ISSN 0146-4116, Automatic Control and Computer Sciences, 2016, Vol. 50, No. 3, pp. 165–178. © Allerton Press, Inc., 2016.

Key Derivation Policy for Data Security and Data Integrity in Cloud Computing[1]

P. Senthil Kumari and A. R. Nadira Banu Kamal

Thassim Beevi Abdul Kader College for Women, Kilakarai, Tamil Nadu 623517, India

e-mail: senthilmathimca@gmail.com

Received November 23, 2015; in final form, March 10, 2016

Abstract—Cloud computing is currently emerging as a promising next-generation architecture in the Information Technology (IT) industry and education sector. The encoding process of state information from the data and protection are governed by the organizational access control policies. An encryption technique protects the data confidentiality from the unauthorized access leads to the development of fine-grained access control policies with user attributes. The Attribute-Based Encryption (ABE) verifies the intersection of attributes to the multiple sets. The handling of adding or revoking the users is difficult with respect to changes in policies. The inclusion of multiple encrypted copies for the same key raised the computational cost. This paper proposes an efficient Key Derivation Policy (KDP) for improvement of data security and integrity in the cloud and overcomes the problems in traditional methods. The local key generation process in proposed method includes the data attributes. The secret key is generated from the combination of local keys with the user attribute by a hash function. The original text is recovered from the ciphertext by the decryption process. The key sharing between data owner and user validates the data integrity referred MAC verification process. The proposed efficient KDP with MAC verification analyze the security issues and compared with the Cipher Text – Attribute-Based Encryption (CP-ABE) schemes on the performance parameters of encryption time, computational overhead and the average lifetime of key generation. The major advantage of proposed approach is the updating of public information and easy handling of adding/revoking of users in the cloud.

Keywords: attribute-based encryption (ABE), cloud computing, data integrity, data security, key derivation policy (KDP), secret key

DOI: 10.3103/S0146411616030032

1. INTRODUCTION

Recently, cloud computing is a significant technology in the Information Technology (IT) and Educational sectors. Cloud computing is a parallel and distributed computing and service-oriented architecture based on the virtualization. The significant features of the cloud computing are high operational efficiency, scalability, flexibility and minimum capital cost. Regardless of the great benefits, security, confidentiality, and regularity have become serious problems in the cloud computing application. The most prominent security concern in the cloud computing is data security and privacy, due to its web-based data storage and management. Users provide data to the cloud service provider for storage and business operations. Moreover, the entrepreneurs will face the critical consequences if their confidential data is disclosed to their business competitors or the public. Many data security techniques are developed to mitigate the security issues in the cloud. Current data security approaches focus only on cryptographic approaches where the solutions are derived by the random key generation processes. But, the prevailing security technique suffers minimum data integrity. Loss of key in the conventional cryptographic techniques crash the original data provided by the data owner. Figure 1 shows the system model of the key encryption process.

Key-based encryption techniques protect the data confidentiality and prevent the data from the unauthorized access. The utilization of encryption alone not provided the required security due to the various access control policies defined in research works. The fine-grained access control policies are defined based on user attributes. Hence, the design of access control policies required the identity attributes of the user. Hence, the research works are shifted into the Attributes Based Encryption (ABE) schemes. The

[1] The article is published in the original.

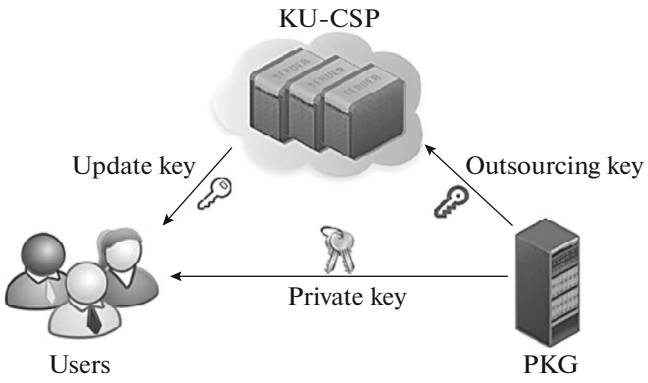

Fig. 1. System model of key encryption process.

ABE is a public key encryption technique that allows users to encrypt and decrypt messages, based on their attributes. In the ABE scheme, the cipher texts are not encrypted for a particular user. Rather, both the cipher texts and decryption keys are associated with a set of attributes or a policy over attributes. The user can decrypt a ciphertext only during the proper matching between the decryption key and the cipher text. ABE schemes are classified into key policy based ABE (KP-ABE) and cipher text-policy based ABE (CP-ABE). The KP-ABE scheme is based on the association of the attributes and decryption keys of the user. The CP-ABE scheme is based on the association of ciphertext policy and decryption keys of the user. In the KP-ABE scheme, a cipher text relates to the set of attributes. The decryption key of the user is associated with a monotonic tree access structure. The user can decrypt the cipher text, only when the user attribute related with the cipher text satisfies the tree access structure. The CP-ABE technique is extended to Hierarchical Attribute Set Based Encryption (HASBE) in order to design the scalable, flexible, fine-grained access control. The HASBE operation includes several processes such as system setup, domain authority grant validation, and file creation. The setup algorithm is used to setup the system public key parameters and master key parameters. The trusted authority domain verifies the new top-level domain authority when it requests to join the system. The administrative domain authority verifies the newly joined domain authority whether it is valid or not. The new encrypted file creation is based on the needs of the owner. The complexity of the new file creation depends upon the size of the domain authority data file. The HASBE technique increases the efficiency of user revocation in multiple values assignment environments. The key escrow problem induced in HASBE technique is considered by Multi-Authority Attribute-Based Encryption (MA-ABE). The main drawback of the ABE technique is the increase in the computational cost for key generation and encryption, and the user privacy encoded information is not protected and they have suffered the handling problem of simultaneous multiple users and multiple keys. To overcome these problems, this paper proposes an efficient Key Derivation Policy (KDP) to ensure data security and integrity in the cloud services. The proposed technique focuses on a robust secret key generation process and Message Authentication Code (MAC) verification process. The novel contributions of proposed efficient key derivation policy are listed as

• The multi-attributes-based secret key generation supports the effective adding and removal of many users.

• The robust key generation mechanism in the proposed efficient Key Derivation Policy (KDP) and the MAC verification based on block size have the capability to solve the simultaneous multi-users/keys handling problem.

• The hash-based mapping and the attributes decomposition-based secret key generation reduces the time complexities and improves the secure data transfer level.

The rest of the paper is structured as follows: section 2 includes the existing work related to the conventional encryption techniques for the cloud computing applications. Section 3 describes the detailed description of the proposed Efficient KDP including a robust secret key generation algorithm. Section 4 illustrates the simulation results of the proposed technique and section 5 presents the conclusion and future work of this paper.

2. RELATED WORKS

This section describes the conventional encryption techniques for the cloud computing applications. Wan et al. [1] proposed a Hierarchical Attribute-Set-Based Encryption (HASBE) technique for the scalable, flexible and fine-grained access control of the outsourced data in the cloud computing. The proposed scheme achieves the scalability and flexibility due to the hierarchical structure. The proposed scheme was efficient and flexible in dealing with the access control of the outsourced data in the cloud computing. Yang and Jia [2] proposed an efficient and privacy-protective auditing protocol for supporting the data dynamic operations in the cloud storage systems. The proposed protocol supports batch auditing for the multiple owners and clouds, without requiring any trusted organization. The efficiency and security of the proposed auditing protocols were improved while reducing the computation cost of the auditing process.

Li et al. [3] suggested a set of data access control mechanisms for the Personal Health Record (PHR) stored in the semi-trusted servers. The PHR file of the patient was encrypted, by using the Attribute-Based Encryption (ABE) techniques. Each user in the PHR system was divided into multiple security domains, to reduce the complexity in the key management for the data owners and users. The analytical and experimental results had shown the efficiency, security, and scalability of the proposed scheme. Wang et al. [4] proposed a secure cloud storage system for the simultaneous privacy-protective public auditing of the multiple users. The security and performance analysis had described that the proposed schemes were secure and highly efficient. Wang et al. [5] suggested a flexible auditing mechanism for the cloud storage, by using the homomorphic token and distributed erasure-coded data. The proposed mechanism was resistant against various failure and malicious attacks. Fast data error localization was achieved, without any increase in the communication and computation cost.

Wei et al. [6] proposed a Sec Cloud protocol for associating the secure storage and computation auditing in the cloud, by using the Designated Verifier Signature (DVS), batch verification and probabilistic sampling techniques. The effectiveness and efficiency of the proposed Sec Cloud were improved. Rewagad and Pawar [7] suggested the combination of the digital signature and Diffie-Hellman key exchange with the Advanced Encryption Standard (AES) algorithm to enable the protection of the data confidentiality in the cloud. The three-way mechanisms of the proposed architecture had made it more difficult to crash the security system. Sun et al. [8] presented an attribute-based keyword search scheme for independently encrypting and outsourcing data of the multiple owners to the cloud server. The owner-enforced access policy on the index of each file had achieved fine-grained search authorization. The proposed scheme was efficient and secure against the keyword attack.

Liu et al. [9] presented a clock-based proxy re-encryption scheme that enables the sharing of a secret key by the data owner and the cloud. The cloud has automatically performed re-encryption of data based on the internal clock, without receiving any command from the data owner. The proposed scheme had achieved scalable user revocation and fine-grained access control in the unreliable clouds. Alshehri et al. [10] suggested the utilization of the ciphertext policy based ABE technique to encrypt and decrypt the Electronic Health Record (EHR). The flexibility and scalability of the proposed approach were realized using the preliminary experimental results. Ruj et al. [11] proposed a distributed access control in the cloud algorithm to support the user revocation without the need for redistribution of the keys to all the cloud users. The computation, communication and storage overheads were reduced by the proposed approach.

Yang et al. [12] designed an access control framework with efficient attribute revocation method to match with the dynamic change in the access privileges of the users in large-scale systems. The proposed scheme was efficient and secure in the random oracle model. Wang et al. [13] proposed a hierarchical encryption scheme combining the identity-based encryption and ciphertext policy based encryption systems, to achieve fine-grained access control. The access rights were efficiently revoked from the users, by applying proxy and lazy re-encryption techniques to the proposed scheme. Li et al. [14] proposed a revocable Identity-Based Encryption (IBE) scheme for deploying a hybrid private key for each user. The efficiency and security of the proposed scheme were improved while achieving a reduction in the key generation complexity.

Zheng et al. [15] proposed a novel verifiable attribute-based keyword search scheme for the outsourced encrypted data. The performance evaluation had depicted that the proposed scheme was practical and deployable. Liu et al. [16] proposed a proxy re-encryption technique based on attribute and ciphertext policy, constructed in the composite order bilinear group. The proposed technique integrated the dual system encryption technology with a selective proof technique. Wu et al. [17] presented a Multi-message Ciphertext-Policy ABE technique, for sharing scalable media based on the attributes of the data users. The scheme was efficient and flexible while achieving a reduction in the computational complexity of the

cloud servers. Xu et al. [18] proposed a novel attribute-based encryption scheme to generate different class security keys for the users. The proposed scheme was simple, efficient and secure by using the hierarchical keys resulting from the one-way function chain. Li et al. [19] proposed Authorized Private Keyword Search (APKS) solution that enables the delegation and revocation of search capabilities. Efficient multi-dimensional keyword search was achieved by the proposed solution. Zhu et al. [20] presented an efficient time-based access control encryption scheme for the cloud services. The effectiveness and security of the encryption scheme were improved by using the cryptographic integer comparison. The traditional key-based encryption scheme such as Efficient Privacy-Preserving Demand Response Scheme (EPPDR) achieves the privacy preservation of demand, adaptive key evolution, and the forward secrecy. The problem in EPPDR is more computational overhead compared to other encryption methods. The Key Derivation Policies (KDP) required an efficient in the key generation process. The quality enhancement in outsourced data, a large number of users and the dynamic changed user to set and policies required the hierarchical process. Chen et al. proposed the new hierarchical key assignment [21] *CloudHKA* observed the user revocation issue. The utilization of *CloudHKA* to encrypt the outsourced data whether it is secure or not against the honest-but-curious cloud servers. They tested the *CloudHKA* scheme with the legal attacks issued by authorized data sources. On the basis of fine-grained access control policies, the selectively sharing of documents is the critical task in the public cloud. Multiple encrypted files on single keys raised the computational costs. Hence, an alternative technique is required to minimize the computational overhead in security applications. Nabeel et al. utilized the principle of dynamic sharing of symmetric keys during decryption avoided the public key cryptography. Based on this, they formalized the Broadcast Group Key Management (BGKM) [22], which provides the secrets to the users. On the basis of these secrets, the BGKM allows the derivation of asymmetric keys. Research works addressed the framework for efficient delivery and resource provisioning was required. Takabi et al. [23] focused the diverse policy management schemes based on the diverse languages. They introduced the policy management as a service designed to provide the unified control point. The overhead and the confidentiality were the important problems addressed. Nabeel et al. [24] performed coarse grained and fine grained two layer encryption. Upon two-layer encryption, decomposition of access control policies was the challenging issue. This problem referred as NP-hard problem. They overcome the problems by using an efficient group key management. The intensive operations such as data searching, multimedia processing in the mobile cloud processing raised the computational burden. Huang et al. [25] presented the new mobile cloud framework through trust management and private isolation. The chief drawbacks of the existing ABE and KDP schemes were expensive pairing operations and increase in the complexity and overhead of the admission policy. The time needed to decipher the cipher text was high, due to the great size of the cipher text. Hence, in order to overcome these limitations, this paper proposes an efficient KDP for enhanced data security and integrity in the cloud.

3. EFFICIENT KEY DERIVATION POLICY

This section describes the proposed efficient KDP for improving the data security and data integrity in the cloud. The proposed technique mainly focuses on the key security for the outsourced data in the cloud servers. The key generation algorithm provides secure access control mechanism and data access policies.

A. Key Derivation Policies

The data owners in cloud structure require the prevention of servers from learning the contents of unauthorized users. The inclusion of user and data attributes in the key derivation process created the new way of access control policies. The secret key generation based on these attributes provides the efficient encryption process, which reduces the time complexities with the secure data transfer. The requirements for efficient key derivation policies are listed as follows:

- Different users are authorized to access the different sets.
- The user access privileges must be revoked in an efficient way whenever required.
- Allowance for changes in pre-defined policies.
- The system should be scalable to a large number of users in terms of storage, computation and key management.

The increase in a number of users and owners increases the management complexities. The proposed efficient key derivation policies overcome the problem in two ways. At first, Attributes Based Encryption (ABE) is adopted to limit the complexity in encryption and user management; second, the division of users in the system into Security Domain (SD). The SD categorized into Public (PUDs) and Personal

Table 1. Notations used

Variables	Description	Variables	Description
λ	Security parameter	MK_k^{OKDP}	Master key for OKDP
g_1, g_2	Generator	PK^{OKDP}	Personal key for OKDP
H	Hash function	i	Each attributes in the set
v_k	Number of secrets	$K_{i,GID}$	Key generation
M	Message	C_t	Coefficients for each t element in the matrix
A	Attribute set	ρ	Mapping attribute
gp	Generators with the order p	α_i, y_i	Random exponents
P_k	Public key	e	Bilinear map function
S_k	Secret key		

(PSD). The PUD consists of a large number of users and multiple Public Attribute Authorities (PAA). The mapping of each PUD with the each sector makes the users acquire the credentials of authorities rather than the interaction with the owner. Initially, users obtain the local keys based on two attributes. The private keys and secret keys are generated by the logical operations (AND, XOR) performed between user and data attributes. Then, the owners in cloud upload the ABE encrypted files to the cloud server associated with the access control policies. Finally, there are two types of user revocation strategies namely, revocation of the user's attributes by using an Attributes Authority (AA) and the updating of access control policies for each document based on information from owner to the server. The two attributes such as data and role attributes are selected for proposed method. The intrinsic properties of data, referred by data attributes and the roles of entities, defined by role attributes.

The description of variables used in efficient key derivation policies is shown in Table1.

The process of encryption implemented by using the following algorithms:

- Global setup (λ)
- Authority setup (gp)
- Encrypt $\{M,(A,),gp,\{P_k\}\}$
- KeyGen (id,G_p,i,S_k)
- Decrypt $\{C_t, gp, \{K_{i,GID}\}\}$

(1) Global setup (λ)

The initial process in the encryption technique is the global setup. The setup process defines the input and output variables for the bilinear group (G) with generator (g). The master key for generators and public key for ABE process are generated by using the hash function. The hash function that maps the identities to the generators is given by following equation:

$$H : \{0,1\} \to g. \tag{1}$$

This hash function describes the random exponents for key generation processes. The master key and public keys are derived by using the following equations:

$$MK_k^{OKDP} = msk_k, \{t_{k,i}\}_{i \in A_k}, \tag{2}$$

$$PK^{OKDP} = Y = H\sum_k y_k, \{y_k, T_{k,i}\}. \tag{3}$$

The global setup defined by the public and personal keys initiate the encryption and decryption process.

(2) Authority setup (gp)

The two random exponents are generated for authority that belongs to each attribute i is given by

$$\alpha_i, y_i \in Z. \tag{4}$$

The equation (2) is used to generate the following keys: Based on the exponents the master key and the personal keys are modified in proposed system to assure the efficiency as follows:

$$\text{Public key } P_k = \left\{ e(g_1, g_2)^{\alpha_i}, g_2^{y_i} \text{ for } i \right\}, \tag{5}$$

$$\text{Secret key } S_k = \{\alpha_i, y_i \text{ for } i\}. \tag{6}$$

The encryption algorithm utilizes the P_k and S_k to generate the necessary policy coefficients

(3) Encrypt

The encryption algorithm uses the message M, $n \times l$ matrix A with ρ mapping of row attributes and global parameters and the public keys. The coefficients for key derivation policy is derived as

$$C_0 = Me(g_1, g_2)^s, \tag{7}$$

$$C_{1,x} = e(g_1, g_2)^{\lambda_x} e(g_1, g_2)^{\alpha_{\rho(x)} r_x}, \tag{8}$$

$$C_{2,x} = g_1^{r_x}, \tag{9}$$

$$C_{3,x} = g_1^{y_\rho(x) r_x} g_2^{\omega_x}. \tag{10}$$

The coefficients of key derivation policy are used to generate the key to correspond to the identity value of the message.

(4) Keygen

A key defines the unique labels for each attribute in the structure. The depth of the key structure is the level of recursions in the set. The members at depth 1 are either attribute elements or sets and members at depth 2 are attribute elements. Let us consider the hash function and the generator and identity. Then, the private key is generated by using the user and data attributes.

$$K_{i,GID} = g_1^{\alpha_i} H(GID)^{y_i}. \tag{11}$$

The generated key from equation (9) is used to decrypt the message signal.

(5) Decrypt

The decrypting process computes the coefficients for retrieving the message from encrypted format with the assumption such that the decryption has secret keys $\{K_{\rho(x),GID}\}$ subset of rows A_x of a matrix A as follows:

$$C_{1,x} \frac{e(H(GID), C_{2,x})}{e(K_{\rho(x),GID}, C_{2,x})} = e(g_1, g_2)^{\lambda_x} e(H(GID), g_2)^{w_x}. \tag{12}$$

The message computed from the coefficients is described by the following equation:

$$M = C_0 / e(g_1, g_2)^S. \tag{13}$$

The key security for the message transmission computed to ensure the security and integrity. The cloud repository is formed by the global setup with the security parameter (λ). After initiating the attribute set and associated keys, randomly taken exponents are used to set up the authority space. The public and secret keys base on the entropy based mapping function. The data owner raised the request through the message, which is decomposed into row, and column attributes. The coefficients are encrypted with the generators preferred. The hash based mapping function and the associated generators are used in key generation mechanisms. The ABE performed on the selected attributes with the generated keys. The hash based mapping, secret key generation based on attributes decomposition of message sequence into row and column format optimized the encryption process, which reduces the time complexities. Figure 2 shows the flow diagram of the encryption process and MAC verification process.

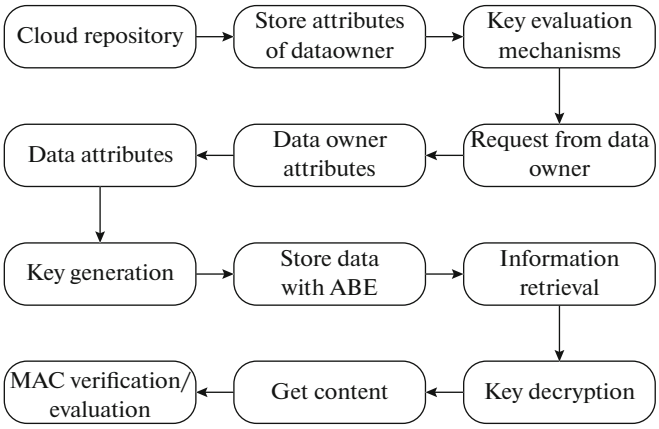

Fig. 2. Flow diagram of the ABE and MAC verification process.

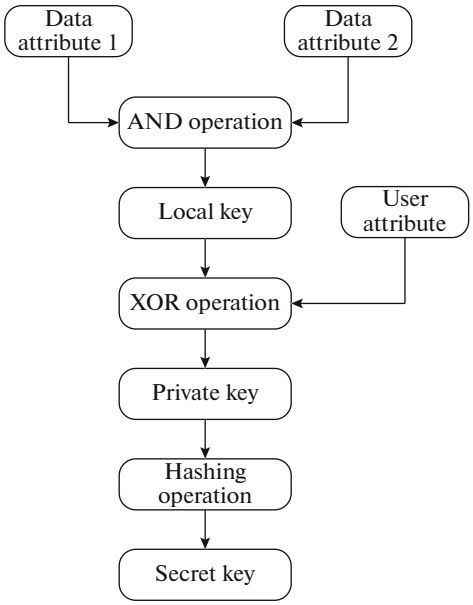

Fig. 3. Flow diagram for the secret key generation process.

B. Secret Key Generation Algorithm

Initially, the extraction of the data and user attributes are performed. Then, any two attributes are randomly selected. The AND operation is performed on the selected attributes. The resultant value of the AND operation is the local key. The exclusive OR (XOR) operation is performed with the local key and user attribute and a private key is generated. Then, the Hashing operation is performed to convert the private key into a secret key. When the users need to retrieve data, their request is transferred to the data owner by the third party provider. The data owner sends the secret key directly to the user. Using this secret key, the user can decrypt the cipher text obtained from the cloud, to get the original plain text. Figure 3 shows the flow diagram for the secret key generation process.

The key generation algorithm depends upon the two issues such as a secret key and the attributes of the user. The attribute authority receives the master key MK. Let (a_1) and (a_2) be the two user attributes. Local key L_k is generated by the intersection of a_1 and a_2 is given by following equation:

$$L_k = a_1 \cap a_2. \tag{14}$$

The private key is generated using the Ex-or operation of the L_k and a_3. The secret key K_E is generated by hashing the private key P_k. The cost function is performed using the secret key and the selected file. Finally, the encryption key is generated. The encryption key can be viewed as the form of equation (15)

$$K_E = H0(H1(F), P_k) \oplus H2(F). \quad (15)$$

Here $H0$, $H1$ and $H2$ are all cryptographic hash functions. The file F is encrypted with another key K, while K will be encrypted with K_E. The selected file (F) is encrypted and decrypted with the key generated (K_E). Finally, the computational cost is calculated for the proposed key generation process. The Boolean logic and the gates based process in the key generation process includes the attributes in the key generation process and derives the necessary efficient key derivation policies.

Secret Key Generation Algorithm

Input: File set Fs, New file Fn, Attributes (a_1, a_2, a_3)
Output: K_E = KeyGen (Hash (P_K))
Step 1: Start
Step 2: SEND UserAttr (Fs, Fn, Fext)
Step 3: RECEIVE UserAttr (Fs, Fn, Fext)
Step 4: Generate localKeyGen by using equation (14)
Step 5: Generate PrivateKeyGen $SK = a_3$ Exor L_k
Step 6: K_E = KeyGen (Hash (P_K))
Step 7: CF = Enc(K_E, F)
Step 8: STORE CF into cloud
Step 9: REQUEST for File Download
Step 10: DF = Dec (K_E, F)
Step 11: GET Cost (DF)
Step 12: Stop

C. MAC Verification Process

The MAC verifies the data integrity by using a secret key shared between the data owner and user. Different hash values are generated to indicate the unawareness of the secret key of the data owner. The MAC standard defines the cryptographic checksum, which is obtained by passing the data through a message authentication algorithm along with the user attributes. The formulation of MAC verification is given by the following equation

$$MAC(K_E, M) = H((K_E \; xor \; pad_o) | H((K_E \; xor \; pad_i) | M)). \quad (16)$$

The hash function H is used to compute the verified parameters such as secret key k, authenticated message m, inner pad and outer pad sequences (pad_o & pad_i). The algorithm to implement MAC verification as follows:

MAC verification

Input: *Key(K_E), message (M)*
Output*: Hash concatenation*
Step 1: Start
Step 2: check the size of keys greater than block size
Step 3: calculate the hash function, otherwise add the zero pads to the hash function
Step 4: Calculate the underlying hash function for within the block and XOR function
Step 5: Calculate the concatenated hash output
Step 6: Stop

The MAC utilizes a session key and message to detect both concatenated data modifications in the hash function. The data owner pre-computes MACs of the file using a set of secret keys and stores them locally, before data outsourcing. For each time during the auditing process, the data owner reveals a secret key to the cloud server and requests for a fresh-keyed MAC for verification. MAC verification process

Table 2. Encryption time vs. attributes

Attributes	Computational time (s)	
	CP-ABE	KDP
1	0.5	0.2
2	0.8	0.5
3	1.1	0.8
4	2.7	1
5	3.1	1.3
6	3.4	1.5
7	3.9	1.6
8	4.3	1.8
9	4.5	2.3
10	5.5	2.9

enables high data integrity since it covers all data blocks. The encrypted file is derived from user key is compared with the file derived from MAC process. If both are equal, then the data are not affected by the attacks. If it is not equal, then it shows the retrieved data what is corrupted by the unauthorized users.

4. PERFORMANCE ANALYSIS

This section presents the comparative analysis of the performance parameters such as computational time, computational overhead and average time to derive the keys with the optimization techniques and average time to generate the keys with optimization on the proposed KDP with the CP-ABE, EPPDR, and pseudo-random key generation subset cover.

A. Security Analysis

This section describes the security analysis of proposed KDP in following cases

• The efficient hash based encryption of data provides the confidentiality to unauthorized users assure resistance of collision.

• The allocation of specific time period to the user to receive the encryption/decryption key in the hash property assures the strong data privacy against non-authorized users.

• Secure revocation of user privileges whenever necessary carried out by the hash-based secret key generation satisfied the assumptions for access control policy formation.

B. Encryption Time

The time required to complete the encryption process is termed as computational time. When the number of attributes involved in the process increases, it increases the encryption time. The encryption time computed with ten key attributes is listed in Table 2.

It shows the variations of the encryption time with the number of attributes involved. The time for encryption increases to the maximum value in the traditional CP-ABE methods. The proposed KDP provides the minimum time required for the encryption process for a different number of attributes.

Figure 4 describes the relationship between the computational times with the number of attributes respectively. For the minimum attributes (1), the encryption time of CP-ABE and the KDP are 0.5 and 0.2 secs, and for maximum attributes (10), they provide 5.5 and 2.9 secs. The proposed KDP algorithm reduces the encryption time by 60 and 47.27% compared to CP-ABE due to the multi-attributes in single key generation.

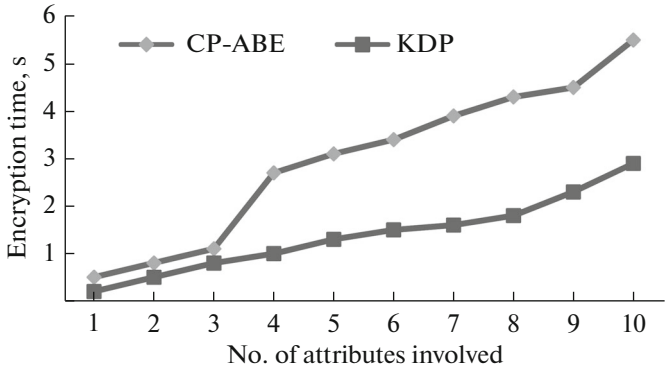

Fig. 4. Encryption time vs. no. of attributes.

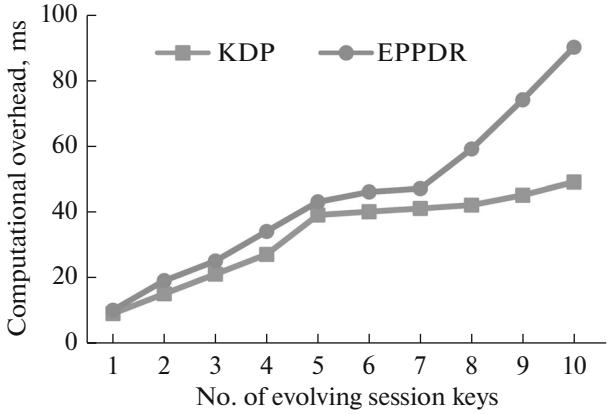

Fig. 5. Computational overhead vs. no. of evolving session keys.

C. Computational Overhead

The measure of the capability of the network to withstand the emulation attackers is called the computational overhead. When the number of attackers increases, the overhead is limited to achieve the authentication. The computational overhead is mathematically represented as follows:

Computational Overhead = Generated Keys + Encrypted Keys

The computational overhead computed with ten session keys is listed in Table 3. It shows the variations of the computational overhead with the number of session keys involved. The overhead increases to the maximum value in the traditional EPPDR methods. The proposed KDP provides the minimum overhead.

The relationship between the computational overhead and the number of session keys is described in Fig. 5. The number of session keys is increased and the network capability in terms of the computational overhead is computed. For the minimum session keys (1), the computational overhead of EPPDR and the KDP are 10 and 9 ms and for maximum session keys the overhead are 90 and 49 ms. The secret key generation through the multi-attributes participation in KDP reduces the overhead by 10 and 45.55% compared to EPPDR for minimum and maximum session keys.

D. Average Lifetime to Derive Keys

The lifetime is the important parameter in the design of the network. The speed of the packet transmission depends upon the lifetime to derive the keys of the data transmission when the network is in high traffic. The interval for key update increases, then the average lifetime to derive the keys is computed using the existing pseudo-random key generation algorithm and the proposed KDP algorithm. The simulation results confirm the effective increase in the lifetime. The average lifetime to derive the keys computed with

Table 3. Computational overhead vs. session keys

Session keys	Computational overhead (ms)	
	EPPDR	KDP
1	10	9
2	19	15
3	25	21
4	34	27
5	43	39
6	46	40
7	47	41
8	59	42
9	74	45
10	90	49

Table 4. Average lifetime for key derivation vs. key update interval

Key update interval	Average lifetime (ms)	
	pseudo random	KDP
1	234	200
2	274	208
3	434	256
4	466	341
5	500	490
6	530	504
7	561	541
8	714	547
9	939	638
10	993	684

ten different key update intervals is listed in Table 4. It shows the measures of the average lifetime for key derivation with the key update interval. The interval for updating process is more than the average lifetime for the derivation of keys. But, using the proposed KDP algorithm provides the minimum average lifetime compared to the pseudo-random key generation algorithm.

The interval for the key update is increased in the network that leads to the high network traffic. The measure of the traffic is expressed as the lifetime of the users. The relationship between the key update interval and lifetime are depicted in Fig. 6. For the minimum interval (1), the life-time for pseudo-random key generation and the multi-attributes key generation are 234 and 200 ms and for maximum intervals (10) the average lifetime values are 993 and 664 ms. The KDP reduces the average lifetime by 14.23 and 33.13% compared to pseudo-random generator for minimum and maximum update intervals.

Average Lifetime to Generate Keys

The key is the important parameter in the design of the network. The time to generate the keys depends upon the key update interval. The interval for key update increases, then the average lifetime to generate keys is computed using the subset cover and the proposed KDP algorithm. The simulation results confirm the effective increase in the average lifetime. The average lifetime to generate the keys computed with ten different intervals is listed in Table 5.

It shows the measures of the average lifetime of a key generation with the key update interval. The interval for updating process is more than the average lifetime for the derivation of keys. For the minimum interval (1), the life-time for subsetcover and the multi-attributes key generation are 200 and 112 ms and

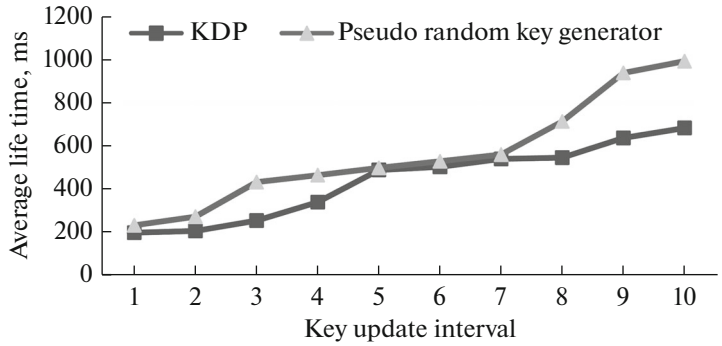

Fig. 6. Average lifetime vs. key update interval.

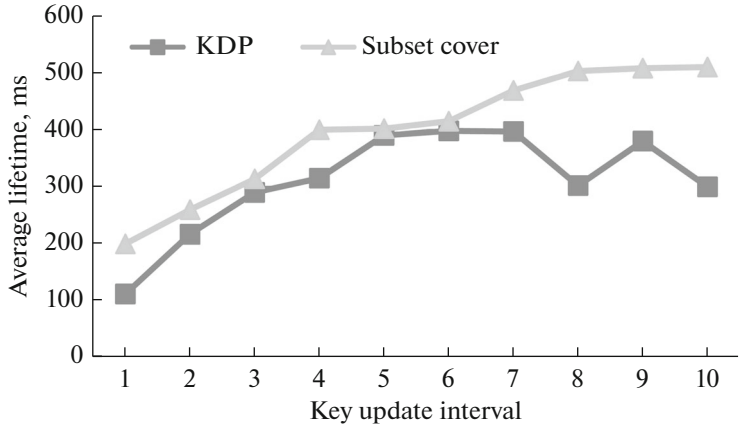

Fig. 7. Average lifetime vs. key update interval.

for maximum intervals (10) the average lifetime values are 510 and 300 ms. The KDP reduces the average lifetime by 44 and 41.18% compared to pseudo-random generator for minimum and maximum intervals.

The increase in the generated keys leads to high network traffic. The measure of traffic is expressed as a lifetime of the users. The relationship between the key update interval and lifetime are depicted in Fig. 7. The proposed method provides the minimum lifetime compared to the subset cover.

Table 5. Average lifetime for key generation vs. key update interval

Key update interval	Average lifetime (ms)	
	subset cover	KDP
1	200	112
2	260	217
3	314	290
4	400	315
5	402	390
6	415	398
7	469	397
8	503	302
9	508	380
10	510	300

5. CONCLUSION AND FUTURE WORK

In this paper, the problem in cloud computing addressed such that data contribution from multiple owners and search process by multiple users is the challenging scenario. Attribute-Based Encryption (ABE) provided the effective encryption based on user and data attributes. The Attribute-Based Encryption (ABE) verified the intersection of attributes to the multiple sets by access control policy. The handling of adding or revoking the users is difficult with respect to changes in policies. The inclusion of multiple encrypted copies for the same key raised the computational cost. This paper proposed an efficient Key Derivation Policy (KDP) for improvement of data security and integrity in the cloud and overcome the problems in traditional methods. The local key generation process in proposed efficient method included the data attributes. The secret key is generated from the combination of local keys with the user attribute by a hash function. The original text is recovered from the ciphertext by the decryption process. The key sharing between data owner and user validates the data integrity referred MAC verification process. The proposed hybrid processes efficient KDP with MAC verification analyze security issues and compared with the other Cipher Text- Attribute-Based Encryption (CP-ABE) schemes on the performance parameters of encryption time, computational overhead and an average lifetime to generate/derive keys. The major advantage of proposed approach is that the updating of public information leads to easy handling of adding/revoking of users and updating in access control policies. Hence, the future work shall be extended to provide an alternative approach to speed up the decryption time for low-end devices.

REFERENCES

1. Wan, Z., Liu, J.E., and Deng, R.H., HASBE: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing, *IEEE Trans. Inf. Forensics Secur.*, 2012, vol. 7, pp. 743–754.
2. Yang, K. and Jia, X., An efficient and secure dynamic auditing protocol for data storage in cloud computing, *IEEE Trans. Parallel Distrib. Syst.*, 2013, vol. 24, pp. 1717–1726.
3. Li, M., Yu, S., Zheng, Y., Ren, K., and Lou, W., Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption, *IEEE Trans. Parallel Distrib. Syst.*, 2013, vol. 24, pp. 131–143.
4. Wang, C, Chow, S.S., Wang, Q., Ren, K., and Lou, W., Privacy-preserving public auditing for secure cloud storage, *IEEE Trans. Comput.*, 2013, vol. 62, pp. 362–375.
5. Wang, C., Wang, Q., Ren, K., Cao, N., and Lou, W., Toward secure and dependable storage services in cloud computing, *IEEE Trans. Serv. Comput.*, 2012, vol. 5, pp. 220–232.
6. Wei, L., Zhu., H., Cao, Z., Dong, X., Jia, W., Chen, Y., et al., Security and privacy for storage and computation in cloud computing, *Inf. Sci.*, 2014, vol. 258, pp. 371–386.
7. Rewagad, P. and Pawar, Y., Use of digital signature with Diffie-Hellman key exchange and AES encryption algorithm to enhance data security in cloud computing, *2013 International Conference on Communication Systems and Network Technologies (CSNT)*, 2013, pp. 437–439.
8. Sun, W., Yu. S., Lou, W., Hou, Y.T., and Li, H., Protecting your right: Attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud, *2014 Proceedings IEEE INFOCOM*, 2014, pp. 226–234.
9. Liu, Q., Wang, G., and Wu, J., Clock-based proxy re-encryption scheme in unreliable clouds, *41st International Conference on Parallel Processing Workshops (ICPPW)*, 2012, pp. 304–305.
10. Alshehri, S., Radziszowski, S.P., and Raj, R.K., Secure access for healthcare data in the cloud using ciphertext-policy attribute-based encryption, *IEEE 28th International Conference on Data Engineering Workshops (ICDEW)*, 2012, pp. 143–146.
11. Ruj, S., Nayak, A., and Stojmenovic, I., DACC: Distributed access control in clouds, *IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2011, pp. 91–98.
12. Yang, K., Jia, X., and Ren, K., Attribute-based fine-grained access control with efficient revocation in cloud storage systems, *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, 2013, pp. 523–528.
13. Wang, G., Liu, Q., Wu, J., Guo, M., Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers, *Comput. Secur.*, vol. 30, pp. 320–331.
14. Li, J., Chen, X., Jia, C., Lou, W., Identity-Based Encryption with Outsourced Revocation in Cloud Computing, 2013.
15. Zheng, Q., Xu, S., and Ateniese, G., Vabks: Verifiable attribute-based keyword search over outsourced encrypted data, *2014 Proceedings IEEE INFOCOM*, 2014, pp. 522–530.
16. Liu, Q., Wang, G., and Wu, J., Time-based proxy re-encryption scheme for secure data sharing in a cloud environment, *Inf. Sci.*, 2014, vol. 258, pp. 355–370.
17. Wu, Y., Wei, Z., and Deng, H., Attribute-based access to scalable media in cloud-assisted content sharing, *IEEE Trans. Multimedia*, 2013, vol. 15, pp. 778–788.

18. Xu, D., Luo, F., Gao, L., and Tang, Z., Fine-grained document sharing using attribute-based encryption in cloud servers, *Third International Conference on Innovative Computing Technology (INTECH)*, 2013, pp. 65–70.
19. Li, M., Yu, S., Cao, N., Lou, W., Authorized private keyword search over encrypted data in cloud computing, *31st International Conference on Distributed Computing Systems (ICDCS)*, 2011, pp. 383–392.
20. Zhu, Y., Hu. H., Ahn, G.-J., Huang, D., and Wang, S., Towards temporal access control in cloud computing, *2012 Proceedings IEEE INFOCOM*, 2012, pp. 2576–2580.
21. Chen, Y.-R., Chu, C.-K., Tzeng, W.-G., and Zhou, J., CloudHKA: A cryptographic approach for hierarchical access control in cloud computing, in *Applied Cryptography and Network Security*, Jacobson, M., Locasto, M., Mohassel, P., and Safavi-Naini, R., Eds., Berlin-Heidelberg: Springer, 2013, vol. 7954, pp. 37–52.
22. Nabeel, M., Ning, S., and Bertino, E., Privacy preserving policy-based content sharing in public clouds, *IEEE Trans. Knowl. Data Eng.*, 2013, vol. 25, pp. 2602–2614.
23. Takabi, H. and Joshi, J.B.D., Policy management as a service: An approach to manage policy heterogeneity in cloud computing environment, *45th Hawaii International Conference on System Science (HICSS)*, 2012, pp. 5500–5508.
24. Nabeel, M. and Bertino, E., Privacy preserving delegated access control in public clouds, *IEEE Trans. Knowl. Data Eng.*, 2014, vol. 26, pp. 2268–2280.
25. Dijiang, H., Zhibin, Z., Le, X., Tianyi, X., and Yunji, Z., Secure data processing framework for mobile cloud computing, *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2011, pp. 614–618.