RESEARCH ARTICLE OPEN ACCESS

Bilinear Hashing With Conditional Attributes For Secured Data Storage in Cloud

M.Buvaneswari [1], Dr.N.Rajendran [2]
Research Scholar [1], Bharathiar University, Coimbatore,
Principal [2], Vivekanandha Arts and Science College for Women, Sankari,
Tamilnadu - India

ABSTRACT
Cloud data security is concern for the client when employing the cloud services supplied by the service provider in cloud environment. Recently, many research works has been developed for securing data in cloud environment. However, cloud data storage and accessing of data still causes security issues. In order to overcome such limitation, Bilinear Hashing based Secured Data Storage (BH-SDS) mechanism is introduced in this paper to develop a secured cloud data storage and fast accessing of the data communications by cloud users. Initially, the request is sent from the clients and conditional attributes are needed to be evaluated for securing the clients data requirement. The conditional attributes are evaluated through client requests. Next, the conditional attributes are encrypted with data by using bilinear mapping transformation function to improve the cloud data security. Then, BH-SDS mechanism is used hash function for mapping client required data into the corresponding users in cloud environment which results in fast accessing of data communications by cloud users. Experimental evaluation of BH-SDS mechanism is done with the performance metrics such as data storage capacity, accessing speed and cloud data security. Experimental analysis shows that the BH-SDS mechanism is able to improve the cloud data security by 8% and to improve the data accessing speed by 22% when compared to the state-of-the-art works.

Keywords:- cloud data storage, bilinear operator, conditional attributes, Hash function, security, bilinear mapping transformation function

I. INTRODUCTION

Cloud computing is a promising technology in the field of information technology. Cloud Computing is the employ of computing resources (hardware and software) which are delivered as a service over an internet. More or less Cloud computing portrays well scalable computing resources supplied as an outer service by internet on pay-as-usability basis.

Currently, numerous institutes outsource their large-scale data storage to the cloud for saving the cost in preserving in-house storage. With cloud storage service, the members of an institute can share data with other members by means of uploading their data to the cloud. The instances of institutes which may profit from this cloud storage and sharing service are various like international enterprises with many employees around the world, collaborative web application providers with a large user base or institutions dealing with big data, healthcare service providers coordinating medical data from doctors, researchers, patients, etc

In Cloud computing, Data owners store the data in the cloud storage and cloud users can access the data in the cloud storage. For accessing the data in the Cloud storage, Cloud users require necessary security for protecting the data. At present, most of researches have been developed for cloud data security. For example, Cooperative Provable Data Possession (CPDP) scheme was developed in [1] depends on homomorphic verifiable response and hash index hierarchy to support dynamic scalability on multiple storage servers in cloud environment. However, CPDP scheme is employed for a large file is affected by using bilinear mapping operations because of its large complexity.

A data integrity checking algorithm was designed in [2] to reduce the third party auditing and to protect static and dynamic data from unauthorized observation, modification, or interference. Data integrity checking algorithm provides better protection in terms of filtering, risk management and deployment of standard information security policies. However, all risks are not reduced via moving operations to a cloud environment. Besides, hash function does not provide the appropriate balance between protection and usability need to be discarded in cloud system.

Privacy Preserving Model was presented in [3] to Prevent Digital Data Loss in the Cloud which in turn assists to cloud requester/users to hope their proprietary information and data stored in the cloud environment. Public auditing protocol was introduced in [4] to provide security cloud storage service to users and to verify data integrity. The Public auditing protocol is secure against the curiosity attack, loss attack and tamper attack. Though, the public auditing model for cloud storage and the model against the pollution attacks for linear network coding should not be considered.

Inconsistency-Tolerant Integrity Checking (ITIC) method was designed in [5] to allow the updates to be fruitfully checked for integrity preservation even in the presence of inconsistency. But, investigation of the interplay between the notions of inconsistency-tolerant repair is not included. It should not be sufficient to be content with partial repairs that tolerate inconsistencies with the query. A dynamic remote attestation framework [6] was developed to evaluate a

target system depends on an information flow-based integrity model. However, dynamic risk evaluation does not explain the tolerable risk level and relevant system properties. In addition, this model is not flexible. Dual-Server Public Key Encryption with Keyword Search (DSPEKS) was presented in [7] which can prevent the inside keyword guessing attack and employ Smooth Projective Hash Function (SPHF) to construct a generic DSPEKS scheme. An effective and secure access control scheme was introduced in [8] to address the security problem in cloud environment. Though, the efficiency of encryption and decryption process is not effective.

A ciphertext-policy attribute-based encryption (ABE) scheme and a proxy re-encryption scheme were designed in [9] to efficiently ad-dress the issues of user revocation and to enable the data owner to dele-gate file re-encryption to cloud servers and delegate user secret key update. A probabilistic challenge–response scheme was developed in [10] to prove that users' files are available and stored in a specified cloud server and to efficiently detect this malicious behavior of cloud servers. But computation and communication overhead is poor in this scheme.

II. RELATED WORKS

In cloud computing, security is essential for protecting data in the cloud storage. Recently, most of the research works has been designed for secured data storage. For instance, a novel public auditing scheme called as MuR-DPA was presented in [11] that integrated a novel authenticated data structure based on the Merkle hash tree. The MuR-DPA scheme can assists fully dynamic data updates, authentication of block indices and efficient verification of up-dates for multiple replicas simultaneously. Though, the MuR-DPA scheme presents improved security against dishonest cloud service providers. However, the secure public auditing of dynamic data and streaming data with constant-sized integrity proofs is remained unsolved.

The remote data auditing scheme was developed in [12] based on algebraic signature properties to verify the integrity of the data stored in cloud computing and to reduce the computational overhead on the client and server side of the cloud. Data Access Control for Multi-Authority Cloud Storage was designed in [13] for effective and secure data access control scheme with efficient decryption and revocation and to improve the cloud data storage. A privacy-preserving and auditing-supporting outsourcing data storage scheme was introduced in [14] by using encryption and digital watermarking. This scheme combines digital watermark technology with encryption methods for outsourcing data storage.

A Hybrid Cloud Security Algorithm (HCSA) was developed in [15] to solve the problems related to secure cloud data storage system modeling based on the grouping of Elliptic Curve based Schnorr (EC-Schnorr) scheme and blooming filter. A public auditing scheme for cloud storage systems was designed in [16] where deduplication of encrypted data and data integrity checking was attained within the same framework. A Symmetric Encryption Algorithm (SEA) was introduced in [17] to provide improved security to the data stored in the cloud storage.

An effective and flexible distribution verification mechanism was introduced in [18] to solve data storage security in cloud computing and to guarantee the correctness of users 'data in cloud data storage. Cryptographic Role-Based Access Control scheme was designed in [19] to protect data privacy and to enhance the system security in a cloud storage system. An efficient encryption technique was developed in [20] for secure access and storage of data on public cloud server.

Based on the aforementioned methods and techniques presented, in this paper, a Bilinear Hashing based Secured Data Storage (BH-SDS) mechanism is proposed. The main goal of BH-SDS mechanism is to improve the cloud data security and to provide fast accessing of data in cloud environment. Initially, BH-SDS mechanism is evaluates the conditional attributes for securing data requirement. Then, BH-SDS mechanism is employed bilinear mapping transformation function for encrypting the conditional attributes with data with objective of improving the cloud data security. Finally, hash function is used to map the data in consequent users in cloud environment. The contributions of BH-SDS mechanism include the following:

- For securing the data requirement, conditional attributes are evaluated.
- To improve the cloud data security, bilinear mapping transformation function is used in BH-SDS mechanism.
- To improve the fast accessing of data communication, Hash function is employed.

The rest of the paper organized as follows. In Section 2, a summary of different cloud data storage mechanisms are explained. In Section 3, the proposed BH-SDS mechanism is described with the help of neat architecture diagram. In Section 4, simulation environment is provided with detailed analysis of results explained in Section 5. In Section 6, the concluding remarks are included.

III. BILINEAR HASHING BASED SECURED DATA STORAGE MECHANISM

In this work, a Bilinear Hashing based Secured Data Storage (BH-SDS) mechanism is introduced to improve the data storage security and to improve the data accessing speed in cloud. Initially, BH-SDS mechanism is evaluates the conditional attributes based on client request in cloud. In BH-SDS mechanism, the conditional attributes are evaluated for securing the client data requirements in cloud. The client data requirement is nothing but user requested data to the cloud database. The conditional attributes contains set of conditions for providing clients required data to users in cloud environment. Then, the determined conditional attributes are encrypted with data with the help of bilinear mapping transformation function for improving the cloud data security. Finally, BH-SDS mechanism is used hash function to map the

client required data into the corresponding users in cloud environment which in turn improves the security of data in an effective manner. Besides, with the help of mapping data into their consequent user, BH-SDS mechanism is significantly improves the data accessing speed. The architecture diagram of Bilinear Hashing based Secured Data Storage mechanism is shown in below Fig. 1.

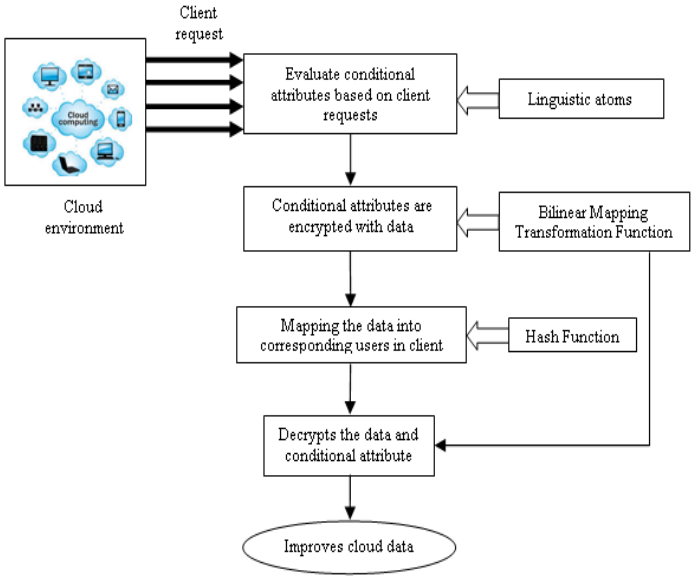

Fig. 1 Architecture diagram of Bilinear Hashing based Secured Data Storage Mechanism

As shown in Figure 1, initially BH-SDS mechanism is evaluates conditional attributes while the request is sent from the clients with the help of Linguistic atoms. Next, BH-SDS mechanism is used Bilinear Mapping Transformation Function to encrypt the conditional attributes with data with the intension of improving the cloud data security. After that, BH-SDS mechanism is used hash function for mapping the client required data into their appropriate users in cloud which results in improved data storage security and fast accessing of data by the cloud users in a significant manner.

A. *Conditional Attributes*

Initially, BH-SDS mechanism is evaluates conditional attributes based on client requests from the client environment to assign some conditions for securing client data requirement. In BH-SDS mechanism, the conditional attribute requested form the client is estimated by using the linguistic atomic variable terms. Each possible value of the linguistic atoms (i.e., attributes) generates the fuzzy set based result. The conditional attribute evaluation using linguistic atoms in BH-SDS mechanism is illustrated as,

$$Linguistic\ Atom\ based\ \textbf{conditional attribute}\ evaluation = \{\text{Attribute}, Trans(\text{Attribute}), D\} \rightarrow S \quad \ldots\ldots\ldots (1)$$

In BH-SDS mechanism, linguistic atoms initially chooses the attributes '$Attribute$' and request to the attributes to be transacted '$Trans\ (Attribute)$' on the total data storage cloud 'D'. Here, 'S' is the syntactic generator to the request query from the client side. Let us consider the linguistic transaction '$Trans\ (Attribute)$' attributes to be encrypted. Then, the client starts the request '$Trans\ (Attribute)$' of data requirement through the cloud applications. The data requirement request made by the client with conditional attributes is need to be encrypted for providing the higher security data transmission in cloud environment. The BH-SDS mechanism is used bilinear mapping transformation function for encrypting conditional attribute with data which is detailed described in following subsection.

B. *Bilinear Mapping Transformation Function for Encryption Process*

Next, BH-SDS mechanism is used Bilinear Mapping Transformation Function to encrypt the conditional attribute with data. In proposed BH-SDS mechanism, data is encoded onto a complex plane by using a bilinear mapping transformation function. Then the data points are transformed into another set of points in another complex plane then generates the cipher text (i.e. encrypted data). The transformation being reversible the original data can be obtained using inverse transformation. At first, client and server acknowledge the request message for data transmission. In BH-SDS mechanism, Bilinear mapping transformation preserves unrelated attributes from the clients to perform secured data transmission in cloud environment. Let us consider two conditional attributes like 'CA_1' and 'CA_2', then the condition attribute is divided and obtained using the following,

$$= CA_1 + CA_2 \quad \ldots (2)$$

With the aid of equation (2), conditional attribute divide, the cipher specific id is produced by using following equation,

$$Cipher\ Specific\ Id\ Generation\ (G) = \frac{CA_1 S + C_1}{CA_1 S + C_2} \quad \ldots\ldots (3)$$

From the (3), 'C_1' And 'C_2' are the constants used while generating the Id and 'S' is the syntactic combined with conditional attributes 'CA_1' and 'CA_2'. The conditional attribute encryption with cipher text, specific id and key generation are detailed explained by using the following algorithmic steps,

// Bilinear Transformation Encryption Algorithm Using Conditional Attributes

Input: plain Text

Output: Encrypted data

Begin

Step 1: divide the plain text into number of blocks and then change into binary form

Step 2: evaluate the conditional attribute splitting from by using (2)

Step 3: attains the cipher specific id from by using (3)

Step 3.1: Cipher Specific id produced with pseudo random numbers employed for the conversion conditional attribute request into cipher text.

Step 3.2: Specific key id generation based on cipher text using setKey () function

Step 4: Converted Cipher text and Specific id is sent to the Cloud Data Storage Server

Step 5: Repeat Step 1 to 4 until entire text is converted

End

Fig. 2 Algorithmic Process of Bilinear Transformation Encryption Algorithm

As shown in Fig. 2, Bilinear Transformation Encryption Algorithm is initially takes plain text as input. The given plain text is divided into number of blocks and then transformed into the binary form. The Bilinear Transformation Encryption Algorithm divides each condition and fetches the accurate information extraction for data transmission in cloud. Then, this algorithm evaluates the conditional attributes 'CA_1,'CA_2',...,'CA_n' into the binary form. The Bilinear Transformation Encryption Algorithm makes the random number with a specified range which performs as an index for the selection of positional attribute information. The Bilinear Transformation Encryption Algorithm set the key size depend on CSP key constraints and allocated to collection framework method for specific key generation. The Bilinear Transformation Encryption Algorithm generates specific key id for corresponding user's cipher text based on key size. The cipher text and specific key id generated in the cloud data storage server is used to extract the information for high secure user data requirement which in turn improve the cloud data security in an effective manner.

C. Hash Function for Mapping Data into Corresponding Users

Next, BH-SDS mechanism is used Hash function for mapping the data into corresponding users in client. Initially, Hash function is converts input message into compressed size data. In BH-SDS mechanism, arbitrary size of data is given as input to hash function and the output generated by the hash function is of fixed size. Hash function returns a value. The value produced by the hash function is termed as Message Digest (MD) in BH-SDS mechanism. MD is also termed as hash value. The Hash value generated by the hash function is smaller than the input message in BH-SDS mechanism. Hash function is also recognized as compression function. The task of hash function is shown in below Fig. 3.

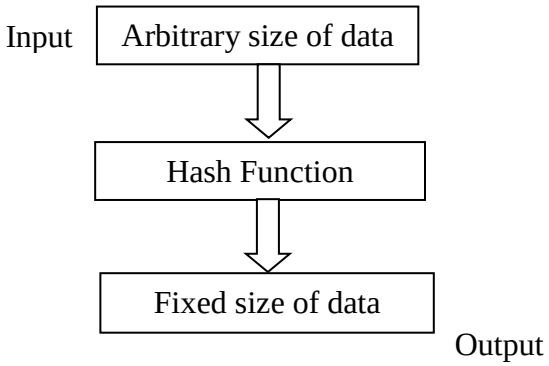

Fig. 3 Task of hash function in BH-SDS mechanism

The property of hash function is pre-image resistance and collision resistance. In pre-image resistance, attackers are prohibited to access the data by having the hash value however it is hard to find the input message. But collision resistance prevents the attacker to identify the two different inputs of arbitrary size with same hash value. So, this collision resistance is also called as collision free hash function. Since the output generated by the hash function is of fixed size, no collision exists between hash function. After converting the fixed size, hash function is used to map the data into appropriate user in cloud environment.

Hash function is used in BH-SDS mechanism for mapping client required data into the corresponding users with aiming at improving the security and accessing of speed of data in cloud. Let us consider cloud database contains numerous data like '$Data1, Data2, ...Datan$' and cloud environment comprises of number of users like '$User1\ User2, ...Usern$'. BH-SDS mechanism is maps the data into corresponding cloud users by using hash function. The Mapping process of hash function is shown in below Fig.4

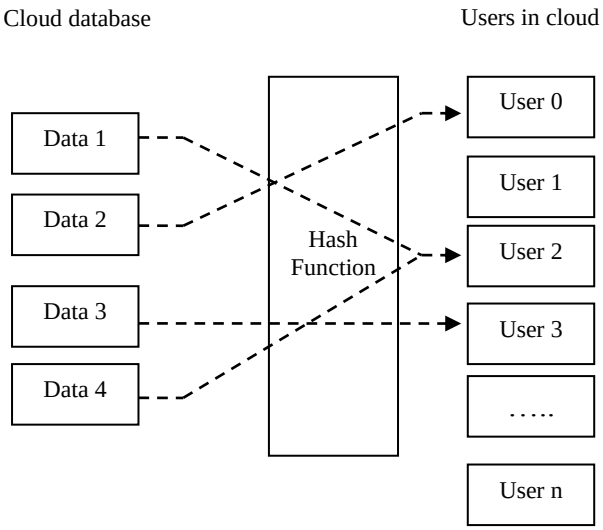

Fig. 4 Mapping Process of Hash Function

In BH-SDS mechanism, Key generated in the Bilinear Transformation Encryption Algorithm needs to be mapped with pre-stored keys in the cloud service provider for identifying corresponding users in could environment. The algorithmic process of mapping using hash function is described in below Fig. 5.

// Hash Function Mapping Algorithm

Input: Specific key id's: $k1, k2 \ldots kn$ and cipher text

Output: Mapping of data into corresponding users in cloud

Begin

Step 1: For all client requests

Step 2: CSP get specific key id's obtained from Bilinear Transformation Encryption Algorithm

Step 3: CSP allocate data location for the corresponding data users

Step 4: Pre-stored keys are generated for the data user location id to perform mapping process

Step 5: Pre-stored keys are mapped to the Specific Key id's for providing data to the user

Step 6: End For

End

Fig. 5 Hash Function Mapping Algorithm

As shown in Fig. 5, Hash Function Mapping Algorithm is used in BH-SDS mechanism to perform mapping process and to improve the data accessing speed in cloud. In Hash Function Users in cloud the cloud service provider gets the specific key ids $k1, k2 \ldots kn$ and cipher text of user's request obtained from Bilinear Transformation Encryption Algorithm to form bilinear transformation. Then the cloud service provider allocates the data location to the data of corresponding users in cloud environment. Next, the pre-stored keys are generated for data user location id to perform mapping processing. In BH-SDS mechanism, Hash Function Mapping Algorithm is performs mapping of pre-stored keys in cloud service provider to the specific key ids for preserves conditional attributes from clients in cloud. With the support of above process, BH-SDS mechanism is effectively performs mapping of client request data into the their corresponding users in cloud environment which results in fast accessing of the data communications by cloud users. This in turn improves the cloud data security in a significant manner.

D. Bilinear Mapping Transformation Function for Decrypting Conditional Attributes and User Required Data

Finally, the proposed BH-SDS mechanism is performs decryption process for separating the user required data and the conditional attribute. On the cloud data storage server side, the conditional attribute is decrypted to complete the data transaction process in BH-SDS mechanism with higher security. In The decryption operation on the cloud server side is used to get the specific user ID with the encrypted conditional attribute. The decryption operation on the cloud server side is formalized as below,

$$Original\ \ Specific\ Id\ Generation\ (M) = \frac{C_1 - GC_2}{GCA_2 - CA_1} \quad \text{-------} \quad (4)$$

In equation (4), 'CA_1'and 'CA_2' signifies the conditional attributes for which the original specific id generation is accomplished by using two constants such as 'C_1' and 'C_2' with 'G' denoting the cipher specific id generation. Every users in cloud environment is used Bilinear Transformation Decryption Algorithm to get the original data. The Bilinear Transformation Decryption Algorithm is described as follows,

// Bilinear Transformation Decryption Algorithm

Input: Cipher text

Output: original data

Begin

Step 1: Change cipher attribute request message into decimal factor

Step 2: Calculate Bilinear Inverse Mapping Transformation by using (4)

Step 3: Original Specific Id produces the original request message into the binary form

Step 4: binary form is changed into text form

Step 5: Repeat the steps 1 to 3 until entire message is decrypted.

End

Fig. 6 Bilinear Transformation Decryption Algorithm

As shown in Fig. 6, Bilinear Transformation Decryption Algorithm is used in proposed BH-SDS mechanism for getting original user required data in cloud environment. The Bilinear Transformation Decryption Algorithmic process comprises of four steps as follows. Initially, this Algorithm change the cipher attribute request message into decimal factor. Then, it calculates Bilinear Inverse Mapping Transformation for generating original specific id. After that, original request message obtained from the Original Specific Id is converted into text form with the aim of obtaining original user required data.

IV. EXPERIMENTAL SETTING

The Bilinear Hashing based Secured Data Storage (BH-SDS) mechanism is implemented in Java language using Amazon Access Samples dataset. The CloudSim simulator toolkit has been used as a simulation platform with 8 GB of RAM and 1 TB of storage space. Amazon Access Samples dataset information is employed on the transaction processing among cloud users and cloud servers. Amazon Access Samples dataset is a sparse data set, less than 10% of the attributes are utilized for each sample. The link is to a '*.tgz' file that includes of two files like [amzn-anon-access-samples-2.0.csv] this file contains the access for users [amzn-anon-access-samples-history-2.0.csv] this file contains the access history for a given user. Amazon Access Samples dataset contains users and their assigned access.

The file contains 4 categories of attributes. 1) [PERSON_{ATTRIBUTE}] which category describes the 'user' who was given access. The [PERSON_ID] column is the primary key column for the file. 2) [RESOURCE_{ID}] which type of attributes are the resources that a users can probably have access to. A user will contain a 1 in this column while they have access to it otherwise it will be 0. 3) [GROUP_{ID}] which type of attributes are the groups that a users can possibly have access to. A user will include a 1 in this column while they have access to it otherwise it will be 0. 4) [SYSTEM_SUPPORT_{ID}] which type of attributes are the organization that a user can probably be supporting.

The performance of BH-SDS mechanism is compared against with the existing methods namely Cooperative Provable Data Possession (CPDP) [1] scheme and data integrity checking algorithm [2]. Experimental evaluation using BH-SDS mechanism is conducted on various factors such as such as storage size, accessing speed and cloud data security.

V. DISCUSSION

The efficacy of BH-SDS mechanism is compared against with exiting two methods namely, Cooperative Provable Data Possession (CPDP) [1] scheme and data integrity checking algorithm [2]. The performance of PC-MLD technique is evaluated along with the following metrics.

A. Measurement of Accessing Speed

In BH-SDS mechanism, the accessing speed is defined as the rate at which the data get accessing from the cloud storage server. The accessing speed is measured in terms of Bytes per second (Bps) and formulated as below,

$$\text{accessing speed} = \frac{\text{accessed data from the cloud stroage}}{\text{accessing time}} \quad \ldots\ldots (5)$$

When the accessing speed is higher, the method is said to be more efficient.

TABLE I
TABULATION FOR ACCESSING SPEED

Number of transaction size (KB)	Accessing Speed (Bps)		
	BH-SDS mechanism	CPDP scheme	Data Integrity Checking Algorithm
50	15	12	9
100	17	14	11
150	19	16	13
200	21	18	15

250	23	20	17
300	25	22	21
350	27	24	23

The performance of data accessing speed with respect to different number of transactions size in the range of 50 and 350 is elaborated in Table 1. From the table value, it is illustrative that the data accessing speed using BH-SDS mechanism is higher as compared to other methods [1], [2].

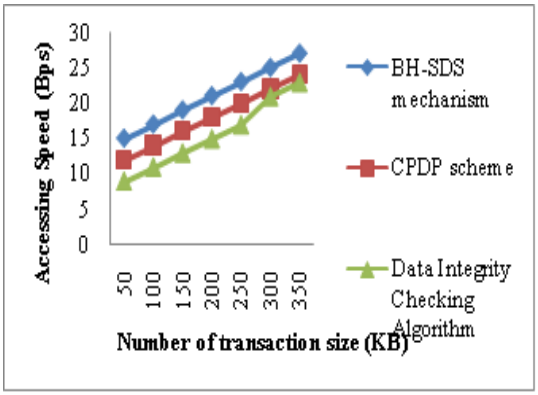

Fig. 7 Measurement of Accessing Speed

Fig. 7 demonstrates the data accessing speed of BH-SDS mechanism versus different number of transactions size in the range of 50-350. As shown in figure, the data accessing speed using BH-SDS mechanism provides better performance as compared to the other two methods namely CPDP scheme [1] and data integrity checking algorithm [2]. Besides, while increasing the number of transaction data size, the data accessing speed is gets also increased. But comparatively the data accessing speed using proposed BH-SDS mechanism is higher. This is because of the application of hash function in BH-SDS mechanism. With the support of hash function, BH-SDS mechanism performs of mapping of data in to their corresponding users in cloud. As a result, BH-SDS mechanism is improves the data accessing speed by 15% as compared to CPDP scheme [1] and 28% as compared to data integrity checking algorithm [2] respectively.

B. Measurement of Cloud Data Security

In BH-SDS mechanism, the clod data security is defined as the amount of security provided to the clients by the cloud server on performing data transmission over cloud servers. When the higher the cloud data security is higher,

TABLE 2
TABULATION FOR CLOUD DATA SECURITY

Number of transaction size (KB)	Cloud Data Security (%)		
	BH-SDS mechanism	CPDP scheme	Data Integrity Checking Algorithm
50	80.12	72.54	64.21
100	83.32	75.46	67.58
150	86.54	78.25	70.21
200	89.24	81.69	73.64
250	92.57	84.97	77.63
300	95.87	87.36	80.89
350	98.41	90.57	83.45

To determine the performance of cloud data security, comparison is made with two other existing methods CPDP scheme [1] and data integrity checking algorithm [2]. In Table 2, the number of transaction size is varied in range of 50 to 350. From the table value, it is illustrative that the cloud data security using the proposed BH-SDS mechanism is higher when compared to other existing methods [1], [2].

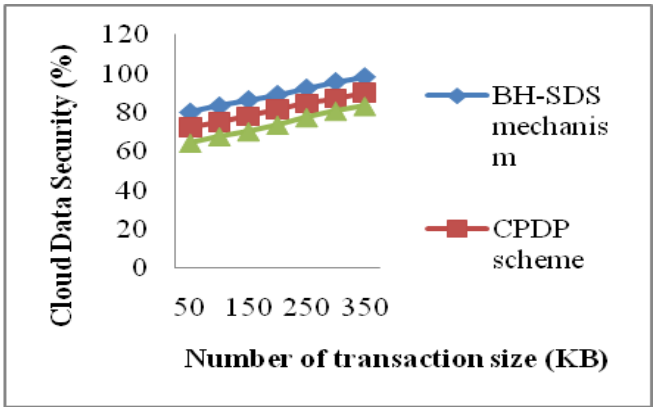

Fig. 8 Measurement of Cloud Data Security

Fig. 8 demonstrates the cloud security for data transmission versus different number of data transactions size in the range of 50-350. As shown in figure, cloud data security using BH-SDS mechanism provides better performance when compared to two other methods namely CPDP scheme [1] and data integrity checking algorithm [2]. Besides, while increasing the number of transaction data size, the cloud data security is gets also increased. But comparatively the cloud data security using proposed BH-SDS mechanism is higher. This is because of the Bilinear Mapping Transformation Function in BH-SDS mechanism which efficiently encrypts

the data with conditional attributes for securing the client required data in cloud environment. As a result, BH-SDS mechanism is improves the cloud data security by 9% as compared to CPDP scheme [1] and 17% as compared to data integrity checking algorithm [2] respectively.

C. Measurement of Cloud Data Storage Capacity

In BH-SDS mechanism, Cloud data storage capacity refers to the data storage performed in cloud environment based on the number of transaction threads, data to be stored and the time taken for data storage. Cloud data storage capacity is measured in terms of kilo bits per second (kbps) and is mathematically formulated as given below.

$$DS = T * d * time \quad \text{-------} \quad (6)$$

From (6), 'DS' refers to the cloud data storage capacity, 'd' states the data to be stored with respect to time '$time$' respectively. When the cloud data storage capacity is higher, the method is said to be more efficient.

TABLE 3
TABULATION FOR CLOUD DATA STORAGE CAPACITY

Number of transaction size (KB)	Cloud Data Storage Capacity (Kbps)		
	BH-SDS mechanism	CPDP scheme	Data Integrity Checking Algorithm
50	85.3	76.5	64.6
100	96.5	87.8	75.6
150	107.4	99.4	86.4
200	118.5	110.8	97.3
250	129.8	121.5	107.2
300	140.9	132.8	118.7
350	151.7	143.5	129.6

The Cloud Data Storage Capacity using BH-SDS mechanism is elaborated in table 3. We consider the framework with different number of transactions size in the range of 50 to 350 is taken for experimental purpose using Java language. From the table value, it is illustrative that the Cloud Data Storage Capacity using PC-MLD technique is higher when compared to the other existing methods [1], [2].

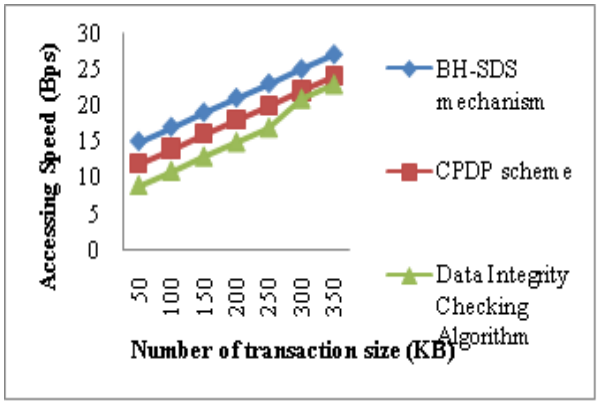

Fig. 9 Measurement of Cloud Data Storage Capacity

Fig. 9 shows the cloud data storage capacity using proposed BH-SDS mechanism and exiting CPDP scheme [1] and data integrity checking algorithm [2]. As shown in figure, the cloud data storage capacity using BH-SDS mechanism provides the better performance when compared to other existing methods [1], [2]. Besides, while increasing the number of transaction data size, the cloud data storage capacity is gets also increased. But comparatively the cloud data storage capacity using proposed BH-SDS mechanism is higher. This is because of the application of hash function in BH-SDS mechanism. Hash function applied in BH-SDS mechanism is efficiently converts input message into compressed size data which in turn improves the cloud data storage capacity in a significant manner. As a result, BH-SDS mechanism is improves the cloud data storage capacity by 7% as compared to CPDP scheme [1] and 9% as compared to data integrity checking algorithm [2] respectively.

V. CONCLUSION

In this work, an effective novel framework is designed called Bilinear Hashing based Secured Data Storage (BH-SDS) mechanism to improve the cloud data security and to improve the data accessing speed in cloud. At first, the request is sent from the clients and then conditional attributes are required to be evaluated for securing the clients data requirement in cloud. After that, the evaluated conditional attributes are encrypted with data, with the aid of bilinear mapping transformation function with the objective of enhancing the cloud data security. Next, BH-SDS mechanism is used hash function to map the client requirement data into their appropriate users in cloud with the objective of improving the data accessing speed. In BH-SDS mechanism, hash function is significantly converts input message into compressed size data which in turn improves the cloud data storage capacity in an effective manner. Finally, BH-SDS mechanism is decrypt the data and conditional attribute, with help of bilinear transformation decryption algorithm which

result improved cloud data security. The performance of BH-SDS mechanism is tested with Amazon Access Samples dataset and compared with existing methods. With the experiments conducted for BH-SDS mechanism, it is observed that of data transmission performed in cloud environment provides more accurate results as compared to state-of-the-art works. The experimental results show that BH-SDS mechanism provides better performance with an improvement of cloud data security by 8% and to improve the accessing speed by 22% when compared to state-of-the-art works.

REFERENCES

[1] Yan Zhu, Hongxin Hu, Gail-Joon Ahn and Mengyang Yu, "Cooperative Provable Data Possession for Integrity Verification in Multi-Cloud Storage", IEEE Transactions on Parallel and Distributed Systems, Volume: 23, Issue: 12, February 2012, Pages: 2231 – 2244.

[2] Dr. Nedhal A. Al-Saiyd and Nada Sail, "Data Integrity in Cloud Computing Security" Journal of Theoretical and Applied Information Technology, 31st December 2013, Volume: 58, Issue: 3, Pages 1-12.

[3] Chandramohan Dhasarathan, Vengattaraman Thirumal, Dhavachelvan Ponnurangam, "A secure data privacy preservation for on-demand cloud service", Journal of King Saud University – Engineering Sciences, Elsevier, 2015

[4] Hongwei Liu, Peng Zhang and Jun Liu, "Public Data Integrity Verification for Secure Cloud Storage", Journal of Networks, Volume: 8, Issue: 2, February: 2013, Pages: 373-380.

[5] Hendrik Decker and Davide Martinenghi, "Inconsistency-Tolerant Integrity Checking", IEEE Transactions on Knowledge and Data Engineering, Volume: 23, Issue: 2, February 2011, Pages: 218-234.

[6] Wenjuan Xu, Xinwen Zhang, Hongxin Hu, Gail-Joon Ahn and Jean-Pierre Seifert, "Remote Attestation with Domain-Based Integrity Model and Policy Analysis", IEEE Transactions on Dependable And Secure Computing, Volume: 9, Issue. 3, May/June 2012, Pages 429-442.

[7] Rongmao Chen, Yi Mu; Guomin Yang, Fuchun Guo, Xiaofen Wang, "Dual-Server Public-Key Encryption with Keyword Search for Secure Cloud Storage", IEEE Transactions on Information Forensics and Security , Volume: 11, Issue: 4, Pages: 789 – 798, Year: 2016

[8] Ke Han, Qingbo Li ,Zhongliang Deng, "Security and efficiency data sharing scheme for cloud storage", Chaos, Solitons and Fractals, Elsevier, Volume 86, May 2016, Pages 107–116

[9] Heng He, Ruixuan Li, Xinhua Dong, Zhao Zhang, " Secure, Efficient and Fine-grained Data Access Control Mechanism for P2P Storage Cloud", IEEE Transactions on Cloud Computing, Volume: 2, Issue: 4, Pages: 471 – 484, Year: 2014

[10] Tao Jiang, Xiaofeng Chena, Jin Li, Duncan S. Wongc, Jianfeng Maa, Joseph K. Liu, "Towards secure and reliable cloud storage against data re-outsourcing", Future Generation Computer Systems, Elsevier, Volume 52, November 2015, Pages 86–94

[11] Chang Liu, Rajiv Ranjan, Chi Yang, Xuyun Zhang, Lizhe Wang, Jinjun Chen, "MuR-DPA: Top-down Levelled Multi-replica Merkle Hash Tree Based Secure Public Auditing for Dynamic Big Data Storage on Cloud", IEEE Transactions on Computers, Year: 2015, Volume: 64, Issue: 9 Pages: 2609 - 2622

[12] Mehdi Sookhaka, Abdullah Gani, Q Muhammad Khurram Khan, Rajkumar Buyya, "Dynamic remote data auditing for securing big data storage in cloud computing", Information Sciences, Elsevier, 2015

[13] Kan Yang, Xiaohua Jia, Kui Ren, Bo Zhang, "DAC-MACS: Effective Data Access Control for Multi-Authority Cloud Storage Systems", INFOCOM, 2013 Proceedings IEEE, Pages: 2895 – 2903, Year: 2013

[14] Xinyue Cao, Zhangjie Fu, and Xingming Sun, "A Privacy-Preserving Outsourcing Data Storage Scheme with Fragile Digital Watermarking-Based Data Auditing", Journal of Electrical and Computer Engineering, Hindawi Publishing Corporation, Volume 2016 (2016), Article ID 3219042, 7 pages

[15] Vinothkumar Muthurajan, Balaji Narayanasamy, "An Elliptic Curve Based Schnorr Cloud Security Model in Distributed Environment", The Scientific World Journal, Hindawi Publishing Corporation, Volume 2016 (2016), Article ID 4913015, 8 pages

[16] Kai He, Chuanhe Huang , Hao Zhou, Jiaoli Shi, Xiaomao Wang, Feng Dan, "Public auditing for encrypted data with client-side deduplication in cloud storage", Wuhan University Journal of Natural Sciences, Springer, August 2015, Volume 20, Issue 4, pp 291-298

[17] Ramalingam Sugumar, Sharmila Banu Sheik Imam, "Symmetric Encryption Algorithm to Secure Outsourced Data in Public Cloud Storage", Indian Journal of Science and Technology, Volume 8, Issue 23, September 2015

[18] Kalpana Batra, Ch. Sunitha, Sushil Kumar, " An Effective Data Storage Security Scheme for Cloud Computing", International Journal of Innovative

Research in Computer and Communication Engineering, Vol. 1, Issue 4, June 2013

[19] Lan Zhou, Vijay Varadharajan, Michael Hitchens, "Cryptographic Role-Based Access Control for Secure Cloud Data Storage Systems", Springer, Security, Privacy and Trust in Cloud Systems, pp 313-344, 2013

[20] Hasan Omar Al-Sakran, "Accessing Secured Data in Cloud Computing Environment", International Journal of Network Security & Its Applications (IJNSA) Vol.7, No.1, January 2015