International Journal of Scientific Research & Engineering Trends
Volume 6, Issue 1, Jan-Feb-2020, ISSN (Online): 2395-566X

Secure Data Sharing in Mobile Cloud Computing

Arshad Azmi, Associate Aarti Gautam Dinkar
Department of Computer Sciences
School of ICT
Gautam Buddha University, Greater Noida, U.P, India
aamee84@gmail.com, aarti@gbu.ac.in

***Abstract* - Mobile cloud computing (MCC) is one of the technologies essential in today's mobile environment run by using mobile devices in cloud environment. MCC has been imagined as the cutting-edge technology of IT enterprise. It combines both the features of mobile computing and cloud computing. It moves the application programming and databases to the unified datacenters, where the administration of the information and administrations may not be completely reliable. With the advancements in the field of MCC, security is a major issue. Some of the common security issues regarding data are like- risk of data theft, violation of privacy rights and loss of physical security, handling of encryption and decryption keys, data storage security. In this paper we provide a solution for data storage security and handling of encryption and decryption keys issues. In this paper we proposed a method which is implemented by combining the concepts of Diffie- Hellman Key Exchange (DHKE) algorithm and Blowfish algorithm. In this proposed scheme, at first a computer user will encrypt a file using a secret key generated by blowfish algorithm. Then using DHKE protocol a shared private key will be generated for two users who are trying to communicate. Presently if the subsequent user needs to decrypt the document encoded by the primary user, he/she needs to use the mutual key for authorization. When the permission is granted, the document can be decrypted by using blowfish algorithm. These proposed mechanisms can securely share and store data in cloud environment.**

***Keywords*- Mobile Cloud Computing, Diffie-Hellman Key Exchange, Third Party Auditor, Cloud Service Provider, Encryption Service Provider, Decryption Service Provider.**

I. INTRODUCTION

Cloud computing, refers to sharing files, software, and data via a network, in this case the Internet. The data is store on physical servers maintained and controlled by a cloud computing. Cloud computing is a rising computing paradigm in which assets of the processing foundation are given as administrations over the Internet. As these paradigms also brings new challenges to data security.

Mobile cloud computing (MCC) is another model in which the cloud computing resources and organizations made accessible for mobile phones. Mobile computing is not meant just for smartphone users but also for wide ranging mobile subscribers as well. This MCC innovation is characterized in the light of three noteworthy ideas: programming mobile applications accessible in the device, equipment mobile devices and correspondence network, information transfer and different protocols.

MCC is a method or model in which portable applications are assembled, controlled and facilitated utilizing distributed computing innovation. A mobile cloud approach empowers engineers to construct applications composed particularly for portable clients without being bound by the portable working framework and the registering or memory limit of the cell phone. Versatile distributed computing focused are for the most part gotten to by means of a portable program from a remote web server, normally without the requirement for introducing a customer application on the beneficiary telephone.

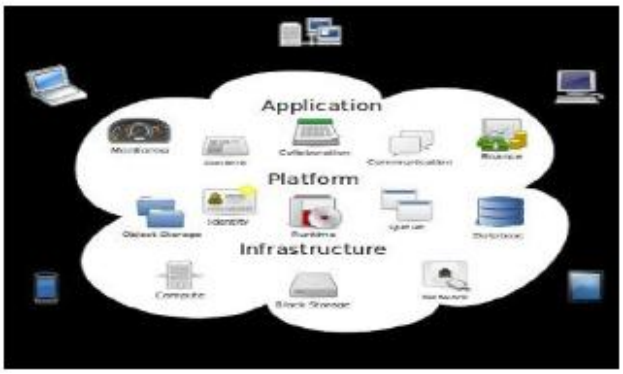

Fig.1. Mobile Cloud Computing Architecture.

1. Security challenges in MCC:
Security has gotten increasingly critical to PC clients, organizations, and the military. Security turned into a significant worry with the coming of web and the historical backdrop of security permits a superior

comprehension of the development of security innovation. The internet structure itself considered numerous security dangers to happen. The modified architecture of the internet can lessen the possible attacks that can be sent over the internet.

With the advancements in the field of versatile mobile computing, security is a significant issue. As discovered while reviewing the existing and proposed frameworks of MCC, several major issues and challenges of MCC were being arranged. Here we present some of the major security challenges in MCC:

- Data Security and Privacy Issues.
- Attack at the End-user Mobile Device.
- Information Security Issues.
- Mobile Cloud Infrastructure Issues.

So these are some issues and challenges in MCC, In this paper work we are concerned with the data security issue where we provide a solution for handling of encryption and decryption keys and data storage security.

II. RELATED WORKS

Qihua Wang et al. in [1] proposed a method to solve the data leakage problem in SaaS collaboration systems by reducing human errors. The authors have designed a series of mechanisms to provide defence in depth against information leakage. First, authors have allowed enterprises to encode their organizational security rules as mandatory, so as to impose restrictions on their employees. Second, design an attribute-based recommender that suggests and prioritizes potential recipients for user's files, reducing errors in the choices of recipients. The authors have implemented a prototype solution and performed experiments on data collected from real-world collaboration systems.

Ruixuan Li et al. in [2] proposed a lightweight data sharing scheme (LDSS) for mobile cloud computing which adopts Ciphertext policy attribute based encryption (CP-ABE), an access control technology used in normal cloud environment. To lessen the user revocation cost, the authors acquaints attribute description fields to apply lazy-revocation, which is a prickly issue in program based CP-ABE systems. Limitation of this technique is high computational overhead because they are using lazy re-encryption technique.

Wang W et al. in [3] proposed an owner-write-users-read applications to encrypt every data block with a different key so that flexible cryptography-based access control can be achieved. Proposed over-encryption and/or lazy revocation to prevent revoked users from getting access to updated data blocks. The authors have design mechanisms to handle both updates to outsourced data and changes in user access rights. But this could lead to high overhead.

Jason Crampton et al. in [4] proposed a group of generic key task plans and compare their advantages. They note that each plan in the writing is basically a case of one of our generic schemes and then conduct an analysis of the Akl-Taylor scheme and propose a number of improvements. The authors introduce a technique for exploiting the respective advantages of different schemes. But this key assignment schemes is not similar for all the broadcast encryption.

Yu Jin et al. in [5] design a secure and lightweight data access control scheme based on Ciphertext-Policy Attribute-based Encryption (CP-ABE) algorithm, which can protect the confidentiality of outsourced data and provide fine-grained data access control in MCC. The proposed scheme can improve the overall system performance by greatly reducing the computation overheads in encryption and decryption operations, provide flexible and expressive data access control policy, and meanwhile enable data owners to securely outsource most of the computation overheads at mobile devices to cloud servers.

Vinothini et al. in [6] proposed a harder encryption with enhanced public key encryption protocol for security and can be implemented into any network to provide better security. The authors have upgraded the hardness in security by improving the Diffie-Hellman encryption algorithm by including some greater security codes in current algorithm.

Saikumar Manku et al. in [7] designed and analysed a Blowfish encryption algorithm for information security. The work is accomplished for networking and communication application for improved network security and safeguard applications. In the proposed Blowfish algorithm reduce rounds of algorithm and proposed single blowfish round. The design is done by Xilinx ISE software using the language of VHDL.

III. PROBLEM IDENTIFICATION

First problem is that, Lightweight data sharing schemes uses Lazy re-encryption technique but this technique brings heavy computational overhead. Lazy re-encryption means re-encrypting the data or information when the user's access privileges to the data are revoked. But this processtakes a lot of time. Second problem is third party auditor. Third party auditor which act as a mediator and its function is to store the keys and verifies key when a user wants to access some data from cloud. But the data is not properly managed by TPA. So we cannot fully trust on third party auditor.

International Journal of Scientific Research & Engineering Trends
Volume 6, Issue 1, Jan-Feb-2020, ISSN (Online): 2395-566X

IV. METHODOLOGY USED

The objective of the research is to design sharing schemes that can be used to secure the data in mobile cloud. The proposed strategy is executed by combining the ideas of Diffie-Hellman key exchange and Blowfish algorithm. In this new strategy at initial a PC user will encrypt a file using a master key created by Blowfish algorithm. At that point using Diffie-Hellman protocol a share private key will be created for two users who are trying to communicate over an insecure channel. Presently if the second user needs to decrypt the data encrypted by the main user he/she needs to use the shared key for that. Once the permit is granted then the data will be decrypted by Blowfish algorithm. The tool used for the whole process is Java NetBeans.

1. Blowfish Algorithm

Blowfish is a symmetric block cipher that can be frequently used for encryption and safeguarding of information. It takes a variable-length key of 32 bits to 448 bits, making it good for securing information. Blowfish was developed in year 1993 by Bruce Schneier as a fast, free contrasting option to existing encryption algorithm.

2. Algorithm description

2.1 Key-expansion- It will change over a key of at most 448 bits into a couple of sub keygroups totalling 4168 bytes. Blowfish uses broad number of sub keys. These keys are creating prior to any information encryption or decoding.The p-array comprises of 18, 32-bit sub-keys: P1, P2, and P18 Four 32-bit S-Boxes comprises of 256 entries each:

S1, 0, S1, 1, S1, 255
S2, 0, S2, 1, S2, 255
S3, 0, S3, 1, S3, 255
S4, 0, S4, 1, S4, 255

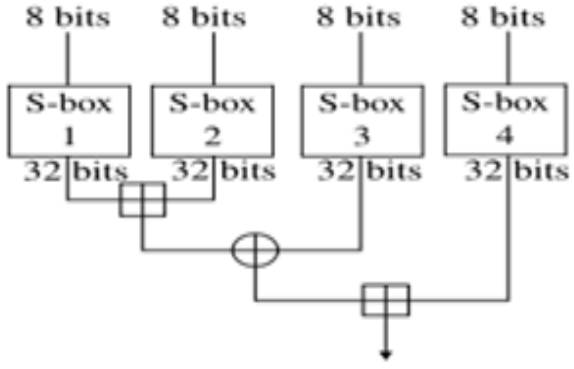

Fig .2. Key expansion.

First initialise the P-array and then four S-boxes, in order with a fixed string. This string comprises of the hexadecimal digits of pi (less the initial 3): P1 = 0x243f6a88, P2 = 0x85a308d3, P3 = 0x13198a2e, P4 = 0x03707344, etc.

Data Encryption: It is having a capacity to repeat 16 times of system. Each round consist of key-subordinate change and a key and information subordinate substitution.The 64 bits cipher is divided into 32 bit halves, left one is called xL and right one is xR. Divide x into two 32-bit halves: xL, xR

For i = 1 to 16: (here i=no. of key)
xL = xL XOR Pi
xR = F (XL) XOR xR (here F= fiestal round)
Swap xL and xR
Swap xL and xR (Undo the last swap)
xR = xR XOR P17
xL = xL XOR P18
Recombine xL and xR

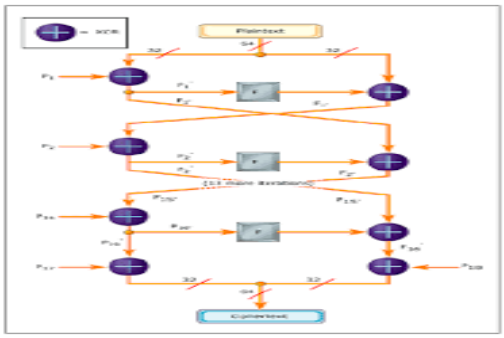

Fig.3. Data encryption.

The above figure shows how the encryption is process is carried out. The plaintext is divided into two halves 32 bit. XOR P1 has first 32 bits of the key, XOR P2 has second 32-bits of the key, and so on for all the bits of the key (up to P14). Repeat the cycle through the bit keys until all the P-array has been XORed with key bits. (For every short key, there is atleast one identical longer key; for example, if A is a 64 bit key, then AA, AAA, and so forth, are equivalent keys.)

3.Diffie Hellman Key Exchange

The motivation behind this algorithm is to empower two clients to safely exchange a key that can be used for encryption of messages. The algorithm itself is restricted to exchange of master keys. The Diffie-Hellman algorithm depends for its effectiveness on the difficulty of computing discrete logarithms. Key Sharing Through DHKE:

Let us consider, p and g be equivalent to 19 and 7 separately.

Client 1 takes private key x = 6 and client 2 takes private key y=9.

Presently client 1 figures open key R1 = (gx) mod p= (76) mod 19=1

Client 2 additionally figures open key R2= (gy) mod p = (79) mod 19=1

For client 1 shared key k1= (R2x) mod p = (16) mod 19=1

For client 2 shared key k2= (R2x) mod p= (19) mod 19=1
So k1 = k2.
Now if an eavesdropper sends a number 3 when user 1 asks for shared key, the request for encrypted file will be denied.

V. RESULT & ANALYSIS

We have analysed securing and sharing using two algorithm-Blowfish and Deffie-Hellman key exchange algorithm. Security and storage are major issues which are kept in mind. When the file is being sent to the recipient, if somehow it gets attacked the attacker still won't be able to see the content as the text/image will remain encrypted and the attacker won't have the secret key of blowfish encryption. Analysis and result are discussed below.

Fig.4. Image to be encrypted.

Fig.5. Encrypted image.

Fig.5 shows the cipher text that is obtained as a result of encryption of the image file shown in fig.4, which when decrypted produces exactly same as the original file.

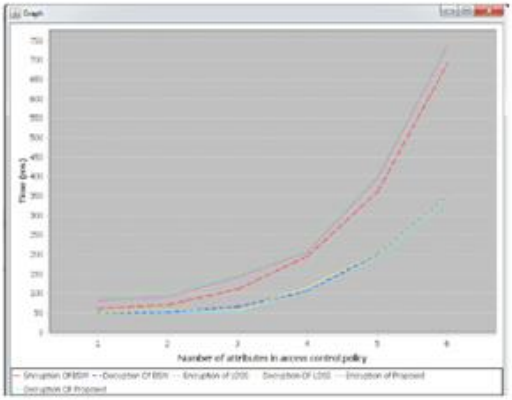

Fig.6. Comparing Time taken in encryption and decryption process.

As it is known that blowfish is a fast algorithm, it encrypts data on large 32-bit microprocessors at a rate of 26 clock cycles per byte reducing the time taken by the existing scheme. It improve the time taken during the encryption and decryption process. Fig.6 shows the results of proposed and LDSS encryption and decryption times.

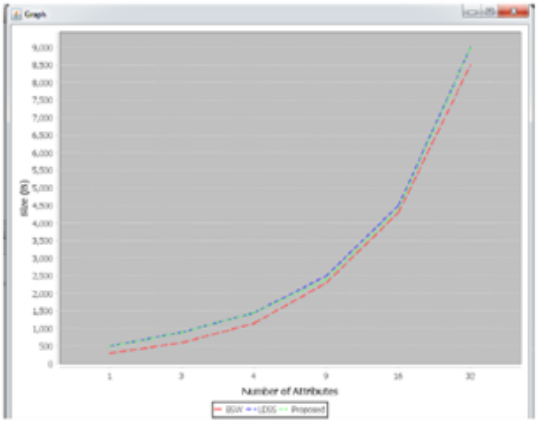

Fig.7. Comparing Storage occupied by attributes Keys.

Cloud is known as a large data space but this large space is not available for all the users. Some spaces are paid and free ones are not large space. By using proposed algorithm you can reduced more spaces in your cloud space. As the proposed system first encrypt the file, generate keys and then it will upload to the cloud, the size of the encrypted files and the keys are very small which makes your cloud spaces free. So Fig.7 shows the result of the storage occupied by keys.

Table I: Encryption Process Time.

USER	Encryption Time (LDSS)	Encryption Time (Proposed)
1	78 ms	74 ms
2	98 ms	94 ms

The above table shows the result of encryption timing by the existing and proposed schemes. By using existing scheme time taken for encryption by one user is 78 ms but by using proposed scheme it is 74 ms, for two user time taken by the existing and proposed scheme is 98 ms and 94 ms respectively.

Table II: Decryption Process Time

USER	Decryption Time (LDSS)	Decryption Time (Proposed)
1	50 ms	47 ms
2	51 ms	48 ms

The above table shows the result of decryption timing by the existing and proposed schemes. By using existing

scheme time taken for decryption by one user is 50 ms but by using proposed scheme it is 47 ms, for two user time taken by the existing and proposed scheme is 51 ms and 48 ms respectively.

Table III: Storage analysis

Key	LDSS	Proposed
1	510 bytes	480 bytes
3	800 bytes	750 bytes

The above table shows the result of storage captured by the existing and proposed schemes. By using existing scheme storage for one key is 510 bytes but by using proposed scheme it is 480 bytes, for three key storage used by the existing and proposed scheme is 800 bytes and 750 bytes respectively.

VI. CONCLUSION & FUTURE SCOPE

With the approach of internet, security of data that is being transferred online has become very difficult to ensure. Diffie Hellman and Blowfish methods when individually applied, faces a lot of security threats like man in the middle attack, data authentication etc. But as we know the cryptosystems used in today's world is based on these two basic algorithms, so we cannot compromise on these. Thus, the proposed system attempts to ensure that the data is read by only intended user by providing a two level security system and overcoming most of the shortcomings faced by existing algorithms.

Our aim always would be to modify this algorithm to increase the level of security. To increase the security aspect, future enhancement can try to prevent replay attacks. Because if someone is repeatedly trying to access the encrypted file with wrong keys, it might very well be possible that the user is trying permutation and combination to get the correct secret base. So we can include timestamp for this reason. If multiple timestamps are being received from a single source it will be easy to comprehend that request is probably coming from an attacker.

REFERENCES

[1]. Qihua Wang, Hongxia Jin. "Data leakage mitigation for discretionary access control in collaboration Clouds" The 16th Acm Symposium On Access Control Models And Technologies (Sacmat), Pp.103-122, Jun. 2011.

[2]. Ruixuan Li, Chenglin Shen, Heng He, Zhiyong Xu, And Cheng-Zhong Xu, "A Lightweight Secure Data Sharing Scheme For Mobile Cloud Computing" Ieee Transactions On Cloud Computing, 2014.

[3]. Wang W, Li Z, Owens R, Et Al. "Secure And Efficient Access To Outsourced Data"In: Proceedings Of The 2009 ACM workshop on Cloud computing security. Chicago, USA: ACM pp. 55-66, 2009.

[4]. Crampton J, Martin K, and Wild P. "On key assignment for hierarchical access control.in: Computer Security Foundations Workshop". IEEE press, pp.14-111, 2006.

[5]. Kan Yang, Xiaohua Jia, Kui Ren, Bo Zhang, and Ruitao Xie: "DAC-MACS: Effective Data Access Control for Multiauthority Cloud Storage Systems". IEEE Transactions on Information Forensics and Security, Vol. 8, No. 11, pp.1790-1801, 2013.

[6]. Yu S., Wang C., Ren K. et al. "Attribute based data sharing with attribute revocation" in: Proceedings of the 5th International Symposium on Information, Computer and Communications Security (ASIACCS), New York, USA: ACM press pp. 261-270, 2010.

[7]. Di Vimercati S D C, Foresti S, Jajodia S, et al. "Over-encryption: management of access control evolution on outsourced data" in: Proceedings of the 33rd international conference on Very large data bases. Vienna, Austria: ACM, pp. 123-134, 2007.

[8]. Zhibin Zhou and Dijiang Huang, "Efficient and Secure Data Storage Operations for Mobile Cloud Computing", Network and service management (CNSM), 2012 8th internationalconference and 2012 workshop on Systems virtualization management (SVM), 13 December 2012.

[9]. Bruce Schneier "Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)"Springer-Verlag, 1993.

[10]. Yu S., Wang C., Ren K., Lou W. "Achieving Secure, Scalable, and Fine-grained DataAccess Control in Cloud Computing". INFOCOM 2010, pp. 534-542, 2010.

[11]. Bethencourt J, Sahai A, Waters B. "Ciphertext-policy attribute-based encryption" in:Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP). Washington,USA: IEEE Computer Society, pp. 321-334, 2007.

[12]. Jia W, Zhu H, Cao Z, et al. "SDSM: a secure data service mechanism in mobile cloud computing" in: Proceedings of 30th IEEE International Conference on Computer Communications. Shanghai, China: IEEE, pp. 1060-1065, 2011.

[13]. Ashwak ALabaichi, Faudziah Ahmad "Security Analysis of Blowfish algorithm" in: Proceedings of IEEE conference. IEEE 2013.

[14]. Abdul Raoof Wani, Q.P. Rana, Nitin Pandey "Cloud Security Architecture Based on User Authentication and Symmetric Key Cryptographic Techniques" in: Proceedings of 6th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), Sep. 20-22, 2017.

[15]. Amina H Gamlo, Ning Zhang, Omaimah Bamasag "Mobile Cloud Computing: Security Analysis" in:

Proceedings of 5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering 2017.

[15]. Naseer Amara, Huang Zhiqui, Awais Ali "Cloud Computing Security Threats and Attacks with their Mitigation Techniques" in: Proceedings of 2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery 2017.

[16]. Saikumar Manku1 and K. Vasanth. "Blowfish Encryption Algorithm for Information Security". ARPN Journal of Engineering and Applied Sciences, JUNE 2015.

[17]. Tingyuan Nie and Teng Zhang. "A Study of DES and Blowfish Encryption Algorithm". IEEE 2009

[18]. Gurjeevan Singh, Ashwani Kumar, K. S. Sandha. "A Study of New Trends in Blowfish Algorithm". International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622

[19]. Navdeep Singh, Dr. Rahul Malhotra, "Privacy Preservation for File Sharing Scheme Using Secured File Block Id with Binary Trees and Encryption", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, Issue 2, February 2016, ISSN (Print): 2320-9798,

[20]. Honggang Wang, Shaoen Wu, Min Chen, Wei Wang, "Security Protection between Users and the Mobile Media Cloud", Security in Wireless Multimedia Communications, IEEE Communications Magazine. March 2014.

[21]. Athanasios V. Vasilakos, Keqin Li, Fellow, Albert Y. Zomaya, "SeDaSC: Secure Data Sharing in Clouds", IEEE Systems Journal (Volume: 11, Issue: 2, June 2017), DOI:10.1109/JSYST.2014.2379646.

[22]. Yu Jin, Chuan Tian, Heng He, FanWang, "A Secure and Lightweight Data Access Control Scheme for Mobile Cloud Computing", Big Data and Cloud Computing (BDCloud), 2015 IEEE Fifth International Conference, DOI: 10.1109/BDCloud.2015.57, 29 October 2015.

[23]. Vinothini, Saranya, Vasumathi. "A Study on Diffie-Hellman Algorithm in Network Security", International Journal of Engineering and Computer Science ISSN: 2319-7242 Volume 3 Issue 7 July, 2014.

[24]. K.Suganya, K.Ramya. "Performance study on Diffie Hellman Key Exchange Algorithm", International Journal of Engineering and Computer Science ISSN: 2321-9653 Vol. 2 Issue III, March 2014.

Authors

First Author – Arshad Azmi, Post Graduate, School of ICT, Gautam Buddha University and aazmee84@gmail.com.

Second Author – Aarti Gautam Dinkar, Faculty Associate, School of ICT, Gautam Buddha University.