Global Journal of Computer Science and Technology: B
Cloud and Distributed
Volume 17 Issue 1 Version 1.0 Year 2017
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Secure Data Distribution using Secret Splitting Over Cloud

By Sagar P. Jaikar, Ramkrushna C. Maheshwar, Sameer P. Mamadapure & Anand A. Bhosle
International Institute of Information Technology

Abstract- Developments are important to ride the unavoidable tide of progress. A large portion of undertakings are endeavoring to lessen their processing cost through the method of virtualization. This interest of diminishing the computing cost has induced the development of Cloud Computing. Cloud computing provides set of services to the customers over the network on rented basis which can be scaled up or down as per customers requirements. Typically cloud computing administrations are conveyed by an outsider supplier who possesses the foundation. In this paper we are focusing on secure data distribution over cloud using secret splitting. This will help us to achieve data confidentiality, integrity & availability with less overhead. We proposed a secure data distribution scheme using secret splitting to ensure data owners that their data are distributed securely over the cloud.

Keywords: cloud computing; distributed computing; secret splitting; data confidentiality; integrity.

GJCST-B Classification: D.2.7

Strictly as per the compliance and regulations of:

Secure Data Distribution using Secret Splitting Over Cloud

Sagar P. Jaikar [α], Ramkrushna C. Maheshwar [σ], Sameer P. Mamadapure [ρ] & Anand A. Bhosle [ω]

Abstract- **Developments are important to ride the unavoidable tide of progress. A large portion of undertakings are endeavoring to lessen their processing cost through the method of virtualization. This interest of diminishing the computing cost has induced the development of Cloud Computing. Cloud computing provides set of services to the customers over the network on rented basis which can be scaled up or down as per customers requirements. Typically cloud computing administrations are conveyed by an outsider supplier who possesses the foundation. In this paper we are focusing on secure data distribution over cloud using secret splitting. This will help us to achieve data confidentiality, integrity & availability with less overhead. We proposed a secure data distribution scheme using secret splitting to ensure data owners that their data are distributed securely over the cloud.**

Keywords: cloud computing; distributed computing; secret splitting; data confidentiality; integrity.

I. Introduction

Cloud computing offers huge advantages to its adopters, however it additionally accompanies its arrangement of issues and inefficiencies of which security is the greatest concern. Keeping in mind the end goal to influence a remote cloud based foundation; an organization basically gives away private information and data that may be delicate and classified. Secret splitting plans are utilized to confine access to such delicate and classified information. In cloud computing, security is considered to be a critical viewpoint because of the noteworthiness of data put away in the cloud. The information can be private and to a great degree delicate. Thus, the information administration ought to be totally dependable. It is important that the data in the cloud is shielded from various assaults. Security acquires attentiveness toward secrecy, integrity and accessibility of information. Unapproved access to data results in loss of information secrecy. Information integrity and accessibility suffers due to outages of services provided by cloud service providers (CSP's).

Author α: Dept. of Information Technology, International Institute of Information Technology, Hinjawadi, Pune, Maharashtra, India.
e-mail: jaikarsagar@gmail.com
Author σ: Dept. of Computer Engineering, International Institute of Information Technology, Hinjawadi, Pune, Maharashtra, India.
e-mail: remomaheswar1987@gmail.com
Author ρ: Dept. of Information Technology, International Institute of Information Technology, Hinjawadi, Pune, Maharashtra, India.
e-mail: sam16.mamadapure@gmail.com
Author ω: Dept. of Information Technology, International Institute of Information Technology, Hinjawadi, Pune, Maharashtra, India.
e-mail: anand.a.bhosle@gmail.com

Cloud computing is the technique of using a network of remote computing resources hosted on the network, rather than on a local server. This moves user's data from local storage to cloud servers which are placed in third party premises. As advancement in computing technology, now day's user's believes in anytime anywhere computing. They often access documents without knowing where they are stored & how they are stored. They store documents with unknown providers, especially in distributed processing situations. This gives several crucial advantages to users. Firstly they don't need to worry about storage management for the data, second, they can have anytime anywhere access to their data, third, they can avoid expenditure on hardware & software infrastructure. These appealing benefits make cloud fancier to their users but as the data is outsourced into cloud they are no longer in control of data, so it posses threats to data integrity & confidentiality.

Other than this there were several incidents of service outages & security breaches from CSP's. Amazon's data storage service was down for several hours recently[12], Gmail's mass email deletion occurrence[13] are some recent examples of it. This kind of outages causes violation of data availability to corresponding users[12]. Also there are various motivations for CSP's to behave unfairly with clients with respect to outsourced data such as hiding the data loss incidents, deleting infrequent data. It means that though we are uploading the data into cloud, we are held at the mercy of CSP's for data confidentiality, data integrity & data availability. In this paper, we will be discussing the data distribution technique which will allow us to distribute the data among various users without violating data confidentiality & data integrity. Despite of distributing data over cloud we will not be depending on CSP for data confidentiality as with secret splitting we are not disclosing original data into cloud. It will definitely add some input/output overhead but here we are mainly focusing on secure data distribution only.

II. Related Work

Though cloud computing is very attractive to its users it poses many security challenges due to numerous reasons. As users are outsourcing their data to third party servers, they are not in control of it. It means standard cryptographic techniques will not be enough to protect the data. Also cloud is not just data

warehouse, frequent changes will be made to data, so data should be in consistent state.

Therefore, we need to depend upon security policies applied by cloud service provider. Considering different kind of data for different users & the demand for data safety as well as of storage correctness within the cloud becomes more difficult. So we can broadly classify the security concerns in three parameters namely Confidentiality, Integrity & Availability. Confidentiality is a security requirement in which the message must be correctly interpreted by the intended user. To do this, unauthorized access and usage must be prevented. Integrity security requirement can be subdivided as origin & data integrity, where we are concerned about source authenticity & data correctness. Availability requirement is data must be available to all legitimate users of the system.

Secret Sharing approaches are one of the vital strategies used for data distribution over third party servers. Two standard secret sharing schemes are the Shamir's Secret Sharing algorithm and Rabin's information Dispersal Algorithm (IDA)[11]. In Shamir's algorithm to distribute a file *F,* we need to cut it into *n* constituents *F1, F2, F3..Fn.* Here, every file *Fi,* $i \leq n$, is padded with some dummy bits to make it exactly of equal size of that original *F*. To obtain the original file *F* we need *k* out of *n* constituents or else we will not be able to obtain it. Shamir calls this as *threshold (k,n scheme)*[1]. But here we are distributing original file via constituents/shares which is different from secret splitting where we are not distributing original file. Rabin proposed Information Dispersal Algorithm where we can split secret *S* into *n* different pieces in such way that to regenerate the secret we require *x* pieces, where *x* is threshold & $x<n$. Though this algorithm will reduce storage complexity it has limitation if the pieces exhibit some pattern then attacker may obtain the secret[2].

Zage et al. developed an alternative to secret split archives was in which an algebraic-based encoding solution, Matrix Block Chaining (MBC).It is used for maintaining data security when encoding large files. The design of MBC is done accordingly to allow encoding of multiple partitions of the original data in parallel as subsequent encoding operations are independent of the output of previous encoding steps. Their technique was designed specifically for cloud storage, however, and as such cannot maintain data availability in a compromised environment[3]. Huchton et al. presented an approach for sensitive data sharing across mobile devices in a front-line environment. Here they used a similar approach as a way to protect sensitive digital data among troops in the field[4].

There are various cryptographic techniques used in the distributed storage system such as data encipherment, homomorphic encryptions[5], secret sharing & splitting algorithms and Private Information Retrieval[6]. Even though PIR and homomorphic encryption can ensure the confidentiality of data, they induced computational costs. In addition, adversaries can affect both throughput and latency. Furthermore, data encryption is insufficient to ensure the security of data, because it is still threatened by lost, theft or damage making it unavailable[10].

The basic idea of secret sharing scheme proposed by Blakley, is that an administrator dispatches a piece of shares about the secret to each participant such that a group of participants have privileges to recover the secret, but unprivileged group of participants cannot obtain any information about the secret[7].

Secret splitting schemes provide both data availability and a certain degree of data confidentiality, with low computational and storage costs compared with other cryptography techniques[9].

III. Overall System Design

a) Secret Splitting

In our proposed system we are using secret splitting to distribute the data securely over a cloud. In secret splitting, the message is shared among multiple users without breaking the original message into pieces. There are ways to take a message and divide it up into pieces. Each piece by itself means nothing, but when we put them together, the complete message appears.

If each user has a piece of message, then only together they can make the complete message. If any user vanishes with his single piece of the message, his information is useless by itself.

The simplest sharing scheme splits a message between two people. Here's a protocol in which Owner O can split a message between User 1 and User 2:

Step 1: Owner generates a random-bit string, *R*, the same length as the message, *M*.

Step 2: Owner XORs *M* with *R* to generate *S*.

$$M \bullet R = S$$

Step 3: Owner gives *R* to User 1 and *S* to User 2. To reconstruct the message, User 1 and User 2 need their respective pieces:

Step 4: User 1 and User 2 XOR their pieces together to reconstruct the message:

$$R \bullet S = F$$

This technique, if done properly, is absolutely secure. Each piece, by itself, is absolutely worthless. Essentially, Owner is encrypting the message with a one-time pad and giving the cipher text to one person and the pad to the other person[8].

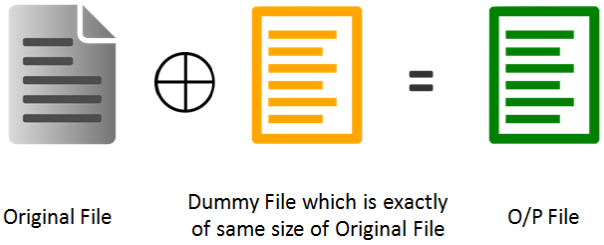

Fig.1: Secret Splitting

As shown in Fig. 1. In secret splitting we have to XOR Original file with dummy file which is exactly of same size of original file. The resultant XOR'ed file & dummy file is shared with two different users. Both the users need to come together with their shares to regenerate original file.

Secret splitting will allow us to achieve confidentiality. But we need to handle the integrity concerns separately. As we are aware if somebody (attacker) modifies the shares of users which we are putting in cloud, there is a possibility that users will not be able to generate the correct file from their respective shares. To guard against such integrity violations we are using SHA-1 hashing algorithm.

b) System Architecture

The data distribution architecture over cloud using secret splitting is illustrated in Fig.2. We are assuming two entities namely data owner & data user. Data owner will distribute their data securely through cloud and different users will get there corresponding shares. To regenerate the secret they need to combine their respective shares. Different entities are mentioned below:

1. *Owner:* These are the entities which will use actual secret splitting technique to upload the file for distribution over a cloud. Owner may choose to update & replace previously updated files.
2. *User:* Users, with whom owner have to share the data over the cloud. They will get their own share from the cloud service provider as they download it from cloud. The users need to come together to regenerate the original file. If anyone of the share is missing they cannot regenerate the original file.
3. *Cloud Service Provider (CSP):* CSP's are those enterprises who have large amount of resources to fulfill clients requirements for storage, processing, platforms etc. They are having their own infrastructure to handle client's data as well as applications. CSP's have the capabilities to scale up/down the resources as per clients needs.

Data owner will take the hash of original file F and encrypt the file with a shared secret key between owner & corresponding user. Owner will also encrypt the hash value using the same key. After encrypting original file & hash value, owner will attach a user id of corresponding user with whom owner wants to share the file. All the three parts are attached together & will be uploaded in cloud. Now the respective user needs to download his corresponding file from the cloud. Also user can verify the correctness of the file by recalculating hash on the file and matching it with attached hash value. Data owner will take a dummy file which is exactly of same size of the original file that owner wants to distribute among users. The Owner then XOR original file F, with dummy file D, producing an output file. The Output file S & Dummy file D is shared with two different users. As number of user's increases, we need to add more dummy files so that everyone will get their own shares.

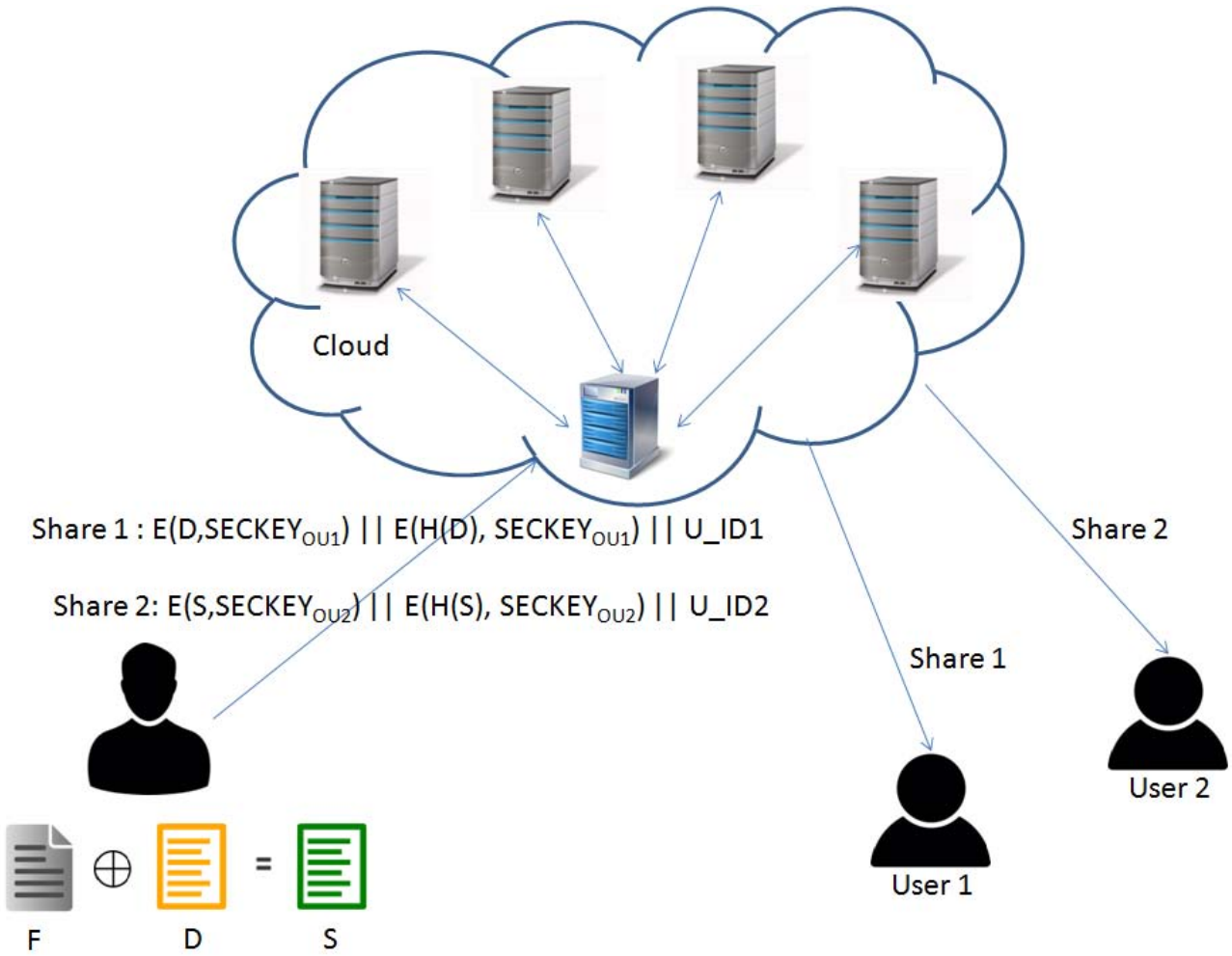

Fig.2: Data Distribution Architecture over Cloud using Secret Splitting

c) Notation & Preliminaries

- F – The original data file to be distributed. We are denoting F as a data matrix of m x n vectors.
- D – The dispersal matrix of x n vectors used as dummy file.
- S – Data matrix constructed using XOR of F & D
- H(file) – Hash value of file.
- SECKEYOU – Share d Secret Key between data owner & data user
- SECKEYOC – Shared Secret Key between data owner & cloud.
- User_ id- Corresponding user id.

In our approach owner will choose the number of dummy files which are exactly less by one than the number of users. We are denoting original file F as a data matrix of m *x n* vectors. Here we are assuming only two users in the system, but it can be used for more number of users also. So we are taking only one dummy file. Dummy file is exactly of same size of original file *F.* The dummy file is represented in a matrix *D* as dispersal matrix.

After *XORing F & D,* the resultant matrix *S* is generated. These matrixes *D & S* are nothing but the shares that are distributed over cloud. As we cannot send it clear over cloud, Data owner need to encrypt it using a secret key shared between him & corresponding user i.e. $SECKEY_{OU}$. Before encrypting it, owner needs to take hash of the particular matrix. i.e *H(file)*. After this both the parts are attached together along with the corresponding *User_ id* & dispatched into cloud. If attacker compromises cloud servers & tries to retrieve original file, the attempt will be thwarted as they are stored in encrypted format.

File Distribution Algorithm:

1. *Begin*
2. *Choose file F to upload & Dispersal matrix D. i.e dummy file.*
3. *Generate Matrix S = F XOR D.*
4. *Compute the hash on H(S) & H(D). Encrypt both S & D using AES algorithm.*

5. *Encrypt the hash values H(S) & H(D) using corresponding $SECKEY_{OU}$ i.e Shared Secret Key between data user & data owner.*
6. *Attach E(S, $SECKEY_{OU}$) || H(S) || $USER_{ID}$ and attach E(D, $SECKEY_{OU}$) || H(D) || $USER_{ID}$. Dispatch one part to User 1 & other part to User 2.*
7. *End*

In above algorithm we are assuming that there are two users in the system & data owner & users have a shared secret key among them.

File Download & Integrity Check:

1. *Begin*
2. *Download the corresponding User share. i.e. E(S, $SECKEY_{OU}$) || H(S) || $USER_{ID}$ or attach E(D, $SECKEY_{OU}$) || H(D) || $USER_{ID}$*
3. *Decrypt S or D using $SECKEY_{OU.}$ Calculate H(S') or H(D') & compare it with attached H(S) or D(S).*
4. *If match*
   data is intact & distributed properly.
   Else
   redistribute.
5. *End.*

After downloading the corresponding shares users can come together with their shares and they can reconstruct the original file. This approach is very secure as no attacker can get original file if he breaks into a cloud server. As every share is encrypted with shared secret key between corresponding user and data owner, only user possessing the valid key can open the share. The main advantage of this scheme is that the data owner doesn't require sending the original file into the third party cloud. As we have observed in data uploading process, owner is not uploading an original file, rather uploading dummy file & XOR'ed output file. So owner's data remains at his premises only, minimizing the risk of data compromise.

IV. Conclusion

We have investigated the information security worries in cloud information storage/distribution, which is a very significant issue. We proposed a secure data distribution scheme to ensure owners that their data will be distributed securely among the users over the cloud. In this approach, there is no need to calculate the tokens as there is no challenge response protocol to verify the data integrity of owner's data because in reality owners are not uploading their original files/data into the third party cloud rather they are uploading shares calculated using XOR operations. Still in this scheme we tried to ensure confidentiality & integrity of owner's files/data by using AES & SHA-1 algorithm respectively. As compared to secret sharing approaches, secret splitting technique will definitely have some input/output overheads which will cost on bandwidth usage as well as on storage utilization, but on the other hand it will allow us to distribute data more securely. Despite of all this we still believe that information security in Cloud is a zone brimming with difficulties and of vital significance.

V. Future Work

This scheme suffers from a drawback where we need to have all the shares to regenerate the original file. If a single share is lost then we cannot regenerate the original file. So we need to take at most care to bring all shares together. Also significant data overhead maybe caused as number of users increases, because in that case we need to upload that much shares. But looking at security benefits we can afford that much data overhead. We can enhance this approach & eliminate the need of bringing all shares together to regenerate original file by using secret sharing threshold scheme. In such schemes we need to set threshold such that out of m shares, n needed to come together to regenerate the original file. Also we can improve this approach by adding verifiable secret sharing where users can verify their shares that they have received from owner are correct & not the false shares.

References Références Referencias

1. Shamir, A.: "How to share a secret"., In: Commun. ACM, vol. 22, no. 11, pp. 612–613 (1979).
2. Rabin, M.O.: "Efficient dispersal of information for security, load balancing, and fault tolerance". In: Journal of The ACM 36(2), pp. 335–348 (1989).
3. D. Zage and J. Obert. "Utilizing linear subspaces to improve cloud security". In Dependable Systems and Networks Workshops (DSN-W), 2012 IEEE/IFIP 42nd International Conference on, pages 1–6, June 2012.
4. S. Huchton, G. Xie, and R. Beverly. "Building and evaluating a k resilient mobile distributed file system resistant to device compromise". In Proceedings of the Military Communications Conference, Nov. 2011.
5. C. Gentry, "Fully homomorphic encryption using ideal lattices"., in Proceedings of the 41st annual ACM symposium on Theory of computing., 2009, p. STOC 09, New York, NY, USA, ACM 169-178.
6. O. Chor, B., Kushilevitz, E., and Goldreich, "Private information retrieval"., ACM, vol. 45, no. 6, pp. 965–981., 1998.
7. G. R. Blakley, "Safeguarding cryptographic keys"., in National Computer Conference, 1979, pp. 313–317.
8. Bruice Schneier, "Applied Cryptography- Protocols, Algorithms & Source code in C", Wiley India Pvt Ltd, ISBN 978-81-265-1368-0.
9. Z. Chen, "An Efficient and Secure Splitting Algorithm for Distributed Storage Systems"., Chain Commun., vol.7, no. 4, pp. 89–95, 2010.

10. J. L. Dautrich and C. V Ravishankar, "Security Limitations of Using Secret Sharing for Data Outsourcing", pp. 145–160, 2012.
11. S. J. Nirmala, S. M. S. Bhanu, and A. A. Patel, "A comparative study of the secret sharing algorithms for secure data in the cloud," vol. 2, no. 4, pp. 63–71, 2012
12. N. Gohring, "Amazon"s S3 down for several hours," Online at http:// www.pcworld.com/ businesscenter/ article/142549/,amazons_s3_down_for_severalhours.html, 2008
13. M. Arrington, "Gmail Disaster: Reports of Mass Email Deletions," Dec. 2006; http://www.techcrunch.com/2006/12/28/gmail-disasterreports ofmassemail-deletions/