Asian Journal of Information Technology 12 (8): 250-256, 2013
ISSN: 1682-3915

MultiSECPKC: An Authentication Protocol Based on Public Key Cryptography

[1]P. Vijaya Lakshmi, [2]D. Somasundareswari and [3]V. Duraisamy
[1]Department of ECE, Hindusthan College of Engineering and Technology, Tamilnadu, India
[2]Department of Electrical Sciences, Adithya Institute of Technology, Tamil Nadu, India
[3]Maharaja Institute of Technology, Tamil Nadu, India

Abstract: Providing security service in wireless sensor networks using authentication techniques turns out to be a non-trivial task due to network resource constraints. To improve the network security it is often necessary to combine intrusion detection techniques with the key management protocols. Several symmetric key cryptographic solutions proposed earlier are inefficient against node compromise attacks due to delay in the disclosure of secret keys. On the other hand, conventional public key cryptographic techniques suffer from large key sizes introducing large communication overhead and delayed authentication. Researchers propose MultiSECPKC, a highly reliable authentication protocol based on public key cryptography to provide an energy efficient secured environment against node compromise attacks which would otherwise waste the energy of network resources. Any compromised node is detected and isolated in the initial stage itself with minimum overheads and energy consumption. The best and fresh route is formed using intermediate nodes between the source and the sink. In addition, this scheme exploits the use of only one ECDSA signature to authenticate the messages which in turn reduces the overhead of the sink in performing signature verification. Simulation results show that the proposed scheme can provide very low overheads in terms of storage, computation and communication with high energy saving and high reliability. The proposed scheme is more resilient to node compromise attacks and can be applied for different routing techniques for sensor networks.

Key words: Authentication, security, compromised nodes, wireless sensor networks, ECDSA

INTRODUCTION

A Wireless Sensor Network (WSN) in general is a collection of small, low-cost and low battery powered sensor nodes that communicate with each other through wireless link under highly resource constrained hostile environment. These networks find wide applications in the areas of healthcare, environment monitoring and military. Designing security services such as authentication and key management are critical to provide a secure communication between sensor nodes such hostile environments.

LITERATURE REVIEW

Several research surveys were conducted on various public key authentication schemes (Baek *et al*., 2008) and also key distribution schemes (Camtepe and Yener, 2005) used in wireless sensor networks. A key requirement for WSN is source node authentication (Akyildiz *et al*., 2002). Remote user authentication schemes were introduced based on source node identity (Das *et al*., 2004) and also for multicast environments (Leung *et al*., 2003). Techniques for pre-distribution of keys based on group re-keying was proposed to reduce the number of keys used (Zhang *et al*., 2009). Most of the authentication techniques focus on protocol implementations in the network and link layers. Some researches based on RSA algorithms showed optimistic results (Wander *et al*., 2005). TinyPK authentication scheme employs RSA and Diffie-Hellman algorithms to calculate an encrypted public key (Watro *et al*., 2004). This protocol is open to hostile attack like spoofing. Later, a less complex, light-weight, dynamic authentication protocol using a hash function and security features of the IEEE 802.15.4 MAC layer was proposed (Wong *et al*., 2006). TESLA, a secure source authentication for multicast had the problem of time synchronization and delayed authentication (Perrig *et al*., 2001a, b). Later on the same researchers (Perrig *et al*., 2001a, b) proposed a modification in TESLA which allows receivers to authenticate most packets as soon as they arrive.

Several other schemes were also proposed to improve the authentication at thebroadcast level (Chang *et al*.,

Corresponding Author: P. Vijaya Lakshmi, Department of ECE, Hindusthan College of Engineering and Technology, Tamilnadu, India

2006; Ren *et al*., 2010). Public key authentication schemes applied to WSNs had the drawback of large overhead in terms of computational, communication and storage except for some specific cryptographic applications based on elliptic curve primitives (Wang and Li, 2006; Watro *et al*., 2004). Private key agreement schemes proved to be more secure forhetero generous sensor networks (Rahman and El-Khatib, 2010). In recent years, the use of pairing-based cryptographic schemes in WSN environments has beenproposed for stronger protection. These techniques showed faster response using and smaller sized keys (Xiong *et al*., 2010).

However, such schemes still have to address the issues of key revoking since, the sensor nodes can easily be compromised. Researchers propose a novel user authenticationtechnique for wireless sensor networks using ECDSA and MD5 hash function to provide high security for a node joining the network and also for the data transmission between the source and the sink. It is assumed that there is only one sink that is endowed with sufficient energy supply. The sink can not be compromised by any adversaries and it has a secure mechanism to authenticate its messages to all the nodes in the network. Researchers also assume every sensor node has an individual secret key shared with the sink to verify the received messages. Further, there is a unique pair wise key shared between each pair of neighbouring nodes. Contrary to the sink, sensor nodes are resource constrained and are vulnerable to compromised node attacks by adversaries. A reliable transmission mechanism using a link-layer, hop by hop acknowledgment protocol is established so that various types of packets in the scheme are not lost.

PROPOSED SYSTEM DESCRIPTION

Attack model: Researchers assume that an adversary can compromise a small fraction of sensor nodes and can gain control on these nodes. Once a sensor node is compromised, it can disrupt the normal operation of the underlying routing protocol by message dropping on purpose or denial of message attacks to deprive other nodes from receiving messages of the sink. In this study, researchers address the source authentication by using PKC for broadcast authentication and symmetric secret key authentication for data transfer between the source node and the sink to achieve expected security levels with minimized computational and communication costs.

Design goals: The security goal is straightforward; researchers aim to provide immediate authentication to all messages broadcast by the network users so that the bogus messages inserted by the illegitimate users and any compromised sensor nodes can efficiently be rejected and filtered. In particular, researchers focus on minimizing the overheads of the security design in terms of computation, communication and storage costs.

An overview: In this study, researchers present an overview of the MultiSECPKC scheme, designed to provide high security in a distributed wireless senor network. The sink generates the key pair (PR_{sink}, PU_{sink}); PR_{sink} is the private key and PU_{sink} is the public key of the sink. The Pu_{sink} is stored by all sensor nodes in the network during node deployment. Similar to the sink, each sensor node S holds a pair of keys (PR_S, PU_S); PU_S is the public key and PR_S is the private key. It is assumed that each source node S is surrounded by a set of neighbours denoted by a vector $N_S = [n_i]_{i = 1, ..., n}$ Let the data from the source nodes $[S_i]_{i = 1, ..., m}$ to the sink be denoted by a vector $D = [m_i]_{i = 1, ..., k}$, in total k+1 data blocks. The data D is partitioned and organized as k blocks. Each data block m_i is regarded as a message authenticating unit. The first block m_0 contains an authenticator, additional information like MAC or signature to authenticate a data block. Each of the other k-1 blocks also contains a specified authenticator. This study presents a new authentication scheme based on ECDSA and MD5 hash function. The following steps are executed to perform an authenticated data transfer from a source node to sink.

Step 1: The source node initially broadcasts a route request R_{REQ} to the sink through its next hop neighbours. The R_{REQ} message includes source node's identity ID_S, its public key PU_S and a unique sequence number SN, set to 0 initially. SN indicates the freshness of R_{REQ} and is incremented for every new R_{REQ} initiated:

$$R_{REQ} = ID_S \| PU_S \| SN \quad (1)$$

Step 2: The neighbouring nodes check for the authenticity and freshness of R_{REQ} by looking at the Sequence Number Table (SNT). If R_{REQ} is authentic then the request is forwarded to the next neighbour. Otherwise, the source node is registered as a compromised node. Each neighbouring node in the forward path checks for the authenticity and freshness of R_{REQ} and forwards the request to the sink.

Step 3: Sink verifies for the validity of PU_S and generates a Certificate C_S.

Step 4: Sink signs the Certificate C_S using its private key PR_{sink} and sends C_S to authenticate the source node that has send the R_{REQ}. The C_S is of the form shown as:

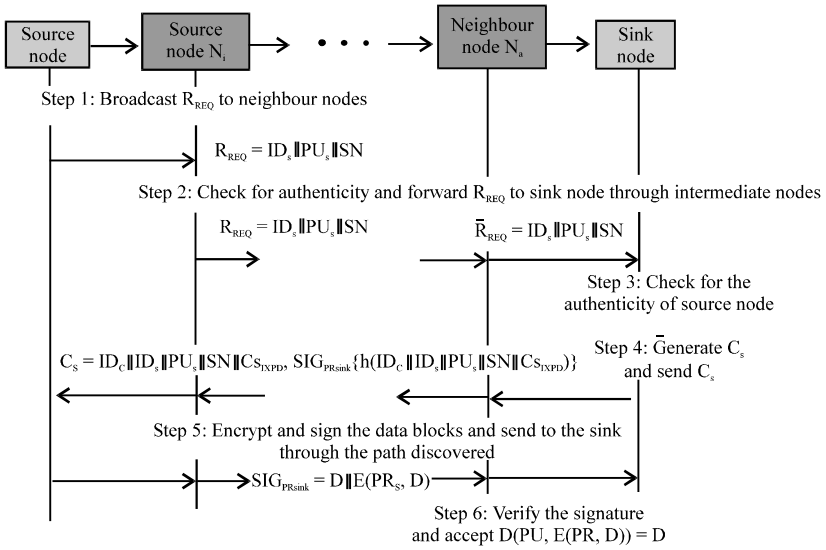

Fig. 1: Step by step procedure describing MultiSECPKC Authentication technique

$$C_S = ID_C \| ID_S \| PU_S \| SN, SIG_{PRsink}, \{h(ID_C \| ID_S \| PU_S \| SN)\}$$

Step 5: The source node computes hash values of all data blocks, obtains the message digest as D = H(CON(D)), where CON(D) = $d_1 \| d_2 \| \ldots \| d_k$. Source NodeS generates a signature for the data to transmit using hash function as:

$$SIG_{PRsink} = D \| E(PR_S, D)$$

The source node sends the signature and the data along with the C_S to the sink of the form:

$$[SIG_{PRsink}, C_S]$$

Step 6: Sink checks for the data authenticity through verification. If D(PU_S, E(PR_S, D)) = D then D is said to be authentic. The sequence of steps involved is shown in Fig. 1.

PROPOSED ALGORITHMS

Researchers propose a novel PKC based authentication scheme for WSNs to meet the following properties:

- Reduced computation and communication overhead almost same as that of HMAC
- It is very difficult for an adversary to compromise the sink to launch a valid authentication
- Use of ECDSA to sign the first block message itself provides the necessary authenticity to the receiver
- More resilience to node compromise attacks

Sensor nodes initialization and deployment: Given the security parameter κ, the sink first chooses an elliptic curve (E(Fp), G, q) defined over Fp, where p is a large prime and G ϵE(Fp) is a base point of prime order q with |q| = κ. Then, the sink selects a secure cryptographic hash function h(), where h: {0, 1}*→ Z*q. Finally, the sink sets the public parameters as params = {E(Fp), G, q, h()}. To initialize sensor nodes S = {S_0, S_1, S_2, ...}, the sink invokes the Algorithm 1. Then, the sink deploys these initialized sensor nodes at a certain region throughvarious ways by air or by land. Researchers assume that all sensor nodes are randomly distributed after deployment.

Algorithm 1 (Network initialization algorithm):

Procedure network initialization:
Input: params = {E (Fp), G, q, h()} and Uninitialized
Network S = {S_i}, i = 1,2,3,4…, n
Output: Network Initialized
  for each sensor node S_iϵS, do
    preload S_i with Tiny ECC, parameters T =
    (p, a, b, G, q, h) and initial energy.
    compute the public key PU_S and private key
    PR_S and install in S_i
end for
return Network Initialization
end procedure

Route discovery and certificate generation: A sensor node that has some data ready to transmit performs the following steps to discover the route from the source node to the sink. The source node broadcasts a route request R_{REQ} to all its 1-hop neighbour nodes as shown in Fig. 2.

The neighbouring nodes check for the authenticity and freshness of the R_{REQ} by looking at the Sequence Number Table (SNT). R_{REQ} includes source node's identity

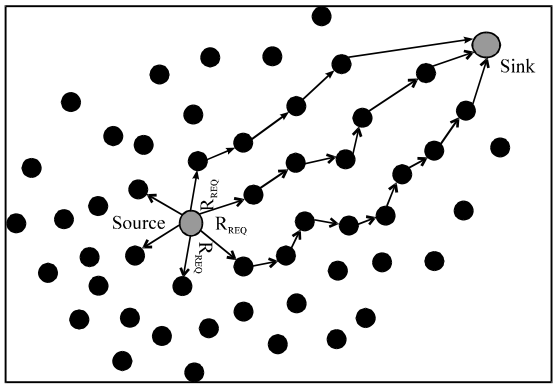

Fig. 2: Route request

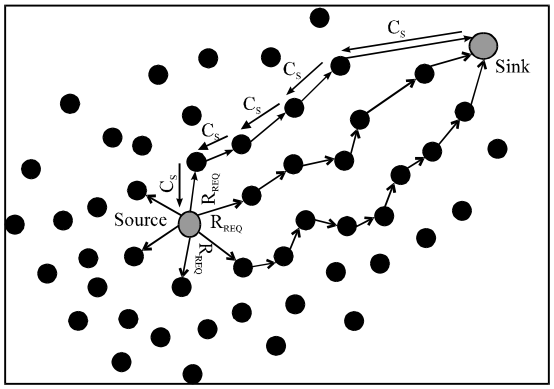

Fig. 3: Route discoveries for certificate generation

ID_S, its public key PU_S and a unique Sequence Number (SN) which is set to 0 initially. SNT stores all the above three entries (i.e., ID_S, PU_S, SN) and broadcasts it's identity B that uniquely identifies a R_{REQ} in the network. If an entry not equal to '0' is present in the SNT for the received R_{REQ}, it is considered as a duplicate and is discarded without further broadcasting. Otherwise, the intermediate node increments the sequence number and updates its routing table for forward path before broadcasting the R_{REQ} message. The neighbour nodes thus create a secured route from source to the sink. Using this route discovery procedure all the available routes between the source and the sink are identified and are stored in the routing table maintained by each node in the network. Before forwarding the R_{REQ} each intermediate node checks for the authenticity of R_{REQ}. Otherwise, it registers the node as a compromised node. The sink generates a certificate C_S for the first fresh R_{REQ} message received and sends this certificate as the response R_{REP} as shown in Fig. 3.

The intermediate nodes are forbidden to send R_{REP} even if they have an active route to sink. The source node transmits the encrypted signed data as soon as it gets

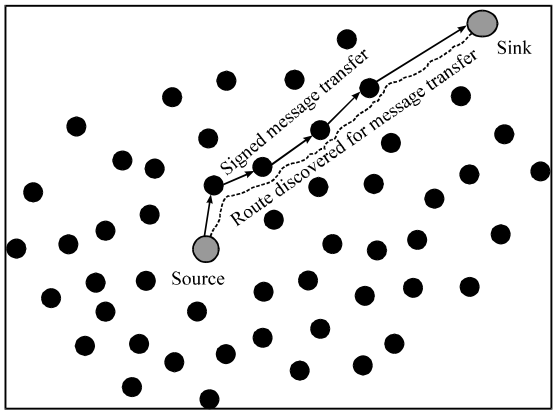

Fig. 4: Signed message transmission

R_{REP} through this main route to the sink as shown in Fig. 4. All the other routes that are discovered will be stored in the routing table as alternate routes. The route selection process is such that whenever a route is required for data transmission, it always selects the main route if it is available. If the main route is not active then the route with lowest hop count from the available alternate routes is selected. Whenever, there is no route available in the routing table, the route discovery process is initiated again and another set of routes are stored. Algorithm 2 shows the sequence of steps involved in route formation and data transmission.

Algorithm 2 (Route formation and data transmission):

1: Procedure route formation and data transmission
Input: $R_{REQ} = ID_S \| PU_S \| SN$
Output: $R_{REP} = C_S = ID_C \| ID_S \| PU_S \| SN \| E(PR_{sink}, ID_C \| ID_S \| PU_S \| SN)$
2: Broadcast R_{REQ} to each neighbour node N_i, i = 1, 2, ..., n
if SN = 0 #check for the freshness in the
3: entry of the seen table (SNT) #
4: then Forward R_{REQ}.
5: Else SN = SN+1
6: Forward R_{REQ}# Increment the entry in the SNT and forward the R_{REQ} #
7: end if
8: If N_i = sink
9: then Generate C_S, Sign C_S and send R_{REP}
10: Else Go to Step 5
11: end if
12: Select the main route
13: Perform data transfer
14: return route formation and data transmission
15: end procedure

Message authentication: In resource constrained environments like wireless sensor networks it is necessary to provide efficient authentication mechanisms to have aneffective control on attacks like eavesdropping, node compromise and DoS, etc. during which an adversary is

able to intercept messages, change the intercepted messages and retransmit them. Sometimes, the adversary may fabricate messages and inject them to the network that may result in heavy traffic causing destruction of the entire network. The proposed MultiSECPKC authentication scheme uses one ECDSA signature to authenticate all the transmitted messages. This in turn reduces the risk in overheads to be controlled by the sink. In MultiSECPKC only one signature is used to authenticate the authenticator in D_0. The authenticator in D_0 is used to authenticate D_1. This process continues until D_k. Authenticators for data blocks are generated using collision resistant hash functions using Algorithm 3.

Algorithm 3 (Generation of signed data):

Procedure; generation of signed data
Input: Data D, a string of random characters
Output: Signed Data $SIG_{PRsink} = \{D \| E(PRs, D)\}$
  Partition D into k blocks $d_1, \ldots, d_k$, $D = [d_i]^T_{i = 1, 2, \ldots, k}$
  Initialize d with a string of random characters
    for i = 1 to k; do
    Concatenate the data blocks d_i to generate
    $CON(D) = d_1 \| d_2 \| \ldots \| d_k$
    Compute Digest D
    D = H(CON(D)) with a collision resistant
    hash
    end for
  Generate $SIG_{PRsink} = D \| E(PR_S, D)$
  return generation of signed data
  endprocedure

Transmitting signed messages: Source node S sends the data D and the certificate C_S to the sink. On receiving the data, the neighbour nodes checks whether D belongs to current source node. If D belongs to S_i, the receiver tries authenticating S_i and D. After authentication of S_i, D will be transmitted to the next node along the route after a short back-off. This process continues until the message reaches the sink.

Verifying extended blocks: According to the algorithm, SM is authenticated by the signature that is if the decryption $D(PU_S, E(PR_S,D)) = D$ then D is said to be authentic.

PERFORMANCE ANALYSIS

In this study, researchers first analyze the security aspects of the proposed MultiSECPKC protocol. Researchers then analyse the overheads in computation, communication and storage of the proposed protocol.

Security: As the proposed scheme is based on PKC, it is inherently more resilient to node compromise attacks. Authentication sensitive information like PU_{sink}, PU_S and the digest will be known to an adversary only if the sink is compromised. However, the adversary can neither create a valid certificate using PU_{sink} nor generate the signature for d_0 with PU_s. H being collision resistant hash an adversary cannot forge d_i using the digest D. Thus, an adversary cannot impersonate a valid source node even after compromising it. In the proposed technique ECDSA is used to authenticate d_0. The remaining k-1 data blocks are authenticated by the digest of its previous data block. Thus, the authenticity of d_i is equivalent to authentication by an ECDSA signature. Researchers see that d_i is the input to compute d_i in d_{i-1} and d_{i-1} is the input to compute d_{i-1} in d_{i-2} and finally, d_1 is the input to compute d_1 in d_0. As the above computation is performed using a collision resistant hash it is computationally infeasible for an adversary to forge D_i without changing d_1. Finally, as authenticity of each data block is equivalent to being authenticated by an ECDSA signature, the proposed scheme is assecure as conventional PKC based authentication scheme.

Overhead: In the sensor network, researchers aim is to find the optimum local connectivity that will provide maximum resilience against compromised node with minimum computational overhead. The average time required to authenticate one data block in Dat the source side is T_{source} as shown in Eq. 1:

$$T_{source} = \frac{T_{sign} + kT_{hash}}{m} \quad (1)$$

The average time to authenticate one broadcast message at the sink side is T_{sink} is given in Eq. 2:

$$T_{sink} = \frac{T_{verify} + kT_{hash}}{m} \quad (2)$$

It is shown in Fig. 5 that MultiSECPKC has a low delay compared to the basic MultiSEC scheme. The computation and communication overhead of the proposed scheme is to the same degree of HMAC.

It can be shown that the computation overhead involved in processing the mathematical expressions for providing message privacy by a sensor processor is greatly reduced. Figure 6 shows the relation between the storage and message count. Figure 7 shows the relation between the storage and message overhead. There is considerable reduction in the storage requirement due to less computation overhead. The average authenticator size per message is L_{av} can also be given as in Eq. 3:

$$L_{av} = \frac{L_{sig} + (k+1)L_d}{m} \quad (3)$$

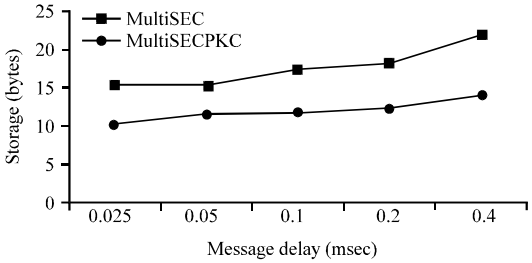

Fig. 5: Relation between storage and message delay

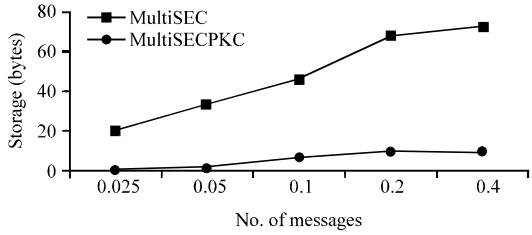

Fig. 6: Relation between storage and No. of messages

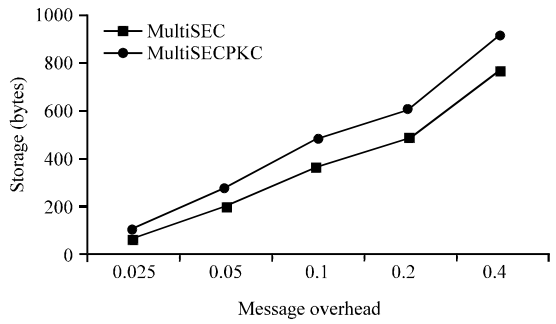

Fig. 7: Relation between storage and message overhead

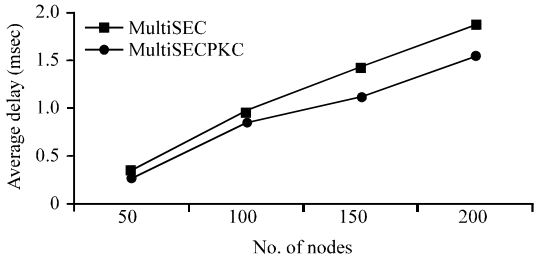

Fig. 8: Delay characteristics

Suppose the arrival time of the first message and last message of data block d_i, $1 \le i \le k$ is denoted by $t_{(i-1)b+1}$ and t_{ib}, respectively. The time delay to authenticate d_i at the receiver side is Ti as shown in Eq. 4:

$$T_i = t_{ib} - t_{(i-1)b+1} \quad (4)$$

Figure 8 shows the delay characteristics of MultiSECPKC compared with its basic scheme. Figure 9

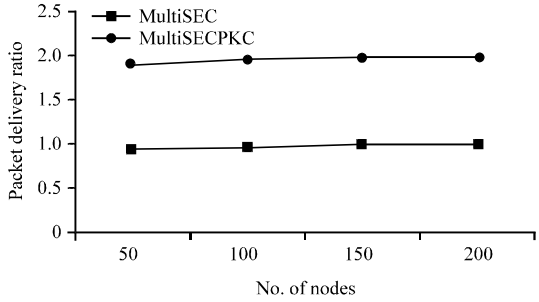

Fig. 9: Packet delivery ratio

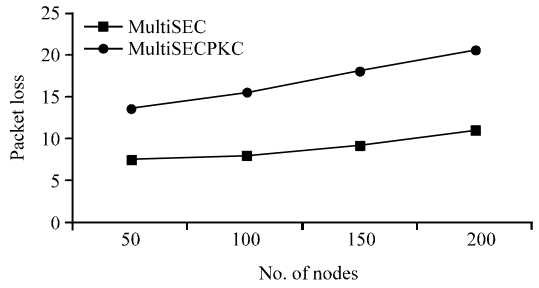

Fig. 10: Packet loss

shows a large improvement in Packet Delivery Ratio (PDR) when compared to its basic scheme. As shown in Fig. 10, the packet loss experienced in MultiSECPKC is also comparably very less to that of the basic protocol.

CONCLUSION

Design of security techniques for WSNs requires more efficient methods to perform mutual authentication in an insecure network environment. In this study, researchers have proposed an efficient authentication scheme based on ECDSA together with compromised node detection to provide a mutual authentication between any pair of sensor nodes in a wireless sensor network. The proposed protocol can also protect inside security andoutside security. Furthermore, it not only inherits the merits of ECC-based mechanism but alsoenhances the WSN authentication with higher security than other protocols.

Therefore, the proposed protocol is more suited to WSNs environments. The protocol has the following features; it is suitable for both static and dynamic WSNs. The system is scalable and resilient against node compromise. After comparing with the basic protocol, the protocol could save about 40% in communication with less memory cost than those pre-distribution schemes without incurring in a considerable amount of communication.

ACKNOWLEDGEMENTS

Researchers would like to thank the anonymous reviewers for their careful reading; insightful comments and suggestions that have helped us improve the presentation of this study.

REFERENCES

Akyildiz, I.F., W. Su, Y. Sankarasubramaniam and E. Cayirci, 2002. Wireless sensor networks: A survey. Comput. Networks, 38: 393-422.

Baek, J., H.C. Tan, J. Zhou and J.W. Wong, 2008. Realizing Stateful Public Key Encryption in Wireless Ensor Network. In: IFIP 20th World Computer Congress, IFIP SEC'08, September 7-10, 2008, Milano, Italy, Jajodia, S., P. Samarati and S. Cimato (Eds.). Springer-Verlag, Germany, pp: 95-108.

Camtepe, S.A. and B. Yener, 2005. Key distribution mechanisms for wireless sensor networks: A survey. Technical Report TR-05-07. Department of Computer Science, Rensselaer Polytechnic Institute, Troy, New York, USA.

Chang, S.M., S. Shieh, W.W. Lin and C.M. Hsieh, 2006. An efficient broadcast authentication scheme in wireless sensor networks. Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security, March 21-24, 2006, Taipei, Taiwan, pp: 311-320.

Das, M.L., A. Saxena and V.P. Gulati, 2004. A dynamic id-based remote user authentication scheme. IEEE Trans. Consumer Elect., 50: 629-631.

Leung, K.C., L.M. Cheng, A.S. Fong and C.K. Chan, 2003. Cryptanalysis of a modified remote user authentication scheme using smart cards. IEEE Trans. Consumer Elect., 49: 1243-1245.

Perrig, A., R. Canetti, D. Song and J.D. Tygar, 2001a. Efficient and secure source authentication for multicast. Proceedings of the Internet Society Network and Distributed System Security Symposium, February 23-26, 2001, San Diego, CA., USA., pp: 35-46.

Perrig, A., R. Szewczyk, J.D. Tygar, V. Wen and D.E. Culler, 2001b. SPINS: Security protocols for sensor netowrks. Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, July 16-21, Rome, Italy, pp: 43-50.

Rahman, S.M.M. and K. El-Khatib, 2010. Private key agreement and secure communication for heterogeneous sensor networks. J. Parallel Distrib. Comput., 70: 858-870.

Ren, K., W. Lou, K. Zeng and P.J. Moran, 2010. On broadcast authentication in wireless sensor networks. IEEE Trans. Wireless Communi., 6: 4136-4144.

Wander, A.S., N. Gura, H. Eberle and V. Gupta, 2005. Energy analysis of public-key cryptography for wireless sensor networks. Proceedings of the 3rd International Conference on Pervasive Computing and Communication, March 8-12, 2005, Kauai Island, Hawaii, pp: 324-328.

Wang, H. and Q. Li, 2006. Efficient Implementation of Public Key Cryptosystems on Mote Sensors. In: Information and Communications Security, Ning, P., S. Qing and N. Li (Eds.). Springer-Verlag, Berlin, Heidelberg, pp: 519-528.

Watro, R., D. Kong, S. Cuti, C. Gardiner, C. Lynn and P. Kruus, 2004. TinyPK: Securing sensor networks with public key technology. Proceedings of the 2nd ACM Workshop Security of Ad Hoc and Sensor Networks, October 25, 2004, Washington DC., USA., pp: 59-64.

Wong, K.H.M., Y. Zheng, C. Jiannong and W. Shengwei, 2006. A dynamic user authentication scheme for wireless sensor networks. Proceedings of the International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing, Volume 1, June 5-7, 2006, Taiwan, pp: 244-251.

Xiong, X., D.S. Wong and X. Deng, 2010. TinyPairing: A fast and lightweight pairing-based cryptographic library for wireless sensor networks. Proceedings of the Wireless Communications and Networking Conference, April 18-21, 2010, Sydney, Australia.

Zhang, W., S. Zhu and G. Cao, 2009. Predistribution and local collaboration-based group rekeying for wireless sensor networks. J. Ad Hoc Networks, 7: 1229-1242.