A new approach for Evolution of end to end Security in Wireless Sensor Network

S. Anjali Devi,
Asst.Professor,
Department of Computer Science & Engineering,
Gayathri Vidya Parishad College of Engineering for Women,
Andhra Pradesh, India.
swarnaanjalidevi@gmail.com
R. Venu Babu,
Asst .Professor,
Department of information technology,
GITAM University,
Andhra Pradesh, India.
Venubabu.r@gmail.com
B. Srinivasa Rao,
Asst .Professor,
Department of information technology,
GITAM University,
Andhra Pradesh, India.
sreenivas.battula@gmail.com

***Abstract:-* A wireless sensor network (WSN) is a network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions such as temperature, sound, vibration, pressure, motion or pollutants, at different locations. Data security is essential for these mission-critical applications to work in unattended and even hostile environments. So, we have to take providing desirable data security, that is, confidentiality, authenticity, and availability, in wireless sensor networks (WSNs) as a challenge. WSN consists of a large number of sensor nodes. These sensor nodes are mini, low-cost, smaller memory sizes and low bandwidth. Existing security designs are vulnerable to many types of Denial of Service (DoS) attacks, such as report disruption attacks and selective forwarding attacks. In this paper, we seek to overcome these vulnerabilities for large-scale static WSNs.**

Key-Words:- Wireless sensor network, routing protocols, sensor nodes, bogus filtering data.

1. Introduction

1.1 Introduction to WSN's:-A wireless sensor network (WSN) is a network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions such as temperature, sound, vibration, pressure, motion or pollutants, at different locations. The development of wireless sensor networks was originally motivated by military applications such as battlefield surveillance. However, wireless sensor networks are now used in many industrial and civilian application areas, including industrial process monitoring and control, machine health monitoring, environment and habitat monitoring, healthcare applications, home automation, and traffic control.

In addition to one or more sensors, each node in a sensor network is typically equipped with a radio transceiver or other wireless communications device, a small microcontroller, and an energy source, usually a battery. A sensor network normally constitutes a wireless ad-hoc network, meaning that each sensor supports a multi-hop routing algorithm (several nodes may forward data packets to the base station).

1.2 Architecture of wireless networks

Infrastructure-based networks

- Cellular mobile communication system
- WLAN

Non-infrastructure networks

- Ad hoc networks
- Wireless Sensor networks

The comparison of two types of wireless networks

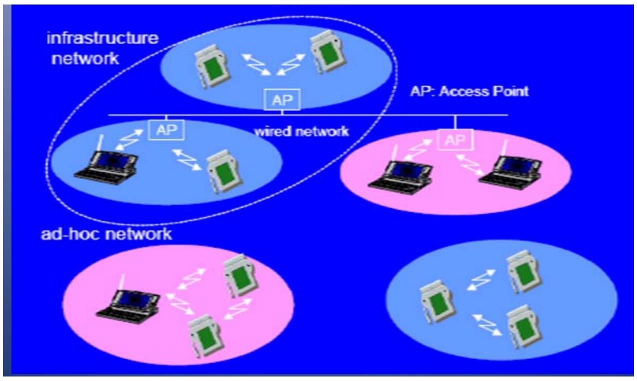


Fig.1.2 comparison of wireless networks

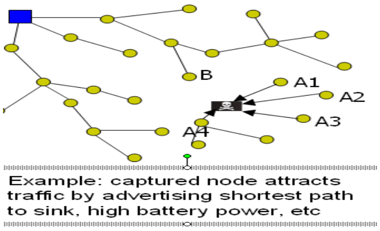
1.3 Attacks on WSN

Main types of attacks on WSN are:

- spoofed, altered, or replayed routing information
- selective forwarding
- wormholes
- HELLO flood attacks
- Acknowledgment spoofing

1.3.1. False routing information

- Injecting fake routing control packets into the network, examples: attract / repeal traffic, generate false error messages
- Consequences: routing loops, increased latency, decreased lifetime of the network, low reliability



Example: captured node attracts traffic by advertising shortest path to sink, high battery power, etc

Fig 1.3.1 false routing

An intruder might add a node to the system that feeds false data or prevents the passage of true data. Such messages also consume the scarce energy resources of the nodes. This type of attack is called "*sleep deprivation torture*".

1.3.2. Selective forwarding

- Multi hop paradigm is prevalent in WSN
- It is assumed that nodes faithfully forward received messages
- Compromised node might refuse to forward packets, however neighbors might start using another route
- More dangerous: compromised node forwards selected packets

1.3.3. Wormholes

- Well placed wormhole can completely disorder routing
- Wormholes can exploit routing race conditions which happens when node takes routing decisions based on the first route advertisement
- Attacker may influence network topology by delivering routing information to the nodes before it would really reach them by multi hop routing
- Even encryption cannot prevent this attack
- Wormholes may convince two nodes that they are neighbors when on fact they are far away from each other

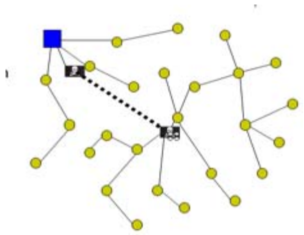


Fig.1.3.3 Wormhole

1.3.4. HELLO flood attack

Many WSN routing protocols require nodes to broadcast HELLO packets after deployment, which is a sort of neighbor discovery based on radio range of the node. Laptop class attacker can broadcast HELLO message to nodes and then advertises high-quality route to sink.

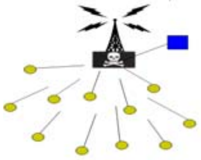


Fig.1.3.4.Hello Flood

1.3.5. Acknowledgment spoofing

- Some routing protocols use link layer acknowledgments.
- Attacker may spoof acknowledgements.
- Consequently weak link may be selected for routing; packets send trough that link may be lost or corrupted.

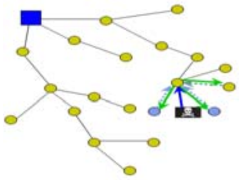


Fig.1.3.5.Acknowledgment Spoofing

2. System Analysis

2.1 Problem Analysis:-

Providing desirable data security, that is, confidentiality, authenticity, and availability, in wireless sensor networks (WSNs) is challenging, as a WSN usually consists of a large number of resource constraint sensor nodes that are generally deployed in unattended/hostile environments and, hence, are exposed to many types of severe insider attacks due to node compromise.

Existing security designs mostly provide a hop-by-hop security paradigm and thus are vulnerable to such attacks. Furthermore, existing security designs are also vulnerable to many types of Denial of Service (DoS) attacks, such as report disruption attacks, Sybil attack,

Selective forwarding attacks, spoofed, altered, or replayed routing information, sinkhole Attack, wormholes, hello flood attacks & acknowledgement spoofing.. Such insider attacks can severely damage network functions and result in the failure of mission-critical applications, induce network congestion, waste the scarce network resources and thus put data availability at stake.

2.2 Proposed Solution:-

In this an integrated security designs providing comprehensive protection over data confidentiality, authenticity, and availability. Our design establishes a location-aware end-to-end data security (LEDS) framework in WSNs.

First here propose a novel location-aware multifunctional key management framework. In LEDS, the targeted terrain is virtually divided into multiple cells using the concept of a virtual geographic grid. Each sensor node obtains its geographic location via a suitable localization scheme. LEDS then efficiently binds the location (cell) information of each sensor into all types of symmetric secret keys owned by that node. What the attacker can do is to misbehave only at the locations of compromised nodes, by which they will run a high risk of being detected by legitimate nodes if effective misbehavior detection mechanisms are implemented.

Second, LEDS provides end-to-end security guarantee. Every legitimate event report in LEDS is endorsed by multiple sensing nodes and is encrypted with a unique secret key shared between the event sensing nodes and the sink. Furthermore, the authenticity of the corresponding event sensing nodes can be individually verified by the sink.

Third, LEDS possesses an efficient en-route false data filtering capability to deal with the infamous bogus data injection attack. As long as there are no more than compromised nodes in each single area of interest, LEDS guarantees that a bogus data report from that cell can be filtered by legitimate intermediate nodes or the sink deterministically.

Last, LEDS provides high-level assurance on data availability by dealing with both report disruption attack [4], [12].and selective forwarding attack [4] simultaneously. By taking advantage of the broadcast nature of wireless links, LEDS adopts a one-to-many data forwarding approach, which is fully compatible with the proposed security framework. That is, all reports in LEDS can be authenticated by multiple next-hop nodes independently so that no reports could be dropped by single node(s).

2.3. Data Security Requirements in WSNs:-

The requirements of data security in WSNs are basically the same as those well defined in the traditional networks, that is, data confidentiality, authenticity, and availability. Data should be accessible only to authorized entities (usually the sink in WSNs), should be genuine, and should be always available upon request to the authorized entities. More specifically, the above three requirements can be further elaborated in WSNs as follows:

2.3.1. Data Confidentiality

Confidentiality means keeping information secret from unauthorized parties. A sensor network should not leak sensor readings to neighboring networks. In many applications (e.g. key distribution) nodes communicate highly sensitive data. The standard approach for keeping sensitive data secret is to encrypt the data with a secret key that only intended receivers possess, hence achieving confidentiality. Since public-key cryptography is too expensive to be used in the resource constrained sensor networks, most of the proposed protocols use symmetric key encryption methods. A sensor network should not leak sensor readings to its neighbors.

2.3.2. Data Authenticity

In a sensor network, data authentication prevents unauthorized parties from participating in the network and legitimate nodes should be able to detect messages from unauthorized nodes and reject them. In the two-party communication case, data authentication can be achieved through a purely symmetric mechanism: The sender and the receiver share a secret key to compute a message authentication code (MAC) of all communicated data. When a message with a correct MAC[2],[4],[16],[9] arrives, the receiver knows that it must have been sent by the sender. If we require a valid report to be collectively endorsed by a number, say, $T(T > 1)$, of sensor nodes that sense the event at the same time, we can protect data authenticity to the extent that no fewer than T compromised nodes can forge a valid report.

2.3.3. Data Availability

Adjusting the traditional encryption algorithms to fit within the wireless sensor network is not free, and will introduce some extra costs. Some approaches choose to modify the code to reuse as much code as possible. Some approaches try to make use of additional communication to achieve the same goal. But all these approaches weaken the availability of a sensor and sensor network for the following reasons:

- Additional computation consumes additional energy. If no more energy exists, the data will no longer be available.
- Additional communication also consumes more energy. What's more, as communication increases so too does the chance of incurring a communication conflict.
- A single point failure will be introduced if using the central point scheme. This greatly threatens the availability of the network.

2.4 End-to-End versus Hop-by-Hop Design:-

In the past few years, many secret key predistribution schemes have been proposed. By leveraging preloaded keying materials on each sensor node, these schemes establish pair wise keys between a node and its neighbors after network deployment for every network node, respectively, and thus form a hop-by hop security paradigm. The security strength of these schemes is analyzed in terms of the ratio of compromised communication links over total network communication links due to node compromise. Two types of node compromise are considered: random node capture[16] and selective node capture[9], according to key distribution information available to the attacker.

Hop-by-hop security design works fine when assuming a uniform wireless communication pattern in WSNs. However, in many applications, node-to-sink communication is the dominant communication pattern in WSNs, that is, data of interest are usually generated from the event happening area and transmitted all the way to the sink. In this case, hop-by-hop security design is not sufficient anymore as it is vulnerable to communication-pattern-oriented node capture attacks[16].

3. Evolution of Security in Wireless Sensor Network

It is an integrated security design providing comprehensive protection over data confidentiality, authenticity, and availability. This design establishes a location-aware end-to-end data security (LEDS) framework in WSNs.

3.1 Design of IGE:-

In this Integrated Geographical Environment the targeted terrain is virtually divided into multiple cells using the concept of **a virtual geographic grid[9],[3],[5]**. It develops a user interface to its relative construction of the required network model. A large number of sensors are scattered over this virtual grid layout. That is, our needed network model and each sensor node obtain its geographic location via a suitable localization scheme.

This will helps to placing the all sensors into those grids or clusters and these sensors having the GPS capabilities to get its required information via longitude and latitude with their positions exactly where those positioned. And, there every cluster fields having maximum of n (n<4) static sensors. One of them is elected and acts as cluster head (CH)[14],[16]. Every sensor has its own id and that information is always kept with cluster heads to monitor vicinity of sensors in same clusters.

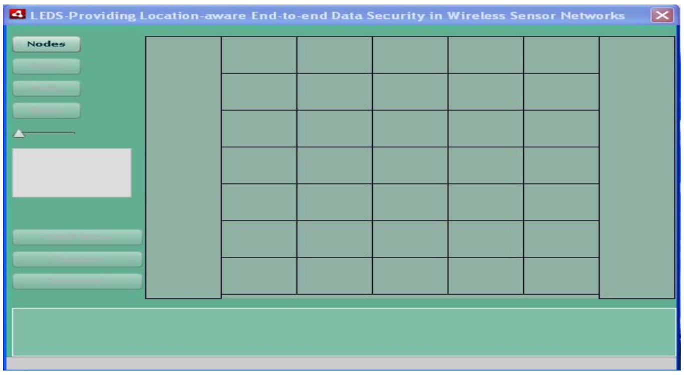

LEDS-Providing Location-aware End-to-end Data Security in Wireless Sensor Networks
Nodes

Figure: 3.1 Application Window Displaying Nodes button

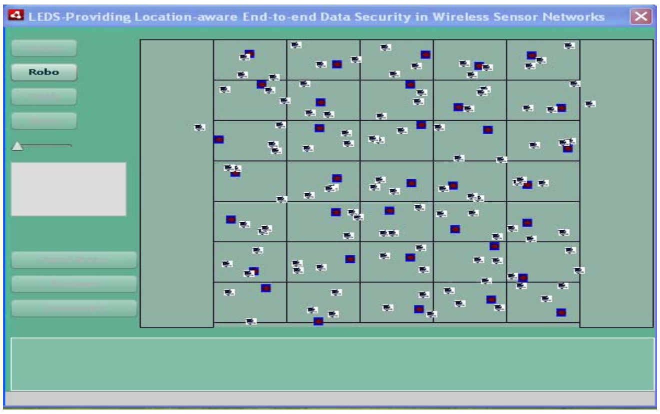

LEDS-Providing Location-aware End-to-end Data Security in Wireless Sensor Networks
Robo

Figure3.1.1 Application Window Displaying Robo button after nodes has been deployed.

3.2 Key Generation and Distribution

This module will target to implement the algorithm of MD5 (Message Digest) which is used to generation of different kinds of security keys for encrypting data for more security providing and sending safely to the destinations.

This LEDS provides end-to-end security guarantees surely. Every legitimate event report in LEDS is endorsed by multiple sensing nodes and is encrypted with a unique secret key shared between the event sensing nodes and the sink. Furthermore, the authenticity of the corresponding event sensing nodes can be individually verified by the sink. This novel setting successfully eliminates the possibility that the compromise of nodes other than the sensing nodes of an event report may result in a security compromise of that event report, which is usually the case in existing security designs.

In this case, each node stores three different types of location-aware keys:

1) A unique secret key shared between the node and the sink that is used to provide node-to-sink authentication.

2) A cell key shared with other nodes in the same cell that is used to provide data confidentiality.

3) A set of authentication keys shared with the nodes in its report-auth cells that are used to provide both cell-to-cell authentication and en-route bogus data filtering.

3.2.1 MD5 Algorithm

The MD5[2],[16] (message-digest) algorithm takes as input a message of arbitrary length and produces as output a 128-bit “fingerprint” or “message digest” of the input. It is conjectured that it is computationally infeasible to produce two messages having the same message digest, or to produce any message having a given prespecified target message digest. The MD5 algorithm is intended for digital signature applications, where a large file must be “compressed” in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem such as RSA.

The MD5 algorithm is designed to be quite fast on 32-bit machines. In addition, the MD5 algorithm does not require any large substitution tables. The MD5 algorithm is an extension of the MD4 message-digest algorithm.

3.2.1.1 MD5 Algorithm Description

We begin by supposing that we have a b-bit message as input, and that we wish to find its message digest. Here b is an arbitrary nonnegative integer; b may be zero, it need not be a multiple of eight, and it may be arbitrarily large. We imagine the bits of the message written down as follows”

m_0 m_1. . . . . . . .m_{b-1}

The following five steps are performed to compute the message digest of the message.

Step 1. Append Padding Bits

The message is “padded” (extended) so that its length (in bits) is congruent to 448, modulo 512. That is, the message is extended so that it is just 64 bits by shy of being a multiple of 512 bits long. Padding is always performed, even if the length of the message is already congruent to 448, modulo 512.

Padding is performed as follows: a single “1” bit is appended to the message, and then “0” bits are appended so that the length in bits of the padded message becomes congruent to 448, modulo 512. In all, at least one bit and at most 512 bits are appended.

Step 2 Append Length

A 64-bit representation of b (the length of the message before the padding bits were added) is appended to the result of the previous step. In the unlikely event that b is greater than 2^64, then only the low-ordered 64 bits of b are used. (These bits are appended as two 32-bit words and appended low-order word first in accordance with the previous conventions)

At this point the resulting message (after padding with bits and with b) has a length that is an exact multiple of 512 bits. Equivalently, this message has a length that is an exact multiple of 16 (32-bit) words. Let M [0 . . . . n-1] denote the words of the resulting message, where N is a multiple of 16.

Step 3 Initialize MD Buffer[2]

A four-word buffer (A,B,C,D) is used to compute the message digest. Here each of A, B, C, D is a 32-bit register. These registers are initialized to the following values in hexadecimal, low-order bytes first):

Word	A:	01	23	45	67
Word	B:	89	ab	cd	ef

Word	C:	fe	dc	ba	98
Word	D:	76	54	32	10

Step 4 Process Message in 16-word Blocks

We first define four auxiliary functions that each take as input three 32-bit words and produce as output one 32-bit word.

F(X,Y,Z)	=	XY vs not (X) Z
G(X,Y,Z)	=	XZ vs Y not (Z)
H(X,Y,Z)	=	X xor Y xor Z
I(X,Y,Z)	=	Y xor (X vs not (Z))

In each bit position F acts as a conditional: if X then Y else Z. The function F could have been defined using + instead of vs since XY and not (X) Z will never have 1's in the same bit position). It is increasing to note that if the bits of X, Y and Z are independent and unbiased, the each bit of F(X,Y,Z) will be independent and unbiased.

The function G, H, and I are similar to the function F, in that they act in "bitwise parallel" to produce their output from the bits of X,Y, and Z, in such a manner that if the corresponding bits of X, Y, and Z are independent and unbiased, then each bit of G(X,Y,Z), H(X,Y,Z) and I(X,Y,Z) will be independent and unbiased. Note that the function H is the bit-wise "xor" or "parity" function of its inputs. This step uses a 64-element table T [1 . . . . ..64]

Constructed from the sine function. Let T[i] denote the i-th element of the table, which is equal to the integer part of 4294967296 times abs (sin (i)), where i is in radians.

Step 5 Output

The message digest produced as output is A, B, C, D. That is, we begin with the low-order byte of A, and end with the high-order byte of D.

3.2.1.2 Unique Key

This key is shared between node and sink of the virtual grid layout (in designed canvas). This key has 32-bit size length.

3.2.1.3 Cell Key

This key is shared between the nodes (sensors) in the same cell or clusters. This key has 32-bit size length.

3.2.1.4 Authentication Key

This key is used for cell to cell authentication or cluster to cluster authentication. This key has 32-bit of size length.

3.2.2 HMAC Algorithm Description

In cryptography, **HMAC** (Hash-based Message Authentication Code)[12],[13],[16], is a specific construction for calculating a message authentication code (MAC) involving a cryptographic hash function in combination with a secret key. As with any MAC, it may be used to simultaneously verify both the *data integrity* and the *authenticity* of a message.

Any cryptographic hash function, such as MD5 or SHA-1[16], may be used in the calculation of an HMAC; the resulting MAC algorithm is termed HMAC-MD5 or HMAC-SHA1 accordingly. The cryptographic strength of the HMAC depends upon the cryptographic strength of the underlying hash function, the size of its hash output length in bits and on the size and quality of the cryptographic key.

An iterative hash function breaks up a message into blocks of a fixed size and iterates over them with a compression function. For example, MD5 and SHA-1 operate on 512-bit blocks. The size of the output of HMAC is the same as that of the underlying hash function (128 or 160 bits in the case of MD5 or SHA-1, respectively), although it can be truncated if desired.

Let:

- **H**(*m*) be a cryptographic hash function
- *K* be a secret key padded to the right with extra zeros to the block size of the hash function
- *m* be the message to be authenticated
- + denote concatenation
- ⊕denote exclusive or (XOR)
- opad be the outer padding (0x5c5c5c…5c5c, one-block-long hexadecimal constant)
- ipad be the inner padding (0x363636…3636, one-block-long hexadecimal constant)

Then **HMAC**(*K*,*m*) is mathematically defined by

$$\mathbf{HMAC}(K,m) = \mathbf{H}((K \oplus \text{opad}) \oplus \mathbf{H}((K \oplus \text{ipad}) \oplus \text{m})).$$

Implementation

The following pseudocode demonstrates how HMAC may be implemented.

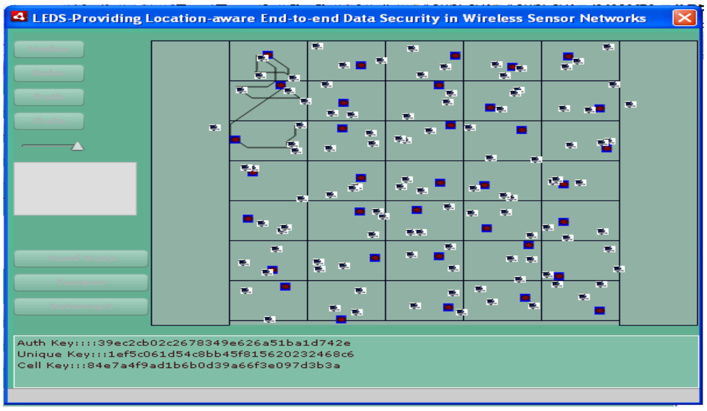
```
function hmac (key, message)
  if (length(key) > blocksize) then
    key = hash(key) // keys longer than blocksize are shortened
  end if
  if (length(key) < blocksize) then
    key = key || zeroes(blocksize - length(key)) // keys shorter than blocksize are zero-padded
  end if
  o_key_pad = [0x5c * blocksize] ⊕ key // Where blocksize is that of the underlying hash function
  i_key_pad = [0x36 * blocksize] ⊕ key // Where ⊕ is exclusive or (XOR)
  return hash(o_key_pad || hash(i_key_pad || message)) // Where || is concatenation
end function
```

3.3 Robot Mechanism

The process of key distribution in wireless sensor network using the robot mechanism. LEDS adopts a robot-assisted network bootstrapping technique. The robot will visit each cell of sensor nodes and distribute the three kinds of keys to each cell. Assume that a group of mobile robots are dispatched to sweep across the whole sensor field along preplanned routes after the deployment of sensors. Mobile robots have GPS capabilities, as well as more powerful computation and communication capacities than ordinary sensors. The leading robot is also equipped with the following bootstrapping parameters:

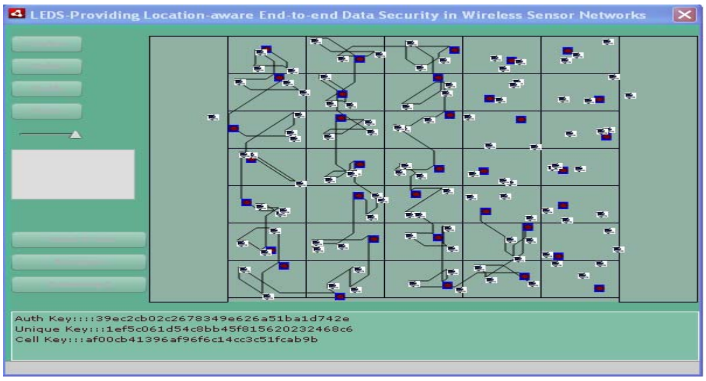
$$\{K^{I}_{M}, K^{H}_{M}, l,(x_0,y_0), (t,T), P\}$$

The robots securely localize every sensor using the secure localization protocol and load each of them with the corresponding location-aware keys in a cell-by cell manner. Specifically, the robots first determine a node u's home cell and then compute a unique secret key $K_{u,}$which u shares with the sink. The robots next compute a set of authentication keys for all the sensors in the same cell. An authentication key is shared among all the sensors in a given cell and its corresponding report-auth cells.


LEDS-Providing Location-aware End-to-end Data Security in Wireless Sensor Networks
Send Data
Tamper
Interrupt

Auth Key::::39ec2cb02c2678349e626a51ba1d742e
Unique Key:::1ef5c061d54c8bb45f815620232468c6
Cell Key:::84e7a4f9ad1b6b0d39a66f3e097d3b3a

Figure 3.3. Window Displaying keys distribution after Robo button has been pressed

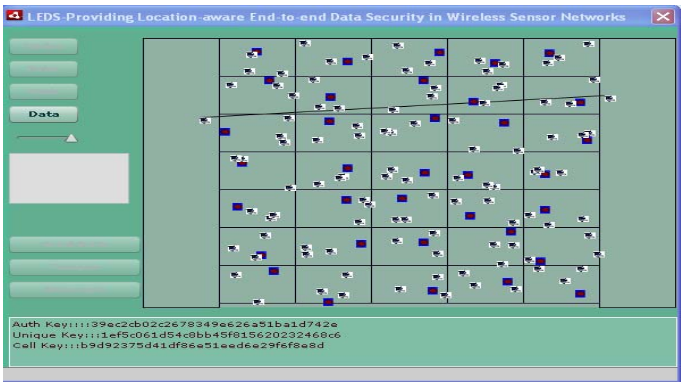

LEDS-Providing Location-aware End-to-end Data Security in Wireless Sensor Networks

Auth Key::::39ec2cb02c2678349e626a51ba1d742e
Unique Key:::1ef5c061d54c8bb45f815620232468c6
Cell Key:::af00cb41396af96f6c14cc3c51fcab9b

Figure 3.4 Application Window Displaying keys distribution after Robo button has been pressed from the first cell to last cell in the Grid.

3.4 Path Selection

This module will show the phase of path selection between source and destination nodes for data sending process. These path selections will be done on the assumption that source and destination know the location of each other. In this mechanism, based on the event of interest sensors, virtual environment can be divided into two partitions and those named as upstream report-auth area[8] and downstream report-auth area[8]. The report-auth cells are determined according to its relative location with respect to the sink. Specifically, a member of the downstream report-auth cells of u is any cell in its downstream report-auth area that is no more than T + 1 cells away from that. Cells in the report-forward route of u are the first of such a cell. The remaining ones for u are those cells within its upstream report-auth area.

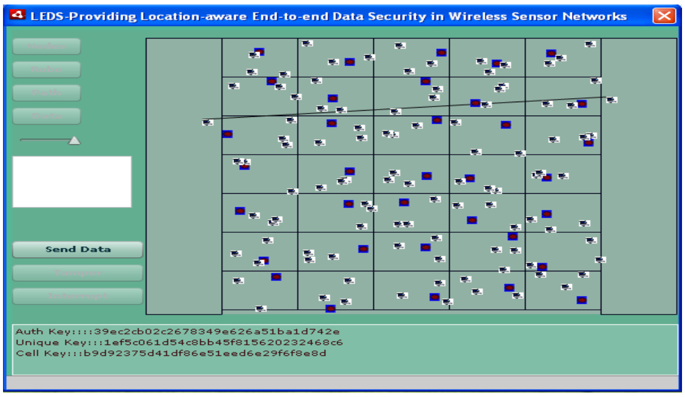

LEDS-Providing Location-aware End-to-end Data Security in Wireless Sensor Networks
Data

Auth Key::::39ec2cb02c2678349e626a51ba1d742e
Unique Key:::1ef5c061d54c8bb45f815620232468c6
Cell Key:::b9d92375d41df86e51eed6e29f6f8e8d

Figure 3.4 Application Window Displaying Data button after the Path has been established between Source and Destination.

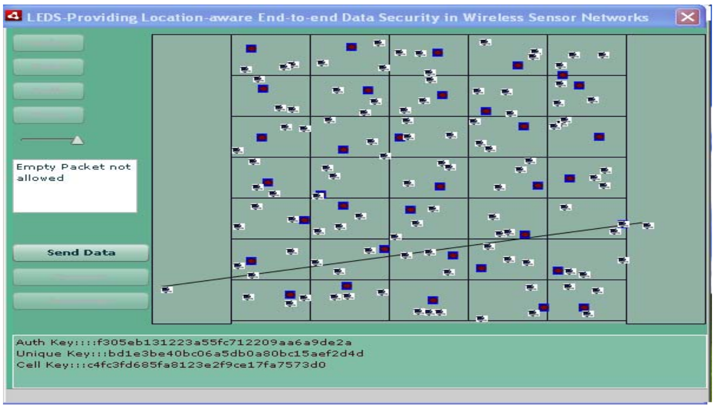
3.5 Data Sending Process

This module is responsible for showing the runtime scenario of data sending between source and destination pair and the behavior of intermediate nodes using three types of keys. Additionally, in this module will show the data encryption done by various keys like cell key, unique key and authentication keys. In these End-to-end data security mechanisms, LEDS seeks to protect data reports in a comprehensive and end-to-end manner.


LEDS-Providing Location-aware End-to-end Data Security in Wireless Sensor Networks
Send Data
Tamper
Interrupt

```
Auth Key::::39ec2cb02c2678349e626a51ba1d742e
Unique Key:::1ef5c061d54c8bb45f81562023246 8c6
Cell Key:::b9d92375d41df86e51eed6e29f6f8e8d
```

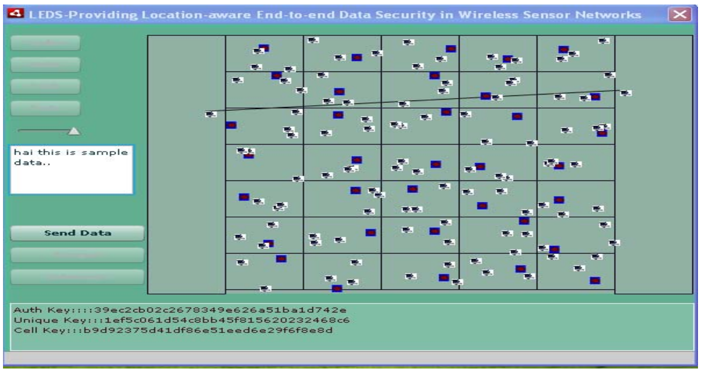
Figure 3.5 Window Displaying Send Data button after the Data button is pressed.


LEDS-Providing Location-aware End-to-end Data Security in Wireless Sensor Networks
Send Data

Empty Packet not allowed

```
Auth Key::::f305eb131223a55fc712209aa6a9de2a
Unique Key:::bd1e3be40bc06a5db0a80bc15aef2d4d
Cell Key:::c4fc3fd685fa8123e2f9ce17fa7573d0
```

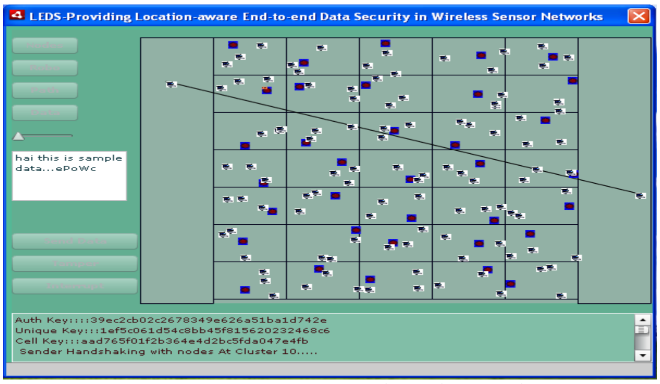
Figure3.5.1 Application Window Displaying that Data should be given in the Text box otherwise it displays a msg as “Empty Packet not Allowed”.


LEDS-Providing Location-aware End-to-end Data Security in Wireless Sensor Networks
Send Data

hai this is sample data..

```
Auth Key::::39ec2cb02c2678349e626a51ba1d742e
Unique Key:::1ef5c061d54c8bb45f81562023246 8c6
Cell Key:::b9d92375d41df86e51eed6e29f6f8e8d
```

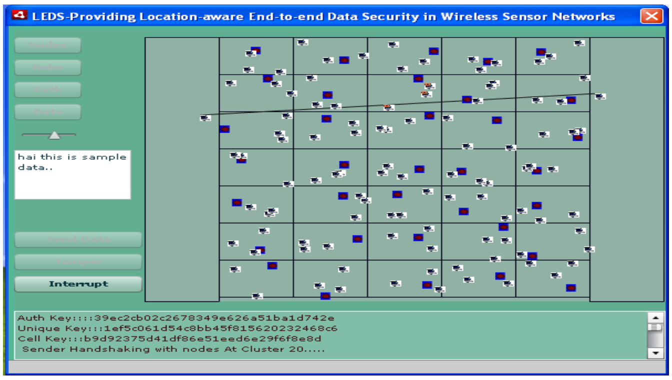
Figure 3.5.2 Window Displaying Send Data button after the data has been entered in the Text box.


LEDS-Providing Location-aware End-to-end Data Security in Wireless Sensor Networks
Nodes
Robo
Path
Data
Send Data
Tamper
Interrupt

hai this is sample data...ePoWc

```
Auth Key::::39ec2cb02c2678349e626a51ba1d742e
Unique Key:::1ef5c061d54c8bb45f815620232468c6
Cell Key:::aad765f01f2b364e4d2bc5fda047e4fb
 Sender Handshaking with nodes At Cluster 10.....
```

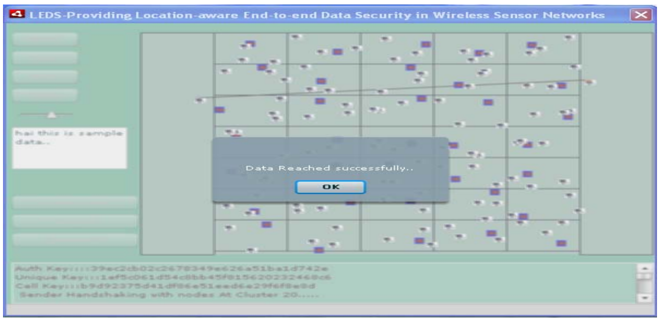
Figure3.5.3 Window Displaying that Data is moving in the form of packets from source to destination path.


LEDS-Providing Location-aware End-to-end Data Security in Wireless Sensor Networks
Nodes
Robo
Path
Data
Send Data
Tamper
Interrupt

hai this is sample data..

```
Auth Key::::39ec2cb02c2678349e626a51ba1d742e
Unique Key:::1ef5c061d54c8bb45f815620232468c6
Cell Key:::b9d92375d41df86e51eed6e29f6f8e8d
 Sender Handshaking with nodes At Cluster 20.....
```

Figure 3.5.6 window Displaying that Interrupt button is being enabled when the Data is being moved from Source To Destination.


LEDS-Providing Location-aware End-to-end Data Security in Wireless Sensor Networks
OK

hai this is sample data..

Data Reached successfully..

```
Auth Key::::39ec2cb02c2678349e626a51ba1d742e
Unique Key:::1ef5c061d54c8bb45f815620232468c6
Cell Key:::b9d92375d41df86e51eed6e29f6f8e8d
 Sender Handshaking with nodes At Cluster 20.....
```

Figure 3.5.7Window Displaying that “Data Reached Sucessfully” When no other button is pressed when data moving from Source To Destination.

3.6 Bogus Data Filtering

This module will show the unnecessary bogus data filtering process. In which cluster head (CH)[13] will verify each data send by node in same cell. And, it looks for the ID of each node of that same cluster or cell. If any one ID is missing then the cluster head (CH) that drop the data of that specified id sensor nodes.

LEDS possesses an efficient en-route false data filtering capability to deal with the infamous bogus data injection attack. As long as there are no more than compromised nodes in each single area of interest, LEDS guarantees that a bogus data report from that cell can be filtered by legitimate intermediate nodes or the sink deterministically. Effective en-route filtering of bogus data packets also results in significant energy savings as unnecessary forwarding is eliminated.

4. Conclusion

Through exploiting the static and location ware nature of WSNs with a location-aware end-to-end security framework to address the vulnerabilities in existing security designs. In this design, the secret keys are bound to geographic locations, and each node stores a few keys based on its own location. This location-aware property successfully limits the impact of compromised nodes only to their vicinity without affecting end-to-end data security.

5. Future Scope

Furthermore, the proposed multifunctional key management framework assures both node-to-sink and node-to-node authentication along report forwarding routes. Moreover, This data delivery approach guarantees efficient en-route bogus data filtering and is highly robust against DoS attacks. And, evaluate this design through extensive analysis, which demonstrates its high resilience against an increasing number of compromised nodes and effectiveness in energy savings.

References

[1] D. Carman, P. Kruus, and B. Matt, “Constraints and Approaches for Distributed Sensor Network Security,” Technical Report 00- 010, NAI Labs, 2000.

[2] A Wood and J. Stankovic, “Denial of Service in Sensor Networks,” Computer, Oct. 2002.

[3] Perrig, R. Szewczyk, V. Wen, D. Culler, and J. Tygar, “SPINS: Security Protocols for Sensor Networks,” Proc. MobiCom, July 2001.

[4] Karlof and D. Wagner, “Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures,” Ad Hoc Networks, vol. 1, no. 2, 2003.

[5] Liu and P. Ning, “Location-Based Pairwise Key Establishments for Relatively Static Sensor Networks,” Proc. ACM Workshop Security of Ad Hoc and Sensor Networks (SASN ’03), Oct. 2003.

[6] Liu and P. Ning, “Establishing Pairwise Keys in Distributed Sensor Networks,” Proc. 10th ACM Conf. Computer and Comm. Security (CCS ’03), Oct. 2003.

[7] Shi and A. Perrig, “Designing Secure Sensor Networks,” Wireless Comm. Magazine, vol. 11, no. 6, Dec. 2004.

[8] H. Chan and A. Perrig, “Security and Privacy in Sensor Networks,” Computer, pp. 103-105, Oct. 2003.

[9] H. Chan, A. Perrig, and D. Song, “Random Key Predistribution Schemes for Sensor Networks,” Proc. IEEE Symp. Research inSecurity and Privacy, 2003.

[10] H. Chan and A. Perrig, “PIKE: Peer Intermediaries for Key Establishment,” Proc. IEEE INFOCOM, Mar. 2005.

[11] L. Eschenauer and V. Gligor, “A Key-Management Scheme for Distributed Sensor Networks,” Proc. Ninth ACM Conf. Computer and Comm. Security (CCS ’02), 2002.

[12] L. Lazos and R. Poovendran, “Serloc: Secure Range-Independent Localization for Wireless Sensor Networks,” Proc. ACM Int’l Conf. Mobile Computing and Networking (WiSe ’04), Oct. 2004.

[13] S. Zhu, S. Xu, S. Setia, and S. Jajodia, “Establishing Pair-Wise Keys for Secure Communication in Ad Hoc Networks: A Probabilistic Approach,” Proc. IEEE Int’l Conf. Network Protocols (ICNP ’03),Nov. 2003.

[14] S. Capkun and J.P. Hubaux, “Secure Positioning in Wireless Networks,” IEEE J. Selected Areas in Comm., Feb. 2006.

[15] S. Capkun and J. Hubaux, “Secure Positioning of Wireless Deviceswith Application to Sensor Networks,” Proc. IEEE INFOCOM,Mar. 2005.

[16] S. Zhu, S. Setia, S. Jajodia, and P. Ning, “An Interleaved Hop-by-Hop Authentication Scheme for Filtering of Injected False Data inSensor Networks,” Proc. IEEE Symp. Security and Privacy, May2004.