Volume No:2, Issue No:1 (June-2016) ISSN No : 2454-423X (Online)

International Journal of Research in Advanced Computer Science Engineering

A Peer Reviewed Open Access International Journal

www.ijracse.com

Secure Erasure Code-Based Storage with Effective Data Sharing

P.Deepa
M.E (CSE),
Dhanalakshmi College of Engineering.
Chennai, Tamilnadu, (India).

Vijayaragavan.P
Associate Professor (CSE),
Dhanalakshmi College of Engineering,
Chennai, Tamilnadu, (India).

ABSTRACT:

High-speed networks and ubiquitous Internet access become available to users for access anywhere at any time. Cloud computing is a concept that treats the resources on the Internet as a unified entity, a cloud. Cloud storage is a model of networked online storage where data is stored in virtualized pools of storage which are generally hosted by third parties. Hosting companies operate large data centers, and people who require their data to be hosted buy or lease storage capacity from them. The data center operators, in the background, virtualized the resources according to the requirements of the customer and expose them as storage pools, which the customers can themselves use to store files or data objects. Physically, the resource may span across multiple servers. Data robustness is a major requirement for storage systems. There have been many proposals of storing data over storage servers. One way to provide data robustness is to replicate a message such that each storage server stores a copy of the message. A decentralized erasure code is suitable for use in a distributed storage system. We construct a secure cloud storage system that supports the function of secure data forwarding by using a threshold proxy re-encryption scheme. The encryption scheme supports decentralized erasure codes over encrypted messages and forwarding operations over encrypted and encoded messages. Our system is highly distributed where storage servers independently encode and forward messages and key servers independently perform partial decryption.

KEY WORDS:

Data confidentiality, data robustness, homomorphism, integrity check, secure decentralized erasure.

INTRODUCTION:

Designing a cloud storage system for robustness, confidentiality and functionality.

The proxy re-encryption scheme supports encoding operations over encrypted messages as well as forwarding operations over encoded and encrypted messages. To provide data robustness is to replicate a message such that each Storage server stores a copy of the message. It is very robust because the message can be retrieved as long as one storage server survives. The number of failure servers is under the tolerance threshold of the erasure code, the message can be recovered from the codeword symbols stored in the available storage servers by the decoding process. This provides a tradeoff between the storage size and the tolerance threshold of failure servers. A decentralized erasure code is an erasure code that independently computes each codeword symbol for a message. A decentralized erasure code is suitable for use in a distributed storage system. A storage server failure is modeled as an erasure error of the stored codeword symbol. We construct a secure cloud storage system that supports the function of secure data forwarding by using a threshold proxy re-encryption scheme. The encryption scheme supports decentralized erasure codes over encrypted messages and forwarding operations over encrypted and encoded messages. Our system is highly distributed where storage servers independently encode and forward messages and key servers independently perform partial decryption.

EXISTING SYSTEM:

In Existing System we use a straightforward integration method. In straightforward integration method Storing data in a third party's cloud system causes serious concern on data confidentiality. In order to provide strong confidentiality for messages in storage servers, a user can encrypt messages by a cryptographic method before applying an erasure code method to encode and store messages. When he wants to use a message, he needs to retrieve the Codeword symbols from storage servers, decode them, and then decrypt them by using cryptographic keys. General encryption schemes protect data confidentiality, but also limit the functionality of the storage system because a few operations are supported over encrypted data.

Volume No:2, Issue No:1 (June-2016) ISSN No : 2454-423X (Online)

International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

A decentralized architecture for storage systems offers good scalability, because a storage server can join or leave without control of a central authority.

LIMITATIONS:

* The user can perform more computation and communication traffic between the user and storage servers is high.
* The user has to manage his cryptographic keys otherwise the security has to be broken.
* The data storing and retrieving, it is hard for storage servers to directly support other functions.

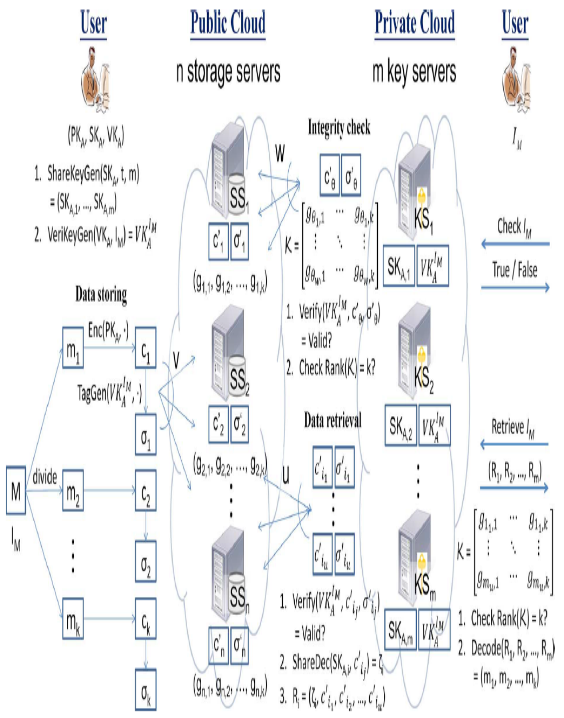
PROPOSED SYSTEM:

In our proposed system we address the problem of forwarding data to another user by storage several servers directly under the command of the data owner. We consider the system model that consists of distributed storage different servers and key servers. Since storing cryptographic keys in a single device is risky, a user distributes his cryptographic key to key servers that shall perform cryptographic functions on behalf of the user. These key servers are highly protected by security mechanisms.
The distributed systems require independent servers to perform all operations. We propose a new threshold proxy re-encryption scheme and integrate it with a secure decentralized code to form a secure distributed storage system. The encryption scheme supports encoding operations over encrypted messages and forwarding operations over encrypted and encoded messages.

ADVANTAGES:

* Tight integration of encoding, encryption, and forwarding makes the storage system efficiently meet the requirements of data robustness, data confidentiality, and data forwarding.
* The storage servers independently perform encoding and re-encryption process and the key servers independently perform partial decryption process using AES Algorithms and Erasure code technique.
* More flexible adjustment between the number of storage servers and robustness.

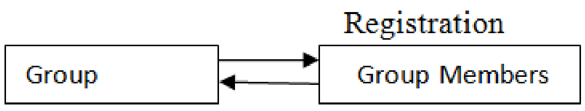
ARCHITECTURE DIAGRAM


User
Public Cloud
Private Cloud
User
n storage servers
m key servers
Integrity check
Data storing
Data retrieval
Check I_M
True / False
Retrieve I_M
divide
1. ShareKeyGen(SK_A, t, m) = (SK_{A,1}, ..., SK_{A,m})
2. VeriKeyGen(VK_A, I_M) = VK_A^{I_M}
1. Verify(VK_A^{I_M}, c'_θ, σ'_θ) = Valid?
2. Check Rank(K) = k?
1. Check Rank(K) = k?
2. Decode(R_1, R_2, ..., R_m) = (m_1, m_2, ..., m_k)
1. Verify(VK_A^{I_M}, c'_{ij}, σ'_{ij}) = Valid?
2. ShareDec(SK_{A,j}, c'_{ij})
3. R_j = (ς, c'_{i_1}, c'_{i_2}, ..., c'_{i_u})

MODULES

Registration:

For the registration of user with identity ID the group manager randomly selects a number. Then the group manager adds into the group user list which will be used in the traceability phase. After the registration, user obtains a private key which will be used for group signature generation and file decryption.



Sharing Data:

The canonical application is data sharing. The public auditing property is especially useful when we expectthe delegation to be efficient and flexible. The schemes enable a content provider to share her data in a confidentialand selective way, with a fixed and small ciphertext expansion, by distributing to each authorized user a single and small aggregate key.

Secure Cloud Storage:

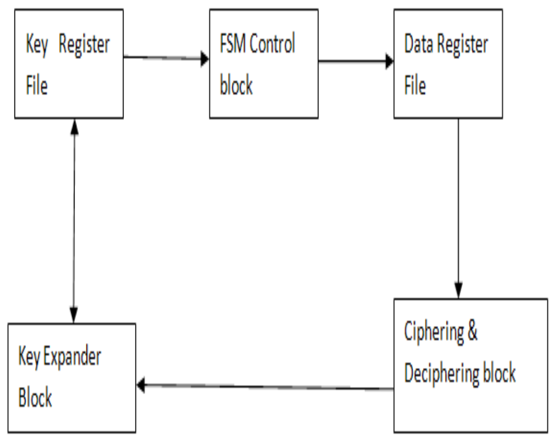
Data robustness is a major requirement for storage systems. There have been many proposals of storing data over storage servers. One way to provide data robustness is to replicate a message such that each storage server stores a copy of the message. A decentralized erasure code is suitable for use in a distributed storage system.

Proxy re-encryption:

Proxy re-encryption schemes are crypto systems which allow third parties (proxies) to alter a cipher text which has been encrypted for one user, so that it may be decrypted by another user. By using proxy re-encryption technique the encrypted data (cipher text) in the cloud is again altered by the user. It provides highly secured information stored in the cloud. Every user will have a public key and private key. Public key of every user is known to everyone but private key is known only the particular user.

Data retrieval:

Reports and data are the two primary forms of the retrieved data from servers. There are some overlaps between them, but queries generally select a relatively small portion of the server, while reports show larger amounts of data. Queries also present the data in a standard format and usually display it on the monitor; whereas reports allow formatting of the output however you like and is normally retrieved.



SYSTEM ANALYSIS FEASIBILITY STUDY:

Feasibility studies aim to objectively and rationally uncover the strengths and weaknesses of the existing business or proposed venture, opportunities and threats as presented by the environment, the resources required to carry through, and ultimately the prospects for success.
In its simplest term, the two criteria to judge feasibility are cost required and value to be attained. As such, a well-designed feasibility study should provide a historical background of the business or project, description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations. Generally, feasibility studies precede technical development and project implementation.

They are 3 types of Feasibility

•Economical feasibility
•Technical feasibility
•perationalfeibility

ECONOMICAL FEASIBILITY:

The assessment is based on an outline design of system requirements in terms of Input, Processes, Output, Fields, Programs, and Procedures. This can be quantified in terms of volumes of data, trends, frequency of updating, etc. in order to estimate whether the new system will perform adequately or not. Technological feasibility is carried out to determine whether the company has the capability, in terms of software, hardware, personnel and expertise, to handle the completion of the project

•Whether the required technology is available or not
•Whether the required resources are available
•Manpower- programmers, testers & debuggers
•Software and hardware

Once the technical feasibility is established, it is important to consider the monetary factors also. Since it might happen that developing a particular system may be technically possible but it may require huge investments and benefits may be less. For evaluating this, economic feasibility of the proposed system is carried out.

Volume No:2, Issue No:1 (June-2016)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

OPERATIONAL FEASIBILITY:

Operational feasibility is mainly concerned with issues like whether the system will be used if it is developed and implemented. Whether there will be resistance from users that will affect the possible application benefits? The essential questions that help in testing the operational feasibility of a system are following.
•Does management support the project?
•Are the users not happy with current business practices?
•Will it reduce the time (operation) considerably? If yes, then they will welcome the change and the new system.
•Have the users been involved in the planning and development of the project? Early involvement reduces the probability of resistance towards the new system.
•Will the proposed system really benefit the organization?

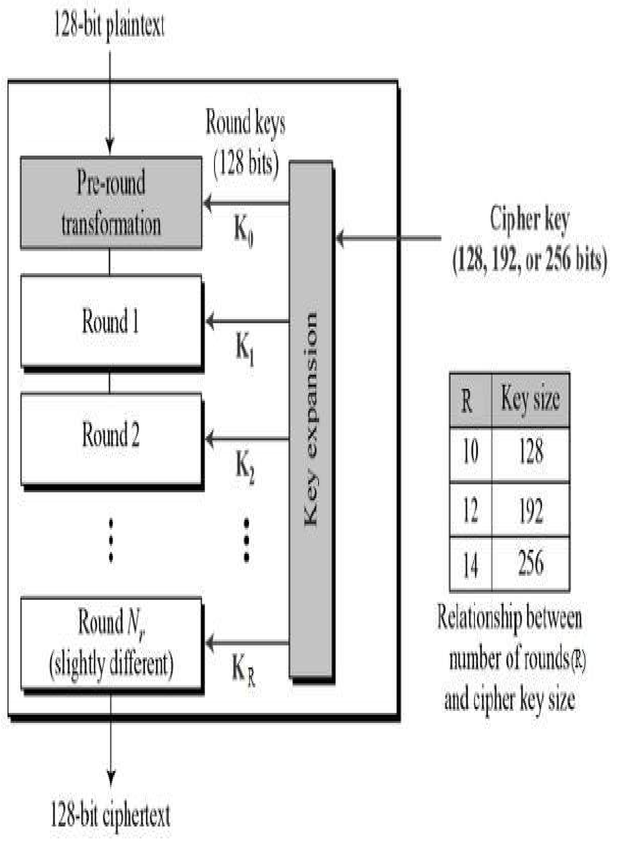
AES:

A replacement for DES was needed as its key size was too small. With increasing computing power, it was considered vulnerable against exhaustive key search attack. Triple DES was designed to overcome this drawback but it was found slow.
The features of AES are as follows −
•Symmetric key symmetric block cipher
•128-bit data, 128/192/256-bit keys
•Stronger and faster than Triple-DES
•Provide full specification and design details
•Software implementable in C and Java

Operation of AES:

AES is an iterative rather than Feistel cipher. It is based on ‘substitution–permutation network’. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations). Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix − Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.

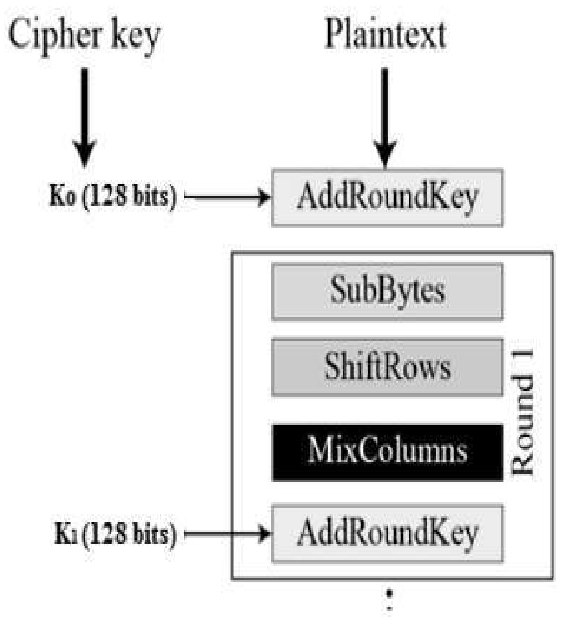
The schematic of AES structure is given in the following illustration −


128-bit plaintext
Round keys (128 bits)
Pre-round transformation
K_0
Cipher key (128, 192, or 256 bits)
Round 1
K_1
Round 2
K_2
Key expansion
Round N_r (slightly different)
K_R
128-bit ciphertext
Relationship between number of rounds (R) and cipher key size

R	Key size
10	128
12	192
14	256

Encryption Process:

Here, we restrict to description of a typical round of AES encryption. Each round comprise of four sub-processes. The first round process is depicted below −



Byte Substitution (SubBytes)

The 16 input bytes are substituted by looking up a fixed table (S-box) given in design. The result is in a matrix of four rows and four columns.

Shiftrows

Each of the four rows of the matrix is shifted to the left. Any entries that ‘fall off’ are re-inserted on the right side of row. Shift is carried out as follows −

•First row is not shifted.
•Second row is shifted one (byte) position to the left.
•Third row is shifted two positions to the left.
•Fourth row is shifted three positions to the left.
•The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other.

MixColumns

Each column of four bytes is now transformed using a special mathematical function. This function takes as input the four bytes of one column and outputs four completely new bytes, which replace the original column. The result is another new matrix consisting of 16 new bytes. It should be noted that this step is not performed in the last round.

Addroundkey

The 16 bytes of the matrix are now considered as 128 bits and are XORed to the 128 bits of the round key. If this is the last round then the output is the ciphertext. Otherwise, the resulting 128 bits are interpreted as 16 bytes and we begin another similar round.

Decryption Process

The process of decryption of an AES ciphertext is similar to the encryption process in the reverse order. Each round consists of the four processes conducted in the reverse order −

•Add round key
•Mix columns
•Shift rows
•Byte substitution

Since sub-processes in each round are in reverse manner, unlike for a Feistel Cipher, the encryption and decryption algorithms needs to be separately implemented, although they are very closely related.

AES cipher: Pseudocode

```
Cipher(byte in[16], byte out[16], key_array round_key[Nr+1])
begin
byte state[16];
state = in;
AddRoundKey(state, round_key[0]);
 for i = 1 to Nr-1 stepsize 1 do
SubBytes(state);
 ShiftRows(state);
 MixColumns(state);
 AddRoundKey(state, round_key[i]);
 end for
SubBytes(state);
 ShiftRows(state);
 AddRoundKey(state, round_key[Nr]);
end
```

AES Analysis

In present day cryptography, AES is widely adopted and supported in both hardware and software. Till date, no practical cryptanalytic attacks against AES have been discovered. Additionally, AES has built-in flexibility of key length, which allows a degree of ‘future-proofing’ against progress in the ability to perform exhaustive key searches. However, just as for DES, the AES security is assured only if it is correctly implemented and good key management is employed.

CONCLUSION:

Erasure codes are promising for improving the reliability of the storage system due to its space efficiency compared to the replication methods. Traditional erasure codes split data into equalsized data blocks and encode strips in different data blocks. This brings heavy repairing traffic when clients read parts of the data, since most strips read for repairing are not in the expected blocks. This paper proposes a novel discrete data dividing method to completely avoid this problem. The key idea is to encode strips from the same data block. We could see that for repairing failed blocks, the strips to be read are either in the same data block with corrupted strips or from the encoded strips.

Volume No:2, Issue No:1 (June-2016) ISSN No : 2454-423X (Online)

International Journal of Research in Advanced Computer Science Engineering

A Peer Reviewed Open Access International Journal

www.ijracse.com

Therefore, no data is wasted. We design and implement this data layout into a HDFS-like storage system. Experiments over a small-scale test bed shows that the proposed discrete data divided method avoids downloading data blocks that are not needed for clients during the repairing operations.

REFERENCES:

[1] H.-Y. Lin and W.-G. Tzeng, "A secure decentralized erasure code for distributed networked storage," IEEE Trans. Parallel Distrib. Syst., vol. 21, no. 11, pp. 1586–1594, Nov. 2010.

[2] H.-Y. Lin and W.-G. Tzeng, "A secure erasure code-based cloud storage system with secure data forwarding," IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 6, pp. 995–1003, Jun. 2012.

[3] A. Juels and B. S. Kaliski, "Pors, proofs of retrievability for large files," in Proc. 14th ACM Conf. Computer and Commun. Security (CCS'07), 2007, pp. 584–597.

[4] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proc. 14th ACM Conf. Computer and Commun. Security (CCS'07), 2007, pp. 598–609.

[5] G. Ateniese,R.Burns,R.Curtmola, J. Herring,O.Khan, L. Kissner, Z. Peterson, andD. Song, "Remote data checking using provable data possession," ACM Trans. Inf. Syst. Secur., vol. 14, no. 1, pp. 12:1–12:34, Jun. 2011.

[6] C. C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proc. 16th ACM Conf. Computer and Commun. Security (CCS'09), 2009, pp. 213–222.

[7] G.Ateniese, S.Kamara, and J. Katz, "Proofs of storage fromhomomorphic identification protocols," in Proc. 15th Int. Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT'09), 2009, pp. 319–333.

[8] S.-T. Shen and W.-G. Tzeng, "Delegable provable data possession for remote data in the clouds," in Proc. 13th Int. Conf. Information and Commun. Security (ICICS'11), 2011, pp. 93–111.

[9] H. Shacham and B. Waters, "Compact proofs of retrievability," J. Cryptol., vol. 26, no. 3, pp. 442–483, 2013.

[10] H.-Y. Lin, W.-G. Tzeng, and B.-S. Lin, "A decentralized repair mechanism for decentralized erasure code based storage systems," in Proc. 10th IEEE Int. Conf. Trust, Security and Privacy in Computing and Commun. (TrustCom'11), Nov. 2011, pp. 613–620.

[11] K. D. Bowers, A. Juels, and A. Oprea, "Hail, a high-availability and integrity Layer for cloud storage," in Proc. 16th ACM Conf. Computer and Commun. Security (CCS'09), 2009, pp. 187–198.

[12] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, "Remote data checking for network coding-based distributed storage systems," in Proc. 2nd ACM Workshop Cloud Computing Security (CCSW'10), 2010, pp. 31–42.

[13] N. Cao, S. Yu, Z. Yang, W. Lou, and Y. T. Hou, "Lt codes-based secure and reliable cloud storage service," in Proc. 31st IEEE Int. Conf. Computer Commun. (INFOCOM'12), 2012, pp. 693–701.

[14] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," J. Cryptol., vol. 17, no. 4, pp. 297–319, Sep. 2004.

[15] Y. Dodis, S. Vadhan, and D. Wichs, "Proofs of retrievability via hardness amplification," in Proc. 6th Theory of Cryptography Conf. (TCC'09), 2009, pp. 109–127.

[16] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proc. 4th Int. Conf. Security and Privacy in Commun. Netowrks (SecureComm'08), 2008, pp. 1–10.

[17] M. T. Goodrich, C. Papamanthou, R. Tamassia, and N. Triandopoulos, "Athos: Efficient authentication of outsourced file systems," in Proc. 11th Int. Conf. Information Security (ISC'08), 2008, pp. 80–96.

[18] A. Heitzmann, B. Palazzi, C. Papamanthou, and R. Tamassia, "Efficient integrity checking of untrusted network storage," in Proc. 4th ACM Workshop Storage Security and Survivability (StorageSS'08), 2008, pp. 43–54.

[19] Q.Wang, C.Wang, J. Li, K. Ren, andW. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Proc. 14th Eur. Conf. Research in Computer Security (ESORICS'09), 2009, pp. 355–370.