International Journal of Computer Networks and Applications
Volume 2, Issue 2, March – April (2015)

RESEARCH ARTICLE

A Probabilistic Key Management Protocol based on Kryptograph for WSN

Prachi
Associate Professor, Department of CSE & IT
ITM University, Gurgaon, India

Abstract – Security is a matter of extensive research interest with widespread deployment of WSN (Wireless Sensor Network) in various real life applications. Unreliable wireless communication, physically insecure locations and resource exhaustion attacks render sensor vulnerable to several security breaches. Sensors are supposed to operate on battery in hostile and unattended environment over a longer span of time. Taking into consideration conflict in interest between security and energy consumption, effective security implementation is non-trivial in WSN. A number of security schemes were presented in literature for ad hoc networks. However, most traditional security solutions like public key cryptography and trusted third party schemes are infeasible in WSN due to resource stringent nature. Probabilistic key management scheme (PKMS) perfectly suites the requirement of WSN due to their low storage, computation and communication overhead over resource stringent nodes. However, most the earlier presented PKMSs are based on Erdos-Renyi (ER) model of random graphs. ER model doesn't go along well with WSN due to their certain assumptions. In this paper, we present and implement a new PKMS scheme in TinyOS based on kryptograph model. Simulation results illustrate that scheme based on kryptograph model is more secure, memory efficient and connected when compared to scheme based on ER model.

Index Terms – WSN, security, Kryptograph, ER.

1. INTRODUCTION

WSN depicts a novel monitoring and controlling technology that makes computing natural part of real world. WSN comprises of densely deployed sensors spatially distributed over a geographical region to perform sensing for parameters of interest. Their ability to self-organize enables them to deploy in various hazardous situations without predefined infrastructure. Several commercial sensors nodes are available in market. These commercial sensors (also popular as motes) serve as foundation for WSN. In this paper, we simulate our application on Mica2 mote [1]. Mica2 mote is the third generation, tiny, wireless node used to generate low power WSN. Mica2 uses TinyOS[2] as OS (Operating System).

Communication via publicly accessible wireless communication channel makes WSN prone to numerous security attacks. With unreliable source of communication, adversaries can easily intercept the channel to modify existing information or inject false information. Increased deployment of WSN in various critical applications like nuclear power plant, tracking of enemies position in military, e-healthcare, issuance of warning in case of natural disasters such as earthquake make security matter of essence. Implementing security in WSN is focus of recent researchers because of scarcity of resources (limited battery life, storage and computation power), deployment of sensors in hostile environment and sensitivity of information to several type of attack. Since security is an auxiliary operation for WSN, availability of resilient but lightweight security solution is extremely essential.

Taking into consideration this unique requirement of WSN, we propose a new random key pre-distribution scheme based on kryptograph model [3]. All the earlier proposed key management schemes were based on ER model [4]. Also, compare our scheme based on kryptograph model with scheme based on ER model. Simulation and analytical result proves that our scheme offers higher probability of connectivity, resilience and at the same time reduces storage burden of resource stringent nodes.

Remainder of this paper is organized as follows. Related work in this area is discussed in section 2. Section 3 gives detailed description of our scheme. Section 4 presents introduction to simulator TinyOS. Section 5 provides detailed explanation about simulation of our scheme on TinyOS. Mathematical analysis of ER and kryptograph model is presented in Section 6. Performance evaluation and comparison of our scheme based on kryptograph and ER model is performed in Section 7. Section 8 draws conclusion

2. RELATED WORK

Cryptography acts as foundation for any security scheme implementation. Security credentials are linchpin for achieving security. According to security credentials, cryptographic mechanisms can be categorized as symmetric (private) and asymmetric (public) cryptosystem. Asymmetric cryptosystem [5], [6] offers high level of security but major portion of energy is consumed in key generation and computation, followed by communication. Moreover, they increase storage burden due to large size keys. Limited computation, storage and power sources of sensor nodes limit public key cryptosystem deployment in WSN. In order to authenticate nodes, Kerberos, a network authentication

RESEARCH ARTICLE

protocol [7] based on trusted third party was presented by Steiner et al. However, this scheme is extremely expensive in terms of communication and leads to energy depletion. Many other protocols [8-10] were also proposed based on the concept of third party server but none of them is suitable to work with WSN because of limited scalability, huge amount of communication and single point of failure.

To cater the problems associated with trusted third party schemes, pairwise key establishment schemes were introduced. Full pairwise key establishment is one of the most basic solutions to ensure secure communication among two parties. This scheme provides high security, connectivity and node to node authentication but at the same time limits scalability, complicates addition of new nodes in network and magnifies storage burden of each node especially in large size networks. To avoid storage burden of full pairwise scheme, probabilistic approaches were introduced where every node is connected to every other node with certain probability. Eschenauer et al. pioneered first PKMS, also known as EG scheme [11]. EG scheme is divided in three phases. During key distribution phase, key pool of random keys is generated using pseudo random number generator and nodes are loaded with random subset of keys along-with their identifier from pool prior to their deployment in field. In shared key discovery phase, nodes broadcast their list of identifiers to find shared keys with other nodes. Nodes that share keys together establish a secure link. Nodes that don't share a key find node with which both of them are directly linked during path key establishment phase. Many schemes presented in literature are based on EG scheme because probabilistic approaches induce small memory overhead, negligible computation and provide good amount of connectivity with limited keys. Chan et al. proposed three different schemes [12] to secure communication among two parties. Q-composite scheme was proposed as an improvisation of EG scheme where nodes need to share q (q>=1) keys in order to communicate. This scheme improves resilience since eavesdropper needs to capture q keys instead of just one. Though to ensure sharing of q keys among two nodes we need to minimize pool size and with a small size pool impersonating small number of nodes reveal large portion of communication in network. Another variation of EG scheme is multipath reinforcement scheme where basic idea is to update the link keys after shared key discovery phase through multiple disjoint paths. This scheme enhances resilience but at cost of increased communication via multiple paths and computation overhead during generation of random values. As a variation to pairwise scheme, random pairwise scheme was introduced where nodes share pairwise keys with certain probability and p must be chosen carefully to ensure connectivity of network. Random pairwise scheme works well for small size network but not in large networks. EG and q-composite scheme suffer from the fact that small number of compromised nodes effect communication among uncompromised nodes because of random election of keys from pool. So, same keys are chosen by number of nodes. To resolve this problem, threshold based key pre-distribution scheme ([13], [14]) were introduced. When number of compromised nodes remains below threshold, links between uncompromised nodes remain unaffected. When number increases above threshold then whole network is revealed. Blom proposed first threshold based key pre-distribution scheme [13] based on the concept of matrices. Then Blundo et al. proposed polynomial based key pre-distribution scheme [14] where setup server generates a t-degree polynomial. Both the above mentioned schemes increase resilience but on stake of huge storage and computation overhead.

Du et al. proposed a key distribution scheme based on BLOM method and EG scheme [15]. This scheme guarantees that any two nodes can compute pairwise key among them. It makes use of matrices and modular multiplication. Liu et al. combine idea of polynomial based key pre-distribution and EG scheme to introduce polynomial-pool based key pre-distribution scheme [16-17]. However, these schemes also suffer from storage and computation overhead. All the above presented schemes assume unpredictability of network topology and load nodes with huge number of keys to achieve desirable connectivity. From 2003-2006 several key distribution protocols were proposed based on deployment knowledge of sensors. Du et al. exploit location of nodes prior to their deployment in field and avoids unnecessary key assignments [18-19]. This scheme uses non-uniform probability density function i.e. sensor are likely to be deployed in certain areas. Major drawback of this scheme is that it considers deployment point as the point where we want to deploy sensor not the actual location where sensor resides. Difference in predicted and actual value determines feasibility of this scheme. A number of authors presented deployment knowledge based security scheme in [20-22]. All these schemes introduce huge storage overhead.

A pairwise key bootstrapping technique was proposed for large-scale WSN (iPAK) in [23] by Ma et al. claim that their scheme provides high connectivity, strong resilience at low storage overhead though computation overhead is similar to blom based scheme [15].

Hussain et al. proposed a key-hash-chain based scheme to use different keys in each session [24]. It combine EG with multi-reinforcement scheme. Here, each node is preloaded with a set of generation keys. Generation keys are used along-with publicy known seed and hash function to generate distinct key chains. To avoid security threat due to sharing of generation key, generation keys are updated with multipath key reinforcement scheme. This scheme slightly increase resilience but induce more communication and computation overhead. Li et al. proposed a continuous secure scheme [25]

RESEARCH ARTICLE

based on two dimensional backward hash key chain. This chain is divided into n disjoint sub-key pool. This scheme offers high connectivity and resilience but with huge computation burden.

It is clear from the aforementioned discussion that none of the previously presented schemes is suitable for unique requirements of WSN. So, in this paper we try to implement a security scheme that offers higher probability of connectivity and resilience at low communication, computation and storage burden.

3. DESCRIPTION OF SCHEME

Our scheme is divided in three phases: key pool generation phase, direct key establishment and path key establishment. Key pool generation phase is executed prior to deployment of nodes in network field. In this phase, a key pool of random numbers is generated with the help of pseudo random number generator and each node is assigned a fixed size subset of keys (also known as key ring) randomly selected from the pool along-with their identifiers. During direct key establishment phase, nodes that want to communicate with other nodes of network determine whether these two nodes are in communication range. If both the nodes are in communication range then sending node prepares a message that comprises of type of message, sender and destination address and identifiers of all keys belongs to sending node and forward the message to destination. Whenever destination node receives a message it checks type of message to determine its purpose. If purpose is to establish a secure communication channel then destination determines whether it shares any id with sender or not. If a shared id is found, destination sends id of shared key to sender. Now, sender determines key corresponding to shared id and uses this key to secure communication between sender and receiver. If no shared key is found, destination prepares a message that contains key id list of destination and send it to sender. Upon receiving this message, sender initiates path key establishment phase. Sender adds its own key id list in the message sent by receiver and broadcasts it in the network. After receiving message, every node checks sender and receiver address. If node address doesn't matches with sender or receiver then this node determines whether it shares any key with source and destination. If intermediate node doesn't share keys with both the nodes then it broadcasts the message further. If intermediate node shares a key then it selects a new key randomly from the pool and encrypts this key with the key shared with source and destination. For encryption, TinyOS supports Skipjack and RC5 encryption and authentication algorithms. Skipjack needs 48 microseconds and RC5 needs 33 microseconds for encrypting a byte. Here, we encrypt our keys using RC5 algorithm because RC5 algorithm is less time consuming as compared to Skipjack. Now, intermediate node embeds type of message, sender address, destination address, encrypted keys and shared key ids of node with source and destination in packet. Intermediate node forwards this message to sender. Sender determines shared key corresponding to key id sent by intermediate node and decrypts the newly generated key with shared key and forward the message to destination. Now, destination determines shared key with the help of id and decrypts the newly generated key with shared key. Now, this newly generated key will acts as shared key between sender and receiver to secure communication. Note that key k_i is chosen from key pool in order to improve resilience so that even if key ring of intermediate node is compromised, keys generated by intermediate node during path key establishment remain intact. Overhead generated during path key establishment is minimized in a way that indirect path is used only once during path key setup

4. SIMULATION OF OUR SCHEME

4.1. Simulator

We have implemented our scheme on TinyOS. TinyOS is a small, open source, event based and energy-efficient simulator. System, library and applications are written in nesC (nestedC) [26]. TinyOS was developed by University of California, Berkeley in co-operation with Intel Research and Crossbow Technology to develop small, low-cost and low power sensors also popular as motes. Its component based architecture allows rapid application development while minimizing code size in ROM and data size in RAM by avoiding redundant declaration and initialization of variables in order to minimizing code and data size.

4.2. Implementation

To design our security scheme, we use basic set of services offered by TinyOS and add additional components according to requirement of our application. In TinyOS, component can be categorized in two ways: module and configuration. Each module implements one or more interfaces. We reuse some of the earlier defined modules in our scheme because it reduces application development time. In addition to basic set of services, we implement new interfaces (services that a module provides or uses) and bind them with some of the predefined interfaces (Main, TimerC, RandomLFSR, GeneriComm and SecPrimitive). Just like C, Main is first component where execution of any application initiates. Main component passes the control to all other components connected to application. Main is also responsible for starting and initializing FIFO (First In First Out) scheduler. StdControl is the interface that is used to initiate, start and stop various TinyOS components. During booting, Main calls init() and start() function of StdControl interface. Further, start() function of StdControl interface calls start() function of Timer interface and GenerateKey() function. TimerC component is used by various nodes to periodically send and receive packets.

RESEARCH ARTICLE

GenerateKey() function uses RandomLFSR (Linear Feedback Shift Register) component that generates 16-bit pseudo-random numbers (keys) for our application. It generates pool of random 16-bit keys using rand() function of Random interface. GenerateKey() function further calls AssignKey() function to assign keys randomly to nodes. To enable continuous transmission of message, timer is fired using fired() event of Timer interface at regular interval. Fired() event of Timer interface calls EstablishKey() function. EstablishKey() function adds type of message, sender address, destination address and sender's key ids in data part of TOS_Msg packet format (a standard packet format that sends and receives message between two nodes). TOS_Msg acts as a message buffer which contains active message packet and packet metadata. Its default length is 29 bytes. In our application, GenericComm is a component is used to provide SendMsg and ReceiveMsg interfaces that enables us to transmit and receive active messages. Both interfaces are parameterized by active message id. EstablishKey() function calls send() function of SendMsg interface by passing TOS_Msg as argument. Whenever a node receives message, event receive() of ReceiveMsg interface is executed. Receive() event calls ProcessReceivedMessage() that retrieves type of message, sender and receiver address. Check() function is called from ProcessReceivedMsg() that checks whether receiving node shares any key with sender or not. If yes, check() function returns id of matched key and this key is used for secure communication between two parties else destination adds its key in the message and rebroadcasts it to other nodes in transmission range using PathKeyEstablishment() function. In PathKeyEstablishment() function, an intermediate node is found such that intermediate node shares key with sender as well as receiver. If such an intermediate node is found then a new key is selected randomly from pool using rand() function of Random interface. To ensure secure and authentic transmission of newly generated key, we choose RC5 algorithm because RC5 algorithm is less time consuming. RC5 provides BlockCipher and BlockCipherInfo interface. BlockCipher interface provides encrypt() and decrypt() function for encryption and decryption of message. Encrypt() function is called by PathKeyEstablishment() function. This function encrypts the newly selected key using the keys that intermediate node shares with source and destination. Now, intermediate node forwards TOS_Msg to source node that incorporates encrypted keys alongwith sender address, destination address, message type and id of keys shared with source and destination. After receiving message, source node retrieves the shared key id, decrypts the newly generated encrypted key using decrypt() function and forwards the TOS_Msg to destination. Destination node follows the same procedure to retrieve and decrypt the newly generated key. We combine all components together into a component known as configuration.

To provide security and authentication, TinySec provides implementation for RC5 and Skipjack modules. One of the major constraints on implementation of security scheme on TinyOS motes is their restricted small packet size of 29 bytes. For our scheme, we have used TOSSIM [27] emulator to test properties of TinyOS applications before loading applications in motes.

5. MATHEMATICAL ANALYSIS OF ER AND KRYPTOGRAPH MODEL

Here, we present analysis of random graph theory proposed by ER and kryptograph model.

5.1. ER random graph model

A random graph Gn,e,p is a graph with n vertices, e edges and probability p (probability that an edge exists between any two vertices). Initially, graph contains vertices but no edges i.e. p is zero. Graph is fully connected when p is one. Edges are chosen and added randomly in network with probability p from n(n-1)/2 possible edges where each edge has equal possibility of being chosen. Initially, if edge e1 is elected for insertion then next time an edge is chosen from the possible set by excluding the already chosen edge e1. Graph generated with above mentioned independent and random edge generation process is one instance from family of random graphs. Random graphs can be considered as a model of communication for networks where vertices of graph are mapped to nodes of network and edges are mapped to communication links.

EG scheme takes into consideration random graph model and connectivity results from ER model to design a WSN.

According to ER model, for a desired probability of connectivity (Pc), p can be calculated as

$$p = \frac{\ln(n)}{n} + \frac{c}{n}$$

where c is any real constant.

Number of actual neighbors (n') became limited (n'<<n) due to wireless constraints so probability of sharing atleast one key with neighbor node increases from p to p' (p'>>p). For a given value of p', size of key pool (P) and key ring (k) can be determined as:

$$p' = 1 - \frac{((P-k)!)^2}{(P-2k)!P!}$$

Above equation clearly depicts that value of k is a function of P and p'.

In ER model, probability to compromise secret information if a sensor gets captured is not dependent on n but on k and P:

RESEARCH ARTICLE

$$\Pr[\text{link compromised}] = 1 - \frac{\binom{P-k}{k}}{\binom{P}{k}}$$

Authors in [3] demonstrate that model used by EG is not suitable for secure WSN because of its certain unrealistic assumptions:
ER model doesn't takes into consideration location of sensors so it cannot properly depicts physical proximity of sensors (In other words, existence and insertion of edges in ER model is independent of previously inserted edges).
Another difficulty with ER model is that it assumes full visibility. Every node assumes that it can communicate with rest of the nodes of network irrespective of location of sensors. So, it ensures full connectivity even when nodes are not within communication range.
Both these assumption leads to failure of implementation of ER model in secure WSN

5.2. Kryptograph model

A geometric random model to design connected and secure WSN with small number of keys per sensor was proposed in [3]. This model takes into consideration geographical location of nodes so it is more realistic. They use a graph (known as kryptograph). Kryptograph model generates k (key ring) for each node using P (key pool) with replacement k times and distributes nodes randomly over a geographic area. Secure link exists between nodes if two nodes are within distance r and share a common key.

Relation between connectivity and security

Here, we perform mathematical analysis of secure WSN. If P is fixed, increase in value of k enhances connectivity but declines security of network. With large k, capturing few nodes reveal entire key pool so security of entire network compromised. If k is very small then a single key is shared by only few nodes. It enhances security but network tends to be disconnected. So, k and P should be chosen carefully to guarantee both security and connectivity.
If P>=n and

$$\frac{k^2}{P} \sim \frac{\log n}{n}$$

If both the above mentioned condition satisfies then network is connected with high probability. So, probability of connectivity in network is roughly equals to

$$\Pr[sharing\_key] \sim \frac{k^2}{P}$$

If k^2/P is small, network tends to be disconnected. Higher value of k^2/P results in connected network.
Stochastic dependencies of model give rise to unexpected correlations (when a node is captured not only link incident on that key compromise but also some links between uncompromised nodes that are sharing same compromised keys). ER model completely overlooks this factor. Kryptograph tries to find a solution such that damage is only confined upto links incident on compromised nodes. Aim of kryptograph is to ensure that in order to compromise certain number of links we must capture some minimum number of nodes.
To make a network resilient, k and P should be chosen in such a way that

$$\frac{k}{P} \sim \frac{1}{n}$$

A very small value of k/P results in highly secure network that is connected with low probability. A higher value of k/P makes network connected but not secure. So, value of k and P should be chosen in such a way that both connectivity and security can be assured.
In order to satisfy above equations value of k and P can be chosen as

$$k \sim \log n$$

$$P \sim n \log n$$

6. PERFORMANCE EVALUATION OF OUR SCHEME

This section presents performance of our scheme based on kryptograph model in comparison with performance of scheme on ER model. Results illustrates that scheme based on kryptograph model significantly outperforms scheme based on ER model in terms of connectivity, resilience and storage overhead.

6.1. Simulation environment

We randomly deployed nodes over $100*100m^2$. Network is designed using random waypoint model. Radio is used at physical layer of node. Antenna model is omni-directional. Application traffic between sender and receiver is of Constant Bit Rate (CBR) type. Transmission range is restricted to 200m. For generation of key pseudo random number generator is used.

RESEARCH ARTICLE

Table 1: Simulation parameters

Parameter	Value
Area	500*500m^2.
Deployment model	Random way point model
Packet Size	29 Bytes
Mote	Mica2
Traffic Type	Constant Bit rate
Source/Destination address	32 bit
Communication model	Radio
Transmission range	200m
Antenna model	Omni directional
Key size	16 bit

6.2. Simulation parameters

Various scenarios have been executed using variable number of nodes to determine probability of connectivity and probability of compromised nodes versus key ring size. Also, we evaluate how memory burden of a node increases when desired probability of connectivity increases. Results of several simulations are combined together.

- Probability of connectivity vs key ring size

Figure 1 and 2 represent probability of connectivity of scheme based on kryptograph and ER model for various key ring and key pool sizes.

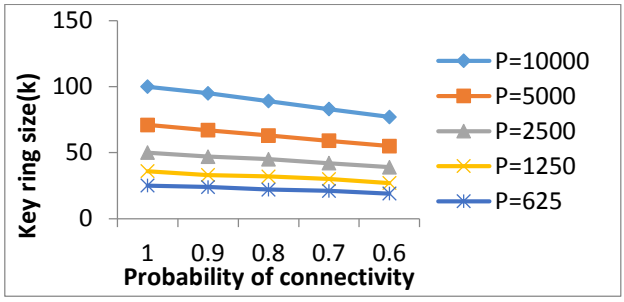

Figure 1 Probability of connectivity in ER model

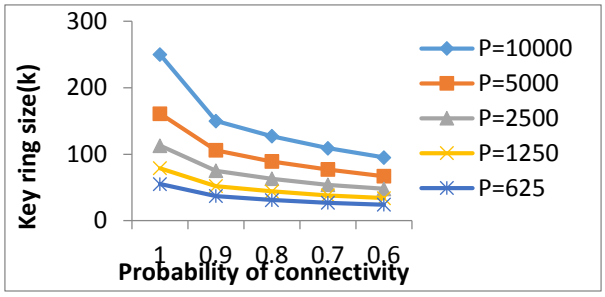

Figure 2 Probability of connectivity in kryptograph model

Figures demonstrate that to achieve same amount of connectivity with same pool size, key ring size required by ER model is more than double of key ring size required by kryptograph model. For example, with pool size of 625, kryptograph model requires key ring size of 25 to achieve full connectivity where as ER model requires key ring size of 55 keys. Similarly, when pool size is doubled i.e. 1250 keys, key ring size in ER and kryptograph model is 79 and 36 respectively.

- Memory overhead vs probability of connectivity

Figure 3 depicts size of key ring in terms of bytes for different probabilities of connectivity based on kryptograph and ER model when pool size is 625 and network size is 110.

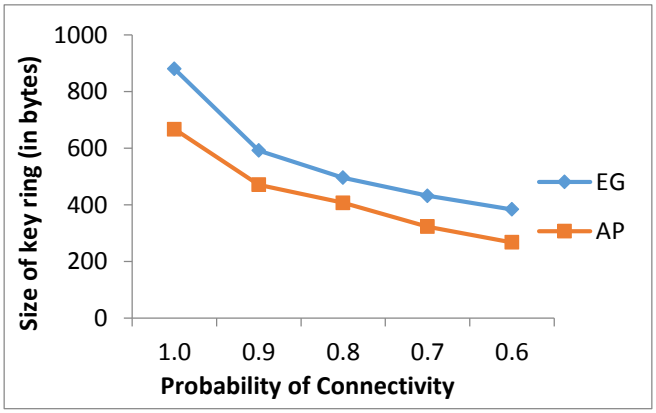

Figure 3 Memory overhead vs probability of connectivity

It is clear from the figure that sensor needs to store small number of bytes if kryptograph model is used. This benefit is especially crucial in WSN because with very limited storage capacity, security is an auxiliary operation for sensor nodes.

- Probability of compromised nodes vs key ring size

Figure 4 and 5 represent probability of compromised nodes under different size key rings.

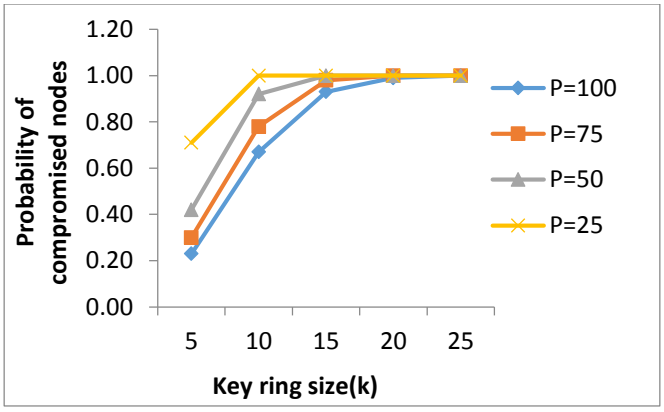

Figure 4 Probability of compromised nodes in ER model

RESEARCH ARTICLE

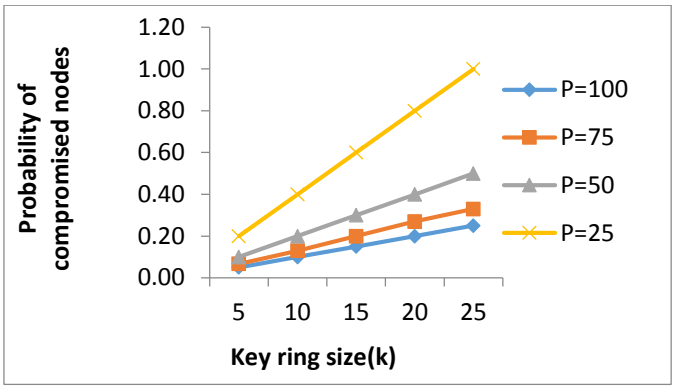

Figure 5 Probability of compromised nodes in kryptograph model

It is clearly depicted that for same size key pool and ring, probability of impersonation in ER is very high when compared to kryptograph model. For example, when pool size is 25 and key ring size is 5 then probability of compromised links is 0.23 in ER model and 0.05 in kryptograph model.

As we already discussed in mathematical analysis of ER and kryptograph model that kryptograph requires small size key ring to achieve same amount of connectivity and above simulation results for connectivity clearly supported this fact. Moreover, figure 4 and 5 depicts that resilience is also high for small size key rings. This is because small size key ring results in less number of shared keys among nodes so disclosure of an entity doesn't allow eavesdropper to impersonate many keys with non-compromised nodes. Small size key rings in our scheme based on kryptograph model not only support similar connectivity but also offer higher resilience and lesser storage burden.

7. CONCLUSION

None of the traditional security scheme such as asymmetric and trusted third party scheme suits the unique requirement of WSN due to their resource intensive operations. In order to survive for long periods WSN requires a lightweight but secure cryptographic mechanism. In this paper we presented a new random key pre-distribution scheme based on kryptograph model on TinyOS simulator. We presented random graph theory based on ER and kryptograph model and evaluate our scheme on both models. Simulation results demonstrate that scheme based on kryptograph is more resilient, offer less memory overhead and highly connected when compared to scheme based on ER model. As a result, our scheme based on kryptograph perfectly claims it suitability for WSN and proves much better when compared to scheme based on ER model.

REFERENCES

[1] MICA2 Wireless Measurement System, Crossbow Technology, http://www.xbow.com, 2011.
[2] www.tinyos.net/
[3] R. D. Pietro, L. V. Mancini, A. Mei, And A. Panconesi, J. Radhakrishnan, Redoubtable Sensor Networks, ACM Transactions on Information and Systems Security, Vol. 11, No. 3, Article 13, Pub. date: March 2008
[4] P. Erdos and A. Renyi, "On the Evolution of Random Graphs" Publ. Math. Inst. Hungar. Acad. Sci., vol. 5, pp. 17-61, 1960.
[5] W. Diffie, M. E. Hellman, "New directions in cryptography", IEEE Transactions on Information Theory, vol. 22, pp. 644–654, November 1976.
[6] R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", Communications of the ACM, vol. 21, no. 2, pp. 120–126, 1978.
[7] John Kohl and B. Clifford Neuman. The Kerberos Network Authentication Service (Version 5). Internet Request for Comments RFC-1510. September 1993
[8] Dierks, T., Allen, C. 1999.: The tls protocol version 1.0.
[9] Adrian Perrig, Robert Szewczyk, J.D. Tygar et al., "SPINS: Security Protocols for Sensor Networks", Wireless Networks 8, 521.534, 2002-2002 Kluwer Academic Publishers Netherlands.
[10] Bocheng Lai, Sungha Kim,Ingrid Verbauwhede. "Scalable Session Key Construction Protocols for Wireless Sensor Networks." Proceedings of the IEEE Workshop on Large Scale RealTime and Embedded Systems (LARTES '02). usa, 2002.
[11] Eschenauer, L. and Gligor, V. D. 2002. A key-management scheme for distributed sensor networks. In Proceedings of the 9th ACM Conference on Computer and Communications Security (Washington, DC, USA, November 18 - 22, 2002). V. Atluri, Ed. CCS '02. ACM, New York, NY, 41-47. DOI= http://doi.acm.org/10.1145/586110.586117.
[12] Chan, H., Perrig, A., and Song, D. 2003. Random Key Predistribution Schemes for Sensor Networks. In Proceedings of the 2003 IEEE Symposium on Security and Privacy (May 11 - 14, 2003). SP. IEEE Computer Society, Washington, DC, 197-213.
[13] R. Blom. An optimal class of symmetric key generation systems. Advances in Cryptology: Proceedings of EUROCRYPT 84 (Thomas Beth, Norbert Cot, and Ingemar Ingemarsson, eds.), Lecture Notes in Computer Science, Springer-Verlag, 209:335–338, 1985.
[14] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung. Perfectly-secure key distribution for dynamic conferences. In Advances in Cryptology – CRYPTO '92, LNCS 740, pages 471–486, 1993.
[15] W. Du, J. Deng, Y.S. Han, and P.K. Varshney, "A Pairwise Key Predistribution Scheme for Wireless Sensor Networks," Proc. 10th ACM Conf. Computer and Comm. Security (CCS '03), pp. 42-51, 2003.
[16] D. Liu and P. Ning, "Establishing Pairwise Keys in Distributed Sensor Networks," Proc. 10th ACM Conf. Computer and Comm. Security (CCS '03), pp. 52-61, 2003.
[17] D. Liu, P. Ning, and R. Li, "Establishing Pairwise Keys in Distributed Sensor Networks," ACM Trans. Information Systems Security, vol. 8, no. 1, pp. 41-77, 2005.
[18] Wenliang Du, Jing Deng, Yunghsiang S. Han, Shigang Chen, and Pramod K. Varshney, A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge, 2004. Twenty-third Annual Joint Conference of the INFOCOM 2004. IEEE Computer and Communications Societies. Volume: 1, 7-11 March 2004, DOI: 10.1109/INFCOM.2004.1354530
[19] Wenliang Du, Jing Deng, Yunghsiang S. Han, and Pramod K. Varshney, A Key Predistribution Scheme for Sensor Networks Using Deployment Knowledge. IEEE Transactions On Dependable And Secure Computing, Vol. 3, No. 1, January-March 2006
[20] Cungang Yang, Jie Xiao, Location-Based Pairwise Key Establishment and Data Authentication for Wireless Sensor Networks, Proceedings of the 2006 IEEE Workshop on Information Assurance United States Military Academy, West Point, NY
[21] Fang Liu, and Xiuzhen Cheng LKE: A Self-Configuring Scheme for Location-Aware Key Establishment in Wireless Sensor Networks, IEEE Transactions On Wireless Communications, Vol. 7, No. 1, January 2008

RESEARCH ARTICLE

[22] Zhen Yu, and Yong Guan, A Key Management Scheme Using Deployment Knowledge for Wireless Sensor Networks, IEEE Transactions On Parallel And Distributed Systems, Vol. 19, No. 10, October 2008
[23] Liran Ma, Xiuzhen Cheng, Member, Fang Liu, Fengguang An, and Jose Rivera, iPAK: An In Situ Pairwise Key Bootstrapping Scheme for Wireless Sensor Networks, IEEE Transactions On Parallel And Distributed Systems, Vol. 18, No. 8, August 2007
[24] Sajid Hussain and Md Shafayat Rahman, Laurence T. Yang, Key Predistribution Scheme using Keyed-Hash Chain and Multipath Key Reinforcement for Wireless Sensor Networks, IEEE Conference on Pervasive Computing and Communication, Galveston, TX, 9-13 March 2009, DOI: 10.1109/PERCOM.2009.4912893.
[25] Sujun Li, Boqing Zhou, Jingguo Dai, and Xingming Sun, A Secure Scheme of Continuity Based on Two-Dimensional Backward Hash Key Chains for Sensor Networks, IEEE Wireless Communications Letters, Vol. 1, No. 5, October 2012
[26] D. Gay, P. Levis, R. V. Behren, M. Welsh, E. Brewer, and D. Culler, "The nesC Language: A Holistic Approach to Networked Embedded Systems", In Proceedings of Programming Language Design and Implementation (PLDI) 2003, June 2003.
[27] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and scalable simulation of entire TinyOS applications", in Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys) 2003, Nov. 2003..

Author

Ms. Prachi has submitted her Ph.D. Thesis in Computer Science at the Banasthali University of Rajasthan, India in 2014. Her current research interests include key agreement in wireless peer-to-peer systems and security in underwater sensor networks. Prachi received the B.Tech. degree from M.D. University, Rohtak in 2007 and the M.Tech. degree in Computer Science from the Banasthali University at Rajasthan in 2009. She is an author of 13 refereed articles in these areas, 6 in reputed international journal and 7 in International Conferences.