Energy Saving Secure Framework for Sensor Network using Elliptic Curve Cryptography

Shish Ahmad
Research Scholar, CSE Dept,
Integral University
Lucknow, India

DR. Mohd. Rizwan beg
CSE Dept,
Integral University
Lucknow, India

Dr. Qamar Abbas
CSE Dept,
Integral University
Lucknow, India

ABSTRACT

Wireless networks of low-power sensing devices are poised to become a ubiquitous part of the computing landscape.In sensor network security, an important challenge is the design of protocols to bootstrap the establishment of a secure communications infrastructure from a collection of sensor nodes, which may have been pre-initialized with some secret information but have had no prior direct contact with each other.. Sensor nodes should be resilient to attacks. Since sensor nodes are resource constrained and run on battery, energy consumption should be low to make it operate for many days. In this paper we propose an energy efficient secure framework that proves the authentication, Integrity, and also provides secure communication among sensor nodes by using public key algorithms.

General Term

Security in Wireless sensor Network

Key Word

S*ecurity, Public Key Cryptography,* Elliptic Curve Cryptography, Sensor Network, attack, RSA.

[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 sensor nodes can be eavesdropped 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, an important challenge is the design of protocols to bootstrap the establishment of a secure communications infrastructure from a collection of sensor nodes which may have been pre-initialized with some secret information but have had no prior direct contact with each other. We refer to this problem as the bootstrapping problem. A bootstrapping protocol must not only enable a newly deployed sensor network to initiate a secure infrastructure, but it must also allow nodes deployed at a later time to join the network securely. The difficulty of the bootstrapping problem stems from the numerous limitations of sensor networks such as limited memory, limited processing power, limited bandwidth, lack of physical security, easy accessibility to adversaries.

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

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

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

Now in future the security in sensor network will be employed by 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.

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

Here we are presenting a Public key algorithm using Elliptic Curve Cryptography algorithm for preventing replay attack in sensor network as well as for data confidentiality and authentication between sensor nodes. Our work shows that our security mechanism successfully prevents 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 neighbours for secure communication as well as it can prove itself as a authenticated node of that network.

Low power consumption and reduction of memory size requirements of implementing secure communication and authentication algorithm.

[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: Authentication means 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: A passive attack is an attack where an unauthorized user monitors or listens in on the communication between two parties.

Active Attacks: Active attacks are attacks in which attacker is not only being able to listen to the transmission but also being able to actively modify or generate false data. Types of Active attacks are -

- Masquerade
- Replay:
- Denial of Service
- Modification of Message

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:

a. Spoofed, Altered, or Replayed Routing Information: Adversaries may be able to create routing loops, attract or repel network traffic, extend or shorten source routes, generate false error messages

b. Selective Forwarding: Malicious nodes may refuse to forward certain messages and simply drop them, ensuring that they are not propagated any further.

c. Sinkhole Attacks: In a sinkhole attack, the adversary's goal is to lure nearly all the traffic from a particular area through a compromised node, creating a metaphorical sinkhole with the adversary at the center.

d. The Sybil attack: In a Sybil attack, a single node presents multiple identities to other nodes in the network.

e. Wormholes: In the wormhole attack, an adversary tunnels messages received in one part of the network over a low latency link and replays them in a different part

f. Hello flood attack: Attacker broadcasting routing or other information with large enough transmission power could convince every node in the network that the adversary is its neighbour.

g. Acknowledgement Spoofing: An adversary can spoof link layer acknowledgments for "overheard" packets addressed to neighbouring nodes.

[3] RELATED WORK

Nodes have been 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

*Evaluation metrics***:** Listed below are several criteria that represent desirable characteristics for a bootstrapping scheme for sensor networks.

- Resilience against node capture
- Resistance against node replication
- Revocation
- Scalability

To secure our sensor network, we have broadly two scheme i.e. symmetric and asymmetric cryptography.

3.1 Key Distribution Techniques In Sensor Networks Using Symmetric Key Method

3*.1.1 Using a single network-wide key*: The simplest method of key distribution is to pre-load a single network wide key onto all nodes before deployment.

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

3*..1.3 Random key predistribution scheme:* By this method we can distribute keys to each of the sensor node before deployment by using Random Key Generator.

In addition a new security protocol have been introduced which is a symmetric method but have close property of asymmetric services.

3.1.4 SPINS has two secure building blocks: SNEP and μTESLA.

■ SNEP provides security primitives: Data confidentiality, two-party data authentication, and data freshness.

■ μTESLA is a new protocol which provides authenticated broadcast for severely resource-constrained environments.

SNEP offers the following nice properties:

■ *Semantic security*: Since the counter value is incremented after each message, the same message is encrypted differently each time. The counter value is long enough that it never

repeats within the lifetime of the node.

- *Data authentication:* If the MAC verifies correctly, a receiver can be assured that the message originated from the claimed sender.
- *Replay protection & Data Freshness*: The counter value in the MAC prevents replaying old messages. Note that if the counter were not present 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.2

µTESLA introduces asymmetry through a delayed disclosure of symmetric keys, which results in an efficient broadcast authentication scheme.

- TESLA requires that the base station and

nodes are loosely time synchronized, and each node knows an upper bound on the maximum synchronization error. 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. When a node gets a packet, it can verify that the corresponding MAC 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. At the time of key disclosure, the base station broadcasts the verification key to all receivers. When a node receives the disclosed key, it can easily verify the correctness of the key (explain below). If the key is correct, the node can now use it to authenticate the packet stored in its buffer.

$$\mathbf{K_i=F(K_{i+1})}$$

3.2 General consideration of using public key method:

The common perception of public key cryptography is that it is complex, slow and power hungry, and as not at all suitable for use in ultra-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 employ 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.

- Key Distribution Techniques In Sensor Networks Using Asymmetric Key Method (Public Key Method)

Once the nodes have been deployed, they perform key exchange. Nodes exchange their respective public keys and master key signatures. Each node's public key is verified as legitimate by verifying the master key's signature using the master public key. Once the public key of a node has been received, a symmetric link key can be generated and sent to it, encrypted by its public key. Upon reception of the session key, key establishment is complete and the two nodes can communicate using the symmetric link key.

3.2.1 Example-RSA

Let The Base station have the master public key (N_{bs},E_{bs}) *and private key* (N_{bs},D_{bs}).

Let A have public key (N_a,E_a) *and private Key* (N_a,D_a).

Let B have public key (N_b,E_b) *and private Key* (N_b,D_b).

Let Node A Want to Varify its Public key to node B
$Dcr_{Nbs,Ebs}(\ Enr_{(Nbs,Dbs)}\ (N_a,E_a)\ )$

Let B want to establish secure communication to A. B follow following Steps.

1. *After verification of the public key of A as above, B generate a Random Session Key* $K_{AB,}$
2. *It Encrypt the session key by applying the public key of A as follows.*

$$Enr_{\ (Na,Ea)}\,(K_{AB})$$

3. *B sends this message to A .*
4. *A find the Session Key by apply the decryption process by its private Key as follows*
$Dcr_{(Na,Da)}\,(.\ Enr_{\ (Na,Ea)}\,(K_{AB})\ )$

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.

3.2.2Rabin's scheme

It is based on the factorization problem of large numbers and is therefore similar to the security of RSA with the same sized modulus. Rabin's Scheme has asymmetric computational cost. The encryption operation is extremely fast, however decryption times are comparable to RSA of the same modulus.

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 E*n;b*(*x*)modulo*x*(*x* + *b*) mod *n*, as .

3.2.3 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 key 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 $\Phi(x)$ from *R* as above.

3. Compute the cipher text polynomial $c(x) = p\ \Phi(x) * h(x) + m(x)$ (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.

3.2.4 Improved Public Key Method

1. The scheme tries to solve the problem of security in WSN by the use of public key cryptography (RSA) as a tool 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)

3.3 Analysis

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

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

Table 1. This measurements on an Atmel ATmega128L low-power 8-bit microcontroller.

3.3.2 following result shows that the energy consumption using Elliptic curve cryptography is less than the RSA

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

Table 2. 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. Elliptic Curve Cryptography(ECC) algorithm in such a way that it consumes less energy as compared to the traditional methods.

[4] PROPOSED SCHEME

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 scheme keeps the data confidential from the source node to the base station at each step as follows.

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

© 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.

ECC minimized the above said attacks at the Sensor Network and prevents the following i,e,

# Confidentiality # Authentication
# Data Integrity #Replay attack
# Non repudiation source

4.2 Assumptions:

- Each Sensor Node has unique id.
- Sensor nodes are homogeneous and Static.
- It should be ensured that adversary cannot compromise Sensor Nodes immediately after nodes are deployed. The adversary takes a few minutes of time to compromise them after they are deployed.
- Each Sensor node has a comparator.

4.3 Scheme

Our scheme is based on Elliptic curve Cryptography algorithm for light weighted and resource constrained Sensor nodes with some modification[Blak99]. 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.

Phase1: Before deployment of the Sensor Nodes

1. The Base Station select a large integer q, which is either prime number p or an integer of the form 2^m and elliptic curve parameter a and b for following equation.

$Y^2 + xy = x^3 + ax^2 + b$.

This defines the elliptic group of points $E_q(a,b)$. 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 $n_1, n_2 \ldots . n_N$ 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.

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.

6. 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.

1. Now if a new Sensor Node is deploys to the exiting one with the same values that is n_R, P_R and G It exchanges the public value P_R 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.

4.4 Results and Analysis

Strength of the proposed algorithm :

1. Here We used the Elliptic Curve Cryptography agorithm 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 a secure algorithm 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 scheme is not even securing the Network at some extent 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. But Sensor Networks are deployed in unattended environment where there is lack of physical security and easily accessible to adversary. 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 proposed scheme shows that 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.

1. It works only for those Sensor Network applications where node has slightly more energy resource compared than nodes have limited energy constrained like iMote Sensor node which is more rich than other sensor nodes in energy resource.
2. By the analysis it is clear that the energy consumption increases rapidly as the number of nodes increases, so it limit the number of nodes.

References:

1. Arain Perrig, Robert Szewczyk. SPINS: Security Protocols for Sensor Networks". 2002 Kluwer Academic Publishers.

2. Bartosz Przydatek, Dawn Song, Adrian Perrigo "SIA:

Secure Information Aggregation in Sensor Networks" .SenSys'03, ACM 2003.

3. Blake, I.; Seroussi, G.; and smart, N. Elliptic Curves in Cryptography. Cambridge; Cambridge University Press, 1999.

4. D. Balenson, D. McGrew, and A. Sherman, "Key Management for Large Dynamic Groups: One-Way Function Trees and Amortized Initialization," IETF Internet draft, August 2000.

5. H.Chan and A Perrig, "Security and Privacy in Sensor Networks," IEEE Computer, October 2003.

6. L.Eschenauer and V.Gligor. "A Key-Management Scheme for Distributed Sensor Networks". In Proc.of ACM CCS 2002.

7. Poosarla R., "Authenticated Route formation and Efficient Key management schemes for Securing Adhoc networks", M.S. Thesis, University of Cincinnati, 2003

8. William Stallings, "Cryptography and Network Security: Principles and Practice, Second Edition Prentice-Hall 1999 .

9. John Paul Walters and Zhengqiang Liang. Wireless Sensor Network Security. In Security in Distributed, Grid and Pervasive Computing, 2006.

10. The Case for Elliptic Curve Cryptography, http://www.nsa.gov/ia/industry/crypto elliptic curve.cfm Dated:31- 05-2008

11. Willliam Stallings " Cryptography and Network Security" Principles and Practice" fifth Edition.

12. Chungen, Yanhong Ge, "The Public Key Encryption to Improve the Security on Wireless Sensor Networks" 2009 Second International Conference on Information and Computing Science,978-0-7695-3634-7/09 $25.00 © 2009 IEEE,DOI 10.1109/ICIC.2009.

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

14. 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))

15. David Carman, Vipin Swarup, Daniel Coffin, Ronald Watro, Bruno Dutertre "Security for Wireless Sensor Networks" Foroum Session, 19th Annual Computer Security Applications Conference, December 08

16. 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.