Hybrid Security Approach for Outsource Data Access in Mobile Cloud Environments

Arivumathi I[1] Ms. Niranjana A[2]
[1]P.G. Student [2]Assistant Professor
[1]Department of Computer Science & Engineering [2]Department of Information Technology
[1,2]Kongunadu College of Engineering and Technology, Thottiam, Trichy, Tamil Nadu, India

Abstract— In mobile cloud computing (MCC), mobile devices can be rely on cloud computing and resources are stored to perform computationally various operations such as searching, mining and accessing the files in secure manner. During the file access in secure network, there are no efficient cryptographic algorithms. For secure the files in cloud, various cryptographic approaches are used but most of them increase the computational cost of file storage. Thus a new technique is required to develop which efficiently used for data storage and data access. Therefore the presented work is intended to find an approach by which the security in file hosting and management can play an important role. Therefore, we give techniques and algorithms for secured access in mobile cloud environment. So we can provide three tier architecture that includes MD5 (Message Digest) algorithm, Advance encryption standard algorithm and Elliptic curve cryptography algorithm. These algorithms can be implementing in real time mobile cloud environment and an experimental result proves with minimal performance degradation.

Key words: Mobile cloud computing, Privacy Preserving, Cryptographic approach, Secure storage network

I. Introduction

Mobile devices are increasingly become an important part of human life as efficient and convenient communication tools are bounded by time and place. Mobile users use various rich services from mobile applications which can be run on the devices and/or on remote servers via wireless networks. The rapid development of mobile computing (MC) happens to a great trend in the progress of information technology, for commerce and industry fields. However, the mobile devices are opposite to many challenges in their resources and communications. The limited possessions noticeably delay the improvement of service characters. Cloud computing has been widely familiar as the next generation's computing infrastructure. Cloud Computing offers some compensation by allowing users to use infrastructure, platforms, and software. Mobile cloud computing is defined as follows: "Mobile Cloud Computing is refers to an infrastructure in where both the data cache space and the data dispensation happen exterior of the mobile device. Mobile cloud application shift the computing power and data storage space forth from mobile and into cloud, fetching the applications and mobile computing to not a Smartphone users but they are much broader range of mobile subscribers".

II. Related work

1) Yung-Hsiang Lu et.al…, [1] suggest that cloud computing be able to potentially save energy for mobile users. However, not all applications are energy efficient when migrate to the cloud. Mobile cloud computing services would be significantly dissimilar from cloud services for desktops because they must offer energy savings.
2) Ayesha Malik, et.al…, [3] analysis to give a solution for the threats that are the major topic for anyone when they want to accept cloud services for their work. For this purpose, a framework should be intended for execution of data and information securely in cloud environment. It will protect users' data, messages, information against various attacks.
3) Shashi Mehrotra Seth et.al…, [4] provide comparative survey for encryption algorithms that includes RSA (Rivest-Shamir-Adleman), DES(Data Encryption Standards) and AES (Advanced Encryption Standards) algorithm. Supported on the text files are used. The experimental result it was concluded that DES algorithm devour least encryption time and AES algorithm has slightest memory usage even as encryption time difference is very minor in case of AES algorithm and DES algorithm. RSA consume longest encryption time and memory usage is also very high but output byte is least in case of RSA algorithm.
4) Siani Pearson et.al…,[9] it will reduces the threat to the cloud computing customer of their separate data actuality pilfer or misused, and also abets the cloud computing donor to conform to privacy law. Cloud computing, in which services are carried out on behalf of customers on hardware that the customers do not own or manage, is an increasingly fashionable business model. The input data for cloud services is uploaded by the user to the cloud, which means that they typically result in users' data being present in unencrypted form on a machine that the user does not own or control. This poses some inherent privacy challenges. There is a risk of data theft from machines in the cloud, by rogue employees of cloud service providers or by data thieves breaking into service providers' machines, or even by other users of the equivalent service if there is incomplete isolation of different customers' data in a machine that they share in the cloud.

III. DES based MCC security

Whenever a user want to store any data or files in smart phone memory it will be prompt to enter the password (key k1).This password is then used to encrypt the data/files. This encrypted data/file is then stored on the cloud server. Storing the file on cloud server ensures that there is no file or data physically present in smart phone memory which could be used by the unauthorized user. As encryption is done at the device end i.e. on smart phone, algorithm is also present along with the key. This algorithm along with key and the encrypted data can be used to again regenerate actual data or files. In order to provide more enhanced

security encrypted data present on the cloud server is again encrypted using another key (key k2) which is given by the user after the encrypted is send from smart phone to cloud server. DES based MCC security provides two tier architecture for securing the data. The steps are as follows:
The steps involved in encryption are:

- User select file to encrypt
- User provide key (key k1) for the file to encrypt
- Encryption performed on the file
- Encrypted file send to cloud server
- Prompt from the cloud server to send another key (key k2)
- Key (key k2) entered by the user
- Key k2 is then send to cloud server
- Another encryption performed on previously encrypted data using key k2 and stored on cloud server

IV. THREE TIER MCC SECURITY

Individuals and enterprises obtain advantage to store enormous amount of data of applications on a cloud. However, problems are integrity and authentication.

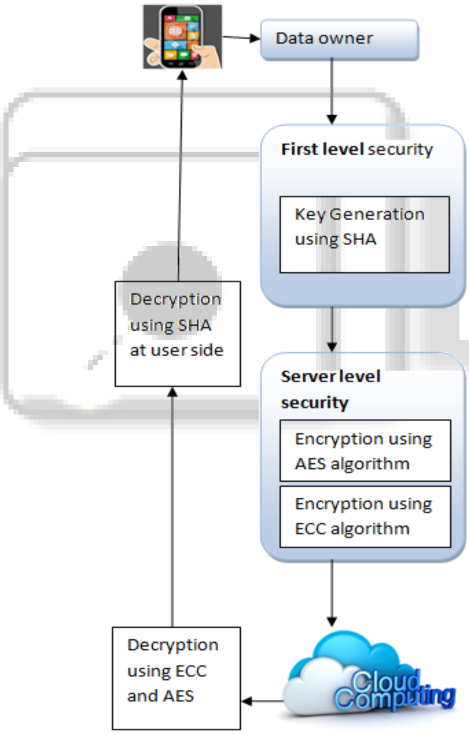
Data owner
First level security
Key Generation using SHA
Decryption using SHA at user side
Server level security
Encryption using AES algorithm
Encryption using ECC algorithm
Decryption using ECC and AES
Cloud Computing

Fig. 1: Three tier framework

- Integrity Every mobile cloud user must certain the precision of their data stored on the cloud. Every access they create must be authenticated and verified. Different approaches in save integrity for one’s information that is accumulating on the cloud is to be proposed. Example, every information stored up by every individual or enterprise in the cloud is labeled bootup them wherein they are the single one to have access such information. Every route they make must be authenticated satisfying that it is their own information and thus authenticating its integrity.
- Authentication Deviating authentication methods have been presented and planned using cloud computing to immune the data access appropriate for mobile environments. Some employ the open standards and even ropes the integration of various authentication methods. For example, the use of contact or log-in IDs, passwords or PINS, authentication requests, etc.
- Digital rights management Illegal distribution and isolation of digital contents such as image, audio, video, and e-book, programs become more and more popular. Some solutions to defend these contents from illegitimate access are applied such as stipulation of encryption and decryption keys to contact these stuffing. A coding or decoding platform must be completed before any mobile user is able to have access to such digital contents. These terms are examined our proposed system and to provide MD5, AES and ECC algorithms for construct secure framework. The flow of the proposed work is illustrated in Fig. 1.

A. *SHA (Secure Hash Algorithm)*

The first version, later dubbed SHA-0, was withdrawn by the NSA. The SHA hash function is similar to the MD4 hash function, but adds some complexity to the algorithm and the block size used was changed. SHA was originally intended as part of the Digital Signature Standard (DSS), a scheme used for signing data and needed a hash function to do so.

- SHA-0: A retronym applied to the original narrative of the 160-bit hash function published in 1993 under the name "SHA". It was detached shortly after publication due to an hiddened "significant flaw" and reinstate by the slightly revised version SHA-1.
- SHA-1: A 160-bit hash function which smack of the earlier MD5 algorithm. This was traced by the National Security Agency (NSA) to be part of the Digital Signature Algorithm. Cryptographic fondness were discovered in SHA-1, and the standard was no longer endorsed for most cryptographic uses after 2010.
- SHA-2: There are two similar hash function, with different block sizes, SHA-256 and SHA-512. They differ from word size; SHA-256 consists of 32-bit words where SHA-512 consists of 64-bit words. There are truncated narratives of each standard, such as SHA-224, SHA-384, SHA-512/224 and SHA-512/256.
- SHA-3: A hash function erstwhile called Keccak, chosen in 2012 after a public rivalry among non-NSA designers. It timbers the same hash lengths as SHA-2, and its internal structure digress significantly from the rest of the SHA family.

1) *Algorithm 1: SHA Algorithm*

- Step 1: Adjoining Padding Bits. The primitive messages are padded, so its length is congruent to 448, modulo 512.
- Step 2: Adjoining Length. 64 bits are adjoined to the end of the padded message to illustrate the length of the primitive message in bytes.
- Step 3: Equiping Processing Functions. SHA1 requires 80 processing
- Step 4: Equiping Processing Constants. SHA1 requires 80 processing constant
- Step 5: Tantalizing Buffers. SHA1 algorithm requires 5 word buffers
- Step 6: Equiping Message in 512-bit Blocks. This are the main task of SHA1 algorithm, which loops through the padded and adjoined message in blocks

of 512 bits each. For each and every input blocks, a number of operations are performed.

B. *AES (Advanced Encryption Standard)*

AES pinpoints a cryptographic algorithm that can be worn to defender electronic data. A symmetric block cipher that will encrypt and decrypt information. Encryption mutates data to an inscrutable form called cipher-text; decrypting the cipher-text shuffle the data reverse into its premitive form, called plaintext. AES allows for three divergent key lengths: 128, 192, or 256 bits. Encryption inhere of 10 rounds,12 rounds,14 rounds of processing contains 128-bit keys, 12 rounds contains 192-bit keys, and 14 rounds contains 256-bit keys. Each of cipher has a 128 bit block size, with different key sizes of bits such as 128, 192 and 256 bits respectively.

1) *High-Level Description of the Algorithm:*

- Key Expansions - round keys are imitatived from the cipher key accepting Rijndael's key schedule. AES desires a dislocated 128-bit round key block for individual round plus one more.
- In initial Round AddRoundKey - individual byte of the state is merged desire a block of the round key accepting bitwise XOR.
- Sub Bytes - a non-linear substitution step for each byte is exchanged with another as maintained by the lookup table.
- Shift Rows - In transposition step, the last three rows of the state can be shifted recurrly a certain number of steps.
- Mix Columns - a mixing proceeding which convey on the columns of the state, coupling the four bytes in individual column.
- Add Round Key- SubBytes step In the Sub Bytes step, all bytes ianothern the state is exchanged desire its access in a fixed 8-bit lookup table, R; jab = R(iab). In the Sub Bytes step, each byte i{a,b} in the state matrix is put back with SubByte R(i{a,b}) using an 8-bit substitution box and the Rijndael S-box. This casualty provides an non-linearity in the cipher key. The S-box is used for imitative from the multiplicative inverse is over from the GF(28),it can be known to have superior non-linearity properties. To avoid barrages based on incomplex algebraic properties, the S-boxes are formulated by combining the contrary function with an revertible affine transformation. The S-box is chosen to avoid any fixed point. While percolating the decryption, Inverse SubBytes step is used, this depends first catching the affine transformation and then inclining the multiplicative inverses.
- Shift Rows step In the Shift Rows step, bytes in individual row of the state are substituted recurrly to the left. The number of places individual byte is substituted digress for each row. The Shift Rows step conveys on the rows of the state; it recurrly substitutes the bytes in individual row by a certain counteract. The first row is left unaffected. All bytes of the second rows will shifted one to the left. Identically, third and fourth row is shifted by counteracts of two and three respectively. For all blocks of sizes 128 bits and 192 bits, the shifting pattern are agnate. Row n is shifted left circular aside n-1 bytes. In the same way,all column of the output state of the Shift Rows step is comprised of bytes from individual column of the input state. For 256-bit block, the first rows are unaffected and shifting for the second, third and fourth rows contains 1 byte, 3 bytes and 4 bytes respectively—this alternation only engages for the Rijndael cipher when used alongside a 256-bit block, an AES doesn’t use a 256-bit block. The expectance of step is to avert the columns actuals linearly independent, in which side, AES sinkings into four independent block ciphers.
- The Mix Columns step In the Mix Columns step, individual column of state is multiplied alongside of a hooked polynomial c(x).In the Mix Columns step, the four bytes of individual column of the state are merged using an revertible linear transformation. The Mix Columns function consists of four bytes as input and outputs, in all individual input byte affects all four output bytes. Well-organised with Shift Rows, Mix Columns provides diversion in the cipher. During this operation, individual columns transformed using a fixed mMatrix multiplications are composed as multiplication and addition of the ingresses. Ingresses are 8 bit bytes acy with regard to accessory of polynomial of order x7. Addition is quietly XOR. Multiplication is a modulo of irreducible polynomial x8+x4+x3+x+1. If process dne bit by bit then after substituting a conditional XOR with 0x1B can be fulfilled if the shifted value is huger than 0xFF. These are unusal case of the usual multiplication in the GF(28). Individual column is feasted as a polynomial at an end GF(28) and is then multiplied by modulo x4+1 with fixed polynomial d(x) = 0x03 · x3 + x2 + x + 0x02. The coefficient is exhibited in their hexadecimal agnate of binary representation of bits in polynomial from GF(2)[x].Mix Columns step can be glimpsed as a multiplication by the shown precise matrix is a finite field GF(28). This process can be contrued further in the article Rijndael mix columns.
- Add Round Key step In the Add Round Key step, individual byte of the state is copuled with a byte of the round sub key using the XOR proceeding (⊕). In the Add Round Key step, the sub key is copuled with the state. For individual round, a sub key is formulated from a main key using the Rijndael's key schedule; each sub key is the same size as the state. Then the sub key is added by coupling individual byte of the state with the coincidencing byte of the sub key using bitwise XOR.

2) *Algorithm 2: AES Algorithm*

```
Input: Cipher(byte in[16], byte out[16],
Output: key_array round_key[Nr+1])
begin
byte state[16];
state = in;
AddRoundKey(state, round_key[0]);
for i = 1 to Nr-1 stepsize 1 do SubBytes(state);
    ShiftRows(state);
    MixColumns(state);
    AddRoundKey(state, round_key[i]);
end for
SubBytes(state);
ShiftRows(state);
AddRoundKey(state, round_key[Nr]);
End
```

C. Elliptic Curve Cryptography (ECC)

Elliptic curve is a form of finite fields and cater fixed number of cipher text length.
Require: N: blended number to be factored,
E: elliptic curve,

$$P_0 = (x_0, y_0, z_0)$$

$\in E(Z_N)$: initial point, B_1: Sleekness bound for phase 1, B_2: sleekness bound for phase 2, $B_2 > B_1$

V. Experimental Results

The proposed work is to provide the secure mechanism for file upload and hosting services thus the additional resource consumption and the requirements are evaluated in these sections. The evaluated performance parameters are reported as:

Response Time The amount of time required to accept the user request and get respond by the server is given as the response time of the system. The experimental result is showed in Fig. 2.

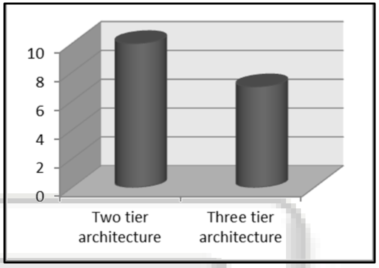

Fig. 2: Response time

A. Space Overhead

The amount of data increases during the file encryption and the data transmission is given as the space overhead. That is evaluated in terms of KB (kilobytes) and reported using the below Fig. 3.

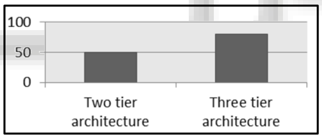

Fig. 3: Space overhead

B. Encryption Time

The amount of time required to encrypt or decrypt an input file is known as the encryption time of the system. The encryption time of the system is measured in terms of seconds and reported in the below figure. In this diagram the red line shows the time consumption of the system during the decryption time and the blue line shows the amount of time consumed during the encryption. And it can be reported below Fig 4.

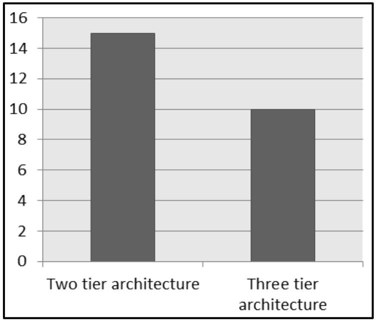

Fig. 4: Time consumed during encryption

VI. Conclusion

In this paper, we implemented a secure data mechanism to solve the problem of data security and privacy in mobile cloud computing. We first implemented MD5 algorithm to realize the access control in cloud computing, and show that the situation when mobile users may arbitrarily join or leave the mobile network makes these approaches not suitable to be used in mobile cloud computing. Afterwards, in this paper we explored AES-encryption scheme to make mobile users easily encrypt the data which are uploaded in cloud system. Then ECC algorithm is implemented successfully and experimental results proved reduced number of response time, space overhead and encryption time.

References

[1] Ayesha Malik, Muhammad Mohsin Nazir, "Security Framework for Cloud Computing Environment: A Review," Journal of Emerging Trends in Computing and Information Sciences, 2012.

[2] Shashi Mehrotra Seth, Rajan Mishra, “Comparative Analysis of Encryption Algorithms for Data Communication," IJCST Vol. 2, June 2011.

[3] Simoens, P., De Turck, F., Dhoedt, B., Demeester, P “Remote Display Solutions for Mobile Cloud Computing” Computer Vol.44 No.8pp.46–53,2011

[4] Shahryar Shafique Qureshi1 , Toufeeq Ahmad1, Khalid Rafique2, Shuja-ul-islam3 “Mobile cloud computing as future for mobile applications – implementation methods and challenging issues” , 2011.

[5] Hoang T. Dinh, Chonho Lee, Dusit Niyato, and Ping Wang. “A Survey of Mobile Cloud Computing: Architecture Applications, and Approaches”, In Wireless Communications and Mobile Computing 2011.

[6] Kumar, K., Lu, Y.-H.: Yung-Hsiang Lu, “Cloud Computing for Mobile Users” Can Offloading Computation Save Energy? Computer, Vol. 43, No.4, pp.51– 56 , 2010 .

[7] Mell P, Grance T ”The NIST definition of Cloud Computing “NIST, Special Publication pp.800–145, Gaithersburg, MD

[8] Zhang Q, Cheng L, Boutaba R “ Cloud Computing: state-of-the-art and research challenges” Journal of Internet Services Applications Vol.1 No.1, pp. 7–18, 2010

[9] Pearson, S., Y. Shen, and M. Mowbray, “A Privacy Manager for Cloud Computing”, in Proceedings of the 1st International Conference on Cloud Computing , Springer-Verlag: Beijing, China , pp. 90-106,2009.

[10] Wang, Q., et al., “Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing”, in Computer Security – ESORICS 2009, M. Backes and P. Ning, Editors, Springer Berlin / Heidelberg , pp. 355-370,2009.