International Journal of Contemporary Research in Computer Science and Technology (IJCRCST) e-ISSN: 2395-5325
Volume 8, Issue 1 (April '2025)

SECURITY AND TIME COMPLEXITY ANALYSIS IN CLOUD COMPUTING USING ASYMMETRIC CRYPTOGRAPHY ALGORITHMS

E.Jansirani,
Research Scholar,
PG and Research Department of Computer Science,
Sri Vijay Vidyalaya College of Arts & Science(Affiliated to Periyar University),
Dharmapuri, Tamilnadu, India.

Dr.N.Kowsalya,
Assistant Professor,
PG and Research Department of Computer Science,
Sri Vijay Vidyalaya College of
Arts & Science (Affiliated to Periyar University),
Dharmapuri, Tamilnadu, India.

Abstract: Now a day, Security is becoming a main concern to maintain confidentiality and integrity of the data. For this purpose, cryptography techniques are used. Cryptography is used to encrypt the data into non readable format and decrypt it again into readable format when needed using specific key. Also for business perspective, cloud computing is very useful. Cloud Computing has become one of the most talked about technologies in recent times and has got lots of attention from media as well as analysts because of the opportunities it is offering. It has different meaning to different uses according to their need. Cloud computing provides an illusion to the customers of using infinite computing resources that are available from anywhere, any time on demand. Cloud computing provide secure data transmission. Security of data becomes a large concern to insure various attribute like integrity, confidentiality, authentication etc., Asymmetric cryptography techniques like DSA,RSA, Elgammal and ECC plays a major roles in protecting the data in those application which are running in a network environment. In this paper providing comparative analysis on various security algorithm which are already available.

Keywords: ***Cryptography, Cloud Computing, Dsa, Rsa, Elgammal, Elliptic curve cryptography, Encryption and Decryption, Asymmetric cryptography***

I.INTRODUCTION

Cloud computing provides a flexible and convenient way for data sharing, offering broad social Advantages as well as individuals. However, there is a natural reluctance for consumers, since data often provide valuable information, to outsource the shared data directly to the cloud server. Cryptographically improved control of access on the shared data should also be enforced. This involves a mechanism that could remove the user from the software until the permission to any user expires. The enhancement of administration, for example, portable learning, human resources control and web-based business, have improved multi-faceted processing in ebb and flow science. Millions of people consistently provide the knowledge and accomplishment that builds their customers. For multipath communication and the transmission of information, the preferred use of distributed computing for the simple multipath correspondence is feasible. Due to the increased number of customers and the strain, the innovation increases more security breaks. Only if penetrations are minimized must the system be considered as steady and secure. Since applications need increased reliability, less pressures and heart-feltness, the correspondence must start to end much better [1].

1.1CLOUD CHARACTERISTICS

- On demand service: cloud is large resource and service pool that you can get service or resource whenever you need by paying amount that you used.
- Ubiquitous network access: cloud provides services everywhere though standard terminal like mobile phones, laptops and personal digital assistants.
- Easy use: the most cloud provider's offers internet based interfaces which are simpler than application program interfaces so user can easily use cloud services.
- Business model: cloud is a business model because it is pay per use of service or resource.
- Location independent resource poling: the providers computing resources are pooled to serve multiple customers using multitenant model with different physical and virtual resources dynamically assigned and reassigned according to demand.

Virtualization is built on top in cloud computing. It is the important concept in cloud computing. First virtualization means to make an implicit version of a resource, and the framework divides the resource into one or more execution environments. If virtualization has securities issues then there also security issues in cloud computing. In cloud computing there are confidentiality, integrity and availability are three major aspects in cloud computing. And also to maintain security on these aspects is a challenging issue.

The process of cryptography Figure 1 displays the process the plaintext passed through before turning into cipher text and then back into plaintext. The plaintext passes through the encryption process to produce a cipher text, while the cipher text passes through the decryption process to produce the plaintext [2].

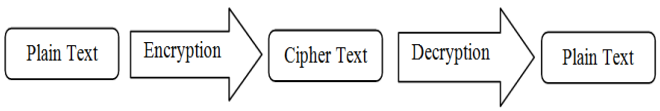


Figure 1: Encryption and Decryption Process

The main purpose of cryptography is to secure data or information from cyber crimes. Cryptography has numbers of security features that's why is it is widely used today. Following are some goals of cryptography

i. **Authentication:** This is a process of proving identity. In this we verify the message security. Authentication is of two types Peer entity authentication and Data origin authentication.

ii. **Privacy:** Privacy means protection against unauthorized manifestation of information. It may be applied to whole message. Privacy provides the conservancy of transmitted data from dormant attacks. iii. Integrity: It assures the receiver that the received original message which has not been exchanged. iv. Non-repudiation: Sender or receiver cannot deny for a transmitted message. When a message is sent, the receiver can verify that the sender in fact sent the message.

One of the main reasons why asymmetric cryptography was invented is because symmetric cryptography is not suitable for communication in a big network with a large number of users. There is a key distribution problem. Each user has to have/remember the secret key of all other users, with whom he communicates.

II. ASYMMETRIC CRYPTOGRAPHY

In an asymmetric, the encryption and decryption keys are different but related. The encryption key is known as the public key and the decryption key is known as the private key. The public and private keys are known as a key pair. Asymmetric key is double unique keys are utilized. The public key is accessible to anybody on the network. The public key gets utilized to encode data. The only private key can decode that data. The private key is reserved secret and it needed in keeping information secure. The pros of utilizing asymmetric key encryption are that it gives better scalability and distribution of key relative to symmetric systems. A few standard Asymmetric Key Algorithms are El Gamal, ECC, RSA, and DSA. The asymmetric encoding algorithm needs more computational processing power if we evaluate it compare to symmetric encoding algorithm. Symmetric approximately 1000 times faster compare to Asymmetric techniques [3].

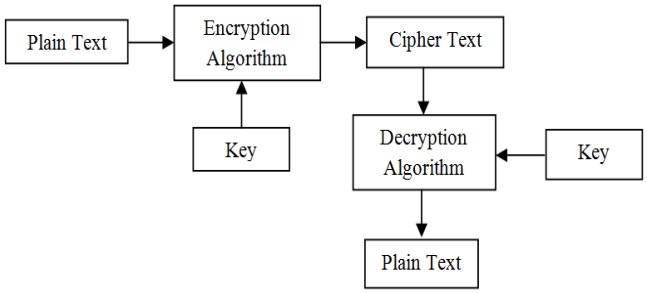


Figure 2: Asymmetric cryptography

Information security is the process of securing the private information in communication channel. Securing data is a challenging issue in today's era. With the incredible growth of sensitive information on cloud, cloud security is getting more important than even before. The cloud data and services can be accessed anywhere. With the rapid growth of the cloud users disastrously happen with a growth in malicious activity in the cloud. Due to those malicious activities, every day, new security advisories are published in order to predict the more and more vulnerabilities which are discovered on the cloud for providing the security over the cloud network. This can be achieved by using the cryptographic algorithms to protect the private information over the cloud. Most of the data travel over the internet and it becomes difficult to make data secure. Securing the data on the cloud is one of the challenging tasks for the researchers. The different cryptographic algorithms are available to protect the data on the cloud. We proposed, in this paper, asymmetric key algorithms are RSA, DSA, Elgammal and ECC on the details of the algorithm are discussed below one of most used algorithm is ECC algorithm. It is simple and computationally faster than the other algorithms.

III. SECURITY ALGORITHMS

3.1 DSA (Digital Signature Algorithm)

Digital signatures are very essential in modern world to verify the sender's identity. Digital signature is an electronic signature which is used for verification and authentication of data. A digital signature is represented as a string of binary digits in computer system. The signature is using a set of rules and parameters (algorithm) such that the identity of the person signing the document as well as the originality of the data can be verified. The signature is generated with the help of a private key. A private key is known only to the sender. The signature is verified by receiver by use of a public key which corresponds to the private key.

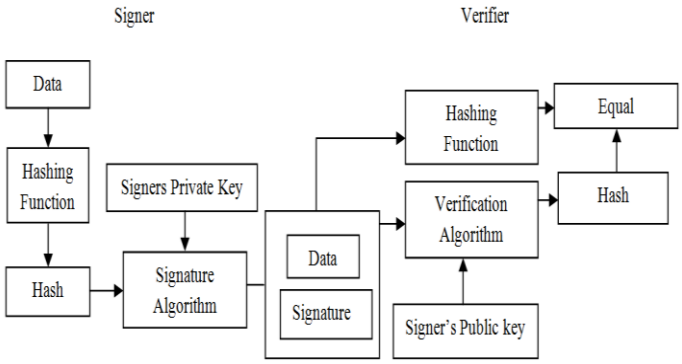


Figure 3: Model of Digital Signature

Digital signature can be used with any kind of data whether it is encrypted or not. Digital signatures are used to detect unauthorized modifications of data by third party. Also, the recipients of a digitally signed document assure that the document was indeed signed by the person who it is claimed to be signed by. This is known as non repudiation, because the person who signed the document cannot repudiate the signature later. Digital signature algorithms can be used in e-mails, electronic funds transfer, software distribution, data storage that assure the integrity , authenticity and originality of data. A hash function is used in the signature generation process to obtain a condensed version of data, called a message digest. The message digest is then input to the

digital signature algorithm to generate the digital signature [4].

3.2 DSA algorithm:

DSA is a variant of the Schnorr and ElGamal signature algorithms, and is fully described in [2]. The algorithm uses the following parameters:

p = a prime number L bits long, when L ranges from 512 to 1024 and is a multiple of 64. (In the original standard, the size of p was fixed at 512 bits [1]. This was the source of much criticism and was changed by NIST [5].)

q = a 160-bit prime factor of $p - 1$.

$g = h(p - 1)/q \bmod p$, where h is any number less than $p - 1$ such that $h(p - 1)/q \bmod p$ is greater than 1.

x = a number less than q.

$y = gx \bmod p$.

The first three parameters, p, q, and g, are public and can be common across a network of users. The private key is x; the public key is y.

To sign a message, m:

1) Sender generates a random number, k, less than q.

2) Sender generates

$r = (gk \bmod p) \bmod q$

$s = (k\text{-}1\ (H(m) + xr)) \bmod q$

The parameters r and s are her signature; she sends these to recipient.

3) Recipient verifies the signature by computing

$w = s\text{-}1 \bmod q$

$u1 = (H(m) \,.\, w) \bmod q$

$u2 = (rw) \bmod q$

$v = ((gu1 \,.\, yu2) \bmod p) \bmod q$

If $v = r$, then the signature is verified.

Real-world implementations of DSA can often be speeded up through precomputations. Notice that the value r does not depend on the message. You can create a string of random k values, and then precompute r values for each of them. You can also precompute k-1 for each of those k values. Then, when a message comes along, you can compute s for a given r and k-1. This precomputation speeds up DSA considerably.

Figure3 Shows how the Encryption of message signal is done, the message signal is sent through Hash function to generate a hash code[6]. A Hash function is a mathematical function that converts an input value into a compressed numerical value – a hash or hash value. The length of the output always depends on the hashing algorithm. Then the hash code and random number „k" is given as an input for signature algorithm along with global public key and sender's private key. Then the message signal and signature will be appended to get an encrypted message.

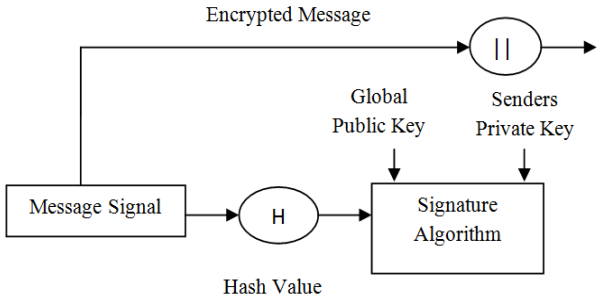


Figure 3: DSA Encryption

Figure 4 shows how the Decryption of message signal is done, once the Encrypted message is received by the receiver; he needs to decrypt the message to get back the original message signal [6].The Encrypted message signal will consist of the original message signal, signature parameters like s and r. The message signal is given to the verifying function, along with it global public key and sender's private key is given to it. And we get the decrypted value that is nothing but the parameter „v".

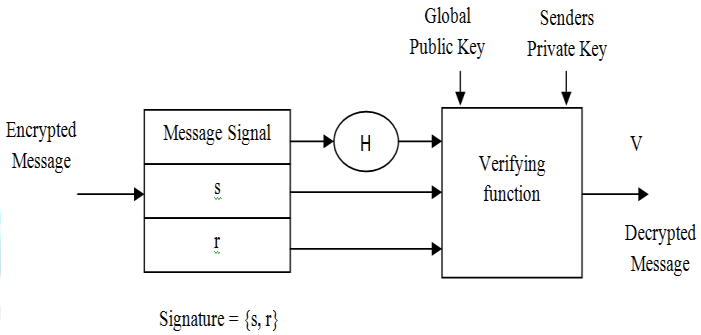


Figure 4: DSA Decryption

The accompanying focuses clarify the whole procedure in detail:

- Each individual receiving this plan has an open private key match.
- Generally, the key sets utilized for encryption/decoding and marking/confirming are distinctive. The private key utilized for marking is alluded to as the mark key and people in general key as the check key [6,7].
- Signer sustains information to the hash work and creates hash of information. • Hash esteem and mark key are then nourished to the mark calculation which delivers the computerized signature on given hash. Mark is annexed to the information and after that both are sent to the verifier.
- Verifier sustains the computerized signature and the confirmation enters into the check calculation. The check calculation gives some an incentive as yield.
- Verifier likewise runs same hash work on got information to produce hash esteem. For check, this hash esteem and yield of confirmation calculation are analyzed. In view

of the correlation result, verifier chooses whether the advanced mark is substantial.

Since advanced mark is made by "private" key of underwriter and nobody else can have this key; the endorser can't renounce marking the information in future. It thought to be seen that as opposed to marking information specifically by marking calculation, more often than not a hash of information is made. Since the hash of information is a remarkable portrayal of information, it is adequate to sign the hash set up of information. The most essential reason of utilizing hash rather than information specifically to sign is productivity of the plan. Give us a chance to accept RSA is utilized as the marking calculation. As examined out in the open key encryption section, the encryption/marking process utilizing RSA includes particular exponentiation. Marking extensive information through secluded exponentiation is computationally costly and tedious. The hash of the information is a moderately little process of the information, subsequently marking a hash is more productive than marking the whole information.

3.3 RSA (Rivest–Shamir–Adleman)

The most common Public Key algorithm is RSA, named for its inventors Rivest, Shamir, and Adleman (RSA). RSA is basically an asymmetric encryption and decryption algorithm. It is asymmetric in the sense, that here public key distributed to all through which one can encrypt the message and private key which is used for decryption is kept secret and is not shared to everyone. which is based on prime numbers. This algorithm works on a public and private key system, the public key is available to everyone to encrypt the information and the person who have private key can able to decrypt the original information.

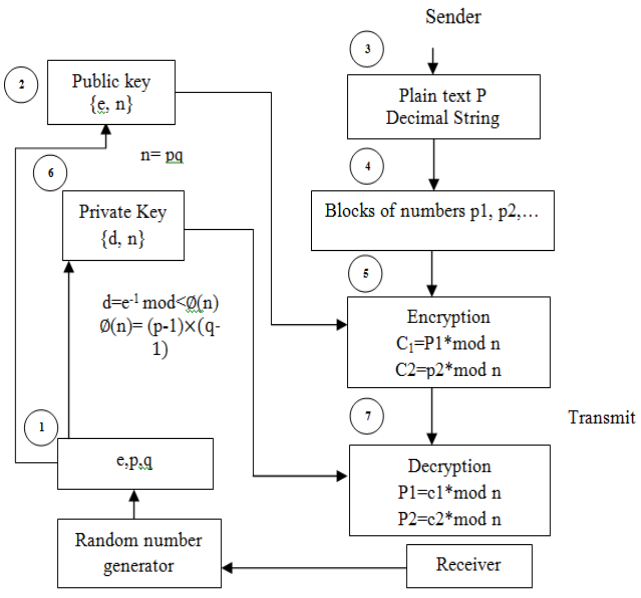


Figure 5: RSA Cryptosystem

The fundamental of RSA is the factorization product of two large prime numbers p and q and the block size depending on the product of these to a prime number. Those keys generated using modulus and exponent operations. The following diagram presents the process of generating the key, encryption, and decryption [8, 9].

First RSA cryptosystem as present in figure 5 begins with Key generation though using two substantial prime numbers, p, and q which be about a similar size in bits. Next, calculate n via the production of the prime number n = pq and calculate Phi of n as $\Phi(n)=(p-1) \times (q-1)$. Then, choose e (public key) which is moderately prime and should stratify this equation were $(1<e< \Phi(n)$ & gcd $(\Phi(n), e) =1)$. After that, calculate d which will represent the value of a private key, using one of a kind multiplicative backward of e modulo $\Phi(n)$ which is $ed=1 \bmod \Phi(n)$. The output is n which will define the size the message M to be encrypted where M<n, public key {e, n} and private key {d, n}. For encryption, the sender will use the public key e and encipher the massage using this formula C=Me mod n. While the receiver uses d private key to decrypt the M using M=Ce mod n. Keep in mind choosing the prime number should be small that will case the factorization of n will be an easy break. Don't use p and q that are relatively close as finding out the common factors reveals the public key [10, 11].

How RSA is going to work in cloud environment is explained as: RSA algorithm is used to ensure the security of data in cloud computing. In RSA algorithm we have encrypted our data to provide security. The purpose of securing data is that only concerned and authorized users can access it. After encryption data is stored in the cloud. So that when it is required then a request can be placed to cloud provider. Cloud provider authenticates the user and delivers the data to user. As RSA is a Block Cipher in which every message is mapped to an integer. In the proposed cloud environment, Public key is known to all, whereas Private Key known only to user who originally owns the data. Thus encryption is done by the cloud service provider and decryption is done by the cloud user or consumer. Once the data is encrypted with the Public key, it will be decrypted using the corresponding Private Key only. RSA algorithm involves three steps:

1. Key Generation
2. Encryption
3. Decryption

RSA algorithm:

Key generation:

Input:Two large distinct primes p and q of the same bits long

Compute N=pq and $\emptyset(N) = (p-1)(q-1)$

Pick an integer e that is relatively prime to $\emptyset(N)$. then determine the integer d satisfying $ed \equiv 1 (\text{mod} \emptyset(N))$.

Output: (ek , pk)

The evaluation key is ek=(N);the private key is pk = (N,e,d).

Encryption

Input:Given pk, let m1,…….,mk ∈ z_w be plaintexts

Compute the ciphertexts c_i as $c_i \equiv m_i^e \pmod N$ where 1≤i≤k

Output: c_i =E(p_k,m_i)

Evaluation

Input: Given ek,let c1,…….,ck ∈ Z_n be ciphertexts

Compute C≡c1×…..×ck(mod N)

Output: C as the result

Decryption

Input: Given pk and a ciphertext C

Compute $c^d \pmod N \equiv m_2 \times \ldots\ldots \times m_k$

Output: $m_1 \times \ldots\ldots\ldots \times mk = D(pk, c)$

Step 1: Generate two distinct primes *p* and *q* each [*n*/*r*+*s*] bits long. Then compute *N* = *prqs* and *L* = *lcm*(*p*−1, *q*−1).

Step 2: Pick an integer *e* that is relatively prime to *L* and to *N*. Then compute *d* ≡ *e*−1 (mod *L*). Step 3: Compute *dp* ≡ *d* (mod *p*) and *dq* ≡ *d* (mod *q*) Step 4: Compute *kp* = *Npr* (*Npr*)−1 and *kq* = *Nqs* (*Nqs*)−1 where *kp* ≡ 1 (mod *pr*) and *kq* ≡ 1 (mod *qs*) The evaluation key is *ek* = (*N*); the private key is *pk* = (*N*, *p*, *q*, *r*, *s*, *e*, *dp*, *dq*, *kp*, *kq*).
Encryption: It encrypts the data as in the Cloud-RSA scheme. Given the private key *pk*, let *m* ∈ ZN be a plaintext. The ciphertext *c* is computed as follows: *c* ≡ *E*(*pk*,*m*) ≡ *me* (mod *N*) (7) **Evaluation:** Given the evaluation key *ek*, let *c*1, *c*2, ..., *ck* be ciphertexts. The cloud provider then compute *Eval*(*ek*, *c*1, *c*2, ..., *ck*) ≡ *c*1 × *c*2 × ... × *ck* (mod *N*) ≡ *m*1*e* × *m*2*e* × ... × *mke* (mod *N*) ≡ (*m*1 × *m*2 × ... × *mk*)*e* (mod *N*) ≡ *E*(*pk*,*m*1,*m*2, ...,*mk*)
Output *C* = *Eval*(*ek*, *c*1, *c*2, ..., *ck*) as the result, where *Eval* is the evaluation function.
Decryption: it is the process of converting the cipher text(data) to the original plain text(data).
Steps:
1. The cloud user requests the Cloud service provider for the data.
2. Cloud service provider verify's the authenticity of the user and gives the encrypted data i.e, C.
3. The Cloud user then decrypts the data by computing, m = Cd (mod n).
4. Once m is obtained, the user can get back the original data
by reversing the padding scheme.

For example when encrypting a text with the numeric value of 0, it would encode as m = 0, which then again computes the ciphertext c = 0, with no concern about the values of n and e. The same goes for the numeric value of 1, which produces the value of 1 in ciphertext. This creates an insecure pattern, which might be analyzed by attackers and easily decrypted after gaining some knowledge about the encryption process. To avoid such problems in the algorithm, it is common to implement a randomized padding into the message before the encryption happens. This is to ensure that the message does not contain some insecure values and that the encrypted ciphertext contains some padded values that generate a larger ciphertext. This increases the level of complexity of the encryption, and will most likely make a dictionary attack harder to succeed.

Once the message arrives on the recipient's side of the communication channel, the ciphertext gets decrypted.using the private key in the following procedure: $m \equiv c^d \pmod n$, where c is the ciphertext. The most important advantage of RSA is ensuring about the privacy of the private key because this key will not be transmitted or revealed to another user. However, this algorithm has some considerable weaknesses. The main computational costs of the RSA are the modular exponentiations found during the key generation, encryption and decryption process [12]. Moreover, this algorithm has some weaknesses against certain attacks (i.e., Brute force, Mathematical attacks, Timing attacks and Chosen Cipher-text attacks) [13]. To reduce these problems, many algorithms have been designed and introduced based on original RSA. RSA Small-e and Efficient RSA are only two of the most popular algorithms identified for improving the main algorithm.

RSA algorithm Merits and De-Merits

Advantages:
1. It provides a safe, secure and protected transfer of data;
2. It makes it difficult for hackers/crackers to crack the file.

Disadvantage:
1. Time required by RSA is greater and makes the process slow when large data are used.

3.4 ElGamal Algorithm

In 1984 Taher ElGamal introduced a cryptosystem which depends on the Discrete Logarithm Problem.The ElGamal encryption system is an asymmetric key encryption algorithm for public-key cryptography which is based on the Di-e-Hellman key exchange.ElGamal depends on the one way function, means that the encryption and decryption are done in separate functions.It depends on the assumption that the DL can't be found in feasible time, while the reverse operation of the power can be computed efficiently .

The first public key system proposed by Diffie and Hellman requires association of both sides to compute a common private key. This poses issues if the cryptosystem should be applied to communication system where both sides are not able to interact in reasonable time because of deferrals in transmission or in accessibility of the receiving party. Means that the proposed scheme by Diffie and Hellman is not a general purpose encryption algorithm as it can only provide secure secret key exchange. Thus it presents a challenge for the cryptologists to design and provide a general purpose encryption algorithm that satisfies the public key encryption standards.So, after Di-e-Hellman, RSA public key cryptosystem came.AFter RSA, The ElGamal solved the Di-e-Hellman key exchange algorithm by presenting a random exponent type k. This exponent is a replacement for the private type of the receiving entity. Because of this simplication the algorithm can be utilized to encode in one heading, without the need of the second party

to take effectively part. The key development here is that the algorithm can be utilized for encryption of electronic messages, which are transmitted by the method for public store-and-forward services. Secure and efficient data storage was needed in the cloud environment in modern era of information technology industry. The cloud verifies the authenticity of the cloud services without the knowledge of user's identity. The cloud provides massive data access directly through the internet. Centralized storage mechanism was followed here for effective accessing of data. Cloud service providers are normally acquires the software and hardware resources and the cloud consumers are avail the services through the internet access in lease basis. Cloud security was enhanced through cryptography technique applied to the cloud security to avoid vulnerability. The intractable computability was achieved in the cloud by using the public key cryptosystem[14] have proposed the approach of applying hyper elliptic curve cryptography for data protection in the cloud with the small key size. The proposed system has the further advantage of eliminating intruder in cloud computing. Efficacy of the system was to provide the high security of the cloud data.

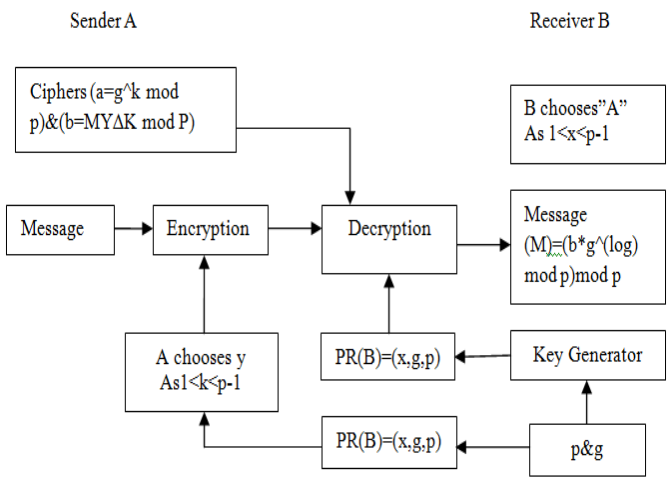


Figure 6: Model of Elgamal's Algorithm

Elgamal Algorithm[15]:

1. Get the File f to be stored on cloud.
2. Call elgmal_encryption( )
3. a. Generate Keys.
4. b. If (flength< p)then
5. E(f)← encrypt the file Elgmal(f)
6. else
7. fpart[x] ← create_file_partion( )
8. E(fpart[x] ← encrypt each fpart[x]
9. Concate each part to single fileE(f)□ +□ E(fpart[0])+E(fpart[1])+…….+E(fpart[n])
10. 3. Upload E (f) to cloud.

Determination Of Public Key And Private Key: A. Choose a large prime number p such that ø(p) has a large prime factor. Choose g: primitive root of p. B. Choose an integer x such that 1 <x< ø(p) . Compute y=g^x mod p. C. The triplet (x,g,p) forms the private key and the triplet (y,g,p) forms the public key of the user'A'. D. User 'A' keeps the private key (x,g,p) secret and makes the public key (y,g,p) available to all those users with whom 'A' intends to communicate.. 2.2 Encryption (Say By User 'B'): When any user (say user B) possessing A's public key (y,g,p) intends to send a message M(0≤M<k< ø(p)). B. Knowing public key (y,g,p) of intended recipient 'A'. user 'B' computes the cipher-text, which comprises of a pair of integers :-

ElGamal cryptosystem for encrypting messages

ElGamal cryptosystem provides the advantage of encrypting large messages. This encryption system provides users with the option of sending any type of messages with more secured access between the users. The ElGamal cryptosystem operates as follows, In ElGamal cryptosystem each user selects their own secret keys such that,

$$k_i \in [1, N-2) \quad (1)$$

Where,
N- Prime number
The related public key is then given by,

$$Pk = c^{ki} \bmod N \quad (2)$$

Where,
C - Primitive root or generator
For encrypting a large message using ElGamal cryptosystem, the sender computes the cipher text *s* and *t* as follows,

$$S = c^{j} \bmod N \quad (3)$$
$$t = M.P_{kj} \bmod N \quad (4)$$

Where, $j \epsilon [1, N-2)$ Now the receiver can decrypt the cipher *s* and *t* by computing,

$$M = t.(s^{ki})^{-1 \bmod N}$$

Based on the above process the ElGamal cryptosystem works and in order to make the encryption process more secure we have utilized some modification in normal ElGamal process by incorporating optimization process.

Decryption: The decryption algorithm works as follows: to decrypt a cipher text (c1,c2) with her private key x,

- A calculates the shared secret s=c1x
- Then computes m'=c2s-1 which he/she then converts back into the plaintext message m, where s-1 is the inverse of s in the group G. (E.g. modular multiplicative inverse if G is a subgroup of a multiplicative group of integers modulo n).

The decryption algorithm produces the intended message, since c2s-1 = m' hy (gxy)-1 = m' gxy g-xy = m'.

3.5 Elliptic curve cryptography

Elliptic curve cryptography is a class of public-key cryptosystem [16] [17]. ECC protocols assume that finding the elliptic curve discrete algorithm is infeasible. ECC provides strong security as RSA with smaller bits key, which implies faster performance and lower computational complexity. A 160-bit key in ECC has the same security level as 1024-bit key in RSA [18]. There are several parameters and algorithm choices which should be considered before implementing ECC system. Several curve

domain parameters (field representation, curve type), algorithm for field arithmetic, elliptic curve arithmetic, and protocol arithmetic can be influenced by security factors, platform, constraints, and communications environment [19]. Algorithms and coordinate systems, which are given in this paper, are used to emphasize the benefit of elliptic curve in cryptosystems and give an insight for technical people to implement a simple elliptic curve cryptosystem. This paper also offers several performance comparisons to show the tradeoffs between coordinate systems. It should be noted that there are several other researches which have tried to optimize ECC algorithms in several different ways [20][21][22].

Elliptical curve cryptography (ECC) is a public key encryption technique based on elliptic curve theory that can be used to create faster, smaller, and more efficient cryptographic keys. ECC generates keys through the properties of the elliptic curve equation instead of the traditional method of generation as the product of very large prime numbers. Because ECC helps to establish equivalent security with lower computing power and battery resource usage, it is becoming widely used for mobile applications. ECC is based on properties of a particular type of equation created from the mathematical group (a set of values for which operations can be performed on any two members of the group to produce a third member) derived from points where the line intersects the axes. Multiplying a point on the curve by a number will produce another point on the curve, but it is very difficult to find what number was used, even if you know the original point and the result. Equations based on elliptic curves have a characteristic that is very valuable for cryptography purposes: they are relatively easy to perform, and extremely difficult to reverse.
In cloud computing, a secured platform can be made by ECC. Based on the EC theory ECC is a public key encryption technology to create a smaller, faster and efficient cryptographic key. The key generation in ECC depends on the property of the EC equation. A 3072-bit RSA public key provides the security of only 256-bit ECC public key. One of the main advantages of ECC is a reduced key size with high security and less storage space. A sample EC is shown in Fig. 2.

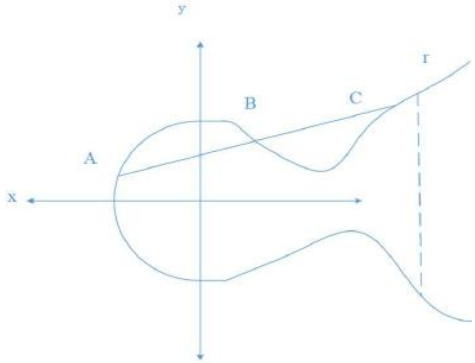


Figure 7: Eliptic curve

Consider an EC

$$y2 = x3 + ax + b \quad (1)$$

Where *a* and *b* are EC coefficients.
Let *Eq a, b* be an EC, consider the equation

$$B = kA \quad (2)$$

where *A* and *B* are two points in curve *A* ∈ *Eq a, b* and *k* < *r*. From the above equation, it is easy to find *B*, where *k* and *A* are given but the difficulty will come into the calculation of *k*. Because it is a trapdoor function or a one-way function and called it as a discrete logarithm.

Key Generation: Primary invention is an essential part where it should create both a public key and a private key. The correspondent encrypts the message to the recipient's social circle and the recipient clicks the private key. Now we have to select a number within "m" width. Use the following equity to generate a public key R = f * t f = random number selected within width (1 to m). 't' is a point of curve. 'f' is the private key and 'R' is the public key. Encryption Let 'l' be the message that, are sending, have to represent this message on the curve. Regard as 'l' have the point of "L" on the curve 'C'. At random choose 'j' since [1 to (m-1)]. Two cipher texts determination be generated let it be CP1 & CP2. CP1 = F*T CP2 = L + F*R CP1 and CP2 will be sending. Decryption To retrieve the message 'm' that was send to us, L = CP2 – f * CP1 M is the original message that we have send. Proof How do to get back the message? L = Cp2 – f * CP1 'L' can be represented as 'CP2 – f * CP1' = CP2 – f * CP1 = (L + F* Q) – d * (f * t) (C2 = L + F * R and C1 = r* t) = L + r * f * t – f * r *t (canceling out r* f * t) = L (Original Message).

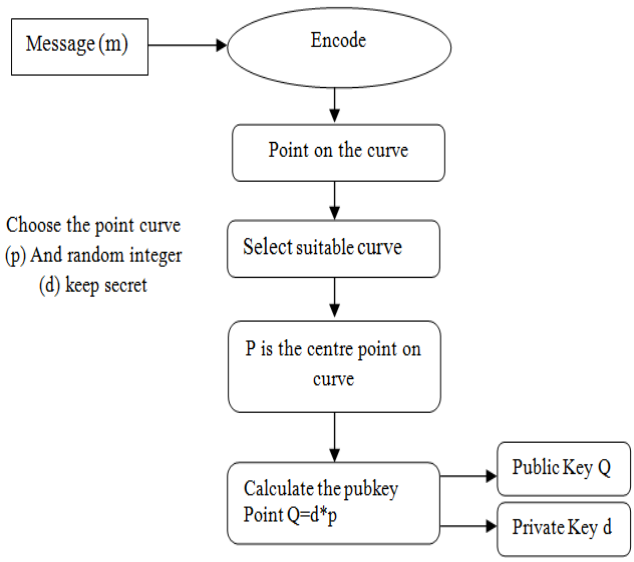


Figure 8: Generation Process Of ECC

Encryption algorithm: Suppose
A wants to send to **B** an encrypted message.
I. A takes plaintext message M, and encodes it onto a point, PM, from the elliptic group.
II. A chooses another random integer, k from the interval [1, p-1]
III. The cipher text is a pair of points : PC = [(kB), (PM + kPB)]
IV. Send cipher text PC to cloud B.

Decryption algorithm: Cloud

B will take the following steps to decrypt cipher text PC

a. B computes the product of the first point from PC and his private key, dB dB * (kB)

b. B then takes this product and subtracts it from the second point from PC (PM + kPB) – [dB(kB)] = PM + k(dBB) – dB(kB) = PM

c. B cloud then decodes PM to get the message, M.

ECC for portable devices and its application When the ECC was first introduced in 1985, there was a lot of skepticism about its security. But, ECC has come a long way since then. After nearly a decade of serious study and scrutiny, ECC has yielded highly efficient and secure. Presently, many product vendors have incorporated ECC in their products, and this number has only been on the rise. Uncertainty still exists among some proponents of traditional cryptographic systems, but they are starting to become more accepting of this promising new technology. RSA Security Inc., for example, has long voiced concern regarding the security of ECC since its introduction. In recent years, however, RSA Security has researched on efficient ECC algorithms, and even acquired a patent on a storage-efficient basis conversion algorithm. Moreover, it has also integrated ECC into some of its products, acknowledging the fact that ECC has begun to establish itself as both secure and efficient.

Example:

Input : Jansirani,Sri Vijay Vidyalaya College of Arts & Science,Dharmapuri

Curve: $y^2 = x^3 + 15$

Point P = (1.35|4.17)

Point Q = (2.2|5.06)

Point R = P + Q = (-2.46|-0.2)

y^2 = x^3 + ax + b, where

a = ffffffffffffffffffffffffffffffeffffffffffffffc,

b =

64210519e59c80e70fa7e9ab72243049feb8deecc146b9b1

field order = fffffffffffffffffffffffffffffffeffffffffffffff

p =

6277101735386680763835789423207666416083908700390324961279

Point P =

(2f415757108c77f29034945a95e8973752cfef858f5bc5c3,

d5b817edbef8e10ca58e4f143cebdcc95a971eb45b936247)

Point Q =

(14833e549cf1429b41b6bb61850169e2d2e7abacabdc9b1c,

104fb018845f8a13a1ec21ea82b09f4950628b1d5250cf00)

R = P + Q

Point R =

(e383100dd46aedf551f617357c04cf28cc78d6f4ba563809,

2011237aa217feb539861c5377525c2388e143f6b6e8e87f)

y^2 = x^3 + ax + b, where

a = ffffffffffffffffffffffffffffffeffffffffffffffc,

b =

64210519e59c80e70fa7e9ab72243049feb8deecc146b9b1

field order = fffffffffffffffffffffffffffffffeffffffffffffff

p = ffffffffffffffffffffffffffffffeffffffffffffffff

Point P =

(a70b9df0b8fe746304a37fc1095f51ac30782a6a2c4c64e1,

299d0e6a7f9088fd679eb14c0554efe284a780f3fc503d3e)

Point Q =

(b8f55b16b83c872ff078c18a8ceb7fcb58555a3a261abf86,

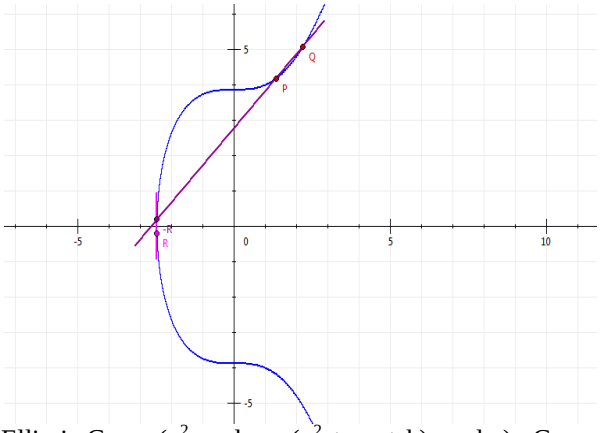
bb40cedfd87c996fa6f4d5653c8bf539b80e78d49ab1bba)

R = P + Q

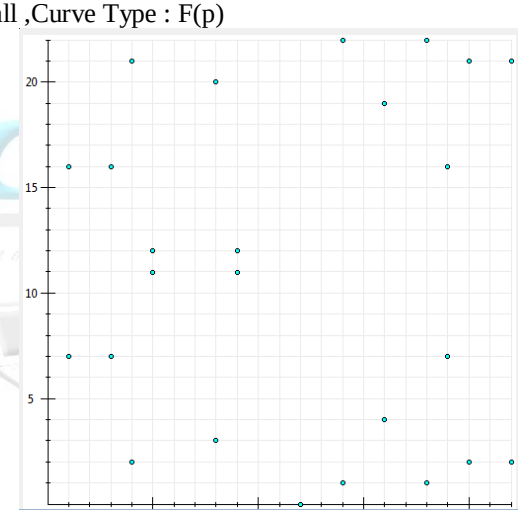
Point R =

(dafebf5828783f2ad35534631588a3f629a70fb16982a888,

dd6bda0d993da0fa46b27bbc141b868f59331afa5c7e93ab)



Elliptic Curve (y^2 mod p = (x^2 + ax + b) mod p) , Curve Size : Small ,Curve Type : F(p)



IV.EXPERIMENTAL RESULTS

Compare asymmetric cryptography algorithms DSA, RSA, ELGAMMAL and ECC using a 4GB -RAM and Intel core i5-2450 M CPU 2.50GHz machine. For encryption and decryption operations 5542 KB block size is used. By applying test data the security algorithms is evaluated in terms of the execution time required to store or retrieve the text data at cloud. The Simulation program inputs are: Algorithm and data block. Subsequent to a successful execution i.e. encryption and decryption process generate an efficient result. The analytical table is formed after the successful encryption / decryption process. To make sure that all the data are processed in the precise way. Basically, it is depend upon execution time (Encryption and Decryption time) as parameters.

Security The key length of our implementation we have used a 160 to 192 bit, which is quit better to protect against

naive attack. The key length is increased for better security by using the encryption and decryption process. The DSA, RSA, algorithms can provide to determine the length of the encryption keys and an arbitrary level of security used for each algorithm. Tables represent the required key length using different encryption algorithms in order to complete a level of security similar to the RSA key length provided by 1024-bit RSA encryption. The times for key generation, signature, and verification algorithms have computed with comparable key sizes for RSA, ECC, ECDH and ECDSA. **The results of the report showed that ECDSA outperformed RSA in both key and signature generations**. However, ECDSA was capable to verify messages much faster than RSA. The key sizes ranged from 1024 to 15360 bits for RSA and 163 to 571 bits for ECDSA algorithms. In **ECDSA key generation are consistently faster than those of RSA.**

Key Generation Time : For performance measure RSA, DSA, ELGAMMAL, ECC implementation is entirely constant time. figure 9. ECC Key generation performs better than DSA,RSA,ELGAMMAL at all key lengths, and is especially obvious when we increase the length of the key.

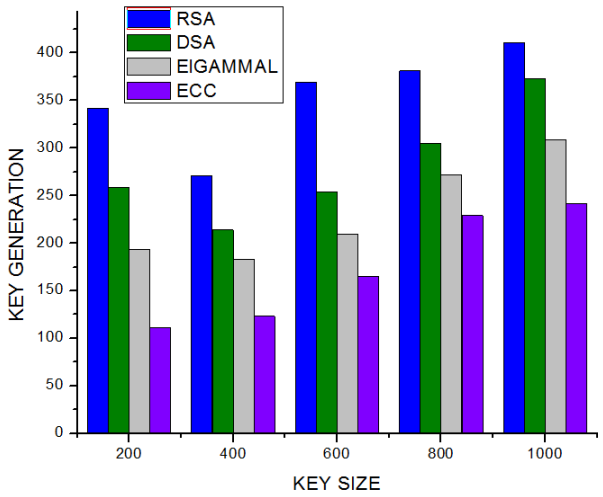


Figure 9:Key Generation

Despite that the RSA,DSA,ELGAMMAL does not have dedicated resources to the computationally intensive generation of prime numbers, but it is superior ECC in speed to produce the public/private key using comparable lengths Compare to other algorithms ECC key generation time was faster, because ECC have different key lengths. The comparison shows very small ratios whatever the security level. This means the key generation with ECC is always faster than the key generation with others. The RSA, DSA, ELGAMMAL keys are generated using large prime numbers thus take significantly longer than the smaller ECC keys that are generated.

Encryption and Decryption Time : Figure 10 and 11 shown, Encryption and decryption performance for the various algorithms are difficult to measure and are heavily influenced by system architecture and software/hardware optimizations.

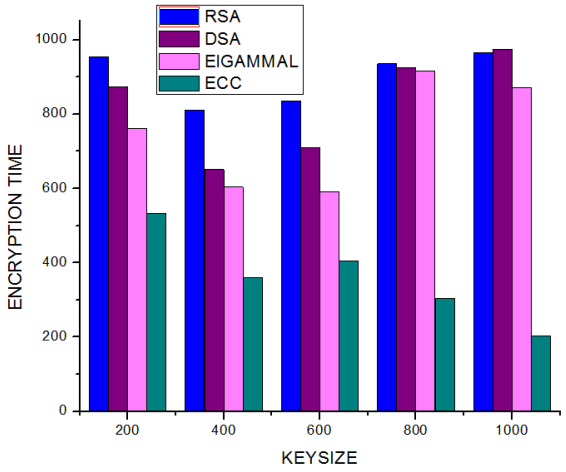


Figure10: EncryptionTime

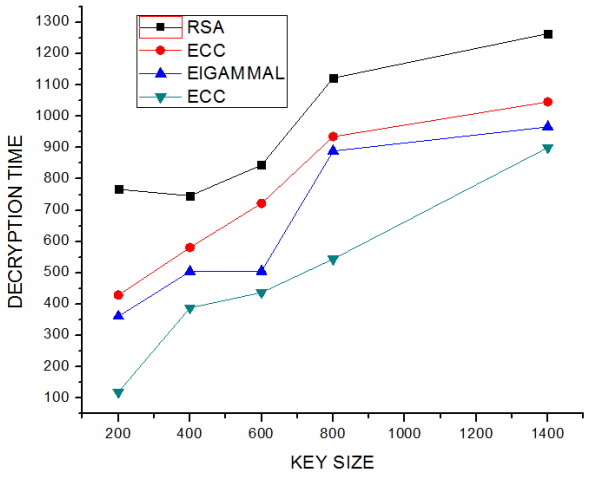


Figure11: Decryption Time

Compared to RSA, ECC offers better key pair generation performance, RSA requiring several time to generate large primes when compared to much smaller ECC key pair, RSA encryption lesser than ECC at RSA bit lengths of 1024 and above, although ECC decryption several times faster than RSA, finally both are efficient enough not to grant a system traffic jam problems. The ECDH and ECDSA algorithms provides similar processing time as ECC because both algorithm implementation is similar to ECC, but both take longer due to multiple exchanges steps are involved. Based on the result shown in figure 2 and 3 the ECDSA algorithm better in both encryption and decryption process compare to other algorithms.

Time complexity of an algorithm quantifies theamount of time taken by an algorithm to run as a function of the lengthof the input. Similarly, Space complexity of an algorithm quantifies theamount of space or memory taken by an algorithm to run as a functionof the length of the input.in figure 3 shows Time Complexity of Security Algorithms (DSA, RSA, ELGAMMAL and ECC), Based on

the result shown in figure 2 the ECC algorithm better in time complexity process compare to other algorithms.

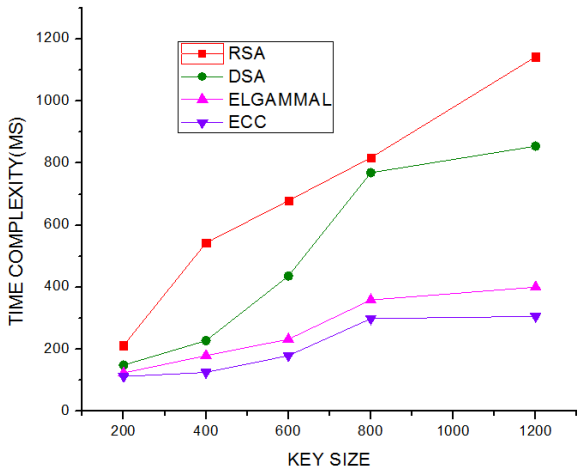


Figure12:Time Complexity

V.CONCLUSION

Cloud computing allows consumers to use applications without installation and access their personal files at any computer with internet access. In cloud computing technology there are a set of important policy issue, which includes issue of privacy, security, anonymity, telecommunications capacity, and reliability among others. But the most important between them is security and how cloud provider assures it. In this paper analyses the importance of security in cloud and time complexity to process the better time to achieve the work. We compared four algorithms, four algorithms for asymmetric algorithm for data security in cloud. Moreover, we concluded that the algorithms implemented are more efficient on cloud environment.

VII. REFERENCES

[1]. Harfoushi, O.; Obiedat, R. Security in Cloud Computing Using Hash Algorithm: A Neural Cloud Data Security Model. Mod.Appl. Sci. 2018, 12, 143–150.

[2]. Alharabi, M.F.; Aldosari, F.; Alharbi, N.F. Review of Some Cryptographic Algorithms In Cloud Computing. Int. J. Comput. Sci.Netw. Secur. 2021, 21, 41–50.

[3]. T. Hardjono and L. R. Dondeti, Security in wireless *LANs and MANs: Artech House*, 2005.

[4]. Y. Lei, D. Chen, and Z. Jiang, "Generating Digital Signatures on Mobile Devices", Proc. 18th International Conference on Advanced Information Networking and Applications, Mar. 2004, pp. 532-536.

[5]. National Institute of Standards and Technology, NIST FIPS PUB 186. „Digital Signature Standard", U.S. Departmentof Commerce, May1994.

[6]. Kaliyamurthie K.P., Udayakumar R., Parameswari D., Mugunthan S.N., Highly secured online voting system over network, Indian Journal of Science and Technology, v-6, i-SUPPL.6, pp-4831-4836, 2013.

[7]. Thooyamani K.P., Khanaa V., Udayakumar R., Efficiently measuring denial of service attacks using appropriate metrics, Middle - East Journal of Scientific Research, v-20, i-12, pp-2464-2470, 2014.

[8]. R. Bhanot and R. Hans, "A Review and Comparative Analysis of Various Encryption Algorithms," vol. 9, no. 4, pp. 289–306, 2015.

[9]. B. Dhanalaxmi, "M ultimedia C ryptography - A R eview," 2017 IEEE Int. Conf. Power, Control. Signals Instrum. Eng., pp. 764–766, 2017

[10]. Z. Hercigonja, "Comparative Analysis of Cryptographic Algorithms," vol. 1, no. 2, pp. 127–134, 2016.

[11]. F. Maqsood and M. A. Shah, "Cryptography : A Comparative Analysis for Modern Techniques," no. August, 2017.

[12]. H. M. Sun, M. E. Wu, W. C. Ting, and M. J. Hinek, ―Dual RSA and Its Security Analysis,‖ *IEEE Trans. on Information Theory*, vol. 53, no. 8, pp. 2922-2933, August 2007.

[13]. A. Alhasib and A. M. Haque, ―A Comparative Study of the Performance and Security Issues of AES and RSA Cryptography,‖ in *Proc. 3rd International Conference on Convergence and Hybrid Information Technology (ICCIT)*, Busan, 2008, pp. 505-510.

[14]. S. Selvi, and R. Ganesan, "A Secured Cloud System using Hyper Elliptic Curve Cryptography", International Journal of Scientific & Engineering Research, Vol. 6, No. 1, 2015.

[15]. Eman M. Mohamed, Hatem S. Abdelkader and Sherif el-etriby, "Enhanced Data Security Model for Cloud Computing," IEEE 8th International Conference on Informatics and Systems (INFOS2012), pp. CC12-CC17 , 2012

[16]. N. Koblitz, "Elliptic curve cryptosystems," Mathematics of computation 48.177, 1987, pp. 203-209.

[17]. V. S. Miller, "Use of elliptic curves in cryptograph,." Advances in Cryptology—CRYPTO'85 Proceedings.

[18]. G. V. S. Raju and Rehan Akbani, "Elliptic curve cryptosystem and its applications," IEEE Systems, Man and Cybernetics, vol. 2, 2003.

[19]. M. Brown, et al., Software implementation of the NIST elliptic curves over prime fields. Springer Berlin Heidelberg, 2001.

[20]. H. Cohen, A. Miyaji, and T. Ono, "Efficient elliptic curve exponentiation using mixed coordinates," Advances in Cryptology— ASIACRYPT'98. Springer Berlin Heidelberg, 1998.

[21]. A. Miyaji, T. Ono, and H. Cohen, "Efficient elliptic curve exponentiation," Information and Communications Security, 1997, pp. 282-290.

[22]. M. Rivain, "Fast and regular algorithms for scalar multiplication over elliptic curves," IACR Cryptology ePrint Archive, 2011, pp. 338.