International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:4, No:6, 2010

Authentication Protocol for Wireless Sensor Networks

Sunil Gupta, Harsh Kumar Verma and AL Sangal

***Abstract*—**Wireless sensor networks can be used to measure and monitor many challenging problems and typically involve in monitoring, tracking and controlling areas such as battlefield monitoring, object tracking, habitat monitoring and home sentry systems. However, wireless sensor networks pose unique security challenges including forgery of sensor data, eavesdropping, denial of service attacks, and the physical compromise of sensor nodes. Node in a sensor networks may be vanished due to power exhaustion or malicious attacks. To expand the life span of the sensor network, a new node deployment is needed. In military scenarios, intruder may directly organize malicious nodes or manipulate existing nodes to set up malicious new nodes through many kinds of attacks. To avoid malicious nodes from joining the sensor network, a security is required in the design of sensor network protocols. In this paper, we proposed a security framework to provide a complete security solution against the known attacks in wireless sensor networks. Our framework accomplishes node authentication for new nodes with recognition of a malicious node. When deployed as a framework, a high degree of security is reachable compared with the conventional sensor network security solutions. A proposed framework can protect against most of the notorious attacks in sensor networks, and attain better computation and communication performance. This is different from conventional authentication methods based on the node identity. It includes identity of nodes and the node security time stamp into the authentication procedure. Hence security protocols not only see the identity of each node but also distinguish between new nodes and old nodes.

***Keywords*—**Authentication, Key management, Wireless Sensor network, Elliptic curve cryptography (ECC).

I. Introduction

SECURITY allows WSNs to be used with assurance. Without security, the use of WSN is any application area would cause in undesirable consequences. Wireless sensor networks are rapidly gaining regard due to low cost solutions to a variety of real world challenges. The basic idea of sensor network is to disperse tiny sensing devices, which are able for sensing some changes of incidents/parameters and communicating with other devices, over a specific geographic area for some particular purposes like surveillance, environmental monitoring and target tracking.

Now a day's sensors can monitor pressure, temperature, humidity, soil makeup, noise levels, vehicular movement, and lighting conditions, the presence or absence of certain kinds of objects or substances, mechanical stress levels on attached objects, and other properties [4]. In case of wireless sensor network, the communication among the sensors is done using wireless transceivers. Basically the major challenge for employing any efficient security scheme in wireless sensor networks is created by the size of sensors, as a result the memory, processing power and type of tasks affected from the sensors. To deal with the important security issues in wireless sensor networks we talk about cryptography, steganography and other basics of network security and their applicability. We investigate various types of threats and attacks against wireless sensor network to save manufacturing cost. A sensor node is usually built as a small device, which has limited memory, a low-end processor, and is powered by a battery. So during the design of any security solution we need to take care of resource constraints like limited energy, limited memory, limited computing power, limited communication bandwidth, limited communication range.

The type of security mechanism that can be hosted on a sensor node platform is dependent on the capabilities and constraints of sensor node hardware. After months of operation or a several weeks, some of the nodes in the network may weaken their power because of the irregular distribution of traffic load. so new node deployment is needed in this case. Besides the natural loss of sensor nodes, a sensor network is also vulnerable to malicious attacks in unattended and hostile environments. Some of the sensor nodes may be destroyed by opponent, so that the entire network may become useless. So, new sensor nodes have to be deployed. on the other hand, an opponent can also position a malicious nodes into the network. These malicious nodes may insert false reports and eavesdrop messages. Recently many schemes [1–7] were proposed to defend the sensor networks. It may prevent external attackers from inserting false reports or eavesdropping messages. But, they can hardly protect against internal attacks [8-12] .In this paper; we evaluate the internal attacks in [8-12]. We observe that these attacks manipulate existing nodes to introduce malicious ‘‘new’’ nodes, which are indistinguishable from legitimate new nodes under current sensor network security technology. Those introduced ‘‘new’’ nodes could be accepted by other normal nodes as legitimate ones. Based on this observation, we design a protocol for sensor networks to prevent malicious nodes. However, most of previously proposed key predistribution schemes cannot be easily implemented as a dynamic access control because all the old secret keys and broadcasting messages of existing nodes should be updated once a new node is added [4,13,14,15]. We introduce the node Security time stamp, which is the time when the new node join the sensor network, into the authentication procedure to differentiate malicious ‘‘new’’ nodes, which are actually old nodes, from legitimate new nodes. Moreover, key establishment is also included in our authentication protocol to help the new node establish shared keys with its neighbors so that it can perform secure communications with them. Com-pared to RSA, ECC can achieve the same level of security with smaller key size. It has been known that 160-bit ECC provides comparable security to 1024-bit RSA and 224-bit ECC provides comparable security to 2048-bit RSA [16] Hence, under the same security level,

smaller key sizes of ECC offer merits of computational efficiency, as well as memory, and bandwidth saving. It is better suited for the resource constrained devices. Owing to the merits of ECC, this Protocol is based on elliptic curve cryptography (ECC).

II. Review Attacks In WSN

A large-scale sensor network consists of thousands of sensor nodes and may be dispersed over a wide area. Typical sensor nodes are small with limited communication and computing capabilities, and are powered by batteries. These small sensor nodes are susceptible to many kinds of attacks. Attacks against wireless sensor networks could be broadly considered from two different levels of views. One is the attack against the security mechanisms and another is against the basic mechanisms.

A. Passive Information Gathering: An opponent with powerful assets can collect information from the sensor networks if it is not encrypted [17].

B. Node Subversion: Capture of a node may tell its information including disclosure of cryptographic keys and thus compromise the whole sensor network [17].

C. False Node: A false node involves the addition of a node by an opponent to inject malicious data, whereby the false node is computationally robust enough to lure other nodes to send data to it [17].

D. Node Malfunction: A malfunctioning node will generate inaccurate data which could put at risk to the integrity of sensor network especially if it is a data aggregating node such as a cluster head [17].

E. Node Outage: Node outage is when a node stops its function. In the case where a cluster head stops functioning, the sensor network protocols should be robust enough to moderate the effects of node outages by providing an alternate route. [17].

F. Message Corruption: Any modification of the content of a message by an attacker compromises its integrity [17].

G. Traffic Analysis: Even when the messages transferred are encrypted, it still leaves a high possibility analysis of the communication patterns and sensor activities can potentially reveal enough information to enable an adversary to cause malicious harm to the sensor network [17].

H. The Sybil attack: In a Sybil attack, a single node presents several identities to other nodes in the network. They pose a significant risk to geographic routing protocols, where location aware routing requires nodes to exchange coordinate information with their neighbors to efficiently route geographically addressed packets. Authentication and encryption techniques can prevent an outsider to launch a Sybil attack on the sensor network. However, an insider cannot be prevented from participating in the network, but it should only be able to do so using the identities of the nodes has compromised. Using globally shared keys allows an insider to masquerade as any node [18,19].

I. Sinkhole attacks: In a sinkhole attack, the goal of opponent is to tempt nearly all the traffic from a particular area through a compromised node, creating a metaphorical sinkhole with the opponent at the center. Sinkhole attacks typically work by making a compromised node look especially attractive to surrounding nodes with respect to the routing algorithm [3].

J. Wormholes: In the wormhole attack, an opponent messages received in one part of the network over a low latency link and replays them in a different part. The simplest instance of this attack is a single node situated between two other nodes forwarding messages between the two of them. on the other hand, wormhole attacks more commonly involve two distant malicious nodes colluding to understate their distance from each other by relaying packets along an out-of-bound channel available only to the Attacker [20].

III. Preliminaries

Before a new conference scheme is proposed, we first introduce for secure authentication and privacy transactions and for secure messaging, an efficient public key communications is needed. The basic choices for public key systems are existing for these applications are:

- RSA
- Diffie-Hellman (DH) or Digital Signature Algorithm (DSA)
- Elliptic Curve Diffie-Hellman (ECDH) or Elliptic Curve Digital Signature Algorithm.

RSA is a system that was published in 1978 by Rivest, Shamir, and Adleman, based on the complexity of factoring large integers. Whitfield Diffie and Martin Hellman proposed the public key system now called Diffie-Hellman Key Exchange in 1976. DH is key agreement and DSA is signature, and they are not directly inter-changeable, although they can be combined to do authenticate key agreement. Both the key exchange and digital signature algorithm are based on the difficulty of solving the discrete logarithm problem in the multiplicative group of integers modulo a prime p. Elliptic curve groups were proposed in 1985 as a substitute for the multiplicative groups modulo p in either the DH or DSA protocols. For the same level of security per best currently known attacks, elliptic curve-based systems can be implemented with much smaller parameters, leading to significant performance advantages.

Such performance improvements are particularly important in the wireless area where memory, computing power, and battery life of devices are more constrained. Prior to a new scheme based on elliptic curves is proposed, we first introduce the properties of elliptic curves that will allow us to discuss the security of the proposed scheme in security analysis [21,22].

Generally elliptic curve given by

$y^2 \bmod P = (x^3 + ax + b) \bmod P$.

The properties of elliptic curves that will allow to the security is

1) If $P=(x, y) \neq O$, then $-P=(x, -y)$.

2) If the number of elements on Equation is n, then for every point P on Equation, it has $nP=0 \bmod q$.

3) Suppose that two points P1=(x1, y1) and P2=(x2, y2).
The rules are as follows:
If x1=x2 mod q, then P1+P2=O. If y1= 0mod q, thenP1= −P1 and 2P1=O.
In other cases, the sum P1+P2 is obtained by computing
$\lambda = (x1-x2/ y1-y2) \bmod q$,
if P1≠P2, or $\lambda = (3x_1^2 + 2ax_1 + b / 2y_1) \bmod q$,
if P1 = P2, and then let $x3 = \lambda 2 - a - x1 - x2 \bmod q$.

IV. Our Protocol

The proposed scheme mainly introduces the security authentication protocol for deployment of new node shown in figure 1 at the commencement of a network, a lot of sensor nodes are deployed in a nominated area. New node deployment is unavoidable because nodes in a sensor network may be lost or destroyed. In this Protocol, we will develop authentication mechanism for new nodes joining sensor networks. Without loss of generality, the proposed method would achieve two tasks:
1. Authentication of new nodes
2. Establishment of key: during authentication, a shared keys should be created between a neighboring nodes and deployed node to provide a secure communication.

The proposed protocol consists of an initialization phase and an authentication phase. The basic idea is stated as follows.

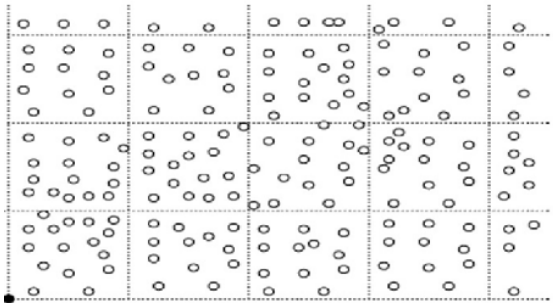


Fig. 1 Node Deployment model

We use Asymmetric technique to provide better authentication and public key certificate to prove the identity of a new node, when the new node is deployed into the sensor network.

A. Assumptions

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 preset security time stamp.
3. Two sensor nodes may have the same Security Time Stamp if they are deployed simultaneously.
4. The base station acts as a controller, is secure, trustworthy and has powerful resources in terms of energy, memory and computation.
5. All the sensor nodes are similar in terms of energy, memory and computation capabilities.
6. The sensor nodes have enough memory to manage with the keying overheads.
7. The nodes sleeping pattern depends on the available energy and event detection.
8. Due to a wireless medium an opponent can listen to the entire network traffic once the network is compromised.

B. How scenario works

Figure 2 shows “b” want to join a network and Its neighbor node is “a”. So “b” have to prove its authentication to join the network Authentication is provided by certificate in addition with security time stamp. Time stamp contains the time to join the network provided by the authority. We say it as bootstrap time.

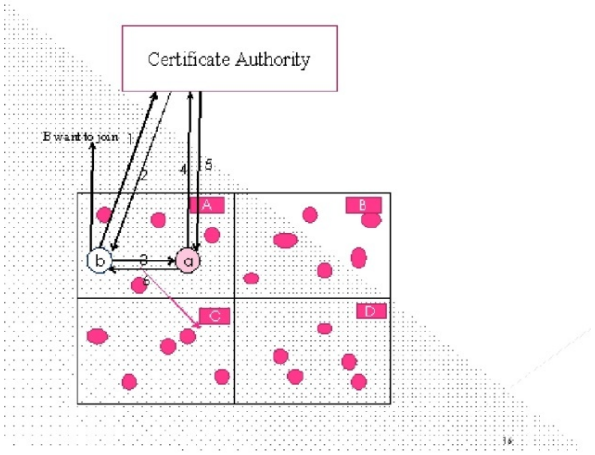


Fig. 2. Authentication scenario

For Authentication between Two nodes (B new node and A old node).

1. B” request to certificate authority for certificate and security time stamp
2. Authority provide a certificate with time stamp to “B” for access network
3. “B” sends this certificate to its neighbor for authentication
4. “A” checks the security time stamp by using certificate
5. “A” checks the validity and identification for authentication of “B” by using authority certificate
6. Exchange the information

C. Complete Framework

Our cryptographic tool uses the Elliptic Curve Cryptography with Diffie Hellman scheme and use, DSA to verify the signature. The reason is that the ECC can attain the same level of security with smaller key sizes. It has been known that 160-bit ECC provides equivalent security to 1024-bit RSA and 224-bit ECC provide similar security to 2048-bit RSA. Under the same security level, smaller key sizes of ECC offer merits of faster computation [16]. Figure 3 shows the complete framework to provide authentication.

D. Predeployment phase

Before a sensor network is deployed, the Certificate authority chooses a set of network parameters including:

1. A finite field F_q, where q is a large odd prime of at least 160 bits.
2. An elliptic curve E over F_q Ep (a ,b) - elliptic curve with parameter a,b and p is prime.
3. G: A cyclic group point on elliptic curve whose order is large value *n* of at least 160 bits.
4. The CA's private key k= {1, 2, 3……, n-1}.
5. The CA's public key Q=kG and the CA never shares its private key with anyone else.

E. Protocol

Steps at Node B

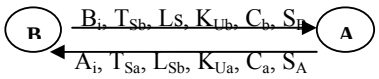
1. B first selects private key n_b and calculates public key $Q=K_{Ub} = n_b*G$
2. B sends its public key to the certificate authority CA.
3. Certificate authority provides certificate to B that include C_b= [B_i, T_e, T_{Sb}, Ls, K_{Ub}].
4. B sends this message using signature to A as, $S_B = K^{-1}(H (B_i||T_{Sb}|| Ls ||K_{Ub})+ n_b* C_b)(mod p)$
   So signature become (C_b , S_B)

Node A checks the validity of security time stamp and identity by verifying the signature.

Steps at Node A

5. A first compare B's security time stamp T_{Sb} With its own security time stamp T_{Sa}. If $T_{Sb} >= T_{Sa}$ then node B might be a new node.
6. To authenticate a new node A proceed to verify node B is a new node by comparing T_{Sb} with its current time t. If T_{Sb} is out of date { $T_{Sb\_}t$} > L_{Sb} node A simply drop the message. Otherwise node A continues to verify B's identity.

If V= C_b node A can make sure that node B is a legitimate new node



Steps at B	**Steps at A**
if $T_{Sa} >= T_{Sb}$	if $T_{Sb} >= T_{Sa}$
if { $T_{Sa\_}t$} > $L_{Sa,}$ reject A_i ;	if { $T_{Sb\_}t$} > $L_{Sb,}$ reject B_i ;
else { calculate verification V;	else { calculate verification V;
if V= C_a { accept A_i ;	if V= C_b {accept B_i ;
calculate K_{AB};}	calculate K_{AB};}
else reject A_i ;}	else reject B_i ;}

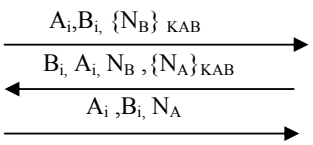


Fig. 3 The Proposed scheme

Verification

$U_1 = H (A_i|| T_{Sb}|| Ls|| K_{Ub})$

$U_2 = S_B^{-1}U_1 (mod p)$

$U_3 = S_B^{-1}n_A (mod p)$

$V = U_2G + U_3Q$

IF V = C_b then node A can make sure that node B is legitimate new node.

$V= U_2G + U_3Q = S_B^{-1}U_1 (mod p)G + S_B^{-1}n_A (mod p)Q$

$= S_B^{-1} H (A_i|| T_{Sb}|| Ls|| K_{Ub}) (mod p) G + S_B^{-1}n_A (mod p) kG$

$= S_B^{-1}(H (A_i|| T_{Sb}|| Ls|| K_{Ub}) +K n_A) G = K_BG= C_b$

V. Security Analysis

Our protocol use ECC quite than RSA.

A. Size trend: A key size selection is one of the important decisions for cryptography. The size of the key actually refers to the size (in bits) of the modulus, N, not the size of any of the public or private keys. Two randomly selected primes, p and q, should be chosen such that they are approximately of the same length to ensure that any attempts to factor the modulus are much more difficult. Key sizes for public-key cryptosystems should be much larger than private-key cryptosystems and so comparisons between the two should not be made. The decision of the key size to be used should be based on a thorough assessment of the security solution requirements for the cryptosystem. This entails an evaluation of the value of the data to be protected as well as the length of time for which it needs to be protected. A corresponding factor is also an appraisal of who might wish to devise such an attack as well as what resources they have available. A best guess can then be made based upon the extrapolation of hardware advances, to hypothesize the computational time possible to break the cryptosystem as well as the cost such a design would involve. Increasing the key size as shown in Table 1 will also cause a corresponding increase in the computational load.

TABLE I COMPARISON OF SECURITY LEVELS

Year	Key Size
1970	39
1978	45
1981	47
1982	51
1983	63
1984	71
1986	87
1987	90
1988	100
1990	111
1991	116
1992	129
1996	130
1998	140
1999	155
2000	162
2002	168
2004	172
2008	192
2009	212

Figure 4 shows the graphical result as increase load and figure 5 shows the increase in difficulty as key size increase. A rough approximation is that doubling the length of the key size will increase public key operations by a factor of four and private key operations by a factor of eight. Public key operations are less sensitive to key size increase because the public exponent can be fixed, while in private key operations the length of the private exponent increases proportionately

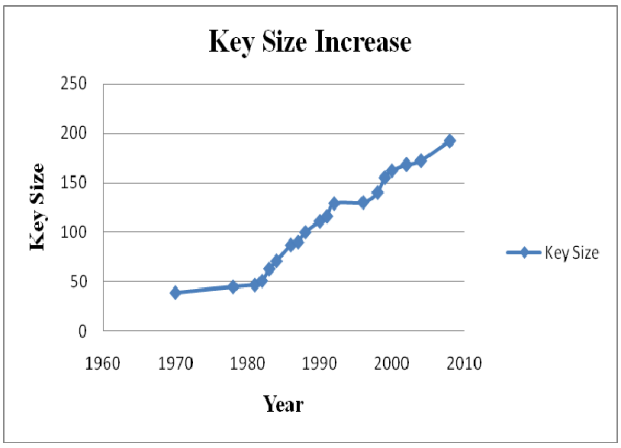


Fig. 4 Size trends

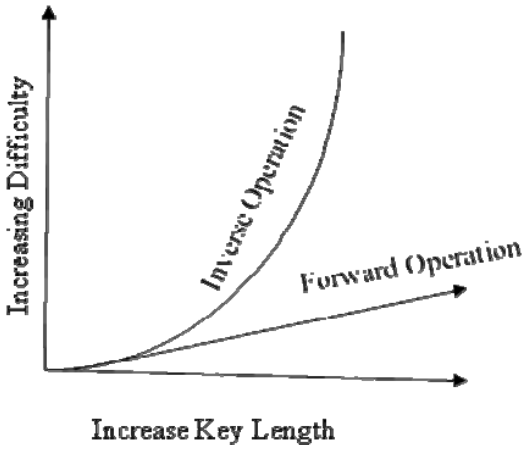


Fig. 5 Difficulty of forward, inverse operation against key length

B. Security levels: At present, for the same level of opposition against the best known attacks, the system parameters for an elliptic-curve-based system can be chosen to be much smaller than the parameters for RSA or mod p systems. For example, an elliptic curve over a 163-bit field currently gives the same level of security as a 1024-bit RSA modulus or Diffie-Hellman prime. The difference becomes even more remarkable as the desired security level increases. For example, 571-bit ECC is currently equivalent in security to 15,360-bit RSA/DH/DSA. Public key protocols are used in combination with symmetric key algorithms.

TABLE II COMPARISON OF SECURITY LEVELS

S.No	Security (Bits)	Symmetric Encryption Algorithm	RSA key Size	ECC key Size	Key Size Ratio	Time to break (MIPS Years)
1	80	Skipjack	1024	160	1:6	3 millions year
2	112	3DES	2048	224	1:9	10^{16} Year
3	128	AES-128	3072	256	1:12	1E+12
4	192	AES-192	7680	384	1:20	1E+ 20
5	256	AES-256	15360	512	1:30	1E +36

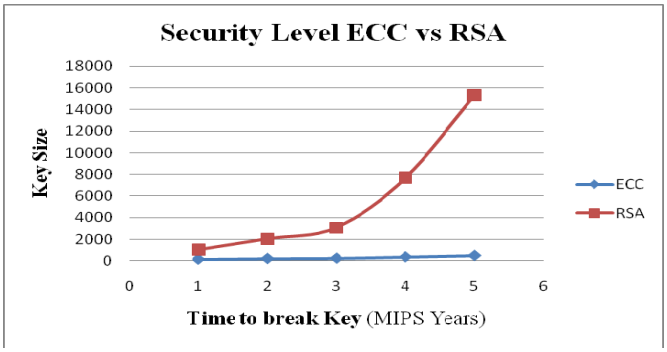


Fig. 6 Comparison Of security level

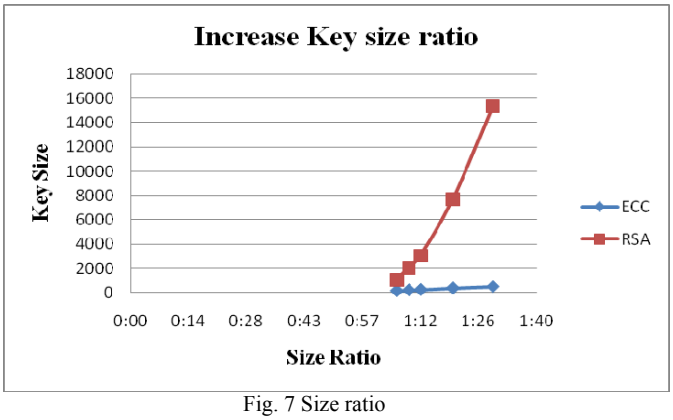


Fig. 7 Size ratio

Table 2 is found in a number of the standards documents [21, 22]. This growing difference in key bit length for equivalent security levels accounts for the performance advantages to be obtained from substituting ECC for RSA/DH/DSA in public key cryptographic protocols. The comparison of these security levels is described by Figure 6 and Figure 7.

In this protocol, the most exclusive operation is the point multiplication of the form kP for k Є Z_n and P Є G.In this every sensor node needs to perform only three point multiplications over an elliptic curve:

- Two for node authentication
- One for key establishment.

TinyPK [7] use RSA for authentication from external parties and Diffie–Hellman over DLP is to establish shared keys between external parties and sensor nodes. It require three modular exponentiation operations over integer rings for each sensor node: one RSA public key operation and one RSA private key operation for node authentication and one DLP operation for key establishment.

It has been made known in [7,23] that a point multiplication wishes less computation time than a modular exponentiation.

A. Comparison with Related Work

At present no solutions can avoid node compromise in sensor network, the best we can do is to limit the contact of node compromise to the locality of the compromised nodes, i.e., avoid opponent from launching network-scale attacks based on compromised nodes. The majority of symmetric key techniques, including ID based keys [3], randomly predistributed keys [4], and location-based keys [5,6] try to improve the flexibility to node compromise by increasing the least number of sensor nodes that an opponent needs to compromise to destroy the entire network security architecture. These schemes can tolerate a certain number of compromised nodes. The protocol may prevent opponent from distribution of the impact of node compromise by launching the Sybil attack, but it cannot notice the node replication attack since the copies of the compromised nodes also have legitimate certificates.

VI. Conclusion

This paper proposes an authentication protocol by using Elliptic curve cryptography to manage a new node joining the sensor network, which is not only legitimate but also truly new to the sensor network. We establish the node security time stamp into the authentication procedure to distinguish malicious nodes from legitimate new nodes. Our authentication protocol can avoid malicious nodes from joining sensor networks at establishment of network. In adding up, key establishment by using elliptic curve is also realized in our protocol to help the new node establish shared keys with its neighbors so that it can perform secure communications with better key management Compared with the RSA, DH, DSA our protocol with ECC can protect against most of the infamous attacks in sensor networks, and achieve better communication and computation performance.

References

[1] Perrig, R. Szewczyk, J.D. Tygar, V. Wen, D.E. Culler, SPINS: Security protocols for sensor networks, Wireless Networks 8 (September) (2002) 521–534.

[2] Karlof, N. Sastry, D. Wagner, TinySec: a link layer se-curity architecture for wireless sensor networks, in: The Second ACM Conference on Embedded Networked Sen-sor Systems (SensSys'04), Baltimore, Maryland, Novem-ber 2004.

[3] L. Eschenauer, V. Gligor, A key management scheme for distributed sensor networks, in: The Proceedings of the 9th ACM Conference on Computer and Communications Se-curity (CCS'02), Washington DC, 2002.

[4] Haowen Chan, Adrian Perrig, Dawn Song, Random key predistribution schemes for sensor networks, in: Proceed-ings of the 2003 IEEE Symposium on Security and Priva-cy (S&P'03), 11–14 May 2003, p. 197.

[5] W. Du, J. Deng, Y.S. Han, P.K.Varshney, A pairwise key pre-distribution scheme for wireless sensor networks, in: The Proceedings of the 10th ACM Conference on Com-puter and Communications Security (CCS'03), Washing-ton, DC, 27– 30 October 2003.

[6] Liu, P. Ning, Establishing pairwise keys in distributed sensor networks, in: The Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS'03), Washington, DC, 2003.

[7] R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, P. Kruus, TinyPK: securing sensor networks with public key technology, in: Proceedings of the 2nd ACM Workshop on Security of Ad hoc and Sensor Networks (SASN'04), Washington, DC, USA, 25 October 2004.

[8] J. Newsome, E. Shi, D. Song, A. Perrig, The sybil attack in sensor networks: analysis & defenses, in: The 3rd In-ternational Symposium on Information Processing in Sen-sor Networks (IPSN'04), Berkeley, California, USA, 26– April 2004.

[9] B. Parno, A. Perrig, V. Gligor, Distributed detection of node replication attacks in sensor networks, in: IEEE S&P'05, 2005.

[10] Y. Hu, A. Perrig, D.B. Johnson, Pachet leashes: a defense against wormhole attacks in wireless networks, in: IEEE INFOCOM'03, 2003.

[11] L. Hu, D. Evans, Using directional antennas to prevent wormhole attacks, in: The 11th Annual Network and Dis-tributed System Security Symposium (NDSS'04), San Di-ego, California, 5–6 February 2004.

[12] W. Wang, B. Bhargava, Visualization of wormholes in sensor networks, in: Proceedings of the 2004 ACM Work-shop on Wireless Security (Wise'04), Philadelphia, PA, USA, 1 October 2004.

[13] S.J. Choi, H.Y. Youn, An efficient key pre-distribution scheme for secure distributed sensor network, The 2005 IFIP International Conference on Embedded and Ubiquit-ous Computing (EUC'2005), LNCS 3823, 2005, pp. 1088–1097.

[14] L. Eschenauer, V.D. Gligor, A key-management scheme for distributed sensor networks, Proceedings of the 9th ACM Conference on Computer and Communication Se-curity, 2002, pp. 41–47.

[15] C.W. Park, S.J. Choi, H.Y. Youn, A novel key pre-distribution scheme with LU matrix for secure wireless sensor networks, International Conference on Computa-tional Intelligence and Security (CIS 2005), Springer-Verlag, Germany, 2005, pp. 494–499, LNAI. 3801, Part I, Dec.

[16] S. Vanstone, Responses to NIST's proposal, Communica-tions of the ACM 35 (July1992) 50–52.

[17] Tanveer Zia and Albert Zomaya,"A security Framework for Wireless Sensor Networks", IEEE Applications Sym-posium, Houston, Texas USA, February 2006.

[18] J. Newsome, E. Shi, D. Song, A. Perrig, The sybil attack in sensor networks: analysis & defenses, in: The 3rd International Symposium on Information Processing in Sen-sor Networks (IPSN'04), Berkeley, California, USA, 26–April 2004.

[19] J.R. Douceur, ''The Sybil attack,'' First International Workshop on Peer-to-Peer Systems (IPTPS'02), March 2002.

[20] Y. Hu, A. Perrig, D.B. Johnson, Pachet leashes: a defense against wormhole attacks in wireless networks, in: IEEE,. 417–426.INFOCOM'03, 2003

[21] N. Koblitz, Elliptic curve cryptosystems, Mathematics of Computation 48 (1987) 203–209.

[22] Miller, Uses of elliptic curves in cryptography, Ad-vances in Cryptology- CPYPTO'85, Lecture Notes in Computer Science, vol. 218, Springer-Verlag, 1986, pp. 417–426.

[23] David J. Malan, Matt Welsh, Michael, and D. Smith, "A publickey infrastructure for key distribution in TinyOS based on elliptic curve cryptography," in *Proceedings of First IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON'04)*, October 2004, pp.145-161.

[24] Crossbow Technology. Available from: <http://www.xbow. com/>.

[25] N. Gura, A. Patel, A. Wander, H. Eberle, and S.C. Shantz, "Comparing elliptic curve cryptography and RSA on 8-bit CPUs," *in: CHES'04*, 2004, pp.67-75.

[26] Atmel Corporation. Available from: <http://www.atmel.- com/>.

Mr. Sunil Gupta has done his Bachelor's degree in computer Science and Master's degree in Computer Science and Engineering from National Institute of Technology Hamirpur. Presently he is working as a research scholar in the area of Information Security at National Institute of Technology Jalandhar India.

Dr Harsh Kumar Verma is currently working as Associate Professor in the department of Computer Science and Engineering at Dr B R Ambedkar National Institute of Technology Jalandhar. He has done his Bachelor's degree in Computer Science and Engineering in May 1993. He did Master's degree in Software Systems from Birla Institute of Technology Pilani in Feb 1998 and Ph.D. from Punjab Technical University Jalandhar India in May 2006. He is presently working in the area of Information Security, Computer Networks and Scientific Computing. He has many publications of international /national level to his credit. e-mail:harsh_verma@yahoo.com

Dr A L Sangal is currently working as Professor in the department of Computer Science and Engineering at Dr B R Ambedkar National Institute of Technology Jalandhar. He has done his Bachelor's degree in Electronics and Communication Engineering from Panjab Engineering College Chandigarh. He did Master's degree in Computer Science from Thapar Institute of Engineering and Technology Patiala and Ph.D. from National Institute of Technology Jalandhar India. He is presently working in the area of Computer Networks, Scientific Computing and Information Security. He has numerous publications of international/national level to his credit.