PRESERVING DATA INTEGRITY WITH OPTIMAL AUTHENTICATION IN CLOUD COMPUTING

S. Muthurajkumar

Assistant Professor,
Department of Computer Technology,
Madras Institute of Technology (MIT) Campus, Anna University, Chrompet, Chennai, India.

Abstract: Cloud storage provides data storage and management service which is a cloud computing system. However, cloud storage also causes a series of security problems. When using cloud storage, users do not actually control the physical storage of their data and it results in the separation of ownership and management of data. Here, the main challenge is that the client has to update his secret keys in every time period, which may inevitably bring in new local burdens to the client. In this work, we focus on how to make the key updates as transparent as possible for the client and propose a new algorithm called cloud storage with Optimal Key Authentication. In this proposed algorithm, the Two Factor Authentication (2FA) is leveraged in many existing public auditing designs, which play the role of authorized party in our case, and make it in charge of both the storage auditing and the secure key updates. In our design, 2FA only needs to hold an encrypted version of the client secret key while doing all these burdensome tasks on behalf of the client. The client only needs to download the encrypted secret key from the 2FA when uploading new files to cloud. The proposed Temporal Reed Hash-Solomon algorithm is designed to divide data into different parts with completely efficient encoding and decoding. It ensures an effective protection of data integrity. Thus, our proposed schemes are efficient from the experiment results.

***Index Terms* - Cloud Computing, Data Integrity, Data Storage.**

I. INTRODUCTION

Nowadays, large number of data owners decided to store their individual data in the cloud which can help them to attain the on-demand high-quality applications and services. It also reduces the cost of data management and storage facility spending. Due to the scalability and high efficiency of cloud servers, the way for public data access is much more scalable, low-cost and stable, especially for the small enterprises. By considering today's technology in mind the cloud computing will have a future trend in many areas. It will give the storing and management of their data. Google Drive, Dropbox, icloud, baidu cloud, etc are cloud storage services provided by companies. Network technology and distributed file system technology are used to make different storage devices to work together.

However coming to the security issues, the privacy problems are significant among these issues. Here, the goal of cloud computing is to improve efficiency and reduce the amount of data that needs to be transported to the cloud for data processing, analysis and storage. If attackers are intelligent and launching attack against cloud system, then it is easy to break cloud user password or attacker is malicious insider and it is possible to hack someone's user password easily and try to getting unauthorized access of cloud system. For each time period, the client has to run the key update algorithm in order to move forward his secret key and it will be consider as a new local burden for them. Some clients for each time period might not like doing extra computations by themselves since they are having limited computation resources. To overcome this problem, we propose different technique for key updation scenarios frequently so that key updation will be transparent as possible for the client by using cloud storage with Optimal Key Authentication Algorithm and to provide security to cloud data from unauthorized user by creating confusion by using Temporal Reed Hash-Solomon algorithm. In Optimal Key Authentication algorithm, key-update operations are not performed by the client, but by an authorized party. The authorized party holds an encrypted secret key of the client for cloud storage auditing and updates it under the encrypted state. The client downloads the encrypted secret key from the authorized party and decrypts it only when he would like to upload new files to cloud. In our design, the Two Factor Authentication (2FA) plays the role of the authorized party who is in charge of key updates. The 2FA does not know the real secret key of the client for cloud storage auditing, but only holds an encrypted version. The system makes the client's file secure and the decryption operation efficient. Meanwhile, the 2FA can complete the key updates under the encrypted state. Another proposed Temporal Reed Hash-Solomon algorithm will produce some excess data block and that will be used for data integrity.

The proposed scheme can provide the privacy from the interior mainly from the Cloud Service Provider (CSP). By using this technology, we can preserve data integrity while storing and retrieving files from cloud. The main objective of the work is to make the key updates as transparent as possible for the client and propose a new algorithm called cloud storage with Optimal Key Authentication. In this proposed algorithm, we leverage the 2FA in many existing public auditing designs that plays the role of authorized party in our case, and make it in charge of both the storage auditing and the secure key updates. In our design, 2FA only needs to hold an encrypted version of the clients secret key while doing all these burdensome tasks on behalf of the client. The client only needs to download the encrypted secret key from the 2FA when uploading new files to cloud. The proposed Temporal Reed Hash-Solomon algorithm is designed to divide data into different parts with completely efficient encoding and decoding. It ensures an effective protection of data integrity of files.

II. LITERATURE SURVEY

There are many researches that include the secure cloud data storage and retrieval methods in recent years. Z. Xia, Y. Zhu, X. Sun, Z. Qin and K. Ren [27] considered that cloud computing provided on-demand access to ample computation and storage resource, which made it a primary choice for image storage and Content-Based Image Retrieval outsourcing. By deploying such image retrieval outsourcing, the data owner is no longer needed to maintain the image database locally. An authorized data user can query the cloud for CBIR service without interacting with the data owner. Despite the tremendous benefits, privacy becomes the biggest concern about CBIR outsourcing. They have studied the privacy-preserving CBIR outsourcing problem and present a practical solution. We exploit techniques from security, image processing and information retrieval domains to achieve secure and efficient searching over encrypted images. The proposed scheme supports local-feature based CBIR with the Earth Mover's Distance (EMD) as similarity metric. In particular, a secure transformation was designed so that the cloud server can solve the EMD problem with the privacy preserved. Local sensitive hash in employed to achieve constant search efficiency. They designed a two-stage structure with LSH and the search efficiency is improved. In the first stage, dissimilar images are filtered out by pre-filter tables to shrink the search scope. In the second stage, the remaining images are compared under EMD metric one by one for refined search results.

T. Wang, J. Zhou, X. Chen, G. Wang, A. Liu and Y. Liu [24] considered the cloud storage, users do not actually control the physical storage of their data and it results in the separation of ownership and management of data. The CSP can freely accessed and searched the data stored in the cloud. Meanwhile the attackers can also attacked the CSP server and obtained the user's data. The above two cases both made the users fell into the danger of information leakage and data loss. Traditional secure cloud storage solutions for the above problems are usually focused on accessed the restrictions or data encryption. All of these solutions cannot solve the internal attack. The privacy protection in cloud storage, they proposed a TLS framework based on fog computing model and design a Hash-Solomon algorithm. Due to the allocated ratio of data blocks stored in different servers reasonably, they ensured the privacy of data in each server. On another hand, cracking the encoding matrix is impossible theoretically. Besides, hash transformation has been protected the fragmentary information. Through the experiment test, this method can efficiently complete encoding and decoding without influence of the cloud storage efficiency. Hence, they designed a reasonable comprehensive efficiency index and achieved the maximum efficiency.

J. Yu, K. Ren and C. Wang [29] considered the cloud storage which provided great benefit to users but also it brought new security challenging problems. One important security problem is how to efficiently check the integrity of the data stored in cloud. Many auditing protocols for cloud storage have been proposed to deal with this problem. The cloud can be able to easily hide the data loss incidents for maintaining its reputation, even discard the client's data rarely accessed for saving the storage space once the client's secret key for storage auditing is exposed to cloud. The client has been executed the key update algorithm to move forward the secret key for each time period. Some clients might not like doing extra computations by themselves since having limited computation resources for each time period. Key updation with verifiable outsourcing using the first cloud storage auditing protocol. So, the authorized party should only hold an encrypted version of the user's secret key for cloud storage auditing and for cloud storage auditing, this authorized party from the encrypted version updated their secret keys. The key updation are transparent to the client and outsourced to the TPA. It should be very efficient for the client to recover the real secret key from the encrypted version that is retrieved from the authorized party and the client has been verified the validity of the encrypted secret key after the client has retrieved it from the authorized party.

C. Lin, Y. Bi, G. Han, J. Yang, H. Zhao and Z. Liu [14] considered that efficient file transfer scheduling schemes in cloud computing are critical for file downloading or uploading. Sometimes, the files are required to be transferred with different Quality of Service (QoS). Enormous increase in the scale of big-file generated by cloud computing has brought many challenges to both computational and transfer infrastructures. For large file transfer in cloud computing, the scheduling with intelligent routing was indispensable and improved the efficiency of network resource, especially when file transfer over multiple paths is allowed. But it is a non-trivial task to efficiently select a set of feasible paths for file transfer under a tight delay constraint. The challenge was feasible strategy to compute flows for each file from the overall. Some file transfers received less bandwidth, but the overall network utilization and the multi-file transfer would be benefitted. They have studied the SFTS and MFTS problems in cloud computing. For SFTS problem, they adopted maximum flow over time issue and auxiliary graph technique and propose d a heuristic and an exact algorithm, respectively. Simulation results have shown that both of their proposed algorithms can solved the SFTS problem. Especially, the heuristic can solve the SFTS problem efficiently although the exact algorithm can achieved better QoS, the transfer delay. For MFTS problem, we proposed a heuristic with an intelligent scheme which can maximize the throughput and schedule the multi-file flow dynamically, by solving the maximum multi-file flow over time problem. Simulation results have shown that our algorithm schedules the multi-file flow dynamically and can achieved high network utilization. The corresponding simulation results demonstrated that the proposed algorithm can efficiently support multiple file transfer over multiple paths in terms of delay and bandwidth utilization.

D. He, N. Kumar, M. K. Khan, L. Wang and J. Shen [9] considered that with user's increasing demand of high services quality, a huge amount of data should be processed in time by their mobile device. The mobile devices resources are limited and they cannot satisfied the user's requirements. The data weakness became a performance bottleneck of various applications based on mobile devices. All the messages are transmitted by using the wireless technology in Mobile Cloud Computing services environment controlled the communication channel easily by intercepting, delaying, and modifying transmitted message. Then, the MCC services environment is more vulnerable to various types of attacks than traditional cloud computing services environment. Only the legal user has been accessed the MCC services and stop the adversary accessing MCC services. The Privacy-Aware Authentication (PAA) scheme is very crucial for address security problem in MCC services environment because the participants' identities and their privacy has been identified and protected. They proposed a new PAA scheme for Mobile Cloud Computing (MCC) services by using an identity-based signature scheme. Security analysis shows that the proposed PAA scheme is able to address the serious security problems existing in Tsai and Lo's scheme and can meet security requirements for MCC services.

X. Li, J. Yuan, H. Ma and W. Yao [13] considered that complementary technology with traditional security mechanism, trust solved the problem of providing corresponding access control based on judging the service behaviour, and it made the traditional security services more robust and reliable by ensuring that all the communicating nodes are trusted during authentication, authorization, or key management. Relying on distributed and intelligent agents, they proposed method for security and QoS-related trust behaviour perceiving and mining. By a plurality of original design, the proposed scheme can efficiently perceived the service behaviour of large-scale virtual machines, and quickly completed the trustworthiness computing of service resources based on these real-time perceiving data. To the best knowledge, this work was the first which provided a lightweight and parallel trust computing scheme based on big data analysis for trustworthy cloud service. Performance analysis and experimental results verified feasibility and effectiveness of the proposed scheme. But key research directions could still be explored in depth. Our proposed system on various cloud collaborative service environment, such as distributed data sharing and remote computing, was evaluated as a key direction and the trust value of cloud resources with different value of the time window is calculated. The proposed system is an innovative and lightweight trust computing scheme based on big data analysis for trustworthy cloud service environment. By a plurality of original design, the proposed scheme can efficiently perceived the service behaviour of large-scale VMs, and quickly complete the trustworthiness computing of service resources based on these large-scale and real-time perceiving data.

Z. Fu, F. Huang, K. Ren, J.Weng, and C.Wang [4] considered that many existing schemes are keyword-based search including single keyword and multi-keywords. Their schemes are allowed the data users and retrieved the interested files and returned the related documents in the encrypted form. Due to connatural localization of keywords as document eigenvectors, the returned results are always imprecise and unable to satisfy intention of users. That means keywords as a document feature had inadequate data which carried relatively little semantic information. And some existing schemes explored the relationships among keywords expanded the retrieval results. When extracting keywords from documents, the relationships among keywords are out of consideration which lead to the limitation of these schemes. So exploring a new knowledge representation with more semantic information compared with keywords realized the searchable encryption is a challenging and essential task. Compared with the previous study, they proposed two more secure and efficient schemes and solved the problem over encrypted outsourced data in privacy-preserving smart semantic search based on conceptual graphs. Considering various semantic representation tools, they have selected Conceptual Graphs as our semantic carrier because of its excellent ability of expression and extension. They used Tregex which simplified the key sentence and made it more generalizable and improved the accuracy of retrieval. They transferred the CG into its linear form with some modification creatively which made the quantitative calculation on CG and fuzzy retrieval in semantic level possible. They used different methods to generate indexes and constructed two different schemes with two enhanced schemes respectively against two threat models by introducing the frame of MRSE. They implemented our scheme on the real data set to prove its effectiveness and efficiency.

Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang [6] considered that data stored in the cloud may suffered from malicious use by cloud service providers since data owners have no longer direct control over data. Data privacy and security is considered as a recommended practice for data owners and encrypted the data before uploading onto the cloud. Although it protected the data security from illegal use both from untrusted cloud service providers and external users and it made data utilization more difficult since many techniques based on plain-text are no longer applicable to ciphertext. Therefore, an efficient search technique for encrypted data is extremely urgent to explore. Most users are typically untrained and casual, they might not input the explicit query terms that accurately matched their true search intentions. As a result, the returned result is definitely not what users really want. In such cases, users want to retrieve more results as similar as possible to query terms. The common technique used in IR is query extension, which can extend the original query terms according to some rules before submitting search request. Similarly, query extension is remained to be adjusted and improved to meet searchable encryption in cloud computing. They studied and solved the problem of personalized multi-keyword ranked search over encrypted data (PRSE) while preserving privacy in the cloud computing. In PRSE, with the help of semantic ontology WordNet, user interest model for individual user is built by analyzing the user's search history. And they adopt a scoring mechanism to express user interest smartly by calculating the similarity score between different types of related words and the keyword.

Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren [26] considered that a large image database usually consisted of millions of images. Content-Based Image Retrieval (CBIR) services typically incurred high storage and computation complexities. Cloud computing offered a great opportunity for the on-demand access to ample computation and storage resources, which made it an attractive choice for the image storage and CBIR outsourcing. By outsourcing CBIR services to the cloud server, the data owner is relieved from maintaining a local image database and interacting with database users online. Despite the tremendous benefits, image privacy became the main concern with CBIR outsourcing. The problem was formulated by considering the work in two types of privacy threats. Firstly, a curious cloud server may look into the owner's database for additional information. Secondly, after receiving the retrieved images, the query user may illegally distribute these images to someone unauthorized for benefits. In cloud computing, they presented privacy-preserving and a scheme called copy- deterrence content-based image retrieval. The secure KNN algorithm is applied to encrypt the visual features. The similarity scores can be directly calculated with the encrypted features by the cloud server, which enables the cloud server to rank the images without the additional communication burden. The locality-sensitive hashing is utilized to improve the search efficiency. For the first time, they considered the dishonest users in an SE scheme and proposed a watermark-based protocol to deter the illegal distribution of images. Overall, the image features are secured against the Ciphertext-only Attack model, the image contents are secured against Chosen-plaintext Attack model, and the search efficiency is improved from O(n) to O(n). Firstly, the proposed watermarking method cannot be regarded as a very robust one. Secondly, a small parameter α will cause a limited number of available watermarks.

Z. Fu, X. Wu, C. Guan, X. Sun, and K. Ren [7] considered that the problem of multi-keyword fuzzy searched over encrypted data problem and did not require a predefined fuzzy set. Their method solved the problems of multi-keyword fuzzy search with high efficiency and accuracy. The most important result was that their scheme does not require a predefined fuzzy set. Some other problems arose in this scheme. First, converting the keyword into a bi-gram set increased the Euclidean distance. Two bi-gram sets are different compared with original sets and the vectors are rarely be threshold into the same bit. Second, the scheme is not effective for other spelling mistakes. A keyword can be misspelled into many forms, not only one-letter mistakes. The spelling mistakes are common and should be considered by the search system. In addition, their scheme could not find the keywords with the same root. Finally, MFSE did not show the relevance between the keywords and files. For the same keyword in different files, its keyword weight should be different, and this difference should be considered during ranking. Thus, the files that are more relevant to the query keyword might not be included in the return results. They investigated the problem of multi-keyword fuzzy ranked search over encrypted cloud data. They proposed a multi-keyword fuzzy ranked search scheme that concretely developed a novel method of keyword transformation and introduced the stemming algorithm. With the two techniques, the proposed scheme is able to efficiently handle more misspelling mistakes. Moreover, the proposed scheme took the keyword weight into consideration during ranking. Their proposed scheme does not require a predefined keyword set and hence enabled the efficient updated file. They also gave thorough security analyses and conducted experiments on real world data sets, which indicated the proposed scheme's potential of practical usage.

Z. Xia, X. Wang, X. Sun, and Q. Wang [25] considered that a general approach to protect the data confidentiality is to encrypt the data before outsourcing. However, the data causes a huge cost in terms of data usability. The existing techniques used for keyword-based information retrieval, which are widely used on the plaintext data, cannot be directly applied on the encrypted data. Downloading all the data from the cloud and decrypt locally is obviously impractical. Researchers have designed some general-purpose solutions with fully-homomorphic encryption but these methods are not practical due to their high computational overhead for both the cloud server and user. The Searchable encryption (SE) schemes have made specific contributions in terms of efficiency, functionality and security. SE schemes enabled the client to store the encrypted data to the cloud and execute keyword search over the ciphertext domain. Some dynamic schemes have been proposed to support inserting and deleting operations on document collection. These are significant works as it is highly possible that the data owners updated their data on the cloud server. But few of the dynamic schemes support efficient multi-keyword ranked search. The proposed a secure tree-based search scheme over the encrypted cloud data, which supports multi-key-word ranked search and dynamic operation on the document collection. In order to obtain high search efficiency and construct a tree-based index structure and propose a Greedy Depth-First Search (GDFS) algorithm based on this index tree.

J. Shen, T. Zhou, D. He, Y. Zhang, X. Sun and Y. Xiang [20] considered that group data sharing in cloud computing has opened up a new area of usefulness to computer networks. With the help of the conference key agreement protocol, the security and efficiency of group data sharing in cloud computing can be greatly improved. Specifically, the outsourced data of the data owners encrypted by the common conference key are protected from the attacks of adversaries. Compared with conference key distribution, the conference key agreement has qualities of higher safety and reliability. The conference key agreement asked for a large amount of information interaction in the system and more computational cost. To combat the problems in the conference key agreement, the symmetric balanced incomplete block design (SBIBD) is employed in the protocol design. They presented a novel block design-based key agreement protocol that supported the group data sharing in cloud computing. Multiple participants can be involved in the protocol and general formulas of the common conference key for participants are derived. They presented an efficient and secure block design-based key agreement protocol by extending the structure of the SBIBD to support multiple participants, which enabled multiple data owners to freely share the outsourced data with high security and efficiency. The SBIBD is constructed as the group data sharing model to support group data sharing in cloud computing. The protocol can provide authentication services and a fault tolerance property. The introduction of volunteers enabled the presented protocol to support the fault tolerance property, thereby making the protocol more practical and secure.

III. Proposed System

The proposed work is based on a cloud computing model, in which data is stored on remote servers and can be accessed from the internet or cloud. The advent of cloud storage enables the organizations and enterprises to outsource their data to third party CSP. Cloud storage provides several benefits to customers that includes cost saving of resources, data availability, simplified convenience, mobility opportunities and scalable service, backup and disaster recovery. These great features attract more and more customers to utilize and store their personal data in cloud storage. The data in the cloud server is maintained, operated and managed by the cloud storage service provider. Moving the data onto the cloud offers significant advantages in resource saving. It also provides great convenience to users as they don’t have to worry about the complexities of hardware, software and their maintenance. Stored files can be accessed from anywhere at any time via internet connection. With cloud computing, they eliminate those headaches that come with storing their own data, because they are not managing hardware and software that becomes the responsibility of an experienced vendor like Salesforce.

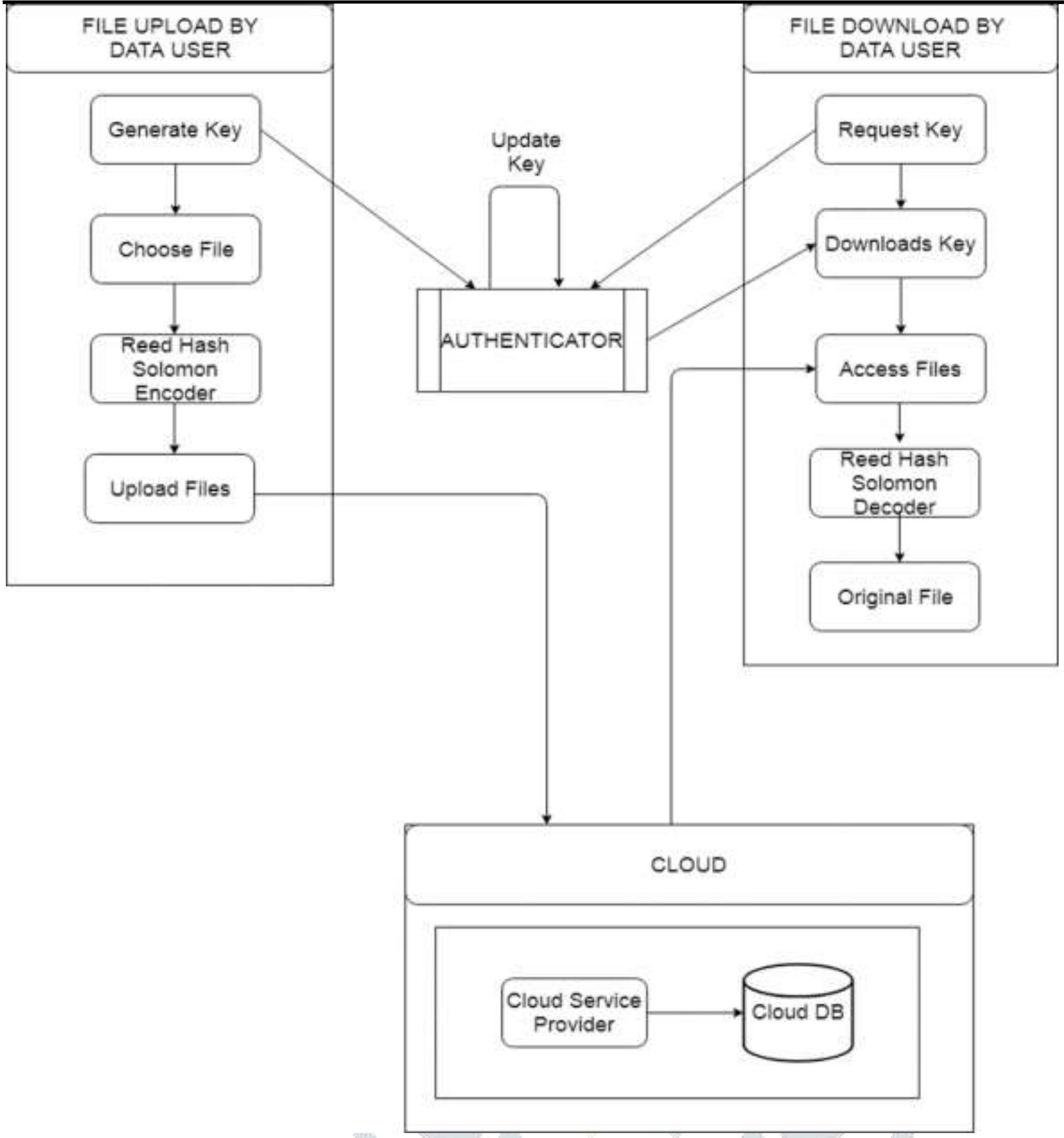


Figure 3.1 Architecture of Preserving Data Integrity with Optimal Authentication

The shared infrastructure means it works like a utility: They only pay for what they need, upgrades are automatic, and scaling up or down is easy. Cloud-based apps can be up and running in days or weeks, and they cost less. With a cloud app, they just open a browser, log in, customize the application, and start using it. Businesses are running all kinds of apps in the cloud, like Customer Relationship Management (CRM), HR, accounting, and much more. Some of the world's largest companies moved their applications to the cloud with Salesforce after rigorously testing the security and reliability of their infrastructure. As cloud computing grows in popularity, thousands of companies are simply rebranding their non-cloud products and services as cloud computing.

As shown in Figure 3.1, we demonstrate that there are three parties in the model: the client, the cloud and the 2FA. The client is the owner of the files that are uploaded to the cloud. The total size of these files are not fixed, that is, the client can upload the growing files to the cloud at different time points. The cloud stores the client files and provides download service for the client. The 2FA plays two important roles: the first is to audit the data files stored in the cloud for the client; the second is to update the encrypted secret keys of the client. Then, the proposed Temporal Reed Hash-Solomon code algorithm helps to encode the data. Data are divided into different pieces of the same size using encoding technology. These parts of the user's data will be stored in the cloud server. Efficient data privacy plays an important role in Cloud Storage. Data User owns the data to be stored in Cloud Server. Data User chooses the data to be stored in Cloud Server and it splits the data into different pieces of the same size in encoded form. Data users upload all the encoded data in the Cloud Server. CSP stores all the data uploaded by data users in Cloud Database. Data users access the CSP to retrieve the data. CSP allows only authorized data users to access the data stored in Cloud Database. The authorization is provided by the respective data owner. The authorized data user accesses the data stored in Cloud Server. After retrieving the data, the data user decodes the data.

The 2FA can be considered as a party with powerful computational capability or a service in another independent cloud. Each file is assumed to be divided into multiple blocks. In order to simplify the description, they do not furthermore divide each block

U	**I**	**O**	**P**
L	**N**	**G**	**S**
M	**R**	**A**	**B**
H	**T**	**W**	**Q**

Figure 3.2 Original File Matrix.

01	**00**	**00**	**00**
00	**01**	**00**	**00**
00	**00**	**01**	**00**
00	**00**	**00**	**01**
1b	**1c**	**12**	**14**
1c	**1b**	**14**	**12**

Figure 3.3 Coding Matrix.

into multiple sectors in the description of our protocol. The 2FA updates the encrypted clients secret key for cloud storage authentication. But the public key keeps unchanged in the whole time period. The client sends the key requirement to the 2FA only when they want to upload new files to the cloud. And then the 2FA sends the encrypted secret key to the client. After that, the client decrypts it to get the real secret key. One downside resolved is that the 2FA should perform the outsourcing computations for key updates under the condition that the 2FA doesn't grasp the secret key of the client. Ancient encoding technique is not appropriate because it makes the key update troublesome to be completed under the encrypted condition. Besides, it will be even additional difficult to alter the client with the verification capability to ensure the validity of the encryption of the secret keys. To address these challenges, we tend to propose to explore the bright technique with homomorphic properties to efficiently encrypt the keys. It permits key updates to be swimmingly performed below the blind version, and additionally makes confirmatory the validity of the encrypted secret keys potential. A cloud storage authentication protocol with secure optimal key updates consists by three algorithms (Algorithm Setup, Encryption key Updation, Decryption), shown below:

3.1 Algorithm Setup

The system setup algorithmic rule is run by the client. It takes as input a security parameter and the associated generates an encrypted initial client's secret key, a secret writing key and a public key. Finally, the client holds Decryption, and sends Encryption Key to the 2FA.

3.1 Encryption Key Updation

The encrypted key update algorithmic rule is run by the 2FA. It takes as input associate degree encrypted client's secret key and also the Public Key (PK) and generates a brand new encrypted secret key.

3.1 Decryption

The key decryption formula is run by the client. It takes as input associate in computing encrypted client's secret key, a Decryption Key (DK) and PK, returns the client's secret key.

IV. Temporal Reed Hash-solomon Algorithm

Temporal Reed Hash-Solomon Algorithm takes a user's data and splits up into n pieces, add k "parity" pieces, and then reconstruct the original data n from (n + k) pieces. In Temporal Reed Hash-Solomon, we should put our data in matrix form.

Data Format:

- Text File
- Audio File
- Video File
- File with valid format

User's Format:

UIOPLNGSMRABHTWQ

Figure 3.2 shows the original file matrix in which each piece is one row of the matrix and the four pieces of the file are **UIOP**, **LNGS**, **MRAB** and **HTWQ** and each 4 bytes long. And here consider n=4 and k=2. Figure 3.3 demonstrates the coding matrix which is created by the Temporal Reed Hash-Solomon algorithm randomly. Next, multiplying with the data matrix to create the coded data. It differs between each and every layer of our proposed framework. In Fig. 3.4, each row of resultant encoding matrix is concatenated to produce the n + k number of encoded files. Here, UIOP are concatenated to produce output.txt.1, [L N G S] are concatenated to produce output.txt.2, [M R A B] are concatenated to produce output.txt.3, [H T W Q] are concatenated to produce output.txt.4, [51 52 53 49] are concatenated to produce output.txt.5, [55 56 57 25] are concatenated to produce output.txt.6. Thus, there will be 6 (n + k) numbers of output encoded files.

Figure 3.5 shows that the data contained in the file can be lost in the cloud server. It is due to the following reasons,

- Technical failure
- Human failure
- Operational failure

$$\begin{bmatrix} 01 & 00 & 00 & 00 \\ 00 & 01 & 00 & 00 \\ 00 & 00 & 01 & 00 \\ 00 & 00 & 00 & 01 \\ 1b & 1c & 12 & 14 \\ 1c & 1b & 14 & 12 \end{bmatrix} x \begin{bmatrix} U & I & O & P \\ L & N & G & S \\ M & R & A & B \\ H & T & W & Q \end{bmatrix} = \begin{bmatrix} U & I & O & P \\ L & N & G & S \\ M & R & A & B \\ H & T & W & Q \\ 51 & 52 & 53 & 49 \\ 55 & 56 & 57 & 25 \end{bmatrix}$$

Figure 3.4 Encoding Matrix Generation.

$$\begin{bmatrix} 01 & 00 & 00 & 00 \\ 00 & 01 & 00 & 00 \\ \cancel{00} & \cancel{00} & \cancel{01} & \cancel{00} \\ \cancel{00} & \cancel{00} & \cancel{00} & \cancel{01} \\ 1b & 1c & 12 & 14 \\ 1c & 1b & 14 & 12 \end{bmatrix} x \begin{bmatrix} U & I & O & P \\ L & N & G & S \\ M & R & A & B \\ H & T & W & Q \end{bmatrix} = \begin{bmatrix} U & I & O & P \\ L & N & G & S \\ \cancel{M} & \cancel{R} & \cancel{A} & \cancel{B} \\ \cancel{H} & \cancel{T} & \cancel{W} & \cancel{Q} \\ 51 & 52 & 53 & 49 \\ 55 & 56 & 57 & 25 \end{bmatrix}$$

Figure 3.5 Data loss.

$$\begin{bmatrix} 01 & 00 & 00 & 00 \\ 00 & 01 & 00 & 00 \\ 1b & 1c & 12 & 14 \\ 1c & 1b & 14 & 12 \end{bmatrix} x \begin{bmatrix} U & I & O & P \\ L & N & G & S \\ M & R & A & B \\ H & T & W & Q \end{bmatrix} = \begin{bmatrix} U & I & O & P \\ L & N & G & S \\ 51 & 52 & 53 & 49 \\ 55 & 56 & 57 & 25 \end{bmatrix}$$

Figure 3.6 Encoding Matrix with Data loss

$$\begin{bmatrix} 01 & 00 & 00 & 00 \\ 00 & 01 & 00 & 00 \\ 8d & F6 & 7b & 01 \\ F6 & 8d & 01 & 01 \end{bmatrix}$$

Figure 3.7 Inversion of Coding Matrix

$$\begin{bmatrix} U & I & O & P \\ L & N & G & S \\ M & R & A & B \\ H & T & W & Q \end{bmatrix} = \begin{bmatrix} U & I & O & P \\ L & N & G & S \\ 51 & 52 & 53 & 49 \\ 55 & 56 & 57 & 25 \end{bmatrix} x \begin{bmatrix} 01 & 00 & 00 & 00 \\ 00 & 01 & 00 & 00 \\ 8d & F6 & 7b & 01 \\ F6 & 8d & 01 & 01 \end{bmatrix}$$

Figure 3.8 Reconstruction of Original Matrix

Multiply the inverse matrix of the coding matrix with the pieces that are available and reconstruct the original data as shown in Fig. 3.6. After multiplying the inverse matrix in both sides of Fig. 3.6, the corresponding coding matrix and its inverse matrix get cancelled which leads to the production of the original data matrix as shown in Fig. 3.7. After reconstructing the original matrix, the input file will get decoded as shown in Fig. 3.8 and the output of the decoding file will be written as output.txt.decoded.

V. Optimal Key Authentication Algorithm

5.1 Algorithm Setup

Step 1: Input a security parameter η.
Step 2: Input Ψ.
Step 3: The client selects α, ß, η ∈ R, and computes E$.
Step 4: Choose two large prime numbers α and ß randomly and independently
of each other such that gcd (αß, (α-1) (ß-1)) = 1. This property is assured if both primes are of equal length.
Step 5: Calculate n,
n = α * ß
Step 6: Calculate λ,
λ = f (α − 1, ß − 1)
Step 7: Calculate r,
r = 1+ n mod n2
Step 8: Choose Ω, x є Z
Step 9: Calculate P$,
P$: (n, r)
Step 10: Calculate D$,
D$: (λ, Ω)
Ensure n divides the order of g by checking the existence of the
following modular multiplicative inverse: μ= L (rλ mod n2)-1 mod n
where function L is defined as L(x)=x-1n
Step 11: E$ = (rΨ) (Ωn) mod n2

5.2 Algorithm for Encryption key Updation

Step 1: Input the E$, P$ to the TPA
Step 2: TPA computes E$ = (rΨ)(Ωn) mod n2
Step 3: TPA updates the E$

5.3 Algorithm for Decryption

```
Step 1: Input the E$, D$, P$
Step 2: Compute Ψ,
Ψ= L (E$λ mod n2) L (rλ mod n2)-1 mod n
        where the constant parameter L(x)=x-1n
```

TEMPORAL REED HASH-SOLOMON ALGORITHM

Algorithm 4.2.1 Reed-Hash Solomon Encoding

```
Step 1: Input the desired file.
Step 2: Input the number of data shards.
Step 3: Input the number of parity shards.
Step 4: Get the size of the input file.
Step 5: Create a buffer holding the file size, followed by the contents of the file.
Step 6: Make the buffers to hold the shards.
Step 7: Fill in the data shards.
Step 8: Creates a Temporal Reed-Hash Solomon codec with the coding loop.
Step 9: Encodes parity for a set of data shards.
Step 10: Create the matrix to use for encoding, given the number of data shards and the number of total shards.
Step 11: Create a matrix, which is guaranteed to have the property that any subset of rows that forms a square matrix is invertible.
Step 12: Build the array of output buffers.
Step 13: Write out the resulting files.
```

Algorithm for Reed-Hash Solomon Decoding

```
Step 1: Input the Chosen File
Step 2: Read in all of the shards that are present.
Step 3: Make empty buffers for the missing shards.
Step 4: Creates a Reed-Hash Solomon codec with the coding loop.
Step 5: if (number Present != total Shard Count)
Step 6:   List all shards, some of which contain data, fills in the ones that don't have data.
Step 7:   Pull out the rows of the matrix that correspond to the shards.
Step 8:   Build a square matrix.
Step 9:   Pull out an array holding just the shards that correspond to the rows of the submatrix.
Step 10:  Invert the matrix.
Step 11:  Pull out the row that generates the shard that has to decode.
Step 12:  Combine the data shards into one buffer for convenience.
Step 13:  Write the decoded file
Step 14: else if (number Present == total Shard Count)
Step 15:  print "All of the shards are present"
Step 16:  Write the decoded file.
Step 17: else
Step 18:  throw Illegal Argument Exception ("Not enough shards present")
Step 19: end
```

VI. RESULTS

We demonstrate that data has been divided into different parts as we mentioned in run time by efficient encoding and generating a new encrypted secret key in Fig. 5.1. The result of the n+k number of splittings of the original file APJ.mp4 is shown in Fig. 5.2. In Fig. 5.4, we demonstrate the decryption by taking the encrypted client's secret key as input and returns the client's secret key. Then, by efficient decoding, reconstructing the divided data and the input file will get decoded. The result of reconstructing the original file APJ.mp4 from the n+k number of encoded files are shown in Fig. 5.6.

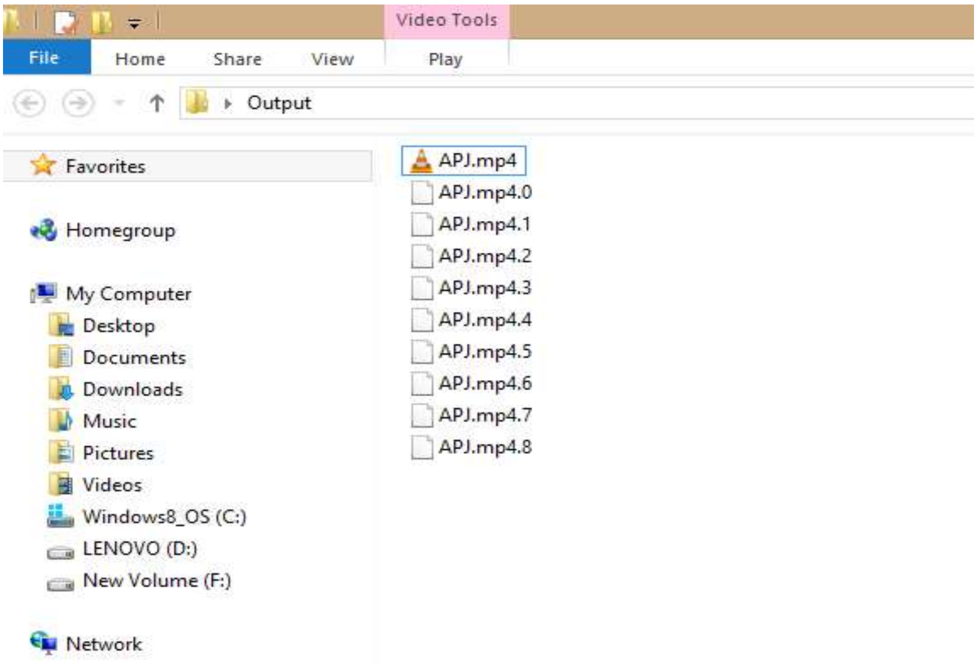
```
Do you Wish to Encode File or Decode File
Press 1 for Encode File and 2 for Decode File and 3 to exit
1

Enter a number to be encrypt:
33

Encrypted text:
46253265851811600008903096190229305353415953975159240046572715133000048471276040751600896544585084862095
96884288117352479296890369318442696926592756364732109700642791012513233887125155625209431886861418980443
5794027326725105601214866356155160797549809823021885675438294679472770565089598689595338116930715748
Updated EncKey:
46253265851811600008903096190229305353415953975159240046572715133000048471276040751600896544585084862095
96884288117352479296890369318442696926592756364732109700642791012513233887125155625209431886861418980443
5794027326725105601214866356155160797549809823021885675438294679472770565089598689595338116930715759
```

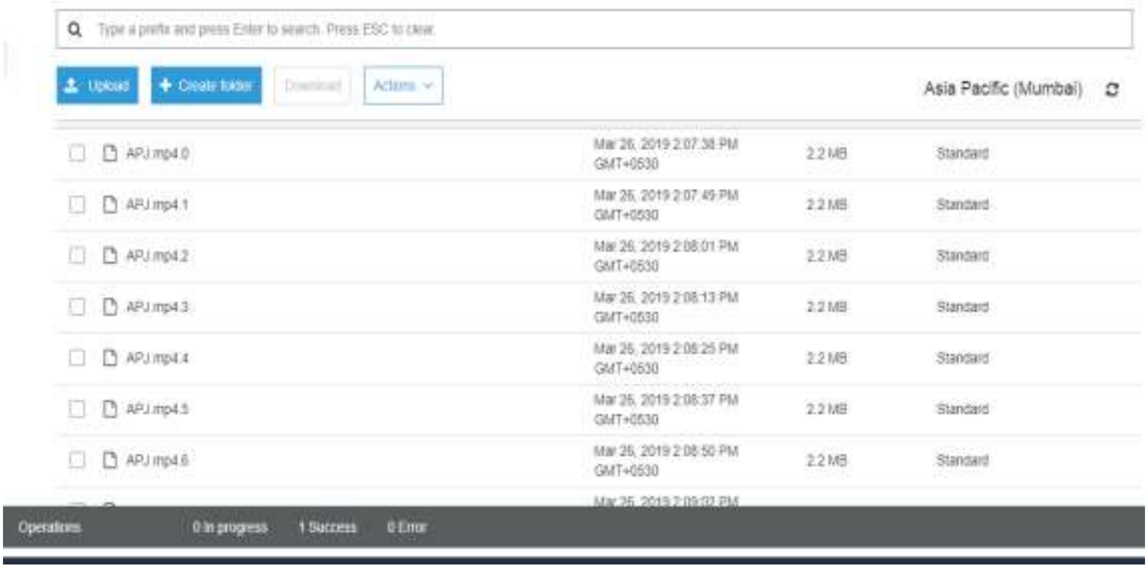
```
Enter Number Of Splittings of original file
6
Enter Number Of Parity Data
4
Total Number Of Files After Encoding: 10
wrote C:\Users\Narmadha\Desktop\Output\APJ.mp4.0
wrote C:\Users\Narmadha\Desktop\Output\APJ.mp4.1
wrote C:\Users\Narmadha\Desktop\Output\APJ.mp4.2
wrote C:\Users\Narmadha\Desktop\Output\APJ.mp4.3
wrote C:\Users\Narmadha\Desktop\Output\APJ.mp4.4
wrote C:\Users\Narmadha\Desktop\Output\APJ.mp4.5
wrote C:\Users\Narmadha\Desktop\Output\APJ.mp4.6
wrote C:\Users\Narmadha\Desktop\Output\APJ.mp4.7
wrote C:\Users\Narmadha\Desktop\Output\APJ.mp4.8
wrote C:\Users\Narmadha\Desktop\Output\APJ.mp4.9
```

Figure 3.1



APJ.mp4
APJ.mp4.0
APJ.mp4.1
APJ.mp4.2
APJ.mp4.3
APJ.mp4.4
APJ.mp4.5
APJ.mp4.6
APJ.mp4.7
APJ.mp4.8

Figure 5.2 Result of Reed-Hash Solomon Encoder.

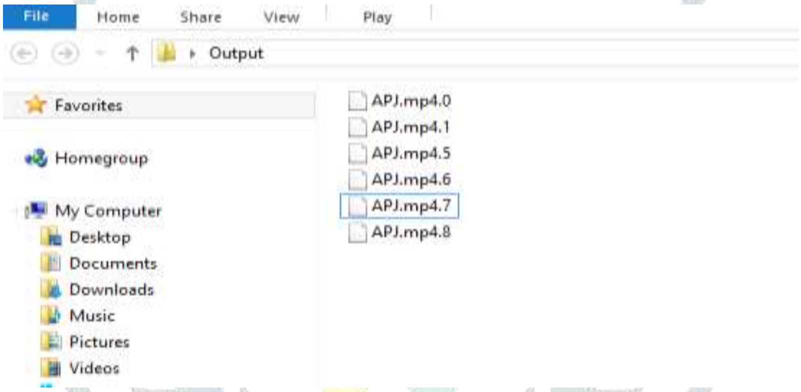


□	APJ.mp4.0	Mar 26, 2019 2:07:38 PM GMT+0530	2.2 MB	Standard
□	APJ.mp4.1	Mar 26, 2019 2:07:49 PM GMT+0530	2.2 MB	Standard
□	APJ.mp4.2	Mar 26, 2019 2:08:01 PM GMT+0530	2.2 MB	Standard
□	APJ.mp4.3	Mar 26, 2019 2:08:13 PM GMT+0530	2.2 MB	Standard
□	APJ.mp4.4	Mar 26, 2019 2:08:25 PM GMT+0530	2.2 MB	Standard
□	APJ.mp4.5	Mar 26, 2019 2:08:37 PM GMT+0530	2.2 MB	Standard
□	APJ.mp4.6	Mar 26, 2019 2:08:50 PM GMT+0530	2.2 MB	Standard
		Mar 26, 2019 2:09:02 PM		

Figure 5.3 Result of Cloud Storage.

```
Do you Wish to Encode File or Decode File
Press 1 for Encode File and 2 for Decode File and 3 to exit
2
Decrypted number matches with the given number!
Decrypted text:
33
Enter Number Of Splittings of original file
6
Enter Number Of Parity Data
4
Read C:\Users\Narmadha\Desktop\Output\APJ.mp4.0
Read C:\Users\Narmadha\Desktop\Output\APJ.mp4.1
Read C:\Users\Narmadha\Desktop\Output\APJ.mp4.2
Read C:\Users\Narmadha\Desktop\Output\APJ.mp4.3
Read C:\Users\Narmadha\Desktop\Output\APJ.mp4.4
Read C:\Users\Narmadha\Desktop\Output\APJ.mp4.5
Read C:\Users\Narmadha\Desktop\Output\APJ.mp4.6
Read C:\Users\Narmadha\Desktop\Output\APJ.mp4.7
Read C:\Users\Narmadha\Desktop\Output\APJ.mp4.8
Read C:\Users\Narmadha\Desktop\Output\APJ.mp4.9
Wrote C:\Users\Narmadha\Desktop\Output\APJ.mp4.decoded
Do you Wish to Encode File or Decode File
Press 1 for Encode File and 2 for Decode File and 3 to exit
3
BUILD SUCCESSFUL (total time: 1 minute 15 seconds)
```

Figure 3.4



APJ.mp4.0
APJ.mp4.1
APJ.mp4.5
APJ.mp4.6
APJ.mp4.7
APJ.mp4.8

Figure 5.5 Result of Data loss.

VII. Conclusion And Future Work

In this work, we studied on how to optimally update the key for cloud storage auditing. We proposed the cloud storage auditing protocol with optimal key updates. Key updates are outsourced to the 2FA and are transparent for the client. In addition, the 2FA only sees the encrypted version of the client secret key, then downloading them from the 2FA. And also, we focussed on how to solve the problem of privacy protection in cloud storage, the proposition of Temporal Reed Hash-Solomon algorithm. The proposed techniques used cloud storage and protected the privacy of data. Cloud Computing has applied to the broader geographical distributions, higher real-time, low latency and sensitive to delay. The proposed Temporal Reed Hash-Solomon code algorithm was designed to divide data into different parts with completely efficient encoding and decoding. The experimented results showed that our proposed schemes were efficient. Through the experiment test, the system completed encoding and decoding without influence of the cloud storage efficiency. Through the experimental evaluation, the feasibility of our scheme has been validated, which was really a powerful supplement to existing cloud storage schemes.

The proposed method has been successful in effectively utilising the storage space allocated to the users by CSP. Our method also supports many users to share a common memory space which helps to save more space than the existing methods. It also prevents the confidentiality of users by checking their proof of ownership. It does not allow unauthorised users to access other files stored in the cloud by random key generation method thus providing more security to the data files. The work can be incorporated in a real time environment for improving the efficiency and flexibility in auditing the files allowing the user to authenticate their file content stored in the cloud server without downloading the entire file. The obfuscation method is another way of providing security which can be implemented in order to improve the security of the user file. Integration of obfuscation techniques with the encryption technique will improve confidentiality. The system can protect the data in the cloud storage from insiders as well as outsiders attack.

References

[1] Agarkhed, J., Anjum, S. (2018) ‘A Dual Secret Sharing Scheme in Cloud for Data Acquisition’, 2018 International Conference on Information, Communication, Engineering and Technology (ICICET), Pune, pp. 1-4.

[2] Apolinario, F., Pardal, M. and Correia, M. (2018) 'S-Audit: Efficient Data Integrity Verification for Cloud Storage', 2018 17th IEEE International Conference on Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), New York, NY, pp. 465-474.

[3] Fowley, F., Pahl, C., Jamshidi, P., Fang, D. X. Liu (2018) 'Classification and Comparison Framework for Cloud Service Brokerage Architectures', IEEE Transactions on Cloud Computing, Vol. 6, No. 2, pp. 358-371.

[4] Fu, Z., Huang, F., Ren, K., Weng, J. Wang, C. (2017) 'Privacy- preserving Smart Semantic Search based on Conceptual Graphs over Encrypted Outsourced Data', IEEE Transactions Inference Forensics Security, Vol. 12, No. 8, pp. 1874-1884.

[5] Fu, Z., Huang, F., Sun, X., Vasilakos, A.C. Yang (2018) 'Enabling Semantic Search based on Conceptual Graphs over Encrypted Outsourced Data', IEEE Transactions on Services Computing, Vol.44, No. 5, pp. 672-832.

[6] Fu, Z., Ren, K., Shu, J., Sun, X. Huang, F. (2016) 'Enabling Personalized Search over Encrypted Outsourced Data with Efficiency Improvement', IEEE Transactions Parallel Distributed System, Vol. 27, No. 9, pp. 2546-2559.

[7] Fu, Z., Wu, X., Guan, C., Sun, X. Ren, K. (2016) 'Toward Efficient Multi Keyword Fuzzy Search over Encrypted Outsourced Data with Accuracy improvement', IEEE Transactions Inference Forensics Security, Vol. 11, No. 12, pp. 2706-2716.

[8] Hahn, C., Kwon, H., Hur, J. (2018) 'Toward Trustworthy in Delegation: Verifiable Outsourced Decryption with Tamper-Resistance Public Cloud Storage', IEEE Transactions on Cloud Computing, Vol. 4, No. 3, pp. 2159-6190.

[9] He, D., Kumar, N., Khan, M. K., Wang, L. Shen, J. (2018) 'Efficient Privacy-Aware Authentication Scheme for Mobile Cloud Computing Services', IEEE Systems Journal, Vol. 12, No. 2, pp. 1621-1631.

[10] Hsu, C., Lu, C. Pei, S. (2012) 'Image Feature Extraction in Encrypted Domain With Privacy-Preserving SIFT', IEEE Transactions on Image Processing, Vol. 21, No. 11, pp. 4593-4607.

[11] Keke Gai, Meikang Qiu, Meiqin Liu (2018) 'Privacy-Preserving Access Control Using Dynamic Programming in Fog Computing', IEEE Transactions on Cloud Computing, Vol. 2, No. 17, pp. 459-467.

[12] Krithikashree, L., Manisha, S. Sujithra, M. (2018) 'Audit Cloud: Ensuring Data Integrity for Mobile Devices in Cloud Storage', 2018, 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bangalore, pp. 1-5.

[13] Li, X., Yuan, J., Ma, H. Yao, W. (2018) 'Fast and Parallel Trust Computing Scheme Based on Big Data Analysis for Collaboration Cloud Service', IEEE Transactions on Information Forensics and Security, Vol. 13, No. 8, pp. 1917-1931.

[14] Lin, C., Bi, Y., Han, G., Yang, J., Zhao, H. Liu, Z. (2018) 'Scheduling for Time Constrained Big-File Transfer Over Multiple Paths in Cloud Computing', IEEE Transactions on Emerging Topics in Computational Intelligence, Vol. 2, No. 1, pp. 25-40.

[15] Lin, L., Liu, T., Li, S., Sarathchandra Magurawalage, C. M., and Tu, S. (2018) 'PriGuarder: A Privacy-Aware Access Control Approach Based on Attribute Fuzzy Grouping in Cloud Environments', IEEE Access, Vol. 6, pp. 1882-1893.

[16] Mantzoukas, K., Kloukinas, C. and Spanoudakis, G. (2018) 'Monitoring Data Integrity in Big Data Analytics Services', 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), San Francisco, CA, pp. 904-907.

[17] Modi, F. M., Desai, M. R. and Soni, D. R. (2018) 'A Third Party Audit Mechanism for Cloud Based Storage Using File Versioning and Change Tracking Mechanism', 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, pp. 521-523.

[18] Rathnayake, R. M. P. H. K., Karunarathne, M. S., Nafi, N. S. Gregory, M. A. (2018) 'Cloud Enabled Solution for Privacy Concerns in Internet of Medical Things', 2018, 28th International Telecommunication Networks and Applications Conference (ITNAC), Sydney, NSW, pp. 1-4.

[19] Shen, Liu, D., Shen, J., Liu, Q. and Sun, X. (2017) 'A Secure Cloud Assisted Urban Data Sharing Framework for Ubiquitous Cities', Pervasive Mobile Computing, Vol. 41, No. 2, pp. 219-230.

[20] Shen, J., Zhou, T., He, D., Zhang, Y., Sun, X. and Xiang, Y. (2018) 'Block Design - based Key Agreement for Group Data Sharing in Cloud Computing', IEEE Transactions on Dependable and Secure Computing, Vol.21, No. 3, pp. 143-546.

[21] Truong, H. and Karan, M. (2018) 'Analytics of Performance and Data Quality for Mobile Edge Cloud Applications', 2018, IEEE 11th International Conference on Cloud Computing (CLOUD), San Francisco, CA, pp. 660-667.

[22] Wang, X., Wang, L., Li, Y. Gai, K. (2018) 'Privacy-Aware Efficient Fine-Grained Data Access Control in Internet of Medical Things Based Fog Computing', IEEE Access, Vol. 6, pp. 47657-47665.

[23] Wang, F., Xu, L. Gao, W. (2018) 'SCLPV: Secure Certificate less Public Verification for Cloud-Based Cyber-Physical-Social Systems Against Malicious Auditors', IEEE Transactions on Computational Social Systems, Vol. 5, No. 3, pp. 854-857.

[24] Wang, T., Zhou, J., Chen, X., Wang, G., Liu, A.and Liu, Y. (2018) 'A Three-Layer Privacy Preserving Cloud Storage Scheme based on Computational Intelligence in Fog Computing', IEEE Transactions on Emerging Topics in Computational Intelligence, Vol. 2, No. 1, pp. 3-12.

[25] Xia, Z., Wang, X., Sun, X. Wang, Q. (2016) 'A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data', IEEE Transactions Parallel Distributed System, Vol. 27, No. 2, pp. 340-352.

[26] Xia, Z., Wang, X., Zhang, L., Qin, Z., Sun, X. Ren, K. (2016) 'A Privacy Preserving and Copy-Deterrence Content-based Image Retrieval Scheme in Cloud Computing', IEEE Transactions Inference Forensics Security, Vol. 11, No. 11, pp. 2594-2608.

[27] Xia, Z., Zhu, Y., Sun, X., Qin, Z. Ren, K. (2018) 'Towards Privacy-Preserving Content-Based Image Retrieval in Cloud Computing', IEEE Transactions on Cloud Computing, Vol. 6, No. 1, pp. 276-286.

[28] Yen, I., Bastani, F., Solanki, N. Huang, Y. (2018) 'Trustworthy Computing in the Dynamic IoT Cloud', IEEE International Conference on Information Reuse and Integration (IRI), Salt Lake City, UT, pp. 411-418.

[29] Yu, J., Ren, K. Wang, C. (2016) 'Enabling Cloud Storage Auditing With Verifiable Outsourcing of Key Updates', IEEE Transactions on Information Forensics and Security, Vol. 11, No. 6, pp. 1362-1375.

[30] Zhao, N., Zhang, Y., Xiong, K. Liu, T. (2018) 'On Massive Data Storage Security in Cloud Computing with Raptor Q codes', 14th IEEE International Conference on Signal Processing (ICSP), Beijing, China, pp. 758-761.