International Journal of Research in Advent Technology, Vol.6, No.8, August 2018
E-ISSN: 2321-9637
Available online at www.ijrat.org

A Cryptographically Secured Lightweight Protocol For Encryption Based Data Sharing For Mobile Cloud Computing

Dr. Mohammed Abdul Waheed[1], Shireen Banu[2]
Associate Professor, Department of Computer Science and engineering, Visvesvaraya Technological University Centre for PG Studies, Kalaburagi, India[1].
Student, Department of Studies in Computer Science and engineering, Visvesvaraya Technological University Centre for PG Studies, Kalaburagi, India[2].
Email: Dr.mawaheed@gmail.com[1], Syedashirin206@gmail.com[2]

Abstract—Mobile devices and its applications have upset the way we store and offer information. It is turning into a stockroom of clients' close to home data. Unfortunately, a large portion of these information are put away in a decoded arrange, inclined to security dangers. In this paper, we propose a lightweight, computationally proficient convention, called CLOAK, for the cell phone. Shroud depends on stream figure and takes the assistance of an outer server for the age and dispersion of cryptographically secure pseudo-arbitrary number (CSPRN). Keeping in mind the end goal to improve the security of our convention, we utilize the idea of symmetric key cryptography. In CLOAK, the center encryption/decoding activity is performed inside the Mobile devices to secure information at its cause. The security of CSPRN is guaranteed utilizing duplicity technique. In CLOAK, all messages are traded safely amongst portable and the server with common character check. We assess CLOAK on Android advanced mobile phones and utilize Amazon Web administrations for producing CSPRN. Furthermore, we display assault examination and demonstrate that the beast drive assault is computationally infeasible for the proposed convention.

***Index terms*—** Mobile device, Cloud Computing, Encryption, Decryption, Mobile Cloud Computing.

1. INTRODUCTION

Developments in portable innovation, imaginative applications and diminishing costs of cell phones, wearable PCs and other Mobile gadgets (MD) have contributed altogether in expanding notoriety of cell phones in our advanced way of life. Since, MDs are intended for individual use, usually utilized as an archive for putting away client's close to home data, for example, client profile, passwords, financial balance data and medicinal records. All the more fundamentally, the information is put away in an unmistakable content arrangement in a MD, which can be effectively recovered and prompt genuine security confusions. Security dangers on MD can be from different sources including malwares, outsider applications, listening in finished remote system, burglary and lost gadgets. Thus, numerous organizations don't enable workers to store corporate information in cell phones or utilize the corporate system through individual gadgets. Versatile distributed computing (MCC) is a developing exploration territory concentrating on supplementing the capacity and computational necessity of MD by using the cloud framework. By connecting with cloud, MD can convey different administrations to the client, for example, medicinal services, portable business and online instruction. Clients can transfer and store information (photographs, medicinal records) from their MD to the cloud and can impart them to others. Moreover, MD can off load calculation concentrated assignments to the cloud to beat its assets restriction and for sparing battery. In any case, security is a noteworthy worry in MCC , especially for portable applications sending decoded individual data over unreliable remote medium to the cloud. Information encryption is likewise required for ensuring client's information against outside and interior assaults inside the cloud condition.

Encryption/decoding calculations are generally utilized for giving security to client's close to home data. Encryption is a procedure of changing over plaintext (PT) information into a vast code called ciphertext (CT) and a decoding calculation is utilized for upsetting the CT to unique PT. In this paper, we concentrate our talk on the encryption and unscrambling for the MD. There are three fundamental methodologies for the same. The encryption/unscrambling activities can be performed inside the MD, which we allude as a versatile driven approach. Creators have examined

the attainability of executing the standard symmetric and awry cryptographic calculations (AES) in the MD. Nonetheless, because of high computational intricacy, the standard encryption calculations are not effective for the asset obliged MDs. The execution can be enhanced by S-Box improvement and decreasing the quantity of rounds however more lightweight encryption/unscrambling calculation is expected of the MDs.

- Secondly, the MD can offload records and play out the calculation serious encryption/decoding errands to the cloud or an outer server (ES). By offloading the undertaking, MD can defeat its asset restrictions and can effectively deal with substantial records in a generally brief time span. Specialists have proposed answers for address the security concerns related with offloading records, for example, utilizing a trusted outsider (TTP), secure channel, versatile VPN, document part and multipath TCP. The greater part of these methods rely on middle of the road server or foundation, which may not be attainable for some MCC applications, similar to moment photograph transferring.

• A moderate approach is to share the calculation by encoding the essential parts of a record in the cell phone and offloading the rest of the assignments to the cloud.

In this paper, we propose a convention for scrambling and decoding inside the MDs in a versatile cloud condition, alluded as CLOAK. We will likely secure individual data put away in MD (pictures, txt, doc), of size in the scope of 5-10 MBs. The CLOAK convention depends on stream figure and takes the assistance of a cloud or an outside server (ES) for creating the key-stream or a cryptographically secure pseudo-arbitrary number (CSPRN). The upside of utilizing stream figure as the premise of our convention, is that it is less calculation escalated contrasted with piece figure and can without much of a stretch be dealt with by existing MDs. Stream figure is a cryptographically secure encryption calculation, utilized as a part of different conventions (WPA, TLS), applications (Internet Explorer, Google Chrome, and Firefox) and in correspondence measures (GSM, 3GPP, LTE). The plan contemplations of our convention are as per the following:

• To outline a lightweight encryption convention for MD. We expect that a solitary size of 5 to 10 MB is satisfactory for pictures and records in txt, doc groups. The convention must have the capacity to deal with such records on most MDs that are right now accessible in the market.

• The encryption/unscrambling task must be performed in a worthy time span.

• The clients must have the capacity to control the encryption unscrambling activity. This is essential for building up client's certainty on the framework.

• All tasks on the PT must be performed locally on the MD and the calculations on ES ought not influence the security of the convention.

• Finally, and above all, the convention must be cryptographically secure.

One of the real difficulties of a stream figure is the age and appropriation of the key-stream or CSPRN (C).In CLOAK, we offload this assignment to an outside server (ES) in the cloud to spare assets of the MDs. What's more, the cloud can be utilized for sharing the encoded records with various beneficiaries. To address the security of the CSPRN (C), we propose two level CSPRN adjustment. Right off the bat, the C is changed to C" by the ES, before transmitting it to the MD. This guarantees the security of C against the vulnerabilities of questionable remote media. Moreover, we have to guarantee that exclusive the proposed beneficiaries ought to have the capacity to unscramble the document. For this, we play out another alteration on C in the MD to create C0. The C0 is utilized for the encoding a document and must be decoded by the beneficiaries having the key k.

Mystery of the key is the fundamental prerequisite of every single cryptographic calculation and enemy may force different assaults to recover the same. Since we are utilizing C0 for the encryption procedure, the age system assumes a urgent part in the security of the convention. In the proposed calculations, we utilize the key (k) for producing C0. We demonstrate that, for an obscure k, it is computationally infeasible for an enemy to produce C0 through a beast drive assault. Furthermore, we likewise demonstrate that the convention can oppose assaults like two time cushion, known plaintext, algebraic, Man-in-the-center, insider, pantomime and DoS. We assess the execution of CLOAK on Android-based cell phones and utilize Amazon Web administrations (AWS) for CSPRN and concentrate the unpredictability of the calculation (i.e., time, space, preparing power) by differing the document measure.

2. RELATED WORK

R.Buyya et.al.[1] identified and analyzed roots and dimensions of heterogeneity in MCC. Also, some of the major MCC challenges are described based on literature. It was argued that MCC is a more heterogeneous domain compared to cloud computing due to divergent computing (mobile computing and cloud computing) paradigms and networking technologies. The taxonomy of heterogeneity roots in MCC was also devised. They analyzed and classified the pivotal roots of heterogeneity and related approaches that handle certain classes of heterogeneity. According to the types of heterogeneity in each landscape, they categorized heterogeneity of cloud computing, mobile computing, and wireless networks into two classes, namely vertical and horizontal. The survey advocated that although there are several academic and industrial solutions, there is no suitable ground yet that can cover the highlighted challenges for end-users, application developers, mobile cloud service providers, and third parties.

R. Kemp et.al.[2] proposed a framework for computation offloading for smartphones, a recently rediscovered technique, which can be used to reduce the energy consumption on smartphones and increase the speed have evaluated the Cuckoo framework with two real world smartphone applications, an object recognition application and a distributed augmented reality smartphone game and showed that little work was required to enable computation offloading for these applications using the Cuckoo framework.

Z. Xiao and Y. Xiao[3] have systematically studied the security and privacy issues in cloud computing based on an attribute-driven methodology. They identified the most representative security/privacy attributes (e.g., confidentiality, integrity, availability, accountability, and privacy-preservability), as well as discussing the vulnerabilities, which may be exploited by adversaries in order to perform various attacks. Defense strategies and suggestions were discussed as well. They believe this review will help shape the future research directions in the areas of cloud security and privacy.

M..R..Baharon et.al.[4] proposed the new Lightweight Homomorphic Encryption (LHE) scheme that allows mobile users to outsource their data in a secure and privacy preserved manner. Moreover, the scheme enables ciphertext data to be processed under both addition and multiplication operations without decryption. They have compared LHE with another related scheme through the experiments that have been developed using Matlab software. There effort was mainly focused on the evaluation of the total execution time of the two schemes by providing comprehensive comparisons between them. The results show that LHE can operate faster than the compared scheme as it has less complexity in terms of computation.

3. IMPLEMENTATION

The CLOAK protocol is based on stream cipher and takes the help of a cloud or an external server (ES) for generating the key-stream or a cryptographically secure pseudo-random number (CSPRN). The advantage of using stream cipher as the basis of our protocol, is that it is less computation intensive compared to block cipher and can easily be handled by existing MDs. Stream cipher is a cryptographically secure encryption algorithm, used in various protocols (WPA, TLS), applications (Internet Explorer, Google Chrome, Firefox) and in communication standards (GSM, 3GPP, LTE).

1. DATA OWNER

Interest which get matches with the put away database fields, on the off chance that it coordinates then client can get the entrance of outside server. The mobile gadget assumes a crucial part here as it gives the interface to the client with the goal that it can send the demand to the outside server keeping in mind the end goal to get the CSPRN key. The key age will happens when client gets approved by the server subsequent to sending the expected accreditations to the outer server (i.e. interesting ID, File name, File Type). For sharing the encoded document, Data proprietor sends the remarkable ID, File Name to Data User so information client can solicitations to the outer server for CSPRN and in the end can unscramble the record.

2. CSPRNG (Cryptographically Secure Pseudo Random Number Generator):

For creating the CSPRN key the most essential thing the client needs to pass three vital parameters, for example, Unique ID, File Name and CSPRN estimate. The security of essential cryptographic components to a great extent relies upon the basic irregular number Generator (RNG) that was utilized. An RNG that is appropriate for cryptographic use is known as a Cryptographically Secure Pseudo-Random Number Generator (CSPRNG). The quality of a cryptographic framework depends intensely on the properties of these CSPRNGs. Contingent upon how the created

pseudo-irregular information is connected, a CSPRNG may need to display a few (or All) of these properties:

- It seems irregular.
- Its esteem is unusual ahead of time.
- It can't be dependably imitated after age

Algorithm 1 CSPRN Generation

```
Function CSPRN_Gen (CSPRN size: cs)
    s ← random_num() ;    /* key or seed */
    sn ← random_num() ;    /* Seq Num */
    CSPRN ← NULL ;    /* Init. CSPRN */
    n ← [cs/128] ;
    while n > 0 do
        CSPRN ← CSPRN + AES(s, sn) ;
        sn ← sn + 1 ;
        n ← n – 1 ;
    return CSPRN :
```

3. MOBILE DEVICE:

In a mobile gadget, XORing is the main activity performed in the MD. For encryption, the PT is XORed with the CSPRN to create the CT and in unscrambling, the CT is again XORed with the same CSPRN to recover the first PT. In our convention, to deal with the memory impediments of the MD, we perform lump insightful XORing activity by incrementally perusing the record and CSPRN in pieces of equivalent sizes. All in all, XORing is a straightforward task with less calculation and memory necessity, which can be effortlessly executed in MD.

4. EXTERNAL SERVER

The part of the outside server is to approve the client by checking every one of the certifications and once the approval work has been done then it gives a novel CSPRN key to the approved client. The External Server can be particularly designed by the necessity of an application and the workload. The correspondences amongst MD and cloud ES can occur through any remote correspondence media, for example, Wi-Fi, 3G, 4G, UMTS, and LTE.5

5. DATA USER:

For sharing the encoded document to information client, the information proprietor needs to pass the extraordinary ID and filename. So at the season of decoding stage information client can really send these documents to outer server keeping in mind the end goal to get the CSPRN key then outside server confirms the information client once it gets approved the outer server will produce the CSPRN key and information client can ready to play out the unscrambling activity.

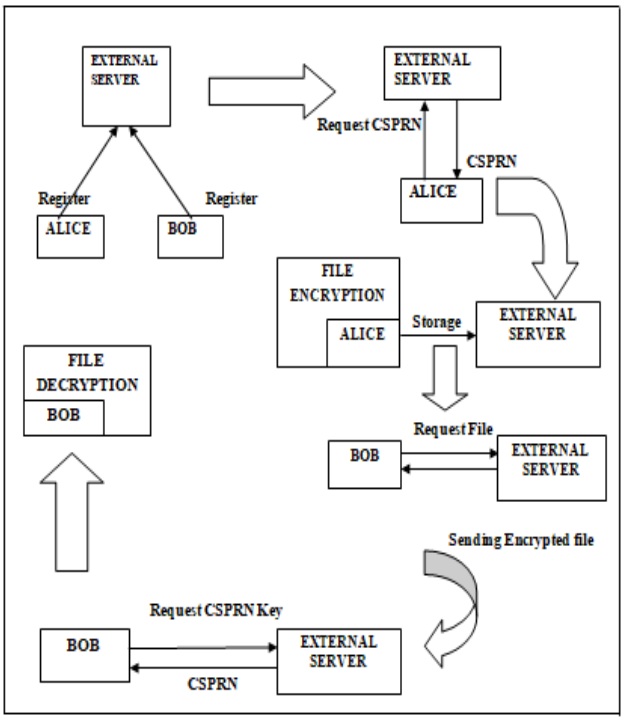

Fig 1 : System Architecture Diagram

ADVANCED ENCRYPTION STANDARD (AES): The more famous and broadly received symmetric encryption calculation liable to be experiencing these days is the Advanced Encryption Standard (AES). It is found no less than six times speedier than triple DES. A substitution for DES was required as its key size was too little. With expanding computing power, it was viewed as defenseless against comprehensive key inquiry assault. Triple DES was intended to defeat this disadvantage yet it was discovered moderate.

The highlights of AES are as per the following −

• Symmetric key symmetric piece figure

• 128-bit information, 128/192/256-piece keys

• Stronger and quicker than Triple-DES

• Provide full determination and configuration points of interest

• Software implementable in C and Java

OPERATION OF AES:

AES is an iterative as opposed to Feistel figure. It depends on 'substitution– change arrange'. It includes a progression of connected activities,

some of which include supplanting contributions by particular yields (substitutions) and others include rearranging bits around (changes).

Strikingly, AES plays out the entirety of its calculations on bytes as opposed to bits. Thus, AES treats the 128 bits of a plaintext hinder as 16 bytes. These 16 bytes are organized in four sections and four columns for preparing as a grid.

Dissimilar to DES, the quantity of rounds in AES is variable and relies upon the length of the key. AES utilizes 10 rounds for 128-piece keys, 12 rounds for 192-piece keys and 14 rounds for 256-piece keys. Every one of these rounds utilizes an alternate 128-piece round key, which is computed from the first AES key.

The schematic of AES structure is given in the accompanying delineation –

ENCRYPTION PROCESS:

Here, we limit to portrayal of a regular round of AES encryption. Each round involve four sub forms. The first round process is delineated underneath –

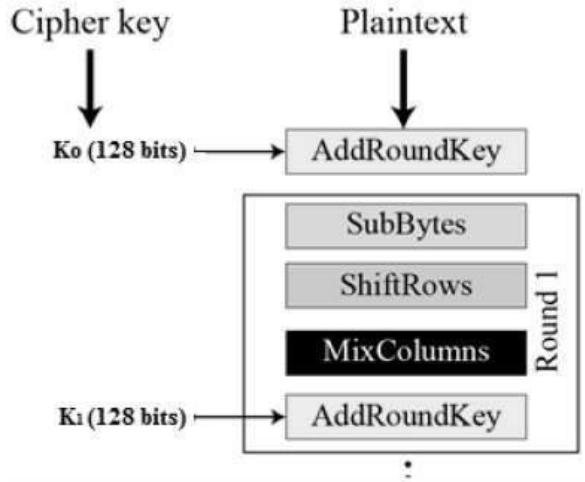

Fig 2 : AES Algorithm

Byte Substitution (SubBytes)

The 16 input bytes are substituted by looking into a settled table (S-box) given in plan. The outcome is in a lattice of four lines and four segments.

a) Shiftrow: Every one of the four columns of the lattice is moved to one side. Any passages that 'tumble off' are re-embedded on the correct side of the column. The move is completed as takes after –

- First push isn't moved.
- Second push is moved one (byte) position to one side.
- Third push is moved two positions to one side.
- Fourth push is moved three positions to one side.
- The result is another grid comprising a similar 16 bytes yet moved regarding each other.

b) MixColumns: Every segment of four bytes is currently changed utilizing an exceptional numerical capacity. This capacity takes as info the four bytes of one segment and yields four totally new bytes, which supplant the first segment. The outcome is another new grid comprising of 16 new bytes. It ought to be noticed that this progression isn't performed in the last round.

c) Addroundkey: The 16 bytes of the lattice are presently considered as 128 bits and are XORed to the 128 bits of the round key. On the off chance that this is the last round then the yield is the cipher text. Something else, the subsequent 128 bits are deciphered as 16 bytes and we start another comparable round.

DECRYPTION PROCESS

The procedure of decoding of an AES cipher text is like the encryption procedure in the switch arrange. Each round comprises of the four procedures directed in the switch arrange –

- Add round key
- Mix segments
- Shift columns
- Byte substitution

Since sub-forms in each round are backward way, dissimilar to for a Feistel Cipher, the encryption and unscrambling calculations should be independently executed, in spite of the fact that they are firmly related.

AES ANALYSIS:

In show day cryptography, AES is broadly received and bolstered in both equipment and programming. Till date, no viable cryptanalytic assaults against AES have been found. Moreover, AES has worked in the adaptability of key length, which permits a level of 'future-sealing' against an advance in the capacity to perform comprehensive key ventures.

Be that as it may, similarly concerning DES, the AES security is guaranteed just in the event that it is accurately actualized and great key administration is utilized.

Steps in AES Encryption process:

You make the accompanying AES strides of encryption for a 128-piece square:

1. Derive the arrangement of round keys from the figure key.

2. Initialize the state cluster with the square information (plaintext).

3. Add the underlying round key to the beginning state cluster.

4. Perform nine rounds of state control.

5. Perform the tenth and last round of state control.

6. Copy the last state cluster out as the encoded information (cipher text).

4. RESULTS AND ANALYSIS

In this work, we assess the appearance of the anticipated system using the stream cipher , that is less computation intensive compared to block cipher.

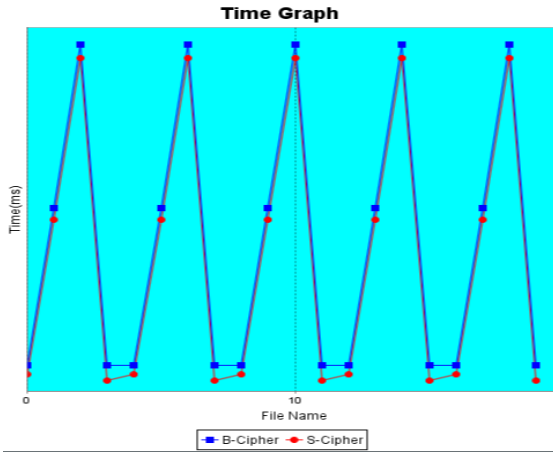

Fig 3 : Time Comparison between Block Cipher and Stream Cipher for Encrypting a Text File

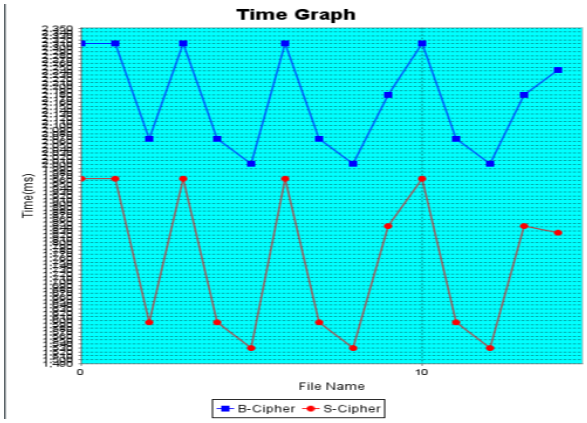

Fig 4: Time Comparison between Block Cipher and Stream Cipher for Decrypting a Text File.

1. Home Page

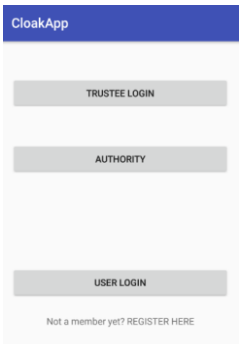

Fig 5 : Home Page

2. User 1 register to the external server and gets a unique user id . For user verification Trustee verify the user by sending the preshared key to the user provided contact number .

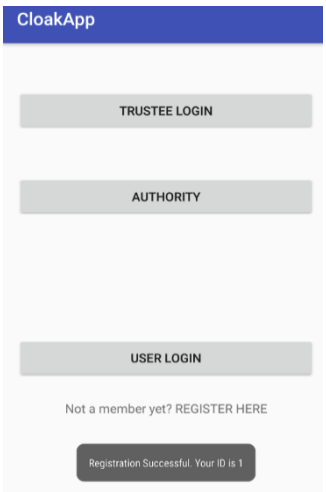

Fig 6 : User 1 registers and get assigned a unique id

3. User 1 encrypts by using requested CSPRN and upload the file to external server.

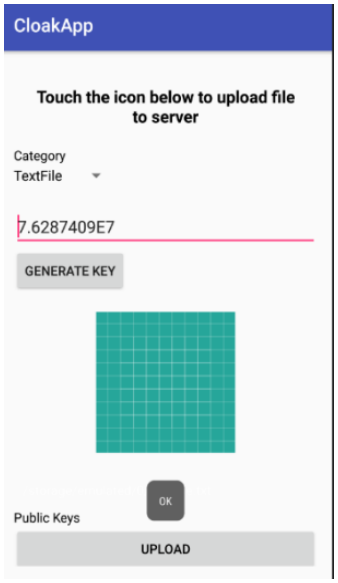

Fig 7 : User 1 Encrypts and uploads the file to server

4. User 2 registers to the external server and request file access to user 1 uploaded file from the list of registered users who uploaded there files to external server,

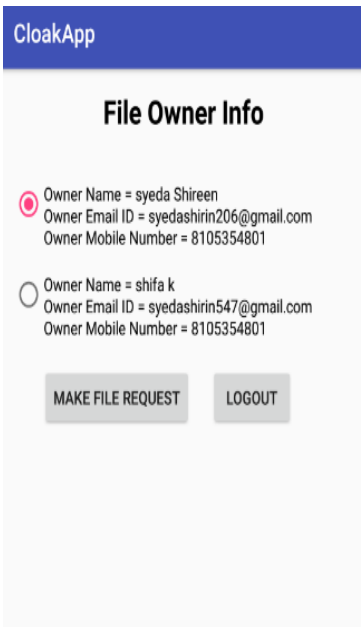

Fig 8 : User 2 request file access

5. User 2 selects the file type and file name for which it needs access.

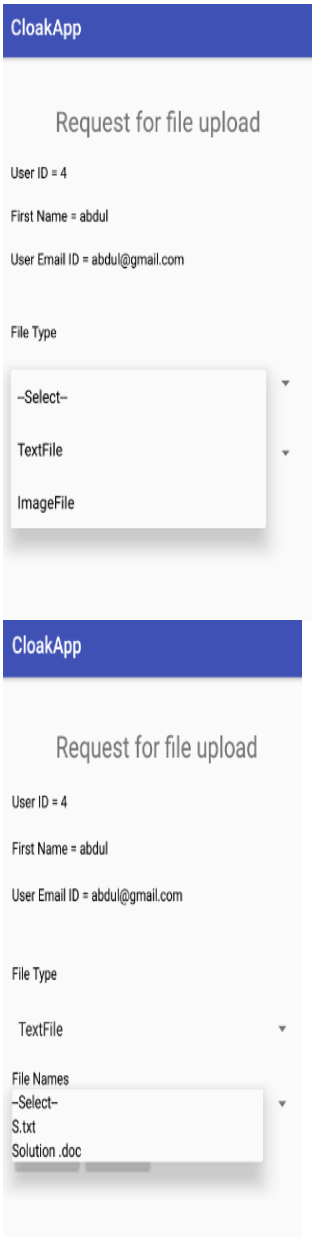

Fig 9 : User 2 selects the file type and file name.

6. User 1 Accept/Deny the user 2's request

Fig 10 : User 1 accepts/deny the user 2 file request

7. Authority grants the access to cloud for file access after user 2 credential verification.

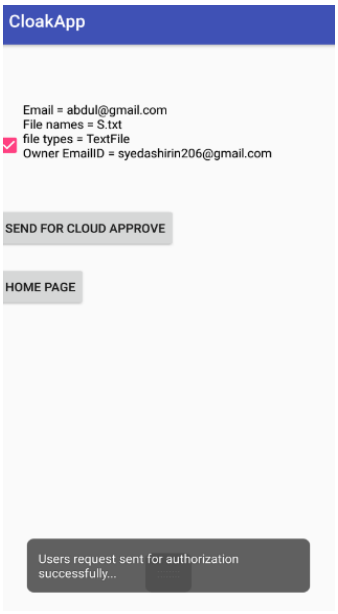

Fig 11 : Authority grants file access

9. Registered Users to the external server

8. User 2 is authorized to decrypt the requested file by requesting the CSPRN key from the external server and for secure key flow it is XORed with the preshared key assigned during registration and is computed back to original key on the mobile device. User 2 can view/download the file.

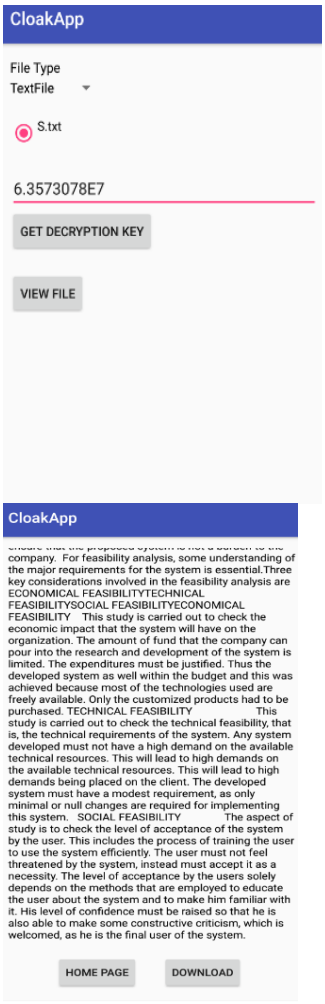

Fig 12 : User 2 decrypt the requested file using the CSPRN key

userid	password	utype
authority@gmail.com	123	authority
cloud@gmail.com	123	cloud
Trustee@gmail.com	123	trustee
abc@gmail.com	123	user
syedashirin206@gmail.com	123	user
abdul@gmail.com	123	user

Fig 13 : Registered users

10. Encrypted Files uploaded by users to the server with there keys and the time taken for encrypting by both block and stream cipher.

file_name	file_type	time_elapsed	email	user_key	stime_elapsed
S.txt	TextFile	5727	syedashirin206@gmail.com	6.3573078E7	5358
Solution .doc	TextFile	10839	syedashirin206@gmail.com	3.2431979E7	10498
rainbow.jpeg	ImageFile	9881	syedashirin206@gmail.com	1.634576+E7	9513
Passwords.txt	TextFile	810	abdul@gmail.com	8.7748945E7	327

Fig 14 : Users uploaded files to the server

11. Time taken for Decryption of requested files

name	type	decr	sdecr
S.txt	TextFile	2312	1964
Solution .doc	TextFile	2066	1595
Passwords.txt	TextFile	2002	1529
rainbowjpeg	ImageFile	2067	1723
Solution .doc	TextFile	2179	1842
S.txt	TextFile	2242	1825
images.jpeg	ImageFile	2090	1604
S.txt	TextFile	2344	2014

Fig 15 : Time taken for decryption of a file

5. CONCLUSION

We exhibited a light-weight, stream figure based encryption/unscrambling convention for the mobile gadgets. The convention is intended for the MCC condition. We handle the difficulties of unreliable remote media by adjusting the CSPRN and securing the message correspondence. We discovered CLOAK can oppose different security challenges like beast drive assault, MIM and Impersonation assaults. Also, we contemplated the security of the messages traded amongst MD and the ES.

REFERENCES

[1] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, ``Heterogeneity in mobile cloud computing: Taxonomy and open challenges," IEEE Commun.Surveys Tuts., vol. 16, no. 1, pp. 369_392, 1st Quart., 2014.

[2] R. Kemp, N. Palmer, T. Kielmann, and H. Bal, ``Cuckoo: A computation of_offloading framework for smartphones," in Mobile Computing, Applications, and Services. Santa Clara, CA, USA: Springer, 2012, pp. 59_79.

[3] Z. Xiao and Y. Xiao, ``Security and privacy in cloud computing," IEEE Commun. Surveys Tuts., vol. 15, no. 2, pp. 843_859, 2nd Quart., 2013.

[4] M. R. Baharon, Q. Shi, and D. Llewellyn-Jones, ``A new lightweight homomorphic encryption scheme for mobile cloud computing," in Proc. IEEE Int. Conf. Comput. Inf. Technol., Ubiquitous Comput.Commun., Dependable, Auton. Secure Comput., Pervasive Intell. Comput. (CIT/IUCC/DASC/PICOM), Oct. 2015, pp. 618_625.

[5] M. Hell, T. Johansson, A. Maximov, and W. Meier, ``A stream cipher proposal: Grain-128," in Proc. IEEE Int. Symp. Inf. Theory, Jul. 2006, pp. 1614_1618.

[6] Y. Chen andW. S. Ku, ``Self-encryption scheme for data security in mobile devices," in Proc. 6th IEEE Consum. Commun. Netw. Conf., Jan. 2009,pp. 1_5.

[7] O. B. Sahoo, D. K. Kole, and H. Rahaman, ``An optimized S-box for advanced encryption standard (AES) design," in Proc. Int. Conf. Adv.Comput. Commun. *(*ICACC*)*, Aug. 2012, pp. 154_157.