International Journal of Engineering Science Invention
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org Volume 3 Issue 7ǁ July 2014 ǁ PP.28-39

A Brief Analysis on Key Management Schemes Using Elliptic Curve Cryptography in Wireless Sensor Network

Usham Robinchandra Singh[1], Sudipta Roy[1], Soram Ranbir Singh[2]
[1]Department of Information Technology, Assam University, Silchar, India.
[2]Department of Computer Science & Engineering, Manipur Institute of Technology, Imphal, India.
(A Constituent College of Manipur University, Imphal, India)

ABSTRACT : *The goal of this paper is to analyse efficient encryption schemes in wireless sensor networks and in devices with low computing power and resources. Embedded devices are also being used for information transfer and hence the need of network security is arising for these domain specific systems. Elliptic Curve Cryptography (ECC) has emerged as the most trusted solution for providing security on such systems. As these systems are classified to be resource constrained, the small key size of ECC makes it effective to implement on such systems. The good thing about ECC is that it can be faster than RSA and uses smaller keys, but still provides the same level of security. The security of ECC relies on the difficulty of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP). A comparative study of ECC with RSA is made in terms of key size, computational power and other factors.*

KEYWORDS: *ECC, Key Management, Public Key Cryptography, Encryption, WSN.*

I. INTRODUCTION

Wireless Sensor Networks play a very important role in the era of pervasive computing. These networks have various energy and computational constraints due to its ad-hoc nature of existence. The scale of deployments of wireless sensor networks requires careful decisions and trade-off among various security measures. Many security protocols designed for sensor networks tend to use symmetric key algorithms. However, constraints in sensor networks impose the need for high speed and less complex cryptographic algorithms which focus on localization, time synchronization and energy efficient routing. The vast majority of the products and standards that use public-key cryptography for encryption and digital signatures use RSA. As we know, the bit length for secure RSA use has increased over recent years, and this has put a very heavier processing burden on different applications using RSA [1]. Energy consumption is very high due to this heavy load processing and this burden has many difficulties, especially for electronic commerce sites that conduct large numbers of secure transactions. Recently, a competing system that has emerged is ECC [2, 3]. ECC was proposed in 1985 by Neal Koblitz and Victor Miller. Public-Key Cryptography (PKC) systems can be used to provide secure communications over insecure channels without exchanging a secret key. In public key cryptography, each user or the device taking part in the communication generally have a pair of keys, a public key and a private key, and a set of operations associated with the keys to do the cryptographic operations. Only the particular user knows the private key whereas the public key is distributed to all users taking part in the communication. Some public key algorithm may require a set of predefined constants to be known by all the devices taking part in the communication. ‘Domain parameters’ in ECC is an example of such constants. Public key cryptography, unlike private key cryptography, does not require any shared secret between the communicating parties but it is much slower than the private key cryptography The key distribution and storage problems, which are very common in private key cryptography is solved by the public key cryptography conception. Previous work shows public key algorithms are a good choice for use in wireless sensor networking. ECC with smaller keys and certificates will be significant in such systems. ECC can be used to achieve authentication and key management. Rest of the paper is organized as follows. We explore concepts of Wireless Sensor Networks in Section II. Cryptography with elliptic curves is briefly explained in Section III. Key Management mechanisms are explained in section IV. Security and elliptic curve discrete logarithm problem are discussed in section V and VI respectively.

II. WIRELESS SENSOR NETWORK

A WSN is a collection of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location [4]. The development of WSNs was mainly motivated by military applications such as battlefield surveillance but today such networks are used in many applications, such as industrial process monitoring and control, machine health monitoring, and so on. Normally, the WSN consists of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one or sometimes several sensors. Each node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the hops of the network can be routing or flooding. In computer science and telecommunications, wireless sensor networks are an active research area with numerous workshops and conferences arranged each year.

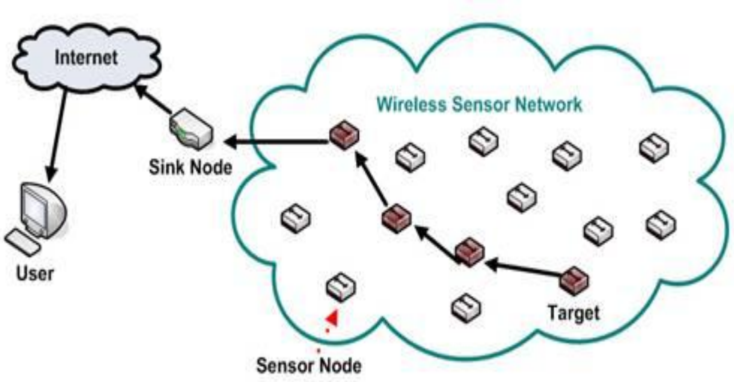

Fig. 1. A wireless sensor network[4]

In many applications, a WSN communicates with a Local Area Network or Wide Area Network through a gateway. The Gateway acts as a bridge between the WSN and the other network [4]. This enables data to be stored and processed by device with more resources, for example, in a remotely located server. The power restrictions of sensor nodes are raised due to their small physical size and lack of wires. Since the absence of wires results in lack of a constant power supply, not many power options exist. Sensor nodes are typically battery-driven. However, because a sensor network contains hundreds to thousands of nodes, and because often WSN are deployed in remote or hostile environments, it is difficult to replace or recharge batteries. The power is used for various operations in each node, such as running the sensors, processing the information gathered and data communication. Energy is the scarcest resource of WSN nodes, and it determines the lifetime of WSNs. WSNs are meant to be deployed in large numbers in various environments, including remote and hostile regions, where ad-hoc communications are a key component. Power limitations greatly affect security, since encryption algorithms introduce a communication overhead between the nodes; more messages must be exchanged, i.e. for key management purposes, but also messages become larger as authentication, initialization and encryption data must be included. Energy Consumption of the sensing device should be minimized and sensor nodes should be energy efficient since their limited energy resource determines their lifetime. In this context the use of ECC comes into play.

III. CRYPTOGRAPHY WITH ELLIPTIC CURVES

ECC is a public key cryptosystem like RSA but the security of it lies on the discrete logarithm problem over the points on an elliptic curve. The main attraction of ECC over RSA is that the best known algorithm for solving the underlying hard mathematical problem in ECC takes full exponential time. RSA take sub-exponential time. This means that significantly smaller parameters can be used in ECC than RSA, but with equivalent levels of security. A typical example of the size in bits of the keys used in different public key systems, with a comparable level of security(against known attacks), is that a 160-bit ECC key is equivalent to RSA with a modulus of 1024 bits. In practical terms, the performance of ECC depends mainly on the efficiency of finite field computations and fast algorithms for elliptic scalar multiplications. In addition to the numerous known algorithms for these computations, the performance of ECC can be increased by selecting particular underlying finite fields and or elliptic curves. For ECC, we are concerned with a restricted form of elliptic curve that is defined over a finite field. Strength of RSA [1] lies in integer factorization problem. That is when we are given a number n; we have to find its prime factors. It becomes quite complicated when dealing with large numbers and this is the strength of RSA.Elliptic Curves are a specific class of algebraic curves. The "Weierstrass form"of an elliptic curve E is the equation [2]:-

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

The constant $a_1,\ a_2,\ a_3,\ a_4,\ a_6$ and the variables x, y can be complex, real, integers, polynomials, or even any other field elements. So, the mathematics of elliptic curve cryptography is so deep and complicated. But in practice we must specify which field, F, these constants and the variables, x, y belong to and $\Delta \neq 0$, where Δ is the discriminant of E and is defined as follows:-

$$\Delta = -d_2^2d_8 - 8d_4^3 - 27d_6^2 + 9d_2d_4d_6$$
$$d_2 = a_1^2 + 4a_2$$
$$d_4 = 2a_4 + a_1a_3$$
$$d_6 = a_3^2 + 4a_6$$
$$d_8 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2$$

We say that E is defined over K when the coefficients $a_1,\ a_2,\ a_3,\ a_4,\ a_6$ (and of course, the variables x and y) of the equations come from the elements of the field K. So, we sometimes write $E(K)$ to indicate that E is defined over K, and K is called the underlying field. If E is defined over K, then E is also defined over any extension field of K.

Elliptic Curve over Galois Fields

We hardly use real numbers in cryptography as it is very difficult to store them precisely in computer and predict how much storage will be needed for them. This difficulty is solved by using Galois Fields. In a Galois field, the number of elements is finite. Since the number of elements is finite [5], we can find a unique representation for each of them, which allows us to store and handle the elements in an efficient way. Galois had shown that the number of elements in a Galois field is always a positive prime power, and is denoted by $GF(p^n)$. Two special Galois fields are very standard in Elliptic Curve Cryptography and they are $GF(p)$ when $n=1$ and $GF(2^n)$ when $p=2$.

1.1 Elliptic Curve over prime Galois Fields

An elliptic curve over a prime Galois Field uses a special elliptic curve of the form

$$y^2(\text{mod } p) = x^3 + ax + b(\text{mod } p)$$

where $a, b \in GF(p), 0 \leq x \leq p$ and $-16(4a^3 + 27b^2)\,\text{mod } p \neq 0.$ The constants ***a*** and ***b*** are non-negative integers smaller than the prime p. The condition that $-16(4a^3 + 27b^2)\,\text{mod } p \neq 0$ implies that the curve has no "singular points"[6].

Group Law

The mathematical property which makes elliptic curves very useful for cryptography is simply that if we take two (distinct) points on the curve, then the chord joining them intercepts the curve in a third point (because we have a cubic curve). If we then reflect that point in the x-axis we get another point on the curve (since the curve is symmetric about the x-axis). This is the "sum" of the first two points. Together with this addition operation, the set of points $E(K)$ forms an abelian group with **0** serving as its identity [7]. It is this group that is used in the construction of elliptic curve cryptographic systems. Algebraic formulae for the group law can be derived from the geometric description and they are reproduced here.

1.2.1 Group law for $y^2 = x^3 + ax + b$ over $GF(p)$.

(1) Identity: $P + 0 = 0 + P = P$ for all $P \in E(K)$.

(2) Negative: If $P = (x, y) \in E(K)$, then $(x, y) + (x, -y) = 0$. The point $(x, -y)$ is denoted by -P and is called the negative of P; note that -P is indeed a point in $E(K)$. Also, $-0 = 0$.

(3) Point addition: Let $P = (x_1, y_1) \in E(K)$ and $Q = (x_2, y_2) \in E(K)$ where $P \neq \pm Q$ .Then $P + Q = R(x_3, y_3)$, where $x_3 = \lambda^2 - x_1 - x_2$, $y_3 = \lambda(x_1 - x_3) - y_1$ and $\lambda = \dfrac{y_2 - y_1}{x_2 - x_1}$.

(4) Point doubling: Let $P = (x_1, y_1) \in E(K)$, where $P \neq \pm P$. Then $2P = R(x_3, y_3)$, where $x_3 = \lambda^2 - 2x_1$, $y_3 = \lambda(x_1 - x_3) - y_1$ and $\lambda = \dfrac{3x_1^2 + a}{2y_1}$.

Geometrical Interpretation of Group Law

1. Negative of a Point

Let's take a point $P = (x_1, y_1)$. The formula for finding $-P$ is $-P = (x_1, -y_1)$ as shown in the fig. 2.

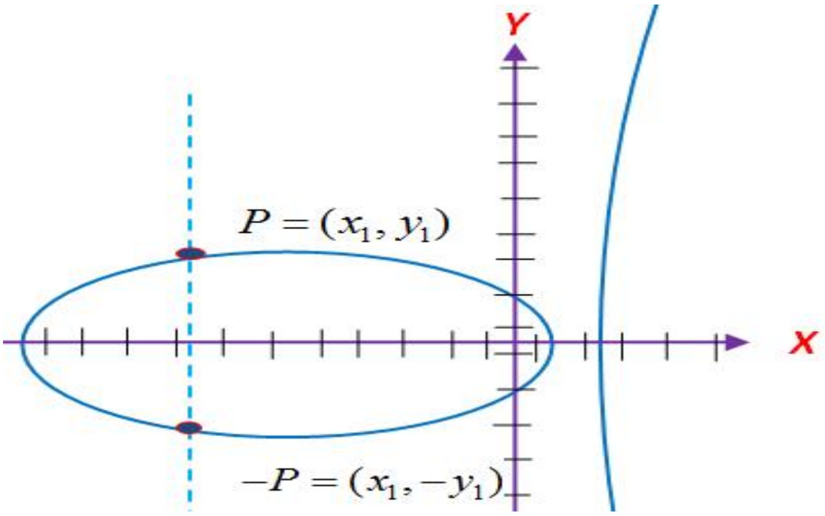

Fig. 2.Negative of a Point

Addition of two points

As mentioned before, we can define the addition of any two points on an elliptic curve by drawing a line between the two points and finding the point at which the line intersects the curve. The negative of the intersection point is defined as the "elliptic sum" of the two points by mathematicians and it is shown in fig. 3.

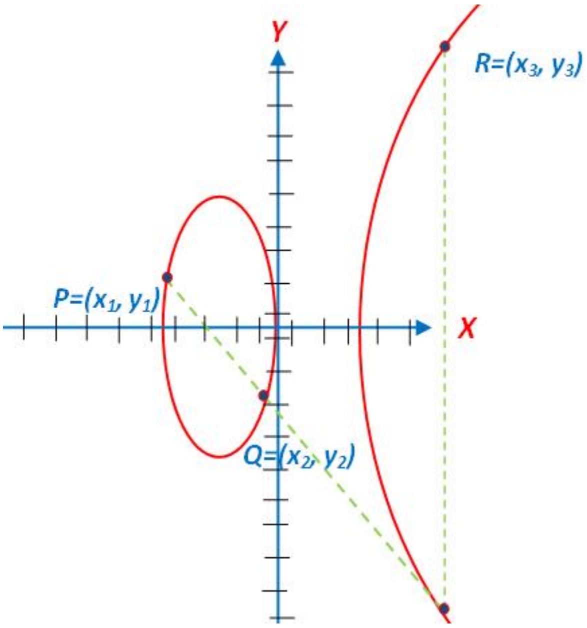

Fig. 3. Addition of two points

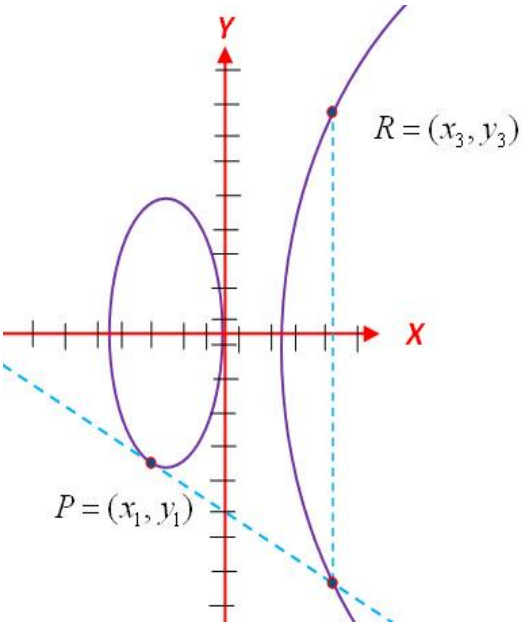

Fig. 4. Doubling a point

Mathematically we write:

$$R = P + Q.$$

This "addition" satisfies all the usual algebraic properties that we associate with integers, provided we define a single additional point "the point at infinity", which plays the role of 0 in the integers. In mathematical terms, we can define a finite additive abelian group on the points of the curve, with the zero being the point at infinity.

Doubling of a point

If $P=(x_1, y_1)$, then the double of *P*, denoted by, $R=(x_3, y_3)$, is defined as follows. First draw the tangent line to the elliptic curve at *P*. This line intersects the elliptic curve in a second point. Then *R* is the reflection of this point in the x –axis. This is depicted in fig. 4. We can extend this idea to define $P+P+P=3P,$ and extending this idea further, we can define $P+P+P+...+k$ times $=kP$, for any integer *k*, and hence define the order of P, being the smallest integer k such that $kP=0$, where 0 denotes the point at infinity[7]. Fig. 5 shows some multiples of $P=(-1,-2)$ on the curve $y^2 = x^3 - 5x.$

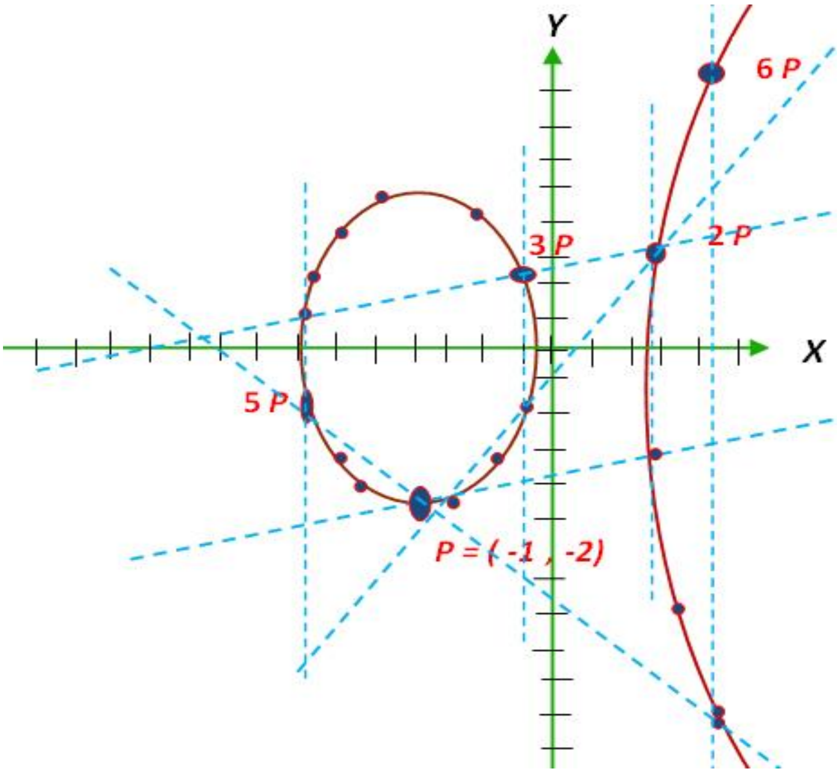

Fig. 5. Some multiples of $P = (-1, -2)$.

To elucidate doubling of a point, consider the elliptic curve

$$y^2 = x^3 + x + 4(\text{mod } 23)$$

defined over $GF(23)$. This curve is represented by $E_{23}(1,4)$. We first note that $4a^3 + 27b^2 = 4 + 432 = 436 \equiv 22(\text{mod } 23) \neq 0(\text{mod } 23)$. The points in $E_{23}(1,4)$ are the following:-

Table 1. Points on the curve $E_{23}(1,4)$

0	*(0,2)*	*(0,21)*	*(1,11)*	*(1,12)*	*(4,7)*	*(4,16)*	*(7,3)*
(7,20)	*(8,8)*	*(8,15)*	*(9,11)*	*(9,12)*	*(10,5)*	*(10,18)*	*(11,9)*
(11,14)	*(13,11)*	*(13,12)*	*(14,5)*	*(14,18)*	*(15,6)*	*(15,17)*	*(17,9)*
(17,14)	*(18,9)*	*(18,14)*	*(22,5)*	*(22,19)*	--	--	--

Let $P = (4,7)$ and $Q = (13,11)$. Then $P + Q = R(x_3, y_3)$ is computed as follows-

$$\lambda = \frac{11-7}{13-4} = \frac{4}{9} = 4\text{X}9^{-1}(\text{ mod } 23) = 4\text{X}18(\text{ mod } 23) = 72 \text{ mod } 23 = 3$$

$$x_3 = 3^2 - 4 - 13 = -8 \equiv 15\,(\text{mod}\,23), \quad \text{and } y_3 = 3(4-15) - 7 = -40 \equiv 6\,(\text{mod}\,23)$$

Hence, $R = (15,6)$.

Again, let $P=(4,7)$. Then $2P=P+P$ is calculated as follows:-

$$\lambda=\left(\frac{3\text{X}4^2+1}{14}\right)=49\text{X}14^{-1}=49\text{X}5=245\,(\bmod 23)=15$$

$x_3=15^2-8=217\equiv 10(\text{mod}\,23)$ and $y_3=15(4-10)=-97\equiv 18(\text{mod}\,23)$.

Hence, $2P=(10,18)$.

2. Elliptic Curve over prime Galois Fields

Let's look at elliptic curves over $GF(2^n)$. Mathematicians say that we cannot use the simplified version of equation, which we used for integer numbers, in our elliptic curve equations over prime Galois fields. They tell us that we need to use either this version:

$$y^2+xy=x^3+ax^2+b \qquad (1)$$

or this version

$$y^2+y=x^3+ax+b \qquad (2)$$

But, the mathematicians say that the second form above, (2), has the advantage that it can be very quickly computed and has some very special properties. These special properties make such curves unsuitable in cryptography.

The curves of equation (1) are excellent for cryptographic applications. We must be careful in choosing the coefficients to get maximum security benefits. Experts argue that a poor choice will create a curve that is easier for the hackers to attack. For equation (1) to be valid, *b* must never be 0. However, *a* can be 0. Here we give the group laws of the first form of the curve [6].

2.1 Group law for $y^2+xy=x^3+ax^2+b$ over $GF(2^n)$

1. Identity: $P+0=0+P=P$ for all $P\in E.$
2. Negative: If $P=(x,y)\in E,$ then $(x,y)+(x,x+y)=0.$ The point $(x,x+y)$ is denoted by -P and is called the negative of P; note that -P is indeed a point in E. Also, $-0=0.$
3. Point addition: Let $P=(x_1,y_1)\in E$ and $Q=(x_2,y_2)\in E$ where $P\neq\pm Q$ .Then $P+Q=R(x_3,y_3)$, where $x_3=\lambda^2+\lambda+x_1+x_2+a$ and $y_3=\lambda(x_1+x_3)+x_3+y_1$ with $\lambda=\dfrac{y_2+y_1}{x_2+x_1}$.
4. Point doubling: Let $P=(x_1,y_1)\in E,$ where $P\neq -P$. Then $2P=R=(x_3,y_3),$ where $x_3=\lambda^2+\lambda+a$ and $y_3=x_1^2+\lambda x_3+x_3$ with $\lambda=x_1+\dfrac{y_1}{x_1}$.

To explain the mathematics behind the group law, let us take an elliptic curve, $y^2+xy=x^3+g^3x^2+1$ over $GF(2^3)$ under the irreducible polynomial $f(x)=x^3+x+1.$ Here the generator, g, satisfies the relation $g^3 + g + 1 = 0$ or $g^3 = g + 1$ as the arithmetic is over $GF(2)$. The following table 2 shows the values of $g's$ and the points on the curve are given in table 3.

Table 2: Possible values of g's

0	*1*	*g*	g^2	$g^3 = g + 1$	$g^4 = g^2 + g$	$g^5 = g^2 + g + 1$	$g^6 = g^2 + 1$
000	*001*	*010*	*100*	*011*	*110*	*111*	*101*

Table 3: Points on the given curve

0	$(0,1)$	$(g^2,1)$	(g^2,g^6)	(g^3,g^2)
(g^3,g^5)	$(g^5,1)$	(g^5,g^4)	(g^6,g)	(g^6,g^5)

Let $P = (0,1)$ and $Q = (g^2,1)$. We have $P + Q = R = (x_3, y_3)$ and it is computed as follows.

$$\lambda = \frac{1+1}{g^2+0} = 0$$

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a = 0 + 0 + 0 + g^2 + g^3 = g^5 \text{ and } y_3 = \lambda(x_1 + x_3) + x_3 + y_1 = 0(0 + g^5) + g^5 + 1$$

$$= g^5 + 1 = g^2 + g = g^4.$$

So, $R = (g^5, g^4) = (111,110).$

Again take $P = (g^2,1)$. $2P = P + P = R(x_3, y_3)$.

$$\lambda = g^2 + \frac{1}{g^2} = g^2 + g^5 = g + 1 = g^3$$

$$x_3 = \lambda^2 + \lambda + a = g^6 + g^3 + g^3 = g^6 \text{ and } y_3 = x_1^2 + \lambda x_3 + x_3$$

$$= g^4 + g^9 + g^6 = g^4 + g^2 + (g^2 + 1)$$

$$= g^4 + 1 = (g^2 + g) + 1 = g^5$$

Therefore, $R = (x_3, y_3) = (g^6, g^5) = (101,111).$

Security of ECC

Let E be an elliptic curve defined over a finite field and let, P be a point (called base point) on E of order n and k is a scalar. Calculating the point $Q = kP$ from P is very easy and $Q = kP$ can be computed by repeated point additions of P. However, it is very hard to determine the value of k knowing the two points: kP and P. This lead leads to the definition of Elliptic Curve Logarithm Problem (ECDLP) [6], which is defined as: "Given a base point P and the point $Q = kP$, lying on the curve, find the value of scalar k". The integer k is called the elliptic curve discrete logarithm of Q to the base P, denoted as $k = \log_P Q$.

IV. KEY MANAGEMENT MECHANISM

Key Management is very much essential for secure communication either in case of symmetric key or asymmetric key algorithms. For the implementation of various security schemes, key distribution is not typical in WSNs, but constraints such as small memory capacity makes centralized key distribution techniques impossible. Straight pair wise key sharing between every two nodes in a network is not suitable for large growing networks. A security scheme in WSNs must use efficient and reliable key distribution for secure communication between all relevant nodes. Various cryptographic solutions based on symmetric and asymmetric algorithms have been proposed. Symmetric algorithms provide confidentiality while fulfilling the power, space and memory requirements of WSN [8]. However they fail to provides authenticity and proper key exchange mechanisms which is achieved through public key cryptography. In symmetric key cryptography data are encrypted and decrypted with a single shared key so it has key exchange problem. Secure key distribution of keys securely to communicating hosts is a significant problem since pre-distributing the keys is not always possible. Asymmetric cryptosystems were not considered as an option for constrained devices due to their extensive mathematical calculations. These calculations require large amount of space and power. An energy efficient key management scheme for WSN using ECC can be designed. A typical WSN can be assumed as a combination of both large number of normal sensor nodes also known as cluster heads and small number of special nodes. Cluster nodes have more computational power than that of special nodes. Cluster nodes can be made according to energy carrying capacity of sensor nodes evolved in the WSN. Before the pre-distribution of the sensor nodes, a server based on ECC can be used to generate both public/private key pair. Due to openness of wireless sensor networks, secure communication between nodes is one of the necessary works in security arrangement. There are many encryptions way in the public key encryption system, the reason why choosing ECC is that under the same working strength request, ECC needs a very shorter key length. At the same time, ECC also has certain superiorities in the computation load, the operating speed and the spatial consumption aspect. Table 4 compares ECC and RSA in term of key length still providing the same security level. Table 5 shows the energy consumption rate [9, 10] of ECC with that of RSA.

Table 4. Comparison of key length in bit of Key of ECC and RSA

RSA	ECC	ECC: RSA
512	112	1:5
1024	160	1:6
2048	224	1:9
3072	256	1:12
7680	384	1:20
15360	512	1:30

Key Exchange

Key exchange can be done in the following manner. A large integer $q = 2^n$ is picked and elliptic curve parameters a and b. This defines an elliptic curve group of points. Now, choose a base point $G = (x_1, y_1)$ in $E(a,b)$ whose order is a very large value n. The elliptic curve E and G are the parameters known to all participants. A key exchange between users A and B can be accomplished as follows:

Table 5: Comparison of energy consumption rate of ECC and RSA

Algorithm	Signature		Key exchange	
	Signature	Verification	Client	Server
RSA-1024	304	11.9	15.4	304
ECDSA-160	22.82	40.093	22.3	22.3
RSA-2048	2302.7	53.7	57.2	2302.7
ECDSA-224	61.54	121.983	60.4	60.4

1. A selects an integer n_A less than n. This is A's private key. A then generates a public key $P_A = n_A G$; the public key is a point on E.
2. B similarly selects a private key n_B and computes a public key $P_B = n_B G.$
3. A generates the secret key $K = n_A P_B$ and B generates the secret key $K = n_B P_A.$

The calculations in step 3 produce the same result. $K = n_A P_B = n_A(n_B G) = n_B(n_A G) = n_B P_A.$
To break this scheme, an attacker would need to be able to compute k given G and kG, which is assumed to be hard.

Encryption using ECC

The plaintext message m is taken as input in the form of bits of varying length. This message m is encoded and is sent in the cryptographic system as x-y point P_m. This point is encrypted as cipher text and subsequently decrypted. As with the key exchange system, an encryption and decryption system requires a point G and an elliptic group $E(a,b)$ as parameters. User A selects a private key n_A and generates a public key $P_A = n_A G$. Similarly, user B selects a private key n_B and generates a public key $P_B = n_B G$. To encrypt and send a message P_m to B, A chooses a random positive integer k and produces the cipher text C_m consisting of pair of points $C_m = \{kG, P_m + kP_B\}.$

Decryption using ECC

To decrypt the cipher text, B multiples the first point in the pair by B's private key n_B and subtracts the result from the second point as shown by equation.

$$P_m + kP_B - n_B(kK) = P_m + k(n_B G) - n_B kG = P_m.$$

V. SECURITY IN WIRELESS SENSOR NETWORKS

The main advantage ECC has over RSA is that the basic operation in ECC is point addition which is known to be computationally very expensive.

Table 6: Comparison of strength of RSA and ECC in breaking the system.

Time to break (in MIPS-years)	RSA key size(in bits)	ECC key size(in bits)
10^4	518	106
10^8	768	132
10^{11}	1024	160
10^{20}	2048	210
10^{78}	21000	600

This is one of the reasons why it is very unlikely that a general sub-exponential attack on ECC will be discovered in the near future, though ECC has a few attacks on a few particular classes of curves. These curves can be readily distinguished and can be avoided. On the other hand, RSA already has a known sub-exponential attack which works in general. We compare the performance of ECC with RSA in terms of key sizes for the same level of security, data sizes, encrypted message sizes, and computational power. RSA takes sub-exponential time and ECC takes full exponential time. For example, RSA with key size of 1024 bits takes $3x10^{11}$ MIP years with best known attack where as ECC with 160 bit key size takes $9.6x10^{11}$ MIP years. Therefore, ECC offers same level of security with smaller key sizes. Data size for RSA is smaller than ECC. Encrypted message is a function of key size and data size for both RSA and ECC. Since ECC key size is relatively smaller than RSA key size, encrypted message in ECC is smaller. As a result, computational power is smaller for ECC. Thus, to maintain the same degree of security in view of rising computing power, the number of bits required in the RSA generated key pair will rise much faster than in the ECC generated key pair.
Menezes and Jurisic**,** in their paper [11], said that to achieve reasonable security, a 1024-bit modulus would have to be used in a RSA system, while 160-bit modulus should be sufficient for ECC. Most attacks on ECC are based on attacks on similar discrete logarithm problems, but these work out to be much slower due to the added complexity of point addition. Also, methods to avoid each of the attacks have already been designed [12].The one thing working against ECC is that though elliptic curves have been a well-researched field, its cryptographic applications have been noticed only recently. This is the only advantage that RSA has over ECC. RSA has been well-researched and has been the topic of many seminal theses. In fact, the cryptographic use for elliptic curves was only discovered in the process of finding out new attacks on the RSA system [13].ECC and some related work about wireless communication that is based on elliptic curve cryptographic techniques. Presently, RSA algorithm demands a key length be not less than 1024bits for long term security and we know that ECC with only a160 bits modulus offers the same level of security as RSA with 1024-bit modulus.

Table 7: Strength of Diffie Hellman vs Elliptic Curve Keys

Security Level(bits)	Ratio of DH key: ECC key
80	3:1
112	6:1
128	10:1
192	32:1
256	64:1

Thus, using ECC in wireless communication system is extremely recommended. In short functional requirement of even such basic electronic gadgets are increasing, into the requirement of more comprehensive software Development platforms. This has resulted into the introduction of embedded operating systems and compilers of various high level languages for embedded systems. The progress is almost on the similar lines how computer systems have evolved into various layers of hardware, operating systems and application programs, which later got clubbed with communication networks. Table 7 shows a comparison of elliptic curve keys with Diffie Hellman Keys [14].

VI. Elliptic Curve Discrete Logarithm Problem

The fastest known technique for breaking the Elliptic Curve Discrete Logarithm is known as the Pollard rho method. Table 8 compares the efficiency of this method with factoring a number into two primes using the GNFS. From Table 8, it can be inferred that a considerably smaller key size can be used for ECC compared to RSA. Furthermore, for equal key lengths, the computational effort required for ECC and RSA is comparable. Thus, there is a Computational advantage by using ECC with a shorter key length when compared to the secure RSA scheme. This work focuses on the performance advantages that can be obtained by using ECC in a wireless environment. ECC over prime fields is implemented for obtaining better performance characteristics in securing SSL. The algorithm for ECC over binary fields is further speeded up by using the Ring representation technique. This algorithm with lesser complexity and higher speed is implemented for sensor networks taking its constraints into account and is found to the ideal in a wireless sensor network environment.

Table 8 Comparision of ECDLP and IFP

ECDLP Using the Pollard rho method		IFP Using General Number Field Sieve	
Key size	MIPS-Years	Key size	MIPS-Years
150	3.8×10^{10}	512	3×10^{4}
205	7.1×10^{18}	768	2×10^{8}
234	1.6×10^{28}	1024	3×10^{11}

VII. CONCLUSION

Wireless sensor networks are devices with low computing power and resources. Elliptic Curve Cryptography (ECC) fits well in such systems. The security of Elliptic Curve Cryptosystem depends on how difficult it is to determine *k* given kP and $P.$ This is referred to as the elliptic curve logarithm problem. The fastest known technique for taking the elliptic curve logarithm is known as the Pollard rho method. It has been seen that a considerably smaller key size can be used for ECC compared to RSA. Thus, there is a computational advantage to using ECC with a shorter key length than a comparably secure RSA. The results show that ECC is efficient in terms of the size of Data files and Encrypted files. The above information is useful for wireless communication due to low data rate transmission and for constrained devices because of low power requirements.

REFERENCES

[1] R.L. Rivest, A. Shamir, and L.M. Adleman, "A Method for Obtaining Digital Signatures and PublickeyCryptosystems ", Communications of the ACM, Volume 21, pages 120-126, February 1978.

[2] N. Koblitz, " Elliptic Curve Cryptosystems ",Mathematics of Computation., Number 48,pages 203-209,1987.

[3] V.S. Miller," Use of Elliptic Curves in Cryptography ",Advances in Cryptology- Proceedings of CRYPTO'85, Springer Verlag Lecture Notes in Computer Science 218, pages 417-426, 1986.

[4] Wireless Sensor Network. [Online] http://en.wikipedia.org/wiki/Wireless_sensor_network. Accessed on June 21st , 2014.

[5] Bhattacharya, Jain, Nagpaul, Basic Abstract Algebra, Cambridge University Press, 2002.

[6] Lawrence C. Washington, Elliptic Curves, Number Theory and Cryptography, CRC Press, 2008.

[7] Ian Blake, Gadiel Seroussi, Higel Smart, Advances in Elliptic Curve Cryptography, Cambridge University Press, 2005.

[8] GuoXiaowang, Zhu Jianyong, Research on Security Issues in Wireless sensor networks,"2011.

[9] An Liu, and Peng Ning. Tiny ECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks. In Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN 2008). SPOTS Track, April 2008.

[10] Cliff Wang. An Liu, Peng Ning, "Cluster-Based Minimum Mean Square Estimation for Secure and Resilient Localization in Wireless Sensor Networks," in Proceedings of the International Conferences on Wireless Algorithms, Systems and Applications(WASA' 07), August 2007.

[11] Aleksandar Jurisic and Alfred J. Menezes. Elliptic curves and cryptography. Dr. Dobb's Journal, 1997.

[12] Henna Pietil¨ainen. Elliptic curve cryptography on smart cards. 30 October 2000.

[13] H. W. Lenstra. Factoring integers with elliptic curves. Annals of Mathematics, 126:649–673, 1987.

[14] The case of Elliptic Curve Cryptography – NSA/CSS [Online] Available: http://www.nsa.gov/business/programs/elliptic_curve.shtml, January 2009.