A Secure Mobile Cloud Storage Environment using Encryption Algorithm

Nitin Nagar
International Institute of Professional Studies, Devi Ahiliya University, Indore, INDIA

Ugrasen Suman
School of Computer Science and IT, Devi Ahiliya University, Indore, INDIA

ABSTRACT
Cloud computing frameworks such as, Google App Engine, Amazon Web Services, Windows Azure, and open source frameworks such as OpenStack have become increasingly popular among practitioners. Also, the growth in usage and deployment of smartphone platforms and applications worldwide is increasing rapidly. Mobile cloud computing promotes use of cloud based services in a mobile environment. Data and complex computing modules are processed in clouds and mobile devices do not need a powerful configuration such as CPU speed, and memory capacity. Mobile devices are unable to utilize resources, communication delay, and unexpected mobile vulnerabilities or attacks. These challenges have great effect in the improvement of service qualities of mobile cloud. In this paper, the survey of different vulnerability and attacks on mobile cloud computing identified and also design a secure mobile cloud storage environment through encryption algorithm. The proposed work focuses on solution for the threats that are the major issues for MCC adoption.

General Terms
Security, smartphone, and encryption.

Keywords
Mobile computing, static partitioning, dynamic partitioning, first factor authentication, M-pin, and TPA.

1. INTRODUCTION
In the past few years, the smartphone has come out as a new computing paradigm that offers a broad scope of applications, powerful operating systems and multifunctional sensors. The smartphone support for multifaceted applications from diverse arenas, such as teaching, entertainment, commercial enterprise, and health care has contributed much to its popularity. As a consequence, the smartphone is becoming a most important computing platform for mobile users [1] [2]. The increasing features and complexity of smartphone applications simultaneously increase with demand of computational ability and energy [3]. Manufacturers release new and enhanced authentic base mobile device models to fulfill the demands of user. However, the mobile devices have size constrained, due to which the improvements in smartphone hardware are unable to cater the users and applications demands [4]. Numerous applications that need high resources are inappropriate for smartphone due to their limited memory, processing power, battery lifetime, and repositioning.

Cloud computing is extensively employed in several areas. It is clear that cloud computing is an efficient solution to cater the smartphone's computation, memory, and energy demands [5] [6]. The role of cloud computing in mobile devices creates a new domain called mobile cloud computing (MCC). MCC builds the mobile devices, energy efficient and resourceful by allowing applications to offload resource intensive computational tasks with uses of cloud computational power. Different applications based on MCC have been acquired and served to users through navigation systems for mobile, voice search, Google's maps, Gmail and various applications on an android platform, MotoBlur from Motorola, and MobileMe from Apple.

MCC threats and vulnerabilities are a foremost challenge in the field of research. The rest of the paper is arranged as follows. Section 2 includes an MCC security challenges and the associated issues in a cloud environment. The section 3 states architecture of mobile cloud computing along with secure proposed framework which includes mobile network, first factor and second factor through M-pin authentication server. In section 4, we include the proposed framework for MCC along with encryption and decryption of data before storing it onto MDSP. In Section 5, we illustrate the work with one example. We also discuss the implementation work to solve the problems different vulnerabilities and attack. The section 6, we have the experiment of proposed work with security in mobile storage evaluation discussion and future work of the proposed work in the mobile cloud computing. In section 7, we state the conclusion and final, references of the paper.

2. MCC SECURITY CHALLENGES AND ISSUES
Mobile cloud computing has recently drawn substantial attention from both academia and industry. Agreeing to a recent study of heavy reading, the direct revenue of the mobile cloud market will rise to nearly $68 billion by 2017 [7]. It is expected to reach $1 trillion in the broader mobile cloud market. A similar forecast has also been made by ABI research, which foreshadows that the number of mobile cloud computing subscribers (i.e., mobile users and mobile application providers) worldwide is expected to rise quickly over the following five years, growing from 42.8 million subscribers in 2008, (approximately 1.1% of all mobile subscribers) to over 998 million in 2014 (about 19 %) [8]. ABI further forecasts that mobile cloud will soon turn a disruptive force in the mobile world, finally becomes the leading area on which mobile applications operate [8].

The general issues of MCC are data theft risk, privacy of data, violation of privacy rights, loss of physical security, handling of encryption and decryption keys, security, and auditing of virtual machines. The vital concern in MCC is data life cycle, which needs to be standardized to motivate the users to adopt mobile cloud data services such as, generation of data, transfer of data, accessing of data, and computer memory. In addition to the data security threats on mobile cloud side, there are some attacks, which are probable at end user smartphone such as, device data theft, virus and malware attacks via wireless devices, and misuse of access rights.

Details of the mobile cloud computing vulnerabilities, attacks, risks and their protection solution are shown the Table 1. Apart from data and information security, the mobile cloud computing have some general issues in terms of their architecture such as, computing off-loading, security for mobile data, improvements in efficiency rate of data access, context aware mobile cloud services, cost and pricing, migration and interoperability and service level agreement (SLA).

In cloud infrastructure, a variety of attacks are identified along with the mobile cloud, which include attacks such as, attacks on virtual machines, vulnerabilities at platform levels, phishing, attack on authorization and authentication level, attacks from local users, and hybrid cloud security management. Some other major challenges and issues have serious in adoption of MCC such as partitioning, execution delay, and communication, which are discussed in subsequent subsections.

2.1 Partitioning Issues

MCC partitioning is based on the concept in which application split into separate components, and preserve the semantics of original application. Partitioning in the mobile based application is difficult to perform. The Identification of component for secure migration in mobile cloud is also difficult in partitioning. There are two ways to perform partitioning in mobile cloud computing environment such as static and dynamic partitioning, the detail description are as follows:

2.1.1 Static Partitioning

In static partitioning, application is separated into a constant number of modules and does not handle the elasticity property of the cloud based resources. MISCO is a static profiling approach for a distributed information processing framework, where high computation applications are partitioned into various tasks [13]. The key value pairs from the input information is obtained using map function and then those pairs are grouped into R partitions using the partitioning function. All pairs in same partition are given in a reduce function for final partitioning results.

2.1.2 Dynamic Partitioning

Dynamic partitioning is depends upon the network bandwidth and connectivity status. Several research works have been performed previously on partitioning off-loading. CloneCloud is the framework used to solve problems such as, energy consumption and execution time through the optimal partitioning of the applications between mobile devices and commercial clouds [9]. Mobile assistance using infrastructure (MAUI) is a system that enables fine-grained energy-aware offloading on mobile code with minimal burden on programmers [10]. The comparison of various existing partitioning strategies based work is shown in Table 2.

Table 1: MCC Vulnerabilities, Attacks, Risks and Protection Solution

Vulnerability	Threat	Risk	Protection Solutions
Over-The-Air (OTA) transmission between a smartphone and cloud environment	Interception of traffic	Identity theft, information disclosure, replay attacks	Trusted platform module (TPM), secure protocols, encryption
Changing or replacing a mobile phone	Configuration and setup complexity	Reduced adoption of the technology	Simplified user interface, security parameters in TPM set by a trusted party
smartphone Internet and geo-location capabilities	Malware on mobile devices, poor data protection controls by merchants	Data disclosure and privacy infringement, profiling of user behavior	User control of geo-location features, cryptographically supported privacy, TPM, authorization and accounting
Weak cryptographic keys or predictable random-number cryptographic APIs provided by development platforms	Cryptanalysis and dictionary attacks	Unauthorized access to restricted functionalities	Secure third-party cryptography APIs with algorithms that can generate unpredictable random numbers and strong cryptographic keys

2.2 Execution Delay

The execution time delay in mobile computing is high as compared to cloud computing due to aspects such as, network and bandwidth. A computation intensive module is being performed in the cloud environment while the mobile user is waiting for the task to be completed and output to be received. Suppose any problem occurs in the processing of the task and the task is delayed for any I/O event to occur or time-out. The mobile user is unaware of the processing progress of the applications running in the mobile cloud [9]. One solution is time checking mechanism through, which the task scheduler in the cloud should keep track of the time spent by the migrated tasks in the cloud. When a module will be migrated to the cloud, a particular amount of time should be set for that module (the time limit for the module to spend in the cloud). The scheduler should keep track of the time spent by the module and compare with the time set for the module (to spend in the cloud). When the time limit exceeds and if the project is still in the cloud, it should be aborted or cancel and send an information to cloud users about the miscarriage.

2.3 Communication

The modules running in the cloud and those running locally on the mobile devices cannot communicate to each other properly, which results the lack of data integrity and data consistency. CloneCloud suggested that the modules which require local resources such as battery life, storage, and bandwidth are not migrated into clouds and only independent modules can be migrated [9]. The upcoming research challenge in MCC is to provide a mechanism, which allows the interaction among the modules of the same application to enhance data integrity and consistency. The modules running in the cloud should be able to access the mobile resources at runtime as well as the data should be transferred between the modules of the same application.

3. ARCHITECTURE VIEW OF MCC

MCC allows access to computing resources in effortless ways which cause a new universe of attack, threats and vulnerabilities. The user name and password is the first necessary authentication process that must be provided before access is granted for the data in any application. Password observed as most weak in current mobile cloud security aspect. A hacker can easily break down the password. Our proposed work provides a solution to the threats that are the major issues for mobile cloud adoption. For this purpose, a secure framework can be designed for the transmission of data and information security in MCC environment. The proposed framework uses two factor authentication mechanisms. The first factor includes policies checker, location tracker, authentication verifier, access point verifier, and partitioning verifier to protect user's data more strongly. Second factor performs M-Pin authentication server; for example, user uses ATM pin for accessibility on data. These mechanisms protect user's data, information from various attacks such as, a key logger attack, malicious insider attack, service hijacking, etc. The proposed framework include four components such as, mobile network, cloud service provider (CSP), third party auditor (TPA) and mobile cloud storage. The general architecture of the MCC is shown in Figure 1.

Table 2: Comparison of various Partitioning Strategies

Paper	Static Partitioning	Dynamic Partitioning	Suitability for Data Stream Application	Consideration of Security Issues	Scalability	Additional Feature
[9]	Yes	Yes	No	No	Lack	Based on traditional computing paradigm
[10]	No	Yes	No	No	Lack	Based on traditional computing paradigm
[11]	No	Yes	Yes	No	Yes	Partitioning is done on cloud side and supports global partitioning
[12]	No	Yes	No	No	Lack	Based on traditional computing paradigm
[13] [14]	No	Yes	No	No	No	Preventive measures are proposed
[14] [15]	No	Yes	No	No	No	Supports encryption and decryption

3.1 Mobile Network

Mobile devices are connected to the mobile networks through access point (e.g., satellite, or base transceiver station (BTS)) that set up and control the connections and functional interfaces between the networks and mobile devices. Mobile user's requests and information (e.g., ID and location) are sent to the central processors that are tied in to servers providing mobile web services. Here, mobile network operators can offer services to mobile users as authentication, authorization, and accounting (AAA) based on the home agent (HA) and user's data stored in databases. AAA describes a framework used for intelligent control access over web resources, implementing policies, and provides necessary data. HA is a router on a mobile node's home network that holds data about

the device's current position through mobile Internet protocol (Mobile IP). It uses tunneling mechanisms to forward Internet traffic so that the device's IP address doesn't have to be changed each time. It also used to connect from a different location. After proper verification of connection now, user's requests are sends to a cloud through the Internet. In a cloud environment, cloud service providers process the different requests of mobile users with the corresponding cloud services.

3.2 Cloud Service Provider

Cloud service provider offers all the services to the users, which are coming from the different mobile locations. CSP also ensure all the policies, which includes at the time of designing the applications. The subcomponents of CSP are first factor authentication, policies checker, location tracker, authentication verifier, access point verifier, and M-Pin authentication server. They are describes as follows:

3.2.1 First Factor Authentication

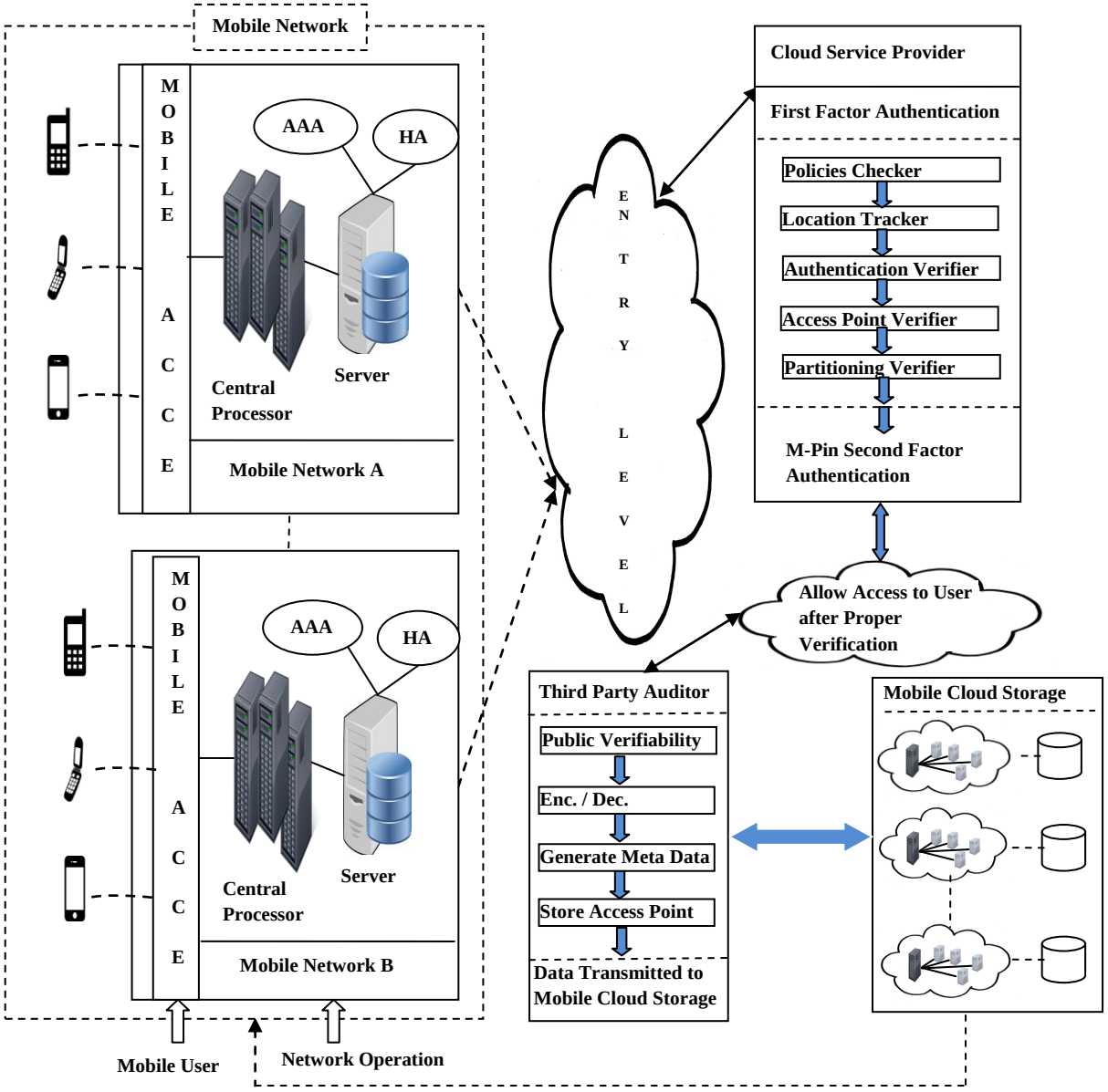

Figure1: Architectural View of Mobile Data Access through Authentication

The proposed work is use to manages user name and password, considered to be more strong first factor authentication that include certain entry level authentication rules such as, policies checker, location tracker, authentication verifier, access point verifier, and partitioning verifier. The following steps are involved in first factor authentication:

3.2.1.1 Policies Checker:

The policies include the IP range, mobile IP, country, proxy and browser. CSP check the policies, which includes at the time of designing the application. They are as follows:

a) Whether the request comes from a particular country or outside country?

b) Whether the application access request comes with proxy or not?

c) Whether the application access request comes from local agent (LA) or global agent (GA)?

d) Whether the application access request comes from desktop or mobile? Different policies glance with succession or not in the log table shown in Figure 2.

3.2.1.2 Location Tracker:

After complete verification of all the policies, location tracker ensures the location detail with ID and location of mobile devices. Location tracker follows all requests through HA and this request will process by GA and LA respectively. LA keeps the status of corresponding VMs. After request assign and update request in VM's memory regarding previous setup. LA gets the resources from GA according to assigned request. GA associated with data center having all information about data such as track of resource availability etc. GA can distribute the resources after getting requests from LA.

```
=>Sun Oct 27 12:27:51 2013, Login Successful! , IP:108.162.222.125, Browser:mobile
=>Sun Oct 27 12:28:03 2013, Login Successful! , IP:108.162.222.125, Browser:mobile
=>Sun Oct 27 13:15:49 2013, Login Successful! , IP:173.245.62.230, Proxy, Browser:desktop
=>Sun Oct 27 13:16:12 2013, Login Unsuccessful! , IP:173.245.62.230, Browser:desktop
=>Sun Oct 27 13:16:19 2013, Login Successful! , IP:173.245.62.230, Browser:desktop
=>Mon Oct 28 08:52:56 2013, Login Successful! , IP:173.245.62.224, Browser:desktop
=>Mon Oct 28 08:54:10 2013, Login Successful! , IP:173.245.62.224, Browser:desktop
=>Mon Oct 28 08:56:53 2013, Login Successful! , IP:141.101.81.114, Browser:desktop
=>Mon Oct 28 09:00:34 2013, Login Successful! , IP:108.162.215.184, Proxy, Browser:desktop
=>Mon Oct 28 09:04:49 2013, Login Unsuccessful! , IP:173.245.62.224, Browser:mobile
=>Mon Oct 28 09:05:13 2013, Login Successful! , IP:173.245.62.224, Browser:mobile
=>Mon Oct 28 09:45:57 2013, Login Unsuccessful! , IP:173.245.62.242, Browser:desktop
```

Figure 2: Mobile Device Access Log

Location trackers also receive position data or other information such as street, city, states, country and device type from where the request is adding up. There are two ways to use location tracker mechanism, one way is to use map location details directly with Google predefined API to pass over the geographical positioning of the user. Another way to use location details with crypt WebDB data to build the application more reliable, stronger, secure and safe. Figure 3 shows the location signature generator and verification [16].

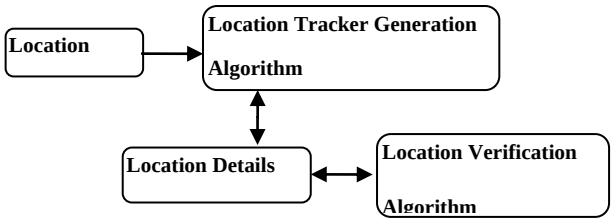

Figure 3: Location Signature Generator and Verification

3.2.1.3 Authentication Verifier:

It confirms all policies and location tracking to ensure the predefined registration details with user name and password with following manner:

- Suppose user A wants to access the mobile cloud storage services.
- The mobile cloud service application is connected with CSP and CSP connected with authentication server (AS), CSP verifies whether the user authorized or not to access the application.
- AS verifies whether the user name and password are correct or not.
  - (i) If it is not correct, the message "Unsuccessful log in" is generated by application.
  - (ii) If it is correct, the message "Successful log in" is generated by the application.

3.2.1.4 Access Point Verifier:

Suppose z is the total allocation volume in GB for cloud data and x, y and V are the initial value, data, and updates respectively. Algorithm 1 can be used to show the access point verification in MCC.

1. If ($V \neq z$ & $\emptyset$)
/* If mobile cloud data server doesn't have any volume */
No restored point is resulted in search
2. Else point is found and restored
3. Then
V = z + (x-y)
4. for V <= 0 to n do
5. V++
6. return V
/* recursively call for each update */

Algorithm 1: Access Point Verification in MCC

In access point verification algorithm, user initially selects V = (v_i, v_{i+1}, v_{i+2},.....v_{i+n}), for every V updated value ($\forall V$). The initial empty states are denoted by 'a' and total allocation volume from CSP's allocated volume to cloud. The algorithm is initially compared with 'z' and '$\emptyset$'.

3.2.1.5 Partitioning Verifier:

The partitioning verifier used to verify whether the partitioning static or dynamic or both.

3.2.3 M-Pin Second Factor Authentication Server

After first factor authentication, users set out for the second factor authentication, i.e., M-Pin authentication server. The following steps are included in the secure second factor M-pin authentication:

- The user is assisted with the PINPAD, which is a javascript pad. It is similar to the PINPAD used in ATM.
- The user enters his 4 to 9 digit pin to authenticate.
- Finally, user server requests for the time permission to accessing the data.

3.3 Third Party Auditor

Third party auditor can be defined as a server for system or an environment in a mobile cloud model to track file throughout its entire lifecycle. In our proposed TPA server collects all the information about the file for auditing and controlling of data from CSP. The subcomponents of TPA include public verifiability, encryption and decryption, generate metadata, and store access point. After completion of all the subcomponents activity, finally data is transmitted to mobile cloud storage server.

3.4 Mobile Cloud Storage

Mobile cloud storage server provides a storage area in which user's data securely store in different cloud host.

4. SECURE MCC STORAGE

Securing MCC storage is crucial to safely store data from unauthorized access. The user's data is encrypted and then partitioned before sending it into different mobile cloud server's storage area. The same data is decrypted as and when user demands. The partitioning of data builds storing easy and effective. It also gives way for flexible access and cost

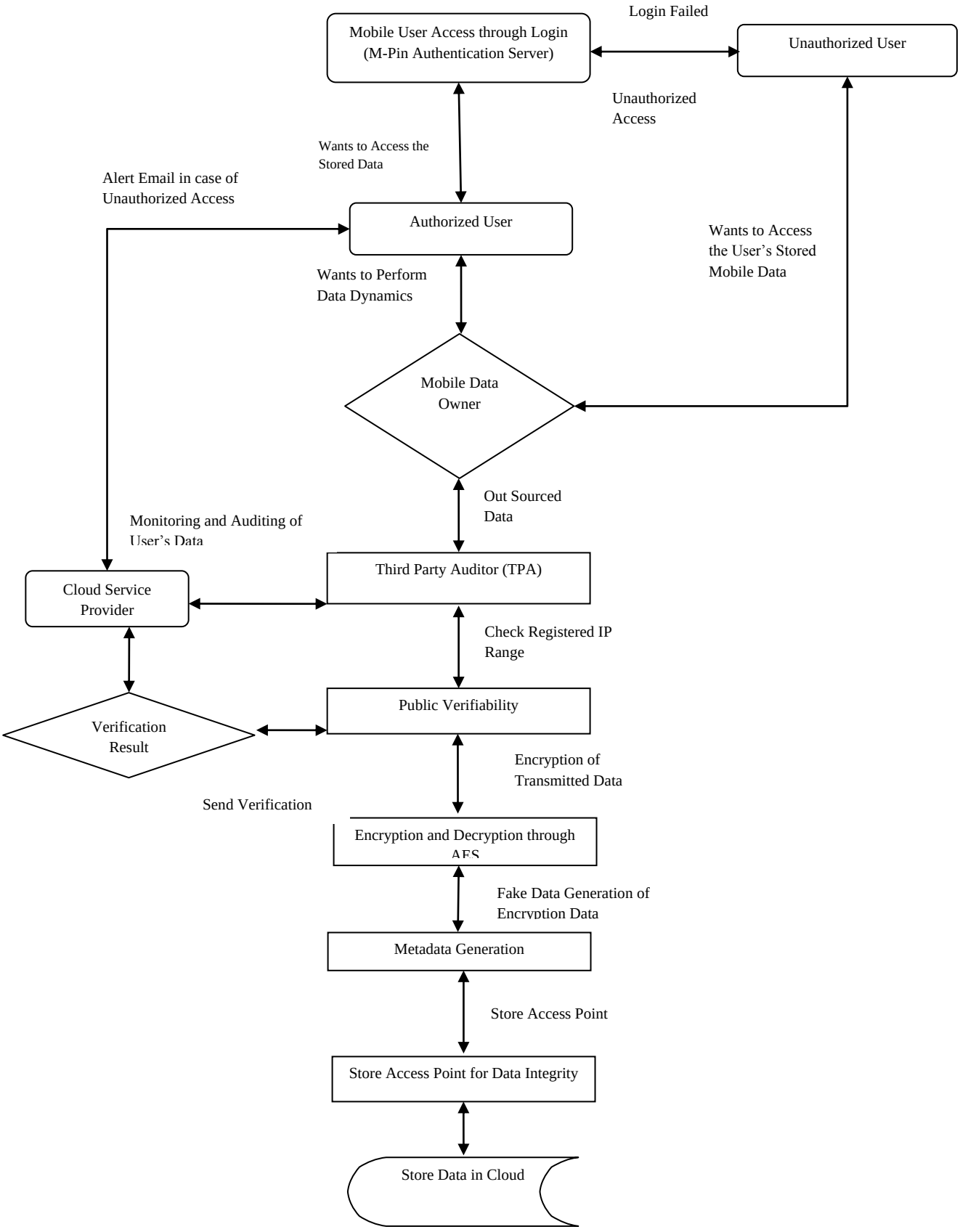

Figure 4: Secure Data Transmission on Mobile Cloud Storage

effectiveness in data storage. The space and time is also effectively reduced during mobile cloud storage. Dynamic operation is another important concept where, encryption and decryption process secures data, when storing into cloud. The remote data integrity checking detects the threats and misbehaving server while storing the data in mobile cloud

ensuring data security. Figure 4 shows the flow of secure data transmission in mobile cloud storage through M-pin authentication server. The proposed framework consists of three parts, which include the mobile user, the CSP and the mobile data storage provider (MDSP), as depicted in Figure 5. The mobile user uses a smartphone to access CSP to perform encryptions, decryptions and integrity checks of their files, before laying in them on MDSP. CSP allows the mobile user to generate public/ private keys, using elliptic curve cryptography (ECC). It also allows the user to encrypt, decrypt and verify the integrity of users files before storing them on MDSP.

Advanced Encryption Standard (AES) is a symmetric block cipher that can encrypt data blocks of 128 bits using symmetric keys 128, 192, or 256. AES uses 10, 12, or 14 rounds. The key size that can be 128,192 or 256 bits depends on the number of rounds. It generates a random AES key to encrypt and decrypt the user's files, while ensuring the integrity of files by using SHA-3 as a hash function. We also use RSA as a public key system designed by Ron Rivest, Adi Shamir, and Leonard Adleman in 1978.

The RSA algorithm is used to encrypt the data to provide security in way that only authorized user can access it. RSA algorithm is only useful to encrypt small amount of data and cannot be utilized for large data or large files [6]. The only disadvantage of using public-key cryptography for encryption is the speed of data transformation. DSP use to stores entire mobile user's account information, files, hash values and AES keys in encrypted configuration. It is also stores the user's public key. These all tasks are performed through TPA.

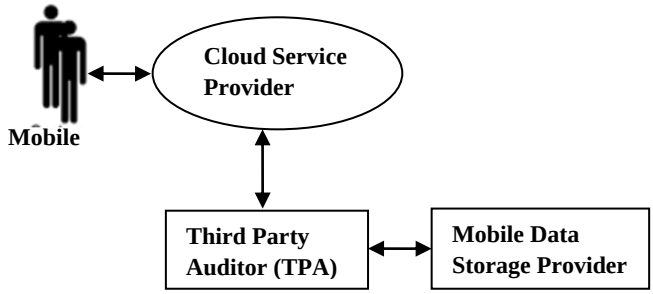

Figure 5: The MDSP Accessibility through TP

4.1 Encryption of Data

Encryption phase allows users to encrypt their files and create the hash value of these files, before storing them on MDSP. The notations used in encryption and decryptions are shown in Table 3. The sample data for encryption and decryption is shown in Table 4. The steps include for the encryptions are:

a. The mobile users login in to CSP to provide the user's authentication [17].

b. CSP retrieves the user's Key (P) from MDSP.

c. The mobile user uploads send data D.

d. CSP allows the mobile user to generate random Key (A).

e. CSP allows the mobile user to compute hash value H of D and encrypt D using Key (A) and encrypt

f. H and Key (A) using the user's Key (P).

Then, CSP stores the encrypted D, H and Key (A) on MDSP.

Table 3: The Notations of Proposed Framework

Notation	Description
D	Mobile user's send data.
Key(A)	Random AES key.
Key(P)	Public Key for user.
Key(R)	Private Key for user.
H	Hash value.

Table 4: Sample Data for Encryption and Decryption

S.No	MDSP_ID	CSP_ID	USER_ ID	Data Format
1	4KMP33	LP6	13BL49	13BL49.docx
2	7DNS45	SK8	12CH59	12CH59.avi
3	8CMQ39	SR5	11MS43	11MS43.doc
4	7KLS41	KM2	65SD83	65SD83.docx
5	8STP25	GM6	63SP48	63SP48.mpeg

4.2 Decryption of Data

Decryption phase allows users to decrypt the required data from MDSP after decrypting and verifying the integrity of these files. Steps require for decryption are as follows:

a. The mobile users logs in to CSP to provide the user's authentication [17]

b. CSP retrieves the user's Key (P) from MDSP.

c. The mobile user selects the required D from provided data list in CSP.

d. CSP retrieves the encrypted D with H and Key (A) from MDSP. CSP allows user to enter Key (R).

e. User decrypts H and Key (A) using Key (R).

f. Then, the user decrypts D using Key (A).

g. User verifies the integrity of D using H.

h. In the last step, user requests data from CSP in plaintext form.

5. ILLUSTRATION

User_ID is used to demonstrate the encryption and decryption process. Table 5 is shows the conversion of User_ID and storage on MDSP.

Table 5: Conversion of User_ID and storage on MDSP

1	13BL49	0011000100110011010000100100110000110 10000111001
2	12CH59	0011000100110010010000110100100000110 10100111001
3	11MS43	0011000100110001010011010101001100110 10000110011
4	65SD83	0011011000110101010100110100010000111 00000110011
5	63SP48	0011011000110011010100110101000000110 10000111000

5.1 Encryption of Mobile Cloud Data

Encryption of mobile cloud data allows encryption of files and creates a hash values for files, before storing them on to mobile cloud storage. Following are the steps are required for storing a data on mobile cloud:

5.1.1: Public key include the pair of keys which include encryption and decryption of message. Here, we obtain a public key as MDSP_ID ((MDSP_ID) = 8CMQ39) which provided through MDSP for public verifiability in mobile cloud storage. We convert the MDSP_ID to its binary equivalents as follows:

{8 = 00111000, C = 01000011, M = 01001101, Q = 01010001, 3 = 00110011, 9 = 00111001}

5.1.2: Secret key is an encryption/ decryption key, known by CSP only in mobile cloud environment. The provided secret key through CSP is CSP_ID which is equivalent to SR5. The binary equivalents of SR5 is as

{S = 01010011, R = 01010010, 5 = 00110101} [0101001101010010 00110101]

5.1.3: User provided data such as User_ID (User_ID = 11MS43) obtained for illustration. Binary equivalents of 11MS43 are as follows:

{1= 00110001, 1 = 00110001, M = 01001101, S = 01010011, 4 = 00110100, 3 = 00110011}

After conversion of SR5 binary equivalent, we concatenate 11MS43 as follow: [00110001 00110001 01001101 01010011 00110100 00110011].

5.1.4: Quotient 'Q' is use for formulation and it is formulated through following formula:

Q = (public key) / (secret key) = (User_ID) / (CSP_ID). After placing all the values according to formula, we get

(00111000010000110100110101010000100110011001110 01) / (010100110101001000110101)

= 10101100110111010110110 1.1100111100110111111011110 1011

Here, we truncated the values right side the decimal part. Now, the new result of Q is

Q = 10101100110111001101101.

5.1.5: We perform ⊕ (XOR) operation with send data (User_ ID) and quotient value (Q => (cipher_text1) = (User_ ID) ⊕ (quotient 'Q'). Binary equivalent result is as follows: 00110001001100010100110101010011001101000011001 1 ⊕ 10101100110111010110110 1

(00 11 00 01 00 11 00 01 01 00 11 01 01 01 00 11 00 11 01 00 00 11 00 11)⊕

(00 00 00 00 00 00 00 00 00 00 00 10 10 11 00 11 01 11 01 01 10 11 01)

We have added zeros 0 to left side of quotient Q for equality of both since it does not affect the operation. Now, we get the new cipher value regarding User_ID.

cipher_text1 = (00 11 00 01 00 11 00 01 01 00 11 01 11 11 11 11 11 10 10 01 01 01 11 10)

5.1.6: The whole cipher_text1 is scanned from left to right for checking the pattern.

00110001001100010100110111111111111010010101111 10 (111 sequences found 5 times)

5.1.7: Cipher_text2 is equivalent Cipher_text1 with bit-stuffing and stuffing using secret key. Since, it is follows 111 keys combination of sequence. We obtain bits from right hand side of secret key for bit stuffing. Secret key bits which are not used have been concatenated at the last.

cipher_text2 = secret key bits = 01010011 01010010 001 10101001100010011000101001101111111011111110010010 1011111001010011 01010010 001

It is second level of encryption. Thus, we get new encrypted form of user_ Id. Similarly, we encrypted all the information of the user and stored them into MDSP.

5.2 Decryption of Mobile Cloud Data

Decryption of mobile cloud data allows decryption of files, when user demands for the mobile cloud stored data. Decryption is the reverse of encryption. We obtained the same input's binary equivalent (cipher_text2 = 00110001001100010100110111111101111111001001) for decryption.

5.2.1: Remove all concatenated bits from cipher_text 2 00110001001100010100110111111101111111001001010111 11001 and we obtained new remove bits 01010011 01010010 001.

5.2.2: Remove each fourth bit followed by the occurrence of triple one (111) sequence. And append these bits at the end of previously removed bits.

5.2.3: If these removed bits are equal and same as secret key the left sequence will become cipher_text1 as original plaintext.

6. EXPERIMENTAL RESULTS

The proposed work is evaluated using various parameters such as, the size of data, computational time, power consumption, mean processing time and throughput. The equipment used to obtain the measurements are an Intel (R) Pentium (R) Core 2 Duo CPU 2.4 GHz (VT enables machine), 8 GB of RAM, and Ubuntu 14.04 (LTS) runs on Linux. To enhance the security of system, we use JAVA language for development of algorithms. We considered availability, integrity, security, authentication, and authorization in network I/O, and network connection for the measurement of performance. The ten text files of different sizes are applied to demonstrate the experimental results. The encryption time is considered the time that an encryption algorithm obtained to produce a cipher text from a plain text. Encryption time is used to calculate the throughput of an encryption scheme, which can be calculated as the total plaintext in bytes encrypted divided by the total encryption time obtained by algorithm. The quick response from encryption algorithm also indicates the more security in storage. The details of various parameters are as follow:

Size of data: Different data sizes are needed for different memory space to perform the operation. The memory space needed by any algorithm is fixed along the basis of input data size, number of rounds, etc.

Computation Time: The time required by algorithm to complete the operation depends on processor speed and algorithm complexity.

Power Consumption: The cycles such as CPU clock cycle and the average current, use for each cycle can easily estimate the entire energy use of cryptographic functions.

Mean Time of Processing: Mean processing time is the difference between the starting and ending times in use of encryption mechanism.

$$Mean\ Time\ of\ Processing = End\ Time\ to\ Encrpt - Start\ Time\ to\ Encrypt$$

Throughput: Throughput of the encryption algorithms is calculated by dividing the total plain text in bytes and total encryption time taken by algorithm. The throughput of the encryption scheme is calculated as, $Throuput = \frac{t_p(byte)}{e_t(second)}$ where t_p is total plaintext (byte) and e_t is the total time for the encryption. The Table 7 shows the comparative studies of existing work and proposed work [18].

Mean processing time is considered as a key factor in mobile cloud computing storage environments to identify the attack status. Usually, it's measured in KB. Figure 6 states the different mean processing time of existing and proposed work. It also required for the measurement the reliability and availability of algorithm.

6.1 Mobile Cloud Storage Security Evaluation Discussion

Mobile security testing is a process that determines the system is protects the mobile cloud storage data and also maintains the integrity of the system. Security is the probability that an attack of a certain type will be removed over a period of time. The integrity is formalized as Integrity = ∑ [(1 - threat) X (1 - security)]. Integrity measures the system's ability to withstand attacks to its security.

In order to measure integrity two additional parameters threat and security are needed. Threat is the probability that an attack of a certain type will happen over a period of time. Where, threat and security are summed over each category of attacks. Figure 7 shows the organization of data security in mobile cloud storage.

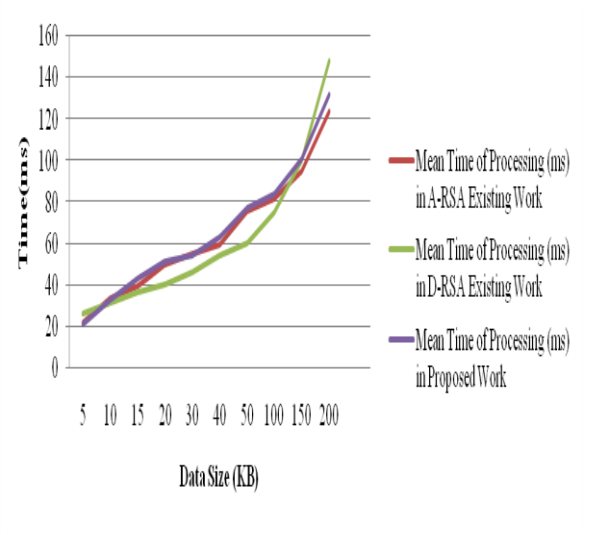

Table 6: Experimental Result with Various Data Size (KB)

Data Size (KB)	Time to Encrypt Plaintext (ms)	Time to Decrypt Plaintext (ms)	Throughput of Encryption (KB/ms)	Throughput of Decryption (KB/ms)	Power Consumption	Mean Processing Time (ms)
05	23	07	0.21	0.71	4.76	23
10	35	11	0.28	0.91	3.57	35
15	43	16	0.34	0.93	2.94	43
20	51	22	0.39	0.91	2.56	51
30	56	29	0.53	1.03	1.88	56
40	63	42	0.63	0.95	1.58	63
50	77	53	0.64	0.94	1.56	77
100	84	67	1.20	1.49	0.83	84
150	101	83	1.49	1.48	0.67	101
200	132	103	1.52	1.95	0.65	132

Table 7: Comparative Studies with A-RSA and D-RSA with Proposed Work

Data Size in (KB)	Power Consumption in Exiting Work	Mean Time of Processing (ms) in A-RSA Existing Work	Power Consumption in Exiting Work	Mean Time of Processing (ms) in D-RSA Existing Work	Power Consumption in Proposed Work	Mean Time of Processing (ms) in Proposed Work
05	4.76	22	3.95	26	4.76	21
10	3.45	34	3.13	31	3.57	33
15	2.5	40	1.90	36	2.94	43
20	2.45	50	2.00	40	2.56	51
30	1.82	55	1.54	46	1.88	54
40	1.52	60	1.35	54	1.58	63
50	1.65	76	1.20	60	1.56	77
100	0.82	82	0.75	75	0.83	84
150	0.63	95	0.67	100	0.67	101
200	0.65	124	0.74	148	0.65	132

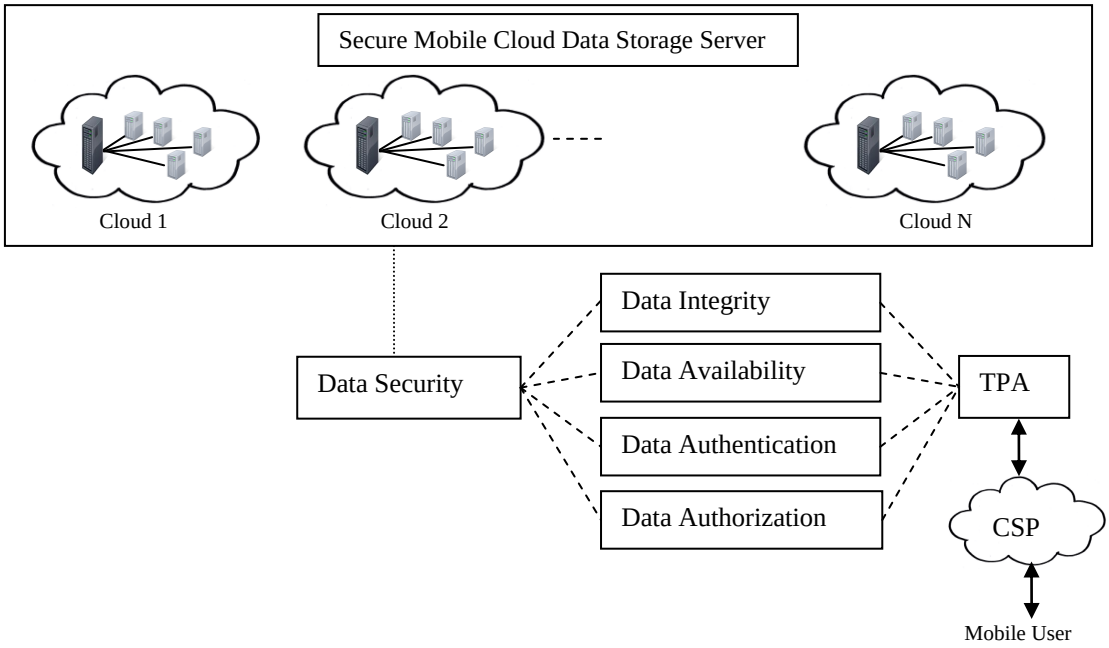

Figure 7: The Organization of Data Security in Mobile Cloud Storage

The existing work fully focuses on security issue and provides some mechanisms to achieve security in MCC through encryption and decryption [14] [15]. Table 8 shows the reliability evaluation for mobile cloud storage server.

We also include four basic security concepts that need to be covered in security testing such as integrity, authentication, availability and authorization. The results show that reliability in MCC cannot be achieved completely, but up to some extent the threats to security can be controlled using security measures.

Table 8: Reliability Evaluation for Mobile Cloud Storage Server

Mobile Cloud Storage Server Access	CS_MTBF (min/year)	CS_MTTR (min/year)	Availability (%)	Unavail ability (%)
Access Stored Mobile Data	78900	78902	99.99747	0.00253
Encryption	78967	78969	99.99747	0.00253
Decryption	77778	77780	99.99743	0.00257
Store Mobile Access Point	34566	34568	99.99421	0.00579

7. CONCLUSION

MCC is very fast and growing technology for the last few years. MCC environment cannot be visualized without cloud based storage. Cloud based storage is also very popular and well known technology since last few decades. MCC creates a new dimension to organization to think forward apart from traditional approaches because they are less effective and costly. But with these new features, MCC also creates new security challenges for the organizations. The smartphone components and environment cannot be protected by existing security mechanisms.

Additional considerations and protections must be kept in place to ensure a strong security mechanism, planning and preparation, as well as training needs to be implemented in advance. Encryption algorithms are used to protect the confidentiality and integrity of user's data at time of uploading or storing data in mobile storage cloud. Encryption algorithm plays an important role in communication security whereas data size, computation time, Throughput, power consumption and mean of processing time are the major issues of concern. Our proposed encryption and decryption of mobile based data before storing it into cloud will ensure the more secure against attack mitigation. The proposed mechanism use to prevent mobile cloud environment from attack will not ensure that the data will more secure although, it gives a new direction and dimensions of security levels in mobile cloud computing. We have many more algorithms to be evaluated and their results can be analyzed with one another to produce the best implemented security algorithm in mobile cloud environment for the future use.

8. REFERENCES

[1] More smartphones Were Shipped in Q1 2013 Than Feature Phones, An Industry First According to IDC, http://www. idc.com/getdoc.jspcontainer Id$=$prUS24085413.

[2] Vallina-Rodriguez, N., Crowcroft, J, "Energy management techniques in modern mobile handsets", IEEE Communications Surveys & Tutorials, 2013, vol.15, issue1, pp 179–198.

[3] Developer Works survey, http://public.dhe.ibm.com/software/dw/survey/2010surveyresults/2010surveresults-pdf.pdf.

[4] Khan, A.R., Othman, M., Madani, S.A., Khan, "S.U.: A Survey of Mobile Cloud Computing Application Models, IEEE Communications Surveys & Tutorials", 2014 vol.16, issue1, pp 393–413.

[5] Ferzli, R., Khalife, I., "Mobile cloud computing educational tool for image/video processing algorithms", *In IEEE Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop*, pp. 529-533.

[6] Zhao,W., Sun, Y., Dai, L., "Improving computer basis teaching through mobile communication and cloud

computing technology", *In International Conference on Advanced Computer Theory and Engineering (ICACTE)*, pp. 452- 454.

[7] Heavy Reading Real World Research (2013) the mobile cloud market outlook to 2017.

[8] ABI (2009) Mobile cloud computing subscribers to total nearly one billion by 2014, Tech. Rep., ABI Research.

[9] B.Chun, S.Ihm, P.Maniatis, M.Naik, and A.Patti., "Clone Cloud: Elastic Execution between Mobile Device and Cloud", *Sixth Conference on Computer Systems*, ACM New York, USA, Apr. 2011, pp 301-314.

[10] E.Cuervo, A.Balasubramanian, D.Cho, A.Wolman, R. Chandra, and P.Bahl, "MAUI: Making smartphones Last Longer with Code Offload", *8th International Conference on Mobile Systems, Applications, and Services*", USA, 2010, pp 49-62.

[11] L.Yang, J.Cao, S.Tang, T.Li, and A.T.S. Chan., "A Framework for Partitioning and Execution of Data Stream Applications in Mobile Cloud Computing", *IEEE Fifth Conference on Cloud Computing*, Honolulu", 2012, pp 794-802.

[12] X.Zhang, A.Kunjithapatham, S.Jeong, and Si.Gibbs, "Towards an Elastic Application Model for Augmenting the Computing Capabilities of Mobile Devices with Cloud Computing", *Mobile Networks and Applications*, 2011, pp 270-284

[13] A. Duo, V. Kalogeraki, D. Gunopulos, T. Mielikainen, and V. Tuulos. Misco, "A mapreduce framework for mobile systems", *Third International Conference on Pervasive Technologies Related to Assistive Environments*, ACM, 2010.

[14] Lipika Goel and Vivek Jain, "A Review on Security Issues and Challenges of Mobile Cloud Computing and Preventive Measures", *International Conference on Advances in Computer Engineering and Applications ICACEA (5)*, March, 2014, pp 22-27.

[15] A. N. Bahar, M. A. Habib and M. Manowarul Islam. "Security Architecture for Mobile Cloud Computing", *Scientific Knowledge*, vol 3, issue3, 2013.

[16] Nitin Nagar, U. Suman, " A Secure Cloud Environment through Location Signature and HTML5 WebDB," Proc. of 3rd International Conference on Advances in Cloud Computing (ACC -2014), Pune (MS), India, Oct. 10, 2014pp.31-35.

[17] Nagar, Nitin, and Ugrasen Suman. "Two Factor Authentication using M-pin Server for Secure Cloud Computing Environment." *International Journal of Cloud Applications and Computing (IJCAC)* vol.4, issue 4, 2014, pp 42-54.