Asymmetric Key Blum-Goldwasser Cryptography for Cloud Services Communication Security

R. Senthilkumar[1], B. G. Geetha[2]

[1] Department of Computer Science and Engineering, Shree Venkateshwara Hi-Tech Engineering College, India
[2] Department of Computer Science and Engineering, K. S. Rangasamy College of Technology, India
{yoursrsk, geethaksrct}@gmail.com

Abstract

Cloud computing is a promising technology that provides different types of services such as data sharing and distribution. The Asymmetric Key Blum–Goldwasser Cryptography (AKBGC) technique is proposed to enhance the communication security of cloud services with minimum overhead. The users transmit requests to a cloud server to get required cloud services. The cloud server provides desired services to users in the cloud with aid of Blum-Goldwasser Cryptography (BGC) to attain higher confidentiality and data security. The AKBGC technique used Probabilistic encryption algorithm of BGC in order to encrypt the user needed cloud data with a public key of the receiver before transmission. This encryption process results in the generation of ciphertext and sent to users in a cloud. At the receiver side, AKBGC technique performs a key authentication process in order to access the original cloud data. The users can reconstruct original data with aid of deterministic decryption process in BGC when public key of both sender and receiver is identical. This helps for AKBGC technique to get higher communication security for cloud service provisioning and performs experimental evaluation using metrics such as data confidentiality, communication overhead, space complexity, and throughput. The result shows that the AKBGC technique is able to increases the data confidentiality and reduce the communication overhead of cloud services.

Keywords: Cloud Services-Communication, Security-Blum, Goldwasser Cryptography-Key, Authentication- Probabilistic Encryption

1 Introduction

Security and privacy are significant concerns in cloud services. The current research issue in cloud services provisioning is the authentication of users to secure communications. There are many cryptographic techniques designed to permit the access of only valid users to their services. In the real world, hundreds of user's access cloud services simultaneously where the authentication process of the user who accesses cloud data must be efficient with minimal overhead.

Hence, there is a requirement for a novel technique to achieve higher communication security in a cloud environment with minimum time.

An Inter-Cloud Virtual Private Network (ICVPN) solution was presented in [1] for secure communication of users who wants cloud services across multiple cloud platforms. The throughput of cloud service using ICVPN was lower. A ciphertext-policy attribute-based encryption (CP-ABE) was presented in [2] to perform confidential communications among fog nodes in a cloud environment. The communication overhead of cloud service using CP-ABE was higher.

A SecureSense was designed in [3] to perform secure end-to-end communication in the cloud-connected internet of things. The data confidentiality of cloud service using SecureSense was not at the required level. A Shibboleth protocol was designed in [4] for enhancing secure communication among fog clients in a cloud environment. The data security using this technique was not adequate.

A novel model was intended in [5] to solve security and privacy risks in the cloud environment. The authentication performance of this model was not efficient for attaining higher security in the cloud. A mobile user authentication scheme was introduced in [6] to secure mobile cloud computing services with the application of cryptographic hash functions. The space complexity involved during cloud service provisioning was not solved.

A novel technique was designed in [7] for attaining secure data privacy preservation for on-demand cloud service. The time required for secure cloud services was more. The Bell-LaPadula Multi-Level security model was designed in [8] to address security requirements in cloud. The average rate of successful data delivery using Bell-LaPadula Multi-Level security model was not sufficient.

A security and privacy issues handled in cloud-assisted wireless wearable communications was analyzed in [9]. A Privacy-Aware Authentication Scheme

*Corresponding Author: R. Senthilkumar; E-mail: yoursrsk@gmail.com

DOI: 10.3966/160792642020072104003

was intended in [10] to minimize authentication processing time of communication between cloud service providers. The confidentiality of cloud data services using a privacy-aware authentication scheme was not solved. In order to solve the above mentioned existing issues, Asymmetric Key Blum–Goldwasser Cryptography (AKBGC) technique is introduced. The main contributions of AKBGC technique are formulated as follows,

To enhance the security of data communication during cloud services provisioning as compared to state-of-the-art works, probabilistic key generation used in AKBGC technique.

This probabilistic key generation process constructs unique public and private key by randomly choosing a large prime numbers for each cloud users. With aid of generated key pairs, AKBGC technique prevents original data from adversaries for getting the decryption key for an encrypted data.

To reduce the communication overhead and to improve the throughput of cloud services, Probabilistic encryption and deterministic decryption algorithm of BGC is employed in AKBGC technique. With aid of Probabilistic encryption algorithm, AKBGC technique produce diverse ciphertexts for each time when it is encrypted using same public key. This supports for AKBGC technique to prevent data from an adversary by comparing them to a dictionary of known ciphertexts on the contrary to conventional works. The proposed BGC is a very efficient for both data encryption and decryption as compared to the conventional RSA encryption. The BGC performs faster data encryption than the existing works as where it based on linear time and BBS stream cipher. Also, The BGC performs quicker decryption for llarger size of data as compared to state-of-the-art works.

The rest of the paper is ordered as follows. In Section 2, the proposed AKBGC technique is explained with the assist of the architecture diagram. The Experimental settings and results analysis is discussed in Section 3 and Section 4. Section 5 explains the related works. Section 6 presents the conclusion of paper.

2 Asymmetric Key Blum-Goldwasser Cryptography Technique for Secured Cloud Services

The Asymmetric Key Blum-Goldwasser Cryptography (AKBGC) technique is designed with the objective of increasing the communication security and improving data confidentiality of cloud services with minimal time. The AKBGC technique used Blum-Goldwasser Cryptography (BGC) in order to attain secure and trustworthy communications during cloud service provisioning. The BGC employed in AKBGC technique that includes three main algorithms such as probabilistic key generation algorithm, a probabilistic encryption algorithm, and deterministic decryption algorithm. The probabilistic key generation algorithm in BGC creates public and private keys for each user in the cloud and thereby avoids an adversary from simply running the key generator to obtain a decryption key. Probabilistic encryption algorithm in BGC helps for AKBGC technique to encrypt the user desired data services before transmission. Besides deterministic decryption algorithm used in BGC assists for decrypting the ciphertext after key authentication process.

On the contrary to existing secured communication techniques designed for on-demand cloud services, a proposed technique used BGC because which is semantically secure based on the considered intractability of factorization during key generation. In addition to that, BGC has numerous advantages over conventional cryptography schemes as follows. The semantic security of BGC reduces solely to integer factorization without the need of any additional assumptions. As well, BGC is efficient in terms of storage, as a constant-size ciphertext expansion regardless of message length. Further, BGC is relatively efficient in terms of computation than conventional RSA (Rivest-Shamir-Adleman).

In BGC, encryption is carried out with help of a probabilistic algorithm. Therefore a given plaintext may generate very different ciphertexts in each time when it is encrypted. This is one of the most significant advantages of using BGC in the proposed technique as it prevents an adversary from recognizing messages by comparing them to a dictionary of known ciphertexts. As a result, proposed AKBGC technique obtains higher security and data delivery than a conventional cryptography scheme. The architecture diagram of AKBGC technique for communication security of cloud services is shown in above Figure 1.

Figure 1 depicts the process of AKBGC technique to secure communication of cloud services with reduced overhead. As demonstrated in the Figure 1, the requests are made from users to the cloud server. The cloud server presents user desired data services to users in a cloud environment. Before providing the user requested services, AKBGC technique encrypts it with aid of probabilistic encryption algorithm in order to preserve the privacy and security of data during transmission. After completing the probabilistic encryption process, AKBGC technique transmits the ciphertext to corresponding users in the cloud environment. Finally, the receiver can decrypt the ciphertext if the user is an authentic person using deterministic decryption process. Thus, AKBGC technique obtains higher communication security and data confidentiality for cloud services provisioning. The detailed process of AKBGC technique is shown in below subsections.

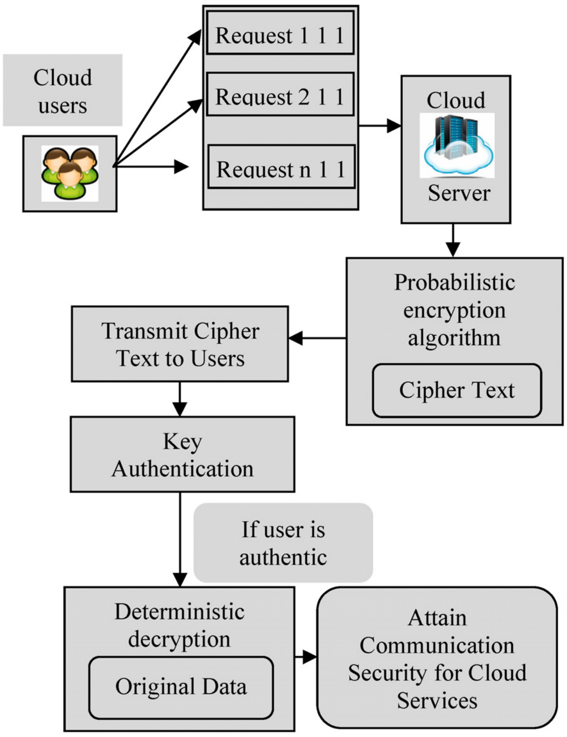

Figure 1. The architecture of AKBGC technique for Securing Communication of Cloud Services

2.1 Blum-Goldwasser Cryptography

The Blum-Goldwasser Cryptography (BGC) is an asymmetric key encryption algorithm that used in AKBGC technique to achieve secure and trustworthy communications during cloud service provisioning. The BGC employs public and private keys to encrypt and decrypt user requested data. One key is shared with everyone i.e. the public key. The other key is kept secret i.e. private key. The BGC encrypt a data with one key (public key) and decrypt a data with another key (private key). The BGC is a probabilistic, semantically secure cryptography technique with a constant-size ciphertext expansion. Here, ciphertext expansion represents the length increase of cloud data when it is encrypted. In BGC, The encryption algorithm executes an XOR-based stream cipher with aid of Blum Blum Shub (BBS) pseudo-random number generator in order to produce the keystream. In BGC, Keystream is a stream of random characters which are united with a plaintext message to generate a ciphertext. Besides, the BGC performs decryption through manipulating the final state of the BBS generator using the private key to identify the initial seed and to recreate random bits.

The BGC algorithm includes three processes such as a probabilistic key generation algorithm, a probabilistic encryption algorithm, and a deterministic decryption algorithm. The detailed process of BGC algorithm for securing communications of cloud services is described in below subsections.

2.1.1 Probabilistic Key Generation

The probabilistic key generation algorithm in BGC produces public and private for each user in a cloud computing environment. The key generator used in BGC is considered to be a probabilistic algorithm which prevents an adversary from simply running the key generator to get the decryption key for a ciphertext.

The probabilistic key generation algorithm at first picks two large prime numbers "a" and "b" such that $a \neq b$, arbitrarily and independently of each other for users in the cloud. From that, the public key is generated using below mathematical formulation,

$$P_k = ab \quad \textbf{(1)}$$

By using equation (1), public key "P_k" is created for each user in the cloud. Followed by, the private key of the user is generated using below formulation,

$$S_k = (a, b, p, q) \quad \textbf{(2)}$$

By using equation (2), private key "S_k" is constructed for each user in cloud environment. Here, $fact(a,b)$ denotes the factorization of (a,b) . For example, let us consider a "$a = 499$" and "$b = 547$. From that, the public key is "$P_k = 499 * 547 = 272953$" Then, probabilistic key generation determines the integers "$p = -57$" and "$q = 52$" satisfying "$pa + qb = 1$". Thus, the public key is "$S_k = (499, 547, -57, 52)$". The following figure shows the probabilistic key generation process of BGC.

As depicted in Figure 2, the cloud server creates a public and private key (i.e. key pairs) for each user by means of randomly picking two prime numbers "a" and "b". After completing probabilistic key generation process, key pairs are sent to users in a cloud environment. With aid of probabilistic key generation, proposed AKBGC technique preserves cloud data which is transmitted over a network from adversaries and unauthorized accesses. Thus, AKBGC technique attains higher communication security for providing user required services in a cloud environment.

2.1.2 Probabilistic Encryption Algorithm

When a user sent requests, the cloud server provides needed data services. Before sending user required services, the cloud server encrypts the data with application of Probabilistic Encryption Algorithm in BGC to enhance the data security. The deterministic encryption generates the same cipher text for given the same data and key. On the contrary to this, Probabilistic encryption produces unique cipher text for each time it is employed using same data and key. In probabilistic encryption, random values are utilized

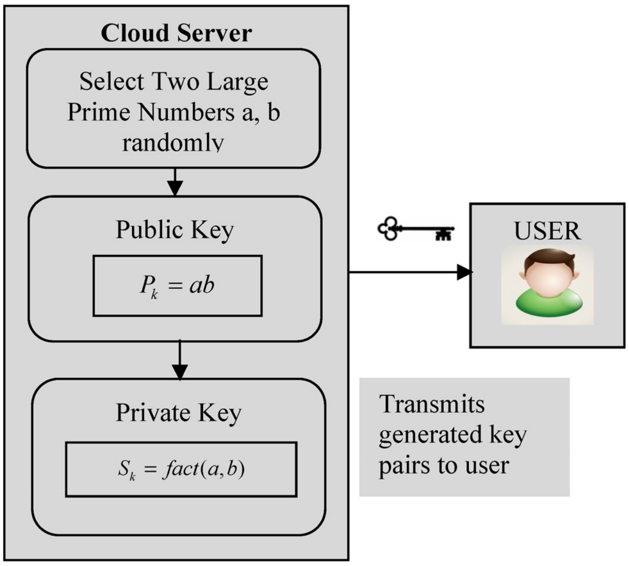
Cloud Server
Select Two Large Prime Numbers a, b randomly
Public Key
$P_k = ab$
Private Key
$S_k = fact(a,b)$
USER
Transmits generated key pairs to user

Figure 2. Probabilistic key generation process

to encrypt a user data. Thus, Probabilistic Encryption Algorithm encrypts a user data for each time by selecting a random value. Therefore, Probabilistic Encryption Algorithm returns different encrypted value (or ciphertext) if it encrypts the same data twice. This means that the ciphertext does not depend only on key and data. From that, Probabilistic encryption is a very strong encryption and semantically secure. Hence, Probabilistic encryption is applied in AKBGC technique.

The cloud server first encodes data '*d*' which is to be transmitted over a network as a string of '*T*' bits as follows,

$$d = (d_0, d_1, \ldots, d_{L-1}) \tag{3}$$

Then, cloud server choose a random number "*x*" where $1 < x < N$ and computes

$$\alpha_0 = x^2 \bmod P_k \tag{4}$$

From equation (4), P_k refers the public key of the receiver. After that, the cloud server uses the BBS pseudo-random number generator to create random bits $\vec{\beta} = \beta_0, \beta_1, \ldots, \beta_{L-1}$ (i.e. keystream). For each i to 'L', the cloud server set β_i equal to the least significant bit of α_i (i.e. $LSB(\alpha_i)$) which is computed as

$$\alpha_i = (\alpha_{i-1})^2 \bmod P_k \tag{5}$$

$$\vec{\beta}_i = LSB(\alpha_i) \tag{6}$$

Followed by, the cloud server generates the ciphertext bits using bits b_i from the BBS which is mathematically formulated as,

$$c = \vec{d} \oplus \vec{\beta} \tag{7}$$

From equation (7), the data bits are XORing with keystream (i.e. random bits) to generate ciphertext. For example, Let us consider a user data to be encrypted "$d = 9121012$" which is first converted into number of strings $(d_0, d_1, \ldots, d_{L-1})$. Where $d_1 = 1001$, $d_2 = 1100$, $d_3 = 0001$, $d_4 = 0000$, $d_5 = 1100$. Then, select a random number "$x = 399$" and computes

$$\alpha_0 = 399^2 \bmod 272953$$
$$\alpha_0 = 159201$$

For each i to 'L', the cloud server create random bits $\vec{\beta} = \beta_0, \beta_1, \ldots, \beta_{L-1}$ and construct cipher text using below,

i	$\alpha_i = (\alpha_{i-1})^2 \bmod P_k$	$\vec{\beta}_i$	$c = \vec{d} \oplus \vec{\beta}$
α_1	180539	1011	0010
α_2	193932	1100	0000
α_3	245613	1101	1100
α_4	130285	1110	1110
α_5	40632	1000	0100

During probabilistic encryption process, data is converted into a cipher text and determines

$$\alpha_6 = \alpha_5^2 \bmod 272953$$
$$\alpha_6 = 40632_5^2 \bmod 272953$$
$$\alpha_6 = 139680$$

The generated cipher text (0010, 0000, 1100, 1110, 0100 and 139680) is sent to user in cloud environment. The process involved in probabilistic data encryption is depicted in below Figure 3.

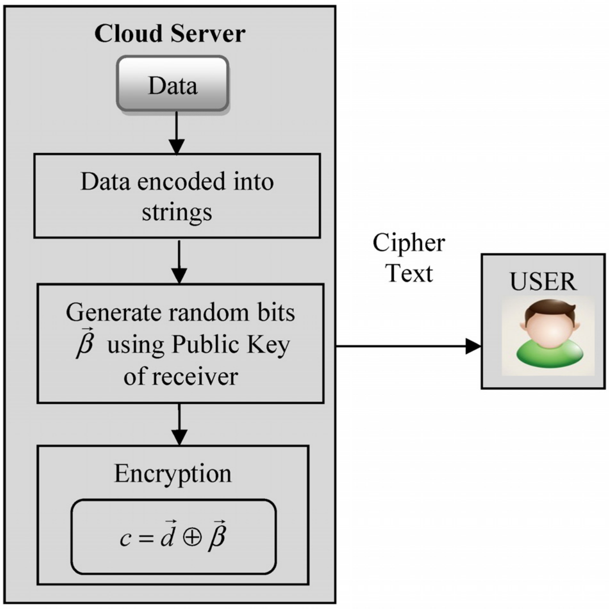
Cloud Server
Data
Data encoded into strings
Generate random bits $\vec{\beta}$ using Public Key of receiver
Encryption
$c = \vec{d} \oplus \vec{\beta}$
Cipher Text
USER

Figure 3. Probabilistic data encryption process

As demonstrated in Figure 3, Probabilistic Data Encryption process initially encodes the data to be sent into a number of string bits. Then, Probabilistic Data Encryption process produces random bits for strings of data with help of public key of the receiver. After that, Probabilistic Data Encryption XORing the data bits with the random bit to generate ciphertext. The algorithmic process of probabilistic data encryption algorithm is demonstrated in below,

Algorithm 1. Probabilistic Data Encryption

```
// Probabilistic Data Encryption Algorithm
Input: Cloud Data "$d_i = d_1, d_2, ..., d_n$", public key "$P_k$"
       of receiver
Output: Cipher Texts $c_i = c_1, c_2, ..., c_{L-1}$
Step 1: Begin
Step 2: For each cloud data $d_i$
Step 3:     Encodes data into a $L$ number of string bits
            using (3)
Step 4:     Select a random number "$x$"
Step 5:     Generate random bits $\vec{\beta}$ with aid of public
            key of receiver using (6)
Step 6:     Generate ciphertext bits using bits $\beta_i$ from
            BBS using (7)
Step 7: End for
Step 8: End
```

Algorithm 1 presents the step by step processes of probabilistic data encryption in BGC. By using above algorithmic processes, proposed AKBGC technique creates ciphertext for data which is broadcasted over a network with minimum time. In addition, the probabilistic data encryption generates constant-size ciphertext. This helps for AKBGC technique to obtain higher data security for cloud services with minimum communication overhead and space complexity.

2.1.3 Deterministic Decryption Algorithm

Whenever a user receives the ciphertext '(($c_i = c_1, c_2, ..., c_{L-1}$), α_L), an authentication process is carried out in order to verifying the identity of a user in a cloud environment with aid of a public key. During the authentication process, the public key of the sender is matched with the public key of the receiver. If both keys are matched, then deterministic decryption is allowed to get the original data. Otherwise, the deterministic decryption process is declined. The deterministic decryption algorithm produce the same output for given a particular input.

When cloud user is authentic, the user recovers original cloud data using the private key S_k of the receiver. With help of prime factorization (a, b) (i.e. private key), the cloud user evaluates the following mathematical expression,

$$x_a = ((a+1)/4)^{L+1} \bmod (a-1) \quad \textbf{(8)}$$

$$x_b = ((b+1)/4)^{L+1} \bmod (b-1) \quad \textbf{(9)}$$

$$u = a_6^{x_a} \bmod a \quad \textbf{(10)}$$

$$v = a_6^{x_b} \bmod b \quad \textbf{(11)}$$

By using equation (8), (9) and (10), (11), then the cloud user determines initial seed α_0 with help of below mathematical formula,

$$a_0 = vpa + vpb \bmod P_k \quad \textbf{(12)}$$

With aid of equation (12), cloud user re-computes the bit-vectors $\vec{\beta}$ as BBS generator used in the probabilistic encryption algorithm. After generating bit-vectors, cloud user obtains original data by XORing random bits with ciphertexts which formulated as,

$$d = \vec{c} \oplus \vec{\beta} \quad \textbf{(13)}$$

From equation (11), original data is reconstructed. For example, to decrypt the encrypted data (i.e. (0010, 0000, 1100, 1110, 0100), the receiver computes the below,

$$x_a = (499+1)/4)^{5+1}(499-1) = 463$$
$$x_b = (547+1)/4)^{5+1}(547-1) = 337$$
$$u = a_6^{x_a} \bmod a = 139680 \bmod 499 = 20$$
$$v = a_6^{x_b} \bmod b = 139680 \bmod 574 = 24$$
$$a_0 = vpa + vpb \bmod P_k$$
$$a_0 = 24(-57)499 + 20(52)574 \bmod 272953$$
$$a_0 = 159201$$

With the help of determined initial seed "a_0", then generates random bit vectors $\vec{\beta}$ and re-create original data using below,

i	$a_i = (a_{i-1})^2 \bmod P_k$	$\vec{\beta}_i$	$d = \vec{c} \oplus \vec{\beta}$
1	180539	1011	1001
2	193932	1100	1100
3	245613	1101	0001
4	130285	1110	0000
5	40632	1000	1100

Thus, $d = 9121012$ is reconstructed. The deterministic decryption process is depicted in below Figure 4.

As depicted in Figure 4, Deterministic Decryption Process initially determines initial seed with assists of the private key of the receiver. Subsequently, Deterministic Decryption Process re-computes random bits with help of BBS generator exploited in the probabilistic encryption algorithm. Afterward, Deterministic Decryption Process gets original data by XORing random bits with ciphertexts. The algorithmic process of deterministic decryption in BGC is depicted in below.

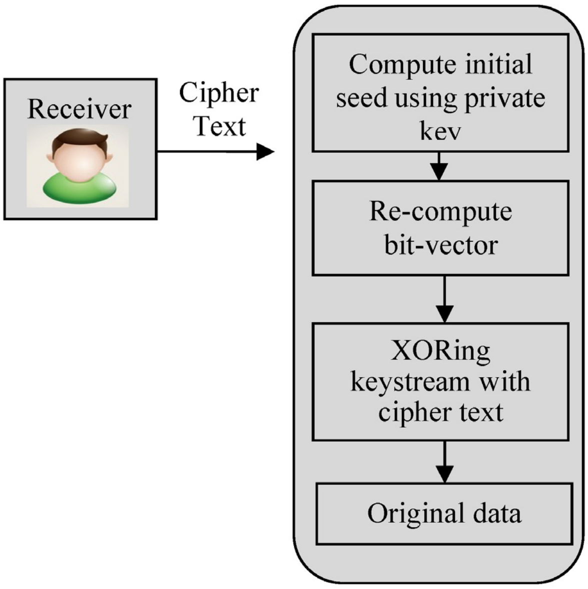
Receiver
Cipher Text
Compute initial seed using private kev
Re-compute bit-vector
XORing keystream with cipher text
Original data

Figure 4. Deterministic decryption process

Algorithm 2. Deterministic Decryption

```
// Deterministic Decryption Algorithm
Input: Cipher Text $c_i = c_1, c_2, ..., c_{L-1}$, Private key $S_k$ of receiver
Output: Obtain original Data ‘$d_i = d_1, d_2, ..., d_{L-1}$’
Step 1: Begin
Step 2: For each received Cipher Text $c_i$
Step 3:     Authentication performed by comparing the public key of the sender with receiver
Step 4:     If both public keys are matched
Step 5:         The deterministic decryption is allowed
Step 6:     Else
Step 7:         The decryption process is declined
Step 8:     Endif
Step 9:     Compute initial seed “$\alpha_0$” with aid of private key of receiver using (12)
Step 10:    Re-compute bit-vector ‘$\vec{\beta}$’
Step 11:    Reconstruct original data ‘$d$’ using (13)
Step 12: End for
Step 13: End
```

Algorithm 2 portrays the step by step process of deterministic data decryption. With aid of above algorithmic process, AKBGC technique allows the users to obtain original data whenever she or he is authentic. This supports for AKBGC technique to prevent data from an adversary or illegal access in a cloud environment. As a result, AKBGC technique enhances data confidentiality and throughput of cloud services.

3 Experimental Settings

The proposed AKBGC technique is implemented in java languages with help of Amazon Access Samples DataSet to evaluate performance. The Amazon Access Samples Data Set [25] is obtained from the UCI machine learning repository to perform experimental evaluations. The Amazon Access Samples DataSet is a sparse data set includes of users and their assigned access. The Amazon Access Samples Data Set comprises the following attributes such as Action, Target_Name, Login, Request_Date, and Authorization_ Date. The performance of AKBGC technique is measured in terms of data confidentiality, communication overhead, space complexity, and throughput. The result of proposed AKBGC technique is compared against with existing Inter-Cloud Virtual Private Network (ICVPN) [1] and Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [2].

4 Results and Discussions

In this section, the performance result of AKBGC technique is discussed. The efficiency of AKBGC Technique is compared against with existing Inter-Cloud Virtual Private Network (ICVPN) [1] and Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [2] respectively. The effectiveness of AKBGC technique is evaluated along with the following metrics with the assist of tables and graphs.

4.1 Performance Result of Data Confidentiality

In AKBGC technique, data confidentiality evaluates the ability of AKBGC technique to secure data services and accessed only by authentic users in a cloud environment. The data confidentiality is measured in terms of percentages (%) and expressed as

$$DC = \frac{data\ correctly\ accessed\ by\ authentic\ users}{Total\ Number\ of\ Cloud\ Data} * 100 \quad \textbf{(14)}$$

From equation (12), data confidentiality ‘*DC*’ of cloud services is determined with respect to a varied number of cloud data. When the confidentiality of data service is higher, the method is said to be more efficient. To evaluate the performance of data confidentiality when providing cloud services, the proposed AKBGC technique considers the framework with a different number of cloud data and users for conducting experimental process. The comparative result analysis of data confidentiality versus a various number of cloud data in the range of 15-150 using three methods namely ICVPN [1], CP-ABE [2] and proposed AKBGC is shown in Figure 5. When considering 75 cloud data for providing services to users in a cloud environment, proposed AKBGC technique obtains 89% data confidentiality whereas existing ICVPN [1], CP-ABE [2] gets 79% and 85% respectively. From these results, it is expressive that the data confidentiality using proposed AKBGC technique is higher than an existing [1-2] works.

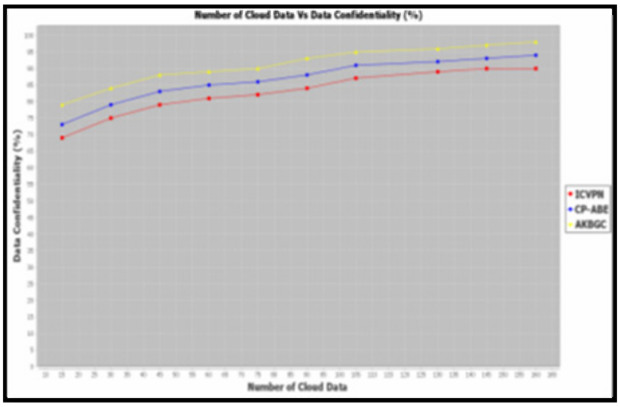

Figure 5. Impact of data confidentiality versus number of cloud data

Figure 5 presents the impact of data confidentiality during cloud services provisioning based on a different number of cloud data in the range of 15-150 using three methods namely ICVPN [1], CP-ABE [2] and proposed AKBGC. As demonstrated in the figure, proposed AKBGC technique provides higher data confidentiality for providing user required services to corresponding users in a cloud environment. As well while increasing the number of cloud data, the data confidentiality of cloud services is also increased using three techniques.

But comparatively, data confidentiality using proposed AKBGC technique is higher than other conventional works. This is because of application of Blum-Goldwasser Cryptography (BGC) in AKBGC technique where it performs probabilistic encryption and deterministic decryption to attain secure data communication. With help of the algorithmic process of BGC, AKBGC technique encrypts the user required data services using the public key of receiver and then sent the ciphertext to users in the cloud. After receiving the ciphertext, authentication is performed through matching the public key of the sender with the public key of receiver. When the user is authentic, the original cloud data is obtained. Otherwise, the decryption process is discarded. From that, only authentic users can access the encrypted data. Further, BGC is semantically secure based on the process of integer factorization during probabilistic key generation process in a cloud environment on the contrary to conventional cryptography techniques.

This helps for AKBGC technique to attain higher data confidentiality for cloud services provisioning. Therefore, proposed AKBGC technique enhances data confidentiality of cloud services by 10% as compared to ICVPN [1] and 5% as compared to CP-ABE [2] respectively.

4.2 Performance Result of Communication Overhead

In AKBGC technique, Communication Overhead measures the amount of time needed for securing cloud data services. The communication overhead is evaluated in terms of milliseconds (ms) and formulated as

$$CO = n * time\,(secured\ data\ service) \qquad \textbf{(15)}$$

From equation (13), communication overhead '*CO*' of cloud data services is determined with respect to a different number of cloud data (*n*). When the communication overhead is lower, the method is said to be more efficient. The proposed AKBGC technique assumes a diverse number of cloud data for carried outing experimental work to evaluate the performance of communication overhead during cloud services provisioning. The performance results analysis of communication overhead based on a different number of cloud data in the range of 15-150 using three methods namely ICVPN [1], CP-ABE [2] and proposed AKBGC is depicted in below Figure 6. While considering 90 cloud data for service provisioning in the cloud environment, proposed AKBGC technique acquires 28 ms communication overhead whereas existing ICVPN [1], CP-ABE [2] obtains 44 ms and 37 ms respectively. Accordingly, it is clear that the communication overhead of cloud data services using proposed AKBGC technique is lower than an existing [1-2] works.

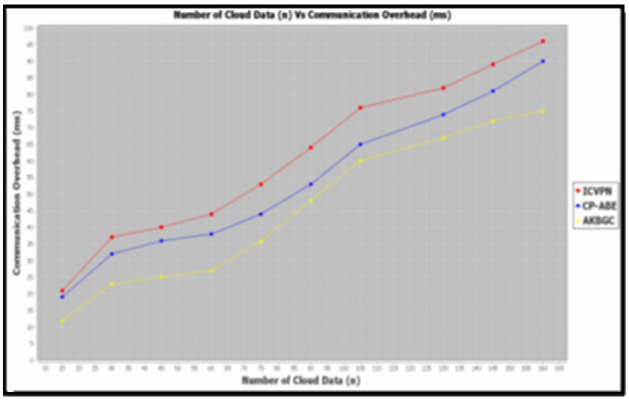

Figure 6. Impact of communication overhead versus number of cloud data

Figure 6 illustrates the impact of communication overhead for cloud services provisioning versus a various number of cloud data in the range of 15-150 using three methods namely ICVPN [1], CP-ABE [2] and proposed AKBGC. As exposed in the figure, proposed AKBGC technique provides lower communication overhead for presenting user needed services to corresponding users in a cloud environment. In addition, while increasing the number of cloud data, the communication overhead of cloud services is also increased using three techniques. But comparatively, communication overhead using proposed AKBGC technique is lower than other existing works. This is owing to application of Blum–Goldwasser Cryptography (BGC) in AKBGC technique.

With the support of BGC process, proposed AKBGC technique securely performs communications among the users in a cloud environment through probabilistic encryption and deterministic decryption. The BGC algorithm takes a minimum amount of time for probabilistic encryption and deterministic decryption process as compared existing ICVPN [1], CP-ABE [2]. Hence, the communication overhead using proposed AKBGC technique is lesser than other existing works. Besides, BGC is comparatively efficient in terms of computation for providing secured data cloud services than conventional RSA. As a result, proposed AKBGC technique lessens the communication overhead of cloud services by 29% as compared to ICVPN [1] and 20% as compared to CP-ABE [2] respectively.

4.3 Performance Result of Space Complexity

In AKBGC technique, space complexity computes the amount of memory space required for storing encrypted cloud data. The space complexity is evaluated in terms of Kilobytes (KB) and represented as,

$$SC = n * memory\ (storing\ encrypted\ cloud\ data) \quad \textbf{(16)}$$

From equation (14), the space complexity *SC* of cloud data services is evaluated with respect to a different number of cloud data (*n*). When space complexity is lower, the method is said to be more effective. The comparative result analysis of space complexity of secured cloud data services versus a varied number of cloud data in the range of 15-150 using three methods namely ICVPN [1], CP-ABE [2] and proposed AKBGC are demonstrated in below Figure 7. When taking 105 cloud data to offer cloud services to users, proposed AKBGC technique consumes 28 KB space complexity for storing encrypted cloud data whereas existing ICVPN [1], CP-ABE [2] acquires 37 KB and 29 KB respectively. From that, it is descriptive that the space complexity of secured cloud data services using proposed AKBGC technique is lower than an existing [1-2] works.

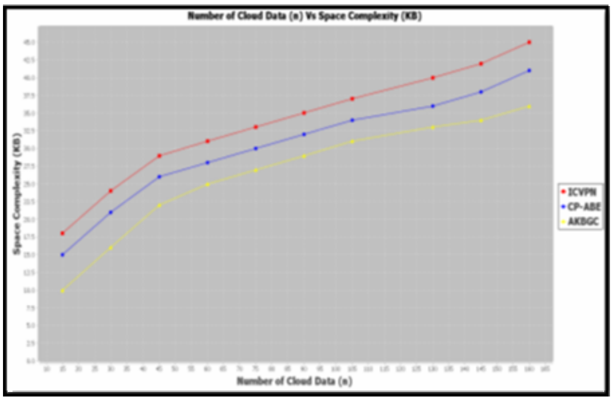

Figure 7. Impact of space complexity versus number of cloud data

Figure 7 describes the impact of space complexity during secured cloud services communications versus a different number of cloud data in the range of 15-150 using three methods namely ICVPN [1], CP-ABE [2] and proposed AKBGC. As depicted in the figure, the proposed AKBGC technique provides minimum space complexity for securely offering user-desired services to users in a cloud environment. Besides while increasing the number of cloud data, the space complexity of cloud services is also increased using three techniques. But comparatively, space complexity using proposed AKBGC technique is lower than other existing works. This is due to the application of Blum–Goldwasser Cryptography in AKBGC technique.

The BGC algorithm consumes a minimum amount of memory space to store encrypted data on the contrary to existing ICVPN [1], CP-ABE [2]. Therefore, the space complexity of cloud services using the proposed AKBGC technique is lesser than other existing works. Further, BGC is relatively efficient in terms of storage because it has a constant-size ciphertext expansion regardless of message length on the contrary to conventional cryptography techniques designed for secure communication of cloud services. Therefore, proposed AKBGC technique minimizes the space complexity of secured cloud services provisioning by 23% as compared to ICVPN [1] and 14% as compared to CP-ABE [2] respectively.

4.4 Performance Result of Throughput

In AKBGC technique, Throughput (*T*) determines the average rate of successful data delivery in a given specified amount of time. The throughput is measured in terms of kbps (kilobit per second) and obtained using below formulation,

$$T = \frac{average\ rate\ of\ successful\ data\ delivery}{Time} \quad \textbf{(17)}$$

From equation (15), the throughput of cloud data services is estimated with respect to a dissimilar number of cloud data (*n*). When the throughput is higher, the method is said to be more effective. In order to examine the average rate of successful data delivery when providing cloud services, the proposed AKBGC technique considers the framework with the varied size of cloud data and for accomplishing experimental works. When considering 80 MB cloud data for experimental evaluation, proposed AKBGC technique achieves 198 kbps throughput whereas existing ICVPN [1], CP-ABE [2] obtains 148 kbps and 165 kbps respectively. Thus, it is illustrative that the throughput using proposed AKBGC technique is higher than existing [1-2] works.

The experimental result analysis of throughput with respect to the dissimilar size of cloud data in the range of 10-100 using three methods namely ICVPN [1], CP-ABE [2] and proposed AKBGC are presented in below Figure 8.

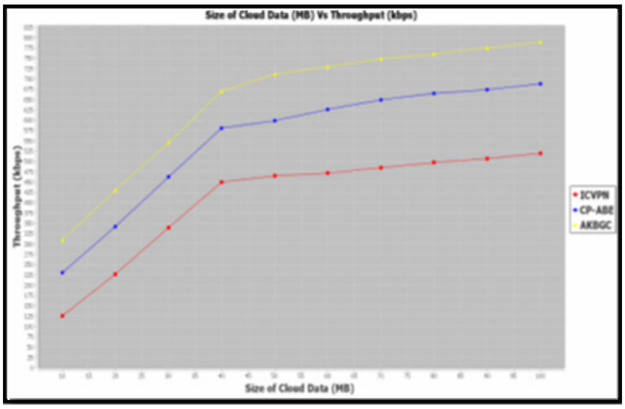

Figure 8. Impact of throughput versus different size of cloud data

Figure 8 portrays the impact of throughput rate for cloud services provisioning versus various sizes of cloud data in the range of 10 MB-100 MB using three methods namely ICVPN [1], CP-ABE [2] and proposed AKBGC. As illustrated in the figure, the proposed AKBGC technique provides higher throughput rate for offering user necessitated services to corresponding users in a cloud environment. Moreover while increasing the number of cloud data, the throughput of cloud services is also increased using three techniques. But comparatively, throughput using proposed AKBGC technique is higher than other existing works. This is because of the process of Blum–Goldwasser Cryptography (BGC) in AKBGC technique.

The probabilistic encryption and deterministic decryption process in BGC assists for proposed AKBGC technique to securely transmit the user needed data services to users in the cloud without any information loss. The BGC helps for proposed AKBGC technique to generate different ciphertexts each time it is encrypted for a given cloud data. Therefore, AKBGC technique safeguards data which is sent over a network from adversaries in a cloud environment. This in turn helps for proposed AKBGC technique to successful delivery the requested data services to users without any loss. Hence, proposed AKBGC technique increases the throughput of cloud services by 59% as compared to ICVPN [1] and 19% as compared to CP-ABE [2] respectively.

5 Related Works

A novel technique was designed in [11] to secure communication among end nodes with aid of biocyber metrics and providing flexible cloud services. The execution time taken for secure communication using this technique was more. The Shamir's secret sharing algorithm was used in [12] for achieving secure communication and increasing the privacy of data in cloud computing. The Shamir's secret sharing algorithm requires more computation time for improving the confidentiality of data.

An Owner-Controlled Cloud Data Sharing was presented in [13] to enhance security and privacy preservation for cloud data services. The space complexity of cloud data sharing was not addressed. A public-key cryptosystem was intended in [14] that generate constant-size ciphertexts to securely distribute data with others in cloud storage. The public-key cryptosystems does not attain higher communication security.

Server-aided anonymous attribute-based authentication was designed in [15] for protecting the privacy of users' data with minimum computational cost. The confidentiality rate of data using attribute-based authentication was lower. A Secure Authentication System was intended in [16] for hybrid cloud service in mobile communication environments to present better performance in terms of confidentiality, integrity, availability.

A secure re-encryption algorithm was presented in [17] with aid of EIGamal algorithm to address security issues in cloud data services. The authentication performance of the EIGamal algorithm was not efficient. A key policy attribute-based encryption method was introduced in [18] for providing privacy and access control during cloud storage services with minimum computational cost. This key policy attribute-based encryption method does not provide higher security services.

A lightweight access control solution was designed in [19] to increase end-user privacy in a cloud environment. However, it takes the more computational cost. A secure energy-efficient and quality-of-service architecture (EEQoSA) was developed in [20] to solve security issues and to support diverse media services in a mobile cloud environment. The EEQoSA provides higher secure communication. But, the amount of time required for secure cloud service provisioning was very higher.

A novel method was presented in [21] that provide solutions for gathering and managing sensors' data in a smart building could. A grey wolf optimization based clustering algorithm was employed in [22] for quality communication and reliable delivery of data. A context-aware framework was introduced in [23] for intelligent power equipment management. A novel technique was developed in [24] for better utilization of resources.

6 Conclusion

The AKBGC technique is developed with the goal of securing communication during cloud services provisioning with minimum overhead. The goal of AKBGC technique is attained with the application of BGC. The designed AKBGC technique allows only authentic users to obtain the original cloud data. Thus, AKBGC attain higher data confidentiality and security than state-of-the-art works. Besides with processes of probabilistic encryption and deterministic decryption, AKBGC reduce the amount of time needed for secured communication and also it enhances the throughput of cloud services when compared to state-of-the-art works. Furthermore with the constant-size ciphertext generation, AKBGC technique minimizes the space complexity involved during secured cloud service provisioning as state-of-the-art works. The efficacy of AKBGC technique is tested with metrics such as data confidentiality, communication overhead, space complexity, and throughput. The experimental results reveal that AKBGC technique provides better performance with an enhancement of data confidentiality and the reduction of communication overhead when compared to the state-of-the-art works. The future work of AKBGC technique can be proceed with different hashing and tree based data structure to increases the security of cloud data storage.

References

[1] A. Sajjad, M. Rajarajan, A. Zisman, T. Dimitrakos, A Scalable and Dynamic Application-level Secure Communication Framework for Inter-cloud Services, *Elsevier Future Generation Computer Systems, Elsevier*, Vol. 48, pp. 19-27, July, 2015.

[2] A. Alrawais, A. Alhothaily, C. Hu, X. Xing, X. Cheng, An Attribute-Based Encryption Scheme to Secure Fog Communications, *IEEE Access*, Vol. 5, pp. 9131- 9138, May, 2017.

[3] S. Raza, T. Helgason, P. Papadimitratos, T. Voigt, Securesense: End-To-End Secure Communication Architectureforthe Cloud-Connected Internet of Things, *Future Generation Computer Systems, Elsevier*, Vol. 77, pp. 40-51, December, 2017.

[4] S. Zahra, M. M. Alam, Q. Javaid, A. Wahid, N. Javaid, S. Ur R. Malik, K. Khan, Fog Computing Over IoT: A Secure Deployment and Formal Verification, *IEEE Access*, Vol. 5, pp. 27132-27144, November, 2017.

[5] M. A. AlZain, B. S. Pardede, A New Model to Ensure Security in Cloud Computing Services, *Journal of Service Science Research, Springer*, Vol. 4, No. 1, pp. 49-70, June, 2012.

[6] S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, N. Kumar, A. V. Vasilakos, On the Design of Provably Secure Lightweight Remote User Authentication Scheme for Mobile Cloud Computing Services, *IEEE Access*, Vol. 5, pp. 25808-25825, October, 2017.

[7] D. Chandramohan, T. Vengattaraman, P. Dhavachelvan, A Secure Data Privacy PReservation for On-demand Cloud Service, *Journal of King Saud University - Engineering Science, Elsevier*, Vol. 29, No. 2, pp. 144-150, April, 2017.

[8] Z. Wen, J. Cała, P. Watson, A. Romanovsky, Cost-Effective, Reliable and Secure Workflow Deployment over Federated Clouds, *IEEE Transactions on Services Computing*, Vol. 10, No. 6, pp. 929-941, December, 2017.

[9] J. Zhou, Z. Cao, X. Dong, X. Lin, Security and Privacy in Cloud-assisted Wireless Wearable Communications: Challenges, Solutions, and Future Directions, *IEEE Wireless Communications*, Vol. 22, No. 2, pp. 136-144, April, 2015.

[10] J.-L. Tsai, N.-W. Lo, A Privacy-Aware Authentication Scheme for Distributed Mobile Cloud Computing Services, *IEEE Systems Journal*, Vol. 9, No. 3, pp. 805-815, May, 2015.

[11] J. Pacheco, C. Tunc, P. Satam. S. Hariri, Secure and Resilient Cloud Services for Enhanced Living Environments, *IEEE Cloud Computing*, Vol. 3, No. 6, pp. 44-52, December 2016.

[12] N. Aggarwal, A. Choudhary, M. Bachani, R. Jain, Framework For Secure Cloud Data Communication, *International Journal of Scientific and Technology Research*, Vol. 4, No. 2, pp. 281-284, February, 2015.

[13] J. Tang, Y. Cui, Q. Li, K. Ren, J. Liu. R. Buyya, Ensuring Security and Privacy Preservation for Cloud Data Services, *ACM Computing Surveys*, Vol. 49, No. 1, pp. 1-39, July, 2016.

[14] C.-K. Chu, S. S.-M. Chow, W.-G. Tzeng, J. Zhou, R. H. Deng, Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 25, No. 2, pp. 468-477, February, 2014.

[15] Z. Liu, H. Yan, Z. Li, Server- Aided Anonymous Attribute-based Authentication in Cloud Computing, *Future Generation Computer Systems, Elsevier*, Vol. 52, pp. 61-66, November, 2015.

[16] J.-M. Kim, J.-K. Moon, Secure Authentication System for Hybrid Cloud Service in Mobile Communication Environments, Hindawi Publishing Corporation, *International Journal of Distributed Sensor Networks*, Vol. 2014, pp. 1-7, February, 2014.

[17] L. Xiong, Z. Xu. Y. Xu, A Secure Re-encryption Scheme for Data Services in a Cloud Computing environment, *Concurrency and Computation: Practice and Experience, Wiley Online Library*, Vol. 27, No. 17, pp. 4573-4585, December, 2015.

[18] E. Tameem, G. Cho, Providing Privacy and Access Control in Cloud Storage Services Using a KPABE System with Secret Attributes, *Arabian Journal for Science and Engineering*, Vol. 39, No. 11, pp. 7877-7884, November, 2014.

[19] N. Fotiou, A. Machas, G. C. Polyzos, G. Xylomenos, Access Control as a Service for the Cloud, *Journal of Internet Services and Applications, Springer*, Vol. 6, No. 11, pp. 1-15, November, 2015.

[20] Q. B. Hani, J. P. Dichter, Energy-efficient Service-oriented Architecture for Mobile Cloud Handover, *Journal of Cloud Computing, Springer*, Vol. 6, No. 1, pp. 1-13, January, 2017.

[21] A. P. Plageras, K. E. Psannis, B. B. Gupta, C. Stergiou, H. Wang, Efficient IoT-based Sensor BIG Data Collection-processing and Analysis in Smart Buildings, *Future Generation Computer Systems, Elsevier*, Vol. 82, pp. 349-357, May, 2018.

[22] M. Fahad, F. Aadil, Zahoor-ur- Rehman, S. Khan, P. A. Shah, K. Muhammad, J. Lloret, H. Wang, J. W., Lee, I. Mehmood, Grey Wolf Optimization Based Clustering Algorithm for Vehicular Ad-hoc Networks, *Computers and Electrical*

Engineering, Elsevier, Vol. 70, pp. 853-870, August, 2018.

[23] C. Choi, C. Esposito, H. Wang, Z. Liu, J. Choi, Intelligent Power Equipment Management Based on Distributed Context-Aware Inference in Smart Cities, *IEEE Communications Magazine*, Vol. 56, No. 7, July, 2018.

[24] W. Huang, P. Wang, L. Lv, L. Wang, An Inventive High-performance Computing Electronic Information System for Professional Postgraduate Training, *International Journal of Computers and Applications*, p. 1, January, 2018.

[25] *Amazon Access Samples Dataset*, https://archive.ics.uci.edu/ml/datasets/Amazon+ Access+Samples.

Biographies

R. Senthilkumar received the Bachelor Degree (B.E), Masters degree (M.E) in Computer Science and Engineering and Currently Pursuing the Ph.D. degree from the Anna University, Tamilnadu, India.His research interests are in Cloud Computing and Secure the Personal Communication. He has published papers in journals and conference proceedings and Member of CSI and IAENG.

B. G. Geetha received the B.E, M.E, Ph.D degree. She is in the editorial board member of International Journals and published many papers. She has acted as jury, speaker at international conferences Spain, Malaysia. Her areas of research include Software Engineering, data mining, Cloud Computing, wireless networks and Big data analytics.