ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 20, April 2016

An Efficient Information Security In Cloud Computing Through Enhanced RSA

A. Pelsita Mary[1], K. Kuppusamy[2]

M.Phil Research Scholar[1], Professor[2]
Department of Computer Science and Engineering,
Alagappa University, Karaikudi, India.
[1]pelistamphil@gmail.com
[2]kkdiksamy@yahoo.com

***Abstract*— Distributed computing world view enables the clients to get to the outsourced information from the Cloud server without the apparatus and programming management. For the feasible use of delicate information from Cloud Service Provider, the information proprietor encodes before outsourcing to the cloud server. To make the information to be secured in cloud, we need to have some security solutions. This paper proposes a productive information security strategy utilizing cryptographic procedures such as Enhanced RSA algorithm. The proposed strategy encodes the delicate information, and also distinguishes the untrustworthy party to get the information using combined hash capacities. This paper explored the effective technique with prominent feature of data integrity and confidentiality as far as capacity, correspondence and computational overheads. The outcome exhibits that the proposed security strategy is more effective than the current security framework.**

***Keywords*-cloud, confidentiality, integrity, enhanced, cryptography.**

I. Introduction

In cloud computing, data is stored in remote massively scalable data centers where compute resources can be dynamically shared to accomplish significant economies of scale. The storage capacity needs to scale with compute resources to manage successfully and getting maximum cloud benefits.

Armbrust[8] explained Cloud is as the Center which offers services to access network resources. It has the following common characteristics; (i) pay-per-usage (ii) flexible size (iii) own-service interface and (iv) resources that are abstracted or virtualized. Storage management is particularly important for organizations into cloud computing. To evade data damage, the cloud system needs to provide protection for data and flexibility. The environment must be able to recover the data swiftly in order to restore access to the cloud services if loss does occur. The storage management and information protection in cloud environment helps to deliver a workload-optimized approach.

Sharing of cloud resources such as providing structure, operating system, application and also network security is controlled by the cloud service provider depending on the cloud deployment model. cloud data is controlled by the users of cloud depending on the cloud service model in their application. An organization classifies the information agreeing to the sensitivity to its loss or disclosure. The data owner defines the level of information sensitivity classification based on the security control.

Cloud computing service model is classified such as infrastructure as service, platform as service and software as service. It contains data provider, authorized users, Communication Access Point (CAP), Security Access Point (SAP), Application Access Point (AAP), Application servers and functional components. Software is represented by various application servers of same and/or different types in software as a service model. The main characteristics of cloud computing model are the data provider and users of cloud do not access servers directly. To access the various services of cloud is based on user request and parameters processing, An Application Access Point Server distributes the service request to the different application server. The users may access cloud services through internet using Communication Access Point.

When a user requests for some application service via CAP to the cloud that request would reach SAP server initially. The server will forward it to the Policy Decision Point (PDP) Server to authenticate and for authorization decision. If both are approved, the request of users' application would be passed to the appropriate application server, where it will be served, and the user will get response.

SAP server, then it forwards to the PDP server, receiving reply back to the SAP server. Finally access to the application server provides the requested service, are performed promptly and clearly to the user. So, the user may not aware of any of these actions, except if some illegal action is attempted.

II. PREVIOUS WORK

To make sure the data integrity of a file consisting of a finite set of data blocks in cloud server several solutions are defined by Qian Wang[4]. The first and straight forward solution is to ensure the data integrity, the data owner pre-computes the MACs for the entire file with a set of secrete keys, before outsourcing data to cloud server. In auditing process, for each time the data owner tells the secret key to the cloud server and asks for new MAC for verification. In this method the number of verifications are restricted to the number of secrete keys. Once the keys are exhausted, the data owner must retrieve the whole file from the cloud server to compute the new MACs for the remaining blocks. This method takes the vast number of communication overhead for verification of entire file, which effect the efficiency of system.

Another solution is to overcome the drawback of previous method, is to create the signatures for each block instead of MACs to get the public audit-ability. This solution can provide probabilistic assurance of data accuracy and public audit-ability, which results in large communication and affects the system efficiency. The above solutions support only static data and they dont deal with the dynamic data updates.

Qian Wang[4]designed an efficient solution to support the public audit-ability without saving the data blocks from server. The design of dynamic data operations is a challenging task in cloud storage system. They suggested a RSA signature authenticator for verification with data. To support the efficient handling for multiple auditing tasks, they drawn-out the technique of bilinear aggregate signature and they introduced a third party auditor to execute the multiple auditing tasks simultaneously.

Junbeom Hur[5] explained the cryptographic based solution for data sharing with a cipher text policy attribute-based encryption (CP-ABF). That would increase the security of the data. The data owners define the access policies on the data. The main drawback of this method is the unauthorized users can access the key for decryption the encryption data.

In the cloud, both data and applications are managed by the data owner and also cloud service provider. To access both the cloud data and applications as a cloud service more securely a data security model has been defined by Mohamed, E.M[6] In this security model, a single default gateway as a platform to the secure user data across public cloud applications. The default gateway only encrypts sensitive data using encryption algorithm, before sending to the cloud server. In this method only authorized users can access the data but the cloud service provider can permit the access for unauthorized users when cheating to the data owner. Therefore, this method reduces the security as proper key management was not implemented.

To increase the income and degree of connectivity from the cloud computing model during access and updating data from data centre to the cloud user, Dubey[7] developed a system using RSA and MD5 algorithms for evading unknown access data from cloud server. The main drawback of this method is the cloud service provider also has an equivalent control of data as the data owner and the computation load for cloud service provider is relational to the degree of connectivity so that the performance of the system can be degraded.

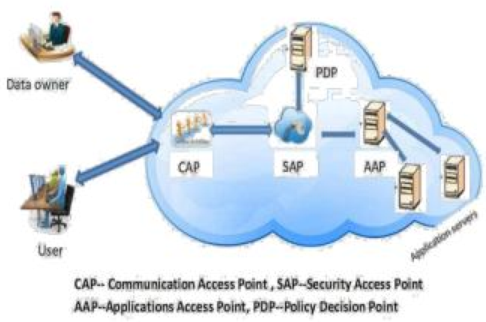

Data owner
User
PDP
CAP
SAP
AAP
Application servers
CAP-- Communication Access Point , SAP--Security Access Point
AAP--Applications Access Point, PDP--Policy Decision Point

Figure 1: Block diagram

III. PROPOSED WORK

Enhanced RSA algorithm The main goal of this algorithm is to improve the security than the previous system. RSA algorithm has the drawback such as duplicate public key, complexity in key generation, low speed and security. In this proposed system key length and blocks are increased up to 256 bits. Encryption and decryption is done up to 16 characters. So hackers or unauthorized cannot access the data. In this proposed system service provider also cannot access the key. As increasing the key size the data security is also improved.

Aims of the proposed system are Design an efficient data privacy algorithm using cryptographic techniques. Detection of the dishonest party using combined hash values verification is done. It reduces the computational overheads of CSP, while introducing TTP and Access the out sourced data, even if data owner is in off-line

.

Block Status Table

The Block Status Table (BST) is a small data structure used to access the encrypted file from the cloud service provider. It consists of two column such as SNj and BNj , where SNj is the sequence number of physical storage of data block j in the file and BNj is the data block number. Initially the data owner stores the table entries as SNj = BNj = j. For insertion of data blocks, the BST is implemented using linked list. The structure of BST for data blocks as shown.

TABLE 1: STRUCTURE OF BLOCK STATUS TABLE

Sequence Number	Block Number
1	1
2	2
3	3
4	4

Setup algorithm

Algorithm Setup(file, key)
Input: Source file and 256 bits key
Output: Number of data blocks(m) and key rotation setup

Step1: Divide the source file in to blocks of equal size.
Number of blocks(m)=file size/block size
Step2: Create circular doble linked list of s16 nodes for key rotation and store one character in each node.

Step3: Create a simple data encoding map table for all the characters.

ALGORITHM for Data Encryption

Input : Source file and KEY for encryption 16 characters
Output : Encrypted file

Step1: Split the characters of the file/string into chunks 128 characters
Step2: Get the binary equivalent of the current Character (process character by character of chunks)
Step3: Take out the first character from the binary value and store it into first character variable and consider the rest of binary value for shift operations.
Step4: Store binary value into the circular double linked list for bit operations
Step5: Select character for a key from ith position, such a way that if 1st chunk character is selected for encryption then select the first character of the key, so i=1.
Step6: If elected chunk character greater than size of key (16) get the modulus of chunk character position.
Step7: Add the stripped off first character to the resultant binary value of circular array after bit operations.
Step8: Add the selected ith key character and (i+2)th key character.
Step9: Right shift the binary bits in circular list by (added value mod 5), if mod 5 is 0 then do by 2.
Step10: Add the selected ith key character value to its previous (i-1)th character value, for instance if 1st key character is selected then the previous character would the 16th character (key is stored into circular array or double way linked list for this operations)
Step11: Get the character equivalent of the binary value
Step12: Get the mapped value from encoded Map, which is the decrypted value of the character.
Step13: Repeat step 2 through 11 for all the chunks with different key.

ALGORITHM for Data Encryption

Input : Encrypted file and KEY for encryption 16 characters
Output : Source file

Step1: Split the characters/ string of the file into chunks of 128 characters
Step2: Get the binary equivalent of the current Character (process character by character of chunks)
Step3: Remove the first character from the binary value and store it into first character variable and consider the rest of binary value for shift operations.
Step4: Store binary value into a circular double linked list for bit operations
Step5: Select a key character from ith position, such a way that if 1st chunk character is selected for decryption then select the first character of the key, so i=1. If elected chunk character greater than size of key (16) get the modulus of chunk character position.
Step6: Add the stripped off first Character to the resultant binary value of circular list.
Step7: Add the selected ith key character and (i+2)th key character.
Step8: Left shift the binary bits in circular list by (added value mod 5), if mod 5 is 0 then do by 2.
Step9: Add the selected ith key character value to its previous (i-1)th character value, for instance if 1st key character is selected then the previous character would be the 16th character (key is stored into circular array or double way linked list for this operations)

Step10: Get the character equivalent of the binary value
Step11: Get the mapped value from encoded Map, which is the decrypted value of the character.
Step12: Repeat steps 2 through 11 for all the chunks with different key.

Data Encryption

□ The information proprietor encodes the record before sending it to the Cloud Service Provider(CSP).□ The encryption calculation has a few stages and made out of key Chooser, Circular Array Inverter and Circular Array shifter. □ The encryption calculation is composed, the data at most astounding component by applying arrangement of turns on each square character and the key is pivoted for each character. □ From this it is guaranteed that same key is not utilized for scrambling each character and thus this calculation is called as key engine encryption calculation.□ The document is partitioned into pieces and privacy is accentuated on each character level of a square. What might as well be called square character is put away in roundabout cluster and number of moves the roundabout exhibit is pivoted is chosen by the CA shifter.□ Where each turn isolates the information by 2 and this will improve the information to its minimum worth and henceforth the protection of information is guaranteed. Since stepper development of CA is diverse for various character it's hard/difficult to decide the genuine estimation of CA as clarified. □ The key segment of calculation is the CA inverter and CA shifter which is performed on each square character lastly on whole piece. □ On the off chance that File has N pieces and if each square has n characters then CAI and CAS is performed by N∗n operations. □ Also, along these lines this calculation has many-sided quality of O(N ∗ n). At the point when the client needs to get to information from cloud server, the client approval and information check method is clarified. □ The unscrambling process happens precisely inverse to encryption which finds a square character from figure content according to mathematical statement. □ The calculation recommends that CA shifter is performed first then Key chooser segment is utilized to choose two keys and they are included before upsetting CA. □ Since CA as of now contains supplemented esteem and supplement of CA now yields unique encoded esteem. □ The Encoding Map (Em) is looked to get its unique character. The Algorithm has same multifaceted nature as Encryption.

Data Decryption

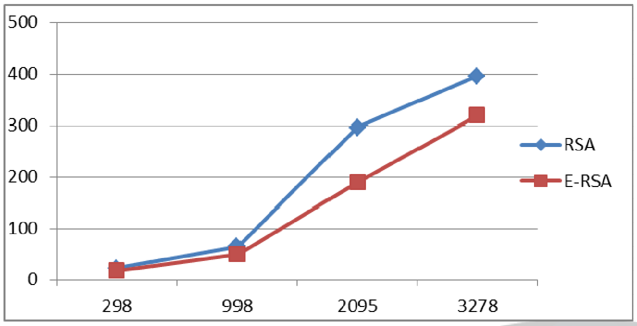
□ The imperative or key operations in both procedures are CA shifter and CA inverter. The ideal opportunity for encryption/decryption is specifically relies on upon these two operations.□ The quantity of developments of CA and its reversal process chooses the accuracy of encryption/decoding. □ Alongside CA shifter the key is additionally turned for each square character. This guarantees same key is not utilized for various characters.□ The examination is performed on various documents and number of developments utilized for moving CA and key stays same. □ For the record with size 313 characters the quantity of developments performed was 646, correspondingly for document with size 3139 it is 6479 developments which is twofold the record size. □ In different words each development considers 0.5 character i.e a large portion of the character which is 8 bits. This induces shift operation is performed on each byte and consequently the information is masked at fine level (byte level). □ The other parameter for investigation is the CA inverter operation. 118 times the supplement is performed for the document size of 313 characters also for the record size of 3139 characters the quantity of supplements performed was 1178 which is around 3 characters. This infers supplement operation is performed for each 3 characters and consequently the information is masked at coarse level (bytes level). □ As CA movement is performed for each character contrasted with supplement operation it has more effect on the encryption/unscrambling process and in this way it can be presumed that encryption is going on at better level i.e byte level. □ The execution time is plotted on the diagram. With expansion in document estimate the quantity of developments and supplements are high and consequently the execution time is straightforwardly corresponding to record size.□ It is watched that unscrambling is taking additional time than encryption process.

IV. RESULTS

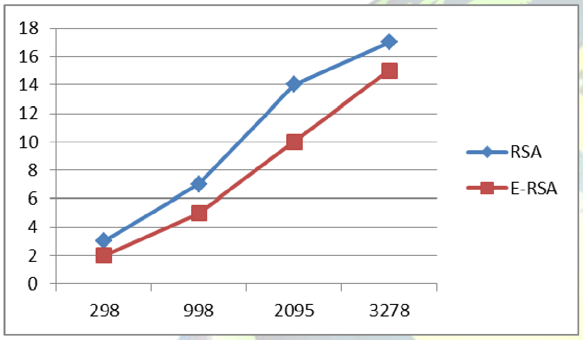
Table 2: Differentiation between RSA and ERSA

File size(bytes)	Algorithm	Encryption Time(sec)	Decryption Time(Sec)	Execution time(Sec)
298	RSA	23	3	26
	E-RSA	18	2	15
998	RSA	65	7	72
	E-RSA	50	5	50
2095	RSA	296	14	310
	E-RSA	190	10	280
3278	RSA	396	17	414
	E-RSA	320	15	400

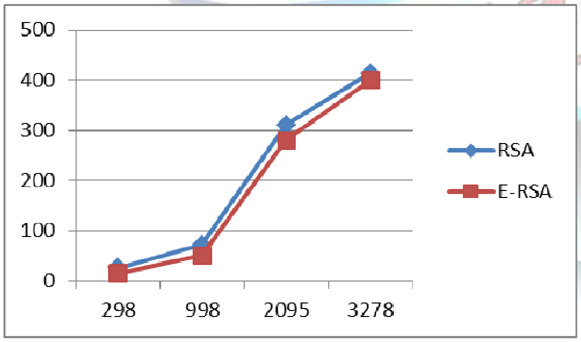
Performance Graph for Encryption Time



Performance Graph for Decryption Time



Performance Graph for Execution time



V. CONCLUSION

This research work has proposed a mechanism to provide secured data in the cloud. Both RSA and ERSA algorithms have been compared according to the execution time. Execution time of Enhanced algorithm is less than the RSA algorithm. Key size is increased to 256 bit(16 char). RSA algorithm has many disadvantages, so to overcome those disadvantages Enhanced RSA is proposed and it produces efficient security to the information in the cloud.

REFERENCES

[1] http://security.setecs.com, security architecture for Cloud Computing environment white pater, 2011

[2] Ayad Barson and Anwar Hasan, Enabling Dynamic Data and Indirect Mutual Trust for Cloud Computing Storage Systems, In IEEE Transaction on Parallel and Distributed Systmes, 2012.

[3] Cong Wang, Kui Ren and Jia Wang, Secure and Practical Outsourcing of Linear Programming in cloud computing, in IEEE International Conference on INFOCOM, pages 820-826,2011.

[4] Q. Wang, C. Wang, J.Li, K. Ren and W.Lou, Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing, in proceeding of 14 European Symposium, Research in Computer-Security(ESORICS 09),pages 355-370,2009.

[5] Junbeom Hur, Improving Security and Effeciency in Attribute Based Data Sharing, in IEEE Transactions on Knowledge and Data Engineering, Volume:25, issue:10, page 2271-2282,2013

[6] Mohamed E M, Abdelkader H S, El-Etriby S, Enhanced Data Security Model for Cloud Computing, Informatics and Systems (INFOS), 8th International Conference on pages 12-17,2012.

[7] Dubey A K, Namdev M, Shrivastava S S, Clouduser Security based on RSA and MD5 Algorithm for Resource Attestation and Sharing in Java Environment, Software Engineering(CONSEG), CSI 6th international Conference on, page 18, September 2012.

[8] M Armbrust, A Fox, R Grifth, A d Joseph and R Katz, Above the Clouds: Aberkeley View of Cloud Computin, U C Barkeley Reliable Adaptive Distributed Systems Labaratory White Paper, 2009.

[9] Cong Wang, Ning Cao, Jin Li, Kui Ren and W Lou, Secure Ranked Keyword Search over encrypted Cloud data, In IEEE 30th International Conference on Distributed Computing Systems (ICDCS), PAGES 253-262, 2010.

[10] Kuyoro S O, Ibikunle F, Awodele O, Cloud Computing Security Issues and challenges, In International Journal of Computer Networks, vol.3, issue 5, 2011.