Volume 4, Issue 11, November 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

An Attempt towards the Analysis of Posteriori Time complexity of Elliptic Curve Cryptography Key Generations over Large Integers

Vaishnavi B
Research Scholar, VIT University,
Vellore, India

Abstract— One of the well known and applied public key cryptosystems over the network field is Elliptic curve Cryptosystem (ECC). In general, ECC has a major advantage over the crypto attacks i.e. ECC is at least four times stronger than the RSA cryptosystem with respect to the exposure of private key. In this paper, we are implementing the ECC with its tuned parameters and comparing its security level with the existing system.

Keywords—Public key cryptosystems, ECC, RSA, Network Security, cryptography.

I. INTRODUCTION

Cryptography basically establishes secured communication between two parties. There are two types of Network cryptographic Methods namely, symmetric key cryptography and asymmetric key cryptography. If the same set of keys involved in both encryption and decryption, then it is said to be symmetric key cryptography. If one pair of keys is used to encrypt and the other pair of keys is used to decrypt the same message (some mathematical calculations will be carried out to interconnect these two key pairs in order to extract the exact message), then it is called as Asymmetric key cryptography. There will be nothing complex in symmetric key cryptography. As it is easily breakable, it is unpopular. Hence, it is not a good idea to use it in network to share sensitive information. So, we move on to Asymmetric key cryptography.

As we already specified, it uses two pairs of keys, one pair is to encrypt and the other pair to decrypt to ensure the security. Though it uses two secret pair of keys, it is vulnerable to attack. Once if the hacker understands the algorithm that generates the keys, then it is easy for him to hack the system of communication. Hence we started generating the random number and use it in mathematical calculation to build the strong keys for Asymmetric key cryptography. Asymmetric key cryptography is also termed as public key cryptosystem. RSA is the first and foremost example of Asymmetric key cryptography. As we have done enough research in RSA, we move on to next popular algorithm called **"Elliptic Curve Cryptography (ECC)"**

II. RSA VS ECC

RSA was proposed in mid-1970s by Ron Rivest, Adi Shamir and Leon Adleman. The ECC was introduced in mid-1980s, by Neal Koblitz and Victor S. Miller. ECC is far better than RSA in key generation. For example, 163 bits of ECC will offer the same security strength as 1024 bits security strength offered by RSA. ECC saves memory usage, bandwidth, and time taken for computation as it uses small keys and it is very hard to break when compared to RSA and other cryptographic algorithms. Bit coin is One of the famous applications of ECDSA.

Five Major application areas based on Elliptic Curve Cryptography:

1. The elliptic Curve Diffie-Hellman (ECDH) Key agreement Scheme.
2. The Elliptical curve encryption scheme.
3. Elliptic curve Digital Signature Algorithm.
4. The Elliptic Curve Menezes-Qu-Vanstone (ECMQV) agreement scheme.
5. The ECQV implicit certificate scheme.

Research area in ECC: Finding vulnerabilities in ECC.

III. PRELIMINARIES AND BACKGROUND

ECC is a public key cryptographic algorithm. Initially we need an equation to draw a curve. The general equation of the elliptic curve is $\mathbf{y^2=x^3+ax+b}$**.** Then, a point is chosen from a curve where the line intersects the axis. By multiplying a number with the point we will get another point on the curve. If we want to increase the complexity, we can do this operation again and again to get more new points. This is the concept behind ECC.

The security lies behind this concept is, even if we know the initial and final point it's difficult to predict or plot the intermediate points and we do not know how many times the mathematical operations are carried over. So, it is easy to create but difficult to break. Here, we have two sorts of calculations 1. Point Addition 2.Point Doubling.

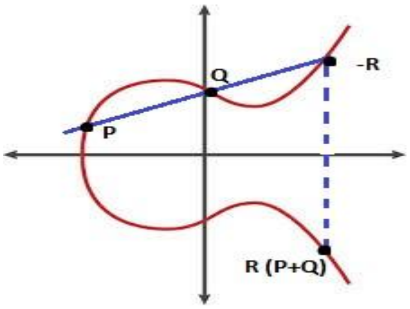


Fig.3.1. Point Addition

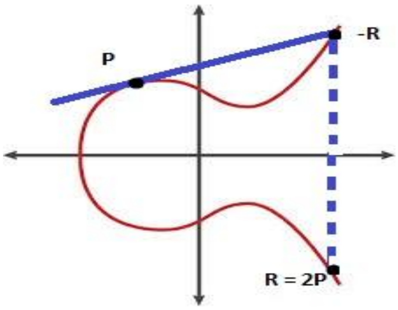


Fig.3.2. Point doubling

A. *Point Addition.*

We draw a small line by touching two points P and Q on an elliptic curve. We manage to get another point by extending the same line. So, we get a new (third) point. The symmetrically opposite point to the third point will be mathematically equal to the addition of first two points (Fig.:3.1).It has its own formulae to find the third point R(x_3, y3).

$$\lambda = \frac{y_2\text{-}y}{x_2\text{-}x_1} \text{ Mod P.}$$

$$x_3 = [(\lambda^2\text{-}\ x_1\text{-}\ x_2) \text{ Mod P}]$$

$$y_3 = [\lambda(\ x_1\text{-}\ x_3)\text{-}y_1] \text{ Mod P}$$

B. *Point Doubling:*

We draw a small line by touching only one point (may be at the edge) on the curve. Again we are extending the line in order to get the new point on the curve. The symmetrically opposite point to the new point will be mathematically equal to two times of the first point (Fig.:2).

$$\lambda = \frac{x_1^2+a}{2y_1} \text{ Mod P}$$

$$x_2 = [(\lambda^2\text{-}\ x_1\text{-}\ x_2) \text{ Mod P}]$$

$$y_2 = [\lambda(\ x_1\text{-}\ x_3)\text{-}y_1] \text{ Mod P}$$

IV. INSTANCE

On Elliptic curve y^2=(x^3+x+1) Mod 13 and e_1 (8, 1), d=3, r=2 , a=1; b=1;
Using these values encrypt the plain text P(3,2) and find the cipher text.

A. *Key generation:*

1. The user chooses point e_1(x_1, y1) in the elliptic curve over GF (P) and a random integer 'd' which is a private key.
2. Then calculate e_1(x_1, y1), e_2(x_2, y2) and E_p (a, b) as Public key.

B. *Encryption:*

1. Sender maps the plain text to a point P on the curve.
2. Finds C1=r *e_1 and C_1 =P +r*e_2 and transmits (C_1, C2) as cipher Text.
3. U -> ASCII P(x_1, y1).
4. V -> ASCII Q(x_2, y2).

C. *Decryption:*

1. The receiver after receiving C_1 and C_2 computes P as Original Text, Plain Text = C_2 - (d * C_1) and

C_1 = r * e_1
= 2 * (8, 1) =>2P
=> a = 1; x1=8; y1=1
=> By applying the point doubling formulae, we get, λ=12; x_2 =11; y_2=2;
C_1 = (11, 2).
C_2 = P+ r*e_2
= (3, 2) + 2*(10, 7) => (3, 2) + 2P
= (3, 2) + (10, 6) =>P+Q = (12, 4)
Plain Text = C_2-(d*c_1)
= (12, 4) – (3 * (11, 2))
= (12, 4) – (10, 6)

$$\begin{aligned} &= (12, 4) + (10, \underset{(+13)}{-6}) \\ &= (12, 4) + (10, 7) \\ &= (3, 2) \end{aligned}$$

V. LITERATURE SURVEY

Neha Tirthani et al. [1] proposed a new encryption mechanism for cloud. In this paper, they have categorized the major service models of cloud and discussed the main security threats in cloud environment. They gave an architecture which will be very helpful in designing cloud. They assured that the proposed algorithm will be more reliable and will also maintain the integrity of data. They have used Diffie-Hellman as it is better in establishing connection when comparing others and ECC (Elliptical Curve cryptography) to propose a new encryption method. They assert that the ECC has sub exponential time complexity. So it is difficult to crack.

Reza Azarderakhsh et al. [2] have compared three double point multiplication algorithms. The compared algorithms were *JT – {±1, ±3}, B-NBC, AK-DAC*. The parameters used to compare are the required area and the time taken to compute the calculation. They have also proposed the hardware architecture for these algorithms to implement. In addition to that, they have also implemented the double point multiplication algorithm over binary elliptical curves. The results showed that the AK-DAC algorithm performs better than other two (i.e.) differential addition chain based schemes are suitable to compute double point multiplication.

Graham Enos [3] has analysed four normal forms of elliptic curves namely Farashahi & Joye's Curve, Wang, Tang, & Yang's Curve, Wu, Tang, & Feng's Curve and Diao & Fouotsa's Curve. According to the view of author, when all these four curves are compared with Edward curves, Edward curves will perform better. The author has categorized the weakness in construction of the four curves into two. One is there is no commutativity (i.e.) the group of laws are not symmetric and the two is their choice of neutral point makes the result ambiguous and affects the result of adding a point and the neutral element. As Edwards's group laws are commutative, they also look simple, complete and unified.

Majid Bayad et al. [4] introduced a new authentication and key agreement protocol for wireless sensor networks based on elliptic curves. While talking about the password protection, the authors assured that the proposed scheme can withstand the dictionary attack. Xue et al already proposed a protocol called "Temporal – Credential based authentication and key". The authors are also reviewed the existing agreement protocol proposed by Xue et al and he argues that the existing protocol is exposed to the risk of being attacked by dictionary attack and stolen smart card attack . They also suggested some ideas to improve security threats that are found in the existing key agreement protocol system.

Joppe W. Bos et al [5] made a study of four Applications of elliptic curve cryptography in order to disclose the mistakes that happen during the implementation of ECC with the help of available ECC data like Key generation, Unexpected weak keys, etc. The four protocols are Bitcoin, secure shell, transport layer security and Austrian e-ID card. The comparison results are categorized based on deployment, weak keys and vulnerable signatures. According to the author, ECC is also not immune to some software bugs. In this paper, author encouraged the researchers to fix some loopholes in ECC and he also pinpointed some weak areas and suggested some ideas to fix it.

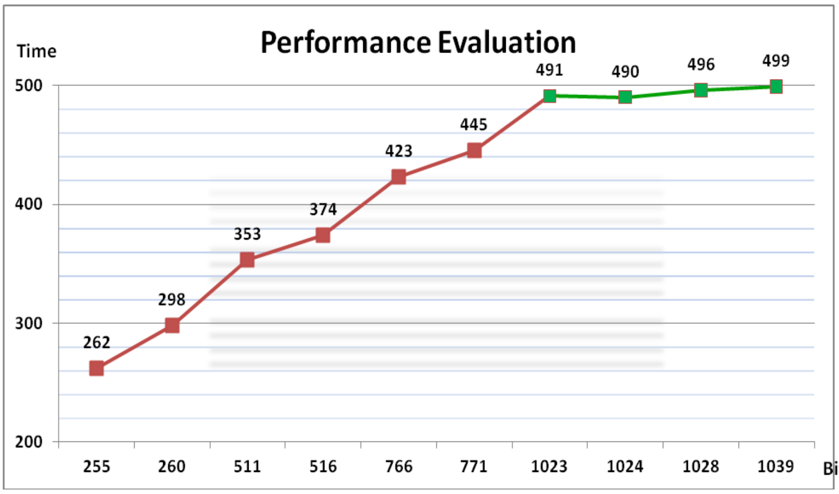
Sharad Kumar Verma [6] et al states some application of ECC. They stated that the ECC has good grip over security. They agree that this algorithm is easy to perform but difficult to reverse (in order to hack). They also explained the cryptographic algorithms based on elliptic curves. The algorithms are1) ECDH – Elliptic curve Diffie Hellman 2) ECDSA – Elliptic curve Digital Signature Algorithm. The clear cut idea of mathematical explanation is been given in this paper. This improves the interest of researchers on implementing the ECC.

Marisa W. Paryasto et al [7] have discussed the major issues while implementing the ECC in real time environment. The authors stated that the ECC can be implemented in software as well in hardware. Implementation ECC in software may consume more power when compared to hardware implementation and can perform in moderate speed. The hardware implementation of ECC in dedicated system can work faster with low power consumption and it is also more secure than software implementation. The main issue discussed in the paper in more power consumption. If this area is handled properly then there will not be any major issues in implementing the ECC.

Vivek B.Kute et al. [8] compared the RSA and ECC in all aspects (i.e.) encryption, decryption and key generation. The authors are also implemented the algorithms and analysed the results comparatively. The Results are displayed graphically. The Result shows that the RSA plays a good role in key generation and encryption only with respect to speed. And ECC is faster in decrypting task. The author agrees that ECC Algorithm is stronger than RSA.

Chandra Segar .T. et al. [9] has proposed an algorithm called Pell's RSA implemented based on the Pell's equation. They also assured that the proposed Pell's RSA would withstand the Wiener's attack. The authors have compared the proposed method with the traditional standard RSA and showed clearly how the proposed algorithm performs far better than conventional RSA.

VI. EVALUATION



Note: X-axis shows the bits: Y-axis shows the time in ns

Fig.6.1. Point doubling

VII. CONCLUSION

In general, the priori time to compute the keys are directly depends on the bit size. The result of the analysis shows that, whenever the bit length of the key increases more than 1000 the time taken to encrypt the text will attain the terminal state. The performance evaluation graph clearly shows that for 1024 to 1040 bits the time taken the limit with 491 to 499. From this observation, one can wisely choose the key bit size above 1024 bits with nominal computations.

REFERENCES

[1] Neha Tirthani, Ganesan R "*Data Security in Cloud Architecture Based on Diffie-Hellman and Elliptical Curve Cryptography*" Reza Azarderakhsh and Koray

[2] Karabina "*A Comparison of Double Point Multiplication Algorithms and their Implementation over Binary Elliptic Curves*"

[3] Graham Enos *"Complete and Unified Group Laws are not Enough for Elliptic Curve Cryptography"*

[4] Majid Bayat , Mohammad Reza Aref "*A Secure and efficient elliptic curve based authentication and key agreement protocol suitable or WSN*"

[5] Joppe W. Bos J. Alex Halderman, Nadia Heninger, Jonathan Moore, Michael Naehrig and Eric Wustrow *"Elliptic Curve Cryptography in Practice".*

[6] Sharad Kumar Verma, Dr. D.B. Ojha "*A Discussion on Elliptic Curve Cryptography and Its Applications"* IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 1, January 2012.

[7] Marisa W. Paryasto, Kuspriyanto, Sarwono Sutikno and Arif sasongko. *"Issues in Elliptic Curve cryptography Implementation".*

[8] Vivek B.Kute, P.R.Paradhi and G.R Bamnote *"A software comparison of RSA and ECC".*

[9] T Chandra Segar, R Vijayaragavan, "*Pell's RSA key generation and its security analysis"* Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT) IEEE, 4-6 July 2013, pp. 1-5, doi:10.1109/ICCCNT.2013.6726659

[10] M. Thangavel, , P. Varalakshmi, Mukund Murrali, and K. Nithya, "An Enhanced and Secured RSA Key Generation Scheme (ESRKGS)", in Journal of Information Security and Applications 6th November 2014, doi:10.1016/j.jisa.2014.10.004