A Novel Approach for Attacks Mitigation in Mobile Ad Hoc Networks Using Cellular Automata

Himadri Nath Saha # [1] , Dr. Debika Bhattacharyya #[2] , Dr. P. K. Banerjee*[3]

Assistant Professor[#1], Professor[#2], Professor*[3]
Department of Computer Science and Engineering, Institute of Engineering &
Management, West Bengal, India[#1, #2]
Department of Electronics and Telecommunication Engineering, Jadavpur University,
West Bengal, India*[3]

Abstract

Many security schemes for mobile ad-hoc network(MANET) have been proposed so far but none of them has been successful in combating the different types of attacks that a mobile ad-hoc network often faces. This paper is providing one way of mitigating attacks in mobile ad-hoc networks by authenticating the node who tries to access this network .This scheme has been applied by using cellular automata (CA). Our simulation results show how cellular automata(CA) is implemented for user authentication and secure transmission in MANET.

Keywords

Ad hoc network; User authentication; Node capturing; Shared key mechanism; Cellular automata.

1. Introduction

Wireless ad-hoc network[2] is a decentralized wireless network which comprises of a large number of sensor nodes. The network is ad-hoc because it does not rely on a pre-existing infrastructure, such as routers in wired networks or access points in managed (infrastructure) wireless networks. Instead, each node participates in routing by forwarding data for other nodes, and so the determination of which nodes forward data is made dynamically based on the network connectivity. Each node has certain computational ability and comprises of a processor, communicational module and a battery supply. These nodes are small, low cost , low power and has functionalities such as communicate over short distances , perform data processing ,sense environmental data, etc.

Wireless ad-hoc network has a wide range of applications. [3,5,15,16] It is used in the military field, in ecological survey, in health related cases such as human physiological data monitoring and many other miscellaneous applications. Most applications where these nodes are used are

DOI : 10.5121/ijasuc.2012.3204

very critical and the data gathered from them are valuable and confidential therefore needs to be protected from outside attacks. There are many possible attacks that one can expect in an wireless environment. A subset of such threats includes Denial of Services (DoS), node capturing, time synchronizing attacks, injecting malicious traffic as well as routing threats. These outside security issues can only be handled by authenticating outside user /nodes. The main aim of authentication is to let sensor nodes themselves detect maliciously injected or spoofed packets. But due to limited resources available in each node it is very challenging to apply user authentication scheme in each node. For this reason we propose to use cellular automata (CA) based components to implement the user authentication scheme in wireless ad-hoc network.

Rest of the paper is organized as follows. We explain attacks in MANET in section 2 and describe the cellular automata section 3,related work in section 4,details of proposed security scheme in section 5,simulation results in section 6 ,analysis in section 7 and finally we present our conclusions in section 8.

2. Attacks in MANET

A. **Denial of Services (DoS).** A DoS attack is any event that diminishes or eliminates a network's capacity to perform its expected function. Hardware failures, software bugs, resource exhaustion, environmental conditions, or any complicated interaction between these factors can cause DoS. IEEE 802.11 wireless networks have fuzzy boundaries, thus allowing an attacker to capture the transmitted packets. The attacker can send large volumes of illegitimate traffic and utilise system resources in a way that makes the system inoperable thus denying access to the authorized users.

B. Eavesdropping. Eavesdropping is a kind of Passive attack. It is similar to injecting some false information into the network.

C. Sinkhole attack. Attacker creates metaphorical sinkhole by advertising for example high quality route to a base station. Almost all traffic is directed to the fake sinkhole. This kind of attack because of the communication pattern: most of the traffic is directed towards sink – single point of failure

D. Node Capture. MANET nodes are usually spread in public areas where an outsider can easily attack them. Due to the low cost of sensor nodes they are not made much damage resistant. This exposes sensor nodes to physical attacks by an advisory.

E. Replay attack. As the medium is wireless the attacker can intercept the message flows easily and replays those to start a new session.

3. CELLULAR AUTOMATA

A dynamic system consisting of a grid of identical finite state machines, whose states are updated synchronously at discrete time steps according to some local update rule, is a cellular automaton. All cells are updated together in an iterative fashion. The process of producing successive generation of grid by updating its cells is called evolution. A d dimensional cellular automata is a structure A=(Zd,φ,N,F) where Zd is a lattice of d-tuples of integer numbers. Each cell in d-dimensional lattice Zd is represented as {z1,z2,…..zd}.

Φ is the set of finite states.

N is a finite subset of Z which is called the neighborhood vector. We assume that there exists a neighborhood function that maps a cell u to the set of neighbors. F: $\Phi m \rightarrow \Phi$ is called the local rule of cellular automata. It computes the new state for each cell from its neighbor cells.In case of general three neighborhood CA present state of any cell $\sigma i(t)$ is dependent on the present state of this cell and also the (i-1)th and (i+1)th cell.

$\sigma i(t+1)=f(\sigma i\text{-}1(t),\sigma i(t),\sigma i+1(t))$,

As the vector $\sigma i(t)$ denotes the configuration of CA at time t. In a two state three neighborhood CA 23 distinct configurations possible and 28 different mappings from all the configurations to the next state. Now each mapping represents a CA rule specified by equivalent decimal number. The top row represents the 8 possible states.

The subsequent rows give the corresponding states of the ith cell at time instant t + 1. Since the output of the first row is the binary equivalent of decimal 30, it is commonly referred to as the CA rule 30. While the subsequent rows give the corresponding states of the ith cell at time instant t + 1. Since the output of the first row is the binary equivalent of decimal 30, it is commonly referred to as the CA rule 30. On minimization, the truth tables for the rules 30 results in the logic function as noted in the right part of the Table 1, where the symbols ⌐,ᵛ, ^ and respectively, denotes for the logical NOT,OR, AND, and XOR operations. For more details about CA the reader can refer to [17, 19]. Our proposed scheme uses a class of cellular automata i.e. second order CA, popularly known as reversible CA (RCA).

A second order CA is an RCA, where the state of a cell at time t + 1 depends not only on its neighborhood at time t, but also on its state at time t-1:

$$\sigma(t + 1) = f(\sigma i(t),\sigma i\text{-}1(t),\sigma i{+}1(t)) \oplus \sigma i(t\ \text{-}1)\ .$$

In fact, all two-states second order rules may be produced in this way. Denoting σ, n and q, respectively, as the configuration at instant t, t -1 and t + 1 the RCA operations can be expressed as

$$\mathrm{RCA}(\xi,\sigma) = \rho \qquad (1)$$

$$\mathrm{RCA}(\rho,\sigma) = \xi \qquad (2)$$

RCA can be implemented quite easily and efficiently. The need to remember the state of cells at t – 1 is not nearly as much of a burden as it may seem, since most conventional CA implementations must use some form of double buffering anyway. Converting a conventional CA implementation into a second order one may be as simple as replacing one assignment operation with an XOR such an RCA with CA rule 30 can be expressed as:

$$\sigma(t + 1) = f(\sigma i\ (t),\sigma i\text{-}1(t),\sigma i{+}1(t)) \oplus \sigma i(t\ \text{-}1)$$

Depending on two initial configurations (σ(0),σ(1)) of second order RCA, the next configuration σ(2) is evolved. Again, the configurations (σ(1),σ(2)) is used to result σ(3) and so on, σ(q + 1) can be obtained from (σ(q -1),σ(q)). Thus by evolving an RCA for q times consecutively, starting with initial configurations (σ(0),σ(1)) would result in σ(q + 1). In reverse, the configuration σ(q - 1) can be obtained from the same RCA loading with initial configurations (σ(q + 1),σ(q)).

Proceeding in similar fashion, evolving the same RCA for q times, the original configuration $\sigma(0)$ can be deduced. The generalized equations for a second order CA evolving for q times RCA is shown below:

$$\sigma(q+1) = RCAq(\sigma(0),\sigma(1)) \qquad (3)$$

$$\sigma(0) = RCAq(\sigma(q+1),\sigma(q)) \qquad (4)$$

The scheme is based on some properties of cellular automata as:

Theorem 1. It is computationally infeasible to guess σ, ξ (two consecutive configurations) from the given output configuration ρ.

Theorem 2. The idea of 3-neighborhood second order RCA can be extended to design a pth order RCA. In pth order RCA every (p + 1)th configuration results from the previous consecutive p configurations. The qth evolution of pth order RCA can be expressed as:

$\sigma(q+p)= RCApq(\sigma(0),\ldots\ldots.\sigma(p-1))$

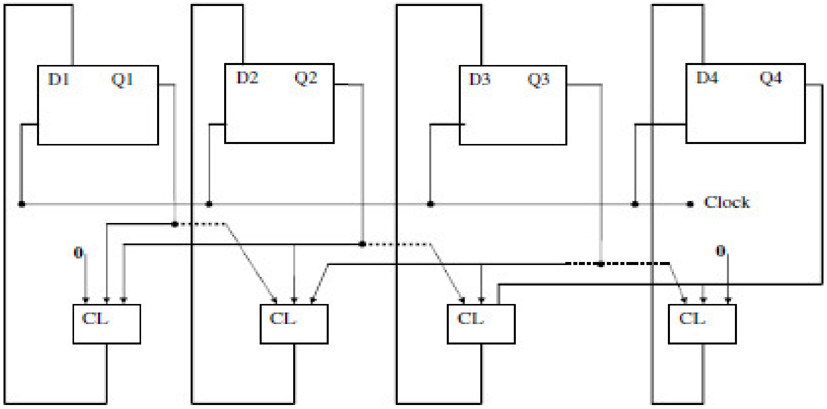


Fig. 1. Evolution of a 3-neighborhood CA.

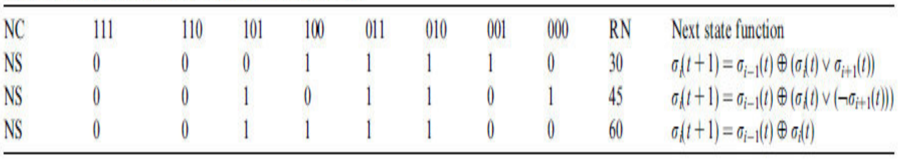
Similarly, one can restore the original configurations from p consecutive configurations by backward evolution as:

$$\sigma(0)= RCApq(\sigma(p+q),\ldots\ldots.\sigma(q))$$

Theorem 3. Let an RCA with non-linear CA rule (rule 30, rule 45) is loaded with two sets of initial configurations (σ,ξ) and (σ',ξ). The following inequality holds.

$$RCA(\sigma,\xi)\oplus RCA(\sigma',\xi) \neq RCA(\sigma,\sigma')$$

Table 1
Next state configuration and logic function of some CA rules



NC	111	110	101	100	011	010	001	000	RN	Next state function
NS	0	0	0	1	1	1	1	0	30	$\sigma_i(t+1) = \sigma_{i-1}(t) \oplus (\sigma_i(t) \vee \sigma_{i+1}(t))$
NS	0	0	1	0	1	1	0	1	45	$\sigma_i(t+1) = \sigma_{i-1}(t) \oplus (\sigma_i(t) \vee (\neg\sigma_{i+1}(t)))$
NS	0	0	1	1	1	1	0	0	60	$\sigma_i(t+1) = \sigma_{i-1}(t) \oplus \sigma_i(t)$

4. Related work

There are many possible attacks which can be expected in a common channel wireless environment. A subset of such threats would include DoS attacks [9], node capturing [18], blackhole attacks [20,21],grayhole attacks[20,21],sybil attacks[22], time synchronization attacks [14], injecting malicious traffic as well as routing threats [12],.Key pre-distribution is an important issue in WSN security. A number of literature is already devoted to secure distribution of keys in WSNs, include [6,8,10,11]. Now a days cellular automata is often used to set a defence mechanism for wireless sensor networks. In a wsn network the nodes are backed up with small memory size, low battery storage and weak processors. There are other ca based schemes which are proposed to develop a wsn security scheme. One of them is CAB which is a cellular automata based key management system that allows sensors to establish pair wise keys during any stage of the network operation using preloaded CAs. It uses simple bitwise OR and XOR. So its computation is very simple. It also has rekeying capabilities and achieves quasi-perfect resilience against node compromise. It considers a large-scale homogeneous sensor network whose nodes are randomly distributed over a region. There is no neighbourhood information available to any sensor before deployment. So a sensor discovers its neighbours and their CA information via local wireless broadcast after deployment.

The broadcast feature of wireless communication allows adversaries to perform a variety of passive and active attacks. In passive listening mode, adversaries silently listen to radio transmissions in order to capture data, security credentials, or other relevant information. For active attacks, adversaries may insert, modify, replay, or delete traffic, or jam part of the network. As a result, adversaries are capable of performing attacks that include session hijacking and man-in-the-middle attacks. Adversaries equipped with powerful communication devices may access any spot of the network from a remote location. However, they cannot monitor the entire deployment region simultaneously at all times. They can gain mobility through the use of robotics or vehicles, and can move inside or outside the network. Also, adversaries can deploy their own sensors and base stations in uncontrolled wireless environments. Further, they are able to capture, replace, compromise, and physically damage existing sensors.

Another scheme that uses CA is LISA or Lightweight Security Algorithm for wsn. This paper is tailored to implement resource restrained sensor node. This scheme can be used to get data authentication and data confidentiality both .

5. Proposed Security Scheme

It is clear that ad-hoc networks are spread over a field and it is possible to capture a node by an adversary. That is why we need to have some authentication before any data communication. To employ the proposed scheme we need to have a base station which will take care of initialization of authentication. As there is no base station in an ad-hoc network thus we need to have some scheme to determine one of the nodes as the base station. After that CA is applied for the authentication purpose. As we have seen this CA mechanism shows high randomness thus it is very difficult to breakdown and also CA involves very little computations like bit wise XOR , AND operations and also storage required is not high. First phase involves choosing of a base station.

Setting up the network

Before setting up a wsn , we must consider some key factor. As the connection is wireless, every node has to broadcast whatever it wants to send. In ad-hoc network there is no base station. But to make things easier we will select one of its nodes as base station .we will discuss the process of selection later on. In an ad-hoc network nodes can be captured or damaged frequently. So we need an efficient algorithm to select base stations when the current base station goes out of control.

A node in an ad-hoc network generally means an electronic device backed up by a battery. So we should not put excess load on a particular node to save battery power.

We will set up our network by following some steps :

The first node wants to communicate becomes current base station. It gets marked with a serial number (for first node it is 0, it’s an unique number) and starts counting it’s age from 0.The base station always stores the serial number of the last node(sln) joined and when another node comes its serial no should be sln+1.The age of each node gets incremented after a specific amount of time that amount of time is constant for the entire network. Base station should keep sending an is-alive packet after a fixed time slot to inform the other nodes that it is alive.

When a node wants to communicate it broadcasts a hello message. The base station receives it and acknowledges it and also sends its serial number. New node gets its new serial no and starts its age counting.

When base station wants to shut down it broadcasts a message to inform it to other nodes. The base station searches for the alive node with lowest serial number and sends a packet to that node to let it know that it is the current base node.

If the base station gets captured or damaged and goes off without notifying other nodes then other nodes stops receiving is alive packets from the base station. At this point base station is selected by broadcasting their own serial numbers.

The base station also checks continually whether any node is showing any kind of malicious activities or not.[1]

Registration Phase

STEP 1: Base Station(BS) chooses a secret key SB. We consider that each node has its own identity IDi and BS distributes an secret key to each node computing

$$Ski=H(IDi \oplus SB) .$$

Whenever a node which was dead earlier has some data to send or receive it needs to register under BS. It sends its identity to the BS. Then BS again computes,

$$Si= H(IDi \oplus SB)$$

and sends it to the node via a secure channel.

STEP 2: Then the node broadcasts its identity to all the nodes. Each node after getting this message generates a Nonce Nik corresponding to IDi and keeps it for a definite time period in its memory. Then it sends Esk(Nik || IDi || IDk), IDk to the BS using the symmetric key cryptography.

STEP 3: BS on receiving the messages from the nodes computes SKk using its own SB . SKk= H(IDk $\oplus$SB) and then decryptes the received message. If the IDk after decryption matches with the received one then BS computes (N1',N2'……Nk') using

$$(N1',N2'……Nk') = RCApq(N1,N2…Nk)$$

Then BS sends these values to the node along with their IDk.

STEP 4: The newly alive node receives those values and computes the nonce values using

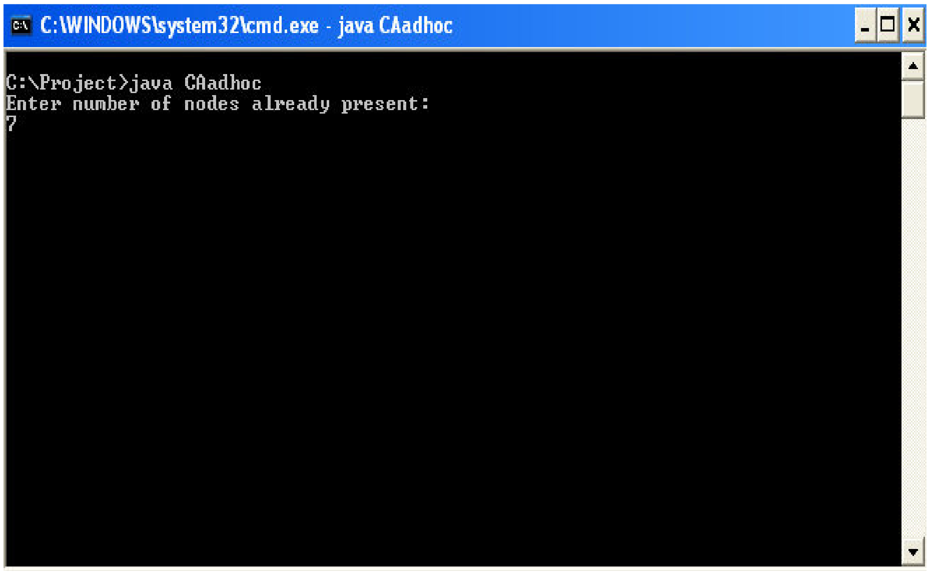
$$(N1,N2……Nk) = RCApq(N1',N2'…Nk')$$

The node then generates a random Nonce N and sends Nki, ID to corresponding sensor with IDk encrypted with the nonce Nik. for mutual authentication.

STEP 5: Each sensor node after receiving decrypts to get the ID and Nki. If the ID matches with the corresponding Nik then node authenticates the new node. If it does not match then node discards the requests from that node and marks the node as the malicious node reports to the BS. These Nonce values are kept for a definite time period after that re-authentication is required.

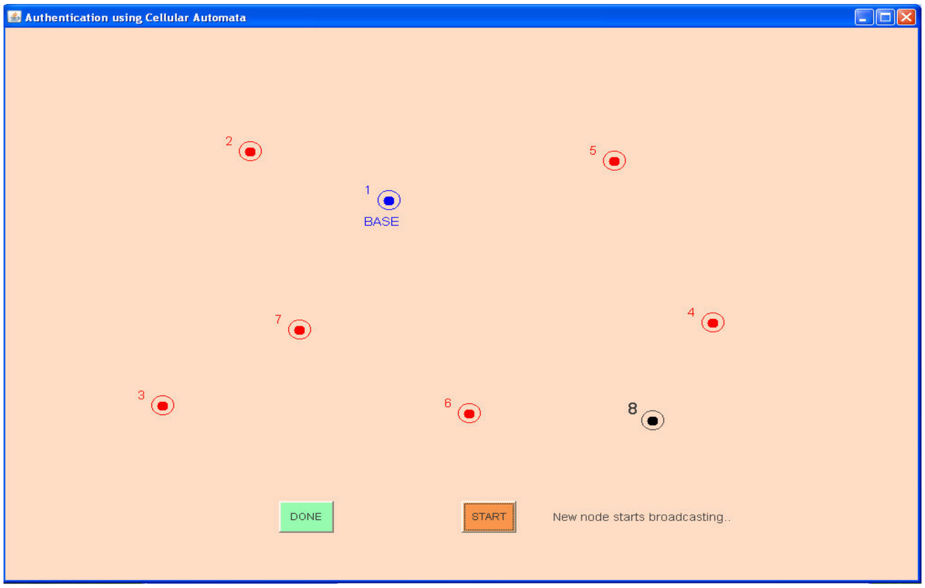
Shared key mechanism. At this point any node in the network has the Nonce of the newly alive node and as this process gets repeated for each node in the network each pair of node is aware of their own unique (Ni,Nj). Now in case of data transmission these nodes use CA rule and q th evolution on the nonce pair to generate a shared key. While transmitting data it encrypts using the shared key and sends own identity along with it . Node after receiving the message gets the nonce corresponding to the ID from memory. And it computes the shared key using same CA rule again this operation is simple and not time consuming but highly random. Any eavesdropper in the middle can get the identity of the sender and if also knows one of the nonce values cannot compute the key or cannot decrypt the message.

6. Simulation Results



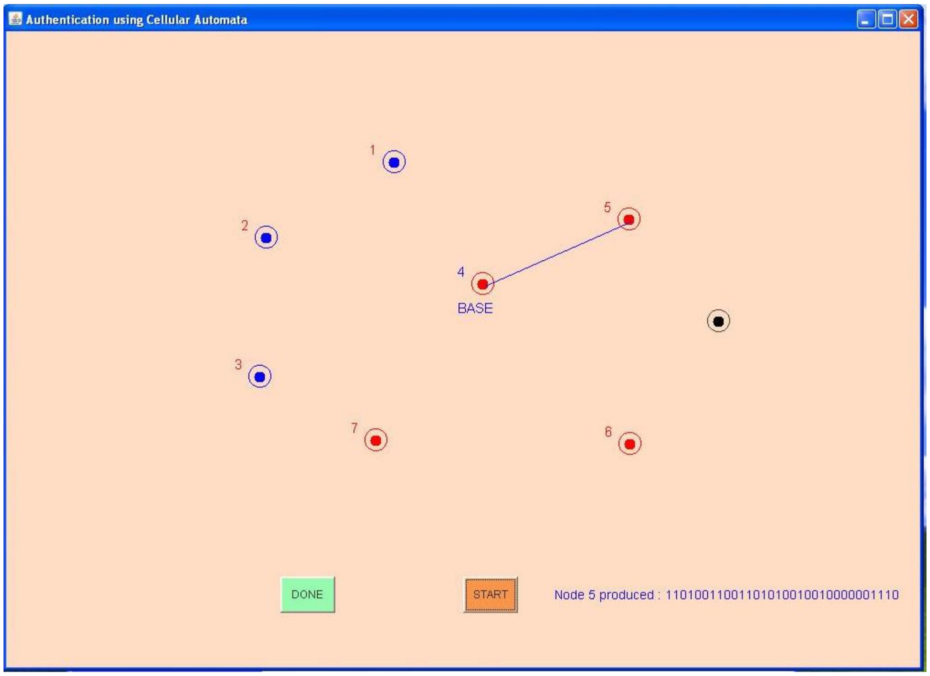
```
C:\Project>java CAadhoc
Enter number of nodes already present:
7
```

Fig. 2. Interface for entering number of nodes which will form the network.



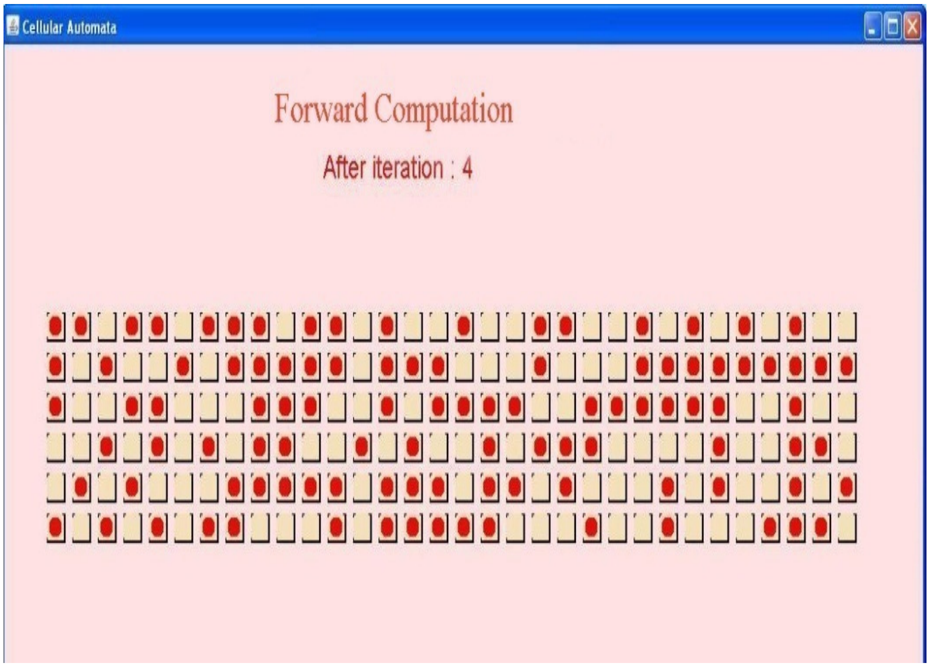
New node starts broadcasting..

Fig 3. New node starts broadcasting.



Node 5 produced : 1101001100110101001001000000001110

Fig 4. Node 5 produced 1101001100110101001001000000001110

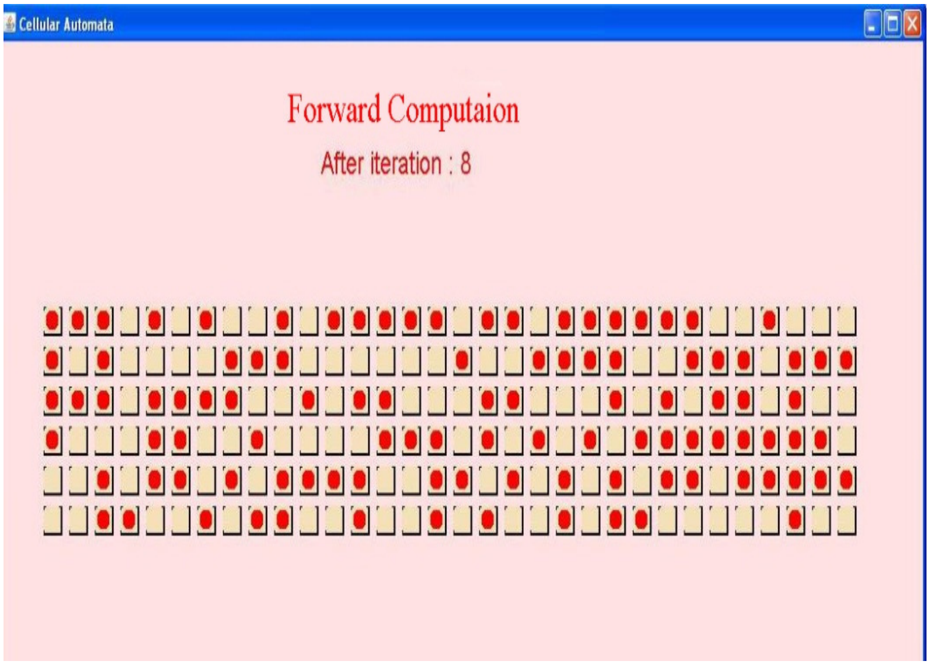


Forward Computation

After iteration : 4

```
●●□●●□●●●□●●□●□□●□□●●□□●□●□●□●□□
●□●□□●□●●●●●□●●●□□□●□□□●●●●●●●●●
●□□●●□□□●●●□□●□●●●●□□●●●●●●□□●□□
□□●□●□●□●●□□●□●□□●□●●●□□□□●□□●●□
□●□●□□□●●●●●□●●●□●●□●□□□●□●□□●□●
●□●□●□●●□□□●□●●●●●□□□●□□●□□□●●●□
```

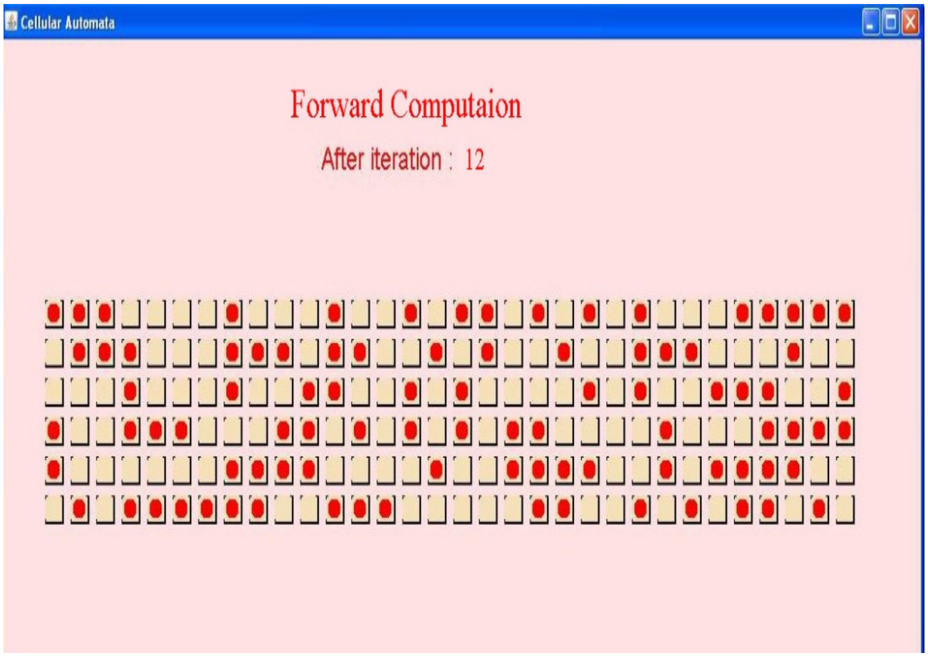
Fig 5. Forward computation after iteration 4.



Forward Computaion

After iteration : 8

Fig. 6. Forward computation after iteration 8.



Forward Computaion

After iteration : 12

Fig. 7. Forward computation after iteration 12.

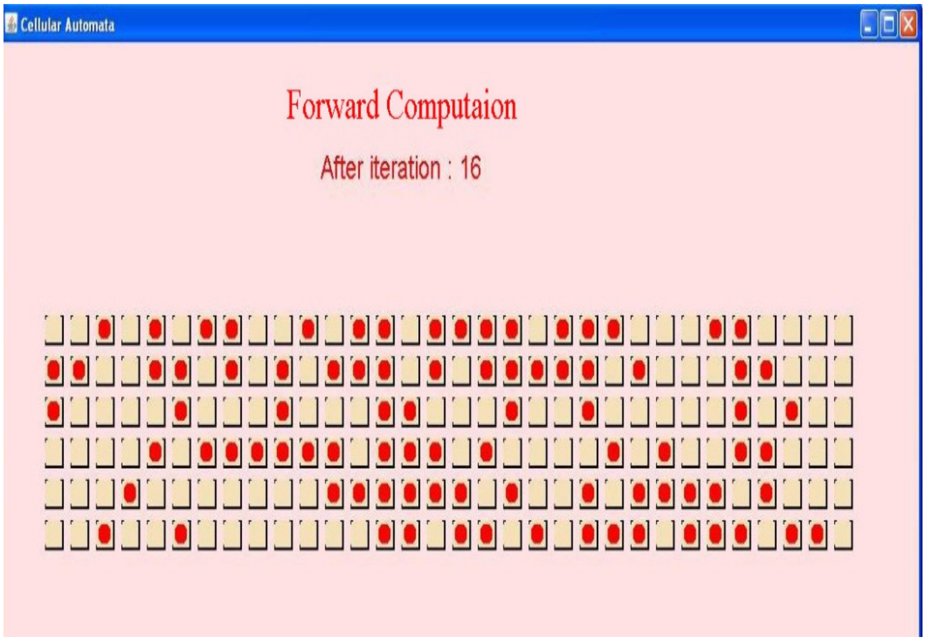


Fig. 8. Forward Computation after iteration 16.

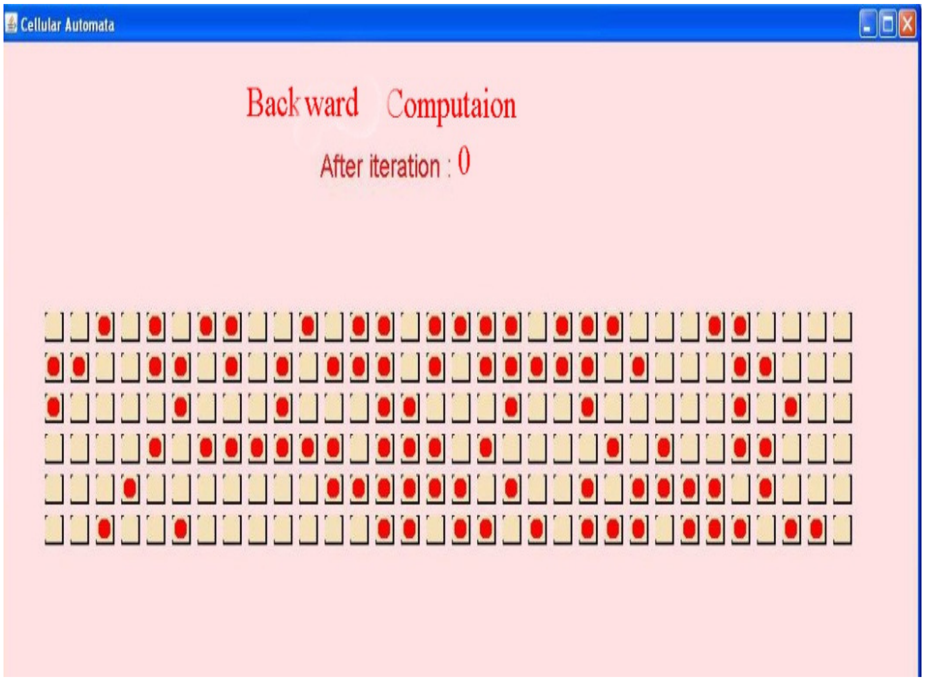
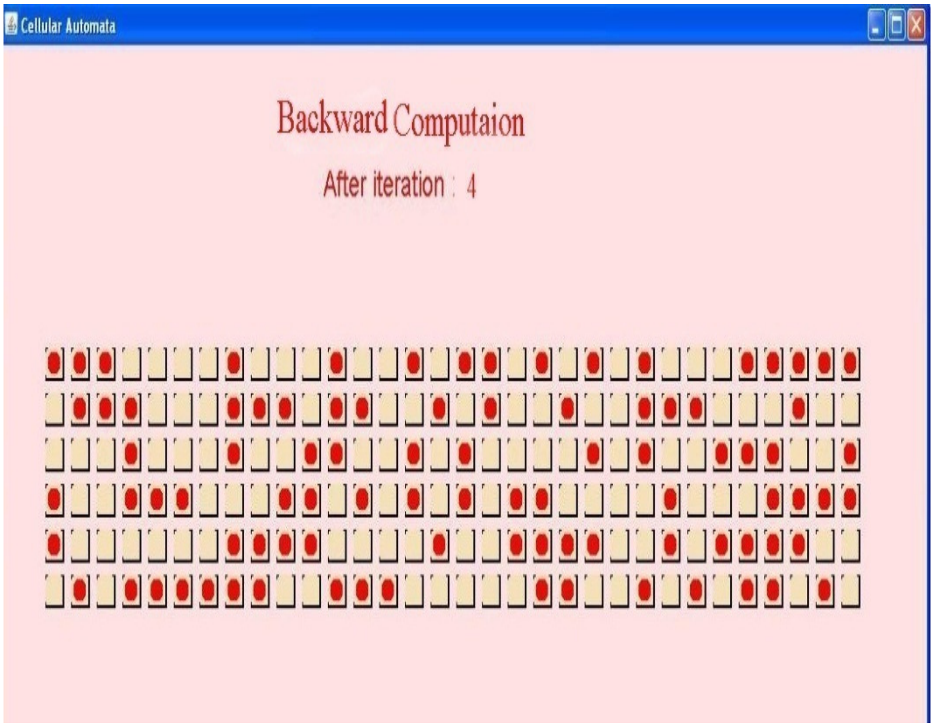


Fig. 9. Backward computation after iteration 0.

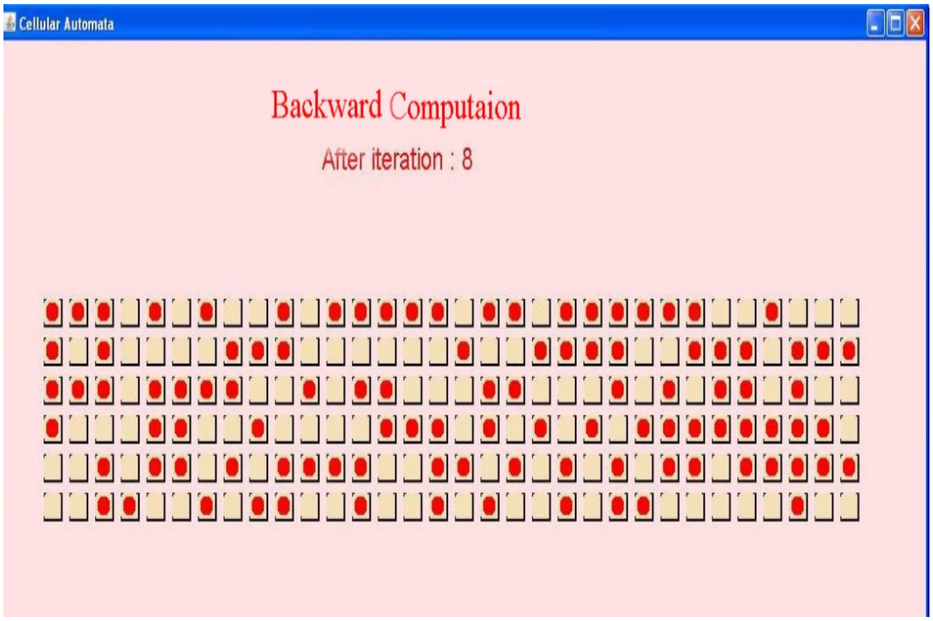


Backward Computaion

After iteration : 4

●	●	●	□	□	□	□	●	□	□	□	●	□	□	●	□	●	●	□	●	□	●	□	●	□	□	□	●	●	●	●	●
□	●	●	●	□	□	□	●	●	●	□	●	●	□	□	●	□	●	□	□	●	□	□	●	●	●	□	□	□	●	□	□
□	□	□	●	□	□	□	●	□	□	●	●	□	□	●	□	●	□	□	□	□	●	□	●	□	□	●	●	●	□	□	●
●	□	□	●	●	●	□	□	□	●	●	□	●	□	●	□	●	□	●	●	□	□	□	□	●	□	□	□	●	●	●	●
●	□	□	□	□	□	□	●	●	●	●	□	□	□	□	●	□	□	●	●	●	●	□	□	●	□	●	●	●	●	□	□
□	●	□	●	●	●	●	●	●	□	□	●	●	●	□	□	□	□	□	●	●	□	□	●	□	●	□	●	●	□	●	□

Fig. 10. Backward computation after iteration 4.

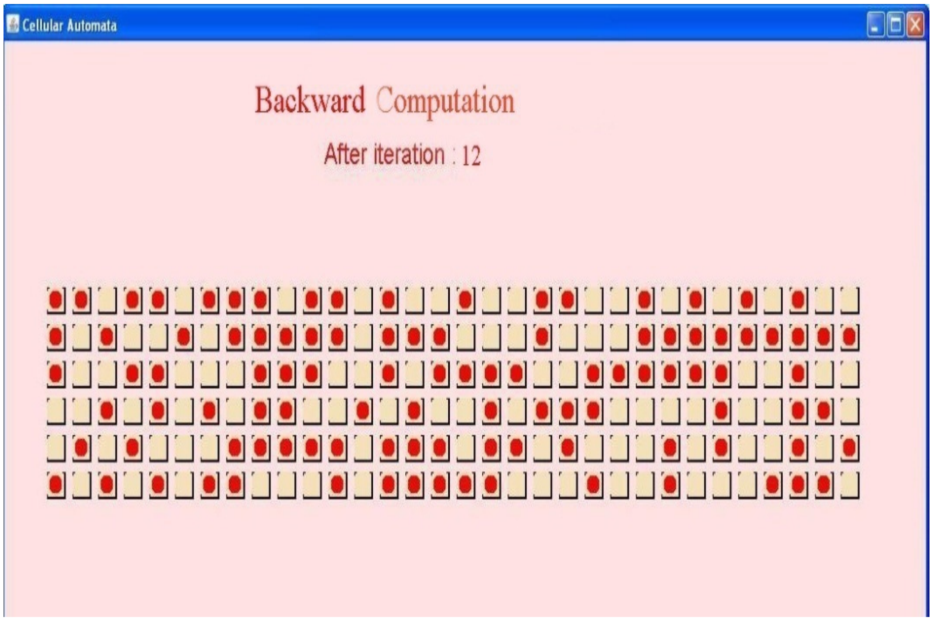


Backward Computaion

After iteration : 8

●	●	●	□	●	□	●	□	□	●	□	●	●	●	●	●	□	●	●	□	●	●	●	●	●	●	□	□	●	□	□	□
●	□	●	□	□	□	□	●	●	●	□	□	□	□	□	□	●	□	□	●	●	●	●	□	□	●	●	●	□	●	●	●
●	●	●	□	●	●	●	●	□	□	●	□	●	●	□	□	□	●	●	□	□	□	●	□	●	□	●	●	□	●	□	□
●	□	□	□	●	●	□	□	●	□	□	□	□	●	●	●	□	●	□	●	□	●	□	●	●	●	●	●	●	●	●	□
□	□	●	□	●	●	□	●	□	●	●	●	●	□	□	●	●	□	●	□	●	□	●	□	●	●	□	●	●	●	●	●
□	□	●	●	□	□	●	□	●	●	□	□	●	□	□	●	□	●	□	□	●	□	●	●	□	□	□	□	□	●	□	□

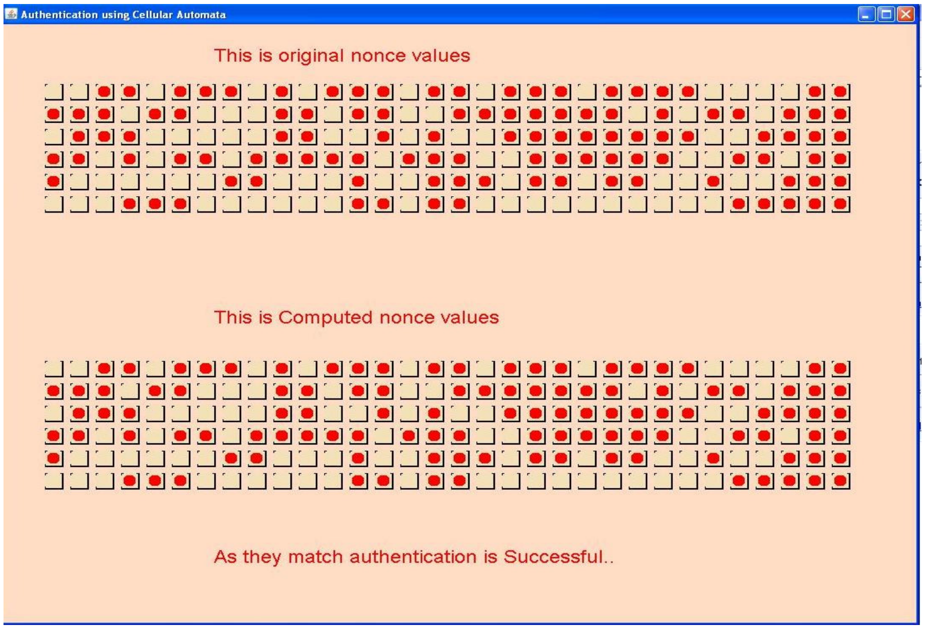
Fig. 11. Backward computation after iteration 8.



Backward Computation

After iteration : 12

Fig. 12. Backward Computation after iteration 12.



This is original nonce values

This is Computed nonce values

As they match authentication is Successful..

Fig. 13. This figure shows the original nounce value and the computed nounce value. As both of them match authentication is successful.

7. Analysis

In this scheme we have a group key authentication initially. Node is authenticated by several nodes. Again CA provides enough randomness thus it is really impossible for attackers to break the keys using dictionary methods, moreover, this is session key implementation so even if attackers able to crack the key, it won't be valid for long enough time. This CA based calculations are very simple and hence less time consuming.

The proposed security component is robust against the following attacks:

Node capturing attack. If an attacker captures a few nodes; the actual information can not be determined. This is because of lack of all correct information from different nodes. In the other way, it is infeasible for an attacker to determine to authenticate itself not knowing nonce values of other nodes.

Denial of service (DoS) attack. DoS is the most generous attack and adversary can disrupt the network services by draining the battery power. It is very difficult to avoid in the environments where, resource constrained devices like sensor nodes are involved. As the computational requirement in our proposed scheme is negligibly small at sensor nodes, attacker cannot make the node busy with computational intensive operations and hence the scheme avoids this form of DoS attack.

Replay attack. The entries in sensor node buffer are valid for a small period of time and therefore, reject the replayed message. On the other hand, the session key established between the sensor nodes is a nonce (number used for once only a standard term in cryptography), so the node also be able to identify the replayed data.

Sink hole or Black hole attack. As there is an strong authentication mechanism thus attacks like sink hole, worm hole are not feasible.

Eavesdropping. All the messages are being encrypted by session keys which are purely random and if nonce values are of 160 bits then It is impossible to break down the system by guessing attack.

8. Conclusion

In this paper we have described what an mobile ad hoc network is. After that we have discussed the different types of attacks that are likely to happen in a wireless ad hoc network. Following that we have introduced the concept of one-dimensional reversible 3-neighbourhood automata for securing wireless ad hoc networks from the previously discussed attacks. The next topic is about the analysis of the network securing schemes. Finally we conclude by saying that work is going on for further improvements in the necessary areas for a better and highly effective protection scheme against outside attacks and remarkable results may be anticipated. This proposed scheme can be further improved by introducing new mathematical concepts.

References

[1] Himadri Nath Saha , Dr. Debika Bhattacharyya , Dr. P. K. Banerjee, A Distributed Administration Based Approach for Intrusion Detection in Mobile Ad Hoc Networks ,IEEE Int. Conference on Science,Technology and Sprituality,Mumbai,2010.

[2] J. Hill, D. Culler, Mica: a wireless platform for deeply embedded networks, IEEE Micro. 22 (2002) 6.

[3] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V.Mittal, H. Cao, M. Demirbus, M. Gouda, Y. Choi, T. Herman, S.Kulkurni, U. Arumugam, M. Nesternko, A. Vora, M. Miyastha, A line in the send: a wireless sensor network for target detection,classification and tracking, Comput. Networks 46 (5) (2004) 605–634.

[4] Z. Benenson, N. Gedicke, O. Raivio, Realizing robust user authentication in sensor networks, in: Proc. Workshop on Real-World Wireless Sensor Networks REALWSN-05, 2005.

[5] R.A.Burne, A.L. Buczak, V.R. Jamalabad, I. Kadar, E.R. Eadan, Selforganizing cooperative sensor network for remote surveillance improved target tracking results, in: Proc. SPIE 4232, 2001, pp. 313–321.

[6] A. Chadha, Y. Liu, S.K. v, Group key distribution via local collaboration in wireless sensor networks, in: Proc. IEEE SECON-05, 2005, pp. 46–54.

[7] A.R. Chowdhury, S. Tripathy, S. Nandi, Securing wireless sensor networks against spurious injections, in: Proc. IEEE Int. Conference on Communication System software and Middleware OMSWARE07,2007.

[8] F. Delghosa, F. Fekri, Key pre-distribution on wireless sensor networks using multivariate polynomials, in: Proc. IEEE SECON 2005, 2005, pp. 118–129.

[9] J. Deng, R. Han, S. Mishra, Defending against path-based DoS attacks in wireless sensor networks, in: Proc. ACM Workshop on Security of Ad Hoc and Sensor Networks SASN-2005, 2005, pp. 89–96.

[10] W. Du, J. Deng, Y.S. Han, P.K. Varshney, A pairwise key predistribution scheme for wireless sensor networks, in: Proc. ACM Conference Computer Communication and Security, (CCS'03), 2003, pp. 42–51.

[11] T. Ito, H. Ohta, N. Matsuda, T. Yoneda, A key pre-distribution scheme for secure sensor networks using probability density function of node, in: Proc. ACM Workshop on security on Ad Hoc and Sensor Networks (SASN), 2005.

[12] C. Karlof, D. Wagner, Secure routing in wireless sensor networks: attacks and countermeasures, in: Proc. IEEE Intl. Workshop on Sensor Network Protocols and Applications (SNPA'03), 2003, pp. 113–127.

[13] M. Luk, A. Perrig, B. Whillock, Seven cardinal properties of sensor network broadcast authentication, in: Proc. ACM Conference on Security of Ad Hoc and Sensor Networks, (SASN 06), 2006, pp. 147–156.

[14] M. Manzo, T. Roosta, Time synchronization attacks in sensor networks, in: Proc. ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN 2005), 2005, pp. 107–116.

[15] K. Martinez, J.K. Hart, R. Ong, Environmental sensor networks, IEEE Comput. 37 (8) (2004) 50–56.

[16] K. Martinez, R. Ong, J. Hart, Glacsweb: a sensor network for hostile environments, in Proc. IEEE SECON 2004, 2004, pp. 81–87.

[17] P. Pal Chaudhuri, D. R. Chowdhury, S. Nandi, S. Chatterjee, 1997. Additive Cellular Automata Theory and Applications 1. IEEE Computer Society Press.

[18] A. Perrig, J. Stankovic, D. Wagner, Security in wireless sensor networks, Commun.+ ACM 47 (6) (2004) 53–57.

[19] S. Wolfram, A New Kind of Sciences, Wolfram media Inc., 2002.

[20] Himadri Nath Saha, Dr Debika Bhattacharyya, Dr. P. K. Banerjee,"A Priority Based Protocol for Mitigating Different Attacks in MANET", International Journal for Computer Science and Communication,Volume I,Number2,pp-299-302,Sept.2010

[21] Himadri Nath Saha, Dr Debika Bhattacharyya, Dr. P. K. Banerjee,"A Distributed Administration Based Approach for Detecting and Preventing Attacks in MANET",International Journal for Scientific and Engineering Reasearch,Volume-2,Issue-3,pp-1-11,Mar-2011

[22] Himadri Nath Saha, Dr Debika Bhattacharyya, Dr. P. K. Banerjee,"Semi-Centralized Multi-Authenticated RSSI Based `Solution to Sybil Attack",International Journal of Computer Science and Emerging Technologies",Volume I,Issue-4,pp-338-341,Dec 2010