Secure Anonymous Routing for MANETs Using Distributed Dynamic Random Path Selection

Vakul Mohanty[1], Dhaval Moliya[1], Chittaranjan Hota[1] and Muttukrishnan Rajarajan[2]

[1] Computer Sc. & Info Systems Group, Birla Institute of Technology and Science,PilaniHyderabad Campus, Hyderabad, Andhra Pradesh, INDIA
[2]School of Engineering & Mathematical Sciences, City University, LONDON
{vakul.mohanty, moliyadhaval}@gmail.com, hota@bits-hyderabad.ac.in, r.muttukrishnan@city.ac.uk

Abstract. Most of the MANET security research has so far focused on providing routing security and confidentiality to the data packets, but less has been done to ensure privacy and anonymity of the communicating entities. In this paper, we propose a routing protocol which ensures anonymity, privacy of the user. This is achieved by randomly selecting next hop at each intermediate. This protocol also provides data security using public key ciphers. The protocol is simulated using in-house simulator written in C with OpenSSL crypto APIs. The robustness of our protocol is evaluated against known security attacks.

Keywords: Anonymity, routing, ad hoc networks, mobile, public key

1 Introduction

Mobile Ad Hoc Networks (MANETs) are autonomous collection of mobile nodes without any fixed infrastructure that communicate over relatively bandwidth constrained wireless links, establishing dynamic communication. The nodes in a MANET may change its' position, adjust transmission and reception parameters causing links to be broken and re-established. A malicious node or an attacker can easily eavesdrop into the wireless channels and infer communication. A malicious node may even drop packets it had otherwise agreed to forward earlier. It may even go the extent of creating denial of service, exploited by injecting large number of unwanted packets into the network. So far, researchers in MANETs have generally studied the routing problem in a non-adversarial network setting, assuming a trusted environment; relatively little research has been done in a more realistic setting in which an adversary may attempt to disrupt the communication.

In this paper we present an anonymous routing protocol for a MANET. The protocol seeks to achieve anonymity with the minimal use of encryption and nullify the requirement of padding of data packet to prevent traffic analysis. In the protocol the next hop is dynamically selected by the router. This makes traffic analysis for a malicious router difficult as the traffic flow is erratic and confuses the adversary.

The rest of this paper is organized as follows: we present related work in Section 2; in Section 3, we present system model of our approach; and using this model, in Section 4, we present our Anonymous routing algorithm; in Section 5 & 6, we present the simulation results and analysis of our protocol respectively; finally, in Section 7, we summarize our work and point out several future research directions.

2. Related Work

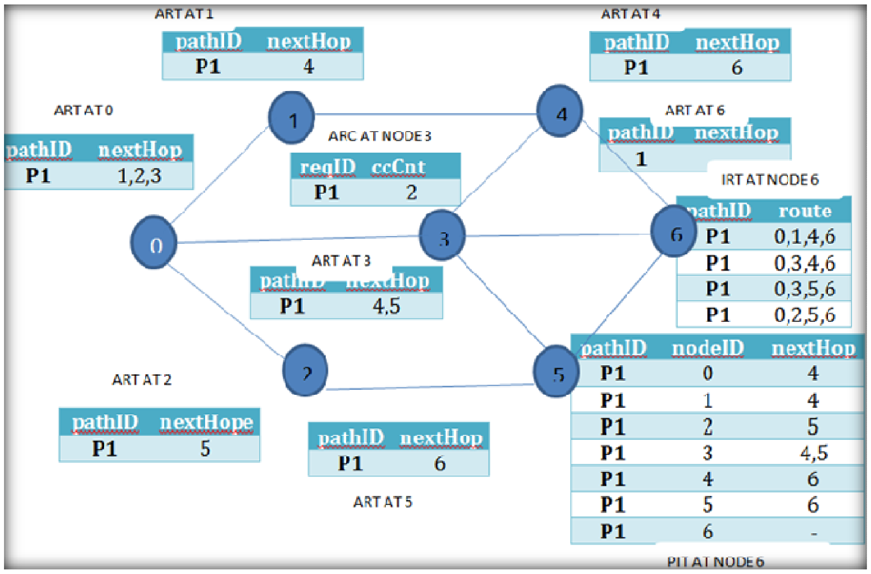
Due to the nature of wireless environment and unavailability of fixed infrastructure, achieving security in MANET routing is a complex task. Onion routing [1,6] uses multiple layers of encryptions wrapped around the message. Each router in the path of the onion receives a message, performs a set of cryptographic operations on the message and then forwards it. Each router uncovers a layer of encryption using its private key, this allows it to access routing instructions for the next router. This process continues until the message reaches the last router. Papadimitriou and Haas [2] proposed a secure routing protocol for MANETs using a security association between source and destination to validate the integrity of a discovered route. Sanzgiri et al. [3] have proposed cryptographic ways to secure routing in MANETs wherein every intermediate node verifies the integrity of the message and then forwards it to the next node. Certificates are used by source and destination nodes to get the public key of each other. ASR [4] uses anonymous virtual circuit in routing and data forwarding where each node does not know its immediate upstream nodes and immediate downstream nodes. Using a special anonymous signaling procedure, the node only knows the physical presence of neighboring ad hoc nodes. The session key of the route between every pair of the intermediate nodes is determined when a node forwards reply packet to its upstream nodes. Although the above mentioned anonymous routing techniques can provide a certain level of anonymity, an external adversary can still monitor the transmitted packets to identify the communication peers [5].

3. System Model

We explain here the notations, assumptions and the system model. An example of our approach is shown in fig. 1.As depicted in fig. 1, every node in the network maintains ART and ARC. Destination maintains PIT and IRT as well. Source node starts with the route discovery message (routeRequest) by flooding it to all neighbors.Request, id is embedded in it.

Notations used:

S	:Sender	**R**	: Receiver
M	:Message	**D**	: Data
X	:Intermediate node	**E**	: Encrypt function
ccCount	:Criss-cross count	**ccTimer**:	Criss-cross timer
PU_N	:Public key of N	**PR_R**	: Private Key of N
D	:Decrypt function	**ccTable**	: Criss cross Timer Table
ART	:Anonymous Routing Table		
IRT	:Intermediate Routing Table<pathID, path_of_message>		
PIT	:Path Info Table <pathID, nodeID, nextHop>		
▯	:Set inclusion, modeled as appending at the end of set (array)		
ARC	:Anonymous Routing Cache <reqID, ccCount>		
exists(x,z)	:returns true if table z has record mapped to x, otherwise false.		
getCnt(x)	: returns ccCount value from the record <x, ccCount> of ARC, if no such record found then return false.		
setCnt(x, y)	: sets ccCount value from the record <x, ccCount> of ARC to y.		
expired(x)	:returns true if timer mapped to x is expired else false.		



ART AT 1

pathID	nextHop
P1	4

ART AT 4

pathID	nextHop
P1	6

ART AT 0

pathID	nextHop
P1	1,2,3

ARC AT NODE 3

reqID	ccCnt
P1	2

ART AT 6

pathID	nextHop
1	

IRT AT NODE 6

pathID	route
P1	0,1,4,6
P1	0,3,4,6
P1	0,3,5,6
P1	0,2,5,6

ART AT 3

pathID	nextHop
P1	4,5

ART AT 2

pathID	nextHope
P1	5

ART AT 5

pathID	nextHop
P1	6

PIT AT NODE 6

pathID	nodeID	nextHop
P1	0	4
P1	1	4
P1	2	5
P1	3	4,5
P1	4	6
P1	5	6
P1	6	-

Fig.1. An Example of Secure Anonymous Routing Protocol.

Data portion D of the message contains pathID, Source(S), Destination (D) and Nonce_s which is encrypted using the public key of destination (PU_D). Message also contains I, set of all nodes traversed by routeRequest message to reach destination, and PU_D.Every entry in I is encrypted using PU_D.

ccCnt indicates the number of more routeRequest messages with same request id that can be flooded by the same node. Initially assigned value to ccCnt is a parameter set by network administrator, subject to tuning. Upon receiving routeRequest message with given reqID first time, node makes an entry in ARC(as shown in fig 1, at node 3), inserting reqID and ccCnt. For subsequent receipts of routeRequest message with same reqID, node checks whether value of ccCnt is zero or not. If zero then ccCnt limit is reached and packet is discarded there itself. If not zero then ccCnt is decremented by 1 and message is forwarded to neighbors by appending its id(encrypted with PU_D) in I field of the message.

Upon receiving the route request message, destination node takes action as explained in previous paragraph with following additional steps: it sets ccTimer for given reqID. ccTimer also acts as a threshold like ccCnt, but it is used to filter optimal paths. It is also subject to tuning by network administrator. Destination decrypts D and I part of message using its private key(PR_D) to extract NONCE_S, pathID and all en-route nodes which are entered into IRT (i.e. <P1, <0,1,4,6>> of IRT at node 6 in fig. 1). There is one to one mapping between pathID and reqID. Whenever ccCnt becomes zero or ccTimer expires for a given reqID, node creates PIT from IRT to be used for updating routing tables of en-route nodes. As depicted in fig. 1, for IRT entry <P1, < 0, 1, 4, 6>>, it updates PIT entries as <P1, 0, <1>>, <P1, 1, <4>>, <P1, 4, <6>>, <P1, 6, <>>. For <P1, < 0, 3, 4, 6>>, it updates PIT as <P1, 0, <1, 3>>, <P1, 3, <4>>, <P1, 4, <6>>, <P1, 6, <>>. This is used to construct routeReply messages, composed of routing table updates of en-route nodes. Destination node encrypts these PIT entries with the public keys of en-route nodes, in the sequence marked in the routeRequest message and onion routing [1, 6] is used to forward these updates to en-route nodes. $NONCE_R$, $F(NONCE_S)$ are also added to message encrypted using public key of Source.

Upon receiving the routeReply message, the intermediate node removes outermost layer from the onion, does appropriate cryptographic operations on it, updates its ART from the update received, and forwards the message to the next hop. Upon receiving the routeReply message, the source node updates it's ART as explained for en-route nodes. It receives $NONCE_R$ and $F(NONCE_S)$, which are used for authentication and preventing the replay attack. For regular data transfer, source uses the pathID, and selects the next hop randomly from its ART. Every en-route node also does the same for selection of the next hop.

Here we assume that any node leaving the network does not cause the partition in the MANET. Every node(X) sends a beacon to its neighbor, and updates the status of neighbors depending upon the reply. If any node $N_L$discovers change in the topology of network then it searches <pathID, Z> in ART such that $N_L \in Z$. If such entry is found then it sends the update message to all nodes in Z and removes N_L from Z. After removing the entry, if Z is empty then node floods the route invalidate message with corresponding pathID. Upon receiving the update message, node updates its ART. In case the node receiving the invalidate message is the one that started the communication with the corresponding pathID, it re-initiates route discovery.

4. Proposed Algorithm

4.1 Path Discovery Phase

```
Source initiates with routeRequest message<reqID, E($PU_{R,D}$), I> ;D=, <pathID, sourceID,
destinationID, $NONCE_S$>, I={E($PU_R$, S)} by sending to all neighbors.

X ($\neq$R), an intermediate node receives routeRequest<reqID, E($PU_R$,D), I>message:
if ( exists(reqID, ARC) ▯ getCnt(reqID) $\neq$ 0)then
    setCnt(reqID, getCnt(reqID) - 1)          /* decrement the ccCount */
    I$\leftarrow$I$\cup${ E($PU_R$, X)}                             /* Append ID to the message*/
    forward  <reqID, E($PU_R$, D), I>  to neighbors except the one from it received.
elif(exists(reqID, ARC) ▯ getCnt(reqID) = 0)then
    Discard routeRequest message as ccCnt limit reached.
else
    ARC $\leftarrow$ARC▯ {<reqID, ccCntUL>}        /* Make entry in ARC */
    I$\leftarrow$I$\cup${ E($PU_R$, X)}                             /* append ID in message */
    forward<reqID, E($PU_R$, D), I> to neighbors.
endif
Send acknowledgement to the node (sourceID) from which message is received.

R receives routeRequest message $M_{RQ}$<reqID, sourceID, destinationID, E($PU_R$, D), I>:
Decrypt each entry of I private key, store decrypted values in I.
if ( exists(reqID, ARC) ▯ expired($ccTimer_{reqID}$) )then
            Discard routeRequest message.    /* Timer Expired */
elif ( exists(reqID, ARC) ▯ $\neg$expired($ccTimer_{reqID}$) ▯ getCnt(reqID) = 0))then
```

```
        Discard routeRequest message.    /* Criss-cross count limit reached */
elif ( exists(reqID, ARC)  ¬expired($ccTimer_{reqID}$)  getCnt(reqID) $\neq$ 0))then
        setCnt ( reqID, getCnt(reqID) - 1)
        pathID$\leftarrow$ D($PR_R$, E($PU_R$, D))
        IRT $\leftarrow$IRT$\cup$ {<pathID, I $\cup$ {R} >}
else    (¬exists(reqID, ARC), ARC)        /* No entry found in ARC for reqID */
        ccTable$\leftarrow$ccTable$\cup$ {<pathID, $ccTimer_{Mrq}$>}        /* Set ccTimer*/
        ARC $\leftarrow$ARC$\cup$ {<reqID, 5>}
        pathID$\leftarrow$ D($PR_R$, E($PU_R$, D)
        IRT $\leftarrow$IRT$\cup$ {<pathID, I $\cup$ {R} >}
endif
```

4.2 Construction of Routing Table entries for intermediate nodes

```
/*Process entries in IRT with reqID for with ccCnt is zero or ccTimer is expired*/
for each <reqID, I> in IRT  do
        for each $x_i$ in I;$x_i \neq$R,do
                if exists(<pathID, $x_i$,  Z>, PIT)then
                        Update Z$\leftarrow$Z$\cup${ $x_{i+1}$} in PIT
                else
                        PIT $\leftarrow$PIT$\cup$ {<pathID,$x_i$, { $x_{i+1}$}> }
                endif
        end for
end for
```

4.3 Updating ART of intermediate nodes

```
Constructing and sending Reply Message:
for each <pathID, I>in IRT do
        I'= 					/* initialize I' as Null*/
        for each $x_i$ in I, i=n…1 do			/* Reverse the path for reply message */
                I'=I' $\cup$ {$x_i$}
        endfor
        temp=< $NONCE_R$ , F($NONCE_S$)>
        for each $x_i$ in I', $x_i \neq$R do
                Search <pathID, $x_i$ , Z> in PIT
                if  i=1 then				/* for source node's case */
                        msg = msg + <S , E($PUx_{i,}$<<pathID, $x_i$ , Z>, temp>) >
                else    msg = msg + <$x_{i-1}$ , E($PUx_{i,}$<pathID, $x_i$ , Z>) >
                endif
        end for
Send routeReply message $M_{RP}$ <R, $x_{n-1}$, msg> to $x_{n-1}$       /*<source, to, msg_data> */
end for

X receives the routeReply message $M_{RP}$<Y, X, msg>:
/* extract the routing info sent by the destination and update ART */
<<pathID, X , Z >, nextHop, E($PU_{nextHop}$ , msg) > = D($PR_{X,}$ msg)
ART = ART $\cup$ {<pathID,  Z>}				/*Update routing table*/
if X=S then
        <pathID, $NONCE_R$ , F($NONCE_S$)> = D($PR_{S,}$ msg)
        Send F($NONCE_R$) to the destination.
else
        forward$M_{RP}$<X, nextHop, msg>
endif
```

4.4 Data Communication Phase

Source-destination pair exchanges session key for regular data transfer. Source sends message with pathID prepended to the message. Every intermediate node will choose the next hop dynamically from its ART corresponding to the pathID in the message. We have employed acknowledgement mechanism for detection of passive nodes.

5. Simulation Results

We have written our simulator using C in UNIX. All cryptographic operations are performed using OpenSSL Crypto API. MANET is constructed using 50 nodes, initially uniformly distributed. Source destination pairs are chosen randomly. Mobility of nodes is random, with constant speed. Once node becomes immobile, it waits there for fixed time. Maximum number of communicating pairs in MANET at a given time is assumed to be 20, chosen randomly. We use cc_cnt, and cc_timer metrics as global tunable parameters which are set by network administrator. As depicted from the cc_cnt vs delay graph in fig. 2, by increasing the value of ccCnt, the number of paths discovered is more. However few of these paths might be longer ones. So the delay incurred on an average to reach the destination also increases.

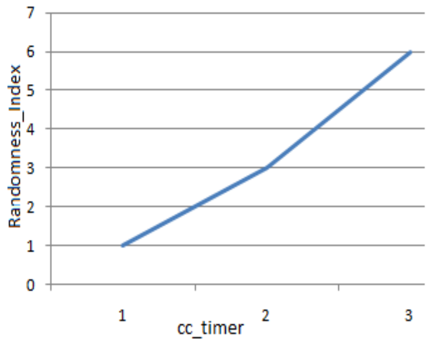


Fig. 2. cc_timer vs Randomness_index

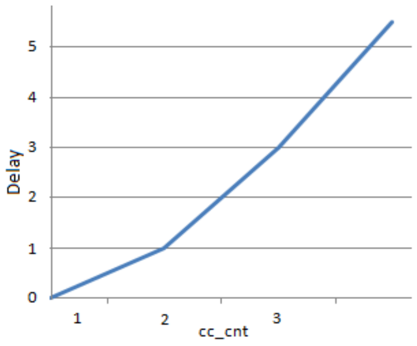


Fig. 3. cc_cnt vs Delay

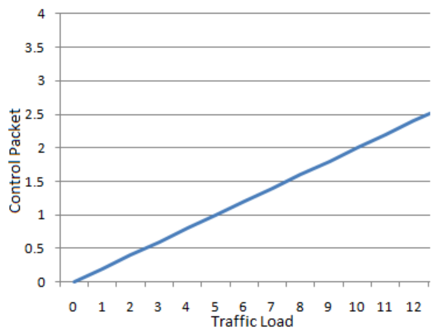


Fig. 4. Traffic Load vs Control Packets

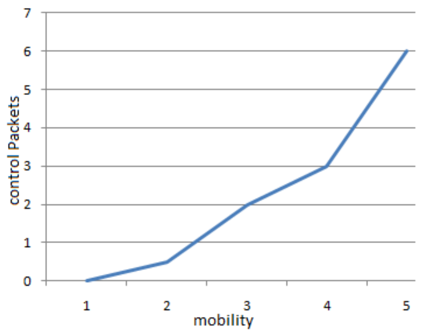


Fig. 5. Mobility vs Control Packets

Fig. 3 shows the cc_timer vs randomness_index, the average number of nodes in each intermediate node's ART, for a given pathID. The linear increase in this randomness_index guarantees higher anonymity because of the fact that more number of paths is now available to make traffic analysis increasingly difficult in the MANET. Fig. 4 shows the mobility vs control packet and fig. 5 shows traffic load vs control packets. Both the figures show that increase in mobility increases number of the control packets. This is because of flooding many packets that include path invalidation, path update and route rediscovery messages.

6. Simulation Analysis

6.1 Anonymity Analysis

Identity Privacy: In our protocol, the identities of source and destination are known to only two communicating parties, as we are using them only in the route request message and with encryption, thereby not revealing them to intermediate nodes. Hence identity privacy is ensured.

Route Anonymity: In our protocol, no adversary can trace a flow of packet because of random selection of next hop and thereby leading to dynamic path selection. Any adversary on the route has no information about the path other than the next hop. As we have employed fixed size padding, we can introduce several dummy packets and reshuffling of actual packets in the buffer to eliminate the possibility of temporal analysis as defined in [12]. Thus all the requirements of route anonymity are satisfied.

6.2 Possible attacks

Route Rediscovery Attack: One possible attack is that adversaries send fake route update or route invalidate packets to fool the intermediate nodes or source to begin route rediscovery process. In our protocol, only the nodes whose routing table has entry for the node leaving the network, can send the route invalidate, route rediscovery or route update messages whichever applicable as explained in the algorithm. So our proposed protocol is less vulnerable to the route rediscovery attack.

Selfish Nodes or Byzantine nodes: Byzantine nodes can intercept packets, create routing loops, selectively drop packets, or purposefully delay packets. Our protocol uses the acknowledgement mechanism. If any node is dropping the packet then acknowledgement will not be sent to sender. Even in presence of live communication link, if node is dropping packets then it can be detected as selfish node. And as we are choosing next hope dynamically at any intermediate node for routing, we can exclude this selfish node from the ART.

6.3 Cryptographic Overhead

In our protocol, we use cryptosystem of the form onion only for path discovery. For data communication, data is encrypted by source with the destinations' public key, i.e. end to end encryption; onion routing is not used here. So there is not much cryptographic overhead involved for normal data communication phase that leads to computational advantage.

7. Conclusions

This has paper has proposed a new routing protocol in mobile ad hoc networks with anonymity and provable security. We have stressed upon the anonymity, which is becoming one of the most important aspect in securing the next generation mobile ad hoc networks. The developed protocol has been evaluated with respect to anonymity and known security threats. Simulation results give the performance of our protocol. Our future work will aim at overcoming Distributed Denial of Service (DDoS) attack, and estimating the cryptographic computation overhead in this type of environment. We will also focus on improving security by adopting strong peer to peer authentication in the route discovery phase using extensive simulations.

References

1. Reed, M., Syverson, P., and Goldschlag, D. Anonymous connections and Onion Routing. IEEE J. Selected Areas in Commun. 16, 4 (May 1998), 482-494.
2. Papadimitratos and Haas: Secure Routing for Mobile Ad hoc Networks. SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002), San Antonio, TX, January 27-31, 2002.
3. K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer: A Secure Routing Protocol for Ad Hoc Networks, In Proceedings of 2002 IEEE International Conference on Network Protocols (ICNP). November 2002.
4. B. Zhu, Z. Van et al. anonymious secure routing in mobile ad hoc networks in 29th IEEE International conference on local computer networks LCN'04, 2004 pp 102-108.
5. D. Huang. Traffic analysis-based unlinkability measure for ieee 802.11b-based communication systems. In Proceedings of ACM Workshop on Wireless Security (WiSe), 2006.
6. Jian Ren, Jie Wu. Survey On Anonymous Communications In Computer Networks. Comput. Commun. (2010), Vol. 33, No. 4 pp. 420-431.