Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 5, May 2014, pg.608 – 616

RESEARCH ARTICLE

Data Protection Outsourcing of Cloud Data to Maintain Trust between Cloud Service and Data Owner Using RC5 Algorithm

Mr. Ajay Bhaisare
Department of CSE, GHRAET
Nagpur (M.S), India,
ajaybhaisare8@gmail.com

Prof. Ashwini Meshram
Department of CSE, GHRAET
Nagpur (M.S.), India
ashwini.meshram@raisoni.net

Abstract – Cloud Service Provider (CSP) provides various types of services .Such as Storage-as-a-Service (SaaS) is a paid facility offered by Cloud Service Provider (CSP), where data owners can outsource their confidential data in the cloud. But there having some problem of ensuring the integrity and security issue of data storage in Cloud. We take the work of allow a Trusted Third Party (TTP), on behalf of the cloud client, to check the security and integrity of the dynamic data stored in the cloud. The data owner securely outsources personal/confidential data in cloud. It allows authoritative users to access the owner's data/file. It keeps going trust between data owner and cloud service provider (CSP).

Key Words— Encryption, Access control, Dynamic environment, outsources data storage

I. INTRODUCTION

Cloud computing is the long dreamed vision of computing as a function, where data owners can slightly store their data in the cloud to benefit from high-quality on-demand services and applications from a shared group of configurable computing resources. While store data in cloud relieves the owners of the burden of local data maintenance and storage, it also discards their physical control of storage security and dependability, which traditionally has been expected by both single/individual and enterprises/organization with high service level requirements. In the current era of digital world, various enterprises/organizations produce a huge amount of confidential/sensitive data including personal information, financial data, records and electronic health. The local organization of such vast amount of data is costly and problematic due to the requirements of high storage capacity and qualified personnel. Therefore, Storage Service provided by cloud service providers (CSPs) come out as a solution to soften the burden of maximum local data storage and minimize the maintenance cost by means of outsourcing data storage [1]-[2]. We consider a scenario in which a huge data, sensitive file's is to be stored securely over cloud storage domain. These storage domain are not reliable in the sense that an attacker may gain access to some of them, but not to all [3].

Usually, traditional access control methods take up the existence of the data vendor and the storage servers in the same trust organization. This statement, conversely, no longer holds when the data is outsourced to a cloud storage which offered by CSP, which takes the full trust of the outsourced data management of owner data, and resides exterior the trust organization of the data owner. A possible solution can be view to enable the owner to inflict access control of the data stored on outsource un-trusted CSP. Through this resolution, the data is encrypted below a certain key, which is communal only with the authoritative users. The

un-authoritative users, including the Cloud Service Provider, are not capable to access the data ever since they do not have the decryption key. This common solution has been broadly incorporated into existing schemes [8]–[9], which take aim at offering data storage security on un-trusted outsource servers. One more class of solutions used attribute-based encryption to get fine-grained access control [10], [11].

Another approach has been investigated that stimuli the owner to remotely store the data and offer certain sort of ensure related to the integrity, access control, and confidentiality of the outsourced data. This approach can prevent and discover malicious measures from the side of CSP. On the other hand CSP needs to be defended from an un-trusted owner, who tries to get illegal corrections by wrongly claiming data fraud over cloud servers. This bother, if not accurately handled, can trigger the CSP through out of business [12].

In order to get the assurance of cloud data integrity and availability and put into effect the quality of cloud storage service from CSP, efficient methods that enable on-demand data correctness verification on behalf of cloud users have to be designed [4]. Security audit is an important solution enabling trace back and analysis of any activities including application activities, security breaches, data accesses, and so on. To ensure there is no attack to compromise the security of verification protocol or cryptosystem by using dynamic data operations [5]. We provide the security to owner's data by using RC5 algorithm& we use SHA & BST(Block Status Table), SHA for computing Hash value of a file that will check data integrity & BST which will used for block level operation that keep track of file update like append, modify, insert & delete [14].

A disadvantage of using public key cryptography for encryption process is speed. There are many secret key encryption functions that are significantly faster than any currently available public-key encryption function. Key generation can be slow. RSA operations are slower than similar symmetric key operations and also there need two prime number for generate key [14].

In this method will allow every data owners to store their data over cloud server/storage with confidence. It will minimize their bother of maintaining this data on local server. Even the confidentiality and integrity of data is maintained.

- It allows the owner of data to outsource sensitive/confidential data to a Cloud Service Provider, and perform full block level dynamic operations on the cloud data, i.e., block deletion, modification, append and insertion.
- It ensures that authorized users (i.e., those who have the right to access the owner's file) receive the latest version of the outsourced data.
- It enables indirect mutual trust between the owner and the CSP.
- It allows the owner to grant or revoke access to the cloud data. We discuss the security problem of the proposed scheme. Also, we justify its performance through imaginary analysis and a prototype implementation on Amazon cloud platform to calculate storage, communication, and computation outlay [14].

Main Contributions:

- The implementation and design of a cloud-based storage scheme that has the following features: (i) It allows to data owner to store data to a CSP, and perform full lively operations at the block-level based, i.e., it supports operations likewise block deletion, insertion, append, and modification; (ii) It ensures the newest property, i.e., the authorized user's get the most recent version of the stored data; (iii) It make indirect trust between the data owner and the Cloud Service Provider since every one party resides in a distinct trust domain; and (iv) It inflict the access control for the store data in cloud.

- We talk about the security features of the planned scheme. Besides, we validate its performance through prototype implementation and theoretical analysis a on Amazon cloud platform to evaluate communication, computation overheads, and storage.

System component and Assumption:

The cloud storage structure consider in this work made of four components as describe in Fig. 1: (i) A owner of data that can be an any Company/organization creates sensitive or confidential data to be outsource in cloud storage and produce available for controlled outside use, (ii) A Cloud Service Provider(CSP) who handle cloud server and grant storage space on its cloud server to store the data and made them available for users, (iii) The group of authorized user of data owner's clients those have the authority of accessing the data from cloud storage; and (iv) A Trusted Third Party (TTP), an unit trusted by data owner, and has abilities to detect unauthorized parties.

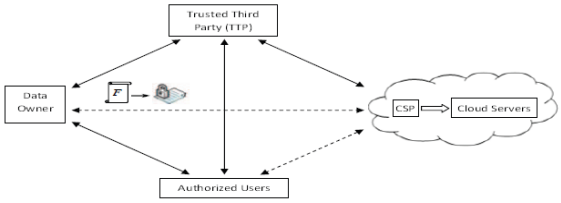


Fig. 1: Component and Cloud Storage

- **RC5 Algorithm**

RC5 has a variable block size (32, 64 or 128 bits), key size (0 to 2040 bits) and number of rounds (0 to 255). The first suggested choices of parameters were a block size of 64 bits, and a 128-bit key and 12 rounds of encryption/decryption. A key

feature of RC5 algorithm is the use of data-dependent rotations; unique goals of RC5 were to provide the study and estimate of such operations as a cryptographic primitive. RC5 algorithm also consists of a number of modular eXclusive OR (XOR)'s and additions. The general model of the algorithm is a Feistel-like network. The RC5 encryption and decryption routines can be stated in a few lines of code. The key schedule, however, expanding the key using an essentially one-way method, is more complex, with the binary expansions of both e and the golden ratio of "nothing up my sleeve number's". The simplicity of the RC5 algorithm with the experience of the data contingent rotations has made RC5 an attractive object of study for cryptanalysts. The RC5 algorithm is simply denoted as RC5-w/r/b where r=number of rounds, b=number of 8-bit byte in the key, w=word size in bits [6].

Here we describe the RC5 algorithm, which consists of three main components: a key expansion algorithm an encryption and decryption algorithm. We demonstrate the RC5 encryption and decryption algorithms first.

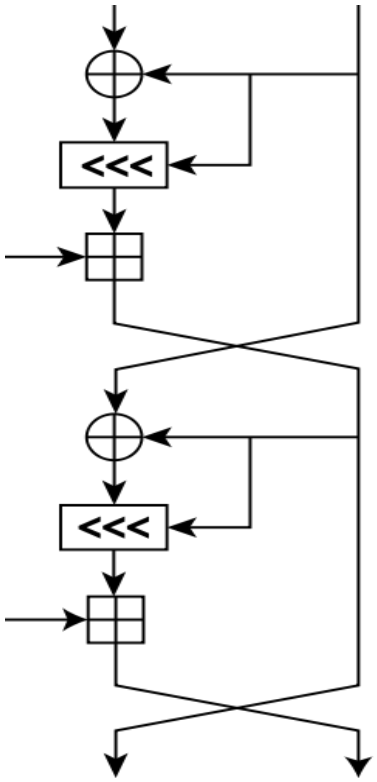


Fig. 2: one round (two half-rounds) of the RC5 block

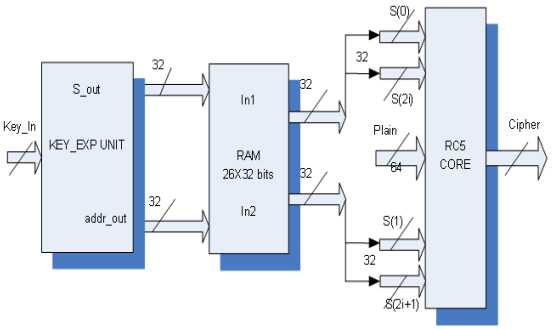


Fig. 3: Architecture of the RC5

Remembering that the plaintext input to RC5 algorithm that consists of two w-bit words, which we denote A and B. Remembering also that RC5 algorithm uses an expanded key table S[0…t-1], consisting of t = 2(r + 1) w-bit word's. The key expansion RC5 algorithm initializes S from the user's given secret key parameter K[6]. (We note that the S table in RC5 encryption algorithm is not an "S-box" such as is used by DES; RC5 algorithm uses the entries in S sequentially one at a time.)

The standard little-endian conventions for packing bytes into a input or output blocks: the first byte fill space the low-order bit locations of register A, and so on, so that the fourth byte fill space the high-order bit locations in A, the fifth byte fill space the low-order bit locations in B, and the eighth (last) byte fill space the high-order bit locations in B.

Encryption:

We take assume that the input block of RC5 is given in two w-bit registers A and B. We also assume that key-expansion of RC5 has already been performed, so that the array S [0…t-1] has been computed. Here is the encryption algorithm in pseudo - code:

A = A + S [0];
B = B + S [1];
for i = 1 **to** r **do**
A = ((A⊕ B) <<< B) + S[2 * i];
B = ((B⊕ A) <<< A) + S[2 * i + 1];

The output is in the registers A and B.

We note the special simplicity of this 5-line algorithm. We also note that each round of RC5 updates both registers A and B; where as a "round" in DES updates only half of its registers. A "half-round" of RC5 algorithm (one of the assignment statements updating A or B in the body of the loop above) is thus perhaps more analogous to a DES round.

Decryption:

The decryption routine is easily derived from the encryption routine.

for i = r **downto** 1 **do**
B = ((B – S[2 * i + 1]) >>> A)⊕ A;
A = ((A – S[2 * i]) >>> B)⊕ B;

B = B – S[1];
A = A – S[0];

Key Expansion:

The key-expansion routine expands the user's secret key K to fill the expanded key array S, so that S resembles an array of t = 2(r + 1) random binary words determined by K. The key expansion algorithm uses two "magic constants," and consists of three simple algorithmic parts [7].

Secure Hash Algorithm (SHA)

The SHA is a group of cryptographic hash method published by the NIST (National Institute of Standards and Technology) as a U.S. FIPS (Federal Information Processing Standard), and may refer to [13]:

- **SHA-0**: A retronym applied to the real version of the 160-bit hash function published in 1993 in the name "SHA". It was take out shortly after publication due to an unreleased "significant flaw" and restored by the slightly revised version SHA-1.
- **SHA-1**: A 160-bit hash function which look like the earlier MD5 algorithm. This was designed by the NSA (National Security Agency) to be part of the Digital Signature Algorithm. Cryptographic fault were discovered in SHA-1, and the standard was no longer approved for most cryptographic uses after 2010.
- **SHA-2**: A family of two alike hash functions, with different block sizes, known as SHA-256 and SHA-512. They differ in the word size; SHA-256 uses 32-bit words where SHA-512 uses 64-bit words. There are also truncated versions of each standard, known as SHA-224 and SHA-384. These were also designed by the NSA.
- **SHA-3**: A hash function formerly called Keccak, chosen in 2012 after a public competition among non-NSA designers. It supports the same hash lengths as SHA-2, and its internal structure differs significantly from the rest of the SHA family [13].

II. IMPLIMENTATION DETAIL

The working environment for secure outsourcing cloud data with proposed algorithm is implemented using PHP with Dreamweaver for coding as frontend and MySQL XAMPP for the database.

The implementation is achieved through modular approach for securely outsource cloud data as a case study. These 3 modules are stated below:

- Data Owner Module

The Data Owner is outsourcing data over cloud, the data that will be passed to the CSP. The Data Owner passes the Data file via TTP where it will be verified, and calculate Hash value of file for integrity.

- TTP Module

The TTP acts as a mediator between Data Owner and CSP. He store Master Key and Calculate Hash value of file and send encrypted file to CSP.

- CSP Module

Here only encrypted file are display and he give access permission to only authorised data owner via TTP.

UML diagrams of the Implementation

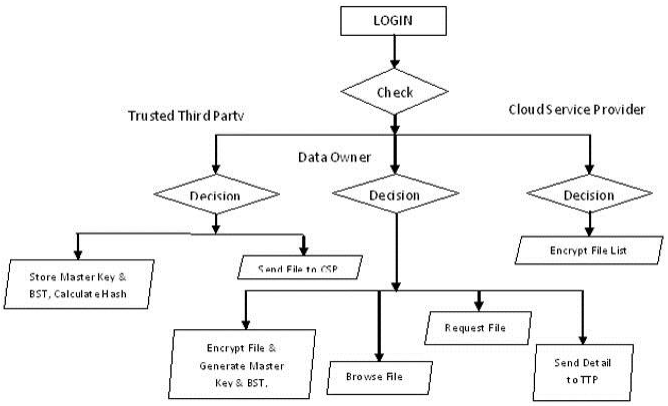
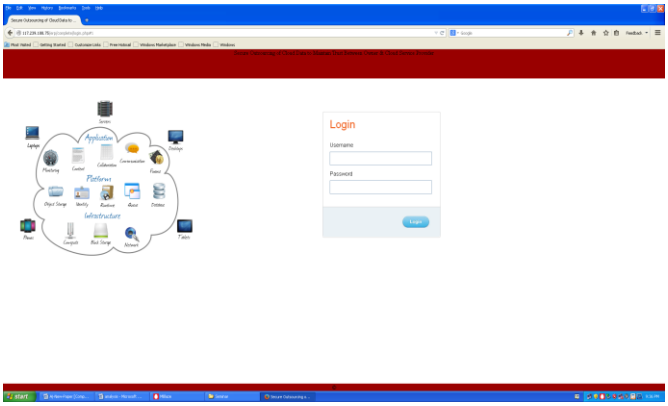


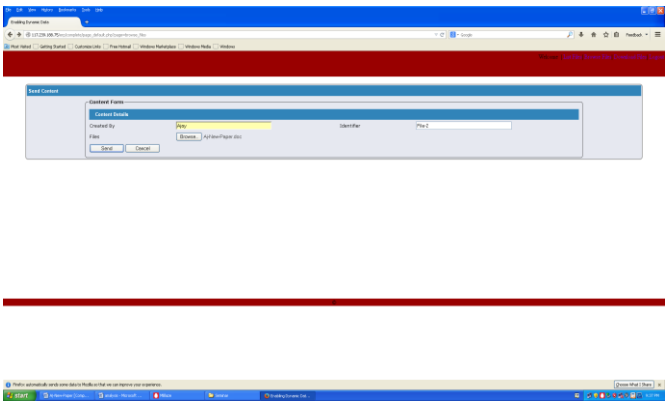
Fig: Data Flow Diagram

Login Page:

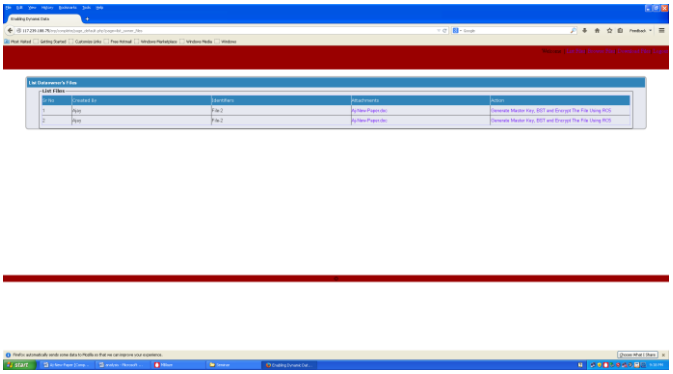


This is a login page for Data Owner, Trusted Third Party (TTP) and Cloud Service Provider (CSP).

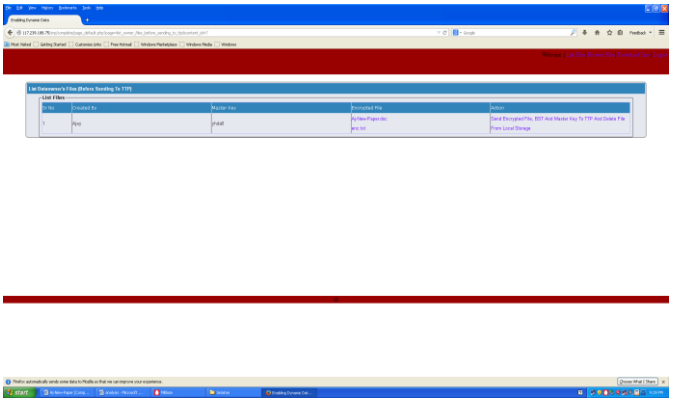
DATA OWNER
Browse File Page:



Owner browse file which he want to upload over cloud.

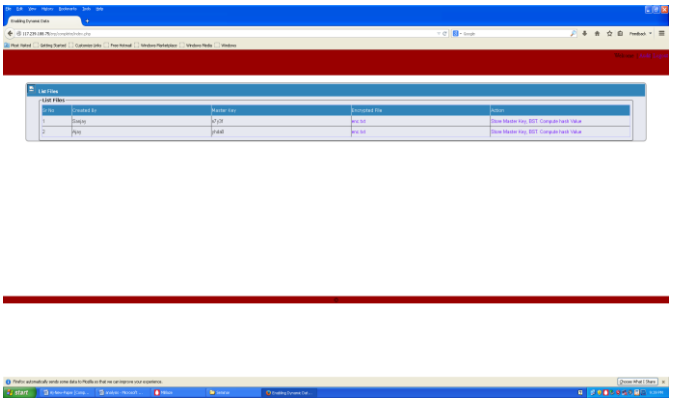


After successful upload file Generate Master or Public key, Block Status Table (BST) and Encrypt file by RC5 private Key.

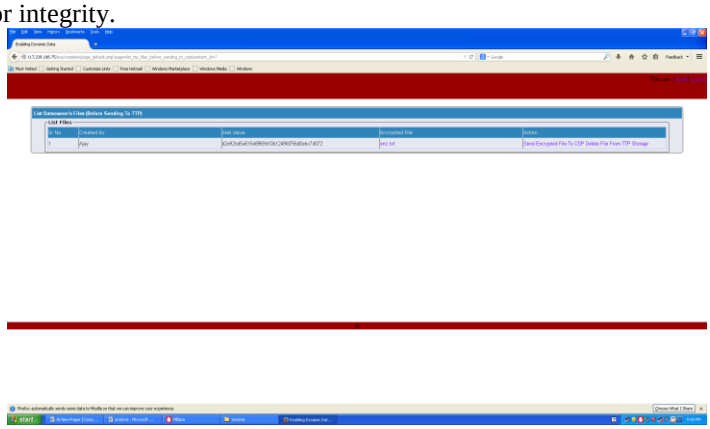


Now send Master Key, BST and Encrypted file to TTP and Delete file from Local Storage.

TRUSTED THIRD PARTY
File List Page:

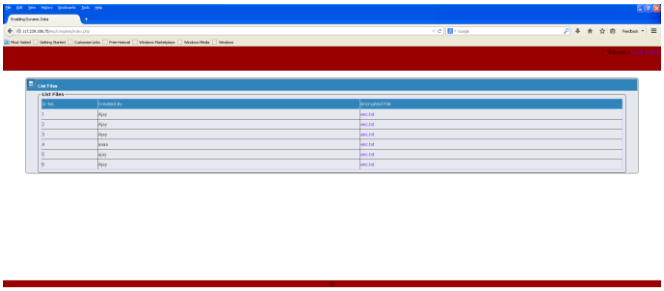


List of files send by Data Owner to TTP. Now, here attribute show File Created By, Master Key and Encrypted file. The TTP do not read the file by using Master Key because for reading the file he need private key also. He store Master Key, BST and Compute Hash value of a file for integrity.

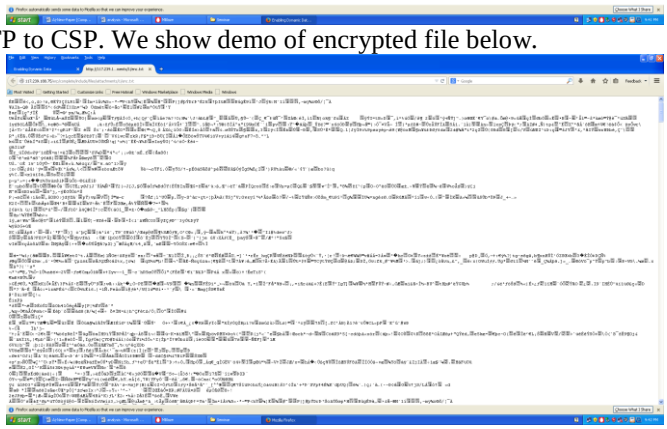


After that he send file to Cloud Service Provider.

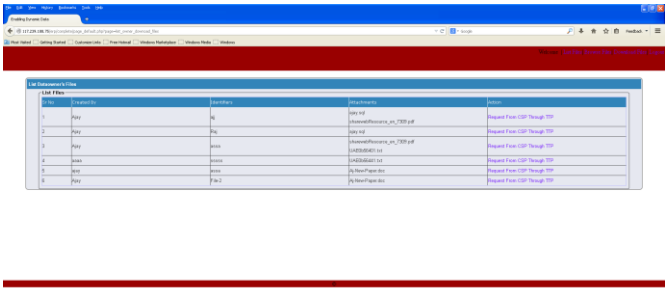
CLOUD SERVICE PROVIDER



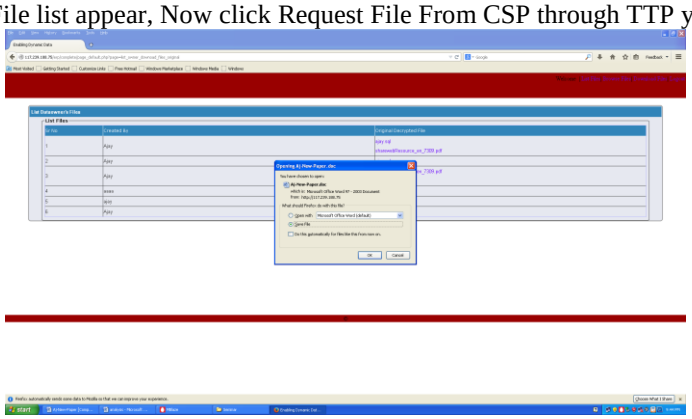
List of encrypted files send by TTP to CSP. We show demo of encrypted file below.



DATA OWNER REQUEST FOR FILE



When click on download abow File list appear, Now click Request File From CSP through TTP you get your decrypted file.



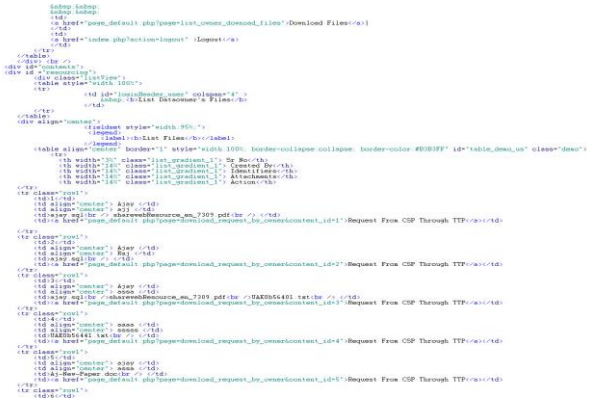


Fig: Code showing stored request file for owner

III. EXPERIMENTAL RESULT

In this paper we analysis the speed, security and trust between data owner and CSP on the basis of security as well as RC5 algorithm. We provide a module that maintains the speed and trust, below we show comparison between RSA and RC5 algorithm.

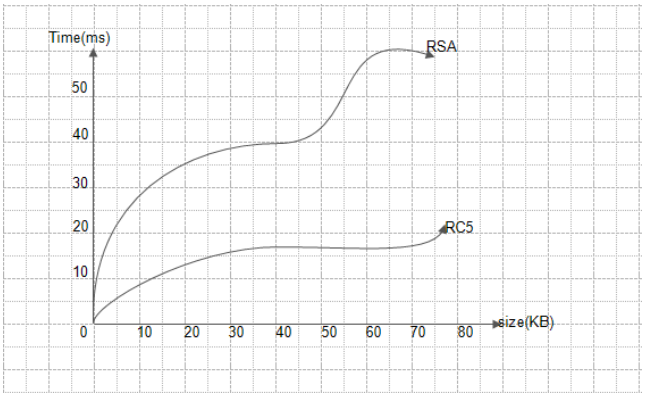


Fig:-graph comparison RSA and RC5 algorithm

The graph show the time required for encryption of file having different size with RSA and RC5 algorithm.

File & Size	RSA	RC5
sharewebResource_en_7309.pdf (76.4 KB)	55 ms	23 ms
UAE0b56401.txt (2 KB)	12 ms	2.3 ms
Ajay.sql (40KB)	40ms	19 ms
Throughput (average)	**35.66 ms**	**14.76s**

IV. RESULT DESCRIPTION

In these paper we propose four important component's/ module's: first O-Module (Owner module), second C-Module (CSP module), third A-Module (Authorized user module), and forth one is T-Module (TTP module). O-Module that runs in the side of data owner is a documentation which is used by the owner to carry out the owner function in the system and file training phase. Furthermore, this documentation is used by the owner at some stage in the dynamic operations on the cloud data.

V. CONCLUSION

The owner of data can't only archiving and accessing the data stored by the Cloud Service Provider (CSP), but also scaling and manipulate this data on the cloud storage. The offered scheme enables the valid/right users to ensure that they are receiving the

recent copy of the cloud data. Furthermore, in case of dispute about data originality/integrity, a TTP is able to determine the lying party.

REFERENCES

[1] Ayad Barsoum and Anwar Hasan, "Enabling Dynamic Data and Indirect Mutual Trust for Cloud Computing Storage Systems," IEEE Transactions on parallel and distributed systems, 2013.
[2] Cong Wang and Kui Ren, "Toward Publicly Auditable Secure Cloud Data Storage Services", IEEE Network, August 2010.
[3] Paulo F. Oliveira, "Coding for Trusted Storage in Untrusted Networks," IEEE Transactions on Infor-mation Forensics and Security, Vol. 7, No. 6, December 2012.
[4] Cong Wang, Qian Wang, Kui Ren, Ning Cao, and Wenjing Lo, "Toward Secure and Dependable Storage Services in Cloud Computing," IEEE Transactions on Services Computing, Vol. 5, No. 2, June 2012.
[5] Yan Zhu, Gail-Joon Ahn, Hongxin Hu, Stephen S. Yau, Ho G. An, and Chang-Jun Hu, "Dynamic Audit Services for Outsourced Storages in Clouds," IEEE Transactions On Services Computing, Vol. 6, No. 2, June 2013.
[6] Ron Rivest, "From Wikipedia, the free encyclopedia", http://en.wikipedia.org/wiki/RC5, March 1997.
[7] Ronald L. Rivest "The RC5 Encryption Algorithm," http://theory.lcs.mit.edu/~rivest/Rivest-rc5rev.pdf, Revised March 20, 1997.
[8] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, "Plutus: Scalable secure file sharing on untrusted storage," in *Proceedings of the FAST 03: File and Storage Technologies*, 2003.
[9] S. D. C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "Over-encryption: Management of access control evolution on outsourced data," in *Proceedings of the 33rd International Conference on Very Large Data Bases*. ACM, 2007, pp. 123–134.
[10] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *CCS '06*, 2006, pp. 89–98.
[11] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *INFOCOM'10*, 2010, pp. 534–542.
[12] R. A. Popa, J. R. Lorch, D. Molnar, H. J. Wang, and L. Zhuang, "Enabling security in cloud storage SLAs with cloud proof," in *Proceedings of the 2011 USENIX conference*, 2011.
[13] United States National Security Agency and is a U.S. Federal Information Processing Standard published by the United States NIST, "From Wikipedia", http://en.wikipedia.org/wiki/Secure_Hash_Algorithm.
[14] Ajay Bhaisare, Prof. Prakash Prasad, Prof. Ashwini Meshram,"Solution to Data Sharing for Confidentiality between Service Provider and the Data Owner," International Journal of Computer Science and Mobile Computing, Vol.3 Issue.2, February- 2014.

