International Journal of Engineering & Technology, 7 (3.27) (2018) 27-31

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Access Policy's Over Encrypted Cloud Storage for Secure Deduplication

G. Kiran Kumar[1], E. Amarnath Reddy[2], B. Mamatha[3*], Kompally Manisha[4]

[1]*Department of Computer Science and Engineering, MLR Institute of Technology, Dundigal, Hyderabad.*
[2]*Department of Computer Science and Engineering, MLR Institute of Technology, Dundigal, Hyderabad.*
[3]*Department of Computer Science and Engineering, MLR Institute of Technology, Dundigal, Hyderabad.*
[4]*Department of Computer Science and Engineering, MLR Institute of Technology, Dundigal, Hyderabad.*
**Corresponding author E-mail:badampudi.mamatha27@gmail.com*

Abstract

Attribute-Based Encryption (ABE) is a basic concept that considers public-key cryptography. Ciphertext-Policy ABE (CP-ABE) is one of the approaches used by ABE for data sharing in the cloud. In CP-ABE scheme, each user's private key has a set of attributes and then the user decrypts a ciphertext if it holds a matching key. Our proposed system provides an extension to CP-ABE by implementing AES. AES uses a symmetric encryption key algorithm for a same set of keys. Our system provides a higher security through AES because of its complexity and helps in generating the content key(C_K). This key is used during the encryption of the original file over the cloud. Our methodology also focuses on deduplication to provide less consumption of cloud storage over the cloud. Another advantage of using this system is to provide an efficient way of data access via access policies for a certain set of credentials.

Keywords: *Access policy, CP-ABE, deduplication, cloud storage.*

1. Introduction

Cloud an important source of data storage that can be maintained, managed and backed up via remotely anytime. These cloud services not only provide easy access but also protects user's data to a greater extent from the third party [1], [2]. On the other hand, we are required to protect confidentiality for a set of users from the third party unauthorized access [3], [4]. In a recent survey conducted states that many users, as well as the organization shifting to the cloud, has doubled in the past few years. That's why our major task is to implement a much more protected system for data storage and certain flexibility to only those concerned users. To implement such a system we are required to store the user's data in an encrypted form. We try to maintain such confidentiality by access control policies such only the respective user is able to access its data in privacy. This data are stored in the cloud in an encrypted format and convenient to decrypt whenever required by the user. One such scheme for encryption is given by Attribute-Based Encryption (ABE). An ABE [5] is defined by a user's private key for a specific attribute set for a message that satisfies access policy for a set of attributes during encryption. Similarly for decrypting the same message the user uses its private key to obtain the required message.

A standard ABE doesn't provide deduplication [6], to achieve such a scheme in our system a methodology has been proposed by [base]. Deduplication majorly removes the redundant data stored over the cloud. Here in this paper, we try to implement such a technique for secure deduplication. There have been many researchers upon deduplication [7], [8], [9], [10] but none of them were implemented with attribute-based encryption as referred in [base]. Because of its bandwidth, cost savings and easy accessibility make it a much reliable system when both ABE and deduplication are implemented together.

The concept for implementing ABE with deduplication for an encrypted data is to store a file only once on the cloud. The goal is to avoid repetitive files stored in the cloud even upon receiving multiple copies of it satisfying the various access policies [11]. For example, if a data owner wants to upload a file on the cloud with some access policy A_A for a set of attributes. He/she encrypts the file to generate ciphertext, this ciphertext is uploaded to the cloud. The user who satisfies attribute set related to that file can access the ciphertext. Similarly, we consider another data owner who uploads the same ciphertext but now with different access policy$A_A{}'$ for a set of attributes. In such a scenario the data owner is able to upload a file on to the server leading to deduplication. This process results in space consumption and the wastage of cost for the cloud.

2. Related Work

Many researchers are conducted upon various ways to provide an efficient way of data storage over the cloud. We have shown interest in few of the related issues that provide a better way of ABE with secure Deduplication. Some of them are as follows,

Symmetric Key Encryption as a Building Block

A cryptographic technique utilizes the same set of keys for communication between two parties over cloud [12]. For example, Alice and Bob are the sender and receiver respectively. Suppose Alice wants to send a file to Bob, she uses Bob shared key with her message to generate a ciphertext. Bob uses this shared key to decrypt the ciphertext and generates the original message. Therefore a simple concept where both Alice and Bob have a same shared key. These keys are issued by the attribute authority as shown in Fig 1.

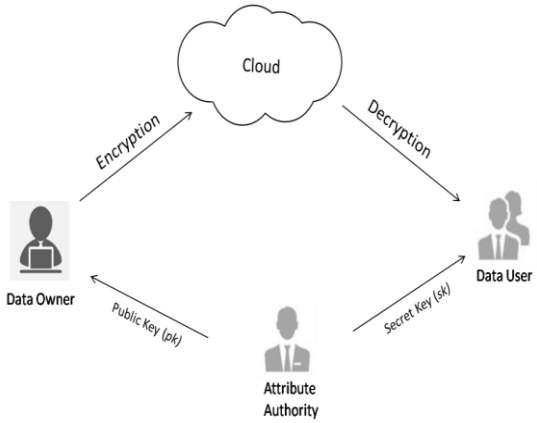

Fig. 1: A simple scenario of storing data over the cloud

Attribute-Based Encryption for Cloud Storage

The Attribute-Based Encryption (ABE) was introduced by [13], and its two basic forms Key-Policy Attribute-Based Encryption (KP-ABE) and Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [14]. There are many issues regarding the KP-ABE scheme for an access structure that is solved up to a certain extent [15], [16]. Overcoming the problem faced in KP-ABE, we use CP-ABE as it's more reliable. [17] defines a CP-ABE construction where the after issuing the private key an access policy is determined. An efficient scheme is described [18] to secure and support AND access structure. With an intention to limit the attribute space for a security parameter [19] considered a prime-order group.

Attribute-Based Encryption Scheme for Secure Deduplication

A novel approach for Attribute-Based Encryption [11] demands to save storage space and network bandwidth by eliminating duplicate copies of data over the cloud. The author considers a hybrid cloud architecture, where a system is managed by two clouds. Private cloud uses a trapdoor key for a particular access policy to generate ciphertext without disclosing the plaintext. In private cloud, the validity of proof is checked once the cloud gets a storage request. Once the proof is valid, the cloud checks for the corresponding tag to prevent storing of repetitive data. Therefore provides a protective way of sharing data with users for an access policy rather than decryption key.

Secure Deduplication over the Cloud

Deduplication is the methodology to avoid storing of repetitive files on the cloud. For example, consider two data owners uploading the same file to the cloud server. The cloud server stores only one copy of the file. Thus implementing deduplication concept for efficient data storage. Therefore reduces the cost and space for the data storage consumption over the cloud. [20] defines such methodology to balance out confidentiality and efficiency by restricting the storage of identical plaintext encrypted to the same ciphertext. [7], [8], [9] considers plaintext distribution based on public parameters. These systems couldn't resist brute force attack which was overcome by [9]. [9] focus on a server-aided encryption for deduplication storage.

Usage of Access Policies

There have been many policies mentioned to protect data over the cloud for a user. These policies provide stability, improve performance and solve user's basic security issues. A number of these policies are given by the author [11] to grant secure and easy encrypting process. Some of them are described below,

1. CP-ABE Policy

Ciphertext-Policy Attribute-Based Encryption (CP-ABE) is one of the major public key encryption schemes. CP-ABE consists of certain implementation steps such as $SetUp$ to set the data owner requirements, $KeyGen$ to generate a private key and master key attributes, Enc to check the validity presence of data, Dec to decrypt encrypted data. These steps are explained in detail,

Step 1: $SetUp(1^{\lambda})$ – Attribute Authority is responsible for generating keys. This step takes input λ as parameter produces output as public parameter $param$ and master private key msk.

Step 2: $KeyGen(param, msk, A)$ - For input public parameter $param$, the master private key msk and the attribute set A gives the output a secret key K_{SA}.

Step 3: $Enc(param, M, \mathrm{A_A})$ – Data owner runs this step, forwards sk_T and CT to the private cloud to ensure security. For input public parameter $param$, message M for an access structure $\mathrm{A_A}$. This leads to a trapdoor key generation sk_T and $CT = (T, L, ct, pf)$ where T denotes Tag, L represents the Label, ct as ciphertext and pf denote a proof of relationship on T, L, and ct for a message.

Step 4: $Dec(param, L, ct, A, sk_A)$ - For input $param$, L, ct, A and a secret key associated with an attribute set sk_A. sk_A should satisfy $\mathrm{A_A}, L, ct$ which is run by the user. Fig. 2 depicts the working of two clouds between the data owner, data user and the attribute authority.

2. Commitment Scheme

The commitment scheme is used to check data integrity. Data integrity ensures maintenance, accuracy, and consistency. This scheme includes generating tags and labels. Tags are used for encrypting data by verifying sk_T and $as\ code$. Labels are used for decrypting data by verifying $file$ and $signature$. The sk_T is generated during Step 3 of the encrypting process, the $as\ code$ is generated by the SHA algorithm for encrypting and decrypting the $file$.

The data owner wants to encrypt the $file$ in the cloud, he/she should generate a $as\ code$ by using a trapdoor secret key sk_T . The unauthorized access to the third party is restricted with the help of this key. For example, consider our $file$ HELLO and the respective hash code generated is 1e275. After encrypting the $file$ with the sk_T, the data owner verifies the $signature$ of the encrypted $hashcode$ i.e. 1e275 with the original $file$ content HELLO. If the $file$ content matches with the $hashcode$ denote that there has been no modification in the $file$. Thus satisfying commitment scheme for the two parties i.e. data owner and the user.

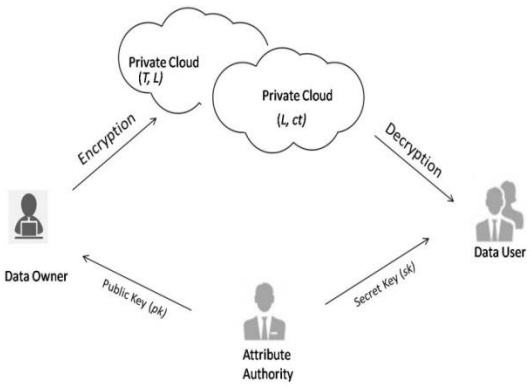

Fig. 2: Working with public and private clouds

3. Zero Proof of Knowledge

This proof identifies whether the stored data is duplicate or not. Consider a set of attributes branch, department, sub-department and job role. For example, branch Mumbai, department

Engineering, sub-department designing and job role as Java. These set of attributes are entered while a data owner registers itself on to the cloud. These attributes should match during file encryption. Here we use && (AND) and || (OR) operators for matching the required set of attributes. Once these attributes match we say the user can decrypt the file from the cloud.

AES

Advanced Encryption Standard (AES) a symmetric key encryption used to ensure higher security for encryption. AES was introduced by NIST to replace DES. When compared with other symmetric block ciphers, it's quite a complex system [21], [22], [23]. All the operations in the AES use a fixed length key size of 8-bit. Depending on the bit length size the AES are called by the wide range of names such as for 128, 192, 256 bits are referred to as AES-128, AES-192, and AES-256 respectively.
As AES is a symmetric block cipher that's depicted by 4 x 4 square matrix that takes the same key for encryption and decryption. The block cipher maintains a state array for each encryption and decryption stage. Finally, the output is copied to the output matrix. Consider a scenario for a 128-bit input, the first four bytes of plaintext are placed in the first column of the matrix after encryption. The second four bytes are placed in the second column and the process continues till the whole length of the input was taken. The AES cipher deals with *N* rounds, the number of rounds depends on key length. For example, 14 rounds for a 256-bit key. Depending on the number of rounds i.e. for *N*-1 rounds consists of four transformation stages. These are Substitute bytes, ShiftRows, MixColumns, and AddRoundKey. Round key of a 4 x 4 square matrix is passed to every transformation stage such that starting from Round 0 Key, Round 1 Key. . . Round *N*-1 Key, Round *N* key. To convert back the ciphertext to plaintext a set of reverse rounds using the same encryption key.

3. System Architecture

Our proposed system consists of three important modules given below,

Data Owner

The data owner is responsible for uploading the encrypted files on the cloud. The data owner always performs encryption process.

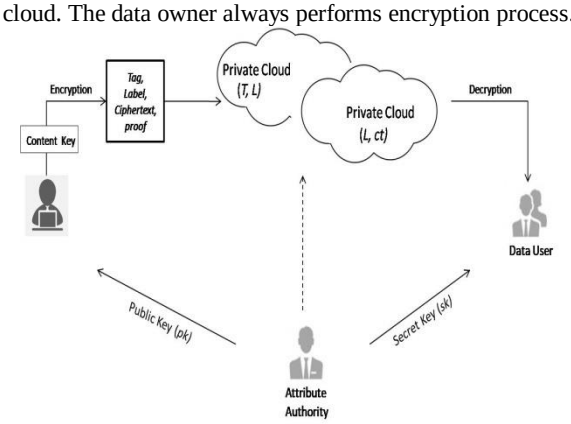

Fig. 3: Secure data storage with AES

Data User

The data user is responsible for decrypting the file from the cloud by using the secret key and the content key (C_K). The data user always performs the decryption process.

Cloud Server

The proposed system utilizes the cloud for data storage. This cloud is divided into two parts,

1. *Public Cloud* – This cloud performs equality test to attribute matching with the data owner. Whenever a mismatch occurs the file had to go through Re-encryption process.
2. *Private Cloud* – This cloud performs all the access policy testing. The file is uploaded to the private cloud when the data owner credentials and attributes match. This uploading of a file is carried out by Attribute Authority.

Attribute Authority

The Attribute Authority is responsible for distributing keys to the data owner and the data user i.e. secret key and public key. After encryption when a secret key is generated, it uses this key to transfer the file to the private cloud.

4. Proposed System

We implement an extension of CP-ABE with AES in addition to provide better security on the cloud. Here, we use AES-256 bit length to generate a ciphertext. This ciphertext is sent to various users, later decrypted by using the same key. We call this key as the content key, generated while file encryption by the data owner. We try to implement the same methodology as in symmetric Key Encryption for restricting the unauthorized access by an attacker. The data owner wants to encrypt the data on the cloud and the data user wants to decrypt the data from the cloud. To decrypt the data, a user wants public key as well as the secret key. These keys are provided by attribute authority. Our methodology provides a better option for higher security throughout the encryption and decryption.

Framework for Extended CP-ABE with AES

Step 1: $SetUp(1^{\lambda})$ – Attribute Authority is responsible for generating keys. This step takes input λ as parameter produces output as public parameter $param$ and master private key msk.
Step 2: $KeyGen(param, msk, A)$ - For input public parameter $param$, the master private key msk and the attribute set A gives the output a secret key K_{SA}.
Step 3: $Enc(C_K)-$ Data owner starts the encrypting process with the help of content key C_K which is generated by AES algorithm. This key is generated by the data owner while file encryption i.e. when converting the plaintext to ciphertext.
Step 4:$Enc(param, M, \mathrm{A_A})$ – Data owner runs this step, forwards sk_T and CT to the private cloud to ensure security. For input public parameter $param$, message M for an access structure $\mathrm{A_A}$. This leads to a trapdoor key generation sk_T and $CT = (T, L, ct, pf)$ where T denotes Tag, L represents the Label, ct as ciphertext and pf denote a proof of relationship on T, L, and ct for a message.
Step 5:$Validity - Test(param, CT)$ – This test is carried out in the private cloud. Input as $param$ and CT, this step performs tests on CT as (T, L, ct, pf) produces output 1 for a valid proof pf for(T, L, ct) or else 0.
Step 6: $Equality - Test(param, (T_1, L_1, ct_1), (T_2, L_2, ct_2))$ – Input as two tuple (T_1, L_1, ct_1) and (T_2, L_2, ct_2) with $param$, if both the tuples match 1 is generated or else 0.
Step 7:$Re-encrypt(param, sk_T, L, ct, A_A)$ – Input as $param$, sk_T, L, ct and an A_A produces an output a new ciphertext ct' with $A_A{}'$ having the same label L. This step is executed by the private cloud.
Step 8: $Dec(param, L, ct, A, sk_A)$ - For input $param$, L, ct, A and a secret key associated with an attribute set sk_A. sk_A should satisfy

A_A, L, ct which is run by the user. This step is executed by the user. C_K the content key is used to decrypt the ciphertext to the plaintext.[24]
Thus we implement an efficient, consistent, privacy and reliable algorithm for extending the CP-ABE with AES. Our proposed scheme presents reduced cost computation and contributes to less storage space.

5. Experimental Results

The following figures represent different steps of the proposed method. Fig 4 describes the usage of content key (C_K) during file encryption. Fig 5 depicts a set of attributes for a particular user. Fig 6 says about various access policies for the matching access structure.[25]

Fig. 4: File uploading with the content key (C_K)

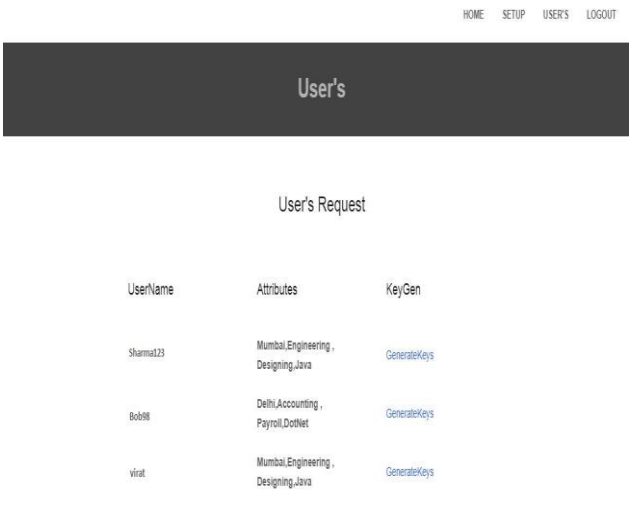

Fig. 5: Various attributes for specific User

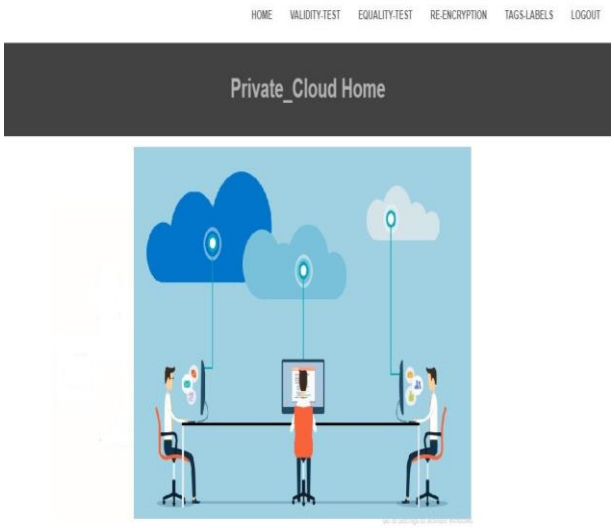

Fig. 6: Different access policies secure data storage

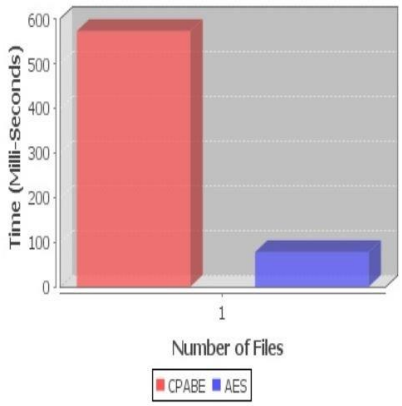

Fig. 7: Encryption performance

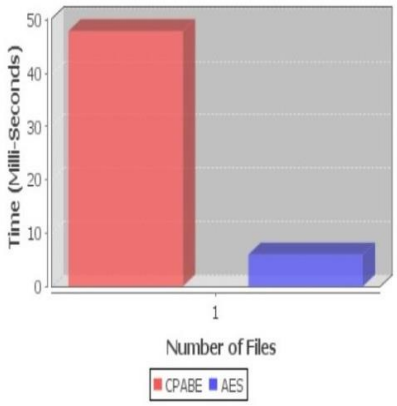

Fig. 8: Decryption performance

Fig 7 and Fig 8 depicts the encryption and decryption performance for the number of files and time taken for execution.

6. Conclusion

Access policies for secure deduplication one of a booming concept to provide security for an encrypted data. For a specific attribute set of a user, various access policies are checked to determine data validity, equality, and efficiency. We also perform re-encryption process for attributes of data owner during setup doesn't match the access structure while encryption. Our proposed work focuses on the extension of CP-ABE scheme with AES. Our method prevents the storage of multiple files on the cloud, resulting in

deduplication avoidance. Therefore reduces the cost of computation and controls the wastage of space over the cloud. Moreover, the usage of AES decreases the security issue and helps in providing a secure and transparent way of data storage.

References

[1] Quick D, Martini B & Choo KR, "Cloud Storage Forensics", *Syngress Publishing / Elsevier*, (2014).

[2] Choo KR, Domingo-Ferrer J & Zhang L, "Cloud cryptography: Theory, practice and future research directions", *Future Generation Comp. Syst.*, (2016), pp.51–53.

[3] Choo KR, Herman M, Iorga M & Martini B, "Cloud forensics: State-of-the-art and future directions", *Digital Investigation*, (2016), pp.77–78.

[4] Yang Y, Zhu H, Lu H, Weng J, Zhang Y & Choo KR, "Cloud based data sharing with fine-grained proxy re-encryption", *Pervasive and Mobile Computing*, (2016), pp.122–134.

[5] Sahai A & Waters B, "Fuzzy identity-based encryption", *Advances in Cryptology –EUROCRYPT*, (2005), pp.457–473.

[6] Zhu B, Li K & Patterson RH, "Avoiding the disk bottleneck in the data domain deduplication file system", *6th USENIX Conference on File and Storage Technologies, FAST*, (2008), pp.269–282.

[7] Bellare M, Keelveedhi S & Ristenpart T, "Message-locked encryption and secure deduplication", *Advances in Cryptology – EUROCRYPT*, Springer, (2013), pp.296–312.

[8] Abadi M, Boneh D, Mironov I, Raghunathan A & Segev G, "Message-locked encryption for lock-dependent messages", *Advances in Cryptology - CRYPTO*, Springer, (2013), pp.374–391.

[9] Keelveedhi S, Bellare M & Ristenpart T, "Dupless: Serveraided encryption for deduplicated storage", *Proceedings of the 22th USENIX Security Symposium*, (2013), pp.179–194.

[10] Bellare M & Keelveedhi S, "Interactive message-locked encryption and secure deduplication", *Public-Key Cryptography - PKC*, Springer, (2015), pp.516–538.

[11] Hui C, Robert HD &Yingjiu L, "Attribute-Based Storage Supporting Secure Deduplication of Encrypted Data in Cloud", *IEEE Transactions on Big Data*, (2016).

[12] Wu EF & Okamoto T, "Secure integration of asymmetric and symmetric encryption schemes", *J. Cryptology*, (2013), pp.80–101.

[13] Quick D & Choo KR, "Google drive: Forensic analysis of data remnants", *J. Network and Computer Applications*, (2014), pp.179–193.

[14] Goyal V, Pandey O, Sahai A & Waters B, "Attribute-based encryption for fine-grained access control of encrypted data", *ACM Conference on Computer and Communications Security, CCS*, Springer, (2006), pp.89–98.

[15] Ostrovsky R, Sahai A & Waters B, "Attribute-based encryption with non-monotonic access structures", *ACM Conference on Computer and Communications Security, CCS*, ACM, (2007), pp.195–203.

[16] Lewko AB & Waters B, "Unbounded HIBE and attribute based encryption", *Advances in Cryptology-EUROCRYPT*, Springer (2011), pp.547–567.

[17] Bethencourt J, Sahai A & Waters B, "Ciphertext-policy attribute-based encryption", *IEEE Symposium on Security and Privacy (S&P)*, (2007), pp.321–334.

[18] Cheung L & Newport CC, "Provably secure ciphertext policy ABE", *ACM Conference on Computer and Communications Security, CCS*, ACM, (2007), pp.456–465.

[19] Rouselakis Y & Waters B, "Practical constructions and new proof methods for large universe attribute-based encryption", *ACM SIGSAC Conference on Computer and Communications Security*, ACM, (2013), pp.463–474.

[20] Douceur JR, Adya A, Bolosky WJ, Simon D & Theimer M, "Reclaiming space from duplicate files in a serverless distributed file system", *ICDCS*, (2002), pp.617–624.

[21] Announcing the Advanced Encryption Standard (AES). *Federal Information Processing Standards Publication*, Vol.197, (2001).

[22] Sai Prasad K, Chandra Sekhar Reddy N, Rama B, Soujanya A & Ganesh D, "Analyzing and Predicting Academic Performance of Students Using Data Mining Techniques", *Journal of Advanced* Research in Dynamical and Control Systems, Vol.10, No.7, (2018), pp.259-266.

[23] Daemen J & Rijmen V, "AES proposal: Rijndael", (1999).

[24] Z Iskakova, M Sarsembayev, Z Kakenova (2018). Can Central Asia be integrated as asean? Opción, Año 33. 152-169.

[25] G Cely Galindo (2017) Del Prometeo griego al de la era-biós de la tecnociencia. Reflexiones bioéticas Opción, Año 33, No. 82 (2017):114-133