



International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Survey on Web Application in Cloud Computing Using Encryption and Decryption

Hemant Kawade, Mohinto Sakhare, Rupesh Mahajan
Department of Computer, Savitribai Phule Pune University,
Pune, Maharashtra, India

Abstract: Migrating from server-attached storage to distributed storage brings new vulnerabilities in creating a secure data storage and access facility. Particularly it is a challenge on top of insecure networks or unreliable storage service providers. For example, in applications such as cloud computing where data storage is transparent to the owner. It is even harder to protect the data stored in unreliable hosts. More robust security scheme is desired to prevent adversaries from obtaining sensitive information when the data is in their hands. We propose a system in which data is divided and out of order key stream is generated. This is novel encryption approach to protect data in distributed storage environments. Outsourcing data to a third-party administrative control, as is done in cloud computing, gives rise to security concerns. The data compromise may occur due to attacks by other users and nodes within the cloud. Therefore, high security measures are required to protect data within the cloud. However, the employed security strategy must also take into account the optimization of the data retrieval time. The Division and Replication of Data in the Cloud for Optimal Performance and Security that collectively approaches the security and performance issues. In the DROPS methodology, we divide a file into fragments, and replicate the fragmented data over the cloud nodes. Each of the nodes stores only a single fragment of a particular data file that ensures that even in case of a successful attack, no meaningful information is revealed to the attacker. Moreover, the nodes storing the fragments, are separated with certain distance by means of graph T-coloring to prohibit an attacker of guessing the locations of the fragments. Furthermore, the DROPS methodology does not rely on the traditional cryptographic techniques for the data security; thereby relieving the system of computationally expensive methodologies. The probability to locate and compromise all of the nodes storing the fragments of a single file is extremely low. We also compare the performance of the DROPS methodology with ten other schemes. The higher level of security with slight performance overhead was observed.

Keywords: Data Security, Distributed Storage, Stream Cipher, Encryption and Decryption.

I. INTRODUCTION

Cloud computing, also known as on-demand computing, is a kind of Internet-based computing, where shared resources, data and information are provided to computers and other devices on-demand. It is a model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third-party data centers. It relies on sharing of resources to achieve coherence and economies of scale, similar to a utility over a network.

Data storage has been recognized as one of the main dimensions of information technology. The prosperity of network based applications leads to the moving from server attached storage to distributed storage. Along with variant advantages, the distributed storage also poses new challenges in creating a secure and reliable data storage and access facility over insecure or unreliable service providers. Aware of that data security is the kernel of information security, a plethora of efforts has been made in the area of distributed storage security. During past decades, most designs of distributed storage chose the form of either Storage Area Networks (SANs) or Network-Attached Storage (NAS) on the LAN level, such as a network of an enterprise, a campus, or an organization. Either in SANs or NAS, the distributed storage nodes are managed by the same authority. The system administrator has the access and control over each node, and essentially the security level of data is under control. The reliability of such systems is often achieved by redundancy, and the storage security is highly depending on the security of the system against the attacks/intrusion from outsiders. The confidentiality and integrity of data are mostly achieved using robust cryptograph schemes. However, such a security system is not robust enough to protect the data in distributed storage applications at the level of wide area networks. The recent progress of network technology enables global-scale collaboration over heterogeneous networks under different authorities.

II. RELATED WORK

1) Project Name: - Self-Encryption Scheme for Data Security in Mobile Devices

Authors: - Yu Chen and Wei-Shinn Ku

Abstract: - The pervasive use of wireless networks and mobile devices has been changing our living style significantly. Along with great convenience and efficiency, there are new challenges in protecting sensitive and/or private data carried

in these devices. The most challenging part lies in a dilemma: while it should be computationally infeasible for adversaries to decrypt the data, the cryptographic operation should be efficient for legitimate users and minimize battery drain. It give information about a novel data encryption and storage scheme to address this challenge. Treating the data as a binary bit stream, our self-encryption (SE) scheme generates a key stream by randomly extracting bits from the stream. The length of the key stream depends on the user's security requirements. The bit stream is encrypted and the cipher text is stored on the mobile device, whereas the key stream is stored separately. This makes it computationally not feasible to recover the original data stream from the cipher text alone.

2) Project Name: - Improved Analysis of the BMGL Key stream Generator

Authors: - Johan Hastad, Mats Naslund

Abstract: - It give information about an improved security analysis of the NESSIE submission BMGL. The new analysis improves also asymptotically some of the theoretical results on which the BMGL key stream generator is based. We also give an alternative, bootstrapped version of the generator which is implementation-wise very close to the original generator and offers even stronger provable security properties.

3) Project Name: - A Survey of Security Services and Techniques in Distributed Storage Systems.

Authors: - Zhiqian Xu, Keith Martin, and Clifford L. Kotnik

Abstract: - The rapid growth of data and data sharing have been driven an evolution in distributed storage infrastructure. The need for sensitive data protection and the capacity to handle massive data sets have encouraged the research and development of secure and scalable storage systems. It give information about identifies major security issues and requirements of data protection related to distributed data storage systems. We classify the security services and techniques in existing or proposed storage systems. We then discuss potential research topics and future trends.

4) Project Name: - Reliable and Secure Distributed Storage of Critical Information

Abstract: - Distributed storage systems aim at providing reliable and secure access to critical data over large networks, by utilizing a distributed collection of parallel storage servers which may be individually unreliable and insecure. Other than the cryptographic approach that relies on secret keys and computational hardness assumptions to provide security, non cryptographic algorithms have been developed as an efficient way to enhance reliability of distributed storage systems. However, such a non-cryptographic approach puts security at risk and is vulnerable under joint reliability and security breaches. It give information about a non-cryptographic algorithm for reliable and secure distributed storage, by invoking results on linear error control codes. This algorithm achieves a combination and tradeoff among three important functionalities: reliability, confidentiality, and integrity, which are collectively, measured using a new unifying metric, resilience vector, give information about. A rigorous security and complexity analysis is provided and allows our algorithm to be optimized under different environments. Implementation and simulation show that our algorithm improves both reliability and security of distributed storage systems by three 'nines' at low computation and storage overhead, requiring only bitwise XOR and table lookup operations.

III. CONTRIBUTION

The entire application is divided into modules according to their functionality. The division allows better understanding of the architecture as a whole.

1) Cloud Client:-

Cloud client should be Data owner or Data user.

- Data Owner:-
Data owner is responsible for uploading file on cloud as well as view files uploaded by him or others. Data owner has information about the placed fragment and its replicas with their node numbers in cloud.
- Data User:-
Data user is the one who is responsible for downloading files or view files uploaded by others. To download file from cloud he has to be authenticated user otherwise he will be considered as attacker.

2) Cloud Server:-

- Key Generator: -
This approach is used for creating key which are used for encryption and decryption. This approach uses key generation algorithm. The input of this approach is plain text or file and it produces the output as binary key generated.
- Encryption with block generation:-
This approach performs two functions such as Encryption and block generation. Encryption algorithm divides the plaintext in fixed length of blocks then by using a key it performs encryption on plaintext and produces the same length of block as a cipher text. In this method all blocks are encrypted with the same key, which degrades security because each repetition in the plaintext will be a repetition in the cipher text.
- Decryption:-
As the data is encrypted in above approach to access the data in original format it is necessary to decrypt that data. For this purpose this approach uses decryption algorithm. The input to this approach is blocks of cipher text and it produces the output as plain text.

- **Cipher Block Allocation:-**
In the encryption approach while encrypting data we are dividing that data into fixed size blocks. As the blocks are created we have to allocate that blocks on cloud server at different nodes. While allocating that block we have to take about security so we are allocating that blocks using T- coloring graph concept. This approach runs the cipher block allocation algorithm.
- **Replication:-**
This approach creates replicas (duplicate copy) of fragments. These replicas are useful when one of fragment is corrupted by attacker then to provide file for user admin replaces its replica at that place and combine all fragments and send file to authenticated user or data owner. To make replicas of file fragments this approach runs replication algorithm which takes input as fragments and produces its replicas as output.

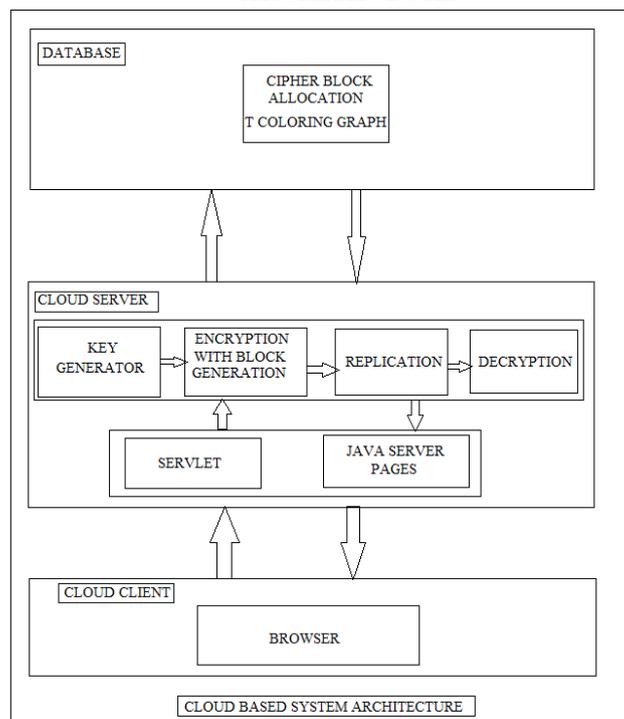
IV. PROPOSED METHODOLOGY AND DISCUSSION

The major design goal of the this strategy is the confidentiality and integrity of the sensitive/privacy data that is stored in the Internet based distributed storage infrastructure such as Grid Storage or Cloud Computing, where the data owner can control neither the reliability/security of the medium, nor the violation of the medium provider or administrators. Either the medium providers or an adversary who has successfully compromised a storage node could do whatever he/she wants to the data in the machine. Therefore, our purpose is to make it computationally infeasible to reveal any meaningful information from each cipher text pieces.

In this scheme first we are constructing the key stream for encryption after key generation encrypting the data. Dividing the cipher text into multiple data blocks with fixed-length, the last block will be stuffed if it consists of fewer bits. Allocating storage nodes in the network and sending each block to one of them.

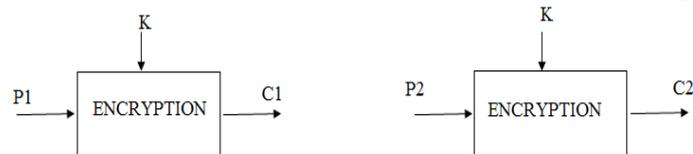
If the hacker tries to get the information from cloud node then its notification information send to the admin.

V. ARCHITECTURE



VI. ALGORITHM WITH EXPLANATION

- **Key stream Generator:-**
Input:-Plain text or file
Output:-Binary Key generated
Procedure:-
Begin
For i=0 to 2047 step by 1
Fixed keys=Generate a random number
End for
Return the generated numbers
End function
- **Encryption Algorithm:-**
Inputs:-Plain text, binary key
Outputs:-Blocks of Cipher-text



Encryption algorithm divides the plaintext in fixed length of blocks then by using a key it performs encryption on plaintext and produces the same length of block as a cipher text as shown in figure. In this method all blocks are encrypted with the same key, which degrades security because each repetition in the plaintext will be a repetition in the cipher text. To counter this issue, modes of operation are used to make encryption more reliable.

Procedure:-

```

Begin
Open ciphered file for writing
Open plain file for reading
While it is not the end of the file do
Read 16 bytes form the input file
Convert 16 bytes to hexadecimal format
Call scheduled key from matrix
Ciphered data block=cipher (input, scheduled key)
Write ciphered data block to output file
End While

```

End Function

- Decryption Algorithm:-

Input:-Blocks of cipher text

Outputs:-Plain Text

Procedure:-

```

Begin
Open ciphered file for reading
Open plain file for writing
While it is not the end of the file
Read 32 bytes as ciphered data block
Scheduled key=Fixed key [Start point.....scheduled key length-1]
Plain text=decipher (schedule key, ciphered data block, scheduled key)
Convert plain text from hexadecimal to string
Write plain block to the plain file
End while
End Function

```

- Ciphered block Allocation:-

Input:-Ciphered blocks

Output:-cloud server nodes with ciphered blocks allocated

Once the ciphered blocks are created it is necessary to selects the cloud nodes for block placement. The selection is made by keeping an equal focus on both security and performance in terms of the access time.

- Replication Algorithm:-

Input:-Ciphered blocks.

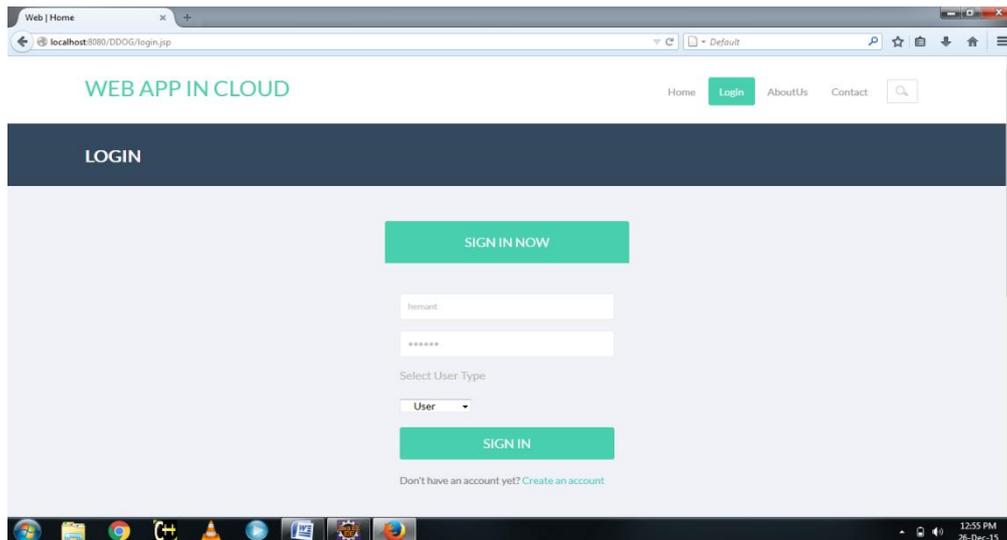
Output:-Replicas of ciphered block.

```

for each  $O_k$  in  $O$  do
select  $S^i$  that has  $\max(R_k^i + W_k^i)$ 
if  $col_{S^i} = open\_color$  and  $s_i \geq o_k$  then
 $S^i \leftarrow O_k$ 
 $s_i \leftarrow s_i - o_k$ 
 $col_{S^i} \leftarrow close\_color$ 
 $S^{i'} \leftarrow distance(S^i, T)$   $\triangleright$  /*returns all nodes at
distance  $T$  from  $S^i$  and stores in temporary set  $S^{i'}$ */
 $col_{S^{i'}} \leftarrow close\_color$ 
end if
end for

```

VII. IMPLEMENTATION



VIII. CONCLUSION

In this system, it give information about a novel steam cipher encryption for data security in distributed storage. In the proposed methodology, a cloud storage security scheme that collectively deals with the security and performance in terms of retrieval time. The data file was encrypted and fragmented and the fragments are dispersed over multiple nodes. The nodes were separated by means of T-coloring. The fragmentation and dispersal ensured that no significant information was obtainable by an adversary in case of a successful attack. No node in the cloud, stored more than a single fragment of the same file.

REFERENCES

- [1] D. J. Bernstein, "Which e STREAM ciphers have been broken?" [http://www.ecrypt.eu.org/ stream/](http://www.ecrypt.eu.org/stream/), submitted 2008.
- [2] Self-Encryption Scheme for Data Security in Mobile Devices Yu Chen and Wei-Shinn Ku.
- [3] Improved Analysis of the BMGL Key stream Generator Johan Hastad NADA Royal Inst. of Technology SE-10044 Stockholm, Sweden Mats Naslund Communications Security Lab Ericsson Research SE-16480 Stockholm, Sweden August 24, 2001.
- [4] A Survey of Security Services and Techniques in Distributed Storage Systems Zhiqian Xu, Keith Martin, and Clifford L. Kotnik Information Security Group, Royal Holloway, University of London, London, UK FedEx Corporation, Memphis, TN, USA.
- [5] Reliable and Secure Distributed Storage of Critical Information Paper ID: 1569169570.
- [6] D. Boru, D. Kliazovich, F. Granelli, P. Bouvry, and A. Y. Zomaya, "Energy-efficient data replication in cloud computing datacenters," In IEEE Globecom Workshops, 2013, pp. 446-451.
- [7] K. Bilal, S. U. Khan, L. Zhang, H. Li, K. Hayat, S. A. Madani, N. Min-Allah, L. Wang, D. Chen, M. Iqbal, C. Z. Xu, and A. Y. Zomaya, "Quantitative comparisons of the state of the art data center architectures," Concurrency and Computation: Practice and Experience, Vol. 25, No. 12, 2013, pp. 1771-1783.
- [8] K. Bilal, M. Manzano, S. U. Khan, E. Calle, K. Li, and A. Zomaya, "On the characterization of the structural robustness of data center networks," IEEE Transactions on Cloud Computing, Vol. 1, No. 1, 2013, pp. 64-77.
- [9] D. Boru, D. Kliazovich, F. Granelli, P. Bouvry, and A. Y. Zomaya, "Energy-efficient data replication in cloud computing datacenters," In IEEE Globecom Workshops, 2013, pp. 446-451.
- [10] Y. Deswarte, L. Blain, and J-C. Fabre, "Intrusion tolerance in distributed computing systems," In Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy, Oakland CA, pp. 110-121, 1991.
- [11] B. Grobauer, T. Walloschek, and E. Stocker, "Understanding cloud computing vulnerabilities," IEEE Security and Privacy, Vol. 9, No. 2, 2011, pp. 50-57.