International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 3 Issue 5, August 2019 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Enhanced Cloud Security Implementation using Modified ECC Algorithm

R. Dhiviya[1], K. Mohamed Amanullah[2]

[1]Research Scholar, [2]Associate Professor

[1,2]Department of Computer Application, Bishop Heber College, Tiruchirappalli, Tamil Nadu, India

How to cite this paper: R. Dhiviya | K. Mohamed Amanullah "Enhanced Cloud Security Implementation using Modified ECC Algorithm" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-3 | Issue-5, August 2019, pp.2225-2230, **https://doi.org/10.31142/ijtsrd27870**

IJTSRD27870

ABSTRACT

Cloud computing is a distributed environment that encompasses thousands of computers that work in parallel to perform a task in lesser time than the traditional computing models. This parallelism enables the low cost virtualization of hardware resources with increased computational performances. Cloud computing provides tremendous opportunity for small and medium scale enterprises to grow their business using IT services with zero deployment cost. Whenever, a task is distributed over web, there encounters a series of potential threats that challenges the security of data such as buffer overflow, session hijacking and black hole attacks.

A cloud computing based services also face such kinds of security issues where applications deployed on cloud can face same kind of attacks as that on client-server model. Storage as a Service (SaaS) based applications are vulnerable to virus attacks .Online operating systems are available on cloud to the user for free .Viruses can spread as attachments of email, of part of the software or can stay in Master Boot Record (MBR) of the operating system available on cloud. Worms residing on one system in cloud can migrate to another system on its own. Trojan horse is software with wrong intentions. Thus the present system needs an effective mechanism to address the problem encountered in cloud computing.

KEYWORDS: *Cloud computing, Security challenges*

I. INTRODUCTION

Cloud computing is a flexible, cost-effective and proven delivery platform for providing business or consumer IT services over the Internet. Cloud computing supports distributed service oriented architecture, multi-users and multi-domain administrative infrastructure, it is more prone to security threats and vulnerabilities. At present, a major concern in cloud adoption is its security and Privacy. Intrusion prospects within cloud environment are many and with high gains. Security and Privacy issues are of more concern to cloud service providers who are actually hosting the services. In most cases, the provider must guarantee that their infrastructure is secure and clients' data and applications are safe by implementing security policies and mechanisms. While the cloud customer must ensure that provider has taken proper security measures to protect their information.

The issues are organized into several general categories: trust, architecture, identity management, software isolation, data protection, availability Reliability, Ownership, Data Backup, Data Portability and Conversion, Multiplatform Support and Intellectual Property. A cloud computing based services also face such kinds of security issues where applications deployed on cloud can face same kind of attacks as that on client-server model. Storage as a Service (SaaS) based applications are vulnerable to virus attacks This thesis is intended to provide an enhanced security service in cloud computing model using an enhanced Elliptic Curve Cryptography algorithm for securing user data over cloud.

The thesis is also extended to present both the theoretical and empirical results of the proposed improved elliptic curve based public key cryptography to prove that the model is better than the traditional AES based schemes in terms of encryption, decryption time and key sizes.

II. EXISTING SYSTEM

Public-key cryptography is based on the intractability of certain mathematical problems. Early public-key systems are secure assuming that it is difficult to factor a large integer composed of two or more large prime factors. For elliptic-curve-based protocols, it is assumed that finding the discrete logarithm of a random elliptic curve element with respect to a publicly known base point is infeasible: this is the "elliptic curve discrete logarithm problem" (ECDLP). The security of elliptic curve cryptography depends on the ability to compute a point multiplication and the inability to compute the multiplicand given the original and product points. The size of the elliptic curve determines the difficulty of the problem.

III. PROPOSED SYSTEM

In this research work AES algorithm is implemented for authentication purpose and Improved ECC algorithm is used for file (document) encryption in Cloud storage. There is facility to block unauthorized user, forget password and secret no. is sent to personal email account along with file encryption, upload, download and decryption. First objective

of proposed work is to make the system secure so that only authorized user can login in the cloud, if any unauthorized user try to access our private cloud here can easily track and permanently block his/her IP and even MAC address of device from where he/she is try to access our private cloud. Second is to make the file sharing in private cloud totally secure using ECC algorithm, and which is hard to decrypt and to make the packets travel securely in network using ECC, so that any hacker cannot intercept or decrypt any packet.

Figure 3.1 shows the complete working for proposed system. It describes that after registration if any user is trying to login and if password is wrong or MAC address is wrong for 5 times the account was blocked. Figure 3.2 describes the possible operations for proposed system; these operations can be applied on document (files) for their security.

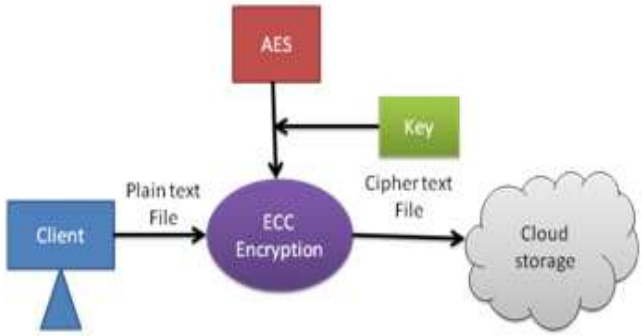

Figure3: Proposed Architecture

A. Elliptic Curves

First of all: what is an elliptic curve? Wolfram MathWorld gives an excellent and complete definition. But for our aims, an elliptic curve will simply be the set of points described by the equation:

$$y^2=x^3+ax+b$$

Where $4a^3+27b^2\neq 0$ (this is required to exclude singular curves). The equation above is what is called *Weierstrass normal form* for elliptic curves.

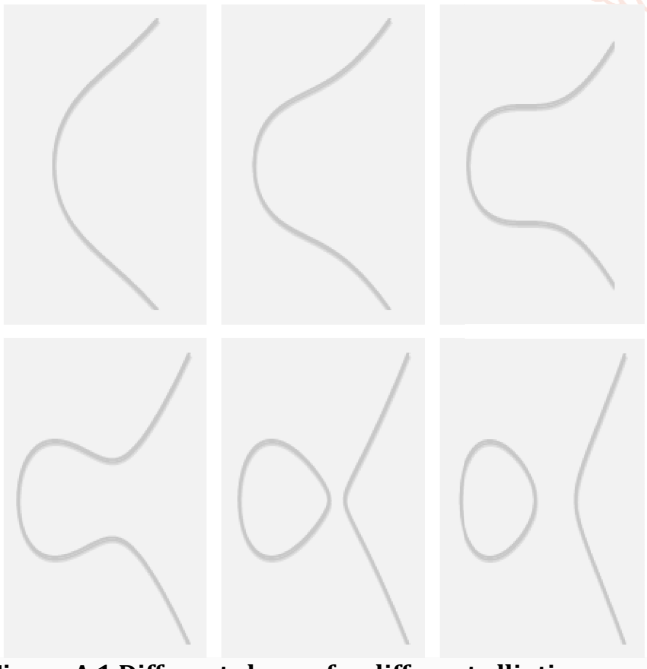

Figure A.1 Different shapes for different elliptic curves (b=1, a varying from 2 to -3).

Figure A.2 Types of singularities: on the left, a curve with a cusp ($y2=x3$). On the right, a curve with a self-intersection ($y2=x3-3x+2$). None of them is a valid elliptic curve.

Depending on the value of a and b, elliptic curves may assume different shapes on the plane. As it can be easily seen and verified, elliptic curves are symmetric about the x-axis.

For our aims, here will also need a point at infinity (also known as ideal point) to be part of our curve. From now on, we will devote our point at infinity with the symbol 0 (zero).

If we want to explicitly take into account the point at infinity, we can refine our definition of elliptic curve as follows:

$$\{(x,y)\in\mathbb{R}^2 \mid y^2=x^3+ax+b,\ 4a^3+27b^2\neq 0\}\cup\{0\}$$

B. Groups

A group in mathematics is a set for which we have defined a binary operation that we call "addition" and indicate with the symbol +. In order for the set G to be a group, addition must defined so that it respects the following four properties:

1. **Closure:** if a and b are members of G, then a+b is a member of G;
2. **Associativity:** (a+b)+c=a+(b+c);
3. There exists an **identity element** 0 such that a+0=0+a=a;
4. Every element has an **inverse**, that is: for every a there exists b such that a+b=0. If we add a fifth requirement.
5. **Commutativity:** a+b=b+a, then the group is called abelian group.

With the usual notion of addition, the set of integer numbers Z is a group (moreover, it's an abelian group). The set of natural numbers N however is not a group, as the fourth property can't be satisfied.

Groups are nice because, if we can demonstrate that those four properties hold, we get some other properties for free. For example: the identity element is unique; also the inversesare unique, that is: for every a there exists only one b such that a+b=0 (and we can write b as −a). Either directly or indirectly, these and other facts about groups will be very important for us later.

The group law for elliptic curves. Here can define a group over elliptic curves. Specifically:

- The elements of the group are the points of an elliptic curve;

- The identity element is the point at infinity 0;
- The **i**nverse of a point P is the one symmetric about the x-axis;
- Addition is given by the following rule: given three aligned, non-zero points P, Q and R, their sumis P+Q+R=0.

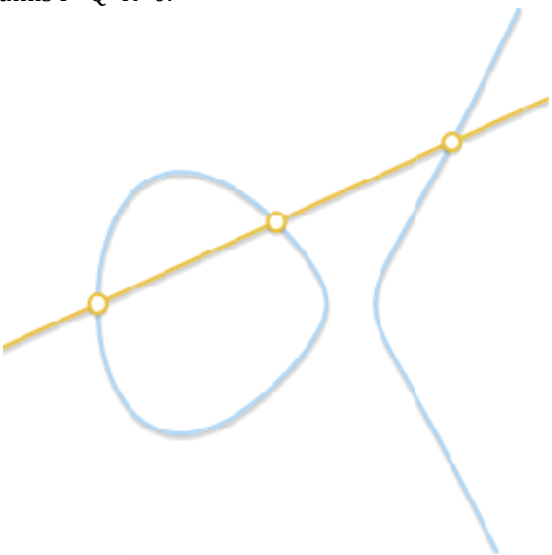

Figure B.1 the sum of three aligned point is 0.

Note that with the last rule, we only require three aligned points, and three points are aligned without respect to order. This means that, if P, Q and R are aligned, then P+(Q+R)=Q+(P+R)=R+(P+Q)=⋯=0. This way, we have intuitively proved that our **+** operator is both associative and commutative: we are in an abelian group.

So far, so great. But how do we actually compute the sum of two arbitrary points?

C. Geometric addition

Thanks to the fact that we are in an abelian group, we can write P+Q+R=0 as P+Q=−R. This equation, in this form, lets us derive a geometric method to compute the sum between two points P and Q: if we draw a line passing through P and Q, this line will intersect a third point on the curve, R (this is implied by the fact that P, Q and R are aligned). If we take the inverse of this point, −R, we have found the result of P+Q.

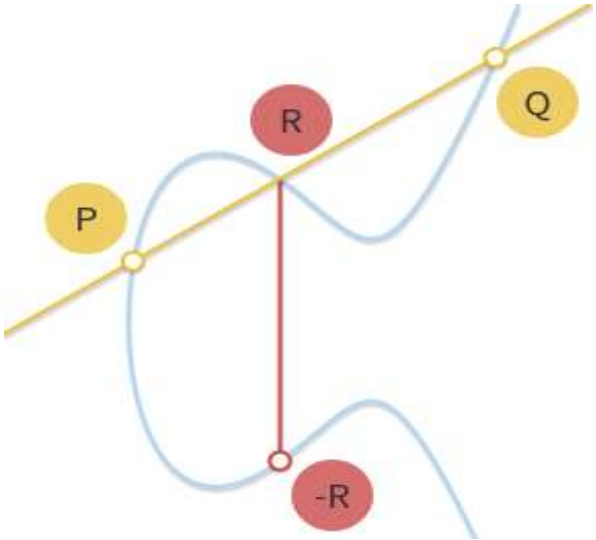

Figure C.1 Draw the line through P and Q. The line intersects a third point R. The point symmetric to it, −R, is the result of P+Q.

This geometric method works but needs some refinement. Particularly, we need to answer a few questions:

- What if P=0 or Q=0? Certainly, we can't draw any line (0 is not on the xy-plane). But given that we have defined 0 as the identity element, P+0=P and 0+Q=Q, for any P and for any Q.
- What if P=−Q? In this case, the line going through the two points is vertical, and does not intersect any third point. But if P is the inverse of Q, then we have P+Q=P+(−P)=0from the definition of inverse.
- What if P=Q? In this case, there are infinitely many lines passing through the point. Here things start getting a bit more complicated. But consider a point Q′≠P. What happens if we make Q′approach P, getting closer and closer to it?

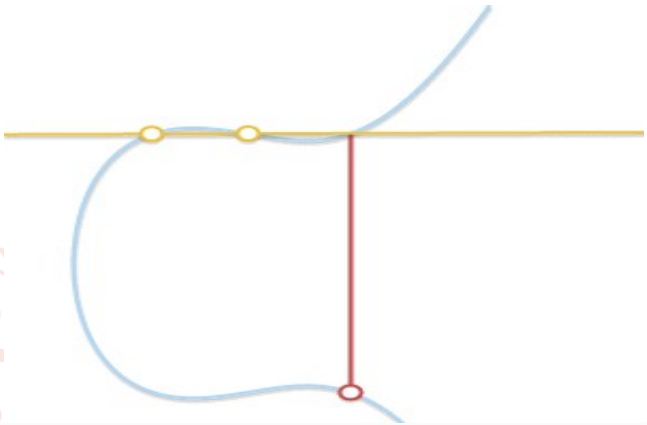

Figure C.2 As the two points become closer together, the line passing through them becomes tangent to the curve.

- As Q′ tends towards P, the line passing through P and Q′ becomes tangent to the curve. In the light of this we can say that P+P=−R, where R is the point of intersection between the curve and the line tangent to the curve in P.
- What if P≠Q, but there is no third point R**?** We are in a case very similar to the previous one. In fact, we are in the case where the line passing through P and Q is tangent to the curve.

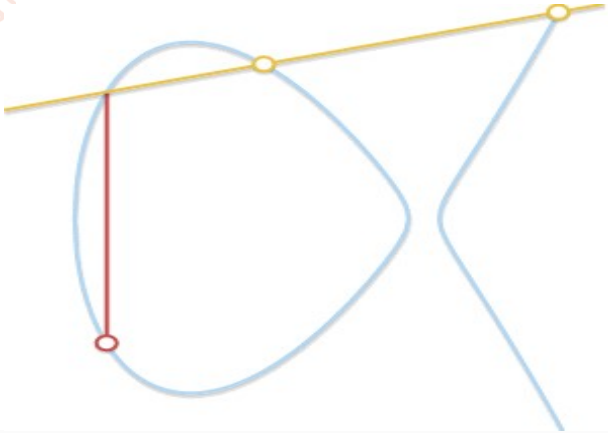

Figure C.3 If our line intersects just two points, then it means that it's tangent to the curve. It's easy to see how the result of the sum becomes symmetric to one of the two points.

- Let's assume that P is the tangency point. In the previous case, we would have written P+P=−Q. That equation now becomes P+Q=−P. If, on the other hand, Q were the tangency point, the correct equation would have been P+Q=−Q.

The geometric method is now complete and covers all cases. With a pencil and a ruler we are able to perform addition involving every point of any elliptic curve. If you want to try, take a look at theHTML5/JavaScript visual tool I've built for computing sums on elliptic curves!

D. Algebraic addition

If here want a computer to perform point addition, we need to turn the geometric method into an algebraic method. Transforming the rules described above into a set of equations may seem straightforward, but actually it can be really tedious because it requires solving cubic equations. For this reason, here I will report only the results.

First, let's get get rid of the most annoying corner cases. We already know that $P+(-P)=0$, and we also know that $P+0=0+P=P$. So, in our equations, we will avoid these two cases and we will only consider two non-zero, non-symmetric points $P=(xP,yP)$ and $Q=(xQ,yQ)$.

If P and Q aredistinct ($xP \neq xQ$), the line through them has slope:

$$m=y_P-y_Q \,/\, x_P-x_Q$$

The intersection of this line with the elliptic curve is a third point $R=(xR,yR)$:

$$xRyR==m2-xP-xQyP+m(xR-xP)$$

or, equivalently:

$$yR=yQ+m(xR-xQ)$$

Hence $(xP,yP)+(xQ,yQ)=(xR,-yR)$ (pay attention at the signs and remember that $P+Q=-R$).

If wanted to check whether this result is right, and would have had to check whether R belongs to the curve and whether P, Q and R are aligned. Checking whether the points are aligned is trivial, checking that R belongs to the curve is not, as we would need to solve a cubic equation, which is not fun at all.

The case P=Q needs to be treated a bit differently: the equations for xR and yR are the same, but given that xP=xQ, we must use a different equation for the slope:

$$m=3x2P+a \,/\, 2Yp$$

Note that, as we would expect, this expression for m is the first derivative of:

$$\surd\, yP=\pm x3P+axP+b$$

To prove the validity of this result it is enough to check that R belongs to the curve and that the line passing through P and R has only two intersections with the curve

E. Scalar multiplication

Other than addition, we can define another operation: scalar multiplication, that is:

$$nP = \underbrace{P + P + \cdots + P}_{n \text{ times}}$$

Where n is a natural number. I've written a visual tool for scalar multiplication too, if you want to play with that.

Written in that form, it may seem that computing nP requires n additions. If n has k binary digits, then our algorithm would be O(2k), which is not really good. But there exist faster algorithms.

One of them is the double and add algorithm. Its principle of operation can be better explained with an example. Take n=151. Its binary representation is 100101112. This binary representation can be turned into a sum powers of two:

$151=1\cdot2^7+0\cdot2^6+0\cdot2^5+1\cdot2^4+0\cdot2^3+1\cdot2^2+1\cdot2^1+1\cdot2^0$

(Here have taken each binary digit of n and multiplied it by a power of two.) In view of this, here can write:

$151\cdot P=2^7P+2^4P+2^2P+2^1P+2^0P$

What the double and add algorithm tells us to do is:

- Take P.
- *Double* it, so that we get 2P.
- *Add* 2P to P (in order to get the result of 21P+20P).
- *Double* 2P, so that we get 22P.
- *Add* it to our result (so that we get 22P+21P+20P).
- *Double* 22P to get 23P.
- Don't perform any addition involving 23P.
- *Double* 23P to get 24P.
- *Add* it to our result (so that we get 24P+22P+21P+20P).

In the end, here can compute 151·P performing just seven doublings and four additions.

If this is not clear enough, here's a Python snippet that implements the algorithm:

```
def bits(n):
    """
    Generates the binary digits of n, starting
    from the least significant bit.

    bits(151) -> 1, 1, 1, 0, 1, 0, 0, 1
    """
    while n:
        yield n & 1
        n >>= 1

def double_and_add(n, x):
    """
    Returns the result of n * x, computed using
    the double and add algorithm.
    """
    result = 0
    addend = x

    for bit in bits(n):
        if bit == 1:
            result += addend
    addend *= 2
    return result
```

If doubling and adding are both O(1) operations, then **this algorithm is** O(logn) (or O(k) if we consider the bit length), which is pretty good. Surely much better than the initial O(n) algorithm.

E.1 Encryption

- Let "m" be the message that we are sending.

- Here have to represent this message on the curve.
- Consider 'm' as the point 'M' on the curve 'E'.
- Randomly select „k" from [1 - (n- 1)].
- Cipher texts will be generated after encryption, let it be C1 and C2.
- C1 = k * p
- C2 = M + k * Q

E.2 Decryption

- The message "M" that was sent is written as following equation,
- M = C2 - d * C1

F. Advanced Encryption Standard (AES)

The Advanced Encryption Standard (AES), also known as Rijndael[1](its original name), is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001.

- AES is based on the Rijndael cipher developed by two Belgian cryptographers, Joan Daemen and Vincent Rijmen, who submitted a proposal to NIST during the AES selection process.[7] Rijndael is a family of ciphers with different key and block sizes.
- For AES, NIST selected three members of the Rijndael family, each with a block size of 128 bits, but three different key lengths: 128, 192 and 256 bits.
- AES has been adopted by the U.S. government and is now used worldwide. It supersedes the Data Encryption Standard (DES),which was published in 1977. The algorithm described by AES is a symmetric-key algorithm, meaning the same key is used for both encrypting and decrypting the data.
- In the United States, AES was announced by the NIST as U.S. FIPS PUB 197 (FIPS 197) on November 26, 2001. This announcement followed a five-year standardization process in which fifteen competing designs were presented and evaluated, before the Rijndael cipher was selected as the most suitable (see Advanced Encryption Standard process for more details).
- AES became effective as a federal government standard on May 26, 2002 after approval by the Secretary of Commerce. AES is included in the ISO/IEC 18033-3 standard. AES is available in many different encryption packages, and is the first publicly accessible and open cipher approved by the National Security Agency (NSA) for top secret information when used in an NSA approved cryptographic module.
- The name Rijndael is a play on the names of the two inventors (Joan Daemen and Vincent Rijmen). It is also a combination of the Dutch name for the Rhine River and a dale.

IV. RESULT

This section contains the working with elliptic curves which are defined over Zp. These are often called the prime curves and can be far simpler to work with as here can reduce modulo p at each stage. Suppose we have an elliptic curve, E, over Zp. In this case we have a cubic equation in which the variables and coefficients take values on the set of integers 0, 1, ... (p − 1) and all calculations are performed modulo p. y 2 ≡ x 3 - Ax - B (mod p) here write Ep(A, B) for the set of integers (x, y) that satisfy the above equation, together with a point at infinity, ∞.

The set E11(1, 6) is the set of integers (x, y) that satisfy

$$y\,2 \equiv x\,3 - x - 6 \pmod{11}$$

Here can see that (x, y) = (7, 9) is in this set as

$$9^{2} \pmod{11} = (73 + 7 + 6) \pmod{11}$$
$$81 \pmod{11} = 356 \pmod{11} \Leftarrow\Rightarrow 4 = 4$$

To find all the points in E11 (1, 6) here find all the possible values x 3 + x + 6 (mod p) and then see what values of y 2 will match. There are 11 choices of x, the integers {0, 1... 10}. Subbing these values in turn into the cubic and reducing modulo 11 will give us the possible values of y^{2}:

x = 0 ⟹ RHS = 6 x = 6 ⟹ RHS = 228 ≡ 8
x = 1 ⟹ RHS = 8 x = 7 ⟹ RHS = 356 ≡ 4
x = 2 ⟹ RHS = 16 ≡ 5 x = 8 ⟹ RHS = 526 ≡ 9
x = 3 ⟹ RHS = 36 ≡ 3 x = 9 ⟹ RHS = 744 ≡ 7
x = 4 ⟹ RHS = 74 ≡ 8 x = 10 ⟹ RHS = 1016 ≡ 4
x = 5 ⟹ RHS = 136 ≡ 4

So we can see that the possible values of y 2 are {3, 4, 5, 6, 7, 8, 9} i.e. y^{2} cannot be 0,1,2 or 10. Next examine the 10 possible values of y and identify which values of x they could be paired with to give a point on the curve.

y = 0 ⇒ y^{2} = 0 ⇒ No Points y = 6 ⇒ y^{2} = 36 ≡ 3 ⇒ x = 3
y = 1 ⇒ y^{2} = 1 ⇒ No Points y = 7 ⇒ y^{2} = 49 ≡ 5 ⇒ x = 2
y = 2 ⇒ y^{2} = 4 ⇒ x = 5, 7,10 y = 8 ⇒ y^{2} = 64 ≡ 9 ⇒ x = 8
y = 3 ⇒ y^{2} = 9 ⇒ x = 8 y = 9 ⇒ y^{2} =81≡ 4 ⇒ x = 5, 7, 10
y = 4 ⇒ y^{2} = 16 ≡ 5 ⇒ x = 2 y = 10 ⇒ y^{2} = 100 ≡ 1 ⇒ No Points
y = 5 ⇒ y_{2} = 25 ≡ 3 ⇒ x = 3

E11(1, 6) = {(2, 4), (2, 7), (3, 5), (3, 6), (5, 2), (5, 9), (7, 2), (7, 9), (8, 3), (8, 8), (10, 2), (10, 9), ∞} An m-file, PC.m, to find and plot all the points on a prime curve was constructed and is stored in Appendix C.2. This m-file takes as its inputs, A, B and p and produces two vectors X, Y which contain all the points (x, y) that lie on y 2 ≡ x 3 + Ax + B (mod p). When run on this example it verified that we had found found all the points in E11(1, 6) and plotted the graph below. Here can see that the points are symmetric about the line y = 5.5

Here can perform the elliptic curve addition operation on prime curves, however we reduce modulo p at each step. For example, still considering E11(1, 6):
If P = (8, 3) then we know that −P = (8, −3). Working modulo 11 we see that −P = (8, 8) which is also a point in E11(1, 6). Let P = (8, 3) and Q = (3, 5). Then to find R = P + Q:

m = (5 − 3) /(3 − 8) = 2/ −5 ≡ 2 /6 = 1 /3 = 1 × 4 = 4
The penultimate step involved taking the multiplicative inverse of 3 in Z11. Now proceed to show that
$x_R = 4^2 - 8 - 3 = 5$, yR = 4(8 − 5) − 3 = 9 So in E11(1, 6) we find (8, 3) + (3, 5) = (5, 9). • Again let P = (8, 3). To calculate 2P = P + P:
m = (3(8^2) + 1) / (2 ∗ 3) = 193 /6 ≡ 6/ 6 = 1 (mod 11)
Then $x_2P = 1^2 - 2(8) = -15 \equiv 7 \pmod{11}$
$y_2P = 1(8 - 7) - 3 = -2 \equiv 9 \pmod{11}$
So in E11 (1, 6) we find 2(8, 3) = (7, 9).

The earlier m-file for performing elliptic curve addition was modified for use with prime curves. It now reduces modulo p at each stage using mod function and find the inverse of elements so the final answer is an element on a prime curve. It contains the same inputs and outputs as m but the user must input p in addition. It makes use of the m-file inve.m

which is stored in Appendix C.4. This m-file takes as its inputs a number N and a prime p and outputs the inverse of N in the group Z_p. The m-file m was used to calculate the remaining entries in the addition table overleaf (Table 2.1). In show that (2, 7) is a generator of this group and so it is isomorphic to Z_{13}.

V. Conclusion

Data has become more important as the methods which are used to ensure security not only need to be strong and efficient but should be easy to implement and execute. Cloud computing is a modern concept that not just speeds up computing and cut costs. However, several challenges still weigh down the technology. Resolving security problems with cloud computing is one such major challenge. It requires an adequate understanding of both the security issues in cloud computing implementation as well as the solutions presently available to address these. The security model is used to improve security without degrading the performance of the system. Main goal of future improvement is providing more security by using more secure algorithm whose security can't be broken.

Simulation results shows that AES algorithm is best for authentication and ECC algorithm used for security has better performance than other techniques. Since ECC has not any known security weak points till now, it can be considered as an excellent standard encryption algorithm. The experimental results reveals that the proposed method offers better performance over previous work.

VI. Future Enhancement

In future here can use ECC algorithm for securing audio and video data. Because, In the area of security, research area of speech is very wide. The Android platform of smarphones is a powerful platform and is used in 80% of smartphones today. The sensors that come with the mobile devices further give a context to cloud applications and opens up a new set of possibilities.

VII. REFERENCES

[1] Arora, Akshay, Abhirup Khanna, Anmol Rastogi, and Amit Agarwal. "Cloud security ecosystem for data security and privacy." In *Cloud Computing, Data Science & Engineering-Confluence, 2017 7th International Conference on*, pp. 288-292. IEEE, 2017.

[2] Thu Yein Win "Big Data Based Security Analytics for Protecting Virtualized Infrastructures in Cloud Computing". *IEEE Transactions on Big Data (Volume: PP, Issue: 99)* 2017.

[3] Xiao-tao, Xu, Chen Zhe, Jiang Fei, and Wang Hui-tao. "Research on service-oriented cloud computing information security mechanism." In *Computer and Communications (ICCC), 2016 2nd IEEE International Conference on*, pp. 2697-2701. IEEE, 2016.

[4] Jayapandian, N., AMJ Md Zubair Rahman, R. B. Sangavee, and R. Divya. "Improved cloud security trust on client side data encryption using HASBE and Blowfish." In *Green Engineering and Technologies (IC-GET), 2016 Online International Conference on*, pp. 1-6. IEEE, 2016.

[5] Ahmad, Naim. "Cloud computing: Technology, security issues and solutions." In Anti-Cyber Crimes (ICACC), 2017 2nd International Conference on, pp. 30-35. IEEE, 2017.

[6] Shokri, Reza, George Theodorakopoulos, Panos Papadimitratos, Ehsan Kazemi, and Jean-pierre Hubaux. "Hiding in the Mobile Crowd: Location Privacy through Collaboration." In IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, SPECIAL ISSUE ON "SECURITY AND PRIVACY IN MOBILE PLATFORMS. 2016.

[7] Chhabra, Sakshi, and Ashutosh Kumar Singh. "Dynamic data leakage detection model based approach for MapReduce computational security in cloud." In Eco-friendly Computing and Communication Systems (ICECCS), 2016 Fifth International Conference on, pp. 13-19. IEEE, 2016.

[8] Bhamare, Deval, Tara Salman, Mohammed Samaka, Aiman Erbad, and Raj Jain. "Feasibility of Supervised Machine Learning for Cloud Security." In Information Science and Security (ICISS), 2016 International Conference on, pp. 1-5. IEEE, 2016.

[9] Yuan, Man, Shuning Pang, and Qiang Gao. "Design and development of data security automatic testing system on public cloud." In Software Engineering and Service Science (ICSESS), 2016 7th IEEE International Conference on, pp. 992-995. IEEE, 2016.

[10] El Makkaoui, Khalid, Abdellah Ezzati, Abderrahim Beni-Hssane, and Cina Motamed. "Cloud security and privacy model for providing secure cloud services." In Cloud Computing Technologies and Applications (CloudTech), 2016 2nd International Conference on, pp. 81-86. IEEE, 2016.