with Li2017's scheme [5]. By identifying unique policy identity for each access policy, revocation is considered for policy level which covers both attribute and user revocation. Because of classifying four updating policy levels in revocation, the number of $\mathrm{Upd_{SK}}$ can be more than one for each plaintext message. In other words, Li2017's scheme [5] has one $\mathrm{Upd_{SK}}$ for encrypting each plaintext message once a time. Moreover, session key is also generated for each user according to each request as in a one-time password. Therefore, the decryption key is never duplicable by integrating the secret key with a unique session key for each authorized user. Especially, classifying $\mathrm{Upd_{level}}$ is directly related to the variation of threshold numbers in the access policy. Therefore, the more threshold in access policy can grow up to one more $\mathrm{Upd_{level}}$ for an even number threshold. Although Li2017's scheme [5] could not include threshold and revocation level ($\mathrm{Upd_{level}}$) variation for an access policy, the proposed EAC scheme can provide scalability of policy management and threshold variation for access policy.

6. Conclusion and Future Work

The proposed EAC scheme enhances the traditional CP-ABE scheme by considering policy revocation. According to the experiments in this paper, key generation times in policy revocation case can be overall time safe because it uses the Base64String format key. Since the decryption phase is performed by double passing for the proxy ciphertext and original ciphertext, every authorized user uses a one-time session key to decrypt the proxy ciphertext and a private secret key for decrypting the original ciphertext. Consequently, the plaintext message can only be accessed by integrating the session key and the secret key of an authorized user. Moreover, the proposed EAC scheme can also be scalable in policy management and it considers threshold variation for classifying revocation levels. As future work, rich access policies are needed. If the more access policies are, the more complex secret keys appear. The concept of the proposed EAC scheme can be extended in the existing applications such as crypt cloud systems or e-cloud data sharing systems in web applications.

References

[1] R. R. Al-Dahhan, Q. Shi, G. M. Lee, and K. Kifayat, "Survey on Revocation in Ciphertext-Policy Attribute-Based Encryption", Sensors (Basel), Vol. 19, Issue. 7, April 9, 2019.

[2] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext- Policy Attribute-Based Encryption", In the Proceedings of IEEE Symposium on Security and Privacy (SP'07), IEEE Computer Society Washington, DC, USA, May 20-23, 2007, pp. 321-334.

[3] M. George, C. S. Gnanadhas, and K. Saranya, "A Survey on Attribute-Based Encryption Scheme in Cloud Computing", International Journal of Advanced Research in Computer and Communication, Vol. 2, Issue 11, November 2013.

[4] S. Jahid and N. Borisov, "PIRATTE: Proxy-based Immediate Revocation of Attribute-based Encryption", Computer Science, Cryptography and Security, August 23, 2012.

[5] J. Li, W. Yao, Y. Zhang, H. Qian, and J. Han, "Flexible and Fine-Grained Attribute-Based Data Storage in Cloud Computing", In Proceedings of IEEE Transactions on Services Computing, Vol. 10, Issue 5, September – October 1, 2017, pp. 785-796.

[6] J. K. Liu, T. H. Yuen, P. Zhang, and K. Liang, "Time-Based Direct Revocable Ciphertext-Policy Attribute-Based Encryption with Short Revocation List", In Proceedings of the 16th International Conference on Applied Cryptography and Network Security (ACNS), Leuven, Belgium, July 2-4, 2018.

[7] P. W. W. Myint, S. Z. Hlaing, and E. C. Htoon, "A Policy Revocation Scheme for Attributes-based Encryption", In Proceedings of the 10th International Conference on Advances in Information Technology (IAIT), ACM, New York, NY, USA, Bangkok, Thailand, December, 2018, pp.16-23.

[8] P. W. W. Myint, S. Z. Hlaing, and E. C. Htoon, "Policy-based Revolutionary Ciphertext-policy Attributes-based Encryption", In Proceedings of the 3rd International Conference on Advanced Information Technologies (ICAIT), Yangon, Myanmar, November, 2019, pp. 227-232.

[9] C. Wang, J. Wu, Y. Yuan, and J. Liu, "Insecurity of Cheng et al.'s Efficient Revocation in Ciphertext-Policy Attribute-based Encryption based Cryptographic Cloud Storage", IEEE International Symposium on Parallel and Distributed Processing with Applications and IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC), Guangzhou, China, December 12-15, 2017, pp. 1387-1393.

[10] X. Xu, J. Zhou, X. Wang, and Y. Zhang, "Multi-authority proxy re-encryption based on CPABE for cloud storage systems", Journal of Systems Engineering and Electronics (JSEE), Vol. 27, Issue. 1, February, 2016, pp 211-223.

[11] K. Yang, X. Jia, and K. Ren, "Attribute-based fine-grained access control with efficient revocation in cloud storage systems", In Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security (ASIACCS), ACM New York, USA, Hangzhou, China, May 8-10, 2013, pp. 523-528.