RESEARCH ARTICLE OPEN ACCESS

Enhancing Data Security in Cloud Computing Using RSA Encryption and MD5 Algorithm

Sudhansu Ranjan Lenka[1], Biswaranjan Nayak[2]
Department of Computer Science and Engineering[1 &2],
Trident Academy of Technology,
Bhubaneswar 751024
Odisha- India

ABSTRACT
Cloud computing is the technology through which people can share resources, services and information among the people through use of internet. Since we share the data through the internet, security is considered as a major issue. In Cloud computing several security issues arises like confidentiality, integrity and authentication. In this paper a new security model have been proposed. The architecture provides a mechanism through which we can get secure communication as well as hiding the information from unauthorized users. In this model we have implemented a combination of RSA encryption and digital signature technique which can easily with all types of cloud computing features like: PaaS, SaaS and IaaS. This combination mechanism provides three way security i.e. data security, authentication and verification. In this paper, we have proposed RSA encryption algorithm for confidentiality of data and for authentication MD 5 algorithm have been implemented.

Keywords: - Cloud Computing, data security, MD5, RSA, encryption and digital signature.

I. INTRODUCTION

In the current scenario, Cloud Computing is being emerged as one of the most powerful and developing networking system which is utilised by developers as well as users [5]. Cloud computing is well suited for the persons who are interested to mould in networking environment. As security is one of the key challenging factor in network platform, providing security in cloud computing also plays a prime concern for its effective utilization.

Cloud computing environment allows its resources to be shared among servers, users and individuals, in turn files or data that are stored in the Cloud are openly accessible to all. Due to this open accessibility factor, the files or data of an individual can be used by other users of the Cloud as a result attacking treat on data or files become more vulnerable [6,15]. Once the intruders get access to data, misuse of it possesses a major risk. The intruder may destroy the original data or disrupt the communication also. Apart from files and data Cloud service providers facilitate critical applications whose security requires a lot of attention [13]. One of the common problem occurs in Cloud is that an individual may not possess the control over the place of data storage. It becomes necessary for a Cloud user to utilize the resource allocation and scheduling facilities provided by the Cloud service provider in turn at the time of processing it becomes essential to protect the data or files of the individuals. To overcome this problem, security in Cloud computing platform should be implemented effectively. We have explored varieties of security aspects in our proposed Cloud computing model.

Till yet, researchers have developed different security models and algorithms applied on them, but these models were unable to solve all types of security threats which has been discussed in [14,10,7]. As this is the era of E-commerce [8] and online business, we are required to imply high capacity security models in Cloud computing environment to have an effective e-commerce and online business processing systems. As it is discussed in [9] that the security models which are developed and used in Cloud computing only provides security for file system not for communication proposes. Although it is found in [11] that present security models sometimes utilizes secured communication channel, the process is not cost effective. One can hardly find a model which can utilize security in main server as well as in transaction. In [12], the researchers have proposed hardware encryption system for augmentation of secured communication system but it provides certain drawbacks like i) the practical implementation is too difficult , ii) hardware encryption is effective for database system and not for other security issues.

In the current networking environment, authenticated user detection technique plays a vital role but the recently used security models hardly emphasizes on this technique in Cloud environment. In the proposed paper, we have proposed the concept of digital signature with RSA encryption algorithm for data encryption purposes at the time of data transfer in the internet. We have combined both the digital signature and RSA for effective handling of authentication and security which can be utilized in Cloud computing environment. In our work, the RSA algorithm has been used for data encryption where the user wants to upload the data in the Cloud and MD5 algorithm has been used for authentication purpose.

II. RELATED WORK

In Cloud computing data are not stored in user's computer but are stored in Cloud storage which is hosted by third parties. Several research work has proposed and done on security of Cloud computing. Ravi Shankar Dhakar et. al[1]

proposed an algorithm called "Modified RSA encryption Algorithm(MREA)" where the researchers have factorized RSA cryptosystem and their implementation compares the existing system with respect to their system with key sizes upto 1024 bits and the researchers have claimed that their work is better than existing system for the brute force. The researchers Suli wang et al. [4] have worked in "File encryption and decryption system based on RSA algorithm" in which they have proposed RSA algorithm for encryption and decryption of smaller size files. The researchers Maryam Savari et al [2] worked on "Comparison of ECC and RSA algorithm in multipurpose smart card application" where comparison is done between the security of RSA 1024 bit key and ECC 160 bit key sizes. The researchers P.R. Vijaylakshmi et al[3] have worked in " Performance Analysis of RSA and ECC in Identity-Based Authenticated New Multiparty Key Agreement Protocol" where the comparison is done between ECC algorithm with 128 bits and RSA algorithm with 1024 bits key sizes.

III. PROPOSED MODEL

In our proposed model, the two algorithms are implemented which are: RSA encryption and MD5 hashing. RSA algorithm is used for secured communication [17, 16] and file encryption and decryption purpose. MD5 algorithm is used for digital signature as well as for covering the tables from unauthorized users [18]. The block diagram of the model is depicted in figure 1. Here, all requests must pass through a secured channel which is connected to the main system server. The system server is connected to other data storage system. The data storage system can be treated as a server or only storage device.

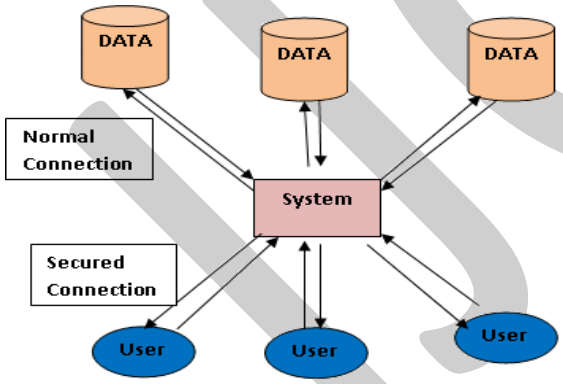

Fig. 1. Proposed Security Model

To provide a secure communication, the user requests are encrypted while sending to the cloud provider. RSA algorithm encrypts the users request by using the systems public key. Similarly, when the user requests for a file, the system sends the encrypted file using the user's public key. When the user receives the encrypted file, the user's browser will decrypt it using the user's private key. So RSA algorithm facilitates secured communication between the user and the system.

In our model as depicted in figure 2, RSA encryption algorithm is also used for encryption and decryption. After getting connected with the system, the user can upload or download files from the server through the system. When the user wants to upload a file, it encrypts the file using the system's public key. For each request the system randomly generates different keys for encryption and decryption. After receiving the encrypted key from the system the user encrypts the file. That particular key is not used further in any instance. In the database table of the system server both the keys along with the user account name is kept. Before inserting into the table the user account name is hashed which is augmented by using MD5 algorithm and it prevents the unauthorized person to access the key and decrypts a particular file for a particular user.

For authentication, the user generates message digest through MD5 algorithm. After receiving the digital signature from the user, the same MD5 algorithm will be used for signature verification. Once verification is done, the encrypted file is stored on the storage server in the user name. So the proposed architecture provides two way securities by means of RSA encryption algorithm for confidentiality and MD5 algorithm for authentication.

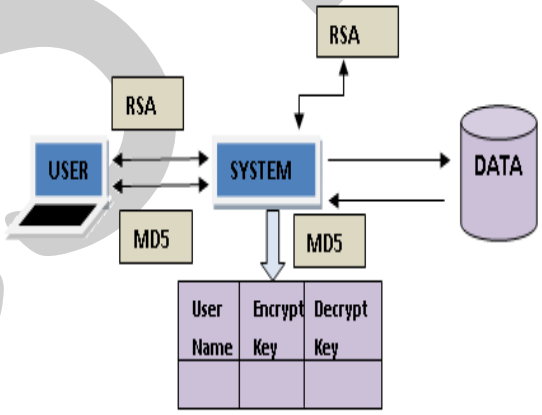

Fig. 2. Working of RSA and MD5 algorithms in the proposed model

IV. WORKING PRINCIPLE OF DIGITAL SIGNATURE WITH RSA ENCRYPTION ALGORITHM

Let's we assume that there are two organizations A and B. Organization B wants a secure data from A's Cloud. So to send the secure data, the following steps are followed to implement Digital Signature along with RSA encryption algorithm and diagrammatically it is depicted in figure 3.

Step1. Using RSA encryption algorithm A will encrypt the message using B's public key.
Step2. The document will be crunched into fixed few lines by using MD5 algorithm to generate message digest.

Step3. A then encrypt the message digest using its own private key to generate Digital Signature.

Step4. B will decrypt the message using his own private key and finally the Signature is verified using A's public key.

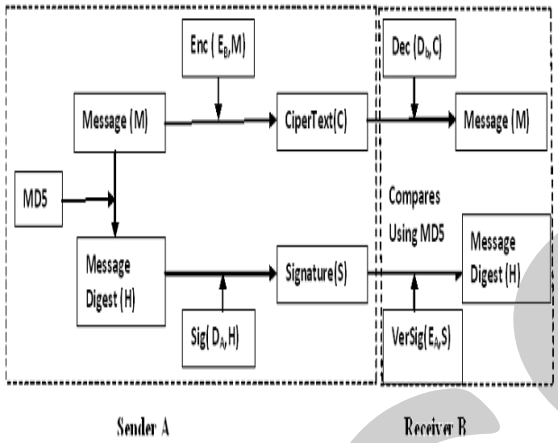

Fig. 3. Encryption, Decryption, Signature and Verification

V. MESSAGE DIGEST ALGORITHM

Message digest functions which are also called as hash functions, used to generate Digital Signature of the information which is known as message digest. MD5 algorithm is used to implement integrity of the message which produce message digest of size 128 bits. These are mathematical functions that process information to produce different message digest for each unique message. Message digest algorithm yields two advantages. Identical messages always generate the same message digest and even if one of the bits of the message changes, then it produce different message digest. The other advantage is that message digests are much shorter than the document from which digests are generated. It processes the message and generates 128- bits message digest. The algorithm consists of the following steps:

1. Append the padding bits
2. Append the length
3. Initialize MD buffer
4. Process message in 512 bit blocks
5. Output generation

VI. PROPOSED ALGORITHM

Here, n is the modulus, e is the encryption exponent and d is the secret exponent or decryption exponent.

The algorithm is divided into 5 steps: Key Generation, Digital Signing, Encryption, Decryption and Signature Verification with their working functions are discussed as below:

Step-1: Key Generation

Randomly generate two large prime nos: p and q.
Compute n=p * q
Compute the totient: $\Phi(n)= (p-1) * (q-1)$
Choose an integer 'e' such that $1< e < \Phi(n)$ and $gcd(e, \Phi(n)) = 1$
Compute d, such that $d * e = 1 \bmod \Phi(n)$
The public key is (n, e) and the private key is (n, d).

Step2: Digital Signing

Generate message digest of the document to be send by using MD5 algorithm.
The digest is represented as an integer m.
Digital Signature S is generated using the private key(n, d), $S= m^d \bmod n$.
Sender sends this signature S to the recipient.

Step 3: Encryption

Sender represents the plain text message as a positive integer m.
It converts the message into encrypted form using the receiver's public key (e, n).
$C= m^e \bmod n$
Sender sends this encrypted message to the recipient B.

Step 4: Decryption

Recipient B does the following operation:
Using his private key (n, d), it converts the cipher text to plain text 'm'.
$m= C^d \bmod n$

Step 5: Signature Verification

Receiver B does the followings to verify the signature:
An integer V is generated using the senders public key (n, e) and signature S

$$V= S^e \bmod n$$

It extracts the message digest M1, from the integer V using the same MD5 algorithm.
It then computes the message digest M2 from the signature S.
If both the message digests are identical i.e. M1= M2, then signature is valid.

VII. EXPERIMENTAL OBSERVATIONS

For experiment purpose, we have some taken some sample data and implemented by using the proposed algorithm through C language. The whole process is discussed as below and the output is shown in figure 4.

Step 1: Key Generation:

1. We have chosen two distinct prime numbers p=23 and q=53.
2. Compute n=p*q, thus n=23*53 =1219.
3. Compute Euler's totient function, Ø(n)=(p-1)*(q-1), thus Ø(n)=(23-1)*(53-1) = 22*52 = 1144.
4. Chose any integer e, such that 1 < e < 1144 that is gcd (e, 1144) =1. Here, we chose e=3.
5. Compute d , d = e-1(mod Ø(n)), thus d=3-1(mod 1144) = 763.
6. Thus the Public-Key is (e, n) = (3, 1219) and the Private-Key is (d, n) = (763, 1219). This Private-Key is kept secret and it is known only to the user.

Step 2: Encryption:

1. The Public-Key (3, 1219) is given by the Cloud service provider to the user who wishes to store the data.
2. Let the message to be send is "hello" which is converted to integer in the following manner:
A=0, B=1 ,a = 27, b=28,c=29 and so on .
So the message "welcome" is encoded to m= 49313829413931
3. Data is encrypted now by the Sender using the corresponding Public-Key which is shared by both the sender and the receiver.
$C=m^e \bmod n$ $C=493138294139313(\text{mod}1219)=625535179657807535$.
4. This encrypted data i.e, cipher text is send to the recipient.

Step 3: Digital Signature and signature verification:

First using MD5 algorithm the message gets converted to message digest i.e. to hexadecimal form.
MD1=H(m)= 0x000c00f0000000f0426f00f0726000f0.
Message digest in decimal form M1= 012024000024066111024011410802 40.
Next digitally signed the message digest MD1 using its own private key d to generate digital signature S.
S=(MD1)dmod n= 088702580002588392960258850124025 8.
Sender then sends the digital signature S to the recipient.
Receiver then computes the integer V using S, e and n.
V= Se mod n= 012024000024066111024011410802 40.
Receiver the computes the message digest from S using MD5 algorithm
MD2 = 012024000024066111024011410802 40.
Since V = MD2, so the Signature is verified.

Step 4: Decryption:

1. The receiver decrypts the data by computing, m = Cd(mod n) = 49313829413931.
3. Once the m value is obtained, user will get back the original message using the same encoding technique.

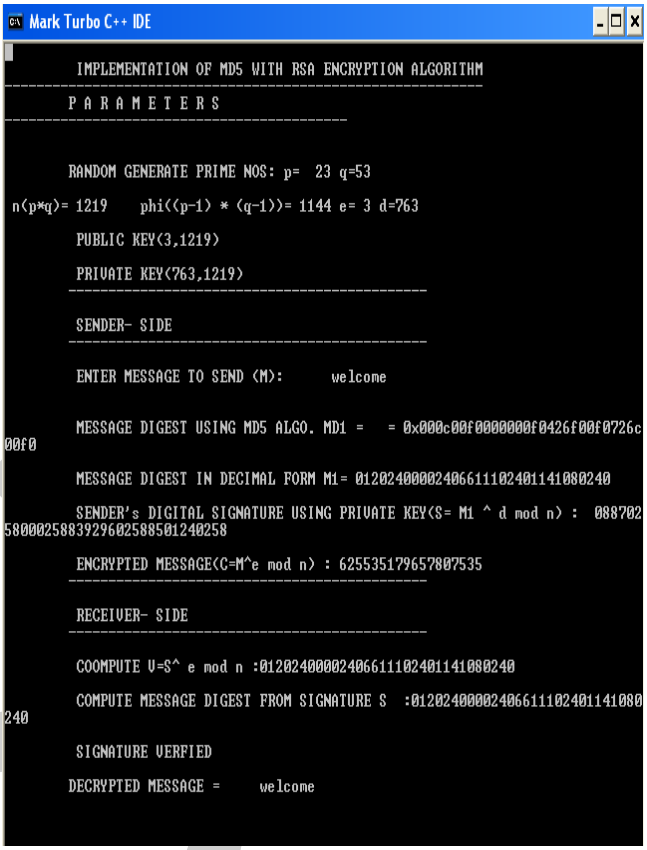

Fig. 4: Output of the proposed Algorithm using C Language

VIII. CONCLUSION

In our work, we have proposed a new security architecture which implements RSA for both encryption and secure communication purposes, whereas MD5 hashing is used for digital signature and hiding key information. This model provides security for the entire cloud computing environment. The specialty of our design approach is that here, each algorithm is executed in different servers which overcomes the problem of slow downing the system. In the proposed system, an intruder cannot easily access or upload the file because the algorithms are executed in different servers at different locations.

For implementation purpose we have combined both RSA encryption and Digital Signatures algorithms as a result a powerful security and data integrity service system is obtained. Although RSA encryption algorithm is quite deterministic but MD5 algorithm makes the model highly secured. In a nutshell we can say that our proposed model can provide a better approach as compared to other works. In the future we would emphasize on finding an encryption algorithm which will be more light and secured for data in Cloud computing.

REFERENCES

[1] Ravi Shankar Dhakar, Amit Kumar Gupta, "Modified RSA Encryption Algorithm (MREA)". Advanced Computing & Communication Technologies (ACCT), Second International Conference, 2012.

[2] Maryam Savari, Mohammad Montazerolzohour and Yeoh Eng Thiam, "Comparison of ECC and RSA Algorithm in Multipurpose Smart Card Application". Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), International Conference, 2012.

[3] P.R. Vijayalakshmi, K. Bommanna Raja, "Performance Analysis of RSA and ECC in Identity-Based Authenticated New Multiparty Key Agreement Protocol". Computing, Communication and Applications (ICCCA), International Conference, 2012.

[4] Suli Wang, Ganlai Liu, "File encryption and decryption system based on RSA algorithm". Computational and Information Sciences (ICCIS), International Conference, 2011.

[5] Yashpal Kadam, "Security Issues in Cloud Computing A Transparent View", International Journal of Computer Science Emerging Technology, Vol-2 No 5 October, 2011 , 316-322.

[6] Rohit Bhadauria, Rituparna Chaki, Nabendu Chaki, Sugata Sanyal, "A Survey on Security Issues in Cloud Computing", 2011.

[7] Ngongang Guy Mollet, "CLOUD COMPUTING SECURITY", Thesis Paper, April 11, 2011.

[8] Gunasekar Kumar, Anirudh Chelikani, "Analysis of security issues in cloud based e-learning", Master's thesis, 2011.

[9] Jiyi Wu, Qianli Shen, Tong Wang, Ji Zhu, Jianlin Zhang "Recent Advances in Cloud Security", JOURNAL OF COMPUTERS, VOL. 6, NO. 10, OCTOBER 2011.

[10] "Cloud Computing: Silver Lining or Storm Ahead?", Volume 13 Number 2, Spring 2010.

[11] Ahmad-Reza Sadeghi, Thomas Schneider, and Marcel Winandy, "Token - Based Cloud Computing Secure Outsourcing of Data and Arbitrary Computations with Lower Latency", TRUST 2010, LNCS6101, pp . 417–429, 2010.

[12] Trusted Computing Group, "Solving the Data Security Dilemma with Self-Encrypting Drives", May 2010.

[13] Ye Hu, Johnny Wong, Gabriel Iszlai, Marin Litoiu, "Resource Provisioning for Cloud Computing", IBM Canada Ltd., 2009.

[14] Daniele Catteddu, Giles Hogben, "Cloud Computing:- Benefits, risks and recommendations for information security", November, 2009.

[15] Mladen A. Vouk, "Cloud Computing – Issues, Research and Implementations", Journal of Computing and Information Technology - CIT 16, 2008, 4, 235–246.

[16] Burt Kaliski, The Mathematics of the RSA Public-Key Cryptosystem, RSA Laboratories , February, 2003.

[17] R.L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Laboratory for Computer Science, Massachusetts Institute of Technology, Cam-bridge, November, 1977.

[18] Ronald Rivest, "MD5 Message-Digest Algorithm", rfc 1321, April 1992.