International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 13 Issue 3 –MARCH 2015.

SECURE PROCESS DATA IN CLOUD STORAGE USING DATA INTEGRITY PROTECTION

Jackulin DuraiRani.A, A.Pushpa Priya, B.Sivaranjani,S.Kavitha

PG Scholar, Department of Computer Science and Engineering, Dr.N.G.P. Institute of Technology, Tamil Nadu, India

PG Scholar, Department of Computer Science and Engineering, Sree Sastha Institute Of Engineering And Technology, Tamil Nadu, India

PG Scholar, Department of Computer Science and Engineering, Dr.N.G.P. Institute of Technology, Tamil Nadu, India

PG Scholar, Department of Computer Science and Engineering, Dr.N.G.P. Institute of Technology, Tamil Nadu, India

Abstract-To secure the data process in cloud storage against duplicity, joining fault tolerance to cloud storage, along with coherent data integrity checking and recovery procedures, becomes critical. Rebuilding (also called repairing) lost encoded fragments from existing encoded fragments provide fault tolerance by striping data across multiple servers occurs less repair traffic than traditional erasure codes during failure recovery. To analyze the problem of checking the integrity of repairing-coded data against corruptions under a real-life cloud storage setting. To implement and evaluate our DIP scheme in a real cloud storage test bed under different parameter choices. It works under the simple assumption of thin-cloud storage and allows different parameters to be fine-tuned for a performance-security trade-off. It is desirable to enable cloud clients to verify the integrity of their outsourced object, in such a case their data have been accidentally corrupted or maliciously compromised by insider/outsider attacks. FMSR-DIP codes preserve fault tolerance and repair traffic saving as in FMSR codes. By combining integrity checking and efficient recovery, FMSR-DIP codes produce a low-cost solution for maintaining data availability in cloud storage. The FMSR-DIP codes, which produces the integrity, fault tolerance, and recovery to cloud storage. Thus the mathematical analysis on the security of FMSR-DIP codes for different parameter choices.

Index Terms: **DIP, FMSR-DIP, PDP, POR.**

I. INTRODUCTION

CLOUD storage offers associate on-demand information outsourcing service model, and is gaining quality as a result of its elasticity and low maintenance value. However, security concerns arise once information storage is outsourced to third party cloud storage suppliers. it is fascinating to change cloud clients to verify the integrity of their outsourced information, in case their information are accidentally corrupted or maliciously compromised by insider or outsider attacks. One major use of cloud storage is long-run deposit, which represents a employment that is written once and barely read. Whereas the hold on information area unit seldom scan, it remains necessary to make sure its integrity for disaster recovery or compliance with legal necessities. Since it is typical to possess a large quantity of archived information, whole-file checking becomes preventive. Proof of irretrievability (POR) and proof of information possession (PDP) have therefore been proposed to verify the integrity of an over sized file by spot checking only a fraction of the file via numerous cryptology primitives. Suppose that we have a tendency to source storage to a server, which could be a storage web site or a cloud-storage supplier. If we detect corruptions in our outsourced information (e.g., when a server crashes or is compromised), then we should always repair the corrupted information and restore the first

information. However, putting all information in a very single server is prone to the single point- of-failure drawback and seller lock-ins. As suggested in a plausible answer is to stripe information across multiple servers. Thus, to repair a failing server, we can 1) scan information from the opposite extant servers, 2) reconstruct the corrupted information of the failing server, and 3) write the reconstructed information to a brand new server. POR and PDP area unit originally projected for the single-server case. MR-PDP and HAIL extend integrity checks to a multi server setting mistreatment replication and erasure writing, respectively. Specifically, erasure writing encompasses a lower storage overhead than replication underneath a similar fault tolerance level. Field measurements show that large-scale storage systems ordinarily expertise disk or sector failures, some of which might lead to permanent information loss. For example, the annualized replacement rate (ARR) for disks in production storage systems is around 2-4 %. Data loss events are found in industrial cloud-storage services. With the exponential growth of deposit data, a little failure rate will imply vital information loss in archival storage. This motivates US to explore high performance recovery thus on cut back the window of vulnerability. Create codes have recently been proposed to attenuate repair traffic (i.e., the number of information being scan from extant servers). In essence, they win this by not reading and reconstructing the entire file throughout repair as in ancient erasure codes, however instead reading a set of chunks smaller than the first file from different surviving servers and reconstructing solely the lost (or corrupted) information chunks. Associate open question is, will we have a tendency to change integrity checks atop create codes

In this paper, we have a tendency to style and implement a sensible knowledge integrity protection (DIP) theme for regenerating-coding based cloud storage. We have a tendency to augment the implementation of functional minimum-storage create (FMSR) codes and construct FMSR-DIP codes, which permit shoppers to verify the authencity of random subsets. FMSR-DIP codes protect the fault tolerance and renovate traffic saving as in FMSR codes. Also, we have a tendency to assume solely a thin-cloud interface, meaning that servers solely got to support customary read and write functionalities. This adds to the immovableness of FMSRDIP codes and permits easy readying normally varieties of storage services. By combining integrity checking and efficient recovery, FMSR-DIP codes offer a inexpensive solution for maintaining knowledge accessibility in cloud storage. In summary, we have a tendency to create the subsequent contributions. We have a tendency to style FMSR-DIP codes that alter integrity protection, fault tolerance, and economical recovery for cloud storage. . We have a tendency to export many tunable parameters from FMSRDIP codes; specified shoppers will create a trade-off between performance and security. . We have a tendency to conduct mathematical analysis on the protection of FMSR-DIP codes for various parameter decisions. . We have a tendency to implement FMSR-DIP codes, and judge their overhead over the present FMSR codes through extensive test bed experiments during a cloud-storage environment. We have a tendency to judge the running times of different basic operations, together with transfer, Check, Download, and Repair, for various parameter choices.

II. RELATED WORKS

We in short summarize the foremost recent and closely connected work here. Additional literature review is found in Section one of the supplementary file, obtainable on-line. We think about the matter of checking the integrity of static knowledge that is typical in long-run depository storage systems. . a serious limitation of the higher than schemes is that they are designed for a single-server setting. If the server is absolutely controlled by Associate in nursing someone, then the higher than schemes will solely give detection of corrupted knowledge, but cannot recover the initial knowledge. This ends up in the look of efficient knowledge checking schemes in an exceedingly many server setting. By striping redundant knowledge across multiple servers, the initial files will still be recovered from a set of servers though some servers are down or compromised. Economical knowledge integrity checking has been planned for various redundancy schemes, like replication, erasure cryptography, and creates cryptography. Specifically, though sub genus Chen et al. additionally thinks about regenerating-coded storage, there are key variations with our work. First, their style extends the single-server compact POR theme by Shacham and Waters. However, such direct adaptation inherits some shortcomings of the single-server theme like an outsized storage overhead, because the quantity of knowledge keep will increase with a more versatile checking coarseness within the theme of. Second, the storage theme of assumes that storage servers have secret writing capabilities for generating encoded data, whereas we have a tendency to think about a thin-

cloud setting, where servers solely got to support normal read and write functionalities or immovableness and ease. The foremost closely related work to ours is HAIL, that stores knowledge via erasure cryptography. As declared in Section one, HAIL operates on a per-file basis and it is nontrivial to directly apply HAIL to regenerating codes. Additionally, our work focuses a lot of on the practical problems, like however totally different parameters will be adjusted for the performance-security trade-off in practical preparation.

III. IMPLEMENTATION

DATA EVALUATION

Data at rest refers to information whereas it is within persistent storage in structured and unstructured forms like databases and file systems. This provides

- All data's are classified as needing protection ought to be encrypted whereas in storage.
- The use of cryptography ought to be supported the sensitively and price of {the information (based upon company data classification requirements).
- Access to keep information taught to be granted on a need-to-know and least privilege basis.
- All access to classified and encrypted information should be compliant with established access management policy.
- In addition to procedural access management, a cryptosystem should support native integrity checking.
- It ought to be attainable to check some encrypted information (one-way hashed passwords) while not jeopardizing the general integrity of the cryptosystem.

Data in transit refers to information whereas it is being transferred from one information repository to a different. Information in transit includes information sent by back-end servers, applications and databases over the network. 2 information repositories will be among an equivalent company network, within the cloud, or a part of fully totally different networks. Information in Transit ought to be managed therefore that:

1. All sensitive information is protected whereas in transit over the wire, air, or the other transport medium.
2. If information supposed to be confidential to any degree is being transmitted unencrypted, secure network communication protocols like IPSec or SSL/TLS ought to be used, particularly once sending information over shared or public networks, together with shared networks of the Cloud Service supplier (CSP). This is often unremarkably observed as 'encrypting the pipe' vs. encrypting the info itself.

There ar many choices for dominant the distribution of cryptography keys for safeguarding information in transit. The CSP ought to provide multiple choices. as an example, cryptography keys may well be distributed to the CSP and each information facility house owners, to the info repository house owners and a 3rd party supplier of such services or to solely the info facility house owners.

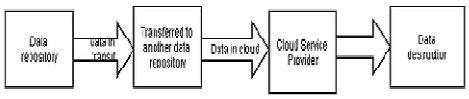

Fig: 4.1 Data Evaluation

Data within the cloud refers to information whereas it is being hold on or processed by a Cloud Service supplier (CSP). Best practices would guarantee that:

1. Constant information classification and science security needs used within a company area unit applied to information hold on or processed by a CSP. It is attainable that handling needs ought to be created even a lot of demanding once data is touched from a "protected" on-premise atmosphere to a public multi-tenant cloud. It is essential that the CSP offer the safety and secret writing protection controls necessary to satisfy those needs.

2. Science security controls might not get replaced by CSP Service Level Agreements.

3. Once information is safely transmitted to a CSP, it\'s hold on during a secure means.

4. The CSP ought to guarantee logical (and if necessary, physical) separation of information supported the owner. For instance, information and science keying material happiness to company a requirement be unbroken break free information happiness to company B.

5. Once information is far from cloud storage, the CSP ought to make sure that it is properly purged, and, all connected science keys the CSP holds or manages should be destroyed in accordance with approved key lifecycle management standards and tips

6. If therefore desired by the user, the CSP ought to permit information to be encrypted during a format

conserving manner. For instance, a sixteen-digit master card variety ought to seem as a string of 16 digits, with every digit painted by 'x'.

7. Secret writing of information should not adversely have an effect on the flexibility of applications to use this information.

The destruction of information implies taking all measures necessary to confirm that data cannot be accessed by anybody in future. The extent to that information destruction processes take away information and keys could rely on the kind of service and also the needs of the info owner. Such needs would so be supported the information owner having each their own data classification methodology and human activity such classification to the CSP.

Some knowledge destruction techniques ensure:

1. Once therefore desired by a certified user, all traces of the information, as well as within the logs, ought to be far away from the system.

2. It will not be potential to achieve any future access to the deleted knowledge.

3. Memory and storage disks area unit overwritten (wiped) before being substitute in use for consecutive client. It is not spare to easily removed pointers. If the information was really encrypted, the key(s) necessary to rewrite the information became the crucial quality to guard, and so destruction of the key(s) ought to answer. Yet, even the cipher text ought to be destroyed.

4. The cytological keys used throughout coding of information will be destroyed. This will be accomplished by overwriting (zeroization of) the media that holds the keys.

SECURING THE CLIENT

Cloud computing services that store encrypted knowledge area unit created doable by a mixture of server and consumer fact parts. Normally there has been a way bigger specialize in server fact parts and, since resources area unit perpetually at a premium, this usually is completed at the expense of consumer facet parts which can be prone to compromise. Purchasers must not be the "forgotten half" of the cloud story. Discussion is building on the problem of shopper devices and also the follow of Bring Your Own Device (BYOD) to business environments. The foremost comprehensive coverage thus integrates consumer and Server as complementary parts of the larger context. , this is often particularly necessary as additional users think about personal consumer devices, like good phones, for accessing cloud services. These devices run the applications that are easy to use, and will address security as associate afterthought. Designers of cloud storage ought to keep this reality in mind and make knowledge protection systems that area unit each suitably secure and stay simple to use with as very little impact on cloud users as possible. Cloud suppliers, vendors and device OEMs ought to provide end-user device security controls and compliance. Purchasers and shopper devices are implemented by native storage encoding. Content must not be rewrites at the cloud edge in order that it to be consumed by devices that can't transfer and decrypt the encrypted content it receives from cloud storage. A backdoor Trojan, keystroke lumberjack, or another sort of malicious software package running on a consumer device will undermine the general security of cloud storage services.

CLIENT-SIDE ENCRYPTION

Encrypting information before it is sent to the cloud storage provides higher levels of security scrutiny to encrypting within the cloud, as a result of it prevents the cloud supplier from seeing or accessing the first information and limiting the harm if information is later compromised at the CSP. It conjointly permits larger user management over the generation and storage of encoding keys. for instance, keys are often hold on during a revolving credit or different hardware tokens that stay within the possession of the user, so providing larger assurance that nobody however the user will decode information hold on within the cloud. Client-side encoding could also be necessary for restrictive compliance, like HIPPA, PCI DSS, and different compliance series. Given this state of maturity of cloud supplier encoding services, encoding of content by the client is a superb mitigation. But it should be remembered that not all cloud customers have the resources to self-provide associate degree encoding capability. Though mature enterprise IT departments could also be ready, several start-ups and tiny and medium size businesses might not.

Endpoint Devices and Applications

Because of their widespread use and universal acceptance as de-facto client applications, web browsers are key elements in providing client-side access to cloud computing services such as data

encryption. However, the open architecture of web browsers, where additional plug-ins and browser helper objects can be installed by users, can introduce security problems. The multitude of available and often required browser plugging and mobile code can create an opportunity for user sessions to be hijacked (for example, through man in the middle attacks) and thereby access to encrypted cloud data can be compromised. Adequate measures should be taken to authenticate users in a secure manner to protect their online session from malicious use. For example, cloud providers and/or vendors should enable multi-factor authentication capability for end-users.

Additionally, mobile device applications increasingly provide a front-end to cloud based services and may be the target of attacks designed to gain access to data and services in the cloud. Care should be taken to ensure that all access points are sufficiently secure to not introduce new vulnerabilities or compromise the security of cloud services, including encryption. Key management is especially important in this area, as keys stored on the mobile application could be exposed if the device is lost.

PROTECTION OF KEYS

Because of their widespread use and universal acceptance as de-facto consumer applications, net browsers area unit key components in providing client-side access to cloud computing services like encoding. However, the open design of net browsers, wherever extra plug-ins and browser helper objects may be put in by users, will introduce security issues. The multitude of obtainable and infrequently needed browser plugging and mobile code will produce a chance for user sessions to be hijacked (for example, through man within the middle attacks) and thereby access to encrypted cloud information may be compromised. Adequate measures ought to be taken to manifest users in a very secure manner to guard their on-line session from malicious use. For instance, cloud suppliers and/or vendors ought to alter multi-factor authentication capability for end-users.

Additionally, mobile device applications more and more offer a front-end to cloud primarily based services and will be the target of attacks designed to achieve access to information and services within the cloud. Care ought to be taken to make sure that everyone access point's area unit sufficiently secure to not introduce new vulnerabilities or compromise the safety of cloud services, together with cryptography. Key management is very necessary during this space, as keys keep on the mobile application might be exposed if the device is lost.

Policy and Enforcement

Security policy is a vital thought once implementing cloud cryptography solutions. Necessities for cryptography to adapt to established security policy do not stop at the on-premise perimeter. The utilization of cryptography through cloud suppliers should effectively extend and support identical security policy necessities as area unit already in use for on-premise process. Addressing policy problems needs acceptable thought on what is encrypted, once it is encrypted, and how. If the client has been in operation while not sound steering and policy documentation supported its own determination of acceptable risk tolerance and expected controls and behaviors, the client has to invest time to know the worth of the info it is on the brink of move to the cloud, confirm its risk craving, and make policy and necessities that it expects the cloud supplier to fulfill. Cryptography is one among the foremost vital and necessary controls for cover of valued information.

As it relates to the utilization of cloud services, the problems of what is encrypted, and when, ought to be self-addressed equally to on-premise ways through the institution of knowledge classification and handling ways. This classification will then be enforced at intervals info workflows mistreatment solutions like information Loss hindrance (DLP) mechanisms that may establish policy violations and so enforce acceptable use of knowledge or a minimum of inform information homeowners that their data was not adequately protected and/or tagged.

The issue of however information is encrypted within the cloud should conjointly align befittingly with on-premise policy. as an example, on-premise policy necessities like those who outline that cryptography algorithms area unit approved to be used, or that confirm what is the minimum complexness needed for passphrase use, should be applied systematically across all on-premise and cloud work out and storage bit points for that organization's valued information. These policies problems and therefore the related to social control measures should touch cowl information because it moves to and from cloud suppliers to make sure that the utilization of cryptography continues to fulfill all mandates for information protection.

Key revocation policy and therefore the associated mechanism area unit vital for all key

management models. Therefore, the key management service ought to change key revocation. As an example, once associate in nursing worker leaves the organization or changes job operate, the key management service has to revoke the key(s) that the worker uses to access that encrypted information that he ought to not have access to.

Data Integrity

A thought that ought to not be unnoticed once implementing cloud services is knowledge integrity. The employment data cryptography alone might not offer enough assurance that encrypted information has not been altered. Files placed into the cloud for storage, significantly if keep over long periods of your time or that are in transit through cloud services, could also be subject to meddling or replacement. Encryption alone cannot sight this. Combining encoding with integrity protections like digital signatures will make sure that knowledge within the cloud remains each non-public and authentic. Wherever on the market, use of trust worthy time ought to be thought of by mistreatment time-stamped signatures on knowledge.

Admin

This module transfer the file and checks whether or not the user file request is attested otherwise user request can neglect. This module show the user request, user details, Network user's mistreatment gird read management. And Admin share the file to given valid user which file key send to their mail Id.

File cacophonic

This module accustomed split the files once admin transfer the files. Files are regenerate to cipher text mistreatment AES rule. As a result of anybody don not hack the info. Then calculate the file size and Split documents supported File size. Then spitted data's are kept in cloud servers.

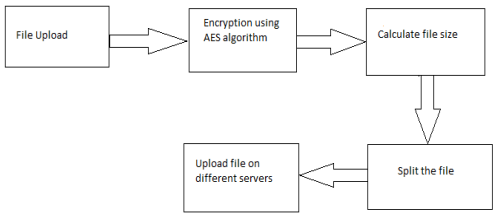

Fig: 4.1 File Splitting

File Sharing

Admin share the file to specified user and they send file key to user mail id. Because Key management is the most complex part of any security system dealing with encryption of data. Encrypted keys are stored in server. User views the files when Admin share the files. Keys should not be changed.

Data Integrity

A consideration that should not be overlooked when implementing cloud services is data integrity. The use of data encryption alone may not provide sufficient assurance that encrypted information has not been altered. Files placed into the cloud for storage, particularly if stored over long periods of time or that are in transit through cloud services, may be subject to tampering or replacement. Encryption alone cannot detect this. Combining data encryption with integrity protections such as digital signatures can ensure that data in the cloud remains both private and authentic.

Securing the Client

Cloud computing services that store encrypted data are made possible by a combination of server and client side components. In general there has been a far greater focus on server side components and, since resources are always at a premium, this typically is done at the expense of client side components which may be vulnerable to compromise. Clients should not be the "forgotten half" of the cloud story. The most comprehensive coverage therefore integrates Client and Server as complementary components of the larger context. This is especially important as more users rely on personal client devices, such as smart phones, for accessing cloud services. These devices run applications that tout ease of use, and may address security as an afterthought. Cloud Providers, vendors should offer end-user device security controls and

compliance "device health" checking. Clients and consumer devices need to implement local storage encryption. Content should not be decrypted at the cloud edge so that it to be consumed by devices that can't download and decrypt the encrypted content it receives from cloud storage. A backdoor Trojan, keystroke logger, or some other type of malicious software running on a client device can undermine the overall security of cloud storage services.

Client-side Encryption

Encrypting data before it is sent to the cloud storage provides higher levels of security comparing to encrypting in the cloud, because it prevents the cloud provider from seeing or accessing the original data and limiting the damage if data is later compromised at the CSP. It also allows greater user control over the generation and storage of encryption keys. For example, keys can be stored in a smart card or other hardware tokens that remain in the possession of the user, thus providing greater assurance that no one but the user can decrypt data stored in the cloud. Client-side encryption may be necessary for regulatory compliance, such as HIPPA, PCI DSS, and other compliance series. Given the current state of maturity of cloud provider encryption services, encryption of content by the customer is an excellent mitigation. However it must be remembered that not all cloud customers have the resources to self-provide an encryption capability. Though mature enterprise IT departments may be able, many start-ups and small and medium size businesses may not.

IV. ALGORITHM DESCRIPTION

FMSR CODE

FMSR codes belong to maximum distance separable (MDS) codes. An MDS code is defined by the parameters (n,k), where k < n. It encodes a file F of size |F| into n pieces of size |F|=k each. An (n,k) - MDS code states that the original file can be reconstructed from any k out of n pieces (i.e., the total size of data required is |F|). An extra feature of FMSR codes is that a specific piece can be reconstructed from data of size less than |F|. FMSR codes are built on regenerating codes, which minimize the repair traffic while preserving the MDS property.

Where each native and code chunk has size $\frac{|F|}{K(n-K)}$ Each code chunk, denoted by Pi (where $i \leq n(n-K)$) , is constructed by a random linear combination of the native chunks. The n(n-k) code chunks are stored in n servers (i.e., n - k code chunks per server), where the k(n-k) code chunks from any k servers can be decoded to reconstruct the original data.

AES ALGORITHM

The Advanced Encryption Standard (AES) is a specification for the encryption of electronic data.AES is based on the Rijndael cipher with different key and block sizes. The algorithm described by AES is a symmetric-key algorithm, meaning the same key is used for both encrypting and decrypting the data. AES is based on a design principle known as a substitution-permutation network, combination of both substitution and permutation, and is fast in both software and hardware. AES operates on a 4×4 column-major order matrix of bytes, termed the state, although some versions of Rijndael have a larger block size and have additional columns in the state.

STEPS

STEP 1: Key Expansions—round keys are derived from the cipher key using Rijndael's key schedule. AES needed a separate 128-bit round key block for each round plus one more.

STEP 2: Initial Round

Add Round Key—each byte of the state is joined together with a block of the round key using bitwise XOR.

STEP 3: Rounds

1. Sub Bytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.

2. Shift Rows—a transposition step where the last three rows of the state are shifted cyclically a certain number of steps.

3. Mix Columns—a mixing operation which operates on the columns of the state, joining the four bytes in each column.

4. Add Round Key

STEP 4: Final Round (no Mix Columns)

The Sub Bytes step

In the Sub Bytes step, every computer memory unit within the state is replaced with its entry

in a very fastened 8-bit search table ,S; bij =S(aij).In the Sub Bytes step, every computer memory unit within the state matrix is substituted with a Sub Byte victimization Associate in Nursing 8-bit substitution box. This operation provides the non one-dimensionality within the cipher.

The Shift Rows step

In the Shift Rows step, bytes in every row of the state are transfer cyclically to the left. The number of places in every computer memory unit is transfer differs for every row. For AES, the first roe is left unchanged. Each computer memory unit of the second row is shifted one to the left. Similarly, the third and fourth rows are shifted by offsets of 2 and 3 severally.

The Mix Columns step

In the combine columns step, each column of the state is increased with a hard and fast polynomial c(x).In the combine Columns step, the four bytes of every column of the state are combined victimization Associate in nursing invertible linear transformation. The Mix Columns perform takes four bytes as input and outputs four bytes, where every input computer memory unit affects all four output bytes.

The Add Round Key step

Within the Add Round Key step, each computer memory unit of state is combined with a computer memory unit of the spherical sub key victimization XOR operation. In the Add Round Key step, the sub key is joined with the state. For each spherical, a sub key springs from the most key victimization Rijindael's key schedule; each sub key is that the same size because the state. The sub key is supplementary by combining every computer memory unit so the state with the corresponding computer memory unit of the sub key victimization bitwise XOR.

V. CONCLUSION AND FUTURE ENHANCEMENT

Given the recognition of outsourcing repository storage to the cloud, it is fascinating to modify purchasers to verify the integrity of their information within the cloud. We have a tendency to style and implement a DIP scheme for the FMSR codes beneath a multi server setting. We construct FMSR-DIP codes that preserve the fault tolerance and reduce traffic using properties of FMSR codes. To know the usefulness of FMSRDIP codes, we analyze the safety strength via mathematical modeling and measure the period of time overhead via test bed experiments. We have a tendency to show however FMSR-DIP codes trade between performance and security beneath completely different parameter settings.

In future work we decided to use advanced encryption standard algorithm is used for encryption and decryption so that the integrity is high compared to the existing system. Also the efficient data recovery in cloud storage and lost data founded easily. To understand the practicality of FMSR-DIP codes, we analyze the security strength via numerical modeling and evaluate the running time overhead via test bed.

VI. REFERENCES

1. B. Chen, R. Curtmola, G. Ateniese, and R. Burns,(2010)"remote data checking for network coding-based distributed storage systems " Proc. ACM Workshop Cloud Computing Security (CCSW '10).
2. T. Schwarz and E. Miller,(2006) "Store, Forget, and Check: UsingAlgebraic Signatures to Check Remotely Administered Storage,"Proc. IEEE 26th Int'l Conf. Distributed Computing Systems,(ICDCS '06).
3. H. Abu-Libdeh, L. Princehouse, and H. Weatherspoon,(2010) "RACS: A Case for Cloud Storage Diversity," Proc. First ACM Symp. Cloud Computing (SoCC '10).
4. H.C.H.Chen and P.P.C. Lee,(2012)"Enabling Data Integrity Protection in Regenerating-Coding-Based Cloud Storage," Proc. IEEE 31stSymp. Reliable Distributed Systems (SRDS '12), 2012.
5. H. Abu-Libdeh L.Princehouse and H.Weatherspoon, "racs: a case for cloud storage diversity"
6. K. Bowers, A. Juels, and A. Oprea,(2009)"HAIL: A High-Availability and Integrity Layer for Cloud Storage," Proc. 16th ACM Conf.Computer and Comm. Security (CCS '09).
7. M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "a view of cloud computing"
8. Jahid S.Mittal and Borisov N.(2011)"EASiER:Encyption based access control in social network eith different revocation,"in ACM ASIACCS,pp.411-415.
9. Wang C.Wang Q.Ren K.Cao N and Lou W,(2012)"Toward Secure and Dependable Storage Services in Cloud Computing",vol.5,no.2,pp.220-232.
10. www.ijarcsse.com/docs/papers/Volume_3/3.../V3I2-0160.pdf
11. Henry C.H. Chen and Patrick P.C. Lee(2014),"Enabling Data Integrity Protection in Regenerating-Coding-Based Cloud Storage: Theory and Implementation "ieee transactions on parallel and distributed systems, vol. 25, no. 2, february 2014 .