Journal of
Computer Science & Systems Biology

Kumar and Singh, J Comput Sci Syst Biol 2015, 8:2
DOI: 10.4172/jcsb.1000176

Research Article Open Access

BMU Routing Algorithm through Smart Role of Intermediate Nodes in WSNs

Sanjay Kumar[1]* and Singh RK[2]

[1]Department of Computer Science, Uttarakhand Technical University, Dehradun, India
[2]Department of Electrical and Computer Engineering, Uttarakhand Technical University, Dehradun, India
[3]Department of Electronics and communication Engineering, Uttarakhand Technical University Dehradun, India
[4]Department of Electrical Engineering, University of West Indies, St. Augustine, Trinidad

Abstract

Unhindered security accomplishment in WSNs has always been threating task. Due to its compelling and tangled nature a lot of research work is going on all around the globe. Out of the various key topics Broadcasting from sink node to other sensor nodes has been the most sophisticated matter which is still untapped. A base station frequently broadcasts messages such as network query, time synchronization, multi-hop routing etc. The basic problem in such communication is source authentication because of untrusted receivers and unreliable communication environment. Therefore a unique topology is in demand. Keeping this theme in mind our proposed paper has come-up with such unique topology and an algorithm which can help sink node not only for broadcasting but also for multicasting and unicasting. Thus helps in improving security to a greater level.

Keywords: Address table; Random number and prime number generator; Describer bits; Dynamic circular-shift function; Intermediate nodes; Parallel searching bit by bit

Introduction

The basic security of any network comes from the way its topology has been designed. A topology can be considered secure if and only if all its components are secured. But in sensor networks such security cannot be incorporated as sensor nodes are to be placed in such adverse areas where reliability cannot be guaranteed. Thus the topology considered under such conditions should be reliable and stable in its random nature. In our proposed topology, we have used a centralized master node called sink node, a few hundreds of Intermediate nodes and a few thousands of sensor nodes. Many proposed paper have used sink nodes and sensor nodes, but in this paper along with them we have introduced a new type of node called Intermediate nodes. These intermediate nodes look alike to sensor nodes in their physical appearance. It means that one cannot identify merely by looking to both types of nodes. For broadcasting purpose, first of all a secure connection is established between Sink node and intermediate nodes in a tree structure format. In latter stage these intermediate node or I_{Nodes} establish connection with sensor nodes and acquire address of all its neighboring sensor nodes.

On time-up sink node sends the actual data to be broadcasted to these Intermediate nodes. Then these Intermediate nodes in turn create copy and send to all those sensor nodes whose address it had stored earlier. In the final stage all Intermediate nodes sends a feed-back to sink. Feed-back results in multicasting or unicasting which can also be done securely and will enhance the security measures of whole network. Whole paper revolves around topology used and routing algorithm implemented. The pictorial representation of broadcasting, multicasting and unicasting will help readers to understand the topology on a broader level.

Related Work

Broadcast authentication is an essential service in WSNs. Symmetric key based message authentication code cannot be directly used for resource-constrained wireless sensor network, since a compromised receiver can easily impersonate the sender. On the other hand, asymmetric key based digital signature schemes which are typically used for broadcast authentication in traditional networks, are too expensive to be used in WSNs, due to high computation involved in signature verification. As a result, several broadcast authentication protocol have been proposed for resource constrained WSNs [1-10]. A broadcast authentication protocol, called BiBa (Bins and Balls) [11-14], have been proposed by Perrig, and it uses one time digital signature scheme to authenticate the source. In BiBa, signer pre-computes some t random values, called SEALs (SELF Authenticating values). For each SEAL s_i, signer generates a public key $f_{si}=F_{si}(0)$, where $F_{si}(0)$ is a one-way function, and these public keys are transferred to the receiver to authenticate SEALs at the receiver end. For each message M, the signer computes GH(M) for all SEALs s_1 to s_t, where GH(M) is a particular instance from a family of one-way function whose range is 0 to n-1 (i.e., n possible output values). The signer generates a signature h_{si}, s_{ji}, where $GH(M)_{(si)}=GH(M)_{(sj)}$ and si6=s_j , and send the message M with the signature h_{si}, s_{ji} to the receiver. After receiving the message, the receiver authenticates the received message by authenticating the signature using previously obtained public keys. The advantage of BiBa is fast verification and a short signature but BiBa takes longer signing time and uses larger public key size to authenticate the signer. To make an improvement over public key size and signing time, Reyzin *et al.* have proposed a new one-time signature scheme called HORS [15-18] (Hash to Obtain Random Subset) which reduces the time needed to sign the message and verify the signature. It also reduces the key and signature sizes in comparison to the ones used in BiBa and make HORS the faster one-time signature scheme. The security

***Corresponding author:** Sanjay Kumar, Department of Computer Science, Uttarakhand Technical University, Dehradun, India, Tel: 9412148830; E-mail: sanjaybenpour@gmail.com

Received September 30, 2015; **Accepted** February 04, 2015; **Published** February 06, 2015

Citation: Kumar S, Singh RK (2015) BMU Routing Algorithm through Smart Role of Intermediate Nodes in WSNs. J Comput Sci Syst Biol 8: 104-111. doi:10.4172/jcsb.1000176

of BiBa depends upon the random-oracle model, while the security of HORS relies on the assumption of the existence of one-way functions. HORS is computation-ally efficient, requiring a single hash evaluation to generate the signature and a few hash evaluations for verification as compared to BiBa. Still this protocol has large public key size, which is not suitable in a WSN environment without additional modifications. Signing each packet would definitely provide secure broadcast authentication, but it still has considerable overhead for signing and verifying packets and also uses more bandwidth. An efficient broadcast authentication scheme, proposed by Shang-Ming, is also based on one-time digital signature scheme. Compared to HORS, this scheme requires less storage and communication overhead at the expense of higher computation cost. In this scheme, key generation is the same as that used in HORS scheme. This scheme makes an improvement over the HORS scheme by reducing the large key size, but still the public key size is large and computational overhead per message is also large. Perrig *et al*. have proposed a broadcast authentication protocol, named TESLA [4], and it is the first proto-col proposed for broadcast authentication in WSNs. This protocol is based on one way hash key chain.

TESLA uses the key chain to emulate public key cryptography with delayed key disclosure. A key is initially chosen, and the remaining keys are generated using one way hash function. The first key of this chain (the last key produced by the hash function) is used to encrypt the first message to be broadcasted by the base station, and this key is distributed to each node of the WSNs apriori. The sender divides the time period for broadcast into multiple intervals and in each interval it uses one key starting with the first key. At the end of each interval, it discloses the next key, which makes it possible to authenticate the messages that were sent encrypted with the previous key. However, the receiving node needs to verify that the next key was not yet disclosed when it received the messages. After receiving a packet, if the receiver can ensure that the packet was sent before the next key was disclosed, the receiver buffers this packet and authenticates it later after receiving the next key. The protocol has certain drawbacks. The protocol requires loose time synchronization between sender and receiver. Individual authentication as well as instantaneous authentication is not available in TESLA. More storage space is required at the receiver side to buffer the packets until the next key is received. Many WSN applications are real time applications. Hence, to minimize the delay in authentication of real time data, the maximum number of additional packets that are received before a packet is authenticated should be small. Nonetheless, there would be some delay before a broadcast packet can be authenticated, and therefore, it is not suitable for real time applications. To increase the scalability of TESLA, Liu and Ning have proposed a multilevel TESLA [15]. The basic idea of this protocol is to predetermine and broadcast the parameters such as the key chain commitments instead of unicast based message transmission used in TESLA. Even though it improves the scalability of TESLA, it still suffers from certain drawbacks like requirement of time synchronization, more buffer storage, etc.

Key Points and Assumptions

Before starting let's have a brief description of various concepts used in this paper.

Definition

Random number generator: This is a special function which is capable of generating random numbers in range from 1 to 100 with no significant link between previously generated number and the one generated at present. It will also check whether generated number is positive or not. If not it regenerates a new number.

Prime number generator: It's Euler's Function which is capable of generating prime numbers with 86 probabilities if the random number feed to it is in the range of 1 to 100. The function is n^2+n+41. Here n is the number generated by Random Number Generator. A procedure attached to the function is triggered if number generated is not a prime. Trigger calls the Random Number Generator again to reproduce a different number.

Dynamic circular shift function: A Dynamic Circular Shift function (DCS) is a function that defines an operation of rearranging the entries in an 8-bit tuple, a special procedure is followed which is explained below, Terms used: LCS << Left Circular Shift, RCS >> Right Circular Shift.

Address table: Every intermediate node has an address table storing address of all its neighboring intermediate nodes. This table can be just updated when sink node adds any new I_{Node} node.

Describer bits: These are the bits which help in identifying the type of packet. Basically two kinds of identification are required for each type of packets. First decision is to check whether the coming packet is for broadcasting or multicasting or unicasting. Second decision is to check the sender of the packet and decide whether to accept the packet or discard it.

Intermediate node's: These are the most important nodes of the whole network. They are costlier then sensor nodes with a lot of in-build security. One thing is clear that no one can get any data from these nodes through physical tampering. There look and feel is alike to sensor nodes thus it's utterly difficult to identify them. They are mainly used to transfer data packets from sink node to sensor nodes in best possible way. Time to time multicasting and unicasting through these I_{Nodes} has improved security.

The procedure is called parallel searching bit by bit (P.S.B.B): This technique runs in two steps, Firstly we commence our search operation by first of all deciding, How many bits actually have to be searched? This job is done by removing all zeros from left-hand side thereby deciding number of bits to be searched. The second step can be accomplished by taking one bit at a time from RHS of left-out part of extracted I_{Node} Address and comparing the same with bits occupied at same position in all I_{Node} Address present in matrix. Let us suppose the address is of four bits but actually addresses are of 8 bits. This concept can be incorporated in with any number of bit addresses. Just for sake of simplicity we have taken 4 bit address. Let's explain the above steps through an example. **Step 1** Extraction of Next 4 bits address, I_{Node} Address extracted is 0010, After Extraction of all regular zeros from LHS:-10 (Two bits) **Step 2** Bits two be compared are 0 at first position 1 at second position. Verification of both bits with I_{Node} Address in matrix is shown diagrammatically (Figure 1).

Encryption decryption technique

Encryption work done at sender node: Let's suppose, we have a priority bit of 1-bit, an address of 14-bits, a function ID of 7-bits and Value out of Function of 19-bits on which circular shift has to be applied dynamically so called Dynamic Circular Shift (DCS). For this we consider values for each. These values are just to explain the concept clearly.

Describer Bits >010

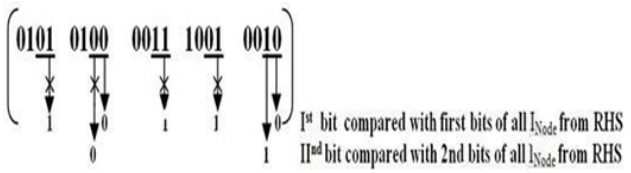

Figure 1: P.S.B.B.

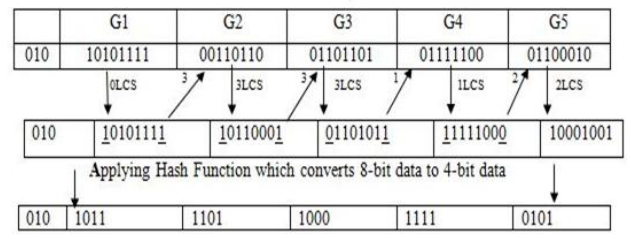

Figure 2: Encryption workdone at Sender Node.

Address >10101110011011001101(Node's address) Value out of Fun > 1010111110001100010 (No of bits can be increased as on demand. 19 bits is just an example which can keep values ranging from 0 to 50000) First of all concatenate them as Address FID-Function Value.

01010101111001101100110110101111110001100010

And then group them in groups of 8-bits as shown leaving the priority bit apart. On G_1 we apply 0 circular shifts. Taking first and last bits of G_1 (10101111) group as shown underlined, (which 11 in binary whose value is 3 in decimal) we come to conclusion that we have to apply 3LCS on G_2 generating 10110001 as result. Similarly, after applying LCS on G_2 we further consider its first and last bit which is again 11 whose decimal value is 3. Thus again we apply 3LCS on G_3 producing 01101011 as the result. This continues till the last group is reached. Above described procedure is explained more vividly through diagrammatic representation. The above 21 bits of data is thus encrypted and ready to be sending to other nodes. We can symbolically represent them as $P_{bit}H[DCS[Concatenate(A_{ddress}F_{ID}Function_{V alue})]]$

Decryption work done at receiver node: Shown below 21 bits will be received by a receiver, on which we will apply Hash function followed Circular Shift function to decrypt the data. Procedure of Circular Shift followed here is same, as used by sender, with a small difference that we apply LCS at sender and RCS at receiver which is a natural phenomenon. The above decrypted data is now grouped accordingly. Ist bit denotes Priority bit, next 14 bits specify the address. Further next 7 bits proposed the function ID. And the left over 19 bits becomes the part value generated through the function. The above decryption can be shown symbolically by P_{bit}DCS[H(Encrypted Bits)] where Encrypted Bits denotes encrypted bits received by receiver (Figures 2 and 3).

Assumptions

1. Time to know random topology created by sink node could not be known till BMU (Broadcasting Multicasting and Unicasting) session is over.

2. None of the nodes know anything regarding its location and its neighboring nodes.

3. Initial deployment should be done safely.

4. Every node either sensor or intermediate should accept or reject the packet on basis of describer bits of the packet.

The Proposed Algorithm

This proposed algorithm is basically efficient enough to securely Broadcast, Multicast and Unicast. In this paper, we have used the basic techniques of encryption-decryption used in our last paper [1]. So the main basic idea is to just create a topology which could be varied from broadcast to broadcast and create such intermediate nodes which would securely transfer packets of data. For such security a routing protocol is a must. The intermediate nodes should also be safeguarded physically (Figure 4). The whole algorithm for broadcasting and multicasting is basically divided in three stages. Unicasting is just a dynamic resilience mechanism which occurs when any sensor node detects any intruder. Broadcasting, multicasting and unicasting are such communications which do not occur occasionally. So it would be more secured if sessions are used and all data used for communication are erased after the session is over. And there should be no link between two sessions. Keeping these all aspects in mind, we first of all deduce a topology which could stand perfect in all scenarios according to our need. The topology is shown below. In the topology shown above it is clear that whole network is sub-divided in small pieces and any kind of communication except the one will be carried on through smart intermediate nodes. Thus a proper routing protocol should be established in these intermediate nodes. So before describing the routing protocol it is necessary to describe the components present in these intermediate nodes. The basic components are

i. A Routing Protocol.

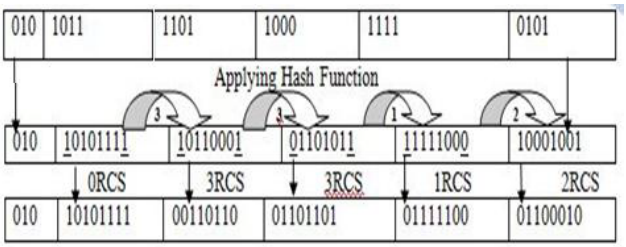

Figure 3: Decryption Work done at Receiver Node.

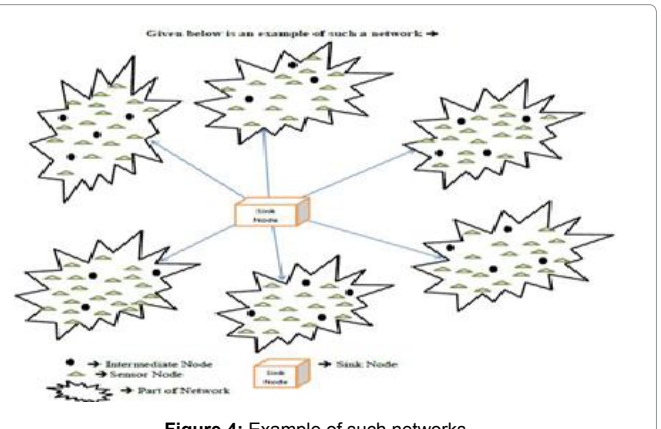

Figure 4: Example of such networks.

ii. Buffer Memory

iii. Common discontinuous function

iv. Dynamic Decrypting Function

v. Volatile Session Address Collector (VSAC)

vi. Address Table holding address of all Intermediate Nodes.

Before working on the arrived packets, it is very necessary to check-out whether they are relevant packets or just a garbage send by an intruder. This can be established by establishing a general rule which each valid node would follow. The rule is explained below. Two bit representation for respective connections.

[00]-Sensor node to Sensor node communication vice-versa (Figure 5).

[01]-Sensor Node to Sink node communication (One way only)

[10]-Sink node to Intermediate communication vice-versa

[11]-Intermediate node to Intermediate node or Intermediate node to Sensor node communication and vice-versa.

In the above shown figure, two bits shown are for secure connection establishment. The above two bits represent that whenever a data packet for connection establishment is send from a sender then only appropriate receiver's will be able to receive them. For example if first two bits are [10], it means data packet send is either form Sink node to Intermediate node or vice-versa. The above explanation just gives us an idea about the packets starting point and its destination point and says nothing about their type whether they are for broadcasting or unicasting or multicasting. For clarity of such a problem we insert one more bit called the Describer Bit [DB]. It is the first bit to be sent for establishing connection. On overall basis we can say that first three bits will describe the data and its purpose thus is DB's. If first bit is zero then the session will be called as broadcast session. And if first bit is one then the session can be multicast or unicast based on the address the packet holds. If the address is of an intermediate node then session is multicast session and if address is of sensor node then it is unicast session. In case of broadcast session whole network just listens to the packets send by sink node via intermediate nodes. In case of multicast just the network part listens to sink node via intermediate node. In unicast no session is created. Just a normal packet of query is send by a sensor node to sink node via intermediate node.

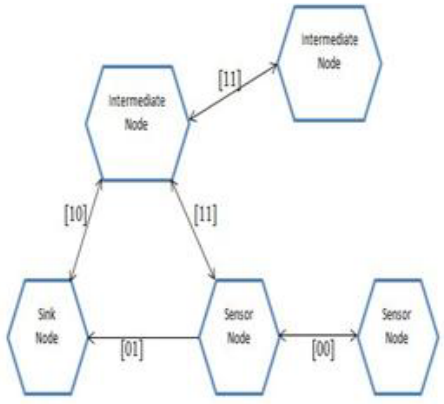

Figure 5: Two bit representation for respective connections.

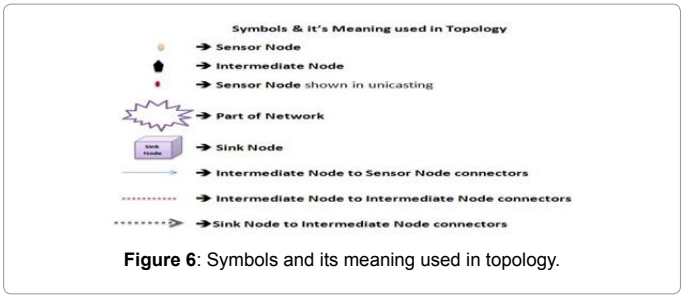

Figure 6: Symbols and its meaning used in topology.

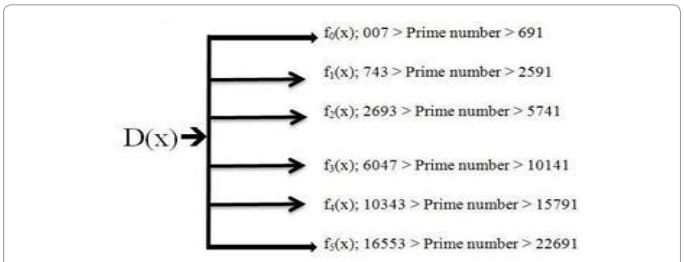

Figure 7: Discontinuous Function.

Proposed Routing Protocol

Though Routing Protocol is a single protocol but works differently in three distinct ways based on the scenario present. Basically it is divided in three sub protocols like broadcasting, multicasting and unicasting Protocol. So a brief example regarding all the three is must. For execution of broadcast and multicast protocol a separate session is created. During this session sensors nodes present in that part of network stops all kinds of communications with its neighboring sensor nodes and intermediate nodes. So let's start with broadcasting protocol algorithm. But before that few basic things should be explained. Like description of discontinuous function, intermediate nodes, generation of FunVALUE, diagram representing symbols and its meaning etc. (Figures 6 and 7). In intermediate node an address table is created holding the address of other intermediate nodes. It also contains a buffer and Volatile Session Address Collector (VSAC). A common discontinuous function is also used as explained in our last paper [1] with little changes. Diagram showing symbols and their meaning used in topology are shown below. The discontinuous function and address matrix of intermediate node is shown below is shown below:-

Generation of FunValue: Procedure is that using random generator function a random number is generated which is feed to prime number generator function to generate a prime number. Then a function is selected from discontinuous function on the range basis i.e. first of all the range is decided in which prime number falls. Then function $f_i(x)$ from that range is selected. From that value Fun VALUE is created. Whole packet of data is encrypted using encryption-decryption technique.

Broadcast Algorithm

Basically progress in three stages i.e. Decision, Propagation and Execution stage

In decision phase: Sink node decides to broadcast some important in-formation. It is this phase where broadcast session begins. For this purpose sink node randomly chooses an intermediate node and calculates FunValue. Intermediate node address is of 16 bit. For FunValue, first of all select a random number say 43 which is feed to

prime number generator (n^2+n+41) to give output as 1993. Sink node then selects a function from discontinuous function where prime number falls in range of 2693-5741. Thus function selected is $f_1(x)$ and let it be as x^2 3985 (Figure 8).

X^2 3985 (Putting the value of X as 1993) 19932-39853968091. Thus after calculation of FunValue sink node creates the DB's and packet to be send to chosen intermediate node. The packet is [DB(010)] H[DSC[$I_{NODE}{}^0$sAddress] [LocFunV alue(3968064)]] which is now ready to be sent.

In propagation phase: Receiving I_{Node} first decides the session by looking up in DB's which a broadcast session is I_{Node} then make appropriate changes in DB's, create multiple copies and send to nearby sensor nodes and I_{Nodes}'s which further distribute the message to their nearby nodes and thus connecting bits are send to all sensor nodes in the entire network. A time-up timer is also set by the sink node within which all sensor nodes have to connect with their neighboring I_{Nodes}. This is shown below with the help of a diagram.

In Execution stage: Sensor nodes firstly checks for the authenticity of data send from I_{Node}. It firstly, checks the describer bits (DB's). If the bits are 011 then sensor nodes understands that a broadcast is going to take place for which it has to connect to its nearby neighboring I_{Node}. It then decrypts the whole message and extracts bits after the address of I_{Node} from where it receives the FunValue and location of function in discontinuous function. The FunValue received is 3968091. The value is then feed to search function to check whether X in that function is a prime number.

X^2=3958+3968091

X^2=3968091+3958

X^2 =3972049

X=$(3972049)^{0:5}$

X=1993 (Value of X is a prime number)

After authentication of bits, sensor node too creates a different FunValue for which a random number say 11. This value is feed to prime number generator function (n^2+n+41) which results as 173. Appropriate function is chosen on the range basis and value 173 is feed to generate a value. The function selected is $f_0(X)$ which is X^2+X+43

X^2+X+43

1732+173+43

29929+173+43

30145

The packet created is $[DB's(011)][I_{Node}\ Address][SensorAddress]$

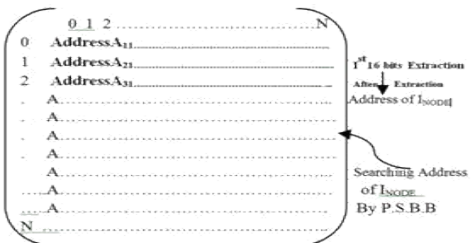

Figure 8: Address Table of I_{Node} holding address of other I_{nodes}.

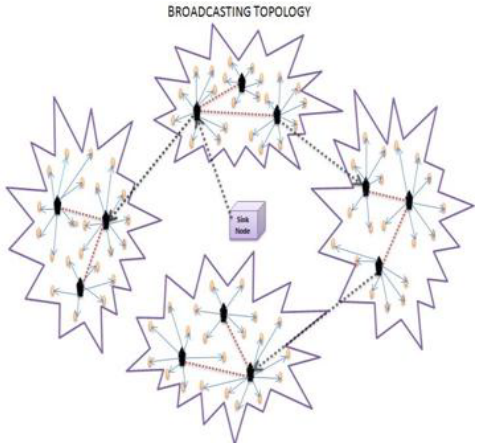

Figure 9: Broadcasting Topology.

$[Loc_{FunValue}(3014)]$

On receiving packet from sensor node, Intermediate node firstly authenticates the data and then stores its address in Volatile Session Address Collector (VSAC) which is explained below. For Authentication:-

$X^2+X+43=30145$

X^2+X 30102=0

X=173 and 174

X= 173 (Value -174 is discarded as it is negative) X=173; (Value of X is prime number)

It stores sensor's address in VSAC. I_{Node} does this for all its neighboring sensor nodes till timer set by sink node expires. After expiry of timer all I_{Nodes} start accepting broadcasting data from sink node and creates multiple copies. These copies are sent to all those sensor nodes whose address is stored in VSAC. On completion of broadcast, I_{Node} report back to sink node by sending all address from VSAC. It then creates a copy of all address and stores it on buffer. All other data used for connection establishment is deleted and session is put to end. Generally broadcast is followed by a multicast. There can be two main reasons behind this purpose. Firstly, sink node would like to scan those sensor nodes who did not respond to broadcast in order to check there authenticity. Secondly, sink node would like to communicate with those sensor nodes who did not respond to broadcast in order to send them the broadcasted message send recently. The occurrence for a multicast is based on the feedback send by I_{Node} (Figures 9 and 10). It stores sensor's address in VSAC. I_{Node} does this for all its neighboring sensor nodes till timer set by sink node expires. After expiry of timer all I_{Nodes} start accepting broadcasting data from sink node and creates multiple copies. These copies are sent to all those sensor nodes whose address is stored in VSAC. On completion of broadcast, I_{Node} report back to sink node by sending all address from VSAC. It then creates a copy of all address and stores it on buffer. All other data used for connection establishment is deleted and session is put to end. Generally broadcast is followed by a multicast. There can be two main reasons behind this purpose. Firstly, sink node would like to scan those sensor nodes who did not respond to broadcast in order to check there authenticity. Secondly, sink node would like to communicate with those sensor nodes who did not respond to broadcast in order to send them the

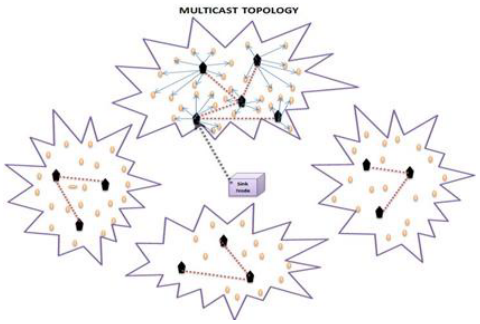

Figure 10: Multicast Topology.

broadcasted message send recently. The occurrence for a multicast is based on the feedback send by I_{Node}.

Multicast Algorithm

Is accomplished in three stages i.e. Decision, Propagation and Execution stage.

Decision Stage: Sink node firstly analyze the feedback and then chooses the most prior part of network to establish multicast session. In this session communication between sensor nodes to sensor node is jammed. The goal of sink node is to get details of those nodes which didn't take part in broadcast. For this purpose sink node randomly chooses an intermediate node from that part of network. It creates DB and an instruction set. Thus creating the package [DB(110)] H[DCS[I_{node}Address][Instruction$_S$et]] and sends to appropriate receiver. Package send is shown in diagram below.

Propagation Stage: I_{Node} accepts the data send from sink node and checks the describer bits. It does so to remove the confusion of broadcasting or multicasting. I_{Node} then creates multiple copies and sends to all I_{Nodes} and sensor nodes present in sub network. Through the message it asks the sensor nodes to send their address and time spend in evaluation of instruction set. Making appropriate changes in DB's, it forwards the packet to all others. The packet send is [DB(111)] H[DCS[I_{node}Address][Instruction$_S$et]].

Execution Stage: I_{Node} keeps on by passing data send by sensor nodes. But before bypassing it does a mandatory check. It transfers only those sensor nodes packets whose address is not present in its immediate buffer storage, thus scanning for unknown address to fulfill the aim of sink node. Other sensors data are put on buffer and deleted as soon as sink node orders to end the session. Appropriate changes in DB's are done before sending. Often after a multicast, a broadcast occurs. But its probability is less. Multicast helps in discovering security breaches and also plays the role of a good friend by helping sensor nodes to get back data not received through broadcasts.

Unicast algorithm: also takes three stages i.e. Decision, Propagation and Execution stage. This algorithm enhances the resilience of each sensor nodes individually. It does not create any overhead like session creation, jamming of sensor node communication etc. The main advantage is that is that it works as a hidden spy and remains inactive. During multicasting, instruction set can request to setup a unicast communication and the desired data. It has no relation with other part of network thus works smartly.

Decision Stage: sensor nodes take the decision whether to establish communication or not? After a positive decision, sensor node creates the packet and sends to the nearby I_{Node}. During this time sensor nodes tops all communications with other sensor nodes and just starts waiting for a unicast reply or a broadcast message. Taking one of the decisions of coming across an unknown address, sensor node can send the packet as shown. The packet contains its own address, unknown sensor address and funValue. For FunValue, first of all select a random number say 23 which is feed to prime number generator (n^2+n+41) to give output as 593. Sink node then selects a function from discontinuous function where prime number falls in range of 007-691. Thus function selected is $f_0(x)$ and let it be as

X^2+X+43

X^2+X+43 (Putting the value of X as 593)

5932+593+43

352285

Thus packet is [DB(111)] H [DCS [Address Own][Address unknown] [FunValue]]. This packet is forwarded to the nearby I_{Node} which instantly forwards to sink node.

Propagation Stage: I_{node} determines that whether communication to be established is unicast or a multicast session. For this I_{Node} observes the DB's and first 16 bit address. If the address is of intermediate node then it is considered as multicast or else if address is of a sensor node then it is considered as unicast. As first address is not found in the address table of intermediate node, it is considered as unicasting. Unicast has highest priority so intermediate node it instantly on receiving. Given below diagram explains the transfer of packet send (Figure 11).

Execution Stage: Sink node first of all checks for the authenticity of the data packet. It first of decrypts the packet and separates the FunValue and location of the function and checks for value of X which is shown below. Function found is X^2+X+43. Thus X^2+X+43=352285 X^2+X 352242=0

X=593or 594 (Value -174 is discarded as it is negative)

X=593 (593, value of X is prime)

After data packet validation sensor node's address and unknown address validation takes place. If un-known address is an authentic one then sink node does a unicast. And sensor node accepts the packet and updates its address table. If unknown address is not present in sink node then quickly a broadcast is planned. Thus occasionally broadcast occurs after a unicast.

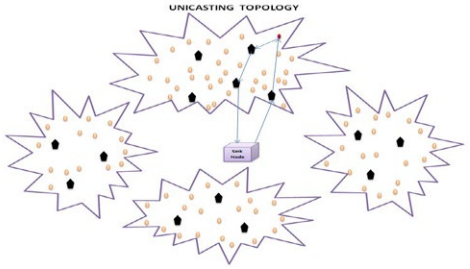

Figure 11: Unicasting Topology.

Network Connectivity Performance Analysis and Overhead Comparison

Analysis of network connectivity

In this section our analysis focuses on how to establish 100% broadcasting. Having a deeper look on the algorithm we can say that combing all the 3 parts i.e. broadcasting, multicasting and unicasting, we can surely establish 100% broadcast. During this process we not only establish 100% broadcast but also able to scan all our sensor nodes. According to the proposed algorithm, we initiate our broadcasting process by establishing a broadcast session. During this session intermediate nodes first of all collects address of all authenticated sensor nodes in VSAC on the request of sink node. It does this job till the timer set by sink node expires. After the expiry of timer intermediate nodes stop accepting data packets from sensor nodes. This means that all those sensor nodes whose addresses are not yet stored will not be able to take part in broadcasting. Latter sink sends the data packet to be broadcasted which is received by all authenticated sensor nodes. The last step of broadcast session is to send a feed-back consisting of address all authenticated sensor nodes. Thus broad-cast session ends. Sending feed-back plays the key role to establish multicast session. As soon as sink node receives feed-back from various intermediate nodes, it goes through and analyses all network parts. It then chooses a network part for multicast purpose on a priority basis. Priority will vary from time to time as the scenario comes. There are two types of multicast. In first case, sink node would like to scan a network to check how many nodes are malfunctioning or captured? Secondly, it chooses a network part and asks intermediate to establish connection with all sensor nodes and resend the broadcasted data. In this way there is 100% chance that all left out sensor nodes which could not get the broadcasted data will receive broadcasted data packets. Practically speaking to achieve 100% connectivity is never possible. Thus if after various multicast few sensor nodes could not receive the broadcasted data packet then unicasting will put forward its hands to bring 100% broadcasting. For this sensor node has to come forward. Suppose a few new sensor nodes were added in the network and a broadcast was done to update their address table with such and such addresses. But few sensor nodes could not update as they were neither able to receive a broadcast nor a multicast. Thus using unicast algorithm they can send a query that they do not know a particular address. Sink node responds by sending the broadcasted message to just that particular node. In this way 100% broadcasting can be done.

Overhead Comparison

Overhead defines the time spend in transmission of data packets. Its analysis helps in solving a lot of security problems. For this purpose we evaluate the overheads at three major points which are base station (Sink node), Intermediate nodes and Sensor nodes and compare it with H2 BSAP scheme. In our proposed scheme we use GAPSOP instead of MAOP which is the procedure followed to transfer data packets securely. GAPS stand for Generalized Authentication Procedure for Security. Overhead of the scheme is greatly reduced due to 2-3 hop coun depth of the network. If topology is studied properly we come to a conclusion that every node (Sink node or I_{Node} or Sensor node) has to send its data packet to 1 hop number depth in or out of the network. Just I_{Node}'s have to use 1or 2 hop-count. The key chaining mechanism has no role in our scheme. Instead of key-chaining we use dynamically generated keys which after authentication are discarded. Thus there is no key-chain storage overhead. Two tables Tables 1 and 2 shown below provide a better explanation for overheads.

Simulation studies

We have studied the performance of our Broadcast authentication scheme using Castalia simulator [17]. All the nodes in the network are randomly deployed. For the simulation purpose, we vary the number of nodes from 50 to 350 in a fixed area of 100×100 meter^2. The network parameters, such as transmission range, transmission rate, sensitivity, transmission power etc., for this simulation study are similar to the parameters specified in CC2420 [18] data sheet and TelosB [19] data sheet. We have taken the initial energy of each node to be 29160 joules for 2 AA batteries as given in the Castalia simulator. Energy consumption for different radio modes used in this simulator is given in Table 3. For this simulation, we assume that clocks of all the nodes are synchronized. The simulation was carried out for both realistic as well as ideal channel. We have used TelosB node hardware platform specification for our simulation and have also used "tunable protocol" provided by Castalia as the MAC layer protocol. The broadcast packet is generated randomly with uniform distribution in every 2 seconds interval at the base station. Figure 12 shows the total number of transmissions to broadcast a packet for different sizes of network. In this figure, we have compared our protocol with the previously proposed scheme with respect to the number of transmissions made. Our approach gives better performance as compared to the previously proposed scheme, because our approach generates a broadcast tree, where only the internal nodes can forward the packet over the network.

This reduces the number of transmissions required to broadcast a packet over the network. Figure 13 shows the number of authenticated nodes and the average number of nodes that received the broadcast packet for different sizes of WSNs. From this figure, we can say that with increase in the density of the network, the percentage age of authenticated nodes also decreases. This happens only due to the collisions of the authentication request packet. It can be reduced by increasing the random delay before the authentication request packet is transferred.

	Overhead at Sink Node	Overhead at I_{NODE} Node	Overhead at Sensor Node
Computation overhead per packet	L_1XGAPS_{OP}		$GAPS_{OP}+(L_3XHASH_{OP})$
Transmission overhead per packet	$L_1XjGAPSj$	$L_2XjGAPSj$	$L_3XjGAPSj$
Verification Delay			

Table 1: Overhead Comparison with GAPS.

	Overhead at Sink Node	Overhead at I_{NODE} Node	Overhead at Sensor Node
Computation overhead per packet	$LXMAC_{OP}$		$MAC_{OP}+[LXHASH_{OP}]$
Transmission overhead per packet	LXjMACj+[LXjKeyj]	(L r)XjMACj+[LXjKeyj]	
Verification Delay			$T_{int}+rXHASH_{OP}$

Table 2: Overhead Comparison with MAC.

Radio mode	Energy Consumption (mW)
Transmit	57.42
Receive	62
Listen	62
Sleep	1.4

Table 3: Radio characteristic.

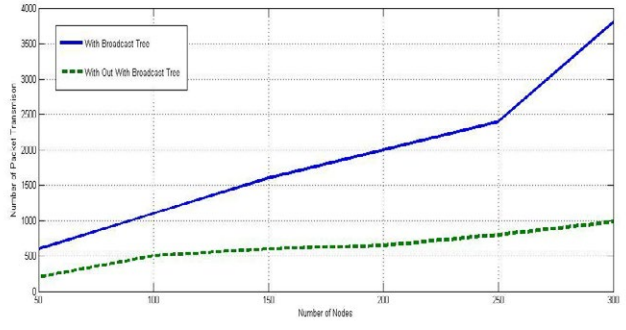

Figure 12: Number of transmission required to broadcast a packet with broadcast tree and without broadcast tree.

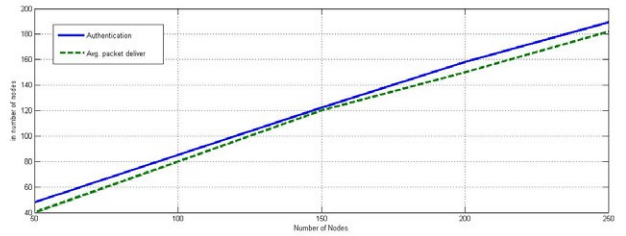

Figure 13: Number of authenticated node and avg. number of node that received the broadcast packet.

Conclusion

The proposed topology and routing algorithm is one the basic approach to broadcast some urgent data packets. In this node to node secure communication has been done securely using the encryption decryption technique proposed. Routing algorithm not only helps complete broadcasting of a message but also in safeguarding network from intruders. Its multicast scanning helps in detecting malfunctioned sensor nodes. Deletion of all calculations used in broadcast or multicast session enhances the security. All three algorithms are made in such a way that each one has no relation with the other. Though there are some overheads but providing such dynamic security and complete broadcast, algorithm can be installed with ease. Future work demands a better timer invocation procedure such that all sensor nodes get authenticated before timer expires.

References

1. Sanjay K, Singh RK (2013) Pair-Wise Key Establishment Using Random Number and Distinct Random Functions in WSNs. IACC100728: Proceedings of the 4th annual international conference on Mobile computing and networking 863-869.
2. Bekara C, Laurent-Maknavicius M, Bekara K (2008) H²bsap: A hop by- hop broadcast source authentication protocol for wsn to mitigate dos attacks. 11th IEEE Singapore International Conference on Communication Systems 1197-1203.
3. Perrig A (2008) The biba one-time signature and broadcast authentication protocol. Proceedings of the 8th ACM conference on Computer and Communications Security. New York, NY, USA: ACM 28-37.
4. Chang SM, Shieh S, WW Lin, Hsieh CM (2006) An efficient broadcast authentication scheme in wireless sensor networks, in ASIACCS06: Proceedings of the 2006 ACM Symposium on Information, computer and communications security. New York, NY, USA: ACM 311-320.
5. Stajano F (2002) Security for Ubiquitous Computing. John Wiley and Sons.
6. Eschenauer L, Gligor VD (2002) A key-management scheme for distributed sensor networks. ACM Conference on Computer and Communications Security 41-47.
7. Gong L (1993) Increasing Availability and Security of an Authentication Service. IEEE Journal on Selected Areas in Communications 11: 657-662.
8. Krawczyk H, Bellare M, Canetti R (1997) HMAC: Keyed-hashing for message authentication. RFC.
9. Blom R (1976) An optimal class of symmetric key generation systems. Proceedings of the EUROCRYPT 84 workshop on Advances in cryptology: theory and application of cryptographic techniques 209: 335-338.
10. Choi SJ, Youn HY (2007) Mkps: A multi-level key pre- distribution scheme for secure wire-less sensor networks. Human-Computer Interaction. Interaction Platforms and Techniques 808-817.
11. Perri (2001) The biba one-time signature and broadcast authentication protocol. Proceedings of the 8th ACM conference on Computer and Communications Security. New York, NY, USA: ACM, 28-37.
12. Zia T, Zomaya A (2006) Security issues in wireless sensor networks. International Conference on Systems and Networks Communications 40-40.
13. Rivest RL, Shamir A, Adleman L (1978) A Method for Obtaining Digital Signatures and Public Key Cryptosystems. Communications of the ACM 21: 120-126.
14. Zhu S, Xu S, Setia S, Jajodia S (2003) Establishing pair-wise keys for secure communication in ad hoc networks: A probabilistic approach. Proceedings of 11th IEEE International Conference on Network Protocols 326-335.
15. Liu D, Ning P (2004) Multilevel μ tesla: Broadcast authentication for distributed sensor networks. ACM Trans Embed Comput Syst 3: 800-836.
16. Reyzin L, Reyzin N (2002) Better than biba: Short one-time signatures with fast signing and verifying. Proceedings of the 7th Australian Conference on Information Security and Privacy 2384: 144-153.
17. Castalia a simulator for wireless sensor networks.
18. 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver. Cc2420 data sheet.
19. Telosb data sheet.