RESEARCH ARTICLE OPEN ACCESS

Improving Secure Routing in Geographic Routing Protocols for Wireless Multimedia Sensor Network

Ghaida Esber [1], Mothanna Alkubeily [2]
Master Student [1], Assistant Professor [2]
Faculty of Mechanical and Electrical Engineering
Tishreen University
Lattakia - Syria

ABSTRACT

The emergence of wireless multimedia sensor networks (WMSNs) is an evolutionary step for wireless sensor networks as audio and visual sensors are integrated into wireless sensor nodes. WMSNs involve a lot of requirements such as energy consumption, QoS requirements, high bandwidth demand in addition to achieve security requirement, so the protocols designed for wireless multimedia sensor network should be aware of the resource constraints nature of WSN and multimedia transmission requirement . In our paper we have discussed the essential geographic routing protocols which have been used in WMSN. After that we have simulated one of those protocols and built GUI to simulate sensor nods by using MATLAB and how they exchange secure routing information, then we have compared between some proposed algorithms in secure routing field. In our study, we takes into account some essential parameters in these networks such as delay and secure routing level. At the end, we have determined the ideal algorithm that satisfies required conditions and present the appropriate proposals to eliminate delay according to the results.

Keywords:-Wireless Multimedia Sensor Network (WMSN), Geographic Routing, Secure Routing (SecuTPGF) protocol, Processing Unit.

I. INTRODUCTION

Wireless multimedia sensor networks (WMSNs) are a newly developed type of sensor network which [1] has the sensor nodes equipped with cameras, microphones, and other sensors producing multimedia data content [2-3].

WMSN enhances existing WSN applications and enables a new large range of applications, like multimedia surveillance, traffic management, automated assistance, environmental monitoring, and industrial process control [4].

WMSN requirements impose more challenge and resource constraints to the routing protocol design, involve energy consumption [5], QoS requirements, multimedia coding techniques, high bandwidth demand in addition to achieve security requirement in WMSN [6-7], these mentioned challenges open many research issues and future research directions to develop protocols to maximize the network lifetime while satisfying the quality of service requirements of the various applications.

Various routing protocols are proposed for WMSN such as geographic routing protocols that are the most convenient protocols for these networks. They use position information for making packet forwarding decisions and they do not need to exchange and maintain routing information [8], each node is aware of its geographic positing and uses packet's destination address as a geographic position to perform routing and forwarding decision [9], some of these protocols have focused on the basic challenges such as resource constraints that involve energy consumption, QoS requirements, and others have focused on secure routing issues which form an important point especially when we use these networks in environmental monitoring and military applications. The rest of this paper is organized as follow. To make this paper self readable we expose in section II the essential geographic routing protocols such as GPSR, GEAMS,TPGF and SecuTPGF.

In section III, we present configuration and general guidelines to simulate TPGF protocol, and build a Guide User Interfaces GUI to simulate sensor nods and study how they exchange secure routing information.

In section IV, the simulation and performance evaluation will be presented. Section V will conclude this paper.

II. RELATED WORKS

A. GPSR

The GPSR (Greedy Perimeter Stateless Routing) [10] was originally designed for MANETs but rapidly adapted for WSNs. The GPSR algorithm relies on the correspondence between the geographic location of nodes and the connectivity within the network by using the location position of nodes to forward a packet. Given the geographic coordinates of the destination node, the GPSR algorithm forwards a packet to destination using only one single hop location information. It assumes that each node knows its geographic location and geographic information about its direct neighbours.

B. GEAMS

GEAMS is a new geographical routing protocol namely (Geographic Energy-Aware Multipath Stream-based) [11] that routes information based on GPSR functionalities (Greedy Forwarding and Perimeter Forwarding) while maintaining local-knowledge for delivering this information on multipath basis.

The GEAMS routing protocol can be seen as an enhancement of the GPSR protocol to support the transmission of video streams over wireless sensor networks. The main idea is to add a load-balancing feature to GPSR in order to increase the lifetime of the network and to reduce the queue size of the most used nodes. In fact, routing with GPSR will always choose the same path (i.e. using the same node which is closer to destination). This will rapidly cause the dying (dropping) of the most used nodes. In GEAMS routing protocol, data streams will be routed by different nodes, decisions are made at each hop avoiding the algorithm to maintain a global knowledge of the topology.

C. TPGF

TPGF (Two Phase geographical Greedy Forwarding) [12] routing protocol is the first to introduce multipath concept in wireless multimedia sensor networks (WMSNs) field.

This algorithm focuses in exploring and establishing the maximum number of disjoint paths to the destination in terms of minimization of the path length, the end-to-end transmission delay and the energy consumption of the nodes. The first phase of the algorithm explores the possible paths to the destination.

A path to a destination is investigated by labeling neighbours nodes until the base station. During this phase, a step back and mark is used to bypass voids and loops until successfully a sensor node finds a next-hop node which has a routing path to the base station [13].

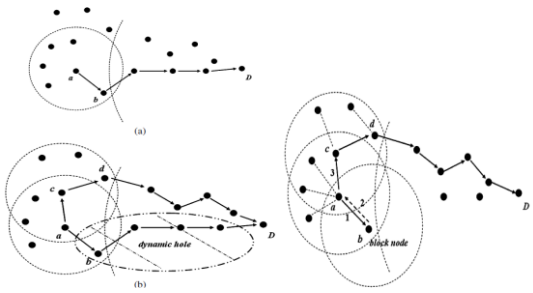

Fig 1: Block situation in TPGF protocol

The second phase is responsible for optimizing the discovered routing paths with the shortest transmission distance (i.e. choosing a path with least number of hops to reach the destination). The TPGF algorithm can be executed repeatedly to look for multiple node disjoint-paths.

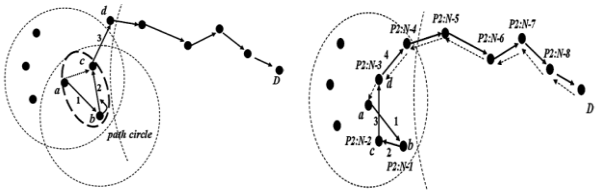

Fig 2: Path optimization in TPGF protocol

D. SecuTPGF

When sensor networks are deployed in a hostile environment, security becomes extremely important due to lack of tamper-resistant infrastructure and the insecure nature of wireless communication channels, Thus, when TPGF is used in WMSNs for transmitting multimedia streaming data, it should be devised in a way that it is resilient to security attacks, since attacks at the networking layer (specifically those against the routing protocols) can disrupt the whole network operation, so the SecuTPGF [14] is a modified version of TPGF that applies Identity-Based Non-Interactive Key Distribution Scheme (IDNIKDS) [15], which provides both node authentication and symmetric key establishment [16].

In SecuTPGF, we mainly secure the neighbour discovery and route discovery. Securing neighbour discovery prevents malicious nodes from joining the WSN and hence nodes establish a neighbour table free of malicious nodes.

Securing route discovery authenticates the intermediate nodes involved in the routing path. SecuTPGF protocol provides the following functions:1) Prevent outside adversaries from joining the network; 2) Limit the impact of insider attack in a localized area; 3) Partially detect insider attack and avoided from the network; 4) Authenticate control messages exchanged between nodes.

E. Discussion

We can summarize the basic parameters which have been taken into account in geographic routing protocol such as energy consumption, delay, secure routing and hole bypassing in the following table:

Table 1 The essential parameters which have been focused in geographic routing protocols.

Geographic Routing Protocol	GEAMS	TPGF	SecuTPGF
Simulator	OMNET++	NETTOPO	NETTOPO
Energy consumption	++	+	+
Delay	+	++	++
Hole Bypassing	-	++	++
Secure Routing	-	-	++

III. CONFIGURATION AND GENERAL GUIDELINES

Through our study to geographic routing protocol, we have found that they take into account the essential requirements from energy consumption and QoS requirement to secure routing, so we can classify our work under these points:

A. Simulation TPGF Protocol through NetTopo

We aim to evaluate TPGF protocol and show the routing phases [17] in it by using NetTopo which is released as an open source sensor network simulator on the Source-Forge [18-19]. Currently, it has been implemented with more than 80 java classes and more than 11,000 Java lines source codes ,we will execute that through these mentioned proposals in this table :

Sensor network	Seed		17
	Node Number		399
	Deploy Network		**500*500**
Hole	Hole1	Radiu of Hole	70
		LocationX	250
		LocationY	500
	Hole2	Radiu of Hole	70
		LocationX	250
		LocationY	400
	Hole 3	Radiu of Hole	70
		LocationX	250
		LocationY	300
	Hole 4	Radiu of Hole	70
		LocationX	250
		LocationY	200
	Hole 5	Radiu of Hole	70
		LocationX	250
		LocationY	140
Transmission Radiu			60
location of sink node			Corner location
Number of source node			1
Sensor node energy			36
Expected lifetime of sensor network			2
Maximum physically allowed transmission radius MTR			60
Agorithm for simulation			Minimize Delay
TPGF routing path topology	NOP (for non-optimal routing path)		
	OP (for optimal routing path)		
Whole network connection topology			

Table 2 General proposals through the simulation of TPGF protocol

B. Simulation of Two Nodes from the Routing Path at the Level of Microprocessors of These Nodes

- We will design GUI to simulate two nodes A,B from one of the routing paths in the network at the microprocessor level of each node using Matlab [20], we supposes that we study the network after applying a routing protocol on it such as TPGF.
- We will focus in our study on information related to the secure routing algorithms which is used to achieve secure routing in wireless multimedia sensor networks.
- We will study two secure routing algorithm, the first one is the secure routing algorithm that have applied in SecuTPGF, and the second one is (Authentication scheme between two individual sensor nodes) algorithm, after that, we will apply these algorithms on our GUI after converting it to assembly instructions.
- Finally, we will compare between them in secure level and delay where delay is the most important factor in QoS parameters in WMSN especially when we used it in military applications and real time application.

B.1 The First Secure Routing Algorithm That We Will Apply on Our Designed GUI : Authentication Scheme Between Two Individual Sensor Nodes[18]

When a sensor node wants to communicate with another sensor node, they need to be authenticated with each other, first, it is assumed that no key information will be stored in individual sensor nodes; instead, every key information will be stored at a high-end sensor node i.e. the master node. By protecting only that master node, we can make the entire sensor network secure. This Figure explain the steps of this secure routing algorithm:

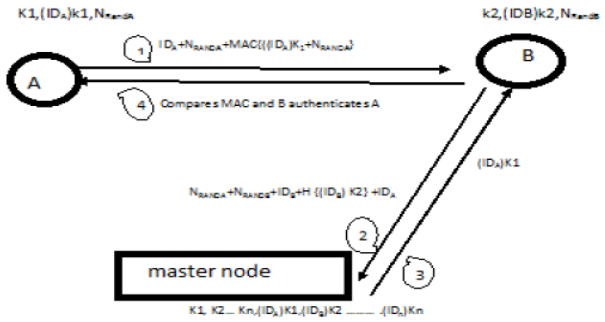

Fig 3: Authentication procedure

B.2 The Second Secure Routing Algorithm That We Will Apply on Our Designed GUI

We can explain the steps of secure routing algorithm which is used in SecuTPGF in this Figure:

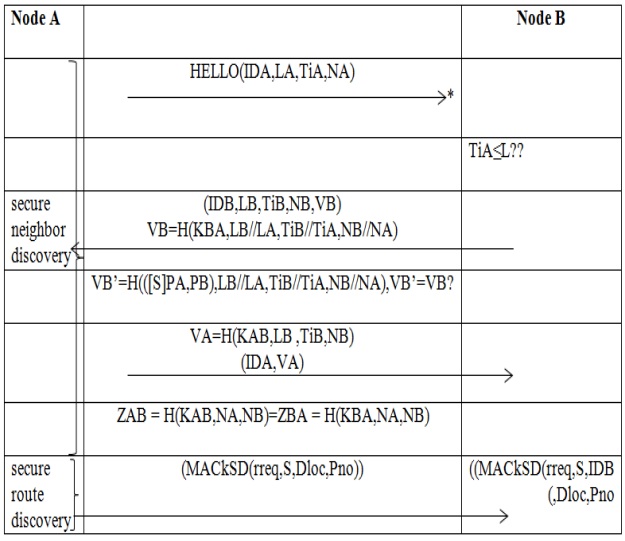

Fig. 4 Secure routing algorithm which is used in SecuTPGF

C. Using of MD5 Hash Function and Elliptic Curve Points During the Simulation

C.1 MD5 Hash Function

During our simulation, we will apply MD5 function as a hash function through the calculation of Message Authentication Code MAC as shown in the figure below.

```
//Process the message in successive 512-bit chunks:
for each 512-bit chunk of padded message
    break chunk into sixteen 32-bit words M[j], 0 ≤ j ≤ 15
//Initialize hash value for this chunk:
    var int A := a0
    var int B := b0
    var int C := c0
    var int D := d0
//Main loop:
    for i from 0 to 63
        var int F, g
        if 0 ≤ i ≤ 15 then
            F := (B and C) or ((not B) and D)
            g := i
        else if 16 ≤ i ≤ 31
            F := (D and B) or ((not D) and C)
            g := (5×i + 1) mod 16
        else if 32 ≤ i ≤ 47
            F := B xor C xor D
            g := (3×i + 5) mod 16
        else if 48 ≤ i ≤ 63
            F := C xor (B or (not D))
            g := (7×i) mod 16
//Be wary of the below definitions of a,b,c,d
        F := F + A + K[i] + M[g]
        A := D
        D := C
        C := B
        B := B + leftrotate(F, s[i])
    end for
//Add this chunk's hash to result so far:
    a0 := a0 + A
    b0 := b0 + B
    c0 := c0 + C
    d0 := d0 + D
end for
```

Fig. 5 Steps of MD5 hash function

We will define a set of values during calculation of MD5 loop as shown in this table:

Table 3: Parameter of MD5 hash function loop

	Values of Node A
S data ki	55146;42104;59591;46934;9248;28891;49597;52974;62844;4015;18811;50730;43056;17939;64838;38145;27008;39128;35652;63407;65535;23473;35164;55230;27536;4386;64920;29075;42617;17294;18868;2081;63006;9570;49261;45888;9822;23121;59830;51114;54831;4189;580;5203;55457;59009;59347;64456;8673;52710;49975;2006;62677;3463;17754;5357;93491;59653;64751;41976;26479;729;36138;19594;65530;14658;34673;63105;28061;24866;64997;14348;24174;59972;19422;53161;63163;19296;48831;48240;10395;32454;60065;10234;54511;12421;1160;7429;55764;53305;59099;39397;8098;31992;50348;22117;62505;8772;17194;65431;43924;9127;64659;41017;25947;22979;32620;52370;65519;62589;34180;24017;28584;32335;65068;59104;41729;17172;19976;45153;63315;32386;48442;62005;10967;53947;60294;54161;
Si	7;12;17;22;7;12;17;22;7;12;17;22;7;12;17;21;5;9;14;20;5;9;14;20;5;9;14;20;5;9;14;20;4;11;16;23;4;11;16;23;4;11;16;23;4;11;16;23;6;10;15;21;6;10;15;21;6;10;15;21;6;10;15;21.
K1	129.
IDA	235.
Initial MAC values	8961; 26437; 43913; 61389; 56574; 39089; 21622; 4146.
Nonce A	31; 301; 251; 77; 513.

C.2 Generation of Elliptic Curve Points

We can use the mathematical equations shown in the figure below to generate elliptic curve points that we use to great keys in SecuTPGF

If E is given by $E : y^2 = x^3 + bx + c \pmod p$ we define

$$(x_3, y_3) = (x_1, y_1) + (x_2, y_2)$$

as

$$x_3 = s^2 - x_1 - x_2 \pmod p \quad \text{and}$$
$$y_3 = s(x_1 - x_3) - y_1 \pmod p$$

where

$$s = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} \pmod p, & \text{if } P \neq Q \\ \frac{3x_1 + b}{2y_1} \pmod p, & \text{if } P = Q \end{cases}$$

Fig. 6 Point addition relationship in elliptic curve

IV. Simulation Results

A. Simulation of TPGF Protocol

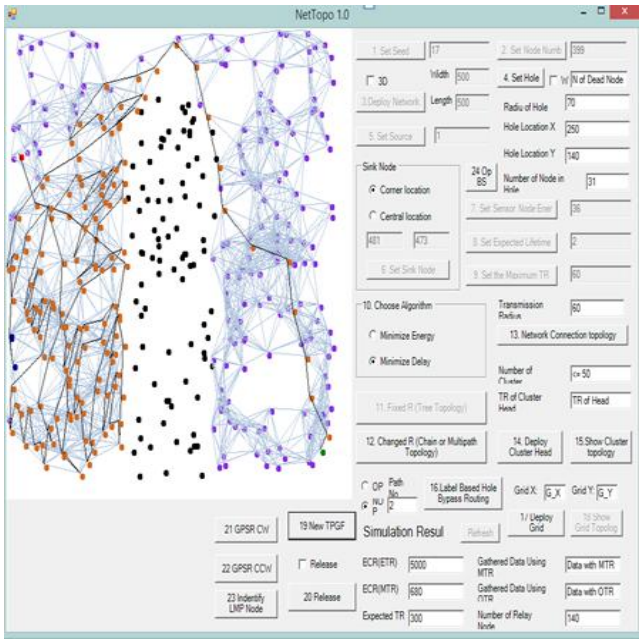

Fig. 7 Explains hole bypassing operation in TPGF protocol.

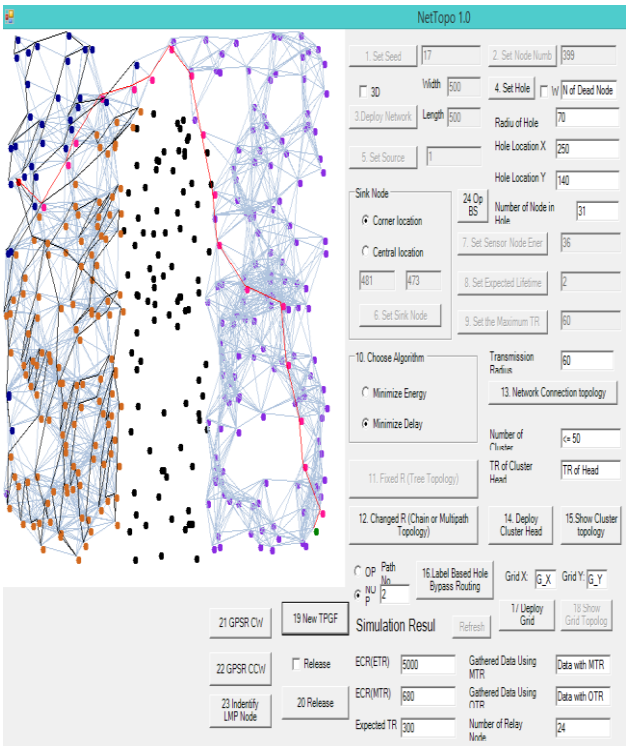

Fig. 8 Path optimization phase

Hence we can notice the mechanism of path circle elimination, and then how it limits the distance inside the route path.

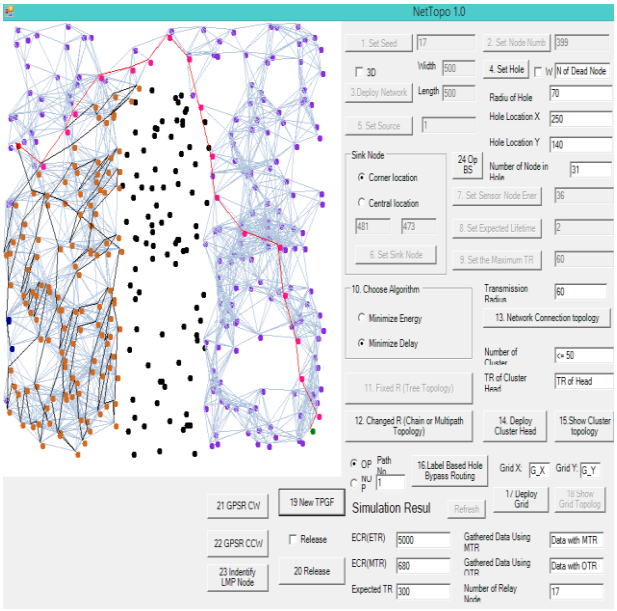

Fig. 9 TPGF protocol for another path

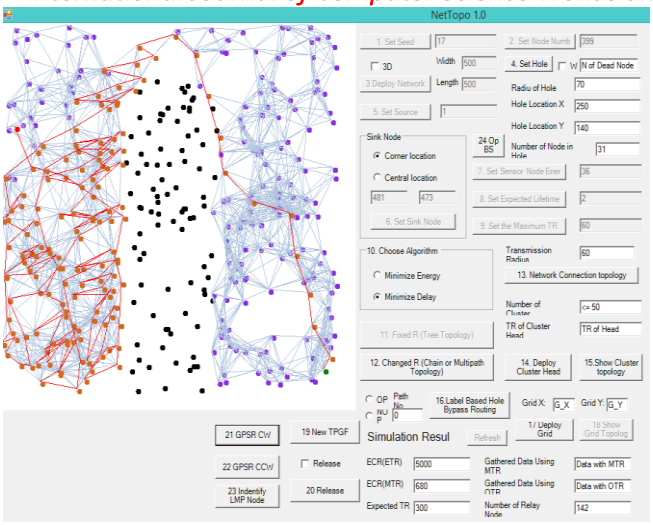

Fig. 10 GPSR protocol

GPSR is the essential protocol in geographic routing where other protocols have improved depending on it.
We can notice the great improvement in the building of the routing paths inside TPGF protocol in compare with this protocol.

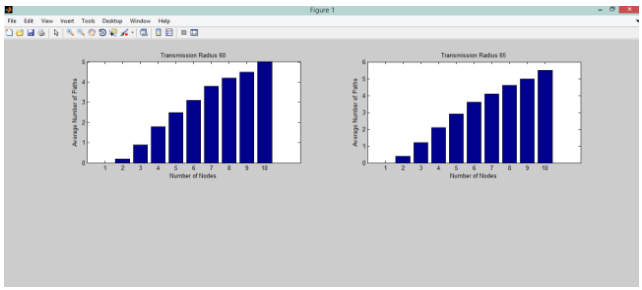

Fig. 11 Relationship Between Average Path Number And the Number of Nodes for Different transmission radius Values.

We can notice the increase of average path number in TPGF with the increase of node numbers in the network for different values of transmission radius

B. Simulation of Two Nodes from the Routing Path at the Level of Microprocessors of These Nodes

The GUI that we have designed by using Matlab enables us to compare between secure routing algorithms in the studied network after converting it to assembly instruction sets, we can compare between them in consumed delay of each algorithm in addition to secure level.
Figure 12 explains the designed GUI by using pointers .

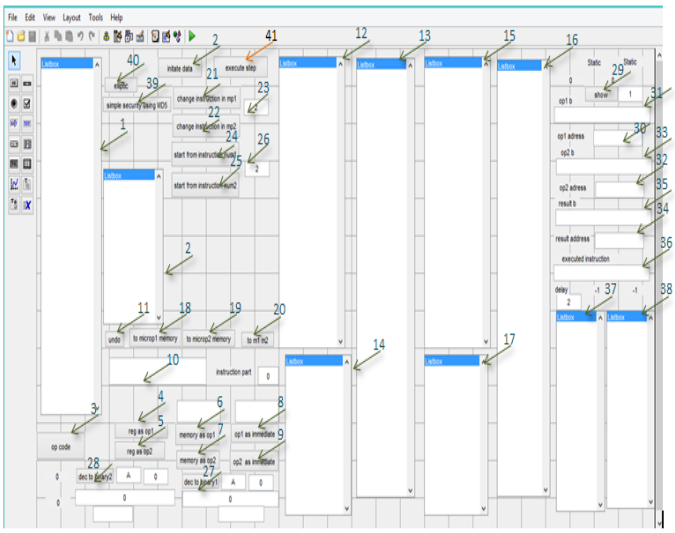

Fig. 12. The designed Graphical User Interface GUI.

We can explain the Graphical User Interface GUI work by the following table which illustrates components that have been mentioned by pointer numbers in GUI

Table 4: Explain the GUI Work.

Pointer number	Name/Type of the Element	
1	**(listbox1)**	List of instruction that we use during converting secure routing algorithms to assembly instruction
2	**(listbox2)**	List of registers that we use during the execution of the instruction
3	**Op code (pushbutton10)**	Ability to choose the instruction that we want to use.
4	**reg as op1 (pushbutton2)**	The first operand that we want to use is a register
5	**reg as op2 (pushbutton4)**	The second operand that we want to use is a register
6	**memory as op1 (pushbutton3)**	The first operand that we want to use is a memory location.
7	**memory as op2 (pushbutton6)**	The second operand that we want to use is a memory location.
8	**op1 as immediate (pushbutton13)**	The first operand that we want to use is an immediate value.
9	**op2 as immediate (pushbutton5)**	The second operand that we want to use is an immediate value.
10	**(edit3)**	Here we determine the operands of the instruction

		in its final format before sending it to the program memory related to the nodes.
11	**Undo (pushbutton11)**	Undo the written instruction if we decide to cancel it and rewrite another one.
12	**(listbox3)**	Program memory of the first node
13	**(listbox6)**	Memory locations of the first node
14	**(listbox5)**	Register values of the first node
15	**(listbox4)**	Program memory of the second node
16	**(listbox8)**	Memory locations of the second node
17	**(listbox7)**	Register values of the second node
18	**to microp1 memory (pushbutton7)**	To send the instruction to the program memory of the first node.
19	**to microp2 memory (pushbutton8)**	To send the instruction to the program memory of the second node.
20	**to m1 m2 (pushbutton22)**	To send the instruction to the program memory of the first and second node at the same time.
21	**change instruction in mp1 (pushbutton16)**	Ability to change the instruction inside the program memory of the first node.
22	**change instruction in mp2 (pushbutton17)**	Ability to change the instruction inside the program memory of the second node.
23	**(edit5)**	The number of the instruction that we want to change in the program memory of the first and second nodes
24	**start from instruction num1 (pushbutton18)**	The ability to begin the execution from a determined number in the program memory of the first node.
25	**start from instruction num2 (pushbutton26)**	The ability to begin the execution from a determined number in the program memory of the second node.
26	**(edit6)**	The number of the instruction that the execution will be start in the program memory
27	**dec to binary1 (pushbutton23)**	When we push this button, we see the value which is
		stored in the memory location of the first node in binary and decimal format .
28	**dec to binary2 (pushbutton24)**	When we push this button, we see the value which is Stored in the memory location of the second node in binary and decimal format .
29	**Show (pushbutton25)**	When we push this button, we can analyze the instruction that we execute in the same moment.
30	**edit17**	The type the first operand in the executed instruction.
31	**edit16**	The value of the first operand before the execution of the instruction.
32	**edit19**	The type the second operand in the executed instruction.
33	**edit18**	The value of the second operand before the execution of the instruction.
34	**edit21**	. The type of the operand that the result is stored in it
35	**edit20**	The result of the instruction execution
36	**edit 22**	The type of the executed instruction.
37	**listbox9**	Interface card1 of the first node.
38	**listbox10**	Interface card1 of the first node.
39	**simple security using MD5 (pushbutton29)**	Instructions which is related to the first secure routing algorithm.
40	**elliptic (pushbutton28)**	Instructions which is related to the second secure routing algorithm
41	**execute step (pushbutton12)**	When we push this button, the execution of the instruction will be start step by step.

B.1 Simulation (Authentication Scheme Between Two Individual Sensor Nodes) Algorithm After Applying it on the Designed GUI

We show in the next figure the memory location values during execution the algorithm:

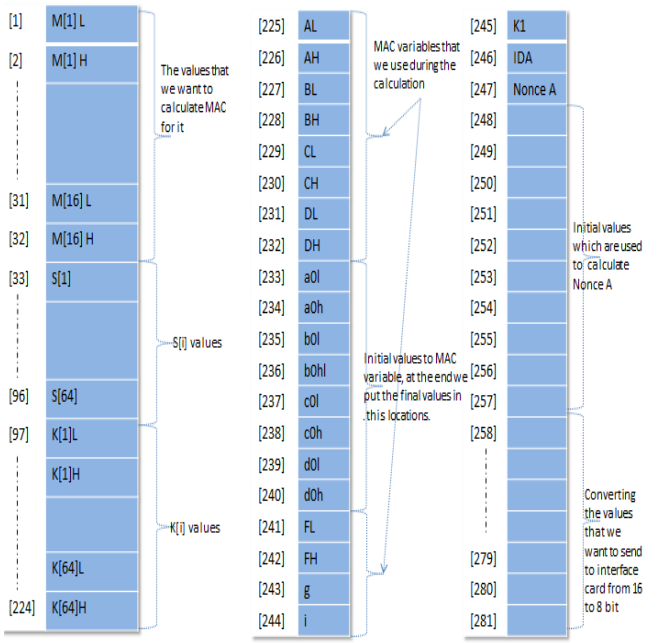

Fig. 13. Memory location values during execution the first algorithm

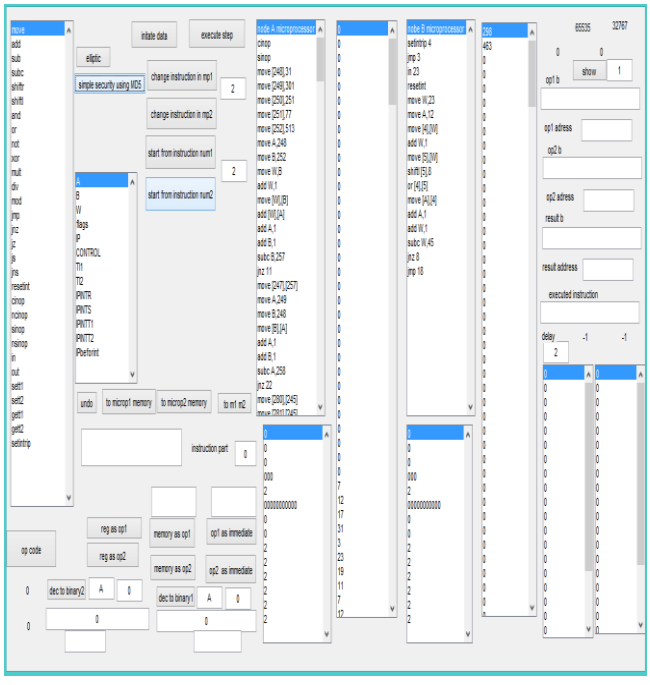

Fig. 14. Initial values when applying the first algorithm before execution

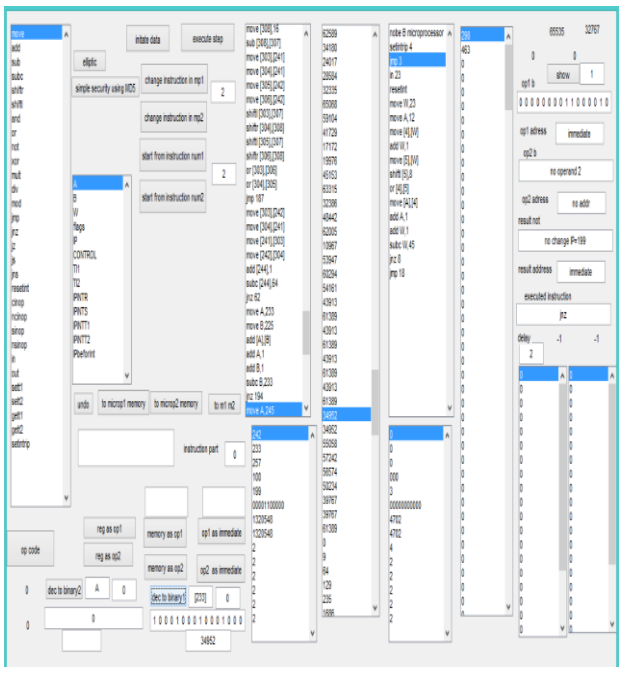

Fig. 15. Calculated MAC values

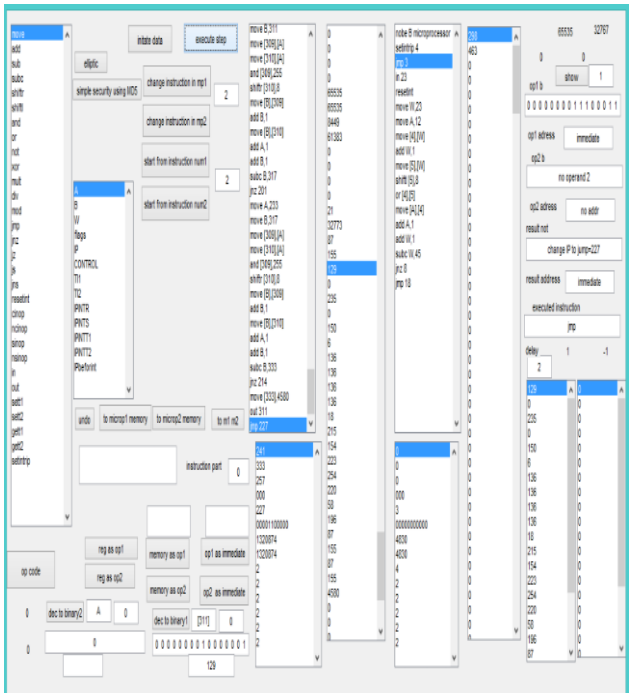

Fig. 16. Message after convert it to 8 bit values for sending

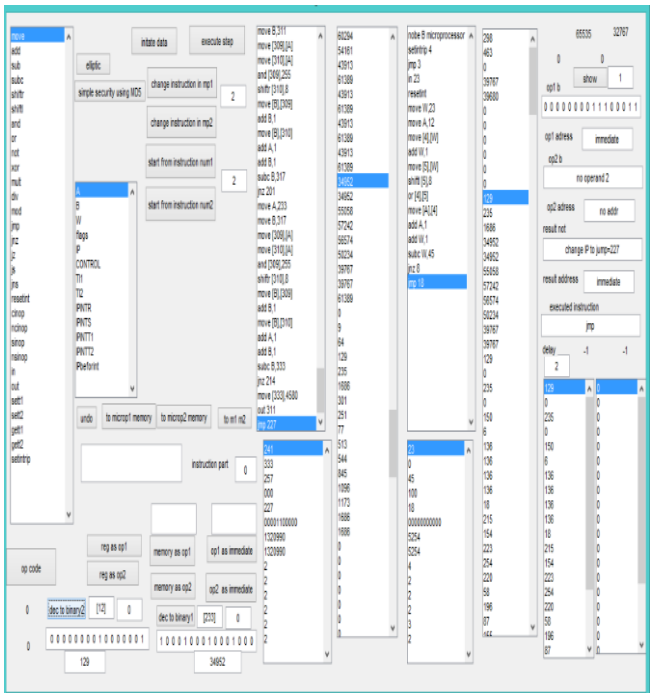

Fig. 17. Values after receiving and reconstruction them

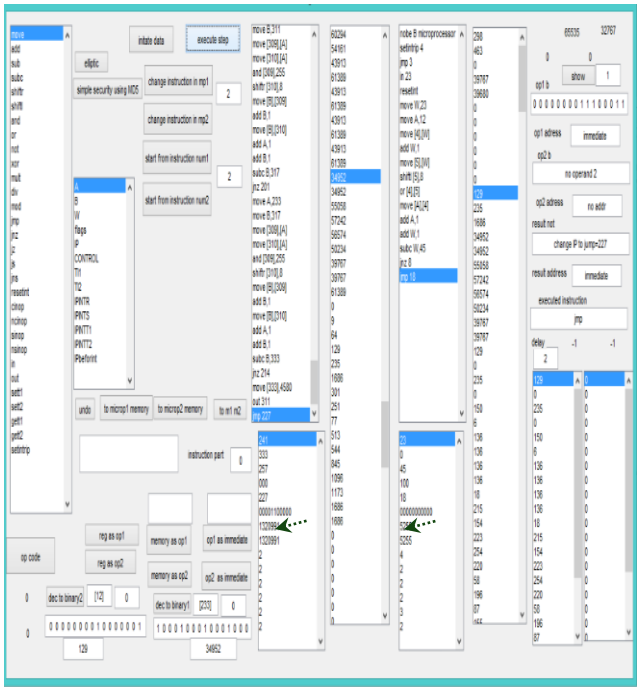

Fig. 18. Values after the end of the execution

We can see through the pointers the clock values for each node:

Number of clocks in the first node: 1320991 clock.

[1]	P1X	[19]	IDB	[37]	high8	[55]	KabX	[70]	rs1
[2]	P1Y	[20]	PBX	[38]	N1	[56]	KabY	[71]	rs2
[3]	P2X	[21]	PBY	[39]	N2	[57]	LAx	[72]	rs3
[4]	P2Y	[22]	Base PAX	[40]	N3	[58]	LAy	[73]	rs4
[5]	P3X	[23]	Base PAY	[41]	N4	[59]	TiA	[74]	rs5
[6]	P3Y	[24]	LBX	[42]	N5	[60]	NA	[75]	rs6
[7]	P4X	[25]	LBY	[43]	N6	[61]	VA	[76]	rs7
[8]	P4Y	[26]	TIB	[44]	N7	[62]	C1	[77]	rs8
[9]	P5X	[27]	Tmax	[45]	N8	[63]	C2	[78]	rs9
[10]	P5Y	[28]	S	[46]	N9	[64]	C3	[79]	rs10
[11]	P6X	[29]	IDA	[47]	N10	[65]	C4	[80]	3840
[12]	P6Y	[30]	LAX	[48]	Kabx	[66]	C5		
[13]	P7X	[31]	LAY	[49]	Kaby	[67]	C6		
[14]	P7Y	[32]	TIA	[50]	LBx||LAx	[68]	C7		
[15]	P8X	[33]	NA	[51]	LBy||LAy	[69]	C8		
[16]	P8Y	[34]	VB	[52]	TiB||TiA				
[17]	Ind PA	[35]	NB	[53]	NB||NA				
[18]	Ind PB	[36]	low8	[54]	VB'				

Number of clocks in the second node: 5255 clock.
In the same way we can simulate the rest of steps using MD5 hash function , so the final clock number of each node:

The final clock number in the first node:

$$5283964 \pm 1000 \text{ clock.}$$

The final clock number in the second node:

$$21020 \pm 1000 \text{ clock.}$$

B.2 Simulation (SecuTPGF) Algorithm After Applying It on the Designed GUI

To simplify the writing of the instruction in this algorithm, we use simple hash function, at the end we can replace it with MD5 hash function.
We show in the next figure the memory location values of node A during execution the algorithm

[1]	P1X	[19]	IDA	[37]	high8	[55]	KabX	[70]	rs1
[2]	P1Y	[20]	PAX	[38]	N1	[56]	KabY	[71]	rs2
[3]	P2X	[21]	PAY	[39]	N2	[57]	LBx	[72]	rs3
[4]	P2Y	[22]	Base PAX	[40]	N3	[58]	LBy	[73]	rs4
[5]	P3X	[23]	Base PAY	[41]	N4	[59]	TiB	[74]	rs5
[6]	P3Y	[24]	LAX	[42]	N5	[60]	NB	[75]	rs6
[7]	P4X	[25]	LAY	[43]	N6	[61]	VA	[76]	rs7
[8]	P4Y	[26]	TIA	[44]	N7	[62]	C1	[77]	rs8
[9]	P5X	[27]	Tmax	[45]	N8	[63]	C2	[78]	rs9
[10]	P5Y	[28]	S	[46]	N9	[64]	C3	[79]	rs10
[11]	P6X	[29]	IDB	[47]	N10	[65]	C4	[80]	3840
[12]	P6Y	[30]	LBX	[48]	Kabx	[66]	C5		
[13]	P7X	[31]	LBY	[49]	Kaby	[67]	C6		
[14]	P7Y	[32]	TIB	[50]	LBx||LAx	[68]	C7		
[15]	P8X	[33]	NB	[51]	LBy||LAy	[69]	C8		
[16]	P8Y	[34]	VB	[52]	TiB||TiA				
[17]	Ind PA	[35]	NA	[53]	NB||NA				
[18]	Ind PB	[36]	low8	[54]	VB'				

Fig. 19. Memory Location Values of Node A,B During Execution the Second Algorithm

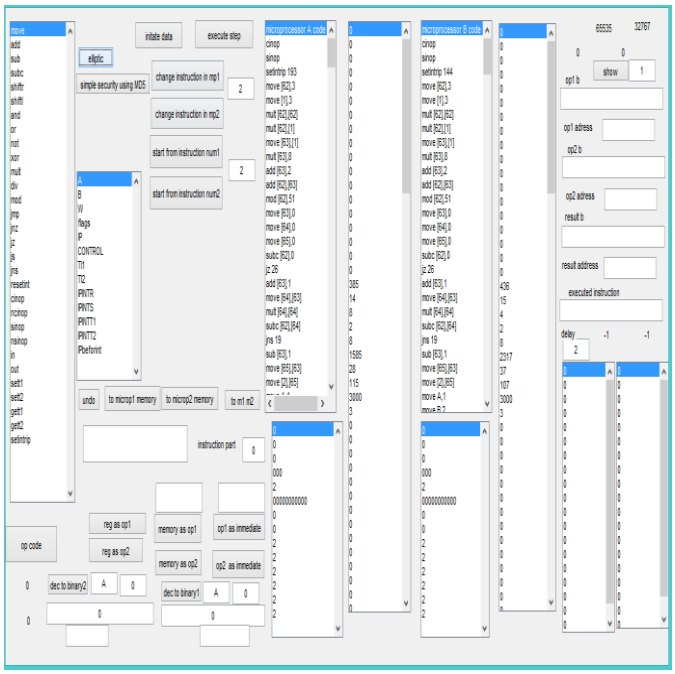

Fig. 20. Initial values when applying the second algorithm before execution

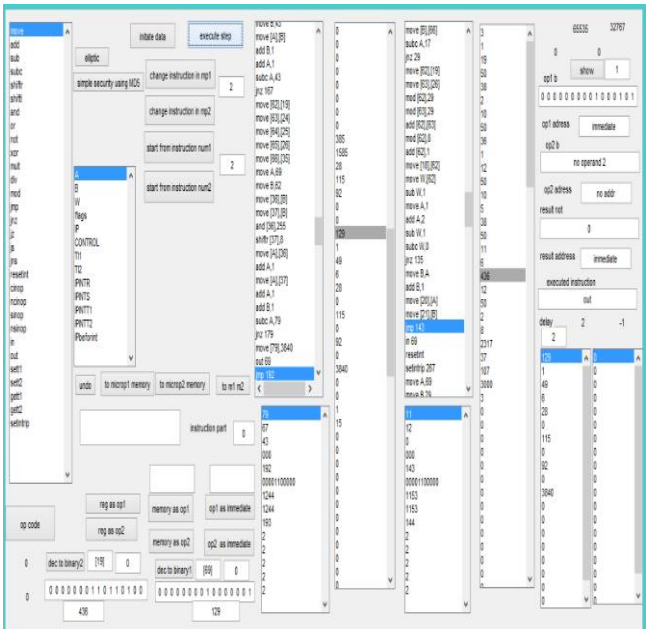

Fig. 21. Hello message which is sent from node A to node B

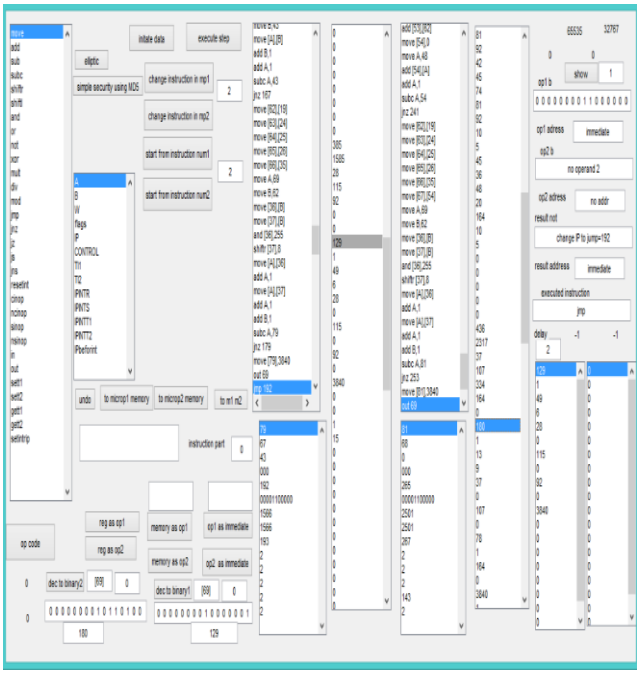

Fig. 22. Values which is sent from B node after calculation VB.

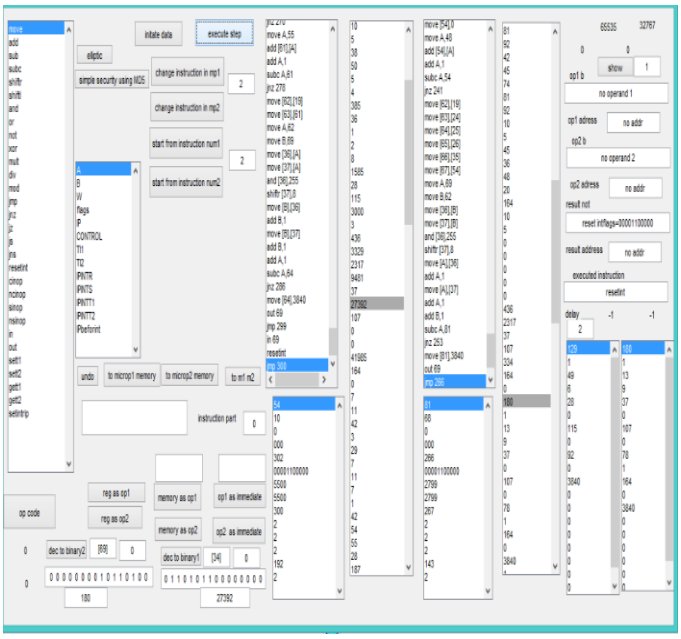

Fig. 23. Values After the End of the Execution.

We can see through the pointers the clock values for each node:
Clock number in the first node: 5500 clock.
Clock number in the second node: 2799 clock.

To compare with the first algorithm, we replace each simple hash function with MD5 hash function during the calculation of PA, Kab, VB',VA in A node and B node, so the final clock number of each node:

The final clock number in the first node:
6610457 ± 1000 clock.
The final clock number in the second node:
5289464 ± 1000 clock.

V. CONCLUSIONS

A. Simulation of TPGF Protocol

After simulation of TPGF protocol, we find that this protocol supports dynamic holes bypassing and elimination of path circles, and that leads to short the routing paths and eliminate end to end delay in compare with other geographic routing protocols.

B. Simulation Two Nodes from the Routing Path

After simulation two nodes from the routing path at the microprocessor level of each node, we find the that (SecuTPGF) algorithm needs relatively higher clocks than (Authentication scheme between two individual sensor nodes) algorithm, but it achieves higher secure level than the second, because (SecuTPGF) algorithm uses elliptic curve to generate keys and ID-NIKDS mechanism, so it prevents the outsider attacks and eliminates the effect of insider attacks, So the trade off between secure routing and delay stays an open issue, and because we choose delay as a basic parameter in our study, we can present a lot of proposals to eliminate delay in SecuTPGF such as:

- The ability to put elliptic curve point values in the memory of the nodes before spread them.
- The calculation of public key in these nodes requires a relatively higher time, so we can eliminate this time by providing the nodes with the keys before spread them.
- Use simple hash function which achieves an acceptable secure level instead of using MD5 hash function that take a relatively higher time.

We can also mention to these recommendations:

- The ability to simulate the transmission of the multimedia information in addition to simulate secure routing information in the future.
- The ability to improve the designed GUI in the future to simulate N nodes as a real network instead of two nodes.

REFERENCES

[1] I.F. Akyildiz, W.Su, Y.Sankarasu bramaniam, E.Cayirci,"Wireless sensor networks: a survey",Computer Network:The International Journal of Computer and Telecommunication Networking,v.38 n.4,p.393-422, 15 March 2002.

[2] R.Verdone, "Wireless Sensor Networks". In Proceedings of the 5th European Conference, Bologna, Italy, 2008.

[3] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," Computer Netw. (Elsevier), vol. 51, no. 4, pp. 921–960, Mar. 2007.

[4] I. F. Akyildiz, T. Melodia, R.Kaushik, Chowdhury" Wireless Multimedia Sensor Networks: Applications and Testbeds", Proceedings of the IEEE,Vol. 96, No. 10, October 2008.

[5] J. Zheng and A. Jamalipour," Wireless Sensor Networks": A Networking Perspective, IEEE Press Editorial Board, John Wiley & Sons, New York, NY, USA, 2009.

[6] M. Guerrero-Zapata, R. Zilan, J. Barcel-Ordinas, K. Bicakci, B. Tavli, "The future of security in Wireless Multimedia Sensor Networks". Telecommunication Systems, December, 2009.

[7] S. Ehsan and B. Hamdaoui, "A survey on energy-efficient routing techniques with QoS assurances for wireless multimedia sensor networks," IEEE Communications Surveys and Tutorials, vol. 14, no. 2, pp. 265–278, 2011.

[8] Camp, J. Boleng, and L. Wilcox. "Location Information Services in Mobile Ad Hoc Networks".In Proc. of IEEE ICC '02, pages 3318–3324, New York City, New York, April 2002.

[9] C. Locherta, M. Mauvea, H. Füßlerb and H. Hartensteinc," Geographic Routing in City Scenarios MobiCom 2004 Poster Abstract"," Mobile Computing and Communications Review, Volume 8, Number.

[10] B. Karp andH. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00), pp. 243–254, August 2000.

[11] S. Medjiah, T. Ahmed, and F. Krief, "GEAMS: a geographic energy-aware multipath stream-based routing protocol for WMSNs," in Proceedings of the Global Information Infrastructure
Symposium (GIIS '09), June 2009.

[12] L. Shu, Y. Zhang, L. T. Yang, Y. Wang, and M. Hauswirth, "Geographic routing in wireless multimedia sensor networks," in Proceedings of the 2nd International Conference on Future
Generation Communication and Networking (FGCN '08), pp. 68–73, December 2008.

[13] B.Leong, S.Mitra, & B.Liskov, "Path vector face routing: geographic routing with local face information". In Proceedings of the 13th IEEE international conference on network protocols (ICNP 2005), Boston, Massachusetts, USA, November 6–9,2005.

[14] T. Mulugeta, L. Shu, M. Hauswirth, M. Chen, T. Hara and S. Nishio" Secured Two Phase Geographic Forwarding Protocol in Wireless Multimedia Sensor Networks," in proceedings IEEE Globecom 2010 at the direction of IEEE Communications Society subject matter experts, 978-1-4244-5638-3/10,IEEE 2010.

[15] R. Sakai, K. Ohgishi, and M. Kasahara, "Cryptosystems based on pairing". In Proceedings of Symposium on Cryptography and Information Security (SCIS'00), pp. 26-28. Okinawa, Japan. January 2000.

[16] A. Joux, "The weil and tate pairings as building blocks for public key cryptosystems". In Proceedings of the 5th International Symposium on Algorithmic Number Theory, Sydney, Australia, July 7-12, 2002.

[17] L. Shu, C. Wu, Y. Zhang, J. Chen, L. Wang, "NetTopo: beyond simulator and visualizer for wireless sensor networks". ACM SIGBED Review, 5(3), 2008.

[18] Mrs S. Sahoo, Mr P. K. Mishra and Prof.Dr.R. N. Satpathy," Secure Routing in Wireless Sensor Networks", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 2, January 2012.

[19] NetTopo .http://lei.shu.deri.googlepages.com/nettopo.

[20] https://en.wikipedia.org/wiki/MATLAB.