ISSN:0975-9646

DCACrypt-A Design of Privacy Preserving in Cloud Storage

Anjul K. S. Rai
M. Tech Scholar
NRI Institute of Information Science and Technology,Bhopal

Dr. Samidha D. Sharma
Head of Department
NRI Institute of Information Science and Technology,Bhopal

Abstract: This article describe how DCACrypt(Divide and Conquer with AES Cryptography) can be deployed for privacy control in cloud Storage. The DCACrypt Split data into blocks and distribute these blocks randomly between multiple cloud Storage provider. The information of location of blocks and relationship between them is kept in an encrypted file using user key. This process will do on user machine this will help prevent cloud machine to overload. And reduced Trust on cloud server provider Due to this individual cloud storage provider cannot disclose any meaningful information, and hence DCACrypt is able to ensure the privacy of user data in the cloud. We present a formal model and simulation result.

Keywords: Cloud Computing, Divide and Conquer, AES, Cryptography, Privacy

1 INTRODUCTION

Extensive analysis efforts are placed on cloud computing and its related technologies, leading to many well acknowledged cloud computing theories and technologies. Cloud computing has been defined as the use of a collection of distributed application, services, information and infrastructure of pools of servers, network and storage. These elements are often quickly arranged, provisioned, enforced and decommissioned using an on request utility like model of sharing and consumption. Basic Cloud computing service models[1] are Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). In 2012 ITU (International Telecommunication Union) officially added two more basic cloud services[2] and these are Network as a service (NaaS) and Communication as a service (CaaS).

Due to the continuous growth in the quantity of digital data that has to be stored, there's a clear incentive for the service providers to explore outsourcing of users' information to the cloud. Potentially there can be many benefits to storing information in the cloud such as availability of services and data, infrastructure flexibility, faster deployment of applications and data, improved throughput, price management, cloud adaptation for actual necessities, etc. The cloud will give scalable superior storage design, and may facilitate us to significantly cut back the value of maintenance of individual services. Since the public cloud is an open platform, and may be subjected to malicious attacks from both insiders and outsiders. Also cloud service provider outsource customer data or analyse without know to customer due to this necessity to safeguard the privacy in the cloud becomes a crucial issue.

Privacy and security is major deterrents to adopting cloud computing. Customer need to be trust on the services provider, that means if the provider is evil the customer data may be used improperly without coming to notices of customer.

Customer and cloud service provider may be in a different jurisdiction. No information about physical location of customer data may worry to the customer. Personal data may be illegal to export in some jurisdiction[3][4]. When provider and customer are different jurisdiction or on a different continent, trust on cloud service provider is reduced.

In this article we explore a scheme where depending on trust will be reduce using divide and conquer approach. The remainder of the article is structured as follows: In Section 2 we outline related work for privacy and security. In section 3 we sketch our solution. In section 4 we present simulation result, outline future work in section 5 and in section 6 offer our conclusion.

2 RELATED WORK

Although practical privacy solutions for the cloud remain elusive, there is an ample body of relevant security knowledge to draw upon. In the following we provide a brief survey on the privacy.

2.1 Privacy through Policies

All major cloud provider and websites with user interaction currently provide privacy policies describing how personal information will be handled. But the fact is these policies are not recognized or read by user. The W3C reference Platform for Privacy Preferences (P3P) design[5] uses a mark-up language to allow cloud provide and websites to announce their privacy policy in a uniform fashion. But it is not guaranteed that services provider follow their privacy policies.

2.2 Privacy by Anonymity

Another way to protects One's privacy is to remain anonymous i. e. do not disclose and identifiable information this is often used in health sector. K-anonymity[6] is one approach to restrict semi-identifiable information such that at least K subjects share any combination. Other methods include general, pseudo and trap-door anonymity. Pseudo anonymity states to conditions where the identities of users are not real identity (e.g. usernames) and in that way deliver protection from things that do not know the link between the alias and the real identity. But some time benefits of services (e.g.

Tweeter, Google+, etc.) are reduces if identity is not provided.

2.3 Privacy by Regulation

Privacy also protected by law, any privacy policy should be according to privacy law in respected country or region. For example India privacy Act provides penalty for breach of confidentiality and privacy[7][8]. Same as EU directives on privacy protection[3] place requirements on member states legislation as to how personal information is stored, controlled and shared. But these regulation failed when government surveillance program like PRISM (USA)[9], CMS (India Surveillance Program)[10].

2.4 Privacy by method

Cryptography is the most important technological from many years. It protects data against interception to a very high degree. There are different types of encryption method such as AES, DES, RSA, etc. Cryptography is the best way to protect data but in some country the use of encryption is limited or restricted or requires a license and can be used in development of software. For example India requires an import license for encryptors[11].

The Free Haven project[12] describes a collaborative distributed storage system, where participants are allowed to store (or publish) data by offering to store data for others, in the same general fashion of peer-to-peer file sharing. Mnemosyne [13] offers steganography storage which not only hides data, but also prevents anyone from determining that there is anything hidden in the first place. Bur this scheme encrypts each block and thus required key managing system.

The POND[14] system is also based on distributed storage, but is not make sure anonymity of the different users. The ShareMind framework [15] offers distributed privacy-preserving a computations, based on the principles of secure multiparty computations. ShareMind is not focused on (anonymous) storage.

Author in [16] proposed a scheme for 'n out of m' secret sharing of information but it does not provide any algorithm for the actual splitting the data. [17] proposed another 'n out of m' scheme with concept of quality of services but this solution not for commercial cloud provider. Again [18] proposed n-out-of- scheme. RACS [19] proposed scheme to prevents vendor lock-in and data loss over failures by acting striping of data between several cloud providers. But it does not deliver privacy or confidentiality.

Jaatun et. al.[20] Proposed RAIN approach in which data split into segments and distributes segments between multiple storage providers using a separate command and control (C&C) provider, and distribution of segments and connection among the distributed segments private. This solution is pure cloud based, as number of user will increase load on C&C and user need to be trust on C&C. there is another problem of indexing and searching.

3 PROPOSED WORK

We developed a new approach where data is split into block and these blocks store on many cloud storage by randomly. Show that network observer cannot find which block is sent on which cloud storage. Splitting of data done on two basis. i) Fixed size of data block. Block size very from 1kb to 100kb depends on file type that means number of block will change as size and type of file changed. ii) Fixed number of block depends on file size and type size of block is change as size of file. Figure 1 show the whole process of proposed method.

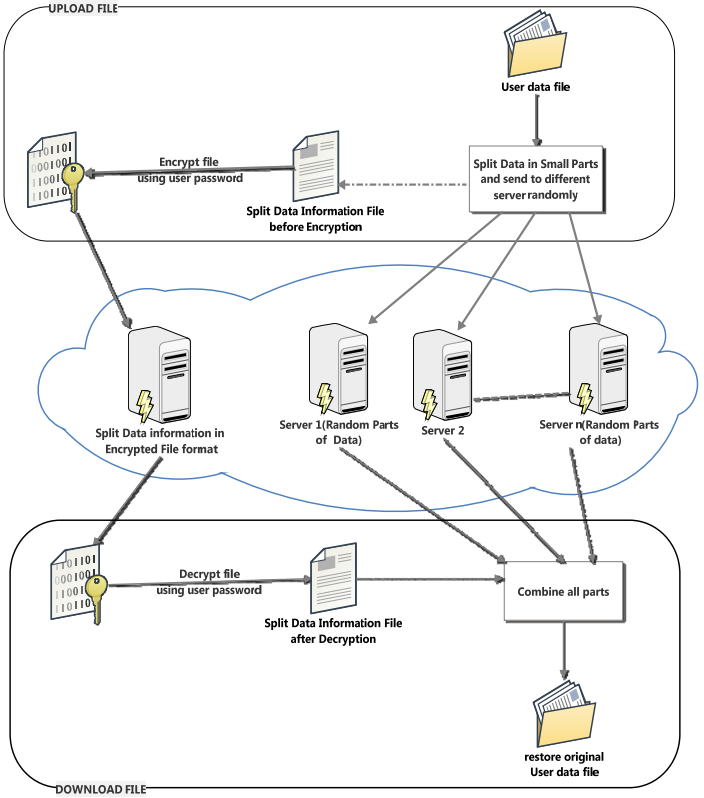

Figure 1 Proposed DCACrypt

3.1 Assumption

DC solution makes the following security assumptions

1. Communication between client and cloud storage is encrypted (SSL, etc.)
2. Cloud storage provider automatically backup each block to backup server.
3. There are enough simultaneous users to make anonymity feasible.
4. We have lightweight authentication mechanism using pool of session key.
5. Client machine is free from malware.
6. Provider will not able to link two different blocks of the same dataset.
7. Integrity of data block does not harm by cloud provide.

3.2 Data Splitting

Data Split of file is based on Divide and conquer approach(DC) in which problem (data file) is divided into small sub problem (small blocks) and applies operations on small problem. here are the overview of DC approach.

- Develops the output directly, for small requests
- Splits large requests to smaller ones, and (recursively) applies the procedure on the smaller requests.
- Combines the results for the sub requests, to produce a result for the original request.

On assumption of the sub problems can be solved independently. For fixed number of block the size of block can be calculated by

$$s = \frac{S}{R} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 1$$

Where 's' is size of block, 'S' is size of data file and 'R' is number of blocks to be makes. As 'R' is high value of 's' decrease and confidentiality is increase. But complexity of system is increase so user needs to be choosing 'R' such a way that confidentiality and complexity is not compromised.

A most simple way to blocking of data is a progressive division, by which the original data is treated as a binary stream and is divided into multiple sub stream. Second way to blocking is treated a data as image and cut the data in two dimensional space.

3.3 Uploading of blocks

Uploading of blocks on multiple cloud storages are randomly. Each block assigns a unique name using UUID function plus a five digit random authentication code using rand function so that no other than authentic user can access data blocks.

3.4 Information of blocks

Information of block position on cloud storage (address to access the block) and relation between them (position in main data) is kept in a simple encoded file. This file is encrypted with AES 256[21] with user key and keep this encrypted file any place where user can access it.

3.5 Data re-assembling

Assembling of original data required two pieces of information

1. Encrypted file associated with original file.
2. Key for decrypting of file.
3.

4 SECURITY ANALYSIS

Evil users can either reconstruct the complete data file or make attempt to decrypt the encrypted file using they can access to original file.

4.1 Compromisation by reconstructing

The re-assembling involves two steps as follows.

1. Gathering all related data block form large data storage
2. Placing the data blocks at proper position.

Supposing that malicious user have access to all cloud storage that containing blocks let there is 'm' number of cloud storage. Each data storage contains 'n_i' number of block and each cloud storage have 'k_i' number of blocks of original data. Then Probability of getting actual block from each storage P_i

$$P_i = \frac{1}{C_{n_i}^{k_i}} \qquad 1 \le i \le m \quad \ldots\ldots\ldots\ldots\ldots\ldots 2$$

Probability of getting all blocks from all 'm' storage

$$P = (P_1)(P_2).....(P_{m-1})(P_m) \quad \ldots\ldots\ldots\ldots\ldots\ldots 3$$

4.2 Compromisation by Decryption

To read information file it need to decrypt. Decryption of AES encrypted file required cryptanalysis. Following table show cryptanalysis method on AES and its computation operation.

Table 1 Cryptanalysis method Operation on AES

Cryptanalysis Method	Required No. of Operation
Brute-Force	1.1×10^{77}
Related-key attack[22]	$2^{99.5}$
Biclique Cryptanalysis[23]	$2^{254.4}$

It is clearly visible that decryption required very high computation power as well as long time.

5 SIMULATION ENVIRONMENT

In this section we describe our work to simulate the proposed method. We implement our proposed method on java language. Java is very flexible and cross platform language. We use Java Enterprise Edition (JavaEE7) on Netbeans IDE. We use CloudBees as cloud service provider for storing data in cloud network

For light weight file upload client we use Awake-library and for AES we used AESCrypt module. Table 2 show the simulation environment and default value

Table 2 Simulation settings

Platform	Java EE 7
Cloud Provider	CloudBees.com
No. of Server Instance	2
No of blocks per file	64
Type of data file	Image(JPG)
Cloud Resource Allocation	128MB RAM/ Instance

6 EXPERIMENTAL RESULTS AND ANALYSIS

This section provides result and analysis of our experiment. We test different size of data file for both uploading and downloading may times. Than calculate average valve of each size of data file. Table 3 show the result of our experiment

Table 3 Experimental Result

Data (MB)	Upload (ms)	Download (ms)
1	2446	1089
1.2	2456	1200
2.4	2566	1312
6.5	4438	1859

It is clearly visible that as size of file increase computation time is increase. But most of the data file for normal user or small business users is limited to 5 MB. We compare our DCACrypt with FADE[24] in which pure AES is used. We get that our approach work very well with 1MB of data file

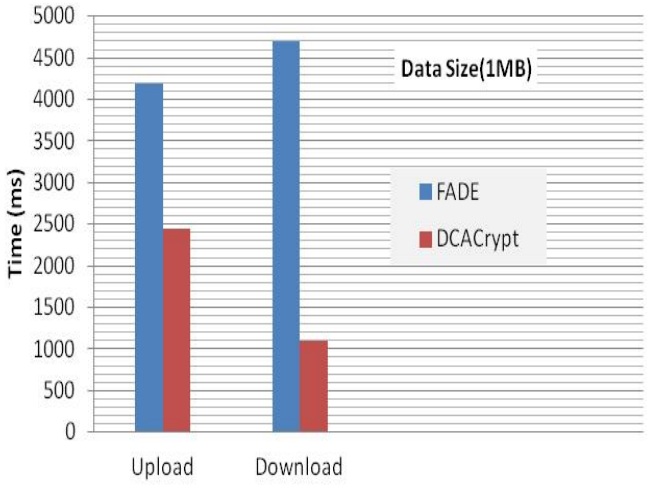

Figure 2 DCACrypt V/S FADE

7 DISCUSSION

We remove C&C node as describe in [20] and load of C&C node is distribute to clients. as we use client computation power, user need to pay less to cloud provider compared to [20]. Searching may be possible if proper index is possible to create, it may be store with encrypted information file. Size of sensitive information can be very small. In these conditions proposed approach may not be suitable. DCACrypt is suitable for sensitive but not classified data. We have not measured load balancing for data storage in cloud storage.

8 FURTHER WORK

Searching and indexing is not provided with our solution but further work can support it .further refinement of our approach to determine the optimal block size policy that can make balance between privacy and efficiency for random data.

CONCLUSIONS

We propose a cloud storage system based on DCACrypt which aim to provide confidentiality. We present a formal model of our approach and implement a prototype of DCACrypt to demonstrate it practically. Our experimental result provides that our approach is acceptable. And this may be used by Home and small business user.

REFERENCES

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology," *NIST Spec. Publ. 800-145*.
[2] "Focus Group on Cloud Computing Technical Report Part 1," 2012.
[3] "European Parliament (1995) Directive 95/46/EC," 1995.
[4] "New draft European data protection regime." [Online]. Available: http://www.mlawgroup.de/news/publications/detail.php?we_objectID=227. [Accessed: 11-Sep-2013].
[5] "The Platform for Privacy Preferences 1.1 (P3P1.1) Specification." [Online]. Available: http://www.w3.org/TR/P3P11/. [Accessed: 05-Sep-2013].
[6] L. SWEENEY, "k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY," *Int. J. Uncertainty, Fuzziness Knowledge-Based Syst.*, vol. 10, no. 5, pp. 557–570, Oct. 2002.
[7] "THE INFORMATION TECHNOLOGY ACT, 2000," INDIA, 2000.
[8] "THE IT (AMENDMENT) ACT 2008." MINISTRY OF LAW AND JUSTICS INDIA, 2009.
[9] "Everything you need to know about PRISM | The Verge." [Online]. Available: http://www.theverge.com/2013/7/17/4517480/nsa-spying-prism-surveillance-cheat-sheet. [Accessed: 16-Oct-2013].
[10] "India's Surveillance Program Stalled - India Real Time - WSJ." [Online]. Available: http://blogs.wsj.com/indiarealtime/2013/07/06/indias-surveillance-program-stalled/. [Accessed: 16-Oct-2013].
[11] "Crypto Law Survey." [Online]. Available: http://www.cryptolaw.org/. [Accessed: 15-Sep-2013].
[12] S. D. Haven and R. R. Dingledine, "The Free Haven Project: Design and Deployment of an Anonymous," in *Master's thesis, MIT*, 2000.
[13] S. Hand and T. Roscoe, "Mnemosyne: Peer-to-Peer Steganographic Storage," in in *Peer-to-Peer Systems*, vol. 2429, P. Druschel, F. Kaashoek, and A. Rowstron, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 130–140.
[14] S. Rhea, P. Eaton, D. Geels, H. Weatherspoon, B. Zhao, and J. Kubiatowicz, "Pond: the OceanStore prototype," in *Proceedings of the 2nd USENIX Conference on File and Storage Technologies*, 2003, pp. 1–14.
[15] D. Bogdanov, S. Laur, and J. Willemson, "Sharemind: A Framework for Fast Privacy-Preserving Computations," in in *Computer Security - ESORICS 2008*, vol. 5283, S. Jajodia and J. Lopez, Eds. Springer Berlin / Heidelberg, 2008, pp. 192–206.
[16] Y. Singh and F. Kandah, "A secured cost-effective multi-cloud storage in cloud computing," in *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2011, pp. 619–624.
[17] J. Luna, M. Flouris, M. Marazakis, and A. Bilas, "Providing security to the Desktop Data Grid," in *2008 IEEE International Symposium on Parallel and Distributed Processing*, 2008, pp. 1–8.
[18] A. Parakh and S. Kak, "Online data storage using implicit security," *Inf. Sci. (Ny).*, vol. 179, no. 19, pp. 3323–3331, Sep. 2009.
[19] H. Abu-Libdeh, L. Princehouse, and H. Weatherspoon, "RACS: A Case for Cloud Storage Diversity," in *Proceedings of the 1st ACM symposium on Cloud computing - SoCC '10*, 2010, pp. 229–240.
[20] M. Jaatun, G. Zhao, and A. Vasilakos, "The design of a redundant array of independent net-storages for improved confidentiality in cloud computing," *J. Cloud Comput.*, vol. 1, no. 1, p. 13, 2012.
[21] J. Daemen and V. Rijmen, *AES Proposal: Rijndael*. National Institute of Standards and Technology, 2003.
[22] Biryukov, Alex and A. Biryukov, *Related-key Cryptanalysis of the Full AES-192 and AES-256*. 2009.
[23] D. H. Lee and X. Wang, Eds., "Biclique Cryptanalysis of the Full AES," in in *Advances in Cryptology – ASIACRYPT 2011*, vol. 7073, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 344–371.
[24] Y. Tang, P. Lee, J. Lui, and R. Perlman, "FADE: Secure Overlay Cloud Storage with File Assured Deletion," in in *Security and Privacy in …*, vol. 50, S. Jajodia and J. Zhou, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 380–397.