A Novel Approach for Secure Keying Protocol for Sensor Networks

B.Srinivas, P.Satheesh, G.Gowthami

***Abstract*— A sensor network is composed of large number of sensor nodes. Each sensor needs to securely communicate with the adjacent sensors in the network. To provide security between every pair of adjacent sensor each sensor in a network needs to store 'n -1'symmetric keys But The storage requirement of this keying protocol is very complex. In this paper, we propose a new keying protocol which reduces the number of keys to be stored in each sensor. we present a fully secure keying protocol here each sensor needs to store (n+1)/2 keys, which is less than the n-1 keys and symmetric keys are used to encrypt and decrypt the exchanged data for secure communication. At least (n-1)/2 keys are required to store in each sensor to secure an entire keying protocol. This new secure keying protocol is very secure against impersonation, eavesdropping and collision attacks.**

***Index Terms*— keying protocol, secure communication, sensor network, symmetric keys**

I. INTRODUCTION

Security is an important issues in any sensor networks In sensor networks, one of the most requirement is to establish cryptographic keys. Earlier, Researchers have proposed a variety of protocols but they are vulnerable to attacks. Comparing the sensor networks from olden days to now a days there is the change in the communication pattern, here sensor nodes required to set up keys with the adjacent nodes. The simple solution for the key establishment is a network wide shared key. Unfortunately, a single node in a network reveals the secret key and allows for decryption. One variant on this idea is to use a single shared key t o establish a set of link keys. However, this variant of the key-establishment process does not allow addition of new nodes after initial deployment.

By using Public-key cryptography each node can set up a secure key with any other node in the network[1],[2] Later, another approach has been implemented with a shared unique symmetric key between each pair of nodes.Each sensor needs to securely communicate with adjacent sensors. we have been using symmetric keys for the communication. Sender must authenticate the receiver and encrypt the messages it sends. To achieve this authentication and encryption is to ensure that the sender and the receiver share a common secret that no other user in the network knows. x and y are two adjacent sensors in a network then these two sensors can use their shared symmetric key Kx,y to authenticate one another by encrypting and decrypting the exchanged data.

Each sensor x in such a network is required to store n - 1 symmetric keys, where n is the total number of sensors in the network. When n is large and the available storage to store keys in every sensor is modest.

The secure keying protocol for sensor networks, that needs to store only (n + 1)/2 shared symmetric keys between two sensors. The number of shared symmetric keys stored in a sensor network with n sensors is n(n + 1)/2, which is close to the optimal number of shared symmetric keys for any key distribution scheme that is not vulnerable to collusion.

It may be noted that in addition to the low number of keys stored, and the ability to resist collusion between sensors, our keying protocol has two further advantages.

Firstly, it is uniform: we store the same number of keys in each sensor. Secondly, it is computationally cheap, and thus suitable for a low-power computer such as a sensor: when two sensors are adjacent to each other, the computation of a shared symmetric key requires only hashing, which is a cheap computation and can be done fast. As our protocol has many desirable properties, such as efficiency, uniformity and security, we call this protocol the secure keying protocol for sensor networks [3][4]and[5].

II. SECURITY ISSUES

A. Authentication and confidentiality:

Authenticity is essential in any communication that the communication is genuine not impersonators. The participants must ensure their identities by using some authentication Mechanisms. without the use of an authentication mechanism the adversary could impersonate and thus gain access to confidential resources.

Network transmission of sensitive information, must requires confidentiality. Confidentiality is that certain information is accessible only to those who have been authorized.

B. Impersonation Attack:

Here the attacker makes independent connections with the victims, making them believe that they are directly talking to each other over a private connection but the entire conversation is controlled by the attacker. When two participants exchanging the messages the attacker may grab the original message and sends the duplicate message.

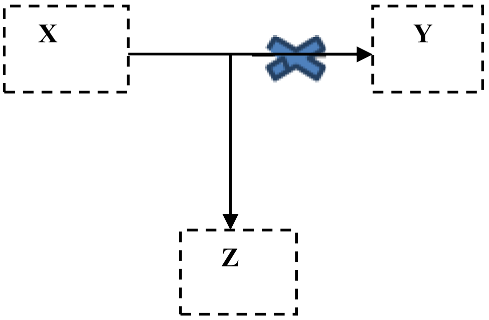


Figure 1 Impersonation Attack

Manuscript received on November, 2012.

B.Srinivas, Computer Science and Engineering, M.V.G.R. College of Engineering, Vizianagaram, India.
P.Satheesh, Computer Science and Engineering, M.V.G.R. College of Engineering, Vizianagaram, India.
G.Gowthami, Computer Science and Engineering, M.V.G.R. College of Engineering,Vizianagaram,India.

C. Eavesdropping Attack:

In sensor network communications, an adversary can gain access to private information by monitoring transmissions between nodes. Example when two adjacent sensors x and y exchange data messages, sensor z notices that it is also adjacent to both x and y and can copy each exchanged data messages between these two sensors.[6],[7].

III. EXISTING SYSTEM

To reduce the number of keys stored in each sensor they were two main keying protocols were proposed in the past. They are probabilistic keying protocol and the grid keying protocol.

Probabilistic keying protocol can defend against eavesdropping but cannot defend against impersonation attack. The grid keying protocol can defend against eavesdropping and impersonation attack but it is vulnerable to collision attack. The In the probabilistic keying protocol[8] , when two adjacent sensors exchange messages they identify which keys are in common and use a combination of the common keys as a symmetric key. By using this symmetric key encryption and decryption of data can provide secure communication. By this it can defend eavesdropping but this problematic keying protocol suffers from another problem that the stored keys in any sensor are independent of the identity these keys cannot be used to authenticate the other sensor in the network. so this probabilistic keying protocol cannot defend against impersonation.

In grid keying protocol [9], [10],[11][12],each sensor is assigned an identifier which coordinates a distinct node in a two-dimensional grid. And also each symmetric key is assigned an identifier which is the coordinates of a distinct node in two-dimensional grid Then a sensor x stores a symmetric key K iff the identifiers of x and K satisfy certain given relation. When two adjacent sensors exchange messages they identify which keys are in common and use a combination of the common keys as a symmetric key. By using this symmetric key encryption and decryption of data can provide secure communication. This grid keying protocol can defend against eavesdropping and impersonation attack but it is vulnerable to collision attack. each sensor in this protocol needs to store only O(log n) symmetric keys. A small gang of adversarial sensors in the network can pool their stored keys together and use the pooled keys to decrypt all the exchanged data messages in the sensor network.

IV. PROPOSED SYSTEM

In this paper, we propose a new keying protocol here each sensor stores only (n+1)/2 keys which is less than (n-1) symmetric keys and this new keying protocol is secure against impersonation, eavesdropping and collision. By using symmetric keys each adjacent pair can encrypt and decrypt the data. By this encryption and decryption we can provide secure communication. To defend against impersonation eavesdropping type of attacks sensors x and y needs to share a symmetric key Kx, y or Kim .The shared key Kx, y need to be stored in both x , y not in any other sensors in the network before these two sensors are deployed in the network. The adjacent sensors can use the shared symmetric key Kx,y to perform the two functions mutual authentication and confidential data exchange.

A. Methodology

Let n is the number of sensors in the network and assume that n is odd positive integer. Each sensor in the network has a unique identifier in the range 0. . . n - 1. We use ix and iy to denote the identifiers of sensors x and y.

Mutual Authentication:

Sensor x authenticates sensor y, and sensor y authenticates sensor x.Mutual Authentication has to be done to ensure that the communication between sensor x and y is genuine not impersonator.

Confidential Data Exchange:

Encrypt and later decrypt all the exchanged data messages between x and y. The sensors x and y can become neighbours in the network in two occasions.

1. The two sensors x and y could be mobile and their movements cause them to become adjacent to one another.
2. The two sensors could be stationary and they are deployed adjacent to one another.

The shared symmetric keys are designed to implement a “special structure”, (n+1)/2 shared symmetric keys are required to store in each senor. Before the implementation of this special structure we have to know two new concepts: 1.universal keys, 2. circular relation. Each sensor x in the network stores a symmetric key, called the universal key of sensor x. The universal key of sensor x, is denoted by ux, is known only to sensor x.

B. Mutual Authentication protocol

If two sensors x,y are adjacent to one another then these two sensors need to execute a mutual authentication protocol so that sensor x proves to sensor y that it is indeed sensor x and sensor y proves to sensor x that it is indeed sensor y. Before the sensors are deployed in a network, each sensor x is supplied with the following items:

1) ix is the distinct identifier in the range 0: n-1
2) One universal key ux
3) (n-1)/2 symmetric keys Kx;y = H(ix,uy)

Each of which is shared between sensor x and another sensor y, where ix is below iy After every sensor is supplied with these items, the sensors are deployed in random locations in the network

```
x→ y : hello(ix, nx)
x← y : hello(iy, ny)
x→ y : verify(ix, iy, H(ix| iy|ny|Kx;y))
x← y : verify(iy, ix, H(iy|ix|nx|Kx;y))
x↔y: H(iy| ix|nx|Kx;y) and H(iy| ix |nx|Kx;y).
 y↔x:  H(ix| iy|ny |Kx;y) and H(ix |iy |ny |Kx;y).
```

(1) Algorithm for mutual authentication Protocol

Sensor X selects a random nonce nx and sends a welcome message to y.After receiving the message from x.sensor y selects a random nonce ny and sends a welcome message to x.

Sensor x determines whether ix is below iy. Then it either fetches Kx;y from its memory or computes it. Finally, sensor x sends a verify message to sensor y.

Sensor y determines whether iy is below ix. Then it either fetches Kx;y from its memory or computes it. Finally, sensor y sends a verify message to sensor x.

Sensor x computes H(iy ix nxKx;y) and compares it with the received H(iy ix nxKx;y). If both are equal then senor x confirms that it is sensor y. If not the conclusion cannot be reached.

Sensor y computes H(ix iynyKx;y) and compares it with the received H(ixiynyKx;y). If both are equal then senor y confirms that it is sensor x. If not the conclusion cannot be reached.

C. Data Exchange Protocol

Encryption and decryption for exchanged data messages. After completion of mutual authentication to sensors x and y, sensors x and y can now start exchanging data messages according to the following data exchange protocol. nx and ny are the two nonce's that were selected at random by sensors x and y, respectively, in the mutual authentication protocol.

$x \rightarrow y$: data(ix, iy, Kx,y (ny text))
$x \leftarrow y$: data(iy, ix, Kx,y (nx text))

(2) Data Exchange protocol

Sensor x concatenates the nonce ny with the text of the data message to be sent, encrypts the concatenation using the symmetric key Kx;y, and sends the result in a data message to sensor y.

Sensor y concatenates the nonce nx with the text of the data message to be sent, encrypts the concatenation using the symmetric key Kx;y, and sends the result in a data message to sensor x.

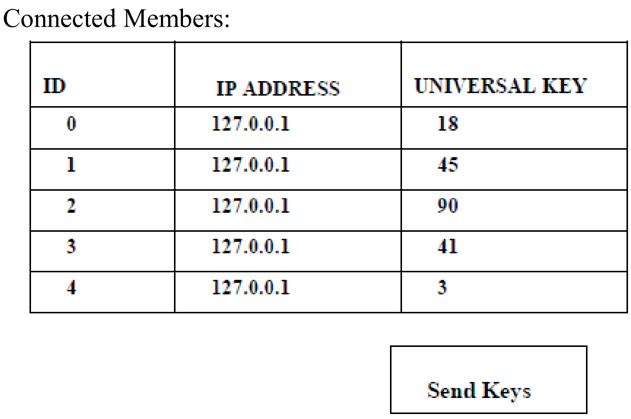
Sensors x and y can repeat Steps 1 and 2 any number of times to exchange data between themselves.

D. Data Security

The main idea for keying Technique is the segregation of key for encryption at sender's end and decryption at receiver's end. When the sender sends a message to the receiver, the receiver must have the decryption key to decrypt the encrypted message. The Encrypted message can only be read by the receiver provide the decryption key.

AES is the Advanced Encryption Standard which uses symmetric key encryption technique. AES provides strong encryption and is very secure to protect information [13][14].

V. EXAMPLE

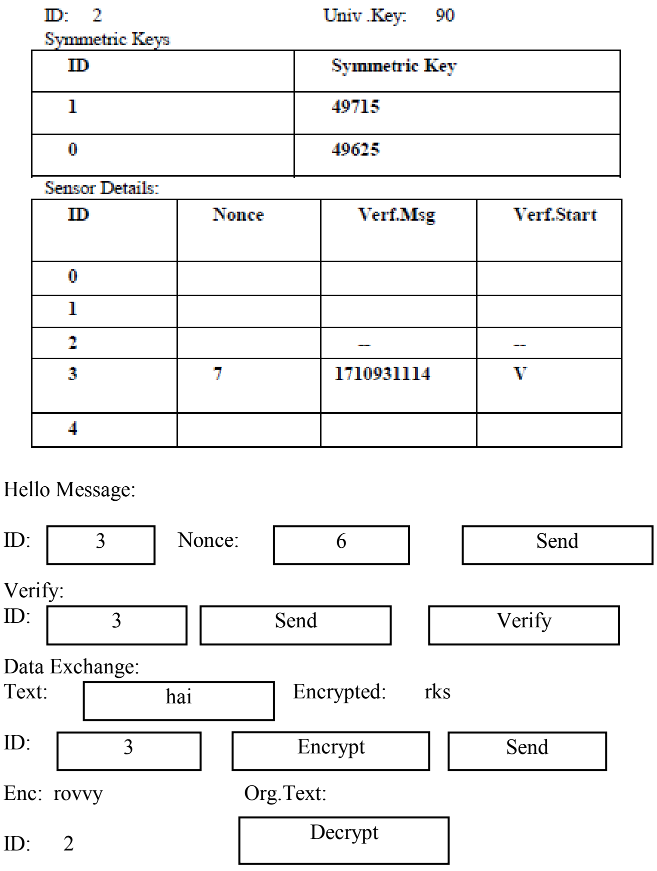


Connected Members:

ID	IP ADDRESS	UNIVERSAL KEY
0	127.0.0.1	18
1	127.0.0.1	45
2	127.0.0.1	90
3	127.0.0.1	41
4	127.0.0.1	3

Figure (1) Sender

To understand how this scheme works we present a simple example here. Each sensor is supplied with one distinct identifier, universal key and Symmetric key.



ID: 2 Univ .Key: 90

Symmetric Keys

ID	Symmetric Key
1	49715
0	49625

Sensor Details:

ID	Nonce	Verf.Msg	Verf.Start
0			
1			
2		--	--
3	7	1710931114	V
4			

Hello Message:

ID: 3 Nonce: 6

Verify:

ID: 3

Data Exchange:

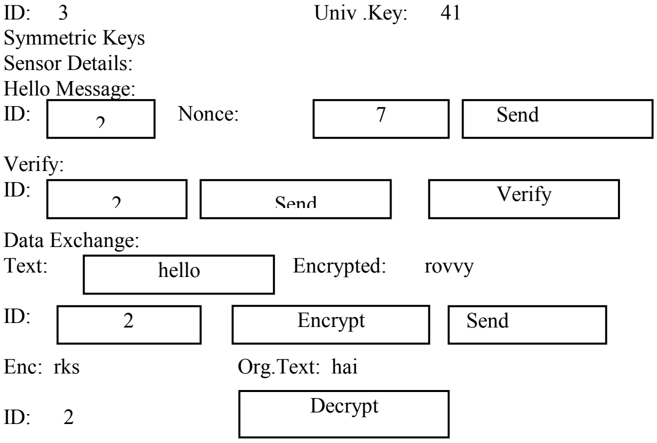
Text: hai Encrypted: rks

ID: 3

Enc: rovvy Org.Text:

ID: 2

Figure(2) Receiver2



ID: 3 Univ .Key: 41

Symmetric Keys

Sensor Details:

Hello Message:

ID: 2 Nonce: 7

Verify:

ID: 2

Data Exchange:

Text: hello Encrypted: rovvy

ID: 2

Enc: rks Org.Text: hai

ID: 2

Figure (3) Receiver3

let us take four receivers in which receiver2 and receiver3 communicate each other. Receiver2 selects a random nonce and sends a hello message to Receiver3.After receiving the message from receiver2.Receiver3 selects a random nonce and sends a message to Receiver2 here the receiver2 and 3 verifies each other. After completion of this mutual authentication now these two exchange data messages. here the Receiver2 sends a hello message that is encrypted using the symmetric key as rovvy to Receiver3. now the Receiver3 decrypt the encrypted message. Symmetric keys are used for both encryption of plaintext and decryption of cipher text.

VI. RESULT ANALYSIS

1) Efficiency:

There is a keying protocol, where each sensor shares a distinct symmetric key with every other sensor in the network, and yet each sensor needs to store exactly (n+1)/2 symmetric keys, before the network is deployed.

2) Optimality:

In every keying protocol, where each sensor shares a distinct symmetric key with every other sensor in the network, each sensor needs to store at least (n-1)/2 symmetric keys, before the network is deployed.

VII. CONCLUSION

In a network, for secure communication each sensor is required to store (n-1) shared symmetric keys. The shared symmetric keys that are stored in the network is n(n-1). , the optimal number of shared symmetric keys for secure communication is n(n-1)/2.

In past so many approaches were introduced to reduce the number of shared keys but the security level is low and they are vulnerable to collusion. In this paper, we provide the secure keying protocol for sensor networks, needs to store only (n + 1)/2 shared symmetric keys between two sensors. The number of shared symmetric keys stored in a sensor network with n sensors is n(n + 1)/2, which is close to the optimal number of shared symmetric keys for any key distribution scheme that is not vulnerable to collusion.

VIII. FUTURE ENHANCEMENT

As we are just transferring only the text messages, we can also transfer the files, images etc. As we are using MUTUAL AUTHANTICATION PROTOCOL for the communication we can also use OPTIMALITY OF KEYING PROTOCOL. In this project we have been used symmetric keys for the communication we can also use ID of the sensors for the communication.

REFERENCES

[1] Dorothy Denning. *Cryptography and Data Security*. Addison-Wesley, 1982

[2] G. Gaubatz, J.-P. Kaps, and B. Sunar. Public key cryptography in sensor networks - revisited. In ESAS, pages 2{18, 2004.

[3] A. Howard, M. J. Mataric, and G. S. Sukhatme, “Mobilesensor network deployment using potential fields: Adistributed, scalable solution to the area coverage problem,”in Proceedings of the International Symposium onDistributed Autonomous Robotic Systems (DARS), 2002, pp. 299–308.

[4] S. Sana and M. Matsumoto, “Proceedings of a wirelesssensor network protocol for disaster management,” inInformation, Decision and Control (IDC), 2007, pp. 209–213.

[5] S. Hynes and N. C. Rowe, “A multi-agent simulationfor assessing massive sensor deployment,” Journal ofBattlefield Technology, vol. 7, pp. 23–36, 2004.

[6] B. Dahill et al., “A Secure Protocol for Ad Hoc Networks,“ IEEE ICNP, 2002X.

[7] Du, H. Chen, "Security in Wireless Sensor Networks", IEEE Wireless Communications, 2008.

[8] L. Eschenauer and V. D. Gligor, “A key-managementscheme for distributed sensor networks,” in Proceedingsof the 9th ACM Conference on Computer and communicationssecurity (CCS), 2002, pp. 41–47.

[9] L. Gong and D. J. Wheeler, “A matrix key-distributionscheme,” Journal of Cryptology, vol. 2, pp. 51–59,January 1990.

[10] S. S. Kulkarni, M. G. Gouda, and A. Arora, “Secret instantiationin ad-hoc networks,” Special Issue of ElsevierJournal of Computer Communications on DependableWireless Sensor Networks, vol. 29, pp. 200–215, 2005.

[11] A. Aiyer, L. Alvisi, and M. Gouda, “Key grids: Aprotocol family for assigning symmetric keys,” in Proceedingsof IEEE International Conference on NetworkProtocols (ICNP), 2006, pp. 178–186.

[12] E. S. Elmallah, M. G. Gouda, and S. S. Kulkarni,“Logarithmic keying,” ACM Transactions on AutonomicSystems, vol. 3, pp. 18:1–18:18, December 2008.

[13] J. Granjal, R. Silva, J. Silva, "Security in Wireless Sensor Networks", CISUC UC, 2008

[14] A. Perrig, R. Szewczyk, V.Wen, D. Culler, and J. Tygar, "SPINS: Security protocols for sensor networks," in Proceedings of MobileNetworking and Computing 2001, 2001.