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. 2, February 2014, pg.575 – 579

RESEARCH ARTICLE

Solution to Data Sharing for Confidentiality between Service Provider and the Data Owner

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

Prof. Prakash Prasad
Dept of Information Technology
Nagpur (M.S.), India
prakashprasad@yahoo.com

Prof. Ashwini Meshram
Department of CSE, GHRAET
Nagpur (M.S.), India
rani11489@gmail.com

Abstract – Cloud Service Provider (CSP) provides various types of services. Such as Storage-as-a-Service (SaaS) is a paid facility provided by CSP, where data owners can outsource their data in the cloud. This having some issue of ensuring the integrity and security of data storage in Cloud. We consider the work of allowing a Trusted Third Party (TTP), on behalf of the cloud client, to verify the integrity and security of the dynamic data stored in the cloud. The data owner securely outsources confidential data in cloud. It allows authorized users to access the owner's file. It maintains trust between data owner and cloud service provider.

Keywords— Encryption, Access control, Dynamic environment, outsources data storage

I. INTRODUCTION

Cloud computing is the long dreamed vision of computing as a utility, where data owners can store their data in the cloud storage to enjoy required high-quality applications and services from a shared pool of configurable computing resources. While data store in cloud server relieves the owners of the worrying of local data storage and maintenance, it also free their physical control of storage security and dependability, which usually has been expected by both organization and singular with high service level requirements. In the current period of digital world, various organizations produce a large amount of sensitive data including personal/confidential information, electronic health records, and economic/financial data. The local management of such large amount of data is challenging and expensive due to the requirements of large storage capacity and qualified staff. Therefore, Storage-as-a-Service offered by cloud service providers (CSPs) emerged as a solution to mitigate the burden of large local data storage and reduce the maintenance cost by means of outsourcing data storage [1]-[2]. We consider a scenario in which a large, confidential file is to be stored securely over cloud networks. These networks are not trustworthy in the sense that an attacker may gain access to some of them, but not to all [3].

In order to achieve the assurances of cloud data integrity and availability and enforce the quality of cloud storage service, efficient techniques that enable required 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 data accesses, security breaches, service activities, 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.

System component and Assumption:

The cloud storage model consider in this work consists of four components as illustrated in Fig. 1: (i) A data owner that can be an any organization/Company generating sensitive data to be stored in cloud storage and make available for controlled external use, (ii) A Cloud service provider who manage cloud server and provides storage space on its service to store the owner's data and made them available for authorized users, (iii) authorized user's: group of owner's clients who have the authority to access the cloud storage data; and (iv) a Trusted Third Party (TTP), an unit who is trusted by data owner, and has capabilities to detect/specify unauthorized parties.

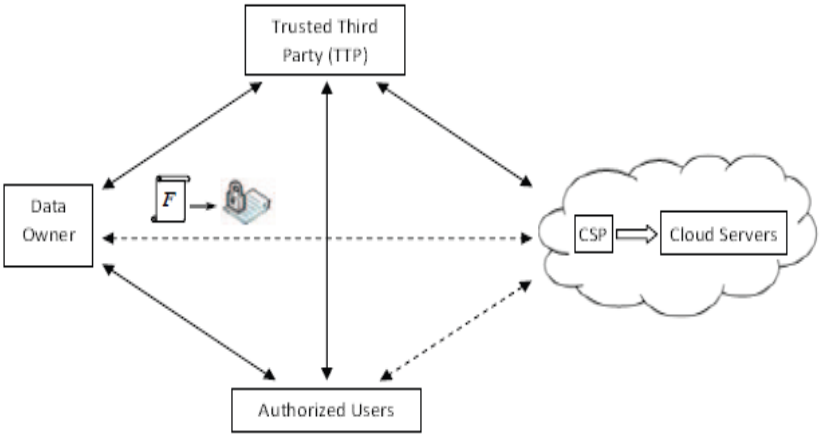

Fig. 1: Component and Cloud Storage

Security Requirement:

Confidentiality: Protect the cloud data from Cloud Service Provider (CSP), unauthorized user and Trusted Third Party (TTP) which have no permission.
Integrity: The cloud server data should remain undamaged i.e. the outsourced data.
Newness: The latest copy of outsourced data should receive.
Access Control: The outsource data access permission are granted to only authorized users.
Defense: The Cloud Service Provider has to be protected from fake allegation that might be claimed by lying owner / user.

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 original suggested choices of parameters were a block size of 64 bits, a 128-bit key and 12 rounds. A key feature of RC5 is the use of data-dependent rotations; one of the goals of RC5 was to prompt the study and evaluation of such operations as a crypto-graphic organelles. RC5 algorithm also consists of a number of eXclusive OR (XOR)'s and the modular's addition. The general arrangement of the RC5 algorithm is a Feistel as a network. The process of encryption and decryption repetitive can be specified in a few numbers of codes. The schedule of key, however, is more complicated, expanding the key using an necessary one path function with the binary growth of both e and the excellent ratio as sources of "nothing up along my sleeves numbers". The inviting ease of the RC5 algorithm together with the experience of the data contingent rotations has made RC5 an attractive goal of study for crypt-analysts. The RC5 is essentially denoted as RC5 - w/r/b where w = word size in bits, r = number of rounds, b = number of 8-bit byte in the key [6].

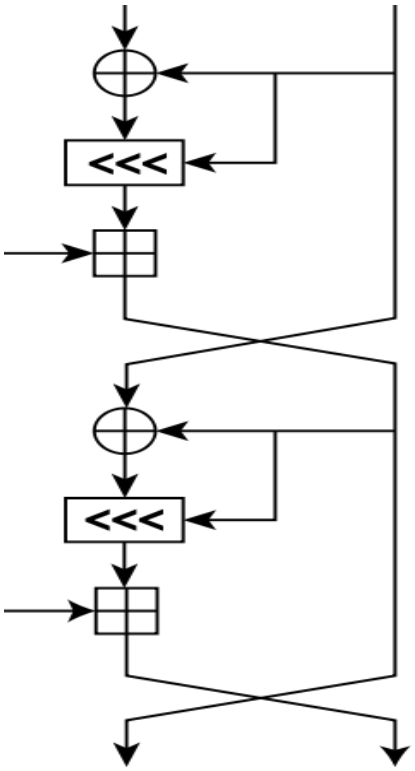

Fig. 2: one round (two half-rounds) of the RC5 block

In this part we brief the RC5 algorithm, which have three components: The RC5 key expansion algorithm for a RC5 encryption algorithm and RC5 decryption algorithm, we summaries the encryption & decryption algorithms first.

Recall the plaintext that input to RC5 algorithm consists of two w - bit words, which is denoted by A and B. Recall also that the RC5 algorithm uses an expanded key table S[0…t-1], consisting of t = 2(r + 1) w - bit words. The RC5 key expansion algorithm initialize S from the secret key parameter K which is given by users. (Note that the S table in RC5 encryption algorithm is not an "S - box" such as is used by DES algorithm; RC5 uses the entries in S sequentially, at a same time.)

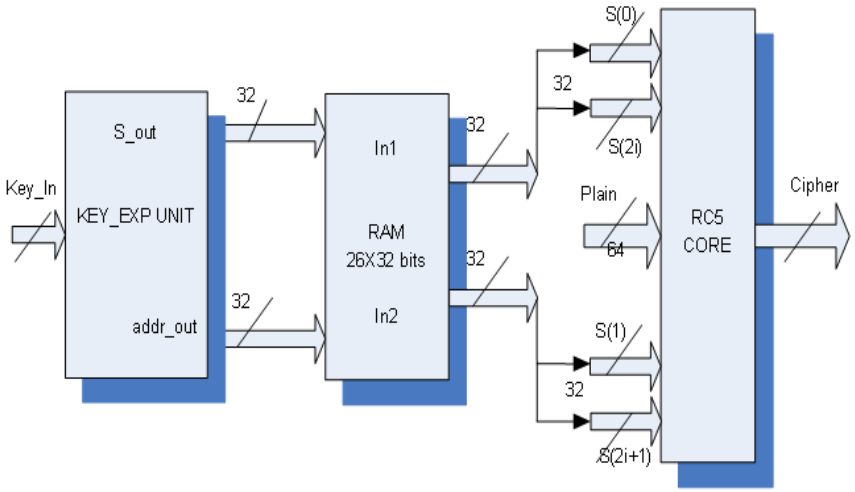

Fig. 3: Architecture of the RC5

We take standard little-endian gathering for packing bytes into I/O (input/output) blocks: the first byte hold the low - order bit positions of register A, and so on, so that the fourth byte hold the high - order bit positions in A, the fifth byte hold the low - order bit positions in B, and the eighth (last) byte hold the high - order bit positions in B.

Encryption:

We assume that the input block is given in two w - bit registers A and B. We also assume that key - expansion 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 outstanding simplicity of this 5 - line algorithm. We also note that each RC5 algorithm round updates both registers A and B, where as a "round" in DES algorithm updates only half of its registers. An RC5 algorithm "half - round" (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 algorithm routine is easily derived from the encryption algorithm 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].

II. PROBLEM DEFINITION

A disadvantage of using public key cryptography for encryption process is speed. There are many secret key encryption techniques that are greatly faster than any currently available public key encryption technique. Key generation can be slow. RSA operations are slower than similar symmetric key operations.

III. OBJECTIVES

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

IV. IMPLEMENTATION

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 on the owner side is a documentation which is used by the owner to carry out the owner function in the system and file training phase. Additionally, this documentation is used by the owner at some stage in the dynamic operations on the cloud data.

V. FUTURE SCOPE

This method will allow lots of data owners to store their data over cloud with confidence. It will minimize their headache of maintaining this data on local. Even the confidentiality and integrity of data is maintained.

VI. CONCLUSION

The data owner can not only archiving and accessing the data stored by the Cloud Service Provider, but also manipulate and scaling this data on the cloud servers. The offered scheme enables the right/valid users to ensure that they are receiving the recent copy of the outsourced data. Additionally, in case of dispute about data integrity/originality, 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.

