ISSN:0975-9646

Towards Secure Cloud Infrastructure with Cloud Validation

Sathiya Moorthy S [1] and Chandrasekar C

[1] *Research Scholar, Manonmaniam Sundaranar University, TamilNadu, India*

[2] *Department of Computer Science, Periyar University, Salem, India*

Abstract-Cloud computing is an emerging standard with various computing ideas and technologies for the Internet, supporting cost-effective organization applications and IT infrastructure. The acceptance of cloud computing demand is gradually increasing over the years and the development of the global market is regularly rising. Security is the major issue often arising and is questioned by users that results in migration of data from one cloud provider to the other. Various cloud storage systems survive presently, but many of existing fails to ensure security during data migration. This lack of security support holds a major problem for the acceptance of cloud computing, more specifically for organization and IT infrastructure. In order to addresses security issue during data migration, this paper presents Cloud-validation, a secure storage system exclusively meant for the cloud. The proposed Cloud-validation based Flexible Distributed (CFD) scheme detects security breaches like integrity, confidentiality. Hence, cloud-validation based flexible distributed scheme efforts wide possibility to allow users to detect and prove cloud misbehavior based on cryptographic implements. The cryptographic implements involve simulcast encryption and signature looping. Simulcast encryption allows a migration to encrypt message to a random subset of a set of users. Signature looping is the process of generating consecutive signatures from an initial signature and a secret head signature. The exchange of the cloud-validations secures the process of data migration. The validations are key elements that permit the users to verify cloud misbehavior and the cloud service providers to defend against the violations. Experimental evaluation proves CFD is high level of access control with cloud-validation in terms of data integrity, confidentiality and validity.

Keywords: **Data Integrity, Migration, validity, cloud-validation and cloud misbehavior**

I. INTRODUCTION

The growth of cloud computing is increasingly tremendous in recent times and gained huge interest because of significant merits and supports. Organizations attempts to reduce computing costs and storage space. Hence, the cloud computing promises to take the organization needs to an innovative level and facilitate them to additionally reduce their costs through better utilization, reduced management, minimizing the infrastructure cost and through quicker deployment cycles. Cloud computing infrastructure aids organizations to attain more well-organized use of IT hardware and software resources.

Data security is another most important research area in cloud computing. At the same time, the service providers generally are not facilitated with access permission for the security system of data centers. More frequently, service providers depend on the cloud infrastructure to obtain a complete data security. Even though, the infrastructure is not completely accomplished the service providers particularly implement for a remotely virtual private cloud security. The security towards the cloud infrastructure for secure data migration is decided based on the confidentiality, integrity and serializability.

Confidentiality defines the secure file access and transfer with the cloud-validation. Data integrity determines whether the security level of applications are corrupted or not for migration. Serializability is the key correctness criterion for the concurrent data migration between cloud infrastructures. Cloud-validation achieves confidentiality using cryptographic implements, integrity through cloud-validation and serializability based on the both cryptographic and cloud-validation.

Some of the existing cloud computing mechanism like [4] builds cloud infrastructure based on virtualization techniques and distributed computing to maintain cost-efficient usage of hardware software resources, highlighting resource lifetime and on demand security services. It provides a unified resource allocation framework for cloud infrastructure. The cloud mapping problem or mixed integer programming (MIP) problem are addressed to reduce the cost-efficiency of the resource with the aid of virtualization. In addition, the method incorporates a heuristic methodology to address the problem. Similarly in [7] presented two useful designs for computing resources namely Reservation plan and On-demand plan.

In reservation plan, cloud consumers pay the advance in prior providing a cheaper means of usage. The optimal advance reservation is achieved based on a Robust Cloud Resource Provisioning (RCRP) algorithm. Certain amount of definite reservations like requirements, money, resource utilization and users cost is handled potentially in RCRP. RCRP was designed to minimize the complete usage cost by considering the reservations, considering the virtualization and virtual machine placement. Both the techniques fail in verifying the confidentiality and integrity. Advanced Cloud Protection System (ACPS) in [11] ensures increased security to cloud resources with violations against confidentiality.

On the other hand, the data confidentiality and integrity is verified in the [8] through trusted cloud computing platform (TCCP). TCCP supports infrastructure as a service providers such as Amazon S3 to offer a closed box effective platform that promises confidential implementation of guest virtual machines. Moreover, TCCP provides a remote validation that ensures the stability of the security service on launching virtual machine (VM).

The virtual machines are remote validation that normally needs a trusted platform module (TPM) to produce non-forgeable system review. During verification state, the encryption uses TPM private key act as the proof of system towards security. But in a cloud like virtualized environment using remote validation directly, is inadequate resulting in wide security breaches.

Storing sensitive data in cloud storage faces severe security hurdles. The cloud probably exposes sensitive data, alter the data or send back incoherent data to different users. The inconsistent event occurrence is due to error bugs, operator faults, collisions and misleads. Moreover, malicious security breaches are more crucial to detect or more harmful than unintentional ones like exterior attackers and interior attackers. Even though the guaranteeing practices of previous implementers and benefits of cloud storage is possible, issues of dependence cause a major hurdle to broad adoption. Present cloud storage services in [5] like Amazon's S3, Google's BigTable, and so on provides security services only in terms of availability but fails in supporting data integrity, confidentiality and serializability.

In most previous work [9, 12] the server holds certain set of un-trusted remote machines that provides un-trusted service. In addition, next issue lies in scalability as most networks needs to sustain the lifetime guaranteeing the cloud data. Organizations are important users of the cloud. In [10], the author analyzed the performance of cloud computing services for evaluating the workloads among different users. The quantification was made in the presence of real time scientific computing workloads for Many-Task Computing (MTC) users, where the users deployed loosely coupled applications consisting of many tasks to achieve their significant goals. As organization involves many staff that needs highly scalable access control and a large amounts of data. Hence, the demand of secure cloud infrastructure is more potential for an organization.

Cloud computing is making a greater level of impact in the IT industry by providing them to access to their infrastructure and services to be used on the basis of subscription. In [12], a framework to measure the quality and prioritize services in cloud was presented. The framework provided a greater impact and resulted in healthier competition between the providers of cloud in order to satisfy their Service Level Agreement (SLA) and finally improve their QoS.

These system offered public-key cryptosystems to generate constant-size cipher texts such that well-organized allocation of decryption rights for any set of ciphertexts was possible. However, the encryption and decryption part still needs enhancements. In this paper we propose cloud-validation based flexible distributed (CFD) scheme with the objective of providing secure cloud infrastructure and detect security to provide significant security service level.

To this end, we design a two-fold scheme in cloud-validation to enhance the level of security during data migration. CFD provides the storage of larger content in the cloud with permissible authentication. However, like most existing schemes where data owner performs the function of authentication and data access, CFD only performs the membership modifications by auditing and definitely not participating during data access. Simulation results shows that

CFD achieves on average 20-25% validity using advanced hash function and is effective at providing security to data owners and users, achieving high validity and higher level of data integrity.

II. Related Work

Most existing cloud systems are an excess of cloud storage systems like Amazon S3, etc. But, majority of these systems struggles to ensure security. More cryptographic techniques are useful in the cloud infrastructure. The techniques denote the proofs of data possession or retrievability (POR) [18] that permits a server to attest to the proprietor of a data in order to store and retrieve file. In addition HAIL [17] permits a shared group of servers to verify data integrity and retrievability of the data at cloud. Such works are definitely purposeful for verifying integrity of outsourced data. But these methods are inadequate as an aggregate complete cloud systems solution.

The reason is that proper retrieval of data on cloud is not sure in providing an exact data upon a request. In case of user request to verify data is corrupted or not, existing techniques uncheck while Cloud-validation successfully checks based on the block of file. Moreover, POR and HAIL is too expensive in order to perform each user request of cloud data and fails in updating resulting in poor scalability. Moreover, these schemes nonsupport serializability and originality with no access control to fetch data from a cloud.

More current works like [6] [16] are related to efficient secure and reliable cloud storage. In [13], data partitioning scheme was used for implementing the aspects related to the security and also addressed the issues related to data partitioning involving the roots of a polynomial in certain areas. In [15], the author discussed the Multi Agent System architecture for cloud using the data encoding mechanism to increase the integrity of Data centers or Cloud Data Storages. The MAS in cloud environment designed an architecture that provided with increased level of integrity of the data present at data centers.

In comparison to these systems, cloud-validation contributes more advantages. Existing cloud storage systems do not support these features, not due to demerits in the design, but due to design made for different reason in cloud setting. Mostly these systems are designed for purpose of official storage dependence on certain remote un-trusted and unaccountable servers. Furthermore, these systems are unable to adopt secure data migration, demanding the need for a better design in cloud infrastructure.

Key-Aggregate Cryptosystem for data sharing in cloud storage as described in [1] supports flexible delegation but holds predefined bound of maximum cipher text classes. Highly decentralized information accountability framework in [2] stay track of the definite usage of the users' data in the cloud. However, the system is unable to leverage the notion of a security. HABE scheme in [3] achieves fine-grained access control and also full delegation in performance but not very expressive in maintaining the security.

With the aforementioned methods and techniques reviewed, the proposed CFD scheme addresses the problem of supporting secure and reliable data migration in cloud infrastructure. The paper is organized as follows: In the next section, certain related work in cloud infrastructure with reference to storage is discussed. In Section 3, the access control in data migration with cloud validation framework is elaborated in detail with the help of an architecture diagram. Performance and security evaluations are conducted in Section 4. Finally, section 5 concludes the proposed work.

III. Research Objectives

In order to enhance level of security during data migration, cloud-validation employed as a proof of verification using the two-fold scheme for flexible distributed system. The two-fold scheme in cloud-validation operates in a cloud storage which forwards attributes like Segment_ID, User_ID and data content using two functions namely, get and put functions. The two-fold here consists of the representation of Segment_ID in the cloud and the representation of User_ID in the specific segment. The attribute Segment_ID (Segment1, Segment2,,..,Segmentn) denotes the segment on the cloud whereas the contents in the segment is identified using the User_ID (U11, U12,…U1m, U21, U22,….,U2m, ….Un1, Un2,….., Unm) as illustrated in figure 1

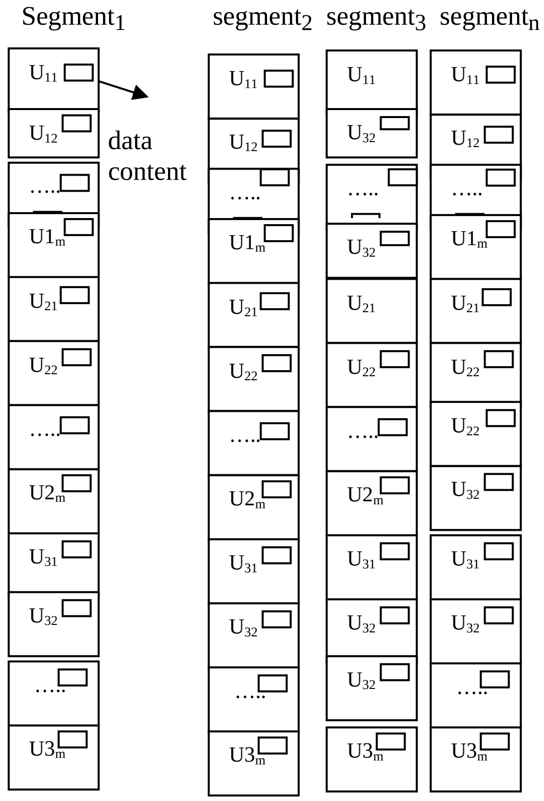

Fig 1 Representation of Segment ID and User ID with user revocation being performed for user User_ID (U22) for Segment_ID (Segment1)

The content of the segment to be read is performed using the command 'get' and the content of the segment to be written is performed with the help of 'put' as illustrated below.

$$\left.\begin{array}{ll} \text{Get(Segment\_ID, User\_ID)} & (1) \\ \text{Put (Segment\_ID, User\_ID, Data\_Content)} & (2) \end{array}\right\}$$

While existing schemes utilized un-trusted file systems, in contrast the proposed cloud-validation utilized the segment id (Segment_ID) along with the user id (User_ID) for performing validation in order to store the data content (Data_Content) in the cloud. This wide use of each 'get' and 'put' command increases the applicability and extend the lifetime of the cloud.

The main objective behind the cloud-validation is to store large content in the cloud with permissible authentication. As the cloud performs huge task, simultaneously validation is performed during the data migration with valid access permission. The task of data owner is to perform membership modifications by auditing (i.e., which user belongs to which segment) and certainly not participating during data access. The architecture of proposed CFD is outlined below in figure 2 (Annexure – A).

Figure 1 (Annexure –A) given above describes the architecture diagram of flexible cloud infrastructure with cloud-validation for secure data migration. The flexible cloud infrastructure with cloud-validation for secure data migration consists of two phases.

The first phase in CFD is the performance of access control in data migration. Here, the users of the data are provided either with read or write access to data on the cloud. For example, users are employees if data owner is an organization. The data owner does not directly interact with the cloud during access. Moreover, with the two commands, 'get' and 'put', the availability of services is ensured even in the absence of data owner. The process of key allocation in the first phase is made innovative incorporating simulcast encryption and signature looping to the cloud. Hence the data owner needs only to change the single segment whenever a user revocation has to be performed instead of changing entire segments wherever the user is present. As illustrated in figure the user revocation for U22 in Segment1 has been performed whereas the U22 in Segment2, Segment3,…,Segmentn remains undisturbed. In order to ensure security, the access controls of the segments running in parallel are restricted.

The second phase involved in CFD is the cloud-validation process which is performed in such a way that the exchanges are made between the data owner, users and cloud service provider in order to provide security during cloud storage access in addition with the use of get or put attributes. As a result, the validation violates the misbehavior, provides security to data owners and users and finally, makes cloud more flexible.

Access Control In Data Migration

Access controls are managed only by data owner that in a larger way restrict the permission to users with the permitted access types consisting of read/write. Each segment possess an access control records (ACR) elaborating the users access permission to the block. The data owner and the cloud service provider holds known public-key as in existing providers.

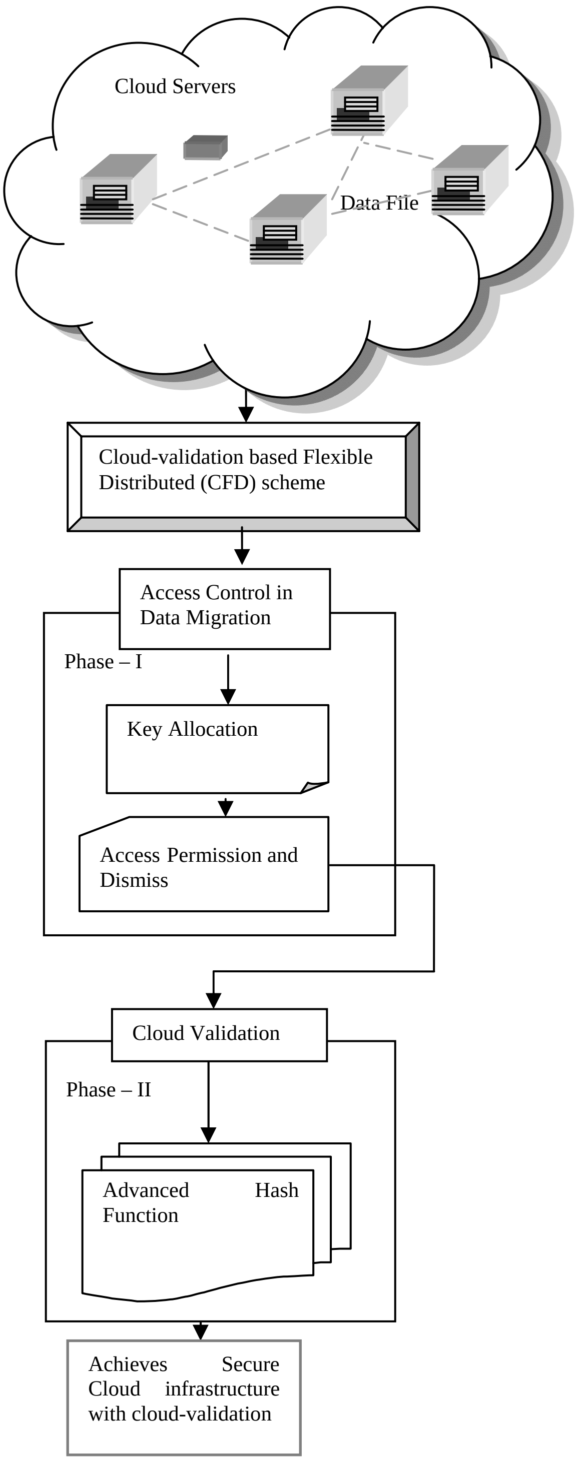
Cloud Servers
Data File
Cloud-validation based Flexible Distributed (CFD) scheme
Access Control in Data Migration
Phase – I
Key Allocation
Access Permission and Dismiss
Cloud Validation
Phase – II
Advanced Hash Function
Achieves Secure Cloud infrastructure with cloud-validation

Fig 2 Architecture Diagram of Flexible Cloud Infrastructure with Cloud-validation for Secure Data Migration

Consider a user is newly entered or removed from the ACR then Respective access permission is also changed. Existing un-trusted cloud access controls rely on data owner to perform the action of user's addition and removal which incurs additional cost and also a time consuming process whereas the cloud-validation using CFD implements cryptographic method like simulcast encryption and signature looping. Simulcast encryption allows a migration to encrypt message to a random subset of a set of users. At the same time, only users in the segment are able to decrypt the transmitted message. Encrypt generates a cipher text of size.

$$\textbf{Encrypt}(\textbf{Message}) = \textbf{Ciphertext}\left[\,\mathbf{O}\,(\sqrt{\textbf{Total no. of users in the segment}})\right] \qquad \textbf{(3)}$$

Signature looping is the process of generating consecutive signatures from an initial signature and a secret signature. The owner of the secret signature holds the permission to generate next consecutive signature. A group of all segments with same ACR is termed as segment class. Whenever a user changes the ACR, then the segment moves to different class. The layout of data segment and segment class signature is figure 3

The figure 3 depicts the segment class with signatures. S denotes the read access signature S_L is the login signature and S_V is the validation key. $E_R()$ denote simulcast encryption to the authorized users in the ACR of a segment class. The segment report represents the number for each segment which is incremented by one on every update with the segment.

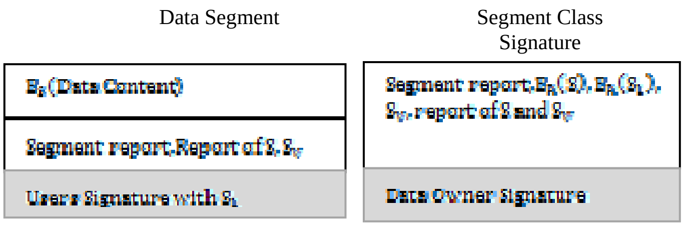
Data Segment
Segment Class Signature
E_R(Data Content)
Segment report, Report of S, S_V
Users Signature with S_L
Segment report, $E_R(S)$, $E_R(S_L)$, S_V, report of S and S_V
Data Owner Signature

Fig 3 Layout of segment formats in a class and of the key segment

In order to avoid unauthorized reads, the data stored in the cloud is encrypted using the stream cipher. The signature of stream cipher is denoted as the read signature. Therefore, a user with read access holds the signature for decryption and has the access to the corresponding data content. Users in the same segment use the same read signature. In a similar way, the write access is performed using a public signature validate and a private signature validate. Everyone knows the public signature whereas the private signature is known only to the users with write access permissions by the ACR.

KEY ALLOCATION

The purpose of key allocation in Cloud Proof based Flexible Distributed (CFD) scheme is to guarantee each user to access the respective segment they are supposed to access. For each segment class, the data owner places single segment termed segment class signature on the cloud containing signature details for that class. Only the data owner holds the rights to modify the block class signature. As mentioned before, each segment shares the same ACR, denoting that each key segment match to a specific ACR as detailed in figure 3. Using simulcast encryption, the data owner encrypts/decrypts the read/write access signature in order to provide decryption read access signature only to users with read/write sets in the ACR. Therefore, only users and read/write sets produce update signature for segments in the respective class.

Access Permission And Dismiss

The data owner attempts to dismiss the access of certain user by discarding them from certain distinct segment ACRs. Moreover, in order to ensure the dismissed user is restricted from data access, two options such as direct dismiss and indirect dismiss is adopted. Direct dismiss indicate that the dismissed user do not have the access to any piece of data from the time of dismissal whereas indirect dismiss denote that the dismissed user do not have the access to any data segments that are updated after removing.

If a segment's ACR changes then direct dismiss is performed in order to move the segment to new segment class signature with new ACR and immediately re-encrypted with new signature segment. On the other hand, if a group's membership changes then indirect dismiss is performed instead of direct dismiss to avoid the malicious data users occurrence due to re-encryption of all segments. The process of indirect dismiss involves signature looping, the data owner loops the signatures then forward to a new report for each of the class corresponding to the affected ACRs. Hence, the segments with ACRs are later re-encrypted indicating that membership changes is independent of the data in the segment class. During block access the user's checks whether the report of the read access key in the segment class signature is larger than the report of the signature with which the current block is prior encrypted. If the read access signature report is found to be larger than signature report then user re-encrypts the segment with new key. As a result, the partition of cloud storage into segment class facilitate the process of dismiss more simple.

Cloud-Validation

The exchange of the validations secures the process of data migration. The validations are key elements that permit the users to verify cloud misbehavior and the cloud service providers to defend against the violations. During process of get ( ) attribute, cloud service provider provides with the users a cloud get validation. Similarly, a users put validation is generated at the time of put ( ) attribute. Naturally, the role of validation is to validate the actions of each user. The validation attached with get indicates the permission to read data. The validation attached with put indicates the permission to write data in turn cloud response with a cloud put validation

Validation Framwork

File validation in Cloud Proof based Flexible Distributed (CFD) scheme is ensured using Advanced Hash Function (AHF). Each validation comprise of combining several data fields that are then hashed using advanced hash function and signed. A unique signature is used as a proof of verification. The figure 4 given below illustrates the framework of validations.

Advanced Hash Function and signed by cloud

Cloud get()	Segment_ ID	Segment Signature	Segment report No.	Segment Hash	User_ ID

Advanced Hash Function and signed with group signature

Cloud put()	Segment _ID	Segment Signature	Updated report No.	New hash	User _ID

Figure 4 Framework of Cloud-Validations

Figure 4 (Annexure –A) illustrates the framework of cloud-validation performed in CloudProof based Flexible Distributed (CFD) scheme. The first field represents the type of validations to be performed [get ()/put ()] with the Segment_ID representing the number for each segment with the corresponding signature for each segments. The segment hash denotes the advanced hash function to be applied with the new hash denoting the hash value after update. The validation is applicable to both get and put validations where Hash () given below returns the valid hash value.

$$\mathbf{Hash\ (U) = U.\ length\ () + AHF\ (U\ [0]) + AHF\ (U\ [U.\ length\ ()\text{-}1])} \quad \textbf{(4)}$$

Where AHF () is constructed using the advanced hash function algorithm as illustrated below with Hash (U) returning the new has value once the update is performed.

Step 1: Cloud-validation is performed using the framework as illustrated in Figure 4

Step 2: Given the Segment_ID and Unique_ID, the first field is obtained as input and accordingly the put () or get () is performed.

Step 3: If the exchange of validation to be performed is get ()

Step 3.1: Evaluate the frequencies of the Unique_ID

Step 3.2: Determine the number of time with which each Unique_ID appears in the respective Segment_ID

Step 4: Sum the Unique_IDs by summing the frequencies appeared in the respective Segment_ID and then sort them in that order.

Step 5: If the exchange of validation to be performed is put ()

Step 5.1: Evaluate the frequencies of the Unique_ID

Step 5.2: Determine the number of time with which each the Unique_ID appears in the respective Segment_ID

Step 6: Sum the Unique_IDs by summing the frequencies appeared in the respective Segment_ID and then sort them in that order.

The algorithm given above provides the detailed steps involved in the designing of advanced hash function to ensure cloud validation with the help of the framework as illustrated in figure 4. The input obtained are the Segment_ID, User_ID along with the put() or get(). The next step consists of the exchange of validation performed with get() or get(). If the exchange of validation to be performed is get(), then the frequencies of the unique id (Unique_ID) is evaluated. Then the number of time the unique id appears in the respective segment is determined. Finally, a summation is performed and sort in the ascending order. In a similar way, put() is performed to ensure data validation.

IV. PERFORMANCE AND SECURITY EVALUATIONS

The security analysis focuses on the protection against adversary attacks during data migration. CFD scheme is implemented in JAVA and further simulated using CloudSim. Moreover, the efficiency of CFD scheme is evaluated with block access permission and dismiss as well as cloud-validation. A benchmark dataset like SMS Spam Collections dataset from UCI Machine Learning Repository is adopted with publicly available 235 datasets for performance evaluations. Further, the CFD scheme is analyzed with dataset in terms of data integrity, confidentiality and validity in comparison with key-aggregate cryptosystem (KAC) scheme [1], HABE scheme [2] and decentralized information accountability (DIA) framework [3].

1. *Security Measure In Terms Of Data Integrity*

Data integrity refers to the validity of data. Data integrity checks the security bugs or threats and error occurrence when data is migrated from one cloud network to another cloud. The optimal way to minimize threats against data integrity is controlling access to data blocks in cloud through security mechanism.

TABLE I
TABULATION FOR DATA INTEGRITY

No. of Data File (MB)	Data Integrity (%)			
	KAC scheme	HABE scheme	DIA Framework	CFD Scheme
2	36	42	45	50
4	43	46	50	55
6	54	56	59	64
8	48	52	56	62
10	61	64	68	73
12	75	77	81	86
14	70	72	76	82

Based on the above table 1 for security evaluation in terms of data integrity, a graph is depicted figure 4

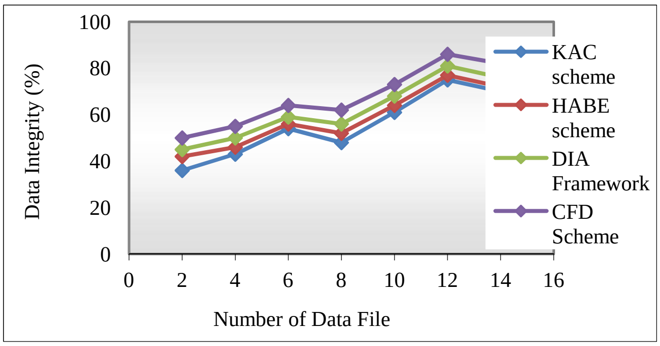

Fig. 4 Number of Data File vs. Data Integrity

Figure 4 describes the level of security in cloud storage in terms of data integrity during data migration. The proposed CFD scheme provides better data integrity of about 17-27 % compared to KAC scheme [1], 11-13% to HABE scheme [2] and 6-11% to DIA framework [3]. This is because the CFD scheme send read/write request along with generated signature derived from AHF of the updated segment using the private signing signature. Moreover a cloud service provider store the generated signature along with the block for further decryption providing with better integrity level for the segment. However, the other existing techniques like [1], [2] and [3] falls into the traditional data integrity protection mechanism with no other verification key proof upon read/write request, resulting in lower data integrity.

2. *Security Measure In Terms Of Confidentiality*

The data migration in a cloud computing system is very open nature due to the system's dynamic property. The confidentiality of a system is ensured by discarding unauthorized access of segments in cloud storage. In data secure systems, the confidentiality characteristic defines authorizations checks to see to that the segments are not accessed by entities with no specific corresponding rights.

TABLE III
TABULATION FOR CONFIDENTIALITY

Block Sizes (KB)	Confidentiality (%)			
	KAC scheme	HABE scheme	DIA Framework	CFD Scheme
50	40	46	44	52
100	32	37	35	40
150	31	35	33	38
200	42	48	44	52
250	36	41	40	45
300	26	29	27	32
350	23	26	24	28

Based on the above table 2 for security measures in terms of confidentiality, a graph is depicted Figure 5

Fig.5: Block Sizes vs. Confidentiality

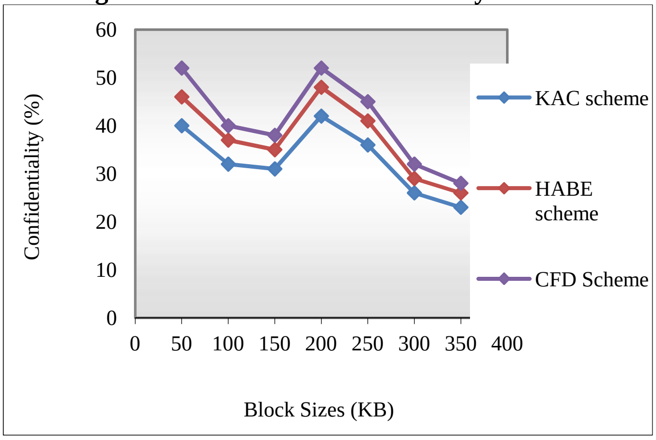

Figure 5 describes the confidentiality of each scheme in providing a secure data migration with access permissions. The proposed CFD scheme address the unauthorized access more potentially than existing schemes with 12-25% KAC scheme [1], 7-13% than HABE scheme [2] and 14-18% than DIA framework [3]. Due to inability of deciding the proper access permission offered to users, the existing schemes struggle in minimizing violations against confidentiality. On the other hand, the proposed CFD scheme involves cloud-validation with cloud's signature to identify the misbehavior. As the proof signature report includes the data segment and signature segment proper detail, the attacker is unable to exhibit the proprietor's signed validation for the key block increasing the confidentiality of data.

3. *VALIDITY IN TERMS OF MISBEHAVIOUR*

Validity is a characteristic of data migration in a cloud schedule. In applicability control of cloud storage and various data migration, distributed data has to be validated for both the data owners and user. Validity is the major accuracy criterion for successive signature looping executions. Validity is derived as

$$\text{Probability}_{\text{read key}}\ (\text{period number, Unique\_ID, read key}) \bmod N = 0 \quad \ldots \quad (5)$$

Where N is an integer included in plain text in the signature metadata by the data owner. If a probability of audit is desired then validity is achieved. The probability of matching the specific period number say particular month, Unique_ID the data content is available and read key attached in the segment with decrypt message decides the accuracy of validity. Audit uses the cloud-validation which the data owner receives from the users.

No. of Data File (MB)	Validity (%)			
	KAC scheme	HABE scheme	DIA Framework	CFD Scheme
5	52	49	56	62
10	37	36	39	46
15	33	31	34	41
20	29	29	30	36
25	41	39	44	51
30	43	41	49	54
35	49	44	53	59

With the determination of validity in the above table 3, a graph (Figure 6 (Annexure –A)) is depicted with respect to the total number of data file.

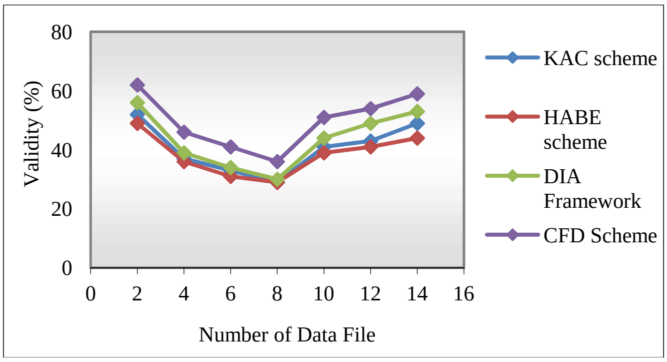

Fig.6 No. of Data File vs. Validity

Figure 6 describes the validity based on the number of data files. Proposed CFD scheme provides better validity of about 20-25% than KAC scheme [1], about 25-30% than HABE scheme [2] and 10-20% than DIA framework [3]. In CFD scheme, the proprietor distinguishes the stored validations based on the segment and sorts them by report number. The data owner demands the cloud to provide cloud-validations for any lost validations and a trustworthy cloud always maintain copies. As a result, CFD scheme checks the validity in terms of secure migration with the exchange of validation. If the cloud does not provide validation, the cloud is penalized for nonconformity during auditing procedure detecting the unmatched access.

Finally, the CFD scheme proves their better efficiency in terms of high data integrity, higher confidentiality and better validity compared to other existing schemes like KAC scheme, HABE scheme and DIA framework.

V. CONCLUSION

The cloud-validation for flexible distributed (CFD) scheme provided a secure users proof during data migration. The principle objective of CFD scheme achieves a large storage in the cloud with authentication permissions with the application of simulcast encryption and signature looping. CFD scheme offers cloud-validation to verify the user's access permission enhancing the security. The wide use of each get and put attribute in the process of access permission with signature allocation increases the applicability and lifetime of the cloud. Each segment in cloud storage holds an access control records (ACR) detailing the users access permission to the requested segment. CFD scheme avoids unauthorized access permission by allowing users to decrypt the encrypted message only using an advanced hash function with private signing key. The partition of cloud storage into segment class facilitate the process of access dismiss more simple. The exchange of the cloud-validations secures the process of data migration. The validation process further permits the users to verify cloud misbehavior and also the cloud service providers to defend against the violations. Through detailed security analysis and wide experiment results justifies that CFD scheme is highly efficient and potential in providing security with cloud-validation during data migration. The performance parameters such as data integrity, confidentiality and validity prove the better performance of CFD scheme compared to other existing schemes.

REFERENCES

[1] S. M. Metev and V. P. Veiko, *Laser Assisted Microtechnology*, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.

[2] Smitha Sundareswaran., Anna C. Squicciarini., and Dan Lin., "Ensuring Distributed Accountability for Data Sharing in the Cloud," IEEE Transactions On Dependable And Secure Computing, vol. 9, no. 4, July/August 2012.

[3] Guojun Wanga., Qin Liu., Jie Wub., Minyi Guo., Science Direct., Elsevier Journal., "Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers," 2011.

[4] C. Papagianni, A. Leivadeas, S. Papavassiliou, V. Maglaris, C. Cervello-Pastor and A. Monje, "On the optimal allocation of virtual resources in cloud computing networks", IEEE Transactions on Computers, Special Section on Optimizing the Cloud, December 2012.

[5] Jayashree Ravi, Zhifeng Yu, Weisong Sh, "A survey on dynamic Web content generation and delivery techniques",

ACM / Elsevier, Science Direct on Network and Computer Applications, September 2009.

[6] T. Sivashakthi, Dr. N Prabakaran, "A Survey on Storage Techniques in Cloud Computing", International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 12, December 2013.

[7] Ms. Lekshmi M Meera, Mrs. Lourdes Mary, "Effective Management of Resource Provisioning Cost in Cloud Computing", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 3, March 2013.

[8] Nuno Santos, Krishna P. Gummadi, and Rodrigo Rodrigues, "Towards Trusted Cloud Computing", ACM, Proceedings conference on Hot topics in cloud computing, 2009.

[9] Zhou Wei, Guillaume Pierre, Chi-Hung Chi, "CloudTPS: Scalable Transactions for Web Applications in the Cloud", IEEE Transactions on Services Computing, Special Issue on Cloud Computing, 2011.

[10] Alexandru Iosup, Simon Ostermann, Nezih Yigitbasi, Radu Prodan, Thomas Fahringer, and Dick Epema, "Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing" IEEE Transactions on Parallel and Distributed Systems, November 2010.

[11] Flavio Lombardi, RobertoDiPietro, "Secure virtualization for cloud computing", Elsevier, Science Direct on Network and Computer Applications, Volume 34, Issue 4, July 2011, Pages 1113-1122.

[12] Saurabh Kumar Garg, Steve Versteeg, Rajkumar Buyya, "A framework for ranking of cloud computing services", Elsevier science Direct on Future Generation Computer Systems, 2013.

[13] Abhishek Parakh, Subhash Kak "Online data storage using implicit security", Elsevier, science Direct on Information Sciences, Volume 179, Issue 19, September 2009, Pages 3323–3331.

[14] B. Sowmya Sri, Mr.S.Vikramphaneendra, "A Secure Way for Data Storage and Forwarding in Cloud", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 9, September 2013.

[15] Satyakshma Rawat, Richa Chowdhary, and Dr. Abhay Bansal, "Data Integrity of Cloud Data Storages (CDSs) in Cloud", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 3, March 2013.

[16] Qian Wang, Cong Wang, Kui Ren, Wenjing Lou, and Jin Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing", IEEE Transactions on Parallel and Distributed Systems, Volume: 22, Issue: 5, May 2011.

[17] Kevin D. Bowers, Ari Juels, and Alina Oprea, "Hail: a high-availability and integrity layer for cloud storage" in ACM Conference on Computer and Communications Security, pp. 187–198. 2009.

[18] Yevgeniy Dodis, Salil Vadhan, and Daniel Wichs, "Proofs of Retrievability via Hardness Amplification," Springer Journal, Computer Science, pp. 109–127, Volume 5444, January 2009.