International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Review Paper

Encryption and Decryption of Attributes Based on One-Time Key Authorization for Secure Message Communication

[1]Mada Prasad*, [2]M. Jagadeeshwar, [3]Dr. D. Shanthi

Submitted: 07/02/2024 **Revised**: 13/03/2024 **Accepted**: 20/03/2024

Abstract: OTKA-AED framework is built with the intention of guaranteeing safe message exchange between owners and requesters of cloud data. Additionally, for every session number, the OTKA-AED framework obtains the encryption and decryption method based on the bilinear mapping transformation and reverse bilinear mapping transformation. In order to provide the public key and secret key, the OTKA-AED framework initially employs the one-time key generation function. This reduces the key generation time and hence increases cloud security. Finally, the permission tag-based encryption and decryption is carried out in accordance with the authorization tag, which ensures authorisation by effectively reducing communication and storage overhead. The proposed OTKA-AED architecture further ensures message processing within the cloud environment by increasing the security of message communication by encrypting key attributes shared among cloud users.

Keywords*: communication, environment, OTKA-AED, transformation*

1. Introduction

Both the academic and business worlds have given the cloud environment a lot of attention. The firm can easily share the various services in a secure manner using CC. The personal information of the cloud data owner is transferred to the cloud servers via numerous middlemen, where it is shared at any time with other cloud data requesters. The cloud environment allows the user to access the data at any time, from any location.

CC is on-demand network access used to assess the information and resources on the internet. The use of the CC services can secure the outsourcing of data security, data privacy, and service availability to 3rd parties. The owners of cloud data can send their message securely and at a high level of security. However, CC uses attribute encryption to ensure the protected message communication while leaving the authorisation factor unresolved.

1Research Scholar, Department of Computer Science, Chaitanya Deemed to Be University, Warangal Urban.

2Prof. M. Jagadeeshwar, Department of Computer Science, Chaitanya Deemed to Be University, Warangal Urban.

3Dr.D. Shanthi, Associate Professor, Department of Information Technology, Maturi Venkata Subba Rao Engineering College, Hyderabad.

**Corresponding authors email ID*

mp4unix@gmail.com

The OTKA-AED architecture is suggested as a solution for the previously mentioned constraint to enhance message security and speed up key generation. The public key and public key are generated among the cloud information proprietors and cloud servers. The public and secret keys are installed among cloud data owners and cloud servers employing a one-time key generation function. With these keys, Authorization Tag-based Attribute Encryption encrypts key attributes using the authorization tag, resulting in ciphertext with reduced computational overhead. This facilitates the execution of Authorization Tag-based Attribute Decryption, which decrypts the cypher text and retrieves the original message, will enable protected message communication in cloud service provisioning.

2. Literature Review

In 2016, Zhu and Jiang pioneered the development of a robust method for secure data sharing, focused on dynamic group-based key distribution and data sharing. The primary aim of this mechanism is outlined as follows: Instead of relying on conventional communication channels, the secure data sharing system establishes a framework to securely share keys. By using group manager, users can successfully access their private

keys without the need for certificate authorities according to user authentication.

The Searchable Symmetric Encryption Scheme with Rankings was developed by Cong Wang et al. (2012) to facilitate the effective usage of Cloud-based remote storage of encrypted data. Ranking search significantly improves system usability and, as a result, the accuracy of file retrieval by enabling the relevance rating of search results. It is crucial to carefully consider ranked search techniques such as relevance scoring, ranked search solution authentication, and command transfer from one to many in order to correctly maintain the sensitive score data.

A KP-ABE approach was presented by Changji Wang and Jianfa Luo (2013) to offer a higher level of security for CC supply. Additionally, the KP-ABE system enables senders to encrypt messages in the attributes, groups, and private keys are connected based on access structures that specify which type of cypher texts the key holder is permitted to decrypt. By utilising the Delerablee identity-based broadcast encryption system, the KP-ABE approach creates the constant cypher text size. This makes it possible to express the access strategy with any form of repetitive access structure. Amount of bilinear pairing is regarded as constant while cypher text size is self-governing to the number of cypher text attributes.

3. Proposed Work

The OTKA-AED framework has been developed to securely outsource confidential messages across cloud servers, utilizing third-party assistance and advanced key generation techniques. The CSP and cloud users (i.e., cloud data owners and cloud data requesters) are described by the OTKA-AED architecture as providing the session key and public keys for achieving message exchange in a secured way. This reduces the storage and communication costs associated with provisioning cloud services.

Let's consider the security factor "SK" when designing the suggested OTKA-AED framework. By selecting the two multiplicative cyclic groups "G1" and "G2" we may construct the bilinear map "f: G1* $G_1 \rightarrow G_2$". The cloud users are formed by assuming the cloud data owners are "$DO_i=DO_1$, DO_2,..., DO_n" and the cloud data requesters are "$DR_i=DR_1$, DR_2,...DR_n". The cypher text is created after the plain text, which is represented by the symbol "M" has been encrypted. The encryption and decryption are carried out by a third party, or "TP" and are recorded in a matrix with attributes set as "$\alpha=\{attr_1, attr_2,..., attr_n\}$" respectively.

3.1. One Time Key Generation

When implementing attribute encryption in the cloud, it's crucial to consider the challenges posed by one-time key generation. The utilization of centralized cloud storage results in nonlinear increases in key generation time and file size due to the accumulation of cloud data. In the suggested OTKA-AED framework, secured message communication is achieved by using the One Time Key Generation function, which generates the through a third party, cloud users can exchange public key and private key. Both the Public Key (PK) and Secret Key (K) are used in every session as the parameters for encryption and decryption. The session key SK_e^i with the i^{th} session for the e^{th} cloud user is represented as the message transmission 'M' together with the cypher text that contains the message to be encrypted.

Figure 3.1 An in-depth explanation of the One Time Key Generation

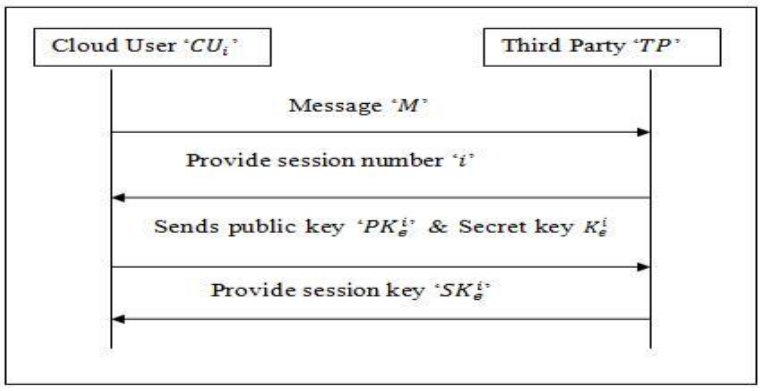

Figure3.1: One Time Key Generation

The suggested OTKA-AED framework serves as the foundation for the one-time key generation function diagram depicted in Figure 3.1. The cloud user "CUi" and the third party "TP" utilise the public key PK_e^i and secret key K_e^i for the e^{th} cloud user by the session number 'i' to assess the key generation function in order to achieve key synchronisation. This leads to the following expression for the public key and secret key's mathematical formula.

$$PK_e^i = RAN(CUi) \quad 3.1$$

$$K_e^i = KG(i, PK_e^{i-1}) \quad 3.2$$

The key generation is denoted by KG in Equations (3.1) and (3.2). Additionally, the suggested OTKA-AED framework is used to generate the novel session key PK_e^i utilising third-party TP at sessions. The session number 'i' is moved through the third party TP by utilising broadcast message MB is represented as, and the cloud user named random number generates the public key.

$$BM = \{i, AT, E_{PK_e^i}(M)\} \quad 3.3$$

The cypher text $E_{PK_e^i}(M)$ is denoted in the question (3.3), and the message "M" is encrypted using the cypher text and session keyPK_e^i. Additionally, the broadcast message "BM" contains the permission tag "AT". All lawful cloud requesters use the authorization tag, which only effectively checks message authentication. For this, the broadcast message is used to describe the public key PK and session keySK_e^i.

$$PK_e^i = \{M \to Attr, K_e^i, n, G_{1,} G_2\} \quad 3.4$$

$$SK_e^i = KG(i, \{E_{SK_e^i}(M),\} K_e^{i-1}) \quad 3.5$$

$$= H\left(i, ||E_{SK_e^i}(M)\right) \quad 3.6$$

K_e^i is the secret key designated to the cloud user "e" within the i^{th} session, according to Equation (3.5), where $M \to Attr$ stands for "attributes", "n" stands for "number of attributes" and "G_1" stands for the multiplicative cyclic group of the bilinear map G1*G1→G2. Therefore, the suggested OTKA-AED system enhances message security by quickly encrypting key qualities between cloud users and third parties. Additionally, Figure 3.2 provides an algorithmic description of the one-time key generation algorithm.

Input: Cloud Data Owners 'DO_i=DO_1,DO_2,...DO_n', Cloud Data Requesters 'DR_i=DR_1,DR_2,...,DR_n', Cloud users 'CU_i=CU_1,CU_2,...,CU_n', Cloud Server 'CS_i=CS_1,CS_2,...,CS_n'
Output: Minimized Key Generation Time
Step1: Begin
Step2: For each session 'i'
Step3: For each Cloud users 'CU_i' with Data Owners 'DO_i' and Data Requesters 'DR_i'
Step4: Measure public key using (3.1)
Step4: Measure secrete key using (3.2)
Step5: Evaluate broadcast message and public key using (3.4)
Step6: Evaluate session key using (3.5)
Step7: End for
Step8: End for
Step9: End

Figure3.2: One Time Key Generation Algorithm

The one-time key creation algorithm is shown in Figure 3.2 above. With the help of a one-time key generation process, the cloud user and cloud server are initially calculated for each session. By utilizing broadcasted messages, the key generation function is assessed for each session based on the session key, public key, and secret key, which reduces the time needed to generate the key effectively.

3.2 Encryption of attributes based on authorization tags

The proposed OTKA-AED uses attribute encryption to communicate the message of the cloud data owner with other requesters using the authorization tag provided by the cloud data owner. Messages delivered through cloud data owners generate the authorization tag "AT" during message broadcasting. After evaluating the authorization tag "AT" provided by cloud users using the suggested

OTKA-AED architecture, the encryption is carried out. Authorization Tag-based Attribute Encryption's activity diagram is shown in Figure 3.3 and is based on the OTKA-AED framework that is suggested below.

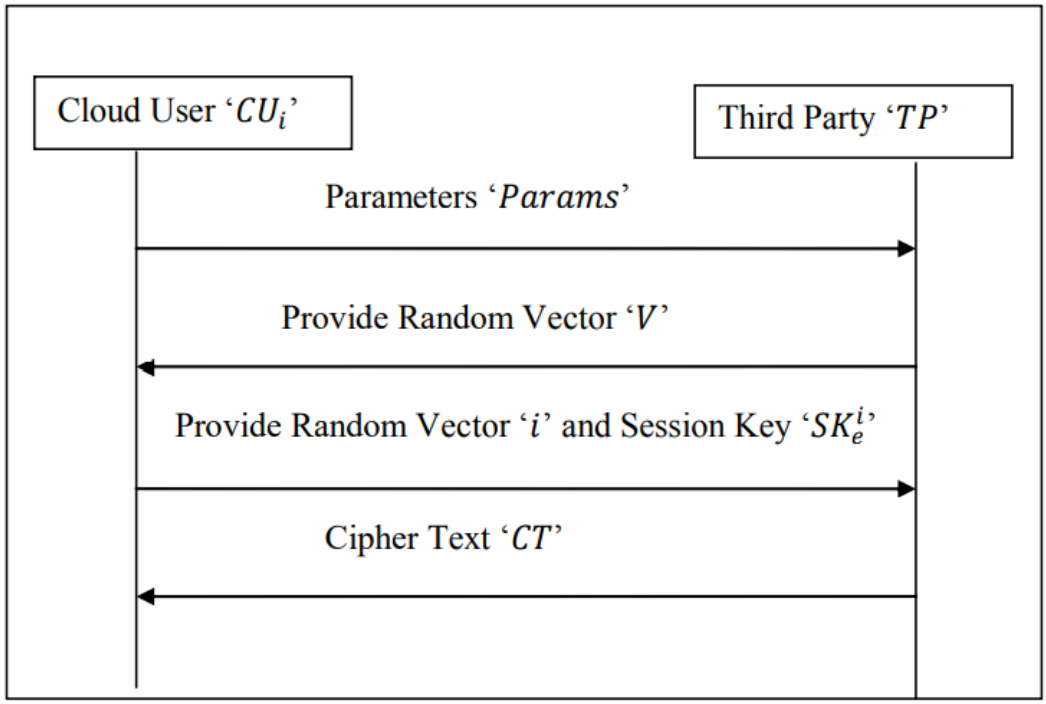

Figure 3.3 Activity Diagram of Authorization Tag-based Attribute Encryption

Figure 3.3 shows how to implement message encryption using the data owner and authorization tag-based attribute encryption. Let's assume that the message "M" the secret key "K" the matrix "MAT" the order "m*n" and 'α' all denote the rows of the matrix "MAT," correspondingly, to the attributes set 'α'.

$$Params = (M, K, MAT, \alpha) \quad 3.7$$

The supplied parameters are from Equation (3.7). The random vector is selected from a set of parameters that serve as the message's input and is symbolised by.

$$V = (e, attr_1, attr_2, \ldots . nattr_n) \in G_2 \quad 3.8$$

$$\beta = V * SK_e^i \quad 3.9$$

According to Equations (3.8) and (3.9), '*V*' is represented as the random vector that shares the encryption with each user e^{th} of the cloud and is evaluated in relation to the attribute values $attr_1, attr_2, \ldots . attr_n$, Additionally, using the OTKAAED framework, $r_i \in G_2$ makes a random choice for the exponent value. The following formalises the cypher test based on the exponent value.

$$C = M * e * f(G_1 * G_1) \quad 3.10$$

$$C_{i=} G_1 * H(\alpha(i))^{-r_i} \quad 3.11$$

$$CT = G_2^{\beta}(M, \alpha), C, C_i \quad 3.12$$

According to Equation (3.12), "CT" refers to the cypher test in which the owner of clou data uses the proposed OTKA-AED architecture to upload an encrypted message to a third party (TP). As a result, the message is considered to be encrypted while evaluating the permission tags that the cloud user has access to, effectively reducing the storage overhead. Additionally, Figure 3.4 illustrates the algorithmic procedure of the proposed OTKA-AED framework-based allowed attribute encryption algorithm.

Input: Message *'M'*, Secret Key *'K'*, set of attributes 'α', Cloud Data Owners 'DO_i=DO_1,DO_2,...DO_n', Cloud Data Requesters 'DR_i=DR_1,DR_2,...,DR_n', Cloud users 'CU_i=CU_1,CU_2,...,CU_n'

Output: Optimized Storage Overhead

Step1: Begin

Step2: For each Cloud users CU_i with Data Owners DO_i, Cloud Data Requesters DR_i, set of attributes 'α'

Step3: If $DOi(AT) = DRi(AT)$

Step4: Extract parameters using (3.7)
Step5: Evaluate random vector using (3.8)
Step6: Obtain cipher text *'CT'* using (3.10), (3.11) and (3.12)
Step7: End if
Step8: End for
Step9: End

Figure 3.4 Authorized Attribute Encryption Algorithm

Figure 3.4 above illustrates how cloud users' permission tags are verified to ensure that the permitted attribute encryption scheme is in place. Next, in order to reduce the computational complexity, the parameters are extracted in the form of a matrix and stored as a vector. Finally, the encrypted message is stored in the cloud server using the OTKA-AED architecture, which significantly reduces the storage overhead for the deployment of cloud services.

3.5.3 Decryption Based on Authorization Tags

The proposed OTKA-AED framework uses authorization tag-based attribute decryption in the cloud environment after conducting authorization tag-based attribute encryption. The approved cloud data requesters (DR_i) can easily access the message of the Cloud Data Owners (DO_i) after receiving the modified cypher text (CT). Additionally, the modified cypher text, or 'CT' is used as the input to data decryption, and the secret key K'^{i}_{e} is linked using the attribute specified in vector '*V*' which helps to satiate the approved tag being implemented with encrypted data. The message 'M' is then obtained based on the calculation shown below.

$$(CT, K_e^i, M) = \frac{(e*f(G_1*G_1)*M)}{(e*f(G_1*G_1))} \quad 3.13$$

$$(CT, K_e^i, M) = M \quad 3.14$$

By implementing the suggested OTKA-AED framework in the cloud, the encrypted data message is decrypted by the cloud requester and results in the actual message 'M" according to Equations (3.13) and (3.14). As a result, the examination of the authorization tag among cloud users, which improves the protected message transfer, results in a reduction in communication overhead. Using the OTKA-AED framework that is suggested below, Figure 3.5 illustrates the algorithmic explanation of the Authorized Attribute decryption algorithm.

Input: Message *'M'*, Secret Key *'K',* Cloud Data Requesters 'DR_i=DR_1,DR_2,...,DR_n', cipher test CT' Cloud Data Owners 'DO_i=DO_1,DO_2,...DO_n'
Output: Reduced Communication Overhead
Step1: Begin
Step2: For each Data Requesters DR'_i and Secret Key K'^{i}_{e}
Step3: $If\ (DR_i(AT)) = (CO_i(AT))$
Step4: $If\ (CT'_i = CT_i(AT))\ and (K'^{i}_{e} = K^i_e)$
Step5: Obtain the decrypted message using (3.13) and (3.14)
Step6: Else
Step7: Function not satisfied
Step8: Go to (2)
Step9: End if
Step10: End if
Step11: End for
Step12: End

Figure3.5: Authorized Attribute Decryption Algorithm

Figure 3.5 explains the approved attribute decryption algorithm, which decrypts the encrypted message based on the authorization tag that was acquired using the cloud requester by using the authorization strategy. Owners of cloud data and those making requests for cloud data both have equivalent authority over the permission tag. There is a decryption procedure used to recover the original data. With the cloud data owner and requester, a secret key is generated using the cypher text. Due to the comparison of permission tags, which results in less communication overhead and increased security in CC, the message is finally decrypted.

3.6 Experimental Otka-Aed Framework Procedure

In this section, the OTKA-AED framework is suggested and put into practise using the Amazon Simple Storage Service (Amazon S3) dataset and the Java programming language to provide secured message communication among cloud users. Data includes files, photos, and other types of essential information are stored in Amazon S3. For cloud service provisioning, an Amazon S3 provides a more dependable, quick, and affordable data storage architecture. For many services, Amazon S3 stores the data objects and enables concurrent read and write access. Information can be easily retrieved thanks to the read and write access for these data items.

The proposed OTKA-AED framework was created with cloud service provisioning in mind. The experimental evaluation is carried out using the OTKA-AED framework, and the outcomes are compared with the following parameters. The number of cloud users ranges from 10 to 100, and the file size is distributed into l pieces with data sizes ranging from 10 KB to 35 KB. The examination of the OTKA-AED architecture for secured message communication among cloud data owners and cloud requesters is evaluated using a variety of simulation metrics, including key generation time, storage overhead, and communication overhead, which are presented below.

3.7 Results And Discussion

The proposed OTKA-AED framework is compared to the existing approaches, including the KP-ABE scheme created by Changji Wang and Jianfa Luo, the Ranked Searchable Symmetric Encryption Scheme created by Wang et al. (2012), and the Secure Data Sharing Scheme created by Zhu and Jiang (2016). (2013). The effectiveness of the suggested OTKA-AED framework is assessed using the values shown in the tables and graphs as well as the following metrics.

3.7.1 Measure of Key Generation Time

The time needed to generate the public key and the secret key to enable secured message transmission among cloud users by a third party is referred to as key generation time in the proposed OTKA-AED framework. The following is a mathematical formulation of key generation time.

KG_{time}=Size of the attributes*Time(public key)*Time(secrete key) 3.15

From Equation (3.15), KG_{time} is defined as the key generation time, which is taken into account while determining the size, public key, and secret key that must be created for a cloud user.

It is quantified in milliseconds (ms). The process is said to be more effective when key generation takes less time.

Table 3.1 Tabulation for Key Generation Time

File size (KB)	Key Generation Time (ms)			
	OTKA-AED	**Existing Secure Data Sharing**	**Existing KP-ABE**	**Existing Ranked Searchable Symmetric Encryption**
10	0.031	0.043	0.058	0.075
20	0.037	0.05	0.064	0.081
30	0.041	0.053	0.068	0.085
40	0.039	0.052	0.067	0.084
50	0.044	0.057	0.072	0.089
60	0.047	0.06	0.075	0.092
70	0.046	0.059	0.074	0.091
80	0.054	0.067	0.081	0.098
90	0.058	0.072	0.086	0.103
100	0.061	0.075	0.089	0.106

Table 3.1 compares the key generation times for current methods such the KP-ABE by Changji Wang and Jianfa Luo and the ranked searchable symmetric encryption scheme by Wang et al. (2012) and the secure data sharing scheme by Zhu and Jiang (2016) with the proposed OTKA-AED framework (2013). The range of file sizes used for conducting experiments is 10 to 100. According to Table 3.1, all approaches have longer key generation times as file sizes grow. When compared to current approaches, the proposed OTKA-AED system, however, dramatically reduces the time required for key generation. Figure 3.6 shows the graph, which is produced based on the data in the table.

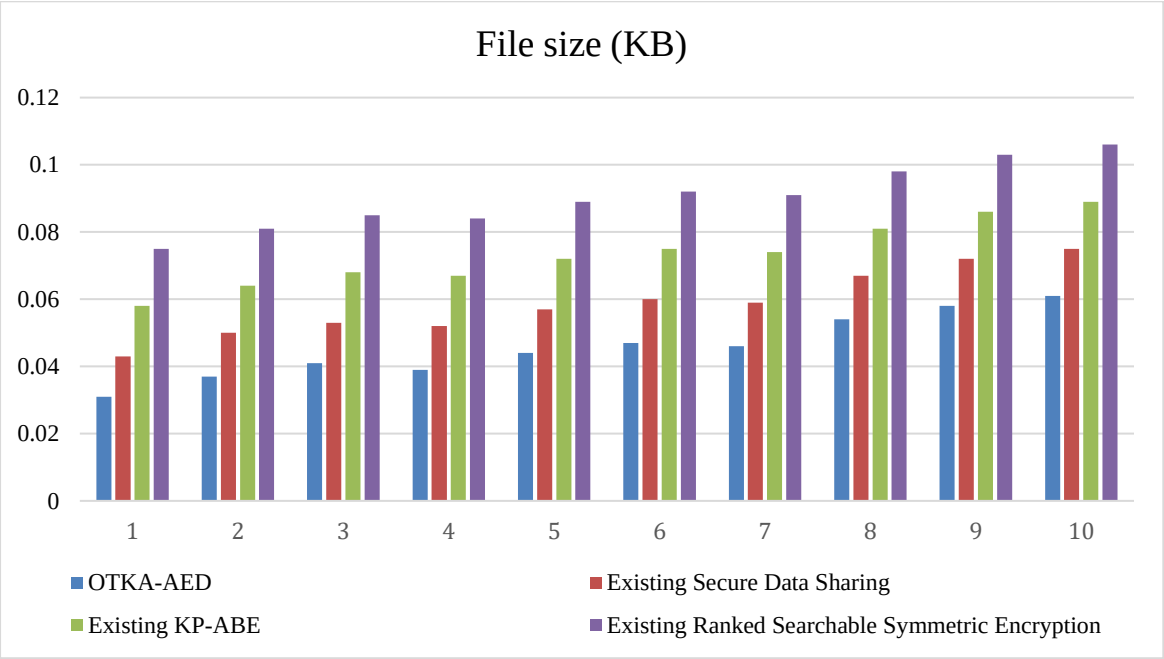

Figure 3.6 Measure of Key Generation Time

The proposed OTKA-AED framework's key generation time is depicted in Figure 3.6 and compared to the state-of-the-art approaches, such as the secure data sharing scheme developed by Zhu and Jiang (2016), the ranked searchable symmetric encryption scheme developed by Wang et al. (2012), and the KP-ABE developed by Changji Wang and Jianfa Luo (2013). The chart shows that, in comparison to current methods, the key generation time is significantly shorter. This is due to the fact that only the key characteristics are addressed by the One Time Key Generation method, and for each key attribute, the generation of the public key and the secret key is calculated with respect to the time, increasing the files ability to be read remotely. As a result, the suggested OTKA-AED framework for secure message exchange can be used in the cloud with confidence. As a result, the proposed OTKA-AED framework's key generation time is decreased by 23%, 38%, and 44%, respectively, in comparison to the existing ranked searchable symmetric encryption scheme by Wang et al. (2012), the existing secure data sharing scheme by Zhu & Jiang (2016), and the existing KP-ABE by Changji Wang & Jianfa Luo (2013).

3.7.2 Measure of Storage Overhead

The difference between the actual message size, the size of the characteristics, and the sum of the sizes of the authorization tags utilising time stamps is known as the storage overhead. It has the following formulation and is measured in kilobits per second (kbps).

$$SO = \frac{(Actual\ message\ size - Attribute\ size + Size\ of\ Authentication\ Tag)}{Timestamp} \quad 3.16$$

According to Equation (3.16), SO is defined as the CSP's storage overhead, which is mostly caused by the properties of the outsourced messages and the size of the authorization tags. Using the OTKA-AED architecture, the method is said to be more efficient if the rate of storage overhead is decreased.

Table 3.2 Tabulation for Storage Overhead

File size (KB)	Storage Overhead (Kbps)			
	Proposed OTKA-AED	Existing Secure Data Sharing	Existing KP-ABE	Existing Ranked Searchable Symmetric Encryption
10	101	168	225	246
20	185	245	300	329
30	235	295	352	372
40	349	409	467	487
50	397	452	504	524
60	449	509	563	583
70	520	580	632	652
80	532	600	646	666
90	554	615	660	680
100	570	630	671	691

Table 3.2 compares the storage overhead in relation to file size utilising the OTKA-AED framework as provided by Wang et al. (2012), the ranked searchable symmetric encryption system by Zhu & Jiang (2016), and the KP-ABE by Changji Wang & Jianfa Luo (2013). The size of the files is between 10 and 100. According to the table, storage overhead increases for all techniques as file size increases. However, it is much diminished in the OTKA-AED system. Figure 3.7 shows the graph, which is produced based on the data in the table.

Figure 3.7 explains the measure of storage overhead for the proposed OTKA-AED framework in comparison to the current approaches, including the secure data sharing scheme by Zhu & Jiang (2016), the ranked searchable symmetric encryption scheme by Wang et al. (2012), and the KP-ABE by Changji Wang & Jianfa Luo (2013).

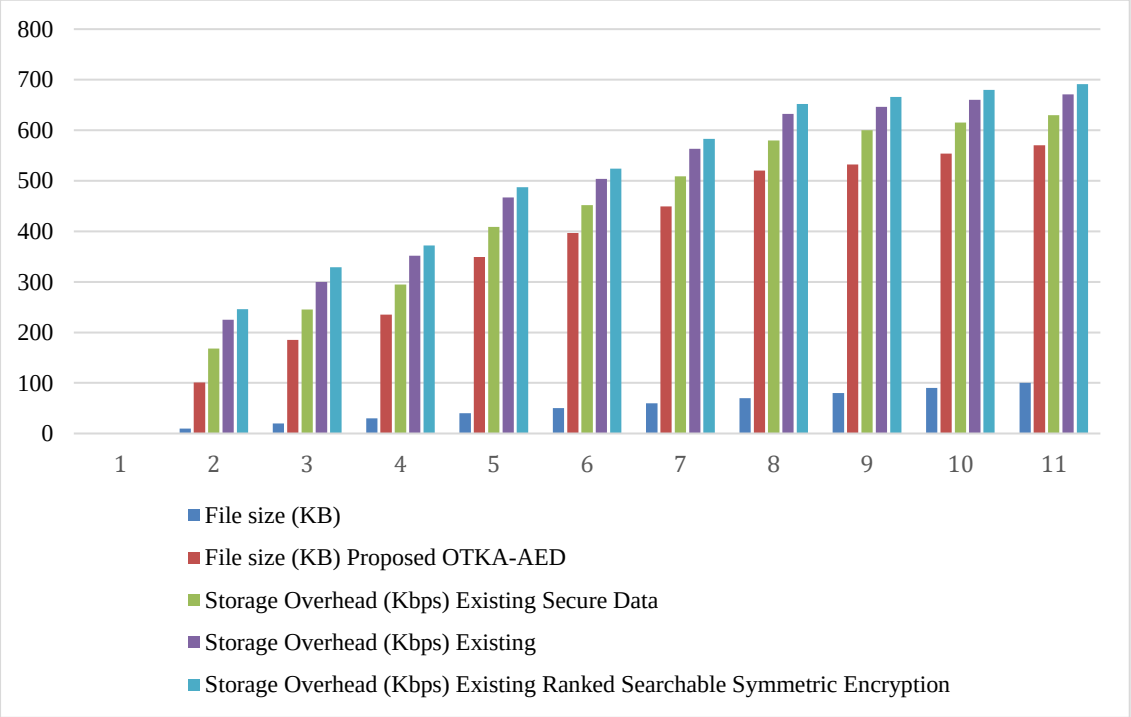

Figure 3.7 Measure of Storage Overhead

In comparison to the currently used approaches, the OTKA-AED system shown in Figure 3.7 above significantly reduces the storage overhead. This is because only the authorization tags are examined to possess the cloud data owners and cloud data requesters due to the Authorization Tag-based Attribute Encryption method. Additionally, utilising the OTKA-AED architecture, the message is only encrypted when the permission tag is accepted by both parties. Thus, when compared to existing secure data sharing schemes developed by Zhu & Jiang (2016), ranked searchable symmetric encryption schemes developed by Wang et al. (2012), and KP-ABE developed by Changji Wang & Jianfa Luo (2013), the proposed OTKA-AED framework reduces storage overhead by 16%, 29%, and 26%, respectively.

3.7.3 Measure of Communication Overhead

The suggested OTKA-AED framework defines the communication overhead as the ratio of the number of cloud users and the amount of message lost with regard to the specified timestamp. The following is a mathematical representation of the communication overhead.

$$CO = \frac{Number\ of\ cloud\ user * Message\ lost}{Timestamp} \quad 3.17$$

The term communication overhead (CO) is taken from Equation (3.17). The efficiency of the method is said to increase with lower communication overhead. Bits per second are used to evaluate it (bps).

Table 3.3 Tabulation for Communication Overhead

No. of Cloud Users	Communication Overhead (bps)			
	Proposed	**Existing Secure Data Sharing**	**Existing KP-ABE**	**Existing Ranked Searchable Symmetric Encryption**
10	10.3	12.4	14.2	17.1
20	16.2	18.3	20.5	23.4
30	21.6	23.8	26.1	29.5
40	27.1	29.9	31.4	35.3
50	32.4	34.1	36.7	39.6
60	38.7	41.4	44.3	47.8
70	47.5	49.8	52.5	55.7
80	52.8	55.5	59.2	62.9
90	58.2	60.2	63.1	66.4
100	62.1	63.7	67.9	70.3

Table 3.3 shows the communication overhead based on file size using the OTKA-AED framework that has been proposed as well as currently used techniques like the secure data sharing scheme developed by Zhu and Jiang (2016), the ranked searchable symmetric encryption scheme developed by Wang et al. (2012), and the KP-ABE developed by Changji Wang and Jianfa Luo (2013). The range of file sizes is 10 to 100. For all of the strategies in Table 3.3, increasing the file size also increases communication overhead. In contrast to current approaches, the suggested OTKA-AED framework effectively reduces the communication overhead. Figure 3.8 shows the graph, which is produced based on the data in the table.

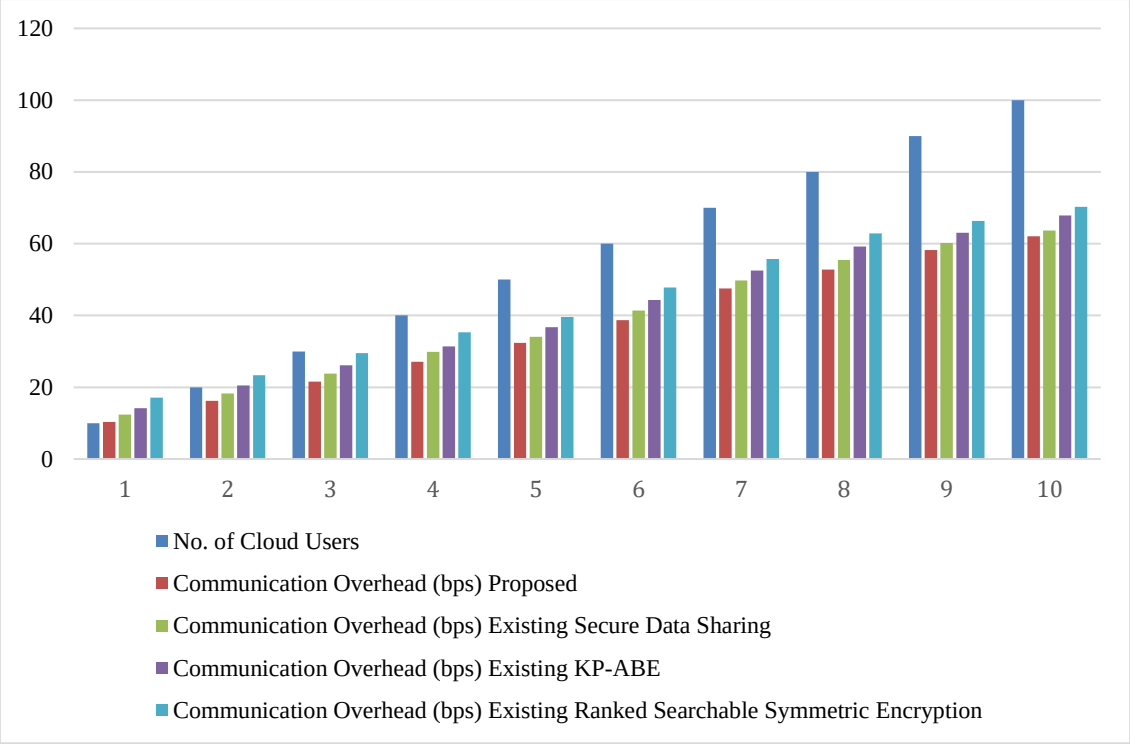

Figure 3.8 Measure of Communication Overhead

The proposed OTKA-AED framework's communication overhead is measured in Figure 3.8 and compared to existing approaches like the secure data sharing scheme developed by Zhu & Jiang (2016), the ranked searchable symmetric encryption scheme developed by Wang et al. (2012), and the KP-ABE developed by Changji Wang & Jianfa Luo (2013). The proposed OTKA-AED framework greatly reduces the communication overhead when compared to current approaches, as demonstrated in Figure 3.8. This is because the Authorized Attribute Decryption technique is being used, which decrypts the encrypted message using the authorization tag that the cloud requester possesses. The encryption content is converted to plain text using the comparison of the permission tag and secret tag, resulting in less communication overhead between cloud users. As a result, when compared to current approaches, the proposed OTKA-AED architecture reduces communication overhead by 7%, 21%, and 14%.

Conclusion:

The proposed OTKA-AED framework is built with the intention of guaranteeing safe message exchange between owners and requesters of cloud data. Additionally, for every session number, the OTKA-AED framework obtains the encryption and decryption method based on the bilinear mapping transformation and reverse bilinear mapping transformation. In order to provide the public key and secret key, the OTKA-AED framework first uses the one-time key generation function. This reduces the key generation time and hence increases cloud security. Finally, the permission tag-based encryption and decryption is carried out in accordance with the authorization tag, which ensures authorisation by effectively reducing communication and storage overhead. The proposed OTKA-AED architecture further ensures message processing within the cloud environment by increasing the security of message communication by encrypting key attributes shared among cloud users.

Additionally, the effectiveness of the suggested OTKA-AED framework is evaluated using the following metrics for cloud service provisioning, including key generation time, storage overhead, and communication overhead. The simulation findings show that the proposed OTKA-AED framework, when compared to state-of-the-art works, decreases key generation time by 35%, storage overhead by 24%, and communication overhead by 14%. However, simply verifying cloud data is insufficient. To provide an optimum cloud service provider, the proposed Fuzzy K-Means and K-Medoids algorithms are created. Additionally, the Algorithms for fuzzy K-Means and K-Medoids,

which are covered in more detail in the next section, greatly reduce the encryption time in CC by authenticating the cloud data.

References:

[1] Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P., & Shi, H. (2008). "Revisiting Searchable Encryption: Consistency residences, Relation to anonymous IBE, and Extensions." journal of Cryptology, 21(three), 350-391.

[2] Atallah, M., Kerschbaum, F., & Du, W. (2003). "comfortable and private sequence Comparisons." complaints of the 2003 ACM Workshop on privateness inside the digital Society (WPES'03), 39-forty four.

[3] Bellare, M., Boldyreva, A., & ONeill, A. (2007). "Deterministic and effectively Searchable Encryption." lawsuits of CRYPTO.

[4] Bloom, B. H. (1970). "area/Time change-offs in Hash Coding with Allowable mistakes." Communications of the ACM, 13, 422-426.

[5] Blum, M., & Goldwasser, S. (1984). "An green Probabilistic Public-Key Encryption That Hides All Partial statistics." lawsuits Crypto'eighty four, Springer-Verlag.

[6] Boldyreva, A., Chenette, N., Lee, Y., & O'Neill, A. (2009). "Order-retaining Symmetric Encryption." court cases of Eurocrypt 09, vol. 5479, LNCS. Springer.

[7] Boneh, D., Crescenzo, G. D., Ostrovsky, R., & Persiano, G. (2004). "Public Key Encryption with keyword search." lawsuits of EUROCRYPT.

[8] Bringer, J., Chabanne, H., & Kindarji, B. (2009). "error-Tolerant Searchable Encryption." complaints IEEE ICC.

[9] Chow, R., Golle, P., Jakobsson, M., Masuoka, R., Molina, J., Shi, E., & Staddon, J. (2009). "Controlling information within the Cloud: Outsourcing Computation with out Outsourcing control." proceedings of the ACM Cloud Computing security Workshop.

[10] Curtmola, R., Garay, J. A., Kamara, S., & Ostrovsky, R. (2006). "Searchable Symmetric Encryption: progressed Definitions and efficient constructions." lawsuits of the ACM CCS.

[11] Goh, E-J. (2003). "cozy Indexes." Technical record 2003/216, IACR ePrint Cryptography Archive. Retrieved from http://eprint.iacr.org/2003/216.

[12] Kahveci, T., & Singh, A. (2001). "An green Index structure for String Databases." court cases of the 27th global convention on Very huge Databases. Morgan Kaufmann, San Francisco, CA, 351-360.

[13] Levenshtein, V. (1966). "Binary Codes able to Correcting Deletions, Insertions, and Reversals." Soviet Physics Doklady, 10(8), 707-710.

[14] Li, J., Wang, Q., Wang, C., Cao, N., Ren, ok., & Lou, W. (2010). "Fuzzy key-word seek over Encrypted records in Cloud Computing." lawsuits IEEE INFOCOM.

[15] RFC. Request For remarks Database. Retrieved from http://www.ietf.org/rfc.html.

[16] D., Wagner, D., & Perrig, A. (2000). "sensible techniques for searching on Encrypted data." court cases of the IEEE Symposium on studies in security and privacy, 44-45.

[17] Stallings, W. (2002). "Cryptography and community safety: standards and practice" fifth ed. Pearson education.

[18] Survey. (2000). "NPD seek and Portal site Survey." Retrieved September 26, 2005, from http://www.searchenginewatch.com/sereport/article.personal home page/2162791.

[19] Wang, P., Berry, M. W., & Yang, Y. (2003). "Mining Longitudinal web Queries: tendencies and styles." magazine of the yank Society for records science and technology, fifty four(8), 743-758.

[20] Winkler, W. E. (1999). "The nation of document Linkage and present day studies problems." records of earnings department, inner revenue provider book R99/04. available from http://www.census.gov/srd/www/byname.html.