ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)
Vol. 2 Issue Special 1 Jan-March 2014
International Journal of Advanced Research in Computer Science & Technology (IJARCST 2014)

An Asymmetric Authentication Protocol for Mobile Hand held Devices using ECC over Point Multiplication Method

[I]Sathish Kumar K, [II]Sukumar R, [III]Karthiyayini M
[I,II]Sethu Institute of Technology, Virudhunagar, Tamilnadu, India
[III]Ultra College of Engineering and Technology, Madurai, Tamilnadu, India

Abstract
Mobile hand –held device is an efficient way to deliver real time data to users in the battle field military applications. The very nature of military applications necessitates the use of security features such as data confidentiality, authentication etc., which are not readily offered by mobile environment. The energy expenditure in such an environment also poses a supplementary performance bottleneck while achieving privacy. Hence, it is necessary to design and implement an energy efficient authentication protocol that accomplishes a high level of security with minimum energy consumption. In this project, we are going to implement energy efficient authentication protocol for mobile handheld devices.

Keywords
Security, Cryptographic algorithms, Rivest, Shamir and Adleman (RSA), Elliptic Curve Cryptography (ECC)

I. Introduction

In an increasingly interconnected world, the interaction among mobile devices, systems, and people is growing rapidly. Businesses need to meet the demands of their employees and customers to allow for greater access to systems and information. By offering a scalable infrastructure model that enables companies to work smarter through more agile and cost-effective access to technology and information. Accessing the internet has become mandatory in many professions. Now mobile phone is also replacing the laptop by enabling internet access through the mobile phone. This has given way to service providers to provide various internet services. Mobile banking and stocks updates have become a common affair for the mobile phone user.

Portable communication system permits mobile users to enjoy global roaming services, and so the system is useful for the conversations conducted over wireless networks [1], [6]. In wireless networks, mobile users send and receive packets by wireless, and thus it is easy for anyone to eavesdrop on communicating messages transmitted over wireless networks. Hence, portable communication systems have more security vulnerabilities than wired systems. Several protocols have been proposed [1], [2], [3], [4], [5], [6], [7], [8] to secure the mobile systems. For secure roaming services, we need many security features such as the secrecy, authenticity, integrity, user privacy and non-repudiation. To achieve the goals, we use cryptographic algorithms such as secret-key systems and public-key systems. Among existing protocols, the great parts of them have been proposed based on secret-key systems since mobile devices have limited capacity. However, secret-key systems cannot provide the non-repudiation. Hence, we should use public-key systems when the property is inevitably required. When we use public-key systems, the cost of underlying mathematical operations is one of the difficulties. However, in these days, mobile devices have more computing power than before, and so it is possible to implement public-key systems on mobile devices. For critical commercial and military applications we propose a secure authentication scheme which incurs high level of security and less time consumption by using ECC over point multiplication method.

II. Related Work

To give greater perspective about the performance of the compared algorithms, this section discusses the results obtained from

A study conducted in [9] observes that this paper investigates the fast developing cryptography researchers and to increase the security development in the field of information security. ECC works with smaller keys to provide high security, high speed in a low bandwidth environment. The security level which is given by RSA, can be provided by ECC using smaller keys.

In [10], ECC is a promising system for the next generation public key cryptosystem. Although ECC's security has not been completely evaluated, it is expected to come into widespread use in various fields because of its compactness and high performance when it is hardware-implemented. ECC has been proven to involve much less overheads when compared to RSA.

A study conducted in [11] observes that the existing authentication protocols, based on RSA asymmetric cryptography are not suitable for such devices due to their confines in computing power, memory capacity, key sizes and cryptographic support. For that reason, an efficient protocol for resource constrained platforms that attain a level of security similar to the one achieved by the protocols in use today is designed and implemented. Elliptic curve asymmetric cryptography and the results demonstrate that the performance achieved is good in contrast to RSA.

In [12], The use of mobile devices demands to accommodate limitations on power and bandwidth, and to provide an adequate level of Security. The ECC used for such constrained environment. Its security comes from the elliptic curve logarithm, which is the DLP in a group defined by points on an elliptic curve over a finite field. This results in a dramatic decrease in key size needed to achieve the same level of security.

In [13], it highlights that the existing authentication protocols, based on RSA asymmetric cryptography, are not appropriate for such devices due to their limitations in computing power, memory capacity, key sizes and cryptographic support. This work shows that it is possible to implement the authentication protocol using HECC in resource constrained mobile devices with reasonable performance compared to RSA. Protocols based on this HECC asymmetric cryptography can be directly used in such devices. This paper addressed the design of a protocol based on HECC asymmetric cryptography. Furthermore, an implementation for J2ME Wireless Tool Kit 2.5.1 is also described.

A study in [14] was conducted for different popular secret key algorithms such as DES, 3DES, AES, and Blowfish. They were implemented, and their performance was compared by encrypting

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

input files of varying contents and sizes. The algorithms were tested on two different hardware platforms, to compare their performance. They had conducted it on two different machines: P-II 266 MHz and P-4 2.4 GHz. The results showed that Blowfish had very good performance compared to other algorithms. Also it showed that AES had a better performance than 3DES and DES. It also shows that 3DES has almost 1/3 throughput of DES, or in other words it needs 3 times than DES to process the same amount of data [15]. .

In [16] a study of security measure level has been proposed for a web programming language to analyze four Web browsers. This study considers the measuring of performances of encryption process at the programming language's script with the Web browsers. This is followed by conducting tests simulation in order to obtain the best encryption, Performance Evaluation of Symmetric Encryption Algorithms versus Web browser. It is concluded in [17] that there is an efficient authentication scheme, which is suitable for mobile devices. It uses an elliptic-curve-cryptosystem for mobile station authentication. This scheme enjoyed both computation efficiency and communication efficiency as compared to known mobile authentication schemes. In this paper, a novel and efficient mobile authentication scheme is proposed, and its security property has been analyzed. The scheme requires one scalar point multiplication operation and two short messages on mobile stations for each session establishment after the initial one-time delegation key verification. It is well suited for low-power mobile devices in wireless networks.

III. Mobile–Server communication Architecture

The authentication protocol must be able to create a secure channel between two principals on top of an insecure network, like the internet. The protocol must ensure the mutual authentication of both parties and the confidentiality and integrity of all the data transmitted through it before data get transmitted.

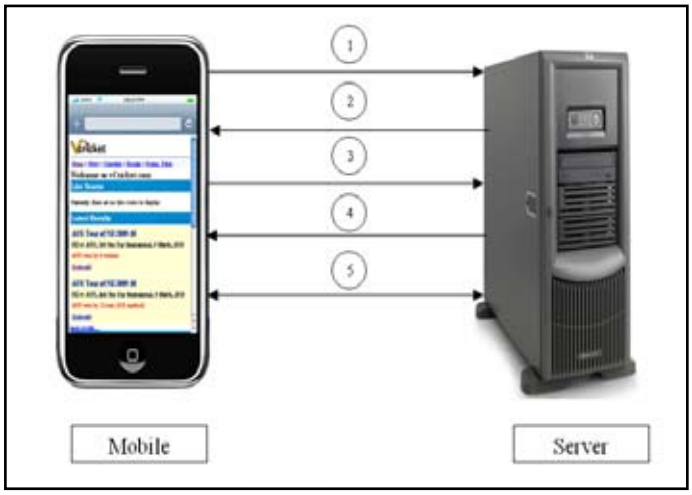

Fig. 3.1

The idea behind this protocol(Fig.3.1) is simple: in step 1, the mobile starts the protocol by sending its ID (e.g. Serial Number) to the server. In step 2, the server stores the mobile's ID for authentication purpose and generates mobile's private key and public key using point multiplication over ECC Public-key Cryptosystem. These keys (private and public key of the mobile) along with the public key of the server are sent to the mobile. Notice that the keys travel from the server into the mobile through a secure channel. To send the key to the respective destination, one can even adopt Diffie-Hellman key exchange algorithm.

In step 3, the mobile generates a challenge and sends it along with its ID to the server, encrypted with a combination of the server's public key and the mobile's private key. The server decrypts the message with mobile's public key and its private key and verifies if this ID matches the ID sent in step 1. This authenticates the client.

In step 4, the server sends the challenge received in the previous step plus one and a randomly generated session key and encrypted with a combination of mobile's public key and server's private key. The mobile then decrypts this message with server's public key and its private key and verifies the challenge. If it matches the one that was sent in step 3, then the mobile can trust that it's indeed talking to the right server. Both encryption and decryption process, specified in step 3 and 4 are done using point multiplication over ECC Public-key Cryptosystem technique. From now on, in step 5, a secure channel has been created and all data is encrypted with a session key. Notice that a new key is setup for each message to prevent replay attacks.

IV. Elliptic Curve Cryptography (ECC)

In ECC Algorithm (Elliptic Curve Cryptography) An elliptic curve is a plane curve defined by an equation of the form $Y^2=x^3+ax+b$Both parties agree to some publicly-known data items

Step 1 : select a suitable curve Ep(a,b)
Step 2 : select base point $G=(x_1,y_1)$ with large order n where nG=O(point of infinity)
Step 3 : A & B select private keys $n_A<n$, $n_B<n$
Step 4 : compute public keys: $P_A=n_AG$, $P_B=n_BG$
Step 5 : compute shared key: $K=n_AP_B$, $K=n_BP_A$
Encode and decode:
Step 1 : Encode any message M as a point on the elliptic curve Pm
Step 2 : Encrypt the message P_m : $C_m=\{kG, P_m+kP_B\}$, k random int number $1<k<p-1$
Step 3 : To decrypt, computes the product of the first point from Pm and his private key, n n_B * (kG)
Step 4 : Takes this product and subtracts it from the second point from $P_m(P_m + kP_B) – [n_B(kG)] = P_m + k(n_BG) – n_B(kG)= P_m$

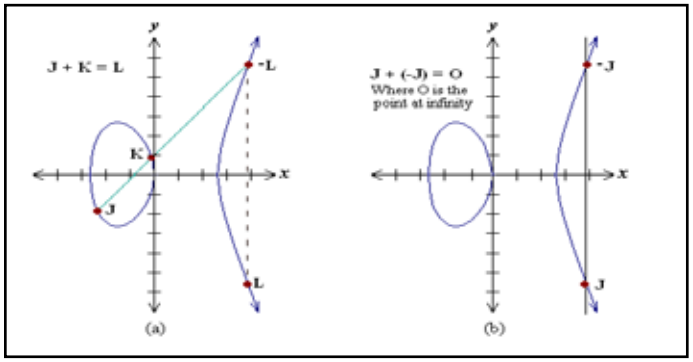

Fig. 4.1

IV. Finite Fields

To make operations on elliptic curve accurate and more efficient, the curve cryptography is defined over two finite fields.

- Prime field F_p
- Binary field F_{2^m}

The field is chosen with finitely large number of points suited for cryptographic operations.[18]

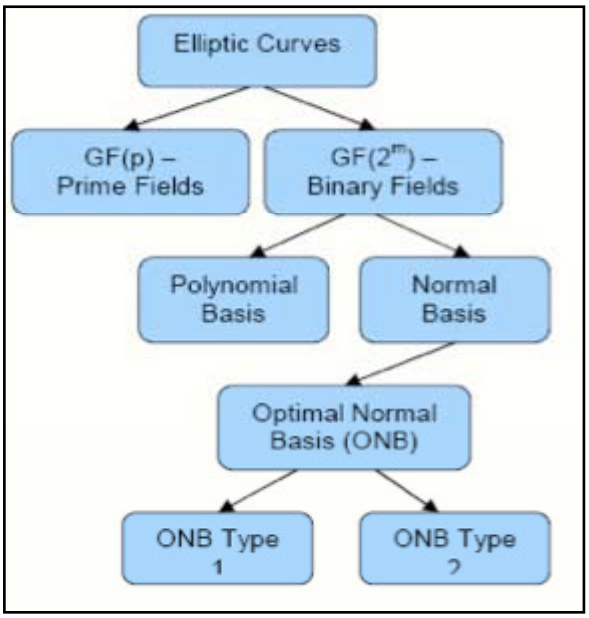

Fig. 4.1: Taxonomy of Elliptic Curves

A. EC On Prime Field

The equation of the elliptic curve on a prime field F_p is $y^2 \bmod p = x^3 + ax + b \bmod p$, where $4a^3 + 27b^2 \bmod p \neq 0$. Here the elements of the finite field are integers between 0 and p – 1. All the operations such as addition, subtraction, division, multiplication involves integers between 0 and p – 1. The prime number p is chosen such that there is finitely large number of points on the elliptic curve to make the cryptosystem secure.

1. Point Addition

Consider two distinct points J and K such that $J = (x_J, y_J)$ and $K = (x_k, y_k)$
Let L = J + K where $L = (x_L, y_L)$, then
$x_L = s^2 - x_J - y_J \bmod p$
$y_L = - y_J + s (x_J, - x_L) \bmod p$
$s = (y_J - y_K)/(x_J - x_K) \bmod p$, s is the slope of the line through J and K.

2. Point Subtraction

Consider two distinct points J and K such that $J = (x_J, y_J)$ and $K = (x_k, y_k)$
Then J - K = J + (-K) where $-K = (x_k, -y_k \bmod p)$
Point subtraction is used in certain implementation of point multiplication such as NAF (Non-Adjacent Form).

3. Point Doubling

Consider a point J such that $J = (x_J, y_J)$, where $y_J \pm 0$
Let L = 2J where $L = (x_L, y_L)$, Then
$x_L = s^2 - 2x_J \bmod p$
$y_L = -y_J + s(x_J - x_L) \bmod p$
$s = (3x_J^2 + a) / (2y_J) \bmod p$

B. EC ON BINARY FIELD F_{2^m}

The equation of the elliptic curve on a binary field F_{2^m} is $y^2 + xy = x^3 + ax + b$, where $b \neq 0$. Here the elements of the finite field are integers of length at most m bits. These numbers can be considered as a binary polynomial of degree m – 1. In binary polynomial the coefficients can only be 0 or 1. All the operation such as addition, subtraction, division, multiplication involves polynomials of degree m – 1 or lesser. The m is chosen such that there is finitely large number of points on the elliptic curve to make the cryptosystem secure.

C. Point Multiplication

In point multiplication a point P on the elliptic curve is multiplied with a scalar k using elliptic curve equation to obtain another point Q on the same elliptic curve i.e. KP=Q Point multiplication is achieved by two basic elliptic curve operations[18]

- Point addition, adding two points J and K to obtain another point L i.e., L = J + K.
- Point doubling, adding a point J to itself to obtain another point L i.e. L = 2J.

Here is a simple example of point multiplication. Let P be a point on an elliptic curve. Let k be a scalar that is multiplied with the point P to obtain another point Q on the cu rve. i.e. to find Q = kP. If k = 23 then kP = 23.P = 2(2(2(2P) + P) + P) + P. Thus point multiplication uses point addition and point doubling repeatedly to find the result. The above method is called 'double and add' method for point multiplication. There are other efficient methods for point multiplication such as NAF (Non – Adjacent Form). In the following sections we review some methods for computing scalar multiplication.

D. Binary Method

For computing kp, the simplest method is binary method[19]. The integer k is represented as
$k = k_{n-1} 2^{n-1} + k_{n-2} 2_{n-2} + \ldots\ldots + k1 + k0$
where ki € {0,1}, n = 0,1,2,……., n -1.
That is

$$k = \sum_{J=0}^{n-1} k_j 2^j \text{ ,where } k_j \in \{0,1\}.$$

This method is called binary method[8] which scans the bits of k either from left-to-right or right-to-left. The binary method for the computation of kP is given in the following algorithm
Algorithm: Binary method
Input: Binary representation of k and point P
$k = (k_{n-1}\ldots k_1 k_0)_2$
Output: kP
1. R ← P
2. For i = n-2 to 0 do
- R ←2R (Doubling)
- If ki =1 then
  R = R + P (Addition)
- i ← i −1

3. Return R

The cost of multiplication depends on the length of the binary representation of k and the number of 1s in this representation. If the representation $(k_{n-1}\ldots k_1 k_0)_2$ has $k_{n-1} \# 0$ then the number of doubling operation is (n - 1) and the number of addition operations is one less than the number of non-zero digits in $(k_{n-1}\ldots k_1 k_0)2$. The number of non-zero digits is called the Hamming weight of scalar representation. In an average, binary method requires n -1 doublings and n -1/ 2 additions.
For example, the integer k = 729 and the binary representation is (1011011001)2, computation of 729P requires 9 doublings and 5 additions. Whenever the bit is 1, two elliptic curve arithmetic operations such as ECDBL and ECADD will be made and if it is 0, only one operation, ECDBL is required. So if we reduce the number of 1s in the scalar representation or hamming weight, we could speed up the above computation.

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

V. System Implementation and Results

Table 5.1 shows the performance measurement of RSA algorithm for different Key size on command prompt execution. Figure 7.1 shows the comparative analysis of execution time of RSA algorithm for various Key sizes.

Table 5.1 Performance measurement of RSA Algorithm

RSA Key Size	Time in milliseconds
128	141
256	219
512	657
1024	4306

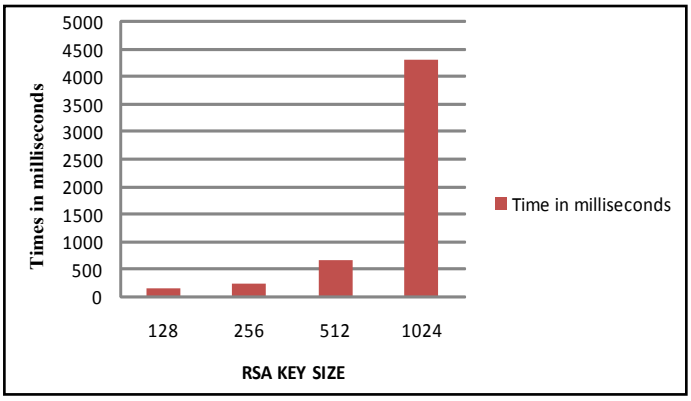

Fig 5.1: Comparative Analysis of Execution Time of RSA Algorithm

Table 7.2 shows the performance measurement of Elliptic Curve Cryptographic algorithm for different Key size on command prompt execution. Figure 7.2 shows the comparative analysis of execution time of Elliptic Curve Cryptographic algorithm for various Key sizes.

Table 5.2 Performance measurement of ECC

ECC KEY SIZE	Time in milliseconds
128	172
160	203
256	282
512	843
1024	5391

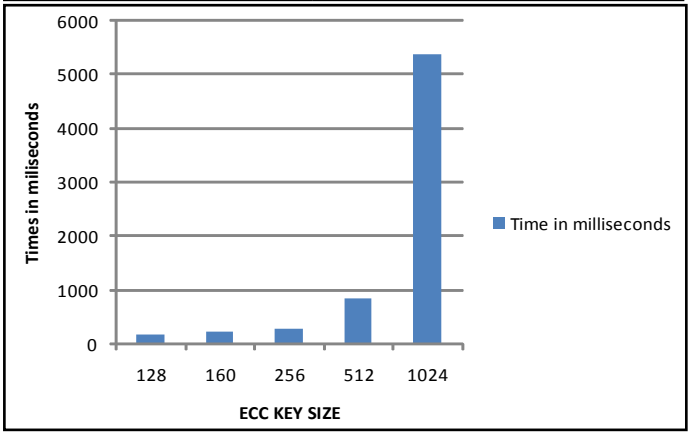

Fig 5.3: Comparative Analysis of execution time of ECC Algorithm

Table 5.4 shows the performance measurement of RSA algorithm for different Key size on J2ME wireless toolkit 2.5.1. Figure 5.5 shows the comparative analysis of execution time of RSA algorithm for various Key sizes.

Table 5.4 Performance measurement of RSA Algorithm in J2ME wireless toolkit 2.5.1

RSA Key Size	Time in milliseconds
128	250
256	578
512	2829
1024	20094

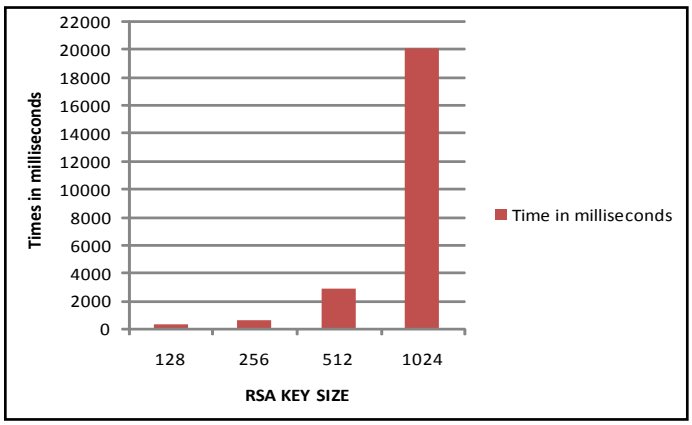

Fig. 5.5: Comparative analysis of execution time of RSA algorithm in J2ME wireless toolkit

Table 5.6 shows the performance measurement of Elliptivc curve cryptographic algorithm for different Key size on J2ME wireless toolkit 2.5.1. Figure 5.7 shows the comparative analysis of execution time of Elliptic curve cryptographic algorithm for various Key sizes.

Table 5.6 Performance measurement of Elliptic curve cryptographic algorithm in J2ME wireless toolkit 2.5.1

ECC KEY SIZE	Time in milliseconds
128	485
160	562
256	766
512	2125
1024	15203

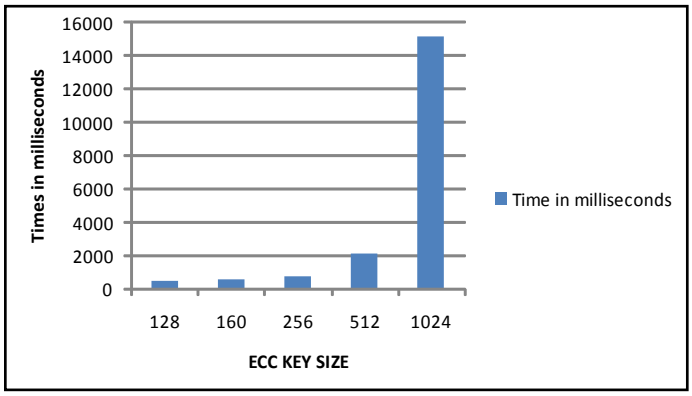

Fig. 5.7: Comparative analysis of execution time of Elliptic curve cryptographic algorithm in J2ME wireless toolkit 2.5.1

Table 5.8 shows the performance measurement of both RSA and Elliptic curve cryptographic algorithm for different Key size on Command prompt. Figure 5.9 shows the comparative analysis of execution time of both RSA and Elliptic curve cryptographic algorithm for various Key sizes.

Table 5.8 Performance measurement of both RSA and ECC algorithm

KEY SIZE	Time in milliseconds(RSA)	Time in milliseconds(ECC)
128	141	172
256	219	282
512	657	843
1024	4306	5391

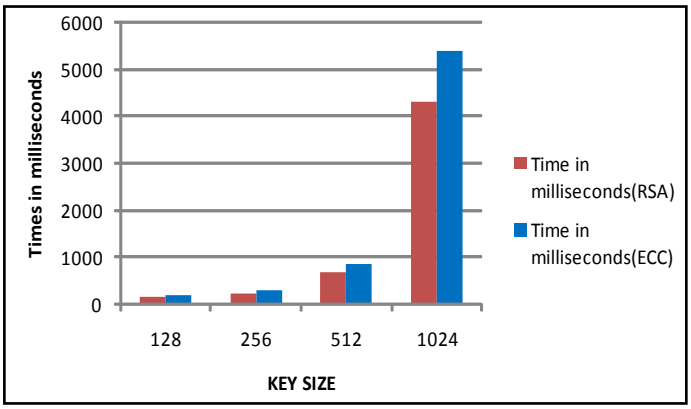

Fig. 5.9: Comparative analysis of execution time of both RSA and ECC algorithm

Table 7.6 shows the performance measurement of both RSA and Elliptivc curve cryptographic algorithm for different Key size on on J2ME wireless toolkit 2.5.1. Figure 5.10 shows the comparative analysis of execution time of both RSA and Elliptic curve cryptographic algorithm for various Key sizes.

Table 5.10 Performance measurement of both RSA and ECC in J2ME wireless toolkit 2.5.1.

KEY SIZE	Time in milliseconds(RSA)	Time in milliseconds(ECC)
128	250	485
256	578	766
512	2829	2125
1024	20094	15203

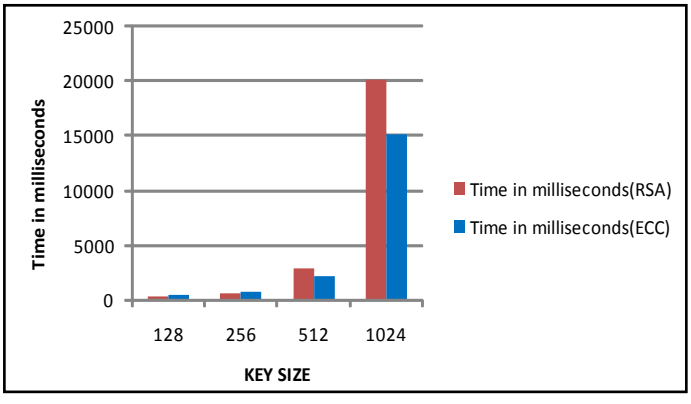

Fig. 5.11 Comparative analysis of execution time of RSA and ECC in J2ME wireless toolkit 2.5.1

The following table 5.12 gives approximate parameter sizes for comparable strength ECC and RSA.

Table 5.12 Comparison of ECC and RSA

Elliptic curve Cryptography	RSA
106 bits	512 bits
132 bits	768 bits
160 bits	1024 bits
224 bits	2048 bits

Table 5.13 Security Analysis of RSA-1024 and ECC160

Asymmetric Protocol (s)	Time (milliseconds)
RSA Key SIZE-1024	20094
ECC Key SIZE-160	562

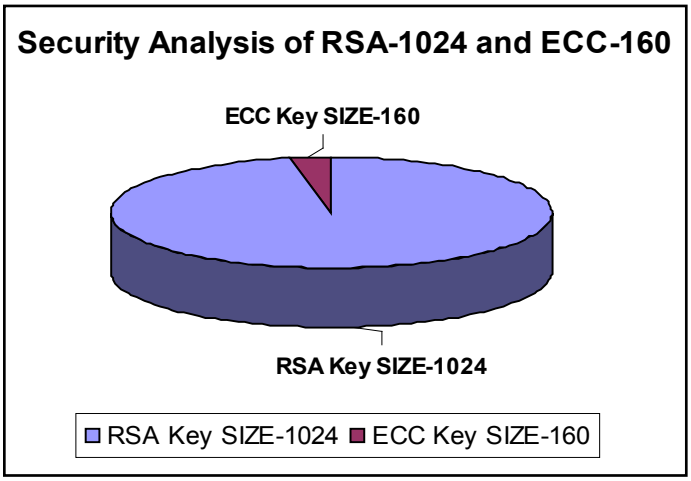

Fig. 5.14: shows the comparative analysis of performance measurements of both RSA and ECC.

Table 5.13 shows the performance measurement of RSA-1024, ECC-160 and Modified ECC-160 on command prompt execution. Figure 5.14 shows the Performance analysis of RSA-1024, ECC-160 and Modified ECC-160.

Table 5.15 Performance measurement of RSA-1024, ECC-160 and Modified ECC-160 in J2ME wireless toolkit 2.5.1.

Algorihms	Time in Milliseconds
RSA Key SIZE-1024	4306
ECC Key SIZE-160	218
Modified ECC Key SIZE-160	171

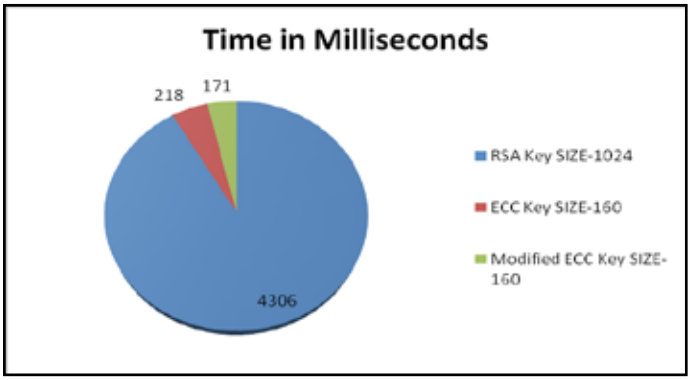

Fig. 5.16: Performance analysis of RSA-1024, ECC-160 and Modified ECC-160 in J2ME wireless toolkit 2.5.1.

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

Emulator Setup

Java Platform, Micro Edition, or Java ME, is a Java platform designed for mobile devices and embedded systems. This can be used to calculate the instantaneous time taken by the mobile while running. Target devices range from industrial controls to mobile phones and set-top boxes. Sun provides a reference implementation of the specification, but has tended not to provide free binary implementations of its Java ME runtime environment for mobile devices, rather relying on third parties to provide their own.

Java ME devices implement a profile. The most common of these are the Mobile Information Device Profile aimed at mobile devices, such as cell phones, and the Personal Profile aimed at consumer products and embedded devices like set-top boxes and PDA's. Profiles are subsets of configurations, of which there are currently two: the Connected Limited Device Configuration (CLDC) and the Connected Device Configuration (CDC).

Designed for mobile phones, the Mobile Information Device Profile includes a GUI API, and MIDP 2.0 includes a basic 2D gaming API. Applications written for this profile are called MIDLETS. Almost all new cell phones come with a MIDP implementation, and it is now the de facto standard for downloadable cell phone games. However, many cell phones can run only those MIDLETS that have been approved by the carrier.

Fig 7.9 & Fig 7.10 illustrates the instantaneous time taken to run those algorithms with J2ME emulator. From the figures, we observe that the proposed authentication algorithm takes minimum time than the existing authentication algorithms. From Fig. 7.9 & Fig.7.10. We observe that the ECC over point multiplication can be used for authenticated mobile communication

Fig.7.9

Fig.7.10

VII. Conclusion

Mobile handheld devices have strict constraints on the resources, such as memory space and time efficiency. It is very challenging authentication protocol by minimizing the running time while maintaining a desirable level of security. In this paper we propose an authentication protocol and calculate the time efficiency to run on mobile handheld devices.

From experimental results, we have the following conclusions:

1. Our proposed scheme is the first step in addressing challenges in terms of time using point multiplication method over ECC. Moreover, our system is more simple, secure and efficient.
2. We observe that point multiplication based ECC authentication protocol reduces the number of addition and doubling operations than conventional ECC protocol. Therefore, in future, This kind of fast and efficient method of authentication protocol can be used for secured mobile communication.

References

[1] *M. J. Beller, L. F. Chang, and Y. Yacobi, "Privacy and authentication on a portable communications system," IEEE J. Sel. Areas Commun., vol. 11, pp. 821-829, 1993.*

[2] *C. C. Lo and Y. J. Chen, "Secure communication mechanisms for GSM networks," IEEE Trans. Consum. Electron., vol. 45, pp. 1074-1080, 1999.*

[3] *T.-F. Lee, C.-C. Chang, and T. Hwang, "Private authentication techniques for the global mobility network," Wireless Personal Commun., vol. 35, no. 4, pp. 329-336, 2005.*

[4] *T.-F. Lee, S.-H. Chang, T. Hwang, and S.-K. Chong, "Enhanced Delegation-Based Authentication Protocol for PCSs," IEEE Trans. Wireless Commun., vol. 8, no. 5, pp. 2166-2171, 2009.*

[5] *H.-Y. Lin and L. Harn, "Authentication protocols with non-repudiation services in personnel communication systems," IEEE Commun. Lett., vol. 3, no. 8, pp. 236-238, 1999.*

[6] *H.-Y. Lin, "Security and authentication in PCS," Comput. Elect. Eng., vol. 25, no. 4, pp. 225-248, 1999.*

[7] *W.-B. Lee and C.-K. Yeh, "A new delegation-based authentication protocol for use in portable communication systems," IEEE Trans. Wireless Commun., vol. 4, no. 1, pp. 57-64, 2005.*

[8] *M. Rahnema, "Overview of the GSM system and protocol architecture,"IEEE Commun. Mag., pp. 92-100, 1993.*

[9] *Dr.R.Shanmugalakshmi, "Research Issues on Elliptic Curve Cryptography and Its applications - IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.6, June 2009, Pg.No :19 – 22.*

[10] *Marisa W. Paryasto , "Issues in Elliptic Curve Cryptography Implementation "- Internetworking Indonesia Journal, Volume 1/No. 1(2009), Pg.No 29-33.*

[11] *S. Prasanna Ganesan, Dr. GRD College of Science, "An Asymmetric Authentication Protocol for Mobile Devices Using Elliptic Curve Cryptography" 978-1-4244-5848-6/10/$26.00 © 2010 IEEE, Pg.No 107-109.*

[12] *Wendy Chou,Dr. Lawrence Washington,"Elliptic Curve Cryptography and Its Applications to Mobile Devices." Department of Mathematics University of Maryland, College Park.*

[13] *S. Prasanna Ganesan, Dr. GRD College of Science, "An Authentication Protocol For Mobile Devices Using Hyperelliptic Curve Cryptography" International Journal of Recent Trends in Engineering and Technology, Vol. 3, No. 2, May 2010.*

[14] *"A Performance Comparison of Data Encryption Algorithms," IEEE [Information and Communication Technologies, 2005. ICICT 2005. First International Conference ,2006-02-27, PP. 84- 89.*

[15] *Results of comparing tens of encryption algorithms using different settings- Crypto++ benchmark- . Retrieved October 1, 2008, from: http://www.eskimo.com/~weidai/benchmarks.html*

[16] S.Z.S. Idrus,S.A.Aljunid,S.M.Asi, "Performance Analysis of Encryption Algorithms Text Length Size on Web Browsers," IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.1, January 2008 ,Pg 20-25.

[17] Caimu Tang, Member, IEEE, and Dapeng Oliver Wu, Senior Member, IEEE" -An Efficient Mobile Authentication Scheme for wireless networks" IEEE transactions on wireless communications, vol. 7, NO. 4, APRIL 2008.

[18] Anoop MS, " Elliptic curve Cryptography", available at http://security.ittoolbox.com/research/elliptic-curvecryptography,5 Jan 2007

[19] J. Lopez, R. Dahab (2000), "An overview of elliptic curve cryptography", Technical report, IC-00-10, May 22. Available at http://www.dcc.unicamp.br/ic-main/public-cation - e.html.

[20]]Standard Specifications for Public Key Cryptography, IEEE Standard 1363, 2000.

Author Biographies

Sathish kumar received his B.E degree in Computer Science and Engineering from Madurai Kamaraj University,Tamilnadu, India, in 2002. He has received his M.E degree in Computer Science and Engineering from Anna University Trichy, Tamilnadu, India, in 2010. He is currently pursuing his Ph.D degree. His research interest includes cryptography and Network Security.

Sukumar received his B.E Degree in Electronics and Communication Engineering from Madurai Kamaraj University, Tamilnadu, India, in 1992. He has received is M.E degree in Computer Science and Engineering from Manonmaniam Sundaranar University, Tirunelveli,Tamilnadu, India,in 2005. He has received his Ph.D Degree from Anna University chennai,Tamilnadu, India,in 2010. His research interest includes cryptography and Network Security and he has published 5 papers in reputed international journals.

Karthiyayini M received her B.E degree in Electronics and Communication Engineering from Anna University Chennai in the year 2009. She has received her M.E degree in Communication Systems from Anna University Chennai, Tamilnadu, India in the year 2012. Her research interest includes Networks and Mobile Security.