ENERGY EFFICIENT PKI SECURE KEY MANAGEMENT TECHNIQUE IN WIRELESS SENSOR NETWORK USING DHA & ECC

Prof(Dr.) Mohd. Rizwan beg[1] and Shish Ahmad[2]

[1] professor & head, CSE dept ,I.U. Lucknow, India
rizwanbeg@gmail.com
[2]Research scholar, CSE dept ,I.U. Lucknow, India,
shish_parv@rediffmail.com

ABSTRACT

Sensor Network are used for variety of application, such as emergency rescue, disaster relief but this is vulnerable to attacks. To make conversation confidential to the adversary so that not able to forge the data. so to provide the security for data exchange between sensors and base station energy consumption should be minimum . For secure communication in sensor network having many resource constrained, symmetric methods are preferred, because of its friendly nature of low resource consumption. But In this paper we propose an energy efficient secure pubic key algorithm that proves the authentication and also provides secure communication among sensor in such a way that energy consumption minimizes.

KEY WORDS:

Sensor Networks, Public key cryptography, Security, Confidentiality, Authentication, Key distribution, Wireless networks, Diffie-Hellman algorithm, Elliptic Curve cryptography, energy saving.

1. INTRODUCTION

Wireless sensor networks are rapidly deployable, self-configurable, and low cost and operate in absence of a pre-deployed infrastructure. Sensor networks are used for a variety of applications , such as emergency rescue, disaster relief, smart homes and patient monitoring, industrial applications, such as structural health monitoring and environmental control, and military applications, such as target identification and tracking. These are often deployed in unattended environments, thus leaving these networks vulnerable to passive and active attacks by the adversary. The conversation between sensors nodes can be eaves dropped by the adversary .The adversary can be aware of the conversation between the sensors and can forge the data. Sensor nodes should be resilient to these attacks. Since Sensor nodes are resource constrained and run on battery, energy consumption should be low to make it operate for many days.

In sensor network security, the challenge is the design of protocols to bootstrap the establishment of a secure communications infrastructure of sensor nodes with some secret information, but have

DOI : 10.5121/ijasuc.2012.3103

had no prior direct contact with each other, referring to this problem as the bootstrapping problem. A bootstrapping protocol not only enable a newly deployed sensor network to initiate a secure infrastructure, but it must allow nodes deployed at a later time to join the network securely. The difficulty of the bootstrapping problem suffers from the numerous limitations of sensor networks such as limited memory, limited processing power, limited bandwidth, lack of physical security and easy accessibility to adversaries.

If the sensors are deployed via random scattering (e.g. from an airplane), the network protocols cannot know beforehand that after deployment which nodes will be within communication range of each other. Even Deployment of nodes by hand, the large number of nodes involved to pre-determine the location of every individual node makes it costly. Hence, any security protocol should priory not assume the knowledge of which nodes will be neighbors in a network.

Secure symmetric encryption will be widely available on the Sensor Network. Effective use of that secure symmetric encryption capability is a critical problem. As is always the case with symmetric encryption, proper key management is a fundamental concern.

The future in sensor security is the public key cryptography ,because it is easy to distribute keys in public key cryptography than symmetric key cryptography because of the random deployment of the sensor nodes in the network, as well as it is also difficult to prove authentication for adversary in public key cryptography.

For implementing public key cryptography care should be taken in Sensor Network because of the constrained of sensor network devices.

The symmetric methods are proffered to provide confidentiality, because it consume less energy for the generation of cipher text as compared to asymmetric method, because public key cryptography method (RSA, DHA) involve power function calculation for the generation of key or cipher text.

The next issue is the security in sensor network. There can be many types of attacks are possible in sensor network. The presented security issues for sensor networks have not been addressed at all. It does not provide assurance for replay attack, authentication, and confidentiality.

In this paper we have applied the public key method in such a way that total energy consumption decreases as compared to conventional public key method for secure distribution of keys or for providing confidentiality.

Here we are presenting a Public key method using Diffie Hellman algorithm & Elliptic Curve Cryptography for preventing replay attack in sensor network as well as for data confidentiality and authentication between sensor nodes. We shows that our security method successfully prevent attacks and prove authentication with some constraints.

Here we are also providing the mechanism for inserting a new sensor node into a pre deployed network such as it can distribute keys to its neighbors for secure communication as well as it can prove itself as a authenticated node of that network.

2. NETWORK SECURITY BACKGROUND

Any security mechanism applied to prevent security attacks will require fundamental basic security services such as authentication, confidentiality, non-repudiation and message integrity.

- Confidentiality: Confidentiality ensures that only sender and the intended receiver should be able to understand the contents of transmitted message.
- Authentication: In authentication that both the sender and receiver should be able to confirm the identity of the other party involved in communication.
- Integrity: Integrity guarantees that the message is not altered.
- Non-repudiation: Non-repudiation ensures an entity to prove the transmission or reception of information by another entity.

2.1 Various Types of Security Attacks

- Passive attacks: In passive attack an unauthorized user monitors on the communication between two parties.
- Active Attacks: In active attacks attacker is not only being able to listen to the transmission but also being able to actively modify(change) or generate false data. Types of Active attacks are [P 03]-
- Masquerade (Impersonation)
- Replay: (Delayed message)
- Denial of Service(unreachable services)
- Modification of Messages (active changes)

2.2 Sensor Network Limitations:

- Partial impracticality of public key cryptosystems
- Vulnerability of nodes to physical capture
- Lack of a-priori knowledge of post-deployment configuration
- Limited memory resources
- Limited bandwidth and transmission power
- Over reliance on base stations exposes vulnerabilities

2.3 Attacks On Sensor Network Routing:

- Spoofing, Alteration, or Replaying Routing Information: Adversaries can create routing loops, attract or repel network traffic, extend or shorten source routes, generate false error messages
- Selective Forwarding: Adversary nodes may refuse to forward some messages and simply drop them, ensuring that they are not propagated any further.
- Sinkhole Attacks: In this attack, the adversary's lure nearly all the traffic from a particular area through a compromised node, creating a metaphorical sinkhole at the center at the adversary.

- The Sybil attack: In a this attack, a single node presents multiple identities(adversary) to other nodes in the network.
- Wormholes: In the this attack, an adversary attarcts messages received in one part of the network and replays them in a different part
- Hello flood attack: Adversary broadcast routing and other information with enough transmission power could convince every node in the network that the he/she is its neighbour.
- Acknowledgement Spoofing: An adversary can spoof the node by using link layer acknowledgments for "overheard" packets addressed to nearby nodes.

3. RELATED WORK

3.1 Key distribution techniques in sensor networks. Nodes pre-initialized with some secret information before deployment, but only after network setup, we know the location of nodes. The node location often determines which nodes need to establish cryptographic keys with which other nodes, so we cannot set up these keys before deployment

3.2 Evaluation metrics. Following are several criteria that represent characteristics for a bootstrapping scheme for sensor networks.

Resilience against node capture

Resistance against node replication

Revocation

Scalability

3.3 Using a single network-wide key: This is the simplest method of key distribution in which a single network wide key onto all nodes before deployment.

3.4 using pair wise-shared keys: In this method, every node in the sensor network shares a unique symmetric key with every other node in the network.

3.5 Random key pre-distribution scheme: By this method we can distribute keys to each of the sensor node before deployment by using Random Key Generator.

3.6 SPINS : It has two secure building blocks: SNEP and µTESLA.

- SNEP provides security services like: Data confidentiality, authentication, and data freshness.
- A new protocol µTESLA which provides authenticated broadcast for resource-constrained environments.
- SNEP provides following properties:
- Semantic security: Since after each message the counter value is incremented, different encrypted message will be generated for same message each time. The value of counter is so long that within the sensor's lifetime it never repeats.

- Data authentication: If the MAC is correct, a receiver can be assured that the message generated from the claimed sender.
- Protection of Replay attack & Data Freshness: The value of counter in the MAC prevents replaying older messages. If the counter were not given in the MAC, an adversary could easily replay messages.
- Low communication overhead: The counter state is kept at each end point and does not need to be sent in each message.
- Asymmetric method through a delayed disclosure of symmetric keys is introduced by µTESLA , which provide an efficient broadcast authentication scheme.
- It requires that the base station and sensor nodes are synchronized by time. To send an authenticated packet, the base station simply computes a MAC on the packet with a key that is secret at that point in time. After getting a packet, node can verify that the corresponding key was not yet disclosed by the base station.
- Since a receiving node is assured that the MAC key is known only by the base station, the receiving node is assured that no adversary could have altered the packet in transit. The node stores the packet in a buffer. The base station sends the verification key to all receivers. The sensor node verifies the correct of the key on receiving it (as below). If the key is correct, the stored packet can be authenticated by the sensor node. Ki=F(Ki+1)

3.7 General consideration of using public key method:

- The common perception of public key cryptography is that it is complex, power hungry and slow, and not suitable for use in low power environment like wireless Sensor Network.
- But in this paper we challenge the basic assumption about public key cryptography in Sensor Networks which are based on the traditional software based approach
- We can impement public key cryptography in Sensor Network for security, provided we use the right selection of algorithms and associated parameters, careful optimization, and low power design techniques.
- Public Key Distribution Techniques In Sensor Networks : After the nodes has been deployed, they perform key exchange by exchanging their respective public keys and signatures of master key. Each public key of a node is verified as legitimate by verifying its master key's signature using the master public key. After the public key of a node has been received, a symmetric key between the link can be generated and sent to it, encrypted by its public key. Upon reception of the session key, establishment of keys are complete and the two sensor nodes can communicate using the symmetric key.

3.7.1 Example-RSA

Let The Base station have the master public key (Nbs,Ebs) and private key (Nbs,Dbs).
Let A have public key (Na,Ea) and private Key (Na,Da).
Let B have public key (Nb,Eb) and private Key (Nb,Db).
Let Node A Want to Varify its Public key to node B DcrNbs,Ebs(Enr(Nbs,Dbs) (Na,Ea))
Let B want to establish secure communication to A. B follow following Steps.
After verification of the public key of A as above, B generate a Random Session Key KAB,
It Encrypt the session key by applying the public key of A as follows.

Enr (Na,Ea) (KAB)

3. B sends this message to A .

4. A find the Session Key by apply the decryption process by its private Key as follows
   Dcr(Na,Da) (. Enr (Na,Ea) (KAB))

But due to the power function calculation the above algorithm consumes much power.
Following the refined algorithms using public key methods that takes less energy.

3.7.2 Rabin's scheme

Through the factorization problem of large numbers and is similar to the security of RSA with the same sized modulus. Rabin's Scheme has pubic key computational cost. The encryption operation is fast, however decryption times are comparable to RSA of the same speed.

Key Generation

1. Choose two large random strong prime numbers.
2. Compute n = p ¢ q.
3. Pick a random number b for which 0 · b < n.
4. The public key is (n; b), the private key is (p; q).

Encryption

1. Represent the message as an integer x for which 0 · x < n
2. Compute the ciphertext En;b(x)modulox(x + b) mod n, as .

3.7.3 The NtruEncrypt Public key cryptosystem

Key Generation

The following steps generate the private key f(x):

1. Choose a random polynomial F(x) from the ring R. F(x) should have small coefficients, i.e. either binary from the set (0,1) (if p = 2) or ternary from (-1; 0; 1) (if p = 3 or p = x + 2).
2. Let f(x) = 1 + pF(x)

The public key h(x) is derived from f(x) in the following way:

1. As before, choose a random polynomial g(x) from R.
2. Compute the inverse f¡1(x) (mod q).
3. Compute the public Qkey as h(x) = g(x) * f¡1(x) (mod q).

Encryption

1. Encode the plaintext message into a polynomial m(x) with coefficients from either (0; 1) or (-1; 0; 1).
2. Choose a random polynomial Φ(x) from R as above.
3. Compute the cipher text polynomial c(x) = p Φ(x) *h(x) + m(x) (mod q).

Decryption

1. Use the private key f(x) to compute the message polynomial m' (x) = c(x) * f(x) (mod p).
2. Map the coefficients of the message polynomial to plaintext bits.

3.7.4 Improved Public Key Method

1. This scheme tries to solves security in WSN by the use of public key cryptography (RSA) for ensuring the authenticity of the base station.
2. RSA is composed of two phases, the first is the sensor to base station handshake in which the base station and a given sensor node setup a session key to secure end to end link between them, this handshake is protected and authenticated using the public key of the base station.
3. The second phase is the use of this session key for data encryption to ensure confidentiality and ensuring the integrity of the exchanged data using the MAC joined to each packet.
4. This increases the security because this method provide end to end encryption with link to link i.e. (Sensor to base station and senor to sensor)

3.8 Analysis

Following result shows the energy consumption of public key algorithm compared with the symmetric one.

Table 1

Algorithm	Energy
RSA-1024	397.7 µJ
AES-128 Enc/Dec	2.49 µJ

This measurements on an Atmel ATmega128L low-power 8-bit microcontroller.

Following result shows that the energy consumption using Elliptic curve cryptography is less than the RSA.

Table 2

Algorithm	Client	Server
RSA-1024	397.7 µJ	390.3 µJ
ECC-160	93.7 µJ	93.9 µJ

Energy consumption on handshake protocol Mica2dot platform.

So here based on the above tables, we are proposing the following method of public key algorithm i.e. Diffie Hellman algorithm & ECC in such a way that it consumes less energy as compared to the traditional methods mentioned above in the table.

4. PROPOSED METHOD

4.1 Problem Statement

When a Sensor node sends the sensed data to the base station, the data must be confidential through the route from the source node to the base station. But if the data is passed through the malicious node it can read or modify the data. Our algorithm keeps the data confidential from the source node to the base station at each step.

(a) A malicious node can enter our network and can send forge or confused data to the base station be pretend as a authorized node.

(b) It can also modify, insert or delete the data during transmission impersonated as a legal node. Our algorithm proves the authentication and keeps data integrity.

(c) Any unauthorized malicious node can send duplicate data and can attempt to repeat authorized data to the base station which is already send. Our protocol also protects replay attack.

(d) The sensor network must be robust menace if let a new node is added to the existing network, it should be added network securely.

So our algorithm minimized the above said attacks at the Sensor Network and prevents the following i,e,

4.2 Proposed Method

4.2.1 Assumptions:

Each Sensor Node has unique id assigned before deployement.
Sensor nodes are homogeneous and Static.
Constant power supply, i.e no change in capacitance, resistance and inductance in hardware
It should be ensured that adversary cannot compromise Sensor Nodes immediately after nodes are deployed. He takes a few minutes of time to compromise them after they are deployed.
Each Sensor node has a comparator also.

4.2.2 Algorithm

Our method is based on Diffie-Hellman algorithm and Elliptic curve cryptography for light weighted and resource constrained Sensor nodes with some modification. Here each node is limited to broadcast the message to only its neighbors. Our algorithm has three phases

(1) Before deployment of the Sensor Nodes,
(2) After deployment of the Sensor Nodes,
(3) Addition of a new node in exiting network.

Deffie Hellman Phases

Phase1: Before deployment of the Sensor Nodes using Deffie Hellman algorithm

1. The Base Station select global elements q and α such that q should be a prime number not more than 1024 bits and α should be less than q and also be the primitive root of q or generator(base) of q.
2. The base station select any private value X that should be less than q for every node differently and for itself also.
3. The base station calculate public value for every station (including itself as node) Y by using following equation
   Y=αx mod , Now we deployed every node with private and public values I,e, X and Y respectively with only public value q.

Phase 2: After deployment of each Sensor Nodes using Deffie Hellman algorithm

1. Now every node of our static network broadcast their public value Y to its neighboring nodes with its id.
2. Now every node calculate its secret key (that will be different for each pair) by using following equation.
   K=Y X mod q
3. Now every node has a secret key to exchange the message to each other with its id. This show confidentiality, data integrity and authentication to each other.
4. Then as first message every node sends a HELLO packet to its neighbors containing its id and a nonce starting with 1 and encrypted with respective Key.
5. Now the receiving node receives and decrypts the HELLO packet and store the Nonce with id.
6. For any next message between them every packet contains the Nonce with a increment of one with the data so that the receiver can verify that the current data is not a duplicate one using comparator. So it can prevent the replay attack.

Phase 3: Addition of a new node in exiting Network using Deffie Hellman algorithm

1. Now if a new Sensor Node is deploys to the exiting one with the same public values that is X, Y & q. It exchanges the public value Y and q to its neighbors.
2. By using above method the neighbors generate the corresponding keys by selecting any random value X that should be less than q.

Elliptic CC Phases

Phase1: Before deployment of the Sensor Nodes using ECC

1. The Base Station select a large integer q, which is either prime number p or an integer of the form 2m and elliptic curve parameter a and b for following equation.
   Y2 + xy = x3 + ax2 + b.
   This defines the elliptic group of points Eq(a,b). The base station also picks a base point G from the above points whose order is a very large value n.
2. The base station select private value n1,n2…..nN for sensor nodes 1,2….n respectively, which is less than n for every station and for itself also. These are the private keys for each of the sensor nodes and base station.
3. The base station generates public keys for each of the sensor nodes and for itself by following equation.
   P = n * G

Where n and P are the private and public values of the nods.

Phase 2: After deployment of each Sensor Nodes using ECC

1. Now every node of our static network broadcast their public value P to its neighboring nodes with its id.
2. Now every node calculate its secret key (that will be different for each pair) by using following equation, let second node is the neighbor of the first node, so by using following equation 1 and 2 generate same key K(symmetric Key) at both the end.
   K=n1 * p2 (at node 1) and K= n2 * p1 (at node 2)
3. Now every node has a secret key to exchange the message to each other with its id. This show confidentiality, data integrity and authentication to each other.
4. Then as first message every node sends a HELLO packet to its neighbors containing its id and a nonce starting with 1 and encrypted with respective Key.
5. Now the receiving node receives and decrypts the HELLO packet and store the Nonce with id.
6. For any next message between them every packet contains the Nonce with a increment of one with the data so that the receiver can verify that the current data is not a duplicate one using comparator. So it can prevent the replay attack.

Phase 3: Addition of a new node in exiting Network using ECC

1. Now if a new Sensor Node is deploys to the exiting one with the same values that is nR, PR and G It exchanges the public value PR and G to its neighbors.
2. By using above method the neighbors generate the corresponding keys by using its previous value that is encrypted with its symmetric key.

4.3 Results and Analysis

So in the above phases, we found and shifted those steps of key/Cipher text generation to the base station, which is not necessarily required at sensor node, because public key method involve power function calculation, which is more power hungry, we minimizes this consumption by shifting it.

Following low level assembly code shows one of the power function calculation for Deffie-Hellman algorithm

After shifting one power function calculation to the base station

Total energy saved ≈ (clock per machine cycle/ frequency of processor) * Total mac\hine cycle to calculate the power_function

4.3.1 Example-:

Assumption-

Using 8051 microcontroller in sensor of XLAT=11.0592 MHz
Continuous power supply i.e. resistance, capacitance & inductance are static 12 clock per machine cycle

Let q, α, private keys & secret key lies between 0-255

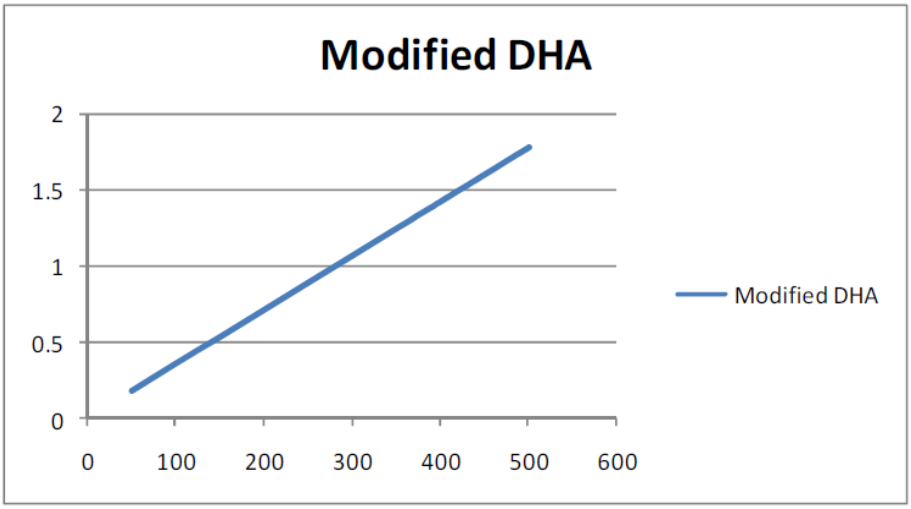
Table 3: Shows the code for power function calculation for DHA

Code	Machine cycle	Time taken of executing the Instruction (µs)
Power_function: ORG 500		
MOV A, #01	1	1.085
MOV R1, # Private key of sender/Receiver	1	1.085
Again: MOV B, # Global public key component(	1	1.085
primitive root of q)	4	4.34
MUL AB	1	1.085
MOB B, # Global public key component q (a	4	4.34
prime number)	1	1.085
DIV AB	2	2.17
MOV A,B		
DJNZ R1, Again		

Total Time taken for executing the above code=(12/11.0592) * 13* private keys of neighboring sensors

So above is the total execution time saved at sensor nodes by shifting that power function to the base station.

And if the sensor nodes are large enough, we can save substantial amount of energy as shown in the graph. Let the private key is 255 at both the end..Following Figure shows the time taken by the sensor for running above code as the number of sensors increases.



1. Here We used the Diffie-hellman algorithm with little modification on Sensor Network. The Diffie-Hellman algorithm depends for the effectiveness on the difficulty of the computing discrete logarithms

4.3.2 Example 2

If we are using ECC in modified fashion, and finding and shifting those function which does not take part in key establishment process/ generation of cipher text at sensor nodes

Assumption-

Using 8051 microcontroller in sensor of XLAT=11.0592 MHz

Continuous power supply i.e. resistance, capacitance & inductance are static 12 clock per machine cycle

Let private keys, Public key & secret key lies between 0-255

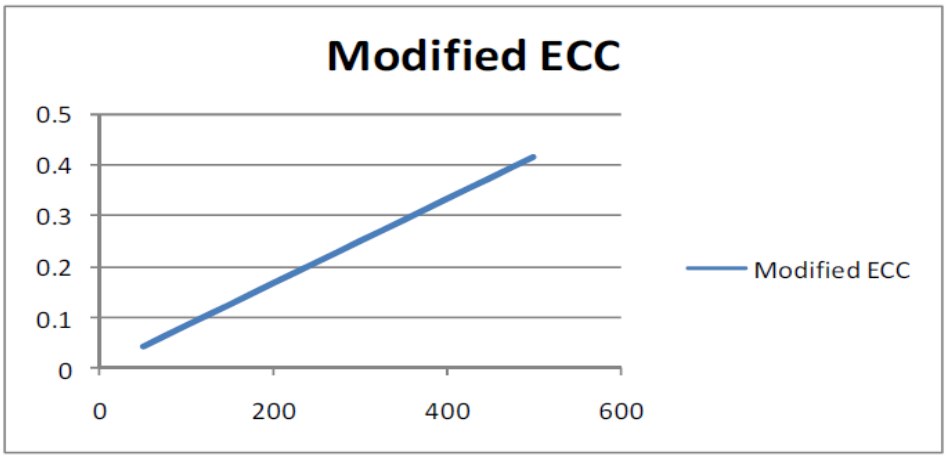
Table 4: Shows the code for power function calculation for DHA

Code		Machine cycle	Time taken of executing the Instruction (µs)
Power_function:	ORG 500		
	MOV A, #00	1	1.085
	MOV R1, # Private key of sender/Receiver	1	1.085
	MOV B, # Global public key component G	1	1.085
Again:	ADD A, B	1	1.085
	DJNZ R1, Again	2	2.17

Total Time taken for executing the above code=(12/11.0592) * 3* private keys of sensors

So above is the total execution time saved at sensor nodes by shifting that multiplication function to the base station.

And if the sensor nodes are large enough, we can save substantial amount of energy as shown in the graph. Let the private key is 255 at both the end..Following Figure shows the time taken by the sensor for running above code as the number of sensors increases.



4.4 Strength of the proposed algorithm

1. Here We used the Elliptic Curve Cryptography algorithm with little modification on Sensor Network. The ECC algorithm depends for the effectiveness on the difficulty of the computing discrete logarithms. This increases the strength.

2. Why Our proposed scheme is better than Conventional Diffie-Hellman and RSA in Sensor Network? Generally all the public key cryptography algorithm is slower than symmetric key cryptography because it involves power function calculation that takes more execution time.
   Our proposed scheme is better than Diffie-Hellman because this algorithm involves at least two step power function calculation for generation of secret key. If the Sensor nodes perform this calculation it consumes more power. As well as RSA involve power function calculation every time it encrypt and decrypt the data so it consume large amount of energy every time.
   But ECC dos not involve power function calculation at sensor nodes as well as the by the above said steps it is clear that most of the calculation is done at the base station, which have enough power for the calculation. So it saves energy at sensor node including almost all the security services.

5. CONCLUSION

In this paper we have presented modified algorithms of DHA & ECC for secure key establishment for wireless Sensor Network. By our framework we identify those steps which are not necessary to perform on the sensor node, we implement those power functions on the base station.. Our proposed algorithm is not even securing the Network at some extend but it also helps to utilize the resources efficiently and also open the options that public key methods can be used for securing the Sensor Network. . In addition Sensor Network has many limitations like limited processing power, limited bandwidth and limited memory. Because of these limitations of Sensor Networks, it is subject to many attacks like active and passive attack, modification of the data, false data injection and node capture by the adversary. So providing Security to Sensor Network is important and challenging task because of its limitations. In this paper We provided the security algorithm to fulfill the above said security parameter at their cost at some extend. The result shows that public key methods can be implemented for securing the Sensor Network at low energy consumption. Our algorithm shows it after adding security by public key method energy

consumption is slightly increased compared to symmetric key cryptography. But my algorithm is limited in following two prospects.

It works only for those Sensor Network applications where node has slightly more energy resource compared than nodes have limited energy constrained, which is more rich than other sensor nodes in energy resource.

By the analysis it is clear that the energy consumption increases rapidly as the number of nodes increases, so it suits for limited numbers of nodes.

REFERENCES

[1] ARAIN PERRIG, ROBERT SZEWCZYK. SPINS. 2002.Security Protocols for Sensor Networks. Kluwer Academic Publishers.

[2] BARTOSZ PRZYDATEK, DAWN SONG, ADRIAN PERRIGO. 2003. SIA: Secure Information Aggregation in Sensor Networks .SenSys'03, ACM.

[3] DIFFIE, W., AND HELLMAN, M. 1976. Multiuser cryptographic Techniques." IEEE Transactions on information Theory. D. BALENSON, D. MCGREW, AND A. SHERMAN, 2000. Key Management for Large Dynamic Groups: One-Way Function Trees and Amortized Initialization, IETF Internet draf.

[4] H.CHAN AND A PERRIG, Security and Privacy in Sensor Networks, IEEE Computer society. JEFFERY UNDERCOFFER, SASIKANTH AVANCHA, ANUPAM JOSHI AMD JOHN PINKSTON. Security for Sensor Network MD 21250.

[5] JOSHI A. 2001. Load balancing, querying, Scheduling schemes in Mobile adhoc networks, M.S. Thesis, University of Cincinnati.

[6] KUROSE J, ROSS K, 2003. Computer Networking- A Top down approach Featuring the Internet, Second Edition Addison Weslay.

[7] L.ESCHENAUER AND V.GLIGOR. 2002. A Key-Management Scheme for Distributed Sensor Networks. In Proc.of ACM CCS 2002.

[8] POOSARLA R., 2003 Authenticated Route formation and Efficient Key management schemes for Securing Adhoc networks, M.S. Thesis, University of Cincinnati.

[9] WILLIAM STALLINGS, 1999. Cryptography and Network Security: Principles and Practice, Second Edition Prentice-Hall.

[10] Gaubatz, J. Kaps, and B. Sunar, "Public KeyCryptography in Sensor Networks", Security in Ad-hoc and Sensor Networks, pp. 2-18, 2005.)

[11] Ho®stein, J., Silverman, J., Whyte, W.: "NTRU report 012, version 2. estimated breaking times for NTRU lattices". Technical Report 12, NTRU Cryptosystems,Inc., Burlington, MA, USA (2003))

[12] Wander, A.S., Gura, N., Eberle, H., Gupta, V., and Shantz, S.C.,Energy Analysis of Public-Key Cryptography for Wireless Sensor Networks", In proceedings of PerCom pp. 324-328, 2005.

[13] JOHN PAUL WALTERS AND ZHENGQIANG LIANG. 2006. Wireless Sensor Network Security. In Security in Distributed, Grid and Pervasive Computing.

[14] SHISH AHMAD, RIZWAN .BEG, S.Q. ABBASS. 2010.Energy Efficient Sensor Network Security Using Stream Cipher Mode of Operation. IEEE ICCCT -2010.

[15] SHISH AHMAD, RIZWAN BEG, S.Q. ABBASS. 2010. Energy Saving Secure framework for Sensor Network using Elliptic Curve Cryptography. IJCA Special issue on MANTES.

[16] Chiasserini, C.F.,Chlamtac, I.,Monti,P.,Nucci,A.: Energy Efficient Design of wireless ad-hoc network.LNCS 2006, vol. 2345, pp. 376-38