

SECURED knowledge TRANSMISSION By Using Minimal KEY EXCHANGE MECHANISM FOR WIRELESS device NETWORKS

Sai Krishna Bokam, Durga Malleswari, Shankar Reddy, Amith

Abstract: Many wireless device networks (mainly networks of mobile sensors or networks that are deployed to watch cataclysm situations) are deployed in Associate in Nursing arbitrary and unplanned fashion. For any device in such a network will find yourself being adjacent node to the other device node within the network. to determine a secure communication between each combine of adjacent sensors node in such a network, every device node x within the network has to store $n - 1$ range of bilaterally symmetric keys that device node x shares with all the opposite device nodes, wherever n is that the range of device nodes within the gift network.

This memory storage demand of the keying protocol is varied, particularly once n is massive and therefore the accessible storage in every device node is modest. Previous efforts to revamp this keying protocol and scale back the amount of keys to be hold on in every device node have made protocols that are liable to impersonation, eavesdropping, and collusion attacks.

In this paper, we have a tendency to gift a totally secure protocol mechanism wherever every device node has to store $(n+1)/2$ keys, that is far but the $n-1$ keys that require to be hold on in every device within the original communication protocol.

we have a tendency to conjointly show that in any absolutely secure keying protocol, every device node has to store a minimum of $(n - 1)/2$ keys.

Keywords: sensor, key sender, encryption, decryption, sensor networks, MWSN, plain-text, cipher-text, grid, probabilistic, keying protocol.

1. Introduction

When sensing element nodes wish to speak with one another, there has got to be a secure association that has got to be established between nodes.

Constant is that the case with the sensing element nodes. Communication between sensing element nodes has got to be shielded from external attacks once establishing a session of communication between one another. Communication nodes (Sensors) area unit small devices with little size, less computation power, and a transmission vary.

Moreover, the positions of Communication nodes (Sensors) needn't be constant and they'll be moving or

dynamically shift with relevance time. So, it's ineluctable that the info has to be transferred and unbroken correct and secure. Within the period once sensors were initial introduced there have been several issues associated with security.

The info that was being communicated Either was too huge in volume or the secure transmission was suffering a scarcity of correct care.

Therefore, the conception of Cryptography was introduced.

Cryptography ensured that the info that's being communicated is secure and additionally the info size is additionally little.

Sensors: Sensor's area unit called devices that calculate a physical amount and convert it into an electrical wave signal which may be understood and skim by a receiver or a tool.

it's typically required to speak confidential knowledge firmly while not being attacked by external world. Examples: Heat Sensors, measuring system (Battle fields), Security Alarm Systems

Sensor Networks: A sensor network is designed with a communication infrastructure planned to record, monitor conditions at multiple locations. Multiple detection stations called sensor nodes exist in a sensor network, each of which is lightweight, tiny, and easy to carry devices. Every sensor node is equipped with a power source and transducer. The transducer produces electrical signals based on sensed phenomena and physical effects. The microcomputer processes and stores the sensor output. The power for each sensor device is obtained from the electric utility or from a battery.

Mobile wireless sensor networks (MWSNs):

Mobile wireless device networks aka MWSNs area unit sometimes outlined as a wireless device network (WSN) within which the device devices (nodes) area unit simply movable.

MWSNs area unit AN rising field of analysis in distinction to their well-established forerunner and that they area unit smaller. Several of their applications area unit similar, like police work or environmental watching.

Key distribution within Sensor Networks:

Within wireless detector network (WSN) style Key distribution is a crucial issue. thanks to power and memory limitations, the keys have to be compelled to be organized to create a totally practical network.

Key distribution (distribution of keys to detector nodes) happens before readying.

Initially, sensors observe they're within sight sensors within the network and transmit the knowledge to the key sender. For secure transmission, the Key Sender then generates secret keys and sends them to individual detector nodes.

Encryption and Decryption of data:

In cryptography, secret writing is that the method of coding messages or info in such some way that solely approved parties will scan it

In Associate in Nursing secret writing theme, the message or info, mentioned as plaintext, is encrypted victimization Associate in Nursing secret writing algorithmic rule, turning it into Associate in Nursing undecipherable ciphertext.

Decryption: the data that has been encrypted into a secret format is understood because the method of secret writing. Decoding needs a secret key or parole; therefore solely licensed users will solely decode knowledge. In easy that means, it's the conversion of cipher text into plain text.

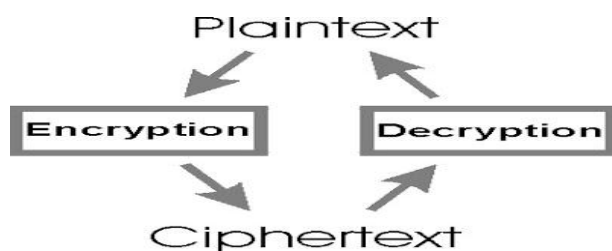


Figure 1: Conversion of plain text to cipher text and vice versa

Two main protocols were advising within the past to reduce the quantity of hold on keys in every sensing element node within the network. We tend to sit down with these 2 protocols because the subjective Grid Keying Protocol and Keying Protocol. Every sensing element within the network stores multiple keys that square measure chosen willy-nilly from an oversized set of keys within the Probabilistic Keying Protocol.

Once 2 aspects by sensing element nodes got to communicate, the 2 sensors nodes acknowledge the shared keys then use a mix of their common keys as a

regular key to cypher and decode their transferred knowledge messages. Since a selected sensing element node solely encompasses a set of shared keys and doesn't have its own universal key it's at risk of impersonation attack.

Every sensing element is allotted a singular number that is employed to coordinate a definite node during a exceedingly in a very} two-dimensional house and every regular key's additionally allotted an symbol that is thought as Grid Keying Protocol,

Then a sensing element node x stores a regular key K if and as long as the identifiers of x and K satisfy definitely given relation.

Once two adjacent sensors have to be compelled to transfer the info, the 2 sensing element nodes establish common keys and use a mix of these shared keys as a regular key to decode and cypher knowledge messages.

The grid keying protocol has 2 benefits. First, this protocol will defend against impersonation - global organization like the probabilistic protocol and might defend against eavesdropping - just like the probabilistic protocol.

Second, every sensing element node during this protocol should store solely $O(\log n)$ regular keys, wherever n is that the no. of sensors within the network. However, the most issue with grid keying protocol is it should not shield itself from collision attack.

In our proposed paper we want to show that there could be a system with a keying protocol that reduces the number of keys maintained within the sensor to $(n+1)/2$ keys. The extra and very predominant feature that has been introduced is that of a Key Sender. All Sensor Networks has a corresponding Key Sender associated with it. Every sensor node has to be registered within the Key Sender. The Key Sender then distributes/shares the keys to the sensor nodes within the network. With the help of the keys distributed by the Key Sender the sensors communicate and transfers with the other sensors.

We use i_x and i_y to symbolize the identifiers of sensor nodes 'x' and 'y', respectively, in this network. Each two sensors, say sensor nodes 'x' and 'y', share a symmetric key denoted $K(x, y)$ or $K(y, x)$. Only the two sensor nodes 'x' and 'y' know their shared key $K(y, x)$. And if sensor nodes 'x' and 'y' ever become neighbors in the network, then they can use their shared symmetric key $K(y, x)$ to perform two functions:

1) **Mutual Authentication:** Sensor 'x' authenticates sensor 'y', and sensor y authenticates sensor 'x'.

2) Confidential Data Exchange: Encrypt and then decrypt all the transferred data messages between 'x' and 'y'. In the rest of this section, we wanted to show that if the shared symmetric keys are designed to have a "special structure", then each sensor needs to store only $(n+1)/2$ shared symmetric keys. We need to introduce two new concepts before we present the special structure of the shared keys, : "Universal Keys" and "a circular relation, named below, over the sensor identifiers". Each sensor node 'x' in the network keeps a symmetric key, called the universal key of sensor 'x'. The universal key of sensor node 'x', denoted 'ux', is known only to sensor node 'x'. Let I_x and I_y be two distinct sensor node identifiers. Identifier I_x is said to be below identifier I_y if one of the below conditions holds:

- 1) $I_x < I_y$ and $(I_y - I_x) < n/2$
- 2) $I_x > I_y$ and $(I_x - I_y) > n/2$

The below relation is better explained by an example. Consider the case where $n / 3$. In this case, the sensor identifiers are 0, 1, 2

We have:

- Identifier 0 is below identifiers 1 and 2.
- Identifier 1 is below identifiers 2 and 0.
- Identifier 2 is below identifiers 1 and 0.

2. Methodology

Theorem 1: If there exists a try of distinct however adjacent sensors 'x' and 'y' with distinctive identifiers 'Ix' and 'Iy' severally then the Below condition holds true as follows :-

- 'Ix' is below 'Iy'
- 'Iy' is below 'Ix'

Theorem 2: Since there exists 'n' sensors, every detector 'x' with symbol I_x has $(n-1)/2$ detector identifiers I_y below it.

Theorem 3: In accordance with Theorem one, the number of detectors with symbols I_x below the sensor 'y' with identifier I_y is $(n-1)/2$.

Theorem 4: If a detector symbol I_x for detector 'Ix' is below a detector symbol 'Iy' then the detector 'x' has to store the biradial Key 'ky, x' / $H(I_x|I_y)$ among it. Then the detector 'y' has to reckon the biradial Key to verify the detector 'x'. The biradial Key 'k(y, x)' is hold on solely in 'x'.

Theorem 5: As mentioned earlier, every detector 'x' has to store single Universal Key and $(n-1)/2$ biradial Keys

'k(y, x)' so as to speak with detector 'y' (NOTE: the detector symbol I_x ought to be below 'Iy').

3. A Mutual Authentication Protocol:

Each and each device 'x' are given the subsequent data before the sensors square measure deployed inside the network: -

- 1) One distinct symbol I_x within the vary $0-(n-1)$
- 2) One universal key married woman
- 3) $(n-1)/2$ trigonal keys $K(y, x) / H(I_x|I_y)$ every of that is shared between device 'x' and another device 'y' (where I_x is below I_y). If the sensors 'x' and 'y' square measure adjacent and need to speak with one another, then they need to implement the Mutual Authentication protocol that has the subsequent steps:-

Step 1: device 'x' selects a random present n_x and sends a hullo message that's received by device 'y'. $x \rightarrow y$: hullo('Ix'| n_x)

Step 2: device 'y' selects a random present Empire State and sends a hullo message that's received by device 'x'. $x \rightarrow y$: hullo ('Iy'| n_y)

Step 3: device 'x' determines whether or not I_x is below I_y . Then it either fetches $K(y, x)$ from its memory or computes it. Finally, device 'x' sends a verify message to device 'y'. $x \rightarrow y$: verify ('Ix'; 'Iy'; $H(I_x | I_y | n_y | K(y, x))$)

Step 4: device 'y' determines whether or not I_y is below I_x . Then it either fetches Bluegrass State, x from its memory or computes it. Finally, device 'y' sends a verify message to device 'x'. $x \rightarrow y$: verify ($I_y | I_x | H(I_y | I_x | n_x | K(y, x))$)

Step 5: device 'x' computes $H(I_y | I_x | n_x | K(y, x))$ and compares it with the received $H(I_y | I_x | n_x | K(y, x))$. If they're equal, then device 'x' concludes that the device claiming to be device 'y' is so device 'y'. Otherwise, no conclusion is reached.

Step 6: device 'y' computes $H(I_x | I_y | n_y | K(y, x))$ and compares it with the received $H(I_x | I_y | n_y | K(y, x))$. If they're equal, then 'y' concludes that the device claiming to be device 'x' is so device 'x'. Otherwise, no conclusion is reached.

4. A Data Exchange Protocol:

Sensors 'x' and 'y' can now start exchanging data according to the following data exchange protocol:-

Step 1: Sensor 'x' combines the nonce n_y with the data to be sent, encrypts the combined data using the symmetric

key $K_{(y, x)}$, and sends the result in a data message to sensor 'y'.

$$x \rightarrow y: \text{data}(I_x | I_y | K_{(y, x)}(n_x | \text{text}))$$

Step 2: Sensor 'y' combines the nonce n_x with the data to be sent, encrypts the combined data using the symmetric key $K_{y,x}$, and sends the result in a data message to sensor 'x'.

$$x \rightarrow y: \text{data}(I_y | I_x | K_{(y, x)}(n_x | \text{text}))$$

5. Optimality of Keying Protocol:

According to our keying protocol, described in Section III, each sensor in the network is required to store only $(n+1)/2$ keys. Thus, the total number of keys that need to be stored within the network is $n(n+1)/2$.

Theorem 6: There should be a minimum of $n(n-1)/2$ keys that are to be stored within the sensor network.

Theorem 7: According to any keying protocol (which is uniform) has to store at least $(n-1)/2$ keys within it to communicate with its adjacent sensors.

6. Triple Data Encryption Algorithm (TDEA):

DES (the encryption Standard) could be a cruciform block cipher developed by IBM. The formula uses a 56-bit key to encipher/decipher a 64-bit block of information. The secret is continually conferred as a 64-bit block, each eighth little bit of that is neglected.

However, it's usual to line every eighth bit so every cluster of eight bits has Associate in Nursing odd range of bits set to one. The formula is best suited to implementation in hardware, in all probability to discourage implementations in computer code, that tend to be slow by comparison.

However, fashionable computers square measure thus quick that satisfactory computer code implementations square measure without delay offered is that the most generally used cruciform formula within the world, despite claims that the key length is just too short.

Ever since DES was 1st declared, contention has raged concerning whether or not fifty-six bits is long enough to ensure security. The key length argument goes like this. Forward that the sole possible attack on DES is to undertake every key successively till the correct one is found, then 1,000,000 machines every capable of testing 1,000,000 keys per second would realize (on average) one key each twelve hours.

Most affordable folks may realize this rather comforting and a decent live of the strength of the

formula. Those who consider the exhaustive key-search attack to be a real possibility (and to be fair the technology to do such a search is becoming a reality) can overcome the problem by using double or triple length keys. In fact, double length keys have been recommended for the financial industry for many years.

Use of multiple length keys leads us to the Triple-DES algorithm, in which DES is applied three times. If we consider a triple length key to consist of three 56-bit keys $K1, K2, K3$ then encryption is as follows:

- Encrypt with $K1$
- Decrypt with $K2$
- Encrypt with $K3$

Decryption is the reverse process:

- Decrypt with $K3$
- Encrypt with $K2$
- Decrypt with $K1$

Setting $K3$ equal to $K1$ in these processes gives us a double length key $K1, K2$. Setting $K1, K2$ and $K3$ all equal to K has the same effect as using a single-length (56-bit key). Thus it is possible for a system using triple-DES to be compatible with a system using single-DES.

7. Data Analysis

Key Sender: It detects the sensors gift in its region and updates their details in its table. Here we have a tendency to used java simulation to form associate interface to key-sender. Symbolically it should look like (before sensors detection and when detection)

ID	IP Address	Universal Key

Figure3: KeySender table before detection

ID is that the distinctive variety given to every device, IP address is that the logical address and universal is the even key used for coding and decoding

Key Sender Table once device detection:

ID	IP Address	Universal Key
0	127.0.0.1	98
1	127.0.0.1	8

Figure4: Key-sender table after clients/sensors are detected

Here we take 2 sensor nodes or clients (say receiver 0 and receiver 1) which are detected by the key sender. The key sender then calculates the universal keys(as shown in fig-4) and sends them to the respective clients

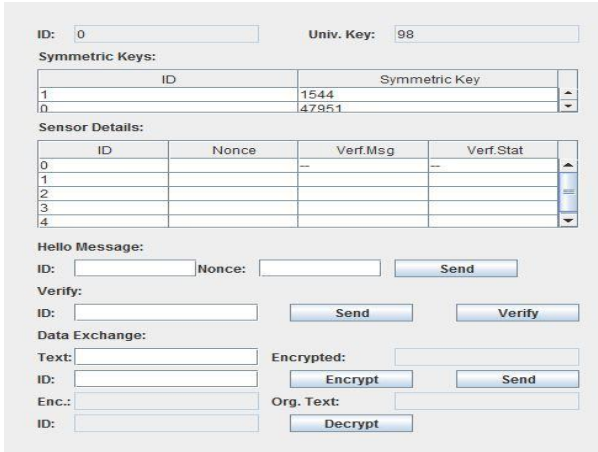


Figure 5 shows the interface for Receiver 0. The 'Univ. Key' is 98. The 'Symmetric Keys' table lists keys for ID 1 (1544) and ID 0 (4795.1). The 'Sensor Details' table shows verification messages for ID 1 (1087523401) with a status of 'V'.

ID	Nonce	Verf.Msg	Verf.Stat
0	--	--	--
1	22	-1087523401	V
2	--	--	--
3	--	--	--
4	--	--	--

Figure5:Receiver0

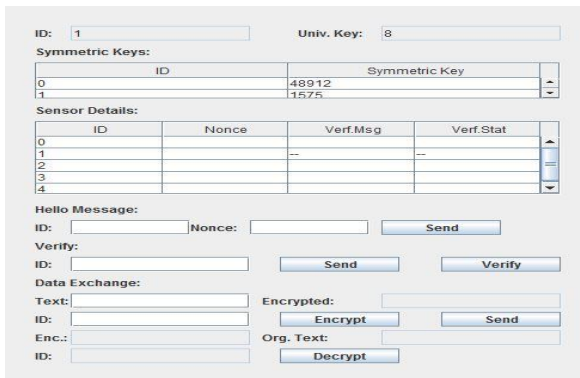


Figure 6 shows the interface for Receiver 1. The 'Univ. Key' is 8. The 'Symmetric Keys' table lists keys for ID 0 (48912) and ID 1 (1575). The 'Sensor Details' table shows verification messages for ID 0 (627086749) with a status of 'V'.

ID	Nonce	Verf.Msg	Verf.Stat
0	--	--	--
1	22	627086749	V
2	--	--	--
3	--	--	--
4	--	--	--

Figure6:Receiver1

After the key sender sends the symmetric keys to both the receivers the clients now look like

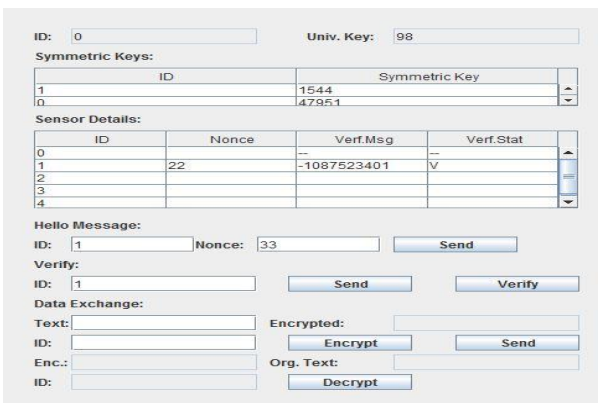


Figure 7 shows Receiver 0 with updated keys. The 'Univ. Key' is 98. The 'Symmetric Keys' table lists keys for ID 1 (1544) and ID 0 (4795.1). The 'Sensor Details' table shows verification messages for ID 1 (1087523401) with a status of 'V'.

ID	Nonce	Verf.Msg	Verf.Stat
0	--	--	--
1	22	-1087523401	V
2	--	--	--
3	--	--	--
4	--	--	--

Figure7:Receiver0 with keys updated

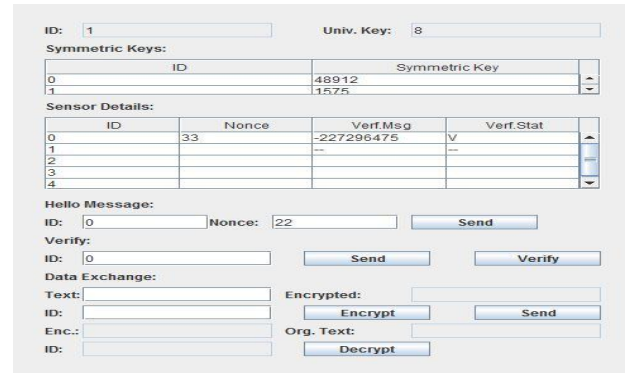


Figure 8 shows Receiver 1 with updated keys. The 'Univ. Key' is 8. The 'Symmetric Keys' table lists keys for ID 0 (48912) and ID 1 (1575). The 'Sensor Details' table shows verification messages for ID 0 (227296475) with a status of 'V'.

ID	Nonce	Verf.Msg	Verf.Stat
0	33	-227296475	V
1	--	--	--
2	--	--	--
3	--	--	--
4	--	--	--

Figure8:Receiver1 with keys updated

After the keys square measure is updated each of the nodes has to be compelled to attest to one another for that they send verification messages and ensure the authentication.

once authentication is finished message transfer is finished victimization secret writing and decoding

Message transfer done by Receiver0-original message is hello, it is encrypted and sent. The encrypted message from Receiver1 is decrypted

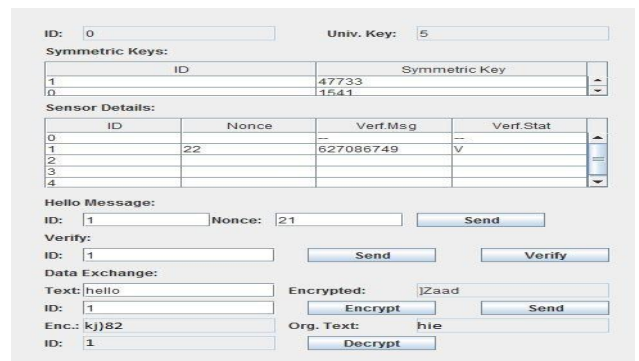


Figure 9 shows Receiver 0 during message exchange. The 'Univ. Key' is 5. The 'Symmetric Keys' table lists keys for ID 1 (47733) and ID 0 (1541). The 'Sensor Details' table shows verification messages for ID 1 (627086749) with a status of 'V'. The 'Data Exchange' section shows the original text 'hello' being encrypted to 'jZaad'.

Figure9: Receiver0 sending and receiving messages

Message transfer done by Receiver1-original message is hie, it is encrypted and sent. The encrypted message from Receiver0 is decrypted

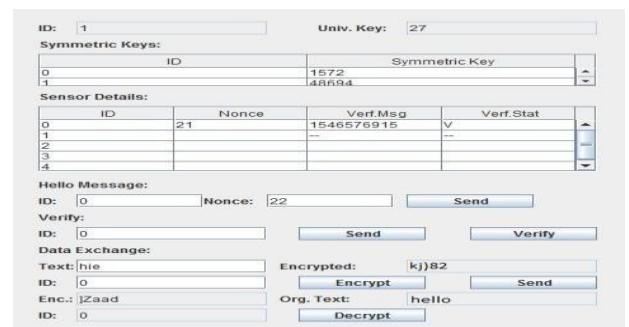


Figure 10 shows Receiver 1 during message exchange. The 'Univ. Key' is 27. The 'Symmetric Keys' table lists keys for ID 0 (1572) and ID 1 (46534). The 'Sensor Details' table shows verification messages for ID 0 (1546576915) with a status of 'V'. The 'Data Exchange' section shows the original text 'hie' being encrypted to 'kj)82'.

Figure10: Receiver1 sending and receiving messages

Whenever a receiver0 desires to speak with receiver2, it cannot communicate directly, 1st of all the receiver0 should communicate with receiver1 so the receiver1 communicates that message with receiver2

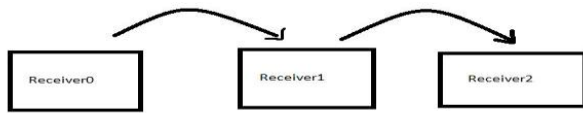


Figure9: nodes communication

8. CONCLUSION:

Typically, each sensor in a sensor network with n sensors needs to store $n - 1$ shared symmetric keys to communicating securely with each other. Thus, the number of shared symmetric keys stored in the sensor network is $n(n - 1)$. However, the optimal number of shared symmetric keys for secure communication, theoretically, is $(n^2) = n(n - 1)/2$.

Although there have been many approaches that attempt to reduce the number of shared symmetric keys, they lead to a loss of security: they are all vulnerable to collusion.

In this paper, we show the Secure minimal or Lightweight Key Agreement protocol for sensor networks, that needs to store only $(n + 1)/2$ shared symmetric keys to each sensor. 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 minimal or Lightweight Key Agreement protocol for sensor networks.

9. REFERENCES

Communication within Sensor Networks by Using Key Distributor by ch.d. naiduijsit 2014.

The remaining references are:

1. Taehwan Choi, H. B. Acharya, and Mohamed D. Gouda, "The Best Keying Protocol ", December 2011 IEEE
2. <http://en.wikipedia.org/wiki/Sensor-Sensors>
3. Sensor Networks by Margaret Rouse.
4. <http://en.wikipedia.org/wiki/Mobilewirelessnetwork-MWSNs>
5. "Key Distribution Mechanisms for Wireless Sensor Networks" by Seyit A., C.Amtepe and BulentYener
6. Communication within Sensor Networks by Using Key Distributor by M.V. Kishore in International Journal of Computer Science and Information Technologies, Vol. 5 (4), 2014, 4906-4910
7. "Cryptography and Network Security", Fourth Edition by William Stallings
8. L. Gong and D. J. Wheeler, "A matrix key-distribution scheme," Journal of Cryptology, vol. 2, pp. 51-59, January 1990.
9. "Advanced Encryption Standard" by Douglas Select
10. http://en.wikipedia.org/wiki/Advanced_Encryption_Standard#cite_note-fips-197-4.
11. "Comparative Study of Energy-Aware QoS for Proactive and Reactive Routing Protocols for Mobile Ad-hoc Networks". International Journal of Computer Applications (0975 - 8887) Volume 31- No.5, October 2011.
12. Steganography Detection using Functional Link Artificial Neural Networks, International Journal of Computer Applications (0975 - 888), Volume 47 No.5 June 2012.
13. Secure Group Communication using Multicast Key Distribution Scheme in Ad-hoc Network, International Journal of Computer Applications. (0975 - 8887) Volume 1 - No. 25, Nov-2010.
14. A Novel Dual-Phase Mechanism for Data Transmission to Provide Compression and Security by M.V. Kishore in International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 12, December 2013 ISSN: 2277 128X

BIOGRAPHIES



Dr.K.N.S. LAKSHMI Currently working as Professor from Department of computer science and Engineering at Sanketika Vidya Parishad Engineering college



B. SAI KRISHNA Pursuing B-tech from Department of computer science and Engineering at Sanketika Vidya Parishad Engineering college



P. DURGA MALLESWARI Pursuing B-tech from Department of computer science and Engineering at Sanketika Vidya Parishad Engineering college



R. SHANKAR Pursuing B-tech from Department of computer science and Engineering at Sanketika Vidya Parishad Engineering college



Ch. AMITH Pursuing B-tech from Department of computer science and Engineering at Sanketika Vidya Parishad Engineering college