Multilevel Onion Tree Routing for Anonymous and Secure Communication in a Wireless Mesh

Abhinav Prakash, Amit Gaur, and Dharma P. Agrawal, *Member, IEEE*

***Abstract*—Although a Wireless Mesh Network (WMN) offers multifold advantages it is also vulnerable to several security and privacy threats being a dynamic open network. Different types of mobile clients such as laptops, cell phones, smart devices can join or leave the network anytime they wish. This opens up issues like fake registrations and packet sniffing. This paper deals with the issues of both security and privacy in great detail by simulating countermeasures for different kinds of attacks in a WMN. First, a perfectly secure network for safe communication is created by using a bi-variate polynomial scheme with low overheads instead of a public-private key mechanism. Further, Ensuring any communication in the network to be rendered anonymous by hiding the node initiating the session by using proposed Multilevel Onion Tree Routing Scheme (MOTR).**

***Index Terms*—Bi-Variate Polynomial, Mesh Networks, Onion Routing, Privacy, Security.**

I. INTRODUCTION

WIRELESS mesh network (WMN) consisting of Internet Gateways (IGWs), mesh routers (MRs) and mesh clients (MCs) seems to be a promising wireless technology. In [1], various challenges and the versatility of WMN are discussed. MCs can communicate with the IGW using multiple hops ad hoc connected MRs or directly with IGW if MC is within its communication range. MRs can act as hosts or packet forwarders in the form of an ad hoc network to the IGW. This ensures a larger coverage at a low cost infrastructure. Hence, MRs are often referred as the last mile network. MCs can consist of different types of devices, like laptops, cell phones, smart devices, etc., working with different types of networks like edge, Ethernet, Wi-Fi, Wi-Max, etc. WMN makes it possible to combine characteristics of all these networks and support different types of devices using only one platform. To enable this, MRs often consists of multiple interfaces in order to perform as Network Bridge or Internet Gateway. For different technologies, MCs generally have a single interface that can either be used while communicating as a host or acting as a network packet forwarder among MCs themselves in the ad hoc network mode. Hence, WMN can be said to be an advanced form of an ad hoc network which is intended to be dynamically self-organized and auto-healing. Existing ad hoc network protocols can be modified for a WMN.

In a WMN, MRs constitutes a wireless backbone by connecting themselves through available wireless channels and MCs connect to the MRs using different interfaces in order to gain access to the internet. This creates a network hierarchy where MRs communicate with each other using a wireless interface at level one while they serve the MCs using different interface at the next level and hence combine the characteristics of two different types of wireless networks. Such a network comprises of devices with varied levels of mobility and different power and computing constraints.

WMN offers several advantages and future potential applications making further research very important. It can be directly used to provide internet access to remote areas which requires availability of low cost infrastructure. Another very important application of WMN is to provide different medium of wireless access. For example, a cell phone in the range of a MR can route it calls through the internet. This enables routing through MR at a cheaper cost instead of using the medium of higher cost low bandwidth cell phone tower. There can be several other applications where MRs can be installed to cover a large region, for example, health monitoring, traffic pattern analysis in airports, etc.

II. RELATED WORK

A. Security

The field of Wireless Mesh Networks is still in its nascent form and hence robust secure and private protocols for a WMN are still in their early stages of development. There are several existing research works investigating various approaches directed towards securing Ad hoc and sensor networks. But the varying level of node mobility in a WMN makes it unique and brings new challenges in the forefront. Therefore, special custom protocols that could address new found vulnerabilities in a WMN. The authors in [2], published in the year 2006 described various attacks like sinkhole and wormhole attacks and also look into some vulnerabilities of a WMN. Some ideas are proposed to combat these attacks like

Manuscript received December 16, 2012.

Abhinav Prakash is with the University of Cincinnati, Cincinnati, OH 45220 USA (phone: 513-546-2280; e-mail: prakasav@mail.uc.edu).

Amit Gaur was with University of Cincinnati, Cincinnati, OH 45220 USA. He is now with the Northwestern University Biomedical Informatics Center, Evanston, IL 60208 USA (e-mail: amitgaur84@gmail.com).

Dharma P. Agrawal is with the Computer Science Department, University of Cincinnati, Cincinnati, OH 45220 USA(e-mail: dpa@cs.uc.edu).

using some kind of elliptic curve cryptography instead of RSA-based public key cryptography. We are already aware of the problem in using an Asymmetric key system for authentication and data communication in a sensor network as it is computationally very expensive. Hence, for the sake of low energy consumption by the mobile devices with limited energy sources, we want our scheme to be light weighted.

Use of symmetric keys seems like a viable option. In [3], Eschenauer and Gligor have proposed a random key distribution scheme to devices in a sensor network. It works by distributing a fixed number of keys to each device randomly selected from a large pool of P key space. This has a major disadvantage that if the fixed number of keys given is small, most of the devices are incapable of communicating with each other as they dont have even one key in common. Hence, lots of disconnected devices are found, whereas in case of a large number of keys given to each device, the scheme becomes vulnerable to a device capture attack. This could lead into large amount of secret being lost, thereby compromising the whole network. Additionally, it is easy to observe that for a network with n-devices, for each pair of devices to have a shared symmetric key each device must be given at least n-1 keys to generate capability of complete connectivity. Instead of just one key, Chan et al. [4], revised the Eschenauers and Gligors model by having at least q (where q>1) keys in common between two adjacent devices have, in order to have a communication link between them. They call their scheme as q-composite random key pre distribution scheme that improves the network resilience attack against the device capture. Blom proposed a symmetric key generation scheme (SKGS) [5], where pair of devices establishes a common key between them by which the amount of secret information they exchange between them, is the least. However, in this approach, there may be dependencies between the keys, and a certain number of users may have to cooperate to resolve the uncertainty of unknown keys. This scheme is not resistant to device capture attack when the number of compromised devices exceed a given threshold value. In [6], Blundo et al. have proposed a secure key distribution scheme for Dynamic conferences where a device may leave or enter the network, thereby constantly changing the network topology. Here, they propose a t-degree bivariate symmetric polynomial pre distribution scheme. The scheme is applicable to any hierarchical networks as well. The communicating wireless devices exchange the polynomials by replacing the variables with their respective IDs. Due to symmetric nature of the polynomial, they are able to compute a common secret key between them. This scheme is k-secure, where k is the degree of the symmetric polynomial.

To establish a common shared key between any communicating entities, Yi Cheng et al. [16] have proposed a pair wise key establishment mechanism (EPKEM) by generating keys and arranging them in the matrix. Each user is allocated a row and a column of keys to form an offline set of keys. The selected elements are then loaded into each device to form its key ring and then they are deployed randomly. In this scheme, two devices discover a common key between them by broadcasting their respective IDs while the indices of keys information are exchanged that are used at the time of key pre distribution. This scheme drastically reduces the number of keys that need to be pre-stored on the wireless devices during the deployment phase, while guaranteeing at least two common keys between any pair of devices.

1) Preliminaries: Polynomial Based Scheme

In [7], we propose a bivariate polynomial function based security scheme. This scheme provides low cost highly scalable dynamic key generation security scheme. In this scheme, we devised a method to establish a secured authenticated connection between any two entities in a WMN. The two nodes can be an IGW, a MR or a MC. The basic concept of this scheme is to provide each node a bivariate polynomial before deployment by the central authority. When deployed, these nodes use this secret mechanism along with some shared information to compute symmetric secure keys and once these keys have been computed, we say that the two nodes have an authenticated association with each other. Before deployment, three different sets are given to a node as follows:

1) A shared key K for initial secure information exchange.

2) A set of Bivariate Polynomial Functions $F_{i,j,k}(x,y)$(where $0 \leq i < l$; $0 \leq j < m$; $0 \leq k < m$) picked randomly from a 3D matrix of polynomials and the indices of the selected polynomial functions.

3) A function *H( )* known as the hash function to compute the shared key from the values received by the bivariate polynomial functions.

The scheme of randomly selecting polynomials from a three dimensional matrix is adapted from Yi Cheng's scheme in [16] and [14]. Every node goes through three stages of

1) Acquiring Secrets.

2) Authenticated Association.

3) Pairwise secure path generation from a Mesh Client to the AAA server or an IGW.

A general Bivariate Polynomial distribution is defined as follows:

$$F_{i,j,k}(x,y) = \sum_{rs=0}^{p} a_{rs} x^r y^s$$

Fig. 1: A three dimensional matrix of bivariate polynomials

the coefficients a_{rs} are randomly selected over a Finite Field $Gf(X)$ where X is a sufficiently large prime number and i, j, k are the indices for the location of the polynomial in a three-dimensional matrix. And p is the degree of the function $F_{i,j,k}(x,y)$. For a Polynomial Function to be Bivariate, it must hold the following property:

$$F_{i,j,k}(x, y) = F_{i,j,k}(y, x)$$

We create a three-dimensional matrix containing randomly selected bivariate polynomials from a large pool of possible polynomials. For simplicity we can visualize this matrix as a set containing i two dimensional $m \times m$ matrices with degree t. Now, we can compute the total number of bivariate polynomials selected from the finite field forming the three dimensional matrix as follows: Total number of polynomials = $l \times m \times m$ such that: $0 \leq t \leq m$ and $0 \leq i \leq l$ So, we can say a function at the position i, j, k in the three-dimensional matrix can be written as $F_{i,j,k}(x, y)$ as displayed in the figure 1. Now, we randomly select a set S of matrices containing bivariate polynomials from l i.e. the total number of m×m two dimensional matrices can be given by: $S \geq \lceil (l+1)/2 \rceil$. where $\lceil x \rceil$ is the ceiling function which gives the smallest integer ≥ x. Using the ceiling function ensures that S is an integer always greater than or equal to $l/2$. This further ensures that two different randomly selected sets S_a and S_b always have atleast one m×m Matrix in common. After selecting the random set S of matrices with one random column and one random row from each of these matrices in S, all the functions contained in the selected row and column are given to a mesh entity. For example refer Figure 2.

Now, since this matrix is of the order m×m it has m rows and m columns. So, the number of polynomials contained in one row and one column selected randomly are $m+(m-1)$. Now, there are S such matrices hence total number of bivariate polynomials given to each MC are:

The total number of bi-variate polynomials = S× (m+ (m-1)) = S× (2m-1)

Fi,0,0()	Fi,0,1()	Fi,0,2()	Fi,0,3()	Fi,0,4()
Fi,1,0()	Fi,1,1()	Fi,1,2()	Fi,1,3()	Fi,1,4()
Fi,2,0()	Fi,2,1()	Fi,2,2()	Fi,2,3()	Fi,2,4()
Fi,3,0()	Fi,3,1()	Fi,3,2()	Fi,3,3()	Fi,3,4()
Fi,4,0()	Fi,4,1()	Fi,4,2()	Fi,4,3()	Fi,4,4()

Fig. 2: A $M \times M$ Matrix S_a

Now, Let us analyze how two clients on the mesh can have common functions to establish a secure communication channel. Since we know two different set of matrices S_a and S_b have atleast one matrix in common. Let the two common

Fi,0,0()	Fi,0,1()	Fi,0,2()	Fi,0,3()	Fi,0,4()
Fi,1,0()	Fi,1,1()	Fi,1,2()	Fi,1,3()	Fi,1,4()
Fi,2,0()	Fi,2,1()	Fi,2,2()	Fi,2,3()	Fi,2,4()
Fi,3,0()	Fi,3,1()	Fi,3,2()	Fi,3,3()	Fi,3,4()
Fi,4,0()	Fi,4,1()	Fi,4,2()	Fi,4,3()	Fi,4,4()

Fig. 3: A $M \times M$ Matrix S_b

matrices be as shown in Figure 2 and Figure 3.

Assuming the highlighted rows and columns were randomly selected row and column for sets S_a and S_b respectively. It is obvious that both these sets will have atleast two functions in common. In a better case, there could be more common matrices, leading to more common functions between two mesh entities. So, this technique of allocating polynomials guarantees any two MCs to have atleast two common bivariate polynomial functions which are used for secured communication.

Generation of Secured Channel for Communication:

When the network is formed, each MC identifies its neighbors and exchanges information to generate a secure key for communication. Say for example, Once two neighbors *C* and *D* find each other, they share each other's node *ID* and the indices of the polynomial functions they possess. This information is encrypted using a common Key *K* which is given to each node for an initial handshake and exchange information to generate a secured key on the fly. Using the polynomial function indices, both the nodes separately determine which function they have in common.

Assuming the node IDs are IDC and IDD and the common functions are $F_{2,7,6}( )$ and $F_{2,3,4}( )$.

At node C, seed values will be computed using the common functions and node IDs of C and D.

$$Seed\ 1_C = F_{2,7,6}\ (IDC\ ,\ IDD)$$
$$Seed\ 2_C = F_{2,3,4}\ (IDC\ ,\ IDD)$$

Similarly, at node D, it would compute its seed values:

$$Seed\ 1_D = F_{2,7,6}\ (IDD\ ,\ IDC)$$
$$Seed\ 2_D = F_{2,3,4}\ (IDD\ ,\ IDC)$$

Since functions $F_{2,7,6}( )$ and $F_{2,3,4}( )$ are bivariate polynomial by this property

$$F_{2,7,6}(x, y) = F_{2,7,6}(y, x)$$

The same is true for any other common function therefore:

$$Seed\ 1_C = Seed\ 1_D$$
$$Seed\ 2_C = Seed\ 2_D$$

This is applicable to any further seeds.

So, the seeds generated independently at both the nodes would be identical. Each node uses a one way hashing function $Hf( )$ that is assigned during the deployment phase. All the seed values are hashed to generate a final secured key for communication and since the same hash function is used at both the nodes and seed values being identical, they both have the same identical unique key for encryption or decryption. This key is never sent over the network and is just used for encryption at the sender end and decryption at the receiver which ensures the key not to be stolen by other entities that might be overhearing the communication.

$$SecureKey = Hf(Seed\ 1_C, Seed\ 2_C, \ldots)$$

If the two nodes have more than two common functions, they generate more than two seed values. This provides more than two seed values to the hashing function which makes it even stronger and more secure. In this fashion, all the nodes establish secure encryption technique with their one hop neighbors. Assuming a node A needs to communicate to the Internet Gateway which is five hops away from it. Each of the four links on the way would be a secured connection using pairwise key. This pairwise secure key establishment ensures end-to-end secured transfer.

B. Privacy

A lot of work has been done in the field of security for a WMN. But, multiple vulnerabilities still remain open in the field of privacy. Several research work attempt to solve the

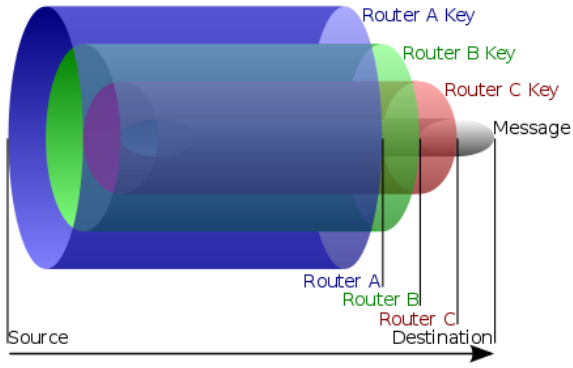

Fig. 4: An Onion Packet [11]

privacy issues related to a wired network. But, there is still a lot of scope in this field. For example, in a wired network, Onion Routing [11] was invented by Michael G. Reed, Paul F. Syverson, and David M. Goldschlag to provide anonymity. In this scheme, a path is pre-computed at the source and the data packet is encrypted in multiple layers with the public key of the forwarding node along the path to destination in a sequential order and each node removes their layer of encryption after receiving the packet and forward the remaining packet called the onion to the next hop and finally the decrypted data with all the layers of encryption removed, received by the destination. Using this approach, each node is only aware of the previous or next hop node which ensures anonymity of the source and the destination.

In [13], Xiaoxin Wu introduces an onion ring protocol for a wireless mesh network where onion rings are formed starting at the gateway router and using all the cycles in the network. Data only travels in one direction and data sessions are only initiated at the gateway router. This provides a good anonymity, while fails on several issues such as a bottleneck is created at the gateway router as all the scheduling is done at the gateway. Moreover, this scheme works on the concept of finding cycles. In a dynamic WMN, having a property of cycles need them to start and end at the MR after traversing

through the network. This problem is found to be an NP-hard to solve. So, it can fail in a realistic scenario. Another problem in this protocol is of the node starvation as all the other nodes in a ring have to wait while one node communicates. The work [9] talks about a layered onion ring approach in which some routers are considered trusted nodes and is similar to [13]. Communication starts and ends at the same node. Hence, anonymity is induced in the network. But, this work strongly relies on finding the cycles in the network and is hence susceptible to similar drawbacks. In [8] and [12], authors propose to implement a phantom routing scheme using a random walk algorithm so as to prevent any attacker to trace back in a Wireless Sensor network. But, there are several challenges in implementing this scheme in a WMN because of its inherent characteristics. The Scheme in [10] inserts dummy traffic in a wireless network to achieve anonymity at the cost of increased communication cost. This helps in hiding the source and the destination by inserting fake encrypted messages to the existing traffic pattern corresponding to the actual data packet stream. This scheme, though otherwise efficient, fails against an active global attacker using a dynamic tracing algorithm.

In [15], authors propose using fake sources to provide spatial *L*-diversity in addition to a traditional *k*-anonymity scheme to tackle with the problem of a global attacker for achieving source location privacy in a Wireless Ad Hoc Network. The parameter *L*-diversity for quantifying privacy is also introduced in this work.

III. MULTILEVEL ONION TREE ROUTING SCHEME

We assume an infrastructure based WMN where the MRs and an IGW form a backbone by connecting each other through the wireless medium and provide service to MCs. MCs can be mobile or static and can join any MR for the internet access. One or more MRs act as the Internet gateway connected directly to the Internet.

In [13] the authors have proposed creating onion rings for private communication in a WMN. But, in a realistic condition, finding cycles (holding the property of an onion ring cycles) in a wireless network is observed to be an NP-hard problem. Moreover, dynamic nature of the WMN makes it even more difficult as cycles have to be found possibly at a regular interval which can be very time consuming and computationally expensive process. In this work, a novel concept of a Multilevel Onion-Tree routing protocol is introduced for a WMN. There are two levels in this protocol, level one (higher level) being at the MR (backbone) level along with IGW and Second level or the lower level comprising of a MR and its MCs.

A. Lower Level Communication

For the lower level at each MR, we find a spanning tree with MR being the starting node such that we cover all the MCs associated with the MR. In [19], authors propose an efficient method of finding a perfectly random spanning tree of a directed graph.

B. Virtual Ring Initiation

The MR attempts to form an onion by informing each MC about their membership in a virtual ring as there might be some overlapping branches and in such a case, some MCs could be a member of multiple virtual rings. We need to have a mechanism to avoid any conflict among multiple virtual rings. Hence, each leaf node (MC) needs to store a table with the information about all the virtual rings it is a member of. The virtual ring initiation onion is created at the MR by encrypting virtual ring information in multiple layers for each member of the virtual ring. Each one of these layers contains virtual ring information unique for a particular node and is encrypted by the shared key between the MR and that node. So, if a layer contains information for node A only or the MR can remove that layer. Hence, the onion is created by performing encryption in multiple layers of a specific order in which the

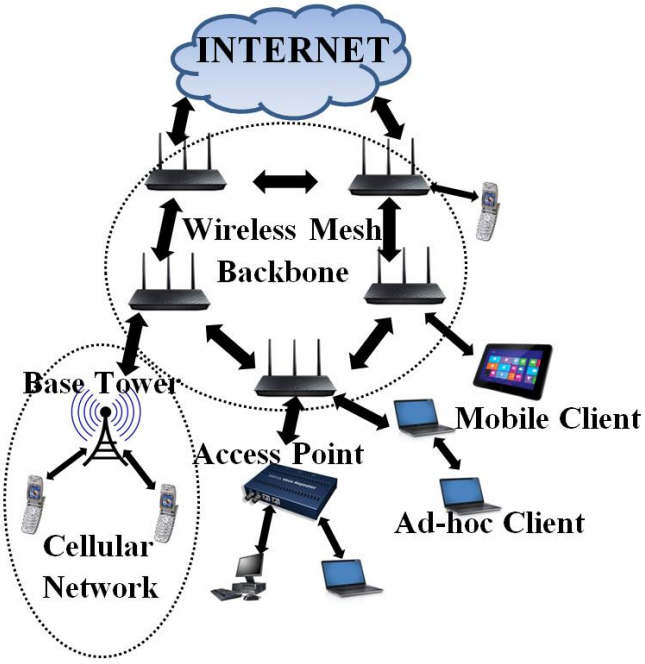

Fig. 5: A Wireless Mesh Network

onion is going to traverse hop-by-hop along the virtual ring. So, when the onion arrives at a node, it decrypts the layer in the onion that belongs to it to get the virtual ring id for that particular virtual ring and information about who is the next hop for that virtual ring, the id of the previous node and the remaining onion that needs to be sent to the next hop. The order of layered encryption is important here because if a node receives an onion in which the layers on top of the layer meant for itself has not been removed, It cannot decrypt the packet to extract information meant for it. Hence, rendering the onion useless. But, this also serves as the most important feature for providing privacy as none of the nodes have information about the complete virtual ring whereas they only know the previous hop and next hop neighbor for a particular virtual ring. Hence,

the network is protected from inside attacks. A typical virtual ring initiation onion looks like:

$E_{MRD}[(VRID,Nextid_D,Previd_D,Level),[E_{MRC}(VRID, Nextid_C, Previd_C,Level),[E_{MRB}(VRID,Nextid_B,Previd_B,Level),[E_{MRA}(VRID, Nextid_A, Previd_A, Level)]]]]$ where E_{MRD} is the key shared between the Mesh Router and the node *D*, *VRID* is the virtual ring identifier, $Nextid_D$ is the identifier of the next hop node for node *D*, and $Previd_D$ is the identifier of the previous hop node for node *D*.

For the first node of any virtual ring, Previd would be the id of the mesh router, and for the last node of any virtual ring $Nextid_i$ would be *NULL*. Level field stores the level of privacy provided by the virtual ring *VRID*.

The level of privacy provided by each virtual ring is directly proportional to the number of nodes present in that virtual ring. Higher privacy is achieved at the cost of larger delay. If a node is only a member in one virtual ring, it has no option but to use that virtual ring for communication when required at the given privacy levels. In case multiple virtual rings are available, a node can use it‘s virtual ring table to decide which virtual ring to use for communication based on the acceptable privacy level and delay tolerance of the application needing the communication. Another deciding factor for this selection is the availability of a career signal in the virtual ring from a MR. A virtual ring which can provide high level of privacy but has non-availability of a career signal, meaning the MR is busy serving another MC in that virtual ring, might be unacceptable due to an added long queuing delay which makes the total delay even longer. Whenever a leaf node receives a virtual ring initiation message, it appends the information to its virtual ring table, in which each row contains information like following:

TABLE I
A VIRTUAL RING TABLE ENTRY

VRID	*Nextid*	*Previd*	*Level*
34	*A3E*	*E47*	*2*

Whenever MC leaves a MR, all the virtual rings need to be deleted for which it was a member. On the other hand, if a new MC joins a MR, new virtual rings need to be created that include the new node. Hence, in both cases, the process of virtual ring initiation is repeated. This process of virtual ring creation or deletion is dynamic and depends on the mobility rate of the MCs. In case the network is static, we want to add new virtual rings at a regular interval. This is done for several reasons, namely, creating new virtual rings enables the usage of links that have not being used previously and could reduce the load of the overused links. This provides a good mechanism of load balancing in the network and increasing the privacy of the network at the same time. The virtual ring table at each node is also dynamic and is constantly updated with new information to keep all the nodes updated with the most recent virtual ring information.

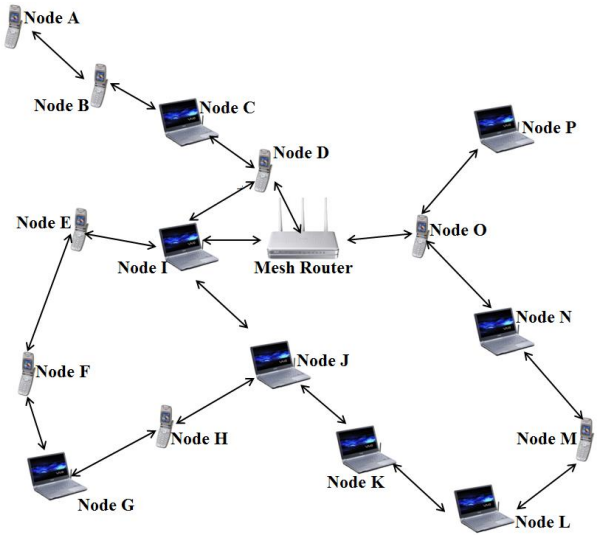

Fig. 6: Mesh Router Level Network (Lower Level)

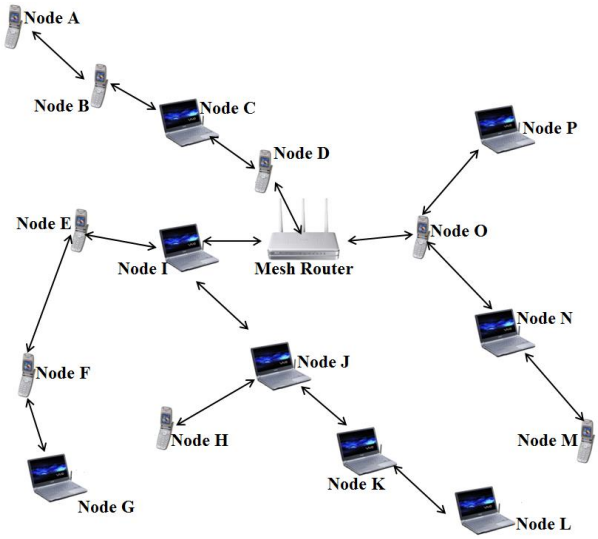

Fig. 7: A Uniform Spanning Tree at a Mesh Router Level

C. Key Revocation

The regular process of virtual ring creation helps us to revoke the old keys and establish new keys every time a new virtual ring is created. Whenever the MR sends a new virtual ring creation request (onion), it can append a new key for every member node. This is our new mechanism for dynamic key renewal on the fly. The onion header in that case looks like this:

$E_{MRD}[(VRID,Nextid_D,Previd_D,Level,NewE_{MRD}),[E_{MRC}(VRID, Nextid_C,Previd_C,Level,NewE_{MRC}),[E_{MRB}(VRID,Nextid_B,Previd_B, Level,NewE_{MRB}),[E_{MRA}(VRID,Nextid_A,Previd_A,Level,NewE_{MRA})]]]]$

D. Virtual Ring Communication

To ensure anonymity, no node is allowed to initiate a

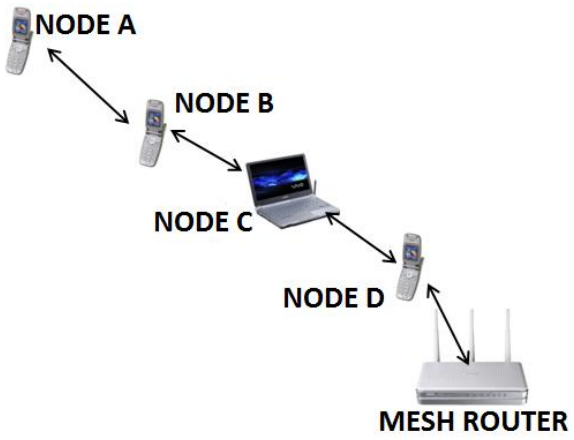

Fig. 8: A Branch starting at a Mesh Router

communication directly which is always started by a MR. To do so, MR sends a data request carrier signal to all the virtual rings at regular intervals, even if none of the nodes have data to send. For example in Figure 8 the MR sends E_{MRD} (req, *VRID*, *dummy*) to Node *D* where req is the identifier flag for a request carrier signal. After receiving the packet, Node *D* decrypts the packet and if Node *D* wants to initiate session, it replaces the dummy with the data request packet and encrypts it again with E_{MRD}. Then, it sends the encrypted packet to the next hop of the virtual ring by looking up from the Nextid in its virtual ring table for the given *VRID* in the format: $E_{CD}[req, VRID, E_{MRD}(request)]$ where E_{CD} is the key shared between *C* and *D* as every two nodes in the network have at least one shared key for communication as per our polynomial based scheme defined in [7] established in the neighbor discovery phase. If Node *D* has nothing to send it sends $E_{CD}[req, VRID, E_{MRD}(dummy)]$ instead.

The total size of the dummy and the request packets are kept consistent so that any intermediate node cannot make a difference between a dummy and a real request packet by the traffic analysis, trying to find out which node is initiating a session. When *C* receives this packet, it decrypts it using the shared key, to find the virtual ring it belongs to and if it has something to send it drops the encrypted request of the previous node (Node *D* in this example) which looks like dummy to it and appends its own request instead by encrypting it with the shared key between itself and the MR associated with the virtual ring identifier *VRID*. It must be observed that a node might be connected to multiple MRs and be a part of multiple virtual rings having roots at different MRs. So, when it has data to send, it sends to the next hop (Node B in example): $E_{BC}[req, VRID, E_{MRC}(request)]$, else it just adds another layer of encryption by sending: : $E_{BC}[req, VRID, E_{MRC}[E_{MRD}(request)]]$ and hence a layer of onion is formed. When this message (onion) reaches the last node, which is detected by the *Nextid* field of the last node being *NULL*, it acts according to its communication needs and identifies the virtual ring traversal is complete. Adding its layer of encryption, the last node routes the onion back in the reverse direction securely to the MR. Once received by the MR, the onion is peeled until a request is found and then the MR decides to grant the request or not and sends a reply. If no request is found, that means no node in that virtual ring is requesting communication or one of the nodes is acting

TABLE II
PACKET PROPAGATION ALONG THE VIRTUAL RING NODES

Hop	*Packet*
MR to D	$E_{MRD}[VRID, E_{MRB}(downlink\ data)]$
D to C	$E_{DC}[VRID, E_{MRB}(downlink\ data)]$
C to B	$E_{BC}[VRID, E_{MRB}(downlink\ data)]$
B to A	$E_{AB}[VRID, E_{MRB}(ACK, uplink\ data\ or\ dummy)]$
A to B	$E_{AB}[VRID, E_{MRB}(ACK, uplink\ data\ or\ dummy)]$
B to C	$E_{BC}[VRID, E_{MRB}(ACK, uplink\ data\ or\ dummy)]$
C to D	$E_{DC}[VRID, E_{MRB}(ACK, uplink\ data\ or\ dummy)]$
D to MR	$E_{MRD}[VRID, E_{MRB}(ACK, uplink\ data\ or\ dummy)]$

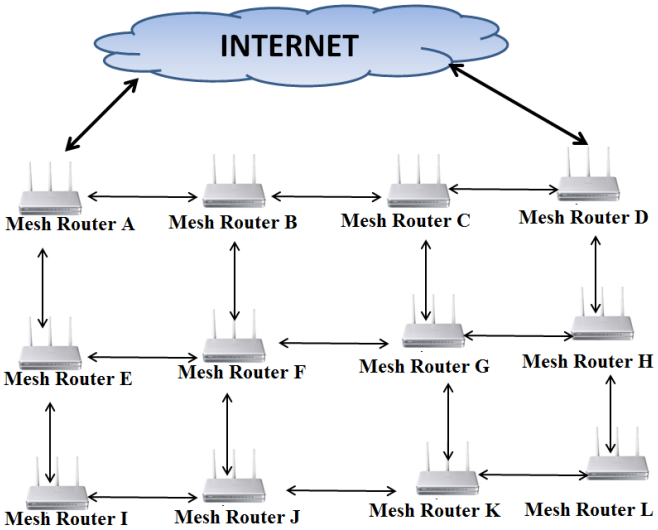

Fig. 9: Higher Level Network with IGW's

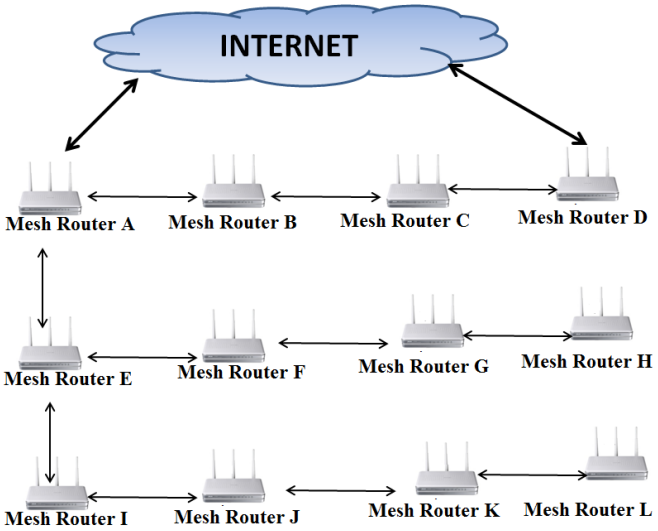

Fig. 10: A Uniform Spanning Tree at Higher (IGW) Level

malicious and voluntarily drop the request and appended a dummy packet which can be easily detected by comparing the received dummy packet with the one that was sent. If both the dummy packets differ, that means one of the nodes in the

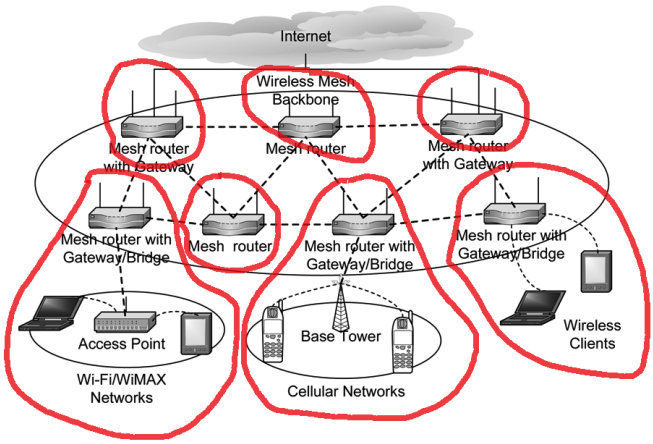

Fig. 11: Lower Level Sets

virtual ring has modified the package and is acting suspicious. It can be clearly seen that the malicious node is the one whose shared key is used to reach the last layer of decryption process in obtaining the fake dummy packet. When an access request is granted, the MR encrypts the downlink data with the shared key between the MR and the destined node. In the example of Figure 8, if *B's* access request is granted the message propagates as follows:

On the reverse cycle, when the packet passes Node *B* again, it verifies the packet to be the same as the one it had sent on the downstream. If not Node *B* reports the branch starting from itself until the end as suspicious to the MR. This data aggregated over time can be utilized to identify malicious nodes in the network and the transmission session is over for Node *B*. The MR keeps sending a request carrier signal at regular intervals until some MC requests an access.

E. Higher Level Communication

At the higher level of routing, a spanning tree is created by the IGW routers, covering all the MRs'. Virtual ring initiation and communication takes place at the manner similar to the lower level. The only difference is the communication between the MRs and IGW instead of the MCs and the MR at the lower level. This forms a hierarchal network in two layers. In the lower level all the MCs send their request to a MR, they are associated with using Onion-Tree routing. Once the requests are at the router level, they are forwarded to the IGW using Onion-Tree routing in a similar fashion. When multiple IGWs are present, multiple trees are formed with overlapping branches. Each MR maintains a virtual ring table to keep record of the entire virtual rings it belongs to. It can decide which virtual ring to use for communication depending on the factors like delay tolerance, level of privacy requirement and availability of request career access signal from an IGW.

IV. IMPLEMENTATION DETAILS

This section looks further deep into our proposed Multilevel Onion Tree Routing (MOTR) scheme. An effort has been made to explain each and every step involved in the implementation of the MOTR scheme in great detail. For a better understanding, WMN is divided into two levels, higher level constituting of the MRs and the IGWs. The lower level constitutes one MR and associated MCs with it. Hence, there are multiple independent sets of entities at the lower level as shown by circles in the Figure 11. Entire WMN can be broken into these sets as subgraphs for lower level computation.

The membership of each of these individual sets is dynamic as MC's keep leaving a MR and joining the network through another MR. This is the first difference between the lower

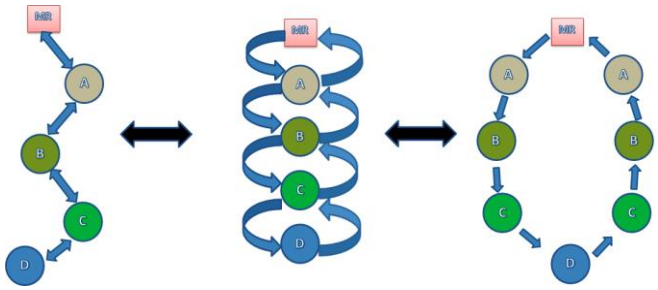

Fig. 12: A Virtual-Ring created from a branch

level network and higher level network as the members of the higher level, which is termed as the network backbone and generally remains invariant. Secondly, the entities in the higher level of the WMN are assumed to have adequate access of energy and computing capability. These are general properties of a WMN, while MOTR does not depend on these factors or utilize such assumptions. Hence, MOTR can be equally applicable to a WMN where all entities have some level of mobility.

When the network is initiated, all entities possess appropriate shared secrets to establish pairwise symmetric key for secure communication as per PBS scheme discussed earlier in Section I-A2. It is critical to ensure security and privacy at the lower level of the WMN as it consists of untrusted user nodes that can be a possible threat, as their association to MRs changes dynamically. Another factor considered is the fact that most of the entities in the lower level network layer have fixed power supply and limited computing capability. Hence, PBS plays a very important role in establishing secure pairwise symmetric keys at a significantly lower cost as compared to an asymmetric public-private key system. Each member set of lower level WMN can be viewed as a graph, with the member entities as vertices', which can be converted into a tree by removing the cycles from it and MR being the root of the tree. This network graph computation is done by the parent MR as it has the complete knowledge of the graph containing the MR and the registered MCs. Every branch starting at the MR and ending in a leaf node of the tree can be viewed as a virtual ring. Figure 12 shows the formation of a virtual ring from a

TABLE III
NUMBER OF POLYNOMIAL FUNCTIONS REQUIRED VERSUS NETWORK SIZE

l	*s*	*M*	*Functions*	*Nodes*
4	*2*	*1*	*2*	*6*
4	*2*	*2*	*6*	*96*
4	*2*	*3*	*10*	*486*
4	*2*	*4*	*14*	*1536*
4	*2*	*5*	*18*	*3750*
4	*2*	*6*	*22*	*7776*
4	*2*	*7*	*26*	*14406*
4	*2*	*8*	*30*	*24576*

branch.

The packets in this virtual ring can be viewed as always traveling in one direction on a circular path, literally along the branch the data travels from the MR traversing the entire branch until the leaf node and then backwards along the same path to the MR. This simulates the effect of a ring, hence named as a virtual ring. This process of virtual ring creation takes place at each MR at regular intervals which is called as the time interval. Repetition of this process is critical to account for the changes in the network structure, leading to disconnections in pre-existing branches and could include newly created links. Adding new branches also increases the number of nodes in the anonymity set, hence achieving a higher level of privacy. So, a low value of time interval can help achieve higher level of privacy and current branch tables leading to increased number of successful transmissions. But, on the other hand, a very low value of can lead to high overhead and could overload the network. Privacy is achieved at a cost of overheads. The value of should be optimal by maintaining a stable network while providing highest level of privacy possible under the given constraints. Once the network has been initiated, the MOTR scheme works at the lower level in the following steps:

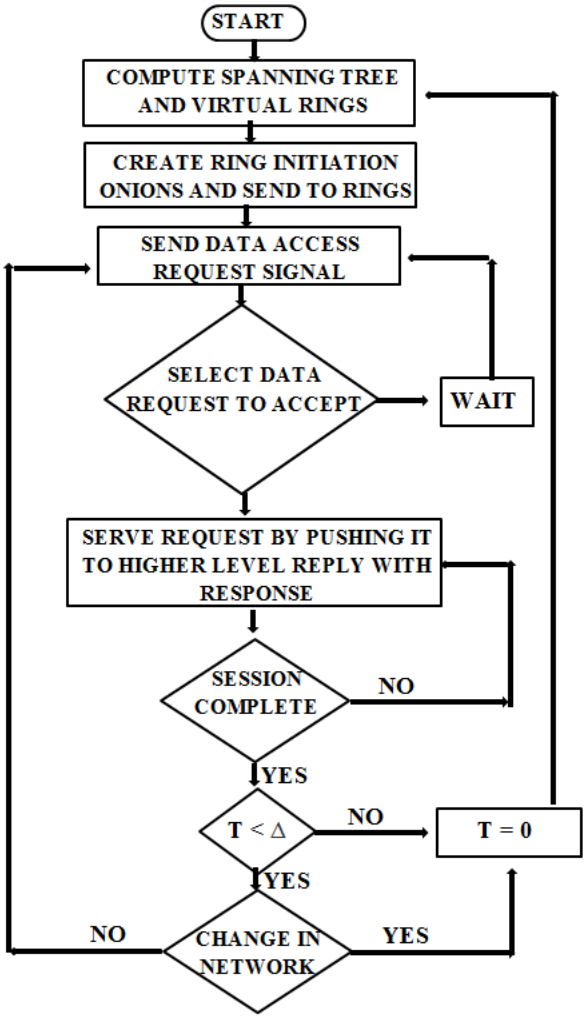

Fig. 13: Work Flow of a MR for Lower Level Network

1) Find a spanning tree for the subgraph with MR being the root.

2) Find branches containing both a leaf node and the MR. We call these branches as virtual rings.

3) MR sends a virtual ring initiation onion packet to all the virtual rings found. In this step, a fully loaded onion is packaged at the MR and is peeled hop by hop as it traverses through a virtual ring while distributing virtual ring information to each member node.

4) Each node receiving the onion decrypts the layer (peels) meant for it and appends the new ring information to its virtual ring table and forwards the remaining onion to the next hop.

5) Once all the virtual rings have been registered at the MR, the MR sends a data request career signal packet to each virtual ring. In this step, each member of the virtual ring adds a layer of encryption to this packet. The onion is complete at the leaf node of the virtual ring which is transmitted back to the MR. Upon receiving this onion, MR decrypts all the layers of the onion to extract a data access request.

6) The MR might receive multiple requests from several virtual rings simultaneously. The MR can decide which requests to serve and in which order. Then, the MR responds with the permission to send. In the communication phase, no onions are required. The data is encrypted by the shared key between the receiver and the MR. When these data packets traverse along the virtual ring, they are encrypted by a second layer of encryption using the shared key between the forwarding node and its previous hop in the virtual ring as shown in the Table II.

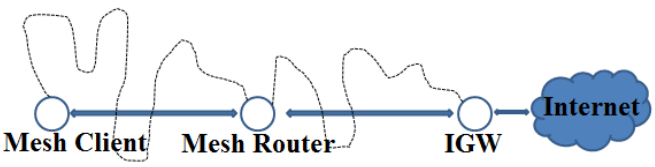

Fig. 14: A Packet Propagation Path from Client to IGW

7) Once data access session is over, steps 5 through 7 are repeated until time interval expires or there is a change in the network structure. Steps 1 through 7 are repeated in that

case.

Onion packets are used to protect the identity of the sender and the receiver. That is why we use it in both the critical steps of Virtual Ring Initiation and Career Access Request Signals. But, we do not use onion packets in the data communication phase, once the Virtual rings are formed, we just use one of the rings to communicate using the Virtual ring table. The only information revealed to an intermediate node is the previous node id and the next hop id. This also reduces the overhead of multiple encryptions and is simple to implement.

Since, MR acts as a bridge between the lower level network and the higher level (backbone) WMN, it possess' at least two interfaces to act as a bridge. All the MRs in the backbone wait to be served by the IGW very similar to MCs in the lower level. MR cannot initiate a data access session as it has to wait for a career access request signal by the IGW and until the time its request has been accepted, only then it can start a data session. During this time, all the requests that have been pushed from the lower level to a MR have to wait in the queue at the MR. This adds up to a queuing delay. But, this is the price we pay for high level of security. According to our assumptions, MRs have sufficient memory available for an unlimited data queuing. The computing and power capabilities of a MR are also assumed to be more than sufficient. Thus, at the higher level (backbone), we need not hesitate to use Public-Private key pairs and onion protocol for every transmission. Techniques like backbone flooding have also been proved to perform well for example as in [20].

V. PERFORMANCE ANALYSIS AND RESULTS

In [7], we have shown that our PBS (polynomial based scheme) is highly scalable and perfectly secure which is provided at a very low cost as we can see in the Table III that with very low number of functions stored on a MC we can support a very large sized network with full connectivity.

The proposed Multilevel Onion Tree Routing (MOTR) takes this to the next level and fills the gap of privacy in case of an attack by a global adversary. Very high level of anonymity is achieved at the cost of some incremental overheads, like multiple encryption and added delay. In case of Multilevel Onion-Tree Routing, the overheads are caused by the redundant paths and Multiple Encryption as onion routing is used in some cases. It can be debated that the onions have a heavy cost of usage. But, in Multilevel Onion-Tree Routing, we only use Onion layers to initiate the branches. Once branches have been established, only two layers of encryption is used, first layer of encryption is done utilizing the shared key between the MR and the client in an active session and then a second layer of encryption is used by the forwarding MCs for secure communication using the shared key among themselves in each hop which is established initially by using the PBS scheme. This helps us keep the cost to bare minimum for multiple encryptions as compared to a pure onion ring routing in [13] and layered onion ring routing approach of [9] where they always use onions for any type of communication.

Another issue of redundant paths can be justified by the logic that all the nodes in a branch are scheduled for sessions of communication by the MR they are registered with, which makes sure that there is a global load balancing and 100% channel access bandwidth is used with CSMA/CA. We can also make sure to keep the delays under a certain level by limiting the length of branches not too long in case of applications that are sensitive to delays and require higher QoS (Quality of Service). This scheme uses more energy than a regular end-to-end communication in a typical WMN. But, this added cost enables us to achieve a very high level of privacy. Redundancy in Multilevel Onion-Tree Routing is much more efficient than Phantom Routing of [12] which is based on flooding to ensure privacy among a group of nodes that requires too many retransmissions. It can also be intuitively observed that in a branch with MCs closer to the MR have lower probabilities of getting served, whereas nodes farther away are given a higher priority to access the communication medium. This strategy serves to our benefit as it assures load balancing as in a typical WMN, as MCs closer to a MR get increased access to the communication medium. Another factor that also makes sure that the MCs, present in the close vicinity of the MR, do not get starved as they have a higher chance to be a member of multiple branches if they exist. Nodes farther away are more spread out and do not have any such benefit.

1) Anonymity

To an outside observer, all the nodes in the WMN along

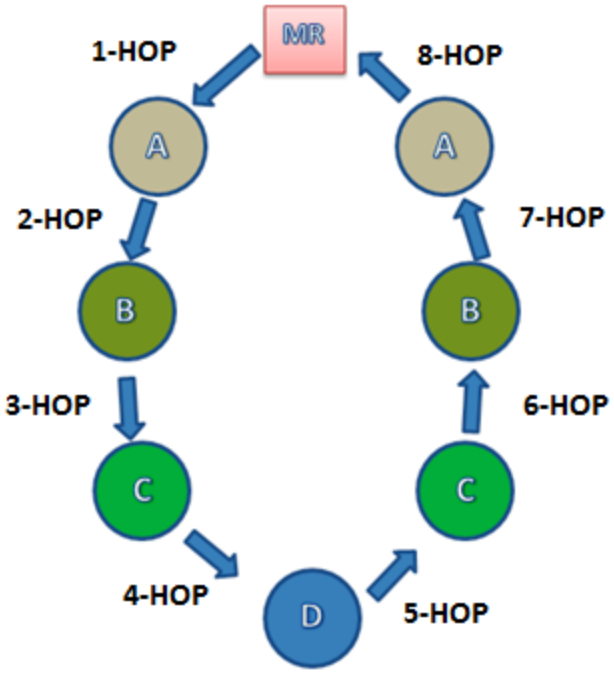

Fig. 15: Analyzing Weighted Hop Count

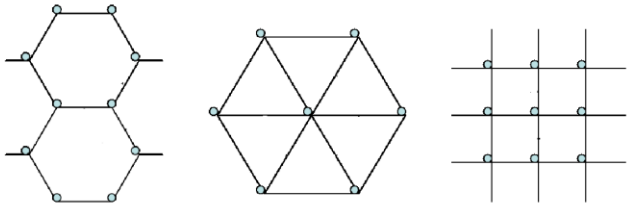

Fig. 16: Regular MC Deployment in Hexagon, Triangle and Square Patterns

both the layers act exactly the same. The encrypted communication combined with the usage of dummy packets makes it impossible even for the global observer to isolate the node initiating the communication session. Additionally, a session is never initiated at the MC as it can only request for a session to a MR and the session is always initiated at the MR. In case of presence of an inside attacker, much information

TABLE IV
END TO END THROUGHPUT IN AN AD HOC NETWORK USING MULTI-HOP TCP

Hops	Throughput (kbits/sec)	Latency (ms)
1	2451	14
2	771	26
3	362	45
4	266	50
5	210	60
6	272	100
7	181	83
8	159	119

cannot be leaked as each MC only knows about the 1-hop information among the branch (previous and next entity). The data packets forwarded by an inside attacker are encrypted and it cannot make a difference between a dummy packet and a data packet as they look exactly the same to an inside attacker.

2) Intersection Attack by Global Adversary

In [9], the authors talk about a new kind of attack that our WMN can be vulnerable to. In this work, they use onion rings for anonymity and consider a global attacker observing all the ongoing transmissions. In the case of such a global attacker, it can monitor the pattern of flow of traffic. In case of low traffic network and fewer numbers of rings, it might be able to isolate rings by monitoring the flow of traffic. Furthermore, if there is

	Total Number of Nodes = 48		
	Deployment		
	Triangle	Square	Hexagonal
Average Hop count End-End	2.72	3.5	3.79
Average Hop count MOTR	3.26	4.33	4.66
Average End-End Throughput (Kbit/sec)	710.64	547.42	472.02
Average MOTR Throughput (Kbit/sec)	377.45	291.71	247.53
Average End-End Latency (ms)	37.36	47.58	49.85
Average MOTR latency (ms)	44.49	59.21	60.85

Fig. 17: Comparison of findings in different deployments

a node that accesses an address visited very rarely by few users and uses two different rings to access this address, the global attacker might be able to isolate the node initiating the session by taking the intersection of the two rings. In our scheme, the flow of traffic goes through two layers of branches which make it impossible to isolate the session initiator. One branch first takes the packet from the MC at the lower layer to the MR. Then, another branch randomly propagates the packet to the IGW. In our case, the selection of branches on different levels is totally independent which makes it difficult to predict what path the packet is going to take. Additionally, availability of multiple branches adds further randomization to the selection of the final path taken. Multiple layers and availability of several branches makes our scheme more or less private and secure. Furthermore, these branches keep on changing dynamically over time.

3) Finding Spanning Trees

In [17], a Spanning Tree Protocol (STP) is described, which has been standardized as IEEE 802.1D and utilized to create spanning trees in a WMN. We propose to use the same for our scheme. We use a Uniform Spanning Tree for our purpose which is a random spanning tree chosen with equal probability among all possible spanning trees in a connected graph. We can use any random walk algorithms to find such Uniform spanning trees. Wilson's Algorithm [18] comes out as a fitting solution with a linear time complexity and hence we use that for our Multilevel Onion-Tree Routing scheme. Hence, we can find Uniform Spanning trees in real time on the fly and create Virtual Rings and revoke old ones dynamically at a regular interval. Since these tasks are performed only at the MRs, we

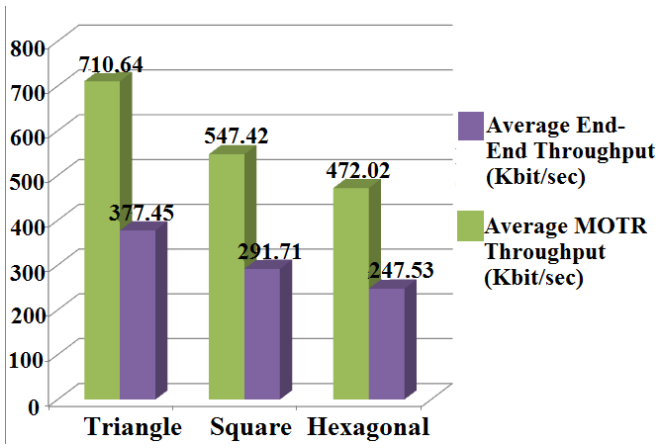

Fig. 18: Comparison of throughput in different deployments

can assume to possess sufficient computing power and energy at the MRs.

4) Performance Analysis

Results from [21] show how the throughput drops in an ad hoc network as the number of hops increase while using TCP and UDP protocols respectively. The number of hops in an ad-hoc network from a source to a destination is the key

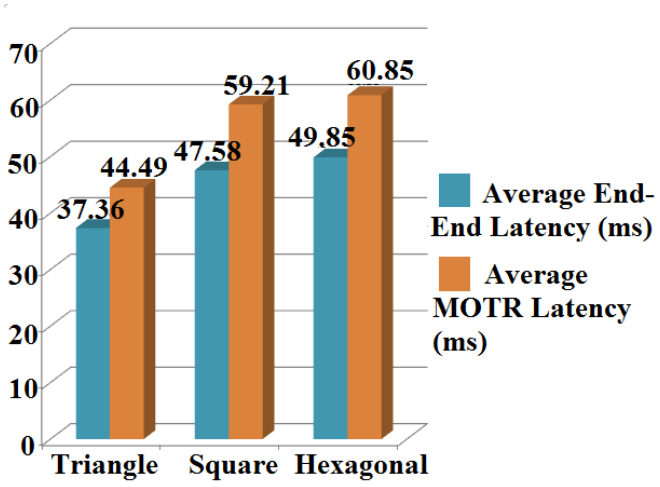

Fig. 19: Comparison of Latency in different deployments

parameter that determines the performance of that path. Using this fact, the added cost of redundancy in a Virtual Ring created in MOTR, can be analyzed in terms of the number of hops. For example, in Figure 15, it can be observed that Node C can receive data from the MR in three hops which would be the same in an end-to-end case. The up-link data sent by Node C to the MR, has to travel five hops to reach its destination. Hence, the redundancy only affects the up-link packets sent to the MR. In a WMN, all the Clients join the network to gain Internet access. So, most of the traffic in a WMN can be assumed to be to the Internet. In the case of Internet traffic, it's a well observed fact that the most of the packets are down-link packets. For the Internet traffic, the ratio between down-link and up-link packets is found to be as high as 10:1. The authors, in [22] have investigated the traffic of smart-phones using Internet. They have found this ratio to be 6:1 in case of modern smart-phones. Internet service providers use a safer ratio of 4:1 while allocating bandwidth to customers for a mixed traffic scenario.

Using this safe ratio of 4:1, we can actually calculate effective hop count in case of using MOTR scheme for the Internet traffic in a WMN. As for every 4 down-link packets one up-link packet is sent, we can allocate weights of 80% to the down-link hop count and 20% to the up-link count. Adding these two values gives us the effective hop count traversed by the packets during a data access session at a particular node in a virtual ring. For example, in Figure 15, we can calculate these weighted hop count values for each node and use those as effective values in case of end-to-end ad-hoc links for further analysis, which will be as follows:

$W_A = 0.2 \times 7 + 0.8 \times 1 = 2.2$ *hops*
$W_B = 0.2 \times 6 + 0.8 \times 2 = 2$ *hops*
$W_C = 0.2 \times 5 + 0.8 \times 3 = 3.4$ *hops*
$W_D = 0.2 \times 4 + 0.8 \times 4 = 4$ *hops*

5) Simulation Results

For simulation purposes we have used regular deployments of MCs around each MR. The MR being at the center. Simulation has been performed at a Lower Level Network entity set for the evaluation purpose. Three regular deployments, Triangle, Square and Hexagon have been used for simulation. C + + is used to code and run the simulation for the simulation results. Distance between each entity has been set to 500 meters. The values in the Table IV have been used as seed values for the end-to-end throughput and latency values in a WMN. Effective (Weighted) hop counts are

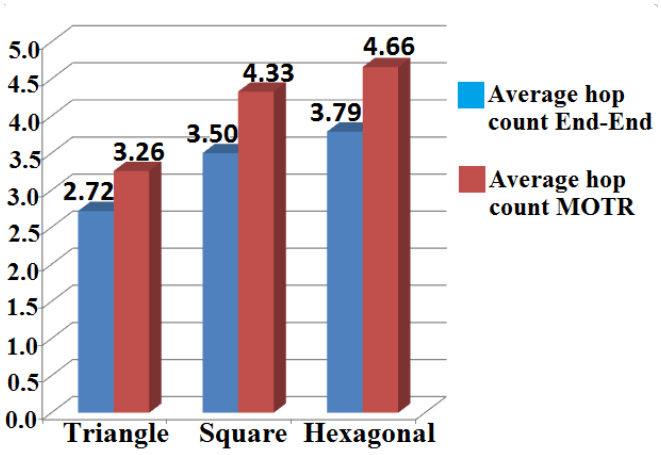

Fig. 20: Comparison of hop counts in different deployments

computed for each MC and this value has been used to compute the Throughput and Latency values as described in the previous section. Any fractional value of the Effective hop count has been rounded off to the next higher integer in calculating the maximal value of redundancy cost (privacy). Furthermore, the Average throughput and latency values have been computed in every different deployment scenario for the subgraph. The comparisons in form of several graphs are included in figures 17 - 20. Number of MCs used are 47 with each MR. A total of 48 wireless entities are used for the simulation.

MOTR results are compared for end-to-end routing in each regular deployment scenario. The results are also compared with each other amongst different regular deployments.

VI. CONCLUSION

We have proposed a scheme that enhances our earlier published PBS scheme by integrating MOTR protocol for privacy. The new scheme provides superior security and privacy at a low cost. Dynamic network management capabilities are also provided like key revocation. New nodes can join the network and old ones leaving without compromising the safety of the entire network. The WMN is secured against all the three major types of attacks inside, outside and intersection attacks. The network is also perfectly secure against a Global Attacker. Analytical and simulation results bolster the superiority of our multilevel scheme above existing protocols. The goal of privacy is achieved even in a dynamic public domain where the traffic traverses through internet.

ACKNOWLEDGMENT

The authors would like to thank Center of Distributed and Mobile Computing Lab (CDMC) at the University of Cincinnati and its members for technical support which made this work possible.

REFERENCES

[1] N. Nandiraju, D. Nandiraju, L. Santhanam, B. He, J. Wang, and D. Agrawal, "Wireless mesh networks: Current challenges and future directions of web-in-the-sky," *Wireless Communications, IEEE*, vol. 14, pp. 79 –89, august 2007.

[2] Y. Zhou and Y. Fang, "Security of ieee 802.16 in mesh mode," in *Military Communications Conference, 2006. MILCOM 2006. IEEE*, pp. 1 –6, oct. 2006.

[3] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM conference on Computer and communications security*, CCS '02, (New York, NY, USA), pp. 41–47, ACM, 2002.

[4] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Security and Privacy, 2003. Proceedings. 2003 Symposium on*, pp. 197 – 213, May 2003.

[5] R. Blom, "An optimal class of symmetric key generation systems," in *Proc. of the EUROCRYPT 84 workshop on Advances in cryptology: theory and application of cryptographic techniques*, (New York, NY, USA), pp. 335–338, Springer-Verlag New York, Inc., 1985.

[6] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," in *CRYPTO*, pp. 471–486, 1992.

[7] A. Gaur, A. Prakash, S. Joshi, and D. P. Agrawal, "Polynomial based scheme (pbs) for establishing authentic associations in wireless mesh networks," *Journal of Parallel and Distributed Computing*, vol. 70, no. 4, pp. 338 – 343, 2010.

[8] C. Ozturk, Y. Zhang, and W. Trappe, "Source-location privacy in energy-constrained sensor network routing," in *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, SASN '04, (New York, NY, USA), pp. 88–93, ACM, 2004.

[9] R. Li, L. Pang, Q. Pei, and G. Xiao, "Anonymous communication in wireless mesh network," in *International Conference on Computational Intelligence and Security, 2009. CIS '09,* vol. 2, pp. 416 –420, dec. 2009.

[10] M. Shao, Y. Yang, S. Zhu, and G. Cao, "Towards statistically strong source anonymity for sensor networks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pp. 51–55, April 2008.

[11] M. G. Reed, P. F. Syverson, and D. M. Goldschlag, "Onion routing network for securely moving data through communication networks," U.S. Patent 6 266 704, Jul 24, 2001.

[12] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk, "Enhancing Source-Location Privacy in Sensor Network Routing," *The 25th International Conference on Distributed Computing Systems (ICDCS)*, June 2005.

[13] X. Wu, N. Li, "Achieving privacy in mesh networks," in *Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks*, SASN '06, (New York, NY, USA), pp. 13– 22, ACM, 2006.

[14] Y. Cheng, M. Malik, B. Xie, and D. P. Agrawal, "Enhanced approach for random key pre-distribution in wireless sensor networks," in *Proceedings of International Conference on Communication, Networking and Information Technology*, 2008.

[15] Y. Yang, S. Zhu, G. Cao, and T. LaPorta, "An active global attack model for sensor source location privacy: Analysis and countermeasures," in *Security and Privacy in Communication Networks* (Y. Chen, T. Dimitriou, and J. Zhou, eds.), vol. 19 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 373–393, Springer Berlin Heidelberg, 2009.

[16] Y. Cheng and D. Agrawal, "Efficient pairwise key establishment and management in static wireless sensor networks," in *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on*, pp. 7 pp. –550, Nov. 2005.

[17] Wikipedia, "Spanning tree — wikipedia, the free encyclopedia," 2012. [Online; accessed 22-May-2012].

[18] Wikipedia, "Loop-erased random walk — wikipedia, the free encyclopedia," 2011. [Online; accessed 22-May-2012].

[19] J. G. Propp, D. B. Wilson, "How to get a perfectly random sample from a generic markov chain and generate a random spanning tree of a directed graph," *Journal of Algorithms*, vol. 27, no. 2, pp. 170 – 217, 1998.

[20] K. Mehta, D. Liu, and M. Wright, "Protecting location privacy in sensor networks against a global eavesdropper," *Mobile Computing, IEEE Transactions on*, vol. 11, pp. 320 –336, Feb. 2012.

[21] S. Bansal, R. Shorey, and A. Kherani, "Performance of tcp and udp protocols in multi-hop multi-rate wireless networks," in *WCNC*, p. 231, 2004.

[22] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin, "A first look at traffic on smartphones," in *Proceedings of the 10th annual conference on Internet measurement*, IMC '10, (New York, NY, USA), pp. 281–287, ACM, 2010.

Abhinav Prakash is a PhD student in the Computer Science Department at the University of Cincinnati. He is working for his doctorate under the guidance of Dr. Dharma P. Agrawal at the Center for Development of Mobile Computing (CDMC) Lab a part of Dept of Computer Science at University of Cincinnati. He has bachelors in Computer Science from Banaras Hindu University in India and a Masters of Science in Computer Science from University of Cincinnati. His research interests include distributed systems, security in sensor and mesh networks, and applications of parallel computing. professional societies other than the IEEE. Finally, list any awards and work for IEEE committees and publications. If a photograph is provided, the biography will be indented around it. The photograph is placed at the top left of the biography. Personal hobbies will be deleted from the biography.

Amit Gaur completed his masters in Computer Science from University of Cincinnati, Ohio in 2010. Currently, he is working as Systems Analyst/Programmer at Northwestern University Biomedical Informatics Center, Northwestern University. His research interests include Implementation of Security in Wireless Sensor Networks and Wireless Mesh Networks with emphasis on key predistribution schemes.

Dharma P. Agrawal is the Ohio Board of Regents Distinguished Professor and the founding director for the Center for Distributed and Mobile Computing in the School of Computing Sciences and Informatics, University of Cincinnati, OH. He is a coauthor of textbooks on Introduction to Wireless and Mobile Systems, 3rd edition and Ad hoc and Sensor Networks, 2nd edition. His recent research interests include resource allocation and security in mesh networks, efficient deployment and security in sensor networks, use of Femto cells, and heterogeneous wireless networks. He has six approved patents, two personal pending patents and twenty four patent filings in the area of wireless cellular networks. He has been the Program Chair and General Chair for numerous international conferences and meetings. He is a Fellow IEEE 1987; ACM 1998; AAAS 2003; WIF 2004, and Charter Fellow, National Academy of Inventors, 2012 and is recipient of the IEEE-CS 2008 Harry H. Goode Memorial Award.