Secure key Pre-distribution and Mutual Node Authentication Protocol in WSN using ECC

Asha Rani Mishra
MDU, Rohtak
BSAITM, Faridabad

Anju Gera
MDU, Rohtak
BSAITM, Faridabad

Bhawna Chauhan
MDU, Rohtak
BSAITM, Faridabad

ABSTRACT
Wireless Sensor Networks (WSNs) has foreseen big changes in data gathering, processing and disseminating for monitoring specific applications such as emergency services, disaster management, and military applications etc. Since large number sensor nodes are deployed to monitor a vast field, where the operational conditions are most often harsh or even hostile so security mechanism in WSN is a greater challenge in WSN. Sensor networks pose unique challenges because of their inherent limitations in communication and computing. Sensor networks are vulnerable to security attacks due to the broadcast nature of transmission. The threats against WSN can be reduced by proper security measures. One of the commonly used methods is the use of cryptographic algorithm. This paper proposes mechanisms for Key Predistribution and mutual Authentication protocol in sensor networks using ECC with respect to constraints of WSN.

General Terms
WSN, ECC, authentication, security issues in WSN.

Keywords
Public Key Cryptography, ECC, WSN, Attacks, Key management, authentication

1.INTRODUCTION
Wireless sensor network consists of a large number of sensor nodes that are able to collect and disseminate data in areas where ordinary networks are unsuitable for environmental and/or strategic reasons. [1] The potential of WSN is emerging with each passing days and hence its use in various fields. Security in wireless sensor networks (WSNs) is an upcoming research field since it is quite different from traditional network security mechanisms. Since not only the resource restriction in conventional wireless networks but security critical applications, security functions are very important issues in WSN. The deployment nature of sensor nodes makes them vulnerable to various threats and hence attacks. They are susceptible to a variety of attacks, including node capture, physical tampering, and denial of service, prompting a range of fundamental research challenges.The threats against WSN can be reduced by proper security measures. To address the critical issues in WSN we use cryptography, secure key management, secure routing protocol, secure data aggregation and intrusion detection. Various security issues can be reduced by using symmetric or asymmetric algorithm. Symmetric algorithms provide only confidentiality. Despite many research efforts, the problems of key distribution and authentication are still open and require new cryptographic solutions. These new security mechanisms have to take advantage of specific sensor network features and meet the strict limitations of WSN hardware platforms. ECC got the attention of researchers due to the possibilities of practical implementation in resource constrained devices. The cryptosystem based on elliptic curve cryptography (ECC) is becoming the recent trend of public key cryptography. ECC can be used to achieve authentication and key management in WSN in a better way keeping all the constraints of wireless sensor network. In next section we will discuss the WSN architecture and characteristics. Section III and IV deal with security requirements and key management issues in WSNs, respectively. Section V will present Elliptic Curve Cryptography. Section VI concludes the paper and proposes some future work. Section VI and VII discusses suitability of ECC in WSNs and key exchange mechanism using ECC.

2. LIMITATIONS AND CHALLENGES OF WSN
There are some limitations for WSN implementation. The sensor nodes have the various limitations such as low battery power, minimum computation capability. Other than energy constraints a major challenge in WSN is to incorporate security mechanisms. Wireless sensor networks are clearly a very challenging environment for applying security services. They differ in many aspects from traditional fixed, networks and standard cryptographic solutions cannot be used in this application space Due to the inherent limitations of sensor nodes like llimited area, nature of links, limited processing, power and memory of WSNs leads to strict constraints on the selection of cryptographic techniques. Traditional security mechanisms with high overhead are not feasible for resource constrained sensor nodes. Other than these factors broadcast nature of wireless communication, frequently changing topology, larger number of interacting nodes, and nature of deployment of sensor nodes makes enabling security in WSNs a different approach. In short vvarious constraints of WSN are: Energy constraints, memory limitations, unreliable communication, and higher latency in communication.

3. CRYPTOGRAPHY IN WSN
Until recently, security solutions for WSNs relied on symmetric encryption algorithms (e.g. RC5) to provide properties such as authentication and confidentiality since, due to their resource constraints, nodes cannot afford to use conventional algorithms of Public Key Cryptography (PKC), e.g. RSA/DSA. Although more efficient than PKC, symmetric cryptosystems have some drawbacks. Firstly, nodes face the key agreement problem, i.e., they must decide on a shared key to communicate securely. This problem is even worse in WSNs due to the open and unattended environments where nodes are commonly deployed. Further, the ideal level of security in these cryptosystems is achieved by using pair wise

keys. However, this scheme is not scalable and thus is inadequate for WSNs which may comprise thousands of nodes. Finally, symmetric cryptosystems do not provide non repudiation. To address these drawbacks, a number of key predistribution schemes have been proposed although effective in trying to achieve a good trade-off between resource consumption and resiliency, these proposals eventually incur some degree of overhead. LEAP assumes that a predistributed key shared among all nodes will not be disclosed during the initial time units of the network operation. Secondly, LEAP assumes that once this key is erased, it cannot be recovered from memory. However, this is not always the case. Lastly, LEAP does not provide digital authentication and repudiation of messages is still possible. Because of all these reasons more efficient method of PKC has to be sought. Elliptic Curve Cryptography (ECC) PKC is indeed feasible in WSNs since ECC consumes considerably less resources than conventional PKC, for a given security level. However, in order to use effectively ECC in WSNs, it is first necessary to enable authentication of public keys. Otherwise, the network shall be vulnerable to man-in the middle attack. These operations, in turn, incur high overheads of storage, communication, and computation and, as a result, are inadequate for WSNs. Identity-Based Encryption (IBE) is an exception where an information that uniquely identifies users (e.g. or email addresses) can be used to both exchange keys and encrypt data, and thus PKI is unnecessary.

4. PUBLIC KEY CRYPTOGRAPHY AND SECURITY IN WSN

Most of the WSN security solutions that have been proposed so far are very restricted and rely only on simple symmetric primitives. Many of them reduce the computational cost, but are not scalable and tend to dramatically increase the communication overhead. They also do not provide a good trade-off between resilience and storage of cryptographic keys. Symmetric key solutions are prone to physical attacks and they do not work when a small number of nodes are captured by attackers. The security of such systems is very limited when compared to Public Key Cryptography solutions. Symmetric key techniques proposed in the literature usually defend against a particular security threat making the sensor network vulnerable to a range of dangerous attacks the need for key pre-distribution, pair-wise key sharing or complicated one-way., Public Key Cryptography provides a unified framework offers a more flexible and simple interface, The functionality of Public Key Cryptography schemes is highly desirable in WSNs to enhance the security level of demanding applications.

5. ELLIPTIC CURVE CRYPTOGRAPHY

Elliptic Curve Cryptography (ECC) is a public key cryptography. where 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. Public-Key cryptography (PKC) systems can be used to provide secure communications over insecure channels without exchanging a secret key. The most popular public-key cryptography systems nowadays are RSA and Elliptic Curve Cryptography (ECC). Elliptic curve cryptography (ECC) was proposed in 1985 by Neal Koblitz and Victor Miller. The mathematical operations of ECC is defined over the elliptic curve $\mathbf{y2 = x^3 + ax + b}$, where $\mathbf{4a^3 + 27b^2 \neq 0}$. Each value of the 'a' and 'b' gives a different elliptic curve. All points (x, y) which satisfies the above equation plus a point at infinity lies on the elliptic curve. The public key is a point in the curve and the private key is a random number [6]. The public key is obtained by multiplying the private key with the generator point G in the curve. The generator point G, the curve parameters 'a' and 'b', together with few more constants constitutes the domain parameter of ECC).The ECC curve is shown in Figure 1.

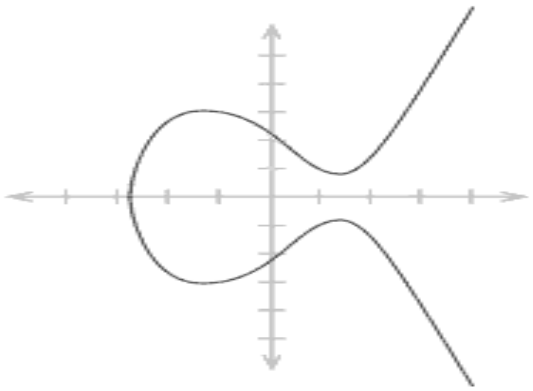

Figure 1: An Elliptic Curve

5.1. Elliptic Curve Discrete Logarithm Problem (ECDLP)

Let P and Q be two points on an elliptic curve such that k*P = Q, where k is a scalar. Given P and Q, it is computationally infeasible to obtain k, if k is sufficiently large. k is the discrete logarithm of Q to the base P. Hence the main operation involved in ECC is point multiplication. i.e. multiplication of a scalar k with any point P on the curve to obtain another point Q on the curve. No sub exponential algorithm to solve ECDLP is known. The security is based on the difficultly of a different problem, which is called the Elliptic Curve Discrete Logarithm Problem (ECD).

6. ECC IN WSN

Wireless sensor networks (WSN) are representative networks using these tiny and low-power sensor devices. Two types of Communications occur in sensor networks: one is between end nodes, and the other is between end node and base station (BS). Since not only the resource restriction in conventional wireless networks but security critical applications, security functions are very important issues in WSN. Main proposes of security in the WSN is not massage encryption but prevent from changing the contents of the massage or disguising sender. It is the most important mutual authentication in order to defend from disguise of sender. Because of limited energy, many researches have done in order to maximize a lifetime of networks. Implementations of symmetric key algorithms are ideal for resource constrained environments of WSN. But symmetric algorithms can only provide confidentiality. Public key cryptosystems are resource hungry but are able to provide a lot more than confidentiality. ECC got the attention of the researchers due to its smaller key size. It offers practical implementation possibilities in resource constrained devices. Previous work shows public key algorithms are a good choice for use in wireless sensor networking, and that the benefits of smaller ECC keys and certificates will be significant in improving energy conservation. ECC is used to achieve

authentication and key management.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 1024 bits for long term security and we know that ECC with only a 160 bits modulus offers an the same level of security as RSA with 1024-bit modulus shown in Table1.Thus, using ECC in wireless communication system is extremely recommended. The key distribution and storage problems, which are typical in secret-key settings it is solved by the ECC cryptography conception.

Table 1. Key size Comparison for ECC and RSA

ECC Key sizes (Bits)	DSA key sizes (Bits)	RSA key sizes (Bits)	Key sizes ratio (Bits)	comment
160	512	1024	1:6	Short period security
256	2048	3072	1:12	Medium period security
384	3072	7680	1:20	Long term security

7. ECC IN WSN

Services such as authentication and key management are critical to communication security in wireless sensor networks as well as the security of sensor network applications. In traditional networks such as the Internet, Public Key Cryptography (PKC) has been the enabling technology underlying many security services and protocols e.g., SSL and IPsec. However, in wireless sensor networks, PKC has not been widely adopted due to the resource constraints on sensor platforms, particularly the limited and deplete able battery power. There has been intensive research aimed at developing techniques that can bypass PKC operations in sensor network applications. For example, there has been a substantial amount of research on random key pre-distribution for pairwise key establishment and broadcast authentication However, these alternative approaches do not offer the same degree of security or functionality as PKC. For instance, none of the random key pre-distribution schemes can guarantee key establishment between any two nodes and tolerate arbitrary node compromises at the same time. As another example, the aforementioned broadcast authentication schemes, which are all based on TESLA requires loose time synchronization, which itself is a challenging task to achieve in wireless sensor networks. In contrast, PKC can address all these problems easily. Pairwise key establishment can always be achieved using, for example, the Diffie-Hellman (DH) key exchange protocol , without suffering from the node compromise problem. Similarly, broadcast authentication can be provided with, for example, the ECDSA digital signature scheme, without requiring time synchronization. Elliptic Curve Cryptography (ECC) has been the top choice among various PKC options due to its fast computation, small key size, and compact signatures.

8. KEY MANAGEMENT SCHEMES IN WSN

Key management and Key establishment is an important issue in wireless sensor networks [5].There are two approaches for it - Centralized and Distributed. In the previous approach there is prior assignment of a unique key to every node and uses the base station as central source of trust while in later approach each sensor node is able to authenticate its neighbors or a subset of them. Most of the traditional techniques in wireless sensor networks are unsuitable because of low power in fact that typical key exchange techniques use asymmetric cryptography, as well called public key cryptography. Usually, Diffie-Hellman which is one of public-key protocols is used to key establishment. In the public-key protocols, data is encrypted with the public key and decrypted only with the private key so it is necessary to maintain both public and private keys. In symmetric protocols data encrypt and decrypt with a single shared key so it has key exchange problem. Secure key distribution of keys securely to communicating hosts is 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 resources seen widespread use is also one of the most accessible illustrations of this principle in action. An efficient key management scheme for WSN using ECC can be designed.

8.1. Properties of Key Distribution in WSN

Other than security requirements, Key distribution mechanisms should support requirements.

8.1.1 Scalability

Must support of large networks and unaffected against the increase of their size.

8.1.2 Efficiency

Consideration of storage, processing and communication limitations on sensor nodes in an efficient way.

8.1.3 Storage Complexity

Amount of memory required to store the keys should be less.

8.1.4 Processing Complexity

Amount of processor cycles required to establish a key should be minimum.

8.1.5 Communication Complexity

Number of messages exchanged during a key generation process

8.1.6 Key Connectivity

Probability that two (or more) sensor nodes store the same key should exist.

8.1.7 Resilience

Should have resistance against node capture (higher resilience means lower number of compromised links).

9. DIFFERENT KEY TEQUNIQUES IN WSN

Both symmetric and asymmetric key management schemes have been suggested to manage and establish keys among the sensor nodes. Most existing key management solutions are based on symmetric cryptography mainly because of its

reasonable energy consumption. Asymmetric cryptography involves the use of a pair of keys (public key and private key) to encrypt and decrypt messages. Each node in the network has a public and a private key, the first is known throughout the network, the second is secret, that is, known only by the node. The source node encrypts messages using the public key of the destination node, and this latter uses its private key to decrypt received messages. In symmetric cryptography, the source and the destination use the same key to encrypt and decrypt messages. Asymmetric cryptography offers better resistance against node compromise attack and allows scalability but requires an additional part on software and hardware of the nodes. Some researchers investigated asymmetric cryptographic tools and propose adapted solutions. Examples of such solutions are Tiny Public Key (*TinyPK*) and Tiny Elliptic Curve Cryptosystem (*TinyECC*). Here is the summarized discussion about both techniques.

9.1. Symmetric Key Management Schemes

9.1.1 Pair -wise distribution

Every node stores a pairwise key with every other node in WSN. If this node is compromised, It will be the only one affected as the communications among the other nodes remain secure. This solution is not scalable.

9.1.2 Master key based predistribution

A master key is predistributed to all nodes in WSN so that the nodes can establish pair wise keys using it along the numbers exchanged with other nodes. Despite of memory saving, WSN is not resilient; once the master key is captured from a node The whole WSN becomes exposed. LEAP [29] addressed this point by erasing the master key from a node after it establishes a pair wise key. But if nodes are captured before the establishment of pair wise keys once added LEAP becomes useless Base station participation: The base station is the central authority which shares a secret key with every node in the WSN. This is called SPINS mechanism. When a node wants to communicate with other node, the base station sends both nodes a pair wise key encrypted with the corresponding shared key. This scheme assumes that base station cannot be compromised and is not scalable due to heavy traffic towards.

9.1.3 Probabilistic key scheme

Every node is provided with a randomly chosen key K from a pool P which is a randomly selected from a huge key space. Two nodes can established a pair wise key by obtaining a common key in their key ring. The resilience of this mechanism is not perfect because once the node is compromised the attacker has the probability k/p to attack successfully between the compromised node and another one.

9.2. Asymmetric Key Management Schemes

Public key management schemes were also examined since it eliminates the key distribution and storage problems. Public /private key pairs the node can establish unique pair wise keys with each other as was done in [30]. Koblitz and Miller developed Elliptic Discrete Logarithm problem (ECDLP) which is a modification of ECC to establish pair wise keys. Pairing based Cryptography (PBC) which is an extension to IBE the practical implementation of ECC in WSNs. The secret key of every node is given as s*N where s is the master key of the network authority, while N is the key derived from its identity.[31] It is impossible to retrieve s from s*N in case of node capturing.

10. AUTHENTICATION MECHANISM IN WSN

An authentication mechanism with low computation and communication overhead is needed to prevent an attacker from performing a Denial-of-Service (DoS) attack by flooding nodes with malicious messages, overwhelming them with the cost of verifying authentication. For instance, for point-to-point authentication of a message, we may use a message authentication code (MAC) and a shared key between the two parties two parties [32].

10.1. Common types of authentication in WSN

10.1.1 One-hop Authentication

A shared link layer key is required between neighboring nodes. The first implemented architecture providing authentication and encryption is TinySec. Though it is full implantation it does not discuss how to establish link-layer keys.

10.1.2 Multi-hop Authentication

End to end shared keys support multi hop authentication. But it fails if one of the nodes in the path is compromised.

10.1.3 Broadcast Authentication

If source node requires to some message like command it broadcasts the message. In this case each packet which is broadcast should be authenticated so that no false data is inserted.

11. PROPOSED SECURE KEY PREDISTRIBUTION SCHEME

An efficient key management protocol for heterogeneous WSN using ECC can be designed using ECC. A heterogeneous WSN can be assumed as a combination of both large number of normal sensor nodes also known as cluster heads (H-node) and small number of special nodes. Cluster nodes having more power computationally more capable than special nodes having more power, larger communication cost and storage. The proposed scheme deals with key predistribution scheme using ECC. An elliptic curve point called as Head key (H_k) is assigned to each node and a private key pool is generated by performing point doubling and addition operations over the Head key before the deployment. Finally, the key pool is predistributed to the sensor nodes prior to deployment. The properties of elliptic curve are used in the generation of private keys of each node. If two nodes share a common private key, link information can be shared .The value of the prime field and key pool size can determine the probability of two nodes sharing the same private key. Suitable values can increased the connectivity between the sensor nodes.The proposed protocol involves three phases Key Generation phase ,Key Predistribution phase and Key Agreement phase.

11.1. Generation phase

The base station generates a suitable elliptic curve and corresponding base elliptic curve parameters are selected to determine Head key (H_k) according to the dimension of sensor network. ECC makes use of elliptic curves in which the variables and coefficients are restricted to elements of a finite field which are prime curves defined over odd prime field *Fp* .A finite field *Fp*, where *p* is an odd prime number, is defined as a set of all integers between 0 and *p*–1. The elliptic curves over a finite field are defined by $y2 \bmod p = x3 + ax + b \bmod p$. where the coefficients *a* and *b* and the variables *x*

and *y* all take values only from the finite field. It can be represented as *Ep*(*a*, *b*). A point on the elliptic curve can be represented as *P*=(*x*, *y*), where *x*, *y* ∈ *Fp*. The modulo *p* function performs a wrapping around operation so that the elements are all within *Fp*.Hence in this phase appropriate value of a, b (elliptic curve coefficients) and p are selected.

11.2. Key Predistribution Phase

Elliptic curve points are generated that satisfy the chosen elliptic curve**.** Each sensor node is allotted a Head key (H_k).A pool of private key is generated by the help of point doubling and point additions.

11.3. Key Agreement Phase

Any two neighboring nodes can establish the communication if they share a common key otherwise they will establish the desired path with the help of intermediate node. Probability of sharing a common private key can be increased by choosing suitable elliptic parameters.

12. RESULTS OF THE PROPOSED PROTOCOL

In this scheme we have considered the value of the prime field and the elliptic curve coefficients a, b and p as the elliptic curve parameters. For a sensor network consisting n nodes, the value of p is a odd prime number larger than n.The Elliptic Curve is represented by the equation is considered for the generation of Head key is given by $Y^2 \bmod 23 = x^3+x \bmod 23$. Here a=1, b=1& p=23.The equation can be assumed for a sensor network consisting of nodes 20.

12.1 Generation of Head keys

For a sensor network of 20 nodes with the value of p as 23, points on the curve can be found as shown in Table 2.

Table 2. Generation of Head keys E_{23} (1, 1)

(0,1)	(0,22)	(1,7)	(1,16)	(3,10)
(3,13)	(4,0)	(5,4)	(5,19)	(6,4)
(6,19)	(7,11)	(7,12)	(9,7)	(9,16)
(11,3)	(11,20)	(12,4)	(12,19)	(13,7)
(13,16)	(17,3)	(17,20)	(18,3)	(18,20)
(19,5)	(19,18)			

Here, 27 Head keys are generated for the selected elliptical parameter. So each sensor node can have unique Head key.

The algorithm *'genPoints'* describes the process of generating the points shown in the above Table 3.

Table 3: Algorithm for generating points on Elliptic Curve

```
{
for(xj=0;xj<p;xj++)
for(yj=0;yj<p;yj++)
caly=yj*yj;
calx=(xj*xj*xj)+(xj);
if(caly%p==calx%p)
Output(xj,yj).
}
}
```

12.2 Generation of private key pool

A private key pool for each sensor node is generated and is distributed to each sensor node prior to its deployment. For each head key, point multiplication operation is used generate private key pool. For every Head key defined by point P, if key pool size is k then using scalar multiplication we can calculate 2P,3P…….,kP. The key pool size determines how many times scalar point multiplication is performed.For example for a Head Key of sensor node A represented by the point P (0, 1) which is a point on elliptic curve of the considered equation $E_{23}(1,1)$, a private key pool of size (k=10) is generated is shown in the Table 4 shown below.

Table 4. Private Key pool for Head key P (0, 1)

1P	(0,1)
2P	(6,19)
3P	(3,13)
4P	(7,11)
5P	(11,3)
6P	(5,19)
7P	(19,18)
8P	(12,19)
9P	(19,5)
10P	(12,4)

The basic idea behind this private key pool generation is to use the EC operations viz, point addition and point doubling. Elliptic curve cryptographic primitives require scalar point multiplication. Say, given a point P(x, y) on an EC, if one needs to compute k*.P, where k is a positive integer is achieved by a series of doubling and addition of P. While generating private key pool to determine 2P (X_{2P}, Y_{2P}) from P (X_P, Y_P), double operation is used. 2P can be written as 2P=P+P, here the points used equal points. So we can use double operation. Equation for double operation is given as

$S = [(3Xp^2 + a)/2Yp] \bmod p$

Then 2P has affine coordinates given by (X_{2P}, Y_{2P})):

$X_{2P} = (S^2 - 2 * Xp) \bmod p$

$Y_{2P} = [S (Xp - X_{2P}) - Yp] \bmod p$

Now to determine 3P, we use addition of points P and 2P ie.3P=2P+1P since the points involved is different therefore we will use addition operation given as

Equation for addition operation

$S = [(Y_Q - Y_p) / (X_Q - X_p)] \bmod p$

$X_{2p} = (S^2 - X_p - X_Q) \bmod p$

$Y_{2P} = [(S (X_P - X_{2p}) - Y_P] \bmod p$

Here Q=2P having coordinates **(**X_Q , Y_Q) is assumed.

Therefore we apply doubling and addition depending on a sequence of operations determined for 'k'. Every point evaluated by doubling or addition is an affine point (points on the Elliptic Curve). The base point P is selected as (0, 1). Base point implies that it has the smallest (x, y) co-ordinates which satisfy the EC, p is another affine point, which is picked out of a series of affine points evaluated for the given EC.

13. SECURE SENSOR NODE MUTUAL AUTHENTICATION PROTOCOL

The proposed authentication protocol is an improved authentication protocol over identity based authentication which provides proper protection on data from unauthorized access. This protocol uses the concept of ECC takes less sensor node's memory and less computational overhead due to which it can be efficiently applicable to wireless sensor network. The proposed scheme mainly introduces the security authentication protocol for deployment of new node at the commencement of a network. In this Protocol, an authentication mechanism for new nodes joining sensor networks is developed. Without loss of generality, the proposed method would achieve two tasks firstly authentication of new nodes and secondly establishment of key during authentication, in which a shared key should be created between a neighboring nodes and deployed node to provide a secure communication. The proposed protocol consists of mainly three phases: A System Initialization phase, Sensor Node Registration phase and a mutual Authentication phase.

13.1 Assumptions regarding System Architecture

The following assumptions are made regarding the proposed security framework:

1. The sensor network is static, i.e. the sensor nodes are not mobile.

2. Each sensor node has Time stamp.

3.The base station acts as a controller a master entity which is responsible for generating private keys and is secure from various kinds of attacks, trustworthy and has powerful resources in terms of energy, memory and computation.

4. All the sensor nodes are similar in terms of energy, memory and computation capabilities.

5. The sensor nodes have enough memory to manage with the keying overheads.

6. The WSN consists of m sensor nodes and n base stations (m>n).The base stations can directly communicate with each other through a secure channel.

7. A system administrator which can directly interact with base station and is responsible for maintaining the identities of all the sensor nodes and updates whenever there is a change.

13.2 Symbols used in proposed protocol

Following symbols has been used in the description of the proposed protocol whose meaning has been described in the Table 5.

Table 5. Meaning of the symbols used in the proposed authentication protocol

Symbol	Meaning
BS	Base station
BS_{pk}	Base station's public
BSs_k	Base station's private
SK_A	Sensor node A's private
PK_A	Sensor node A 's public
SK_B	Sensor node B's private
PK_B	Sensor node B's pubic
ID_A	Identity of sensor node
S_{AB}	Common Shared secret key between A and B
TS	Sending Timestamp
TC	Current Timestamp

14. DETAILED DESCRIPTION OF THE PROPOSED PROTOCOL

The working of the proposed protocol can be described by three phases: A System Initialization phase, Sensor Node Registration phase and a Mutual Authentication phase.

14.1 System Initialization Phase

BS being the master entity uses the Elliptic curve method initializes its base parameters which consists of {F_P, E, G, n, H, BSs_k , BS_{pk}} where F_P is the prime field, E is the elliptic curve, G is the base point of the elliptic curve, n is the order of the elliptic curve and H is the hash function which can map any point to F_P, BSs_k is the base station private key which is selected by the base station itself .Base station keeps its private or secret key and other parameters are broadcast. Functions performed by the BS are summarized below.

1. BS generates its private key BSsk.

2. BS calculates its public key as $BS_{pk} = BSs_k * G$.

14.2 Sensor Node Registration Phase

Every sensor node has to register with the BS before starting communication with BS or other sensor node. Each sensor node has unique identity which is initially obtained from system administrator and stored by it before registration with the BS. Identities of all sensor nodes are assumed to be stored by the system administration in a table. Suppose sensor node A wants to register with the BS. Function performed by the Sensor node A is described below in 3 steps.

1. Sensor node A selects a random number SK_A € F_P as its private key.

2. Sensor node computes its public key $PK_{A=} SK_A * G$.

3. Sensor node sends the parameters (ID_A, TS, PK_A) to BS.

Function performed by the BS is described below in 2 steps.

1. BS uses the hashing function on the identity of sensor node A as H (ID_A) and also computes $T_A = BSs_{k*} H(ID_A)$ and transmits T_A to sensor node A through a secure channel.

2. S broadcast the parameters (H (ID_A), TS ,BSpk)

Verification and acknowledgement by sensor node is done by sensor node after receiving the broadcast message. Verification is done by first calculating $PK_A * T_A * SK_A$. If $PK_A * T_A * SK_A = H (ID_A)* BS_{pk}$ then sensor node registration is successful otherwise not.

14.3 Mutual Authentication Phase

After successful node registration, authentication phase is needed between the sensor nodes or with BS for secure communication. For this phase an assumption is made that sensor node A and sensor node B have already generated public and private keys according to the protocol described as above key pre-distribution protocol. Also BS has already broadcast the hash value of the node's identity.

14.3.1 Pre Authentication Phase

Before actual authentication protocol, pre authentication phase is performed .This phase is explained as below.

Suppose sensor node A has send an authentication request to sensor node B. Authentication request consists of parameters (HID_A, TS).Before starting authentication protocol, first sensor node B will confirm that it has the hash value of sensor node A identity and it matches with the published parameter and vice versa. If it has, then authentication protocol will execute otherwise authentication request is rejected by sensor node B. This can be explained with the help of Figure 2 below.

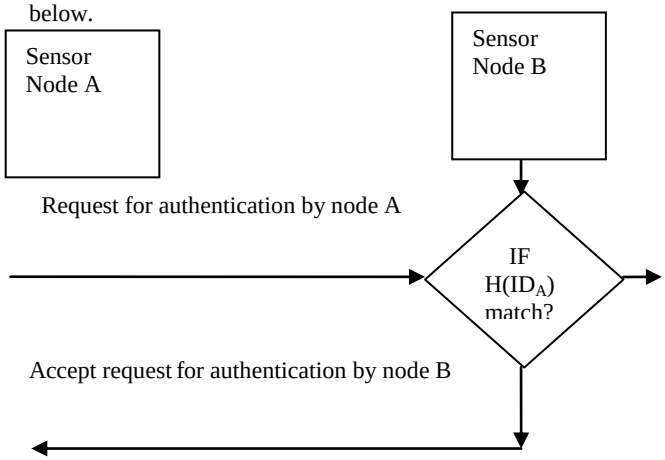

Figure 2: Verification of node registration before authentication

14.3.2 Authentication Protocol

After verifying the node registration of the sensor node A which has sent authentication request in pre authentication phase by sensor node B, authentication begins. The stepwise procedure performed by both the sensor nodes is given as below.

1. Sensor node A selects a random number k_A and gets public key of sensor node B PK_B from BS computes $C_A = k_{A*} PK_{B.}$

2. Sensor node A transmits ($C_{A,}$ TS) to sensor node B.

3. Compute difference between timestamps.

4. If ((TC-TS) > delay) then reject authentication by sensor node B otherwise continue.

5.Sensor node B selects a random number k_B and gets public key of sensor node A PK_A from BS computes $C_B = k_{B*} PK_{A.}$

6. Sensor node B transmits ($C_{B,}$ TS) to sensor node A.

7. If ((TC-TS) > delay) then reject authentication by sensor node A otherwise continue.

8. Sensor node A using $C_{B\ ,}$ computes a common secret key $S_{AB =} k_{A*} C_{A*} C_B$ sends it to sensor node B.

9. Sensor node B using $C_{A\ ,}$ computes a common secret key $S_{BA =} k_{B*} C_{B*} C_A$.

Mutual authentication is successful if both the common secret keys at sensor node A and B are equal.ie $S_{AB =} S_{BA.}$ The steps are represented in the Figure 4.2 below.

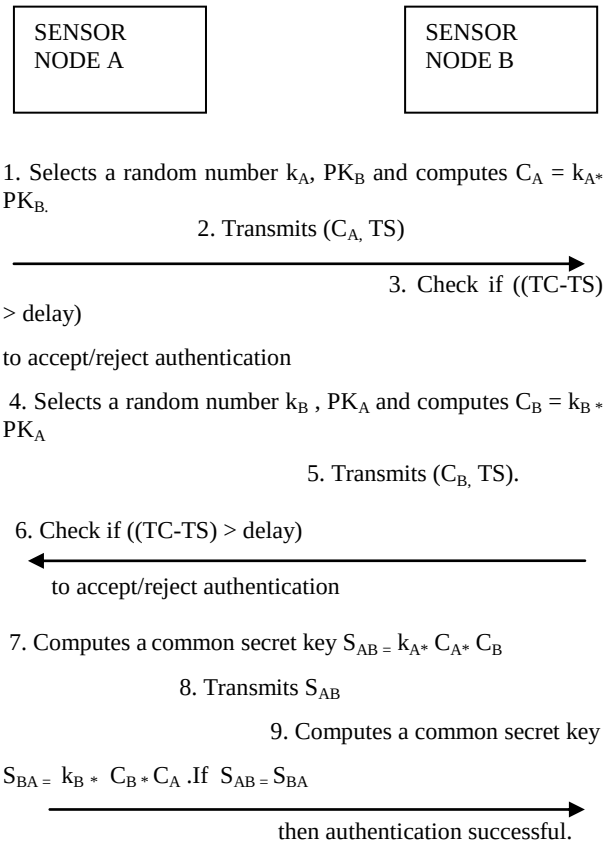

Figure 3: Authentication between two sensor nodes

15. CONCLUSION

A Key Predistribution protocol and mutual node Authentication using Elliptic Curve Cryptosystem is proposed with a goal of increasing the speed and decreasing the power and memory .constraints. By implementing proposed Key Predistribution protocol in WSN, many benefits can be achieved which are summarized as:

15.1 Security

The Implemented method provides the better security. The smaller key size makes possible much more compact

implementations for a given level of security, which means faster cryptographic operations.

15.2 Decreased memory space, power and Bandwidth

Due to the use of ECC operations which uses shorter key size as compared to other public key cryptosystems like RSA it has the above advantages associated with it as mentioned above. This protocol main core operation is ECC multiplication so its efficiency directly determines the performance of the system. It uses the concept of Hash value of identity of sensor node and the concept of timestamp to check for reply attack. It also avoids the tedious calculation for private and public nodes hence the design is simple. Also sensor node stores the hash value instead of identities of sensor nodes to reduce the consumption of memory storage for sensor nodes. In addition to key management and secure communication, public-key cryptography can be the enabling technology for numerous other WSN applications, including securely connecting pervasive devices to the Internet and distributing signed software patches. In future, focus will be on more concrete solution for the node capture attack and implementing the overall protocol to monitor the actual effect in the real environment in terms of different parameters like security, energy consumption, efficiency, durability etc.

16. ACKNOWLEDGEMENT

It is with deep sense of gratitude and reverence that we express our sincere thanks to **Dr. A.K.Sharma,** Dean (PG Study and Research),B.S.A.I.T.M and **Sh. Pawan Bhadana,** HOD (CSE/IT Department) B.S.A.I.T.M, for their guidance, encouragement, help and valuable suggestions throughout.

17. REFERENCES

[1] Guo Xiao Wang, Zhu Jianyong, Analysis and Design of Energy-oriented Security Protocols for Wireless Sensor Networks,"2011

[2] Hero Modares Rosli Salleh Amirhossein Moravejosharieh, Overview of security issues in Wireless Sensor Network,"2011

[3] E.Yoneki and J. Bacon ,"A survey of Wireless sensor Network technologies,"2005.

[4] J.P Walters, et al." Wireless Sensor network security: A survey, "2007

[5] Hero Modares Rosli, Salleh Amirhossein ,Moravejo sharieh, Wireless Network Security Using Elliptic Curve Cryptography," 2011

[6] Moncef Amara and Amar Siad," Elliptic curve Cryptography and its applications", 2011

[7] Guo Xiaowang, Zhu Jianyong, Research o"n Security Issues in Wireless sensor networks,"2011

[8] Yong Wang, Garhan Attebury, and Byrav Ramamurthy A Survey of security issues in wireless sensor networks, University of Nebraska-Lincoln

[9]Pathan,A-S.K.,Alam,M.,Manowar,M.,and Rabbi,F.,"An efficient Routing Protocol for Mobile Ad Hoc networks with Neighbour Awarness and Multicasting.,Proc IEEE E-Tech Karachi I,2004,PP.97-100

[10] C.P Fleeger,Security in Computing,3rd edition,prentice_Hall Inc.NJ.2003