SMART: A Secure Multipath Anonymous Routing Technique

Prateek Jain & **Rupsha Bagchi**

Manipal Institute of Technology, Manipal University, Manipal, 576104, Karnataka, India
E-mail : jainprateek_90@yahoo.com & rupsha.bagchi@gmail.com

Abstract **-** Multipath routing for mobile Ad hoc networks is a technique of concurrent management and utilization of multiple paths for transmitting distributed data evenly across the nodes instead of routing all the traffic along a single path, potentially resulting in longer lifetime along with the benefits of better transmission performance, fault tolerance, increased bandwidth and improved security. In this paper, a secure multipath anonymous routing protocol (abbreviated as SMART) has been proposed. SMART uses non cryptographic ways to help the source find the routes to the destination and dynamic onion routing to intimate the source about these routes. It includes a mechanism of key caching and defines a minimum battery protection threshold for each node to help increase the network lifetime to some extent. In effect SMART is an attempt to strike a balance between the anonymity, security and energy consumption in a network.

***Index Terms* -** *Multipath routing, Dynamic onion routing, Key caching, Minimum threshold*

I. INTRODUCTION

Mobile Adhoc Network (MANET) is a collection of independent mobile nodes that can communicate to each other via radio waves. These networks can work at any place without the help of any infrastructure. The dynamical nature of the network topology increases the challenges of the design of routing protocols in such ad hoc networks.

The nodes in these networks usually have a limited storage and low computational capabilities. They heavily depend on other nodes and resources for data access and information processing. But MANETs are much more vulnerable to attacks than wired network. This is because of the reasons like open medium where eavesdropping is easier than in wired network. Also dynamically changing network topology, implying that mobile nodes come and go from the network, may allow any malicious node to join the network without being detected. Thus, a reliable network topology must be assured through efficient, secure and anonymous routing protocols for these Ad Hoc networks. Routing strategies play an important role in the minimization of energy consumption during the data transmission.[1]Finding a new route on path failure introduces delay along with the possibility of the disclosure of the identity of the source node to passive adversaries. As each radio terminal in the network is usually powered by energy limited power source, an energy efficient multipath routing protocol can be used to overcome this problem. Routing protocols for MANET can be classified as proactive algorithms, reactive (on-demand) algorithms, flow-oriented algorithms, and others. Multipath routing establishes multiple paths between the source-destination pair. Classical multipath routing has been explored for two reasons. The first for load balancing; the traffic between the source and destination is split across multiple disjoint paths. The second use is to increase likelihood of reliable data delivery.

In this paper we describe SMART, which to the best of our knowledge, is the first of its kind. It is an efficient way in which we can achieve anonymity in multipath routing along with economical utilization of energy but at the same time compromising on the latency factor to a certain extent. Rest of the paper is organized as follows. Section II has related work; Section III describes the adversary models, our assumptions, the essential idea of anonymous and secure routing in SMART and the detailed implementation of routing protocol. Section IV gives performance analysis and Section V gives some concluding remarks on our paper.

II. RELATED WORK

In order to provide privacy and protection, plenty of work about anonymous ad hoc routing protocol has been researched. In ASR [3] nodes forwarding a RREQ message keep state information about this RREQ. Later on, they use this state information to decide whether and to whom they have to forward this RREQ. In ANODR [4] each intermediate node adds sufficient information to an onion that is copied by the destination into the RREP message. Nodes can recognize RREP messages they need to forward using this onion. In both [3] and [4], for every RREQ message each intermediate node generates a new public or secret key pair which helps in providing anonymity. Some protocols like SDAR [5] and MASK [6] partially violate the anonymity requisite as they use real identities of participating nodes in order to achieve improved performance. E.g. in [5] nodes use real identities but they are encrypted and known only to sender and receiver, which guarantees the anonymity of intermediate nodes to observers but not to the sender and receiver. In [5], it is assumed that the identity of every node in a broadcast is in plaintext. Every intermediate node has to perform a public key decryption and encryption for every RREQ message it forwards. This may lead to inundation of the network, thereby resulting in congestion. [6] doesn't use a trapdoor unlike [4] but at the same time provides conditional anonymity by exposing the destination's identifier in ARREQ in order to improve routing efficiency. [6] relies on synchronization keys and pseudonyms between nodes. These approaches require all network nodes to perform expensive cryptographic operation in the forward path (broadcasting RREQ message), which results in wasting both computation power and bandwidth, as only a few nodes will be selected as forwarding nodes. ARM [8] improves upon these by using a dynamic index as the index changes on a per-request basis for shared key management and used limited flooding instead of dummy traffic and local mixing of messages to prevent traffic analysis.

CMMBCR [7], a hybrid energy saving protocol considers residual energy of nodes as the metric for route establishment to improve the lifetime of a node. Although several proactive multipath algorithms have been developed, they do not take conservation of power into account and end up generating excessive overhead due to their proactive nature. Their scheme also does not offer any security since the identity of the node is encompassed in its packets.

II. PROTOCOL

The main aim of the proposed routing protocol is to improve upon drawbacks of the anonymous protocol mentioned above, that is to accomplish anonymity, data integrity and security along with reduced energy consumption. It supports reliability by providing node disjoint paths and provides stability by distributing the burden of routing and congestion control. To send data anonymously to a node, a sender node has to discover and establish a reliable and anonymous path that links the two nodes. Both the route discovery and establishment process should be carried out securely and without jeopardizing the anonymity of the communicating nodes. The process is divided into three phases: the *route request* phase, the *route reply* phase and the *data transmission* phase. Distributed information gathering about intermediate nodes that can be used along an anonymous path is carried out during the r*oute request* phase, while passing this information to the source node takes place during the *route reply* phase. The official data exchange is processed during the *data transmission* phase after the construction of the route.

A. Adversary model

Adversaries in a network can be classified into two categories. An external adversary is a wireless link intruder that can intercept all traffic transmitted on all the connections in the network. An internal adversary is a node intruder that can compromise legitimate network members. Both internal and external adversaries exist in the network and rely on trapdoor instead of node identities. Without knowing the node identities, the adversary has no means to break a mobile node's identity. It is assumed that adversaries have unbounded eavesdropping capability but bounded computation and node intrusion capability, as in [6]. Anonymity of network layer alone has been taken into consideration. Attacks in application or physical layer are beyond the scope of this paper.

B. Assumptions

The initial assumptions of [8] hold true for this protocol as well, which are described subsequently. The following assumptions have been made.

- Each node in the network has a permanent unique ID which is known by all other nodes in the network.
- The source N_s and the destination node N_d share a secret key K_{SD} and a secret pseudonym in RREQ messages directed at the destination node.
- Every node has established a broadcast key with its one hop neighborhood which will be used to encrypt the RREP messages. Wireless links between the nodes are symmetric.

To enhance the efficiency of our protocol, it has also been assumed that nodes will only share secret keys and pseudonyms with a limited set of other nodes. At any point of time each node is aware of its residual energy.

C. Notations

N_i	**Node i, where N_s and N_d represent source and destination nodes**
K_{SD}	Secret key shared by the source and the destination
Nm_i, Nm_{i+1}	Two consecutive pseudonyms stored by the destination. Nm_i for current use and Nm_{i+1} for the next use.
T	Type of message: 0 = RREQ; 1=RREP; 2= data
td_D	Trapdoor descriptor for the destination node
$pub_d/priv_d$	Public keys/Private keys
(n_s, k_s)	Link identifiers
h()	Hash function used
e_thresh	Minimum threshold energy defined for each node

Table 1: Notations

D. *Route Request*

The *route request* phase allows a source node to communicate with a node by discovering and establishing routing paths to that node anonymously. This is done using a number of intermediate nodes between the source and destination. Route discovery is triggered when a node wishes to communicate with another node within its network. The source (S) generates a new asymmetric key pair $pub_d/priv_d$ and a secret key k. Next, it generates the trapdoor descriptor td_D that can only be opened by the destination node (D) which has knowledge of the secret key K_{SD}:

$$td_D = K_{SD}[D, k, priv_d], k[Nm_{SD}] .$$

The destination node recognizes its current pseudonym Nm_i and a secret key K_{SD} is shared between the source and the destination. The source now generates a pair of link identifiers (n_s, k_s) and encrypts them with the public key pub_d. It finally constructs a RREQ message in the following format, which is then broadcast to the network.

In the *route request phase*, Nm_{SD} is included in plaintext and the consecutive pseudonym Nm_{SD+1} is used to encrypt a trapdoor. Like Nm_{SD}, ME is also included as plaintext. Upon receiving the RREQ message, an intermediate node does the following things:

- Performs a search in its table (the table contains a list of pseudonyms of various source nodes) to check whether the message was meant for that node. If the corresponding pseudonym is found, the node may be the intended receiver and the next identifier Nm_{SD+1} at the node is then used to open the trapdoor. If the first part of the decryption is not equal to the unique identifier of the node then it is not the destination and once again it broadcasts the request packet. No cryptographic operations are required for a node to be able to recognize whether it is the required recipient.
- If the intermediate node is not the targeted destination then it checks whether Nm_{SD} is present in its routing table. If it is, it discards the packet else it stores (Nm_{SD}, n_i, k_i, k(Nm_{SD})) in its routing table.
- If the received packet has the value of time to live field ttl>1, then the node decrements it, and generates a random pair of link identifiers (n_i, k_i), appends these to the already received encrypted link identifiers, encrypts everything with pub_D, and broadcasts the following RREQ message :

RREQ :

[T=0,Nm_{SD}, td_D, ttl, pub_d, ME, pub_d(. . . (pub_d(n_{i-1}, k_{i-1}), n_i, k_i),padding]

- If the received value in the ttl field is 1, the intermediate node does not broadcast anything.

When an intermediate node broadcasts a RREQ packet, it compares its residual energy with the value present in the ME field. (The first node on the path discovery puts its residual energy in the ME field which is initialized to zero). If the residual energy of the node is less than the value present in ME, then it updates this field by writing its own energy value, else ME is left undisturbed. It also adds other information as mentioned along with random padding in the padding field as per [8]. If the node is the targeted destination then the RREQ packet is stored in its memory and it broadcasts the packet yet again so as to protect its identity. It looks like any normal intermediate node to a passive adversary.

E. *Route Reply*

In order to conserve the energy of the nodes and to increase the network lifetime to some extent, we use multiple routes for data delivery and thus, send information about these routes to the source in the *route reply phase*. During *route request* the destination receives a number of RREQ's from the same source through different routes which are node disjoint. It stores each of these in its memory and compares the value of the ME field with the e_thresh value defined. If the difference between values in the ME field and e_thresh is less than or equal to 1, then that path is rejected. It is because one can conclude that a node on that path is about to reach its threshold energy and die soon as it is possibly being used by other nodes in the network to forward their packets. What a node does when its residual energy equals e_thresh is described in section H.

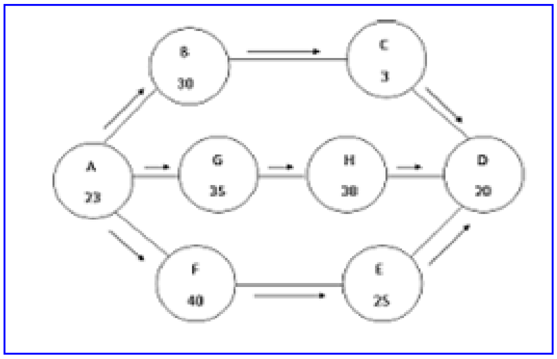

Figure 1: A part of the network with 9 nodes depicting the RREQ packets that reach the destination

For example, in figure 1 there are 3 node disjoint paths that the source can use to send data to the destination. These paths are A-B-C-D(1), A-G-H-D(2) and A-F-E-D(3). When the RREQ packets are broadcasted from the source the ME field of these packets consist of a value zero. The following tables show the values of the ME fields of the RREQ packets as they pass through different nodes.

Node	ME		Node	ME		Node	ME
A	0		A	0		A	0
B	30		G	35		F	40
C	3		H	35		E	25

(1) (2) (3)

Figure 2: The tables show the changing value of ME field of the packets for paths 1,2 and 3.

The destination receives all these three RREQ packets and stores them in its memory. It now compares the ME field with e_thresh. Let for the above example e_thresh is fixed at 2. When the destination finds that for path 1 the difference between ME and e_thresh is 1 it rejects that path for the reason mentioned above. Thus it creates RREP for only two paths namely D-H-G-A and D-E-F-A. It is assumed that there will always be at least one path in which all the nodes have remaining energy such that the difference of their energies and e_thresh is greater than 1.

For each of the valid paths found the destination decrypts the trapdoor identifier td_D and gets information about k and $priv_D$. Using this $priv_D$ it decrypts the link identifiers present in the RREQ packet. After collecting all such link identifiers of intermediate nodes, along with (ns, k_s), the destination creates a route reply onion of the form:

K_n (n_n, $k^1{}_n$, k, k_{n-1}(n_{n-1}, $k^1{}_{n-1}$, k, …k_s(n_s ,k)))

Where $k^1{}_n = h(k_{n-1})$ where h is a hash function. After the construction of the reply onions, node D generates a random value for the time to live field and unicasts them on the discovered routes. (Nm_{SD} is copied from the RREQ message and again serves as a unique identifier). The identifier and ttl field of the RREP message are encrypted with the current broadcast key of node D to hide them from a global passive adversary.

Each intermediate Node N_i which receives the RREP strips one layer of the reply onion, puts a new value in the time to live field and sends it to the next node in the route after encrypting the present header with its current broadcast key. N_i stores a pair of secret keys (k_i, $h(k_i)$) in its routing table. These keys are shared with the previous hop and the next hop respectively where $k_i = h(k_{i-1})$. Padding is applied in the padding field as in RREQ. Finally, these RREP's reach the source which it then stores in its memory. Thus, multiple node disjoint paths are successfully established between the source and the destination.

F. *Data Transmission*

Once the routes have been established, now the source prepares to forward the data. Shamir's modified (k, n) threshold scheme used in [9] has been used to maintain the security of the data being forwarded to the destination. It states the following:

Assume that data D is a secret to be exchanged between the source and destination. Then divide D into n pieces $D_1, D_2, \ldots, D_n$ such that:

a) With the knowledge of any k or more pieces, D_i, D can be computed easily.

b) The knowledge of less than k pieces will leave D completely undetermined (in the sense that all its possible values are equally likely).

c) There is only a small possibility that any k-1 participants can fabricate new pieces $D_{11}, D_{12}, \ldots, D_{1k-1}$ that deceive a kth participant. Here deceiving kth participant means that from $D_{11}, D_{12}, \ldots, D_{1k-1}$ and D_k , the secret D_1 reconstructed is legal but incorrect (i.e. D_1 is not equal to D).

The data is divided into n parts by the source node. To do so, a polynomial of degree k-1 of the form given below is required to be chosen.

$q(x)=a_0+a_1 x+\ldots+a_{k-1}x^{k-1}$

in which a0=D and evaluate

$D_1 = q(1) \ldots\ldots D_i = q(i) \ldots\ldots D_n = q(n)$.

Given any subset of k of these D_i values, the coefficients of q(x) can be found by interpolation. The values $D_1, \ldots D_n$ are computed modulo p where p is a prime larger than both D and n.Thus, the data is divided into n parts. Source node sends each part to the destination through n different routes which have already been established in the route discovery phase. If D_i is long, it can be split up into shorter blocks of bits to avoid multiprecision arithmetic operations as per Shamir[10].

After the destination node receives k random shares it can generate the entire data D by using Lagrange's interpolation(equation 1) and evaluate D=q(0).

$$f(x)=h(x)=\sum^{k}_{i=1} f(i_i)\prod^{k}_{l=1,l\neq i}((x-i_l)\div(i_i-i_l)) \bmod p \;\ldots.\; (1)$$

To make sure that there are no fake shares in the reconstruction process, we assign each share an unforgeable signature such as that proposed in [9].A signature system is said to be strongly unforgeable if the signature is existentially unforgeable and, given signatures on some message M, the adversary cannot produce a new signature on M. In order for that to happen, computation of h(M) for the entire message M is done initially. Each share x then, is of the form

$$D_x= (q(x), h(x), n)$$

If it is assumed that an adversary manages to get hold of any k-1 shares then for every share D_i it can construct only a single polynomial of degree k-1 such that q'(0)=D' and q'(i) = D_i for the given k-1 arguments. By construction, the p possible polynomials are equally likely to be generated which makes the discovery of the real data impossible.

After receipt of k shares, the destination rebuilds the data and recalculates the value of h(M) and examines whether it matches that of the original. If it does, then the shares are legitimate and so is the data. Else it indicates that the shares have been tampered with and there is an active attack going on. In this case the data generated is discarded.

This protocol has two more features which contribute in saving some energy given the computations that need to be carried out for data transmission. We use the concept of key caching and minimum threshold energy which shall be discussed about subsequently.

G. *Key Caching*

With regard to key caching, caching of the following keys is done:

- The key that the source node and the destination node share during route discovery.
- The link identifiers/secret key shared between any two one hop nodes.

After certain period of time the same keys are used instead of generating a set of new keys each time. As generating keys is expensive, the overhead of this operation can be reduced by caching. Also, if the communication takes place between the same source and destination, using the same $priv_D$ and pub_D keys after a considerable time with a certain probability (ε>0 of one out of every few establishments between the same source and destination) definitely helps in cutting down on energy consumption.

The hash function generated during data transmission phase can also be cached and reused later for some other destination. This alleviates the burden on the source as it need not compute new hash functions each time it has to send data.

H. *Minimum Threshold Energy*

Every node in the network has a minimum battery protection threshold energy defined. Unlike other routing protocols in which a node is used till the time it dies or goes out of energy, in our protocol a node whose energy level becomes equal to e_thresh will stop forwarding packets to all other nodes. It will use this left energy to perform its internal computations and also act as a source if it wants. But as mentioned it will discard any RREQ, RREP or data packets that are received on behalf of other nodes. If the environment where the network is established supports charging of the nodes (as in using batteries or solar cells etc.) then the node can again be active and accept packets when its energy level goes up the defined e_thresh.

IV. PROTOCOL ANALYSIS

A. *Anonymity*

This protocol provides a high level of sender, receiver and intermediate node anonymity. Nodes inside the network will not be able to determine whether the node they received a message from is the source of this message or forwarding it, nor will the nodes be able to examine the messages that they forward. The sender and receiver anonymity is achieved due to the secret key shared between the source and the destination which no other node would know of. Public keys in the route discovery phase are self-generated at each node on per-session basis, so that adversaries cannot link them to real identities over time. Padding and random ttl techniques have been applied to prevent nodes in the network to discover the hop distance by message coding and message volume analysis.

This protocol uses multipath routing, which diverts the data flow and makes the traffic analysis based detection more difficult. A comparison with three other protocols namely ANODR, MASK, CMMBCR has been provided in Table II. It can be observed that the proposed protocol provides anonymity for all nodes as well as the data in the network with an aim to reduce the energy consumption to a certain extent.

Characteristic	ANODR	MASK	CMMBCR	SMART
Sender Anonymity	Yes	Yes	No	Yes
Receiver Anonymity	Yes	No	No	Yes
Forwarding node anonymity	Yes	Yes	No	Yes
Key Caching	No	No	No	Yes
Residual energy	No	No	Yes	Yes

TABLE II: Comparison

A. *Message Compromising*

A share is compromised if it is relayed by a compromised node. If k shares are compromised, it implies that the message too is determined since the knowledge of k or more pieces is sufficient to construct the message. This protocol requires encrypting the shares that are transmitted. So if the attacker wants to compromise a message, enough shares (at least k) must be intercepted and then decrypted.

B. *Security Analysis*

The proposed routing protocol is secure against some of the most common passive and active attacks in MANETs like replay attack, identity spoofing, eavesdropping. In this protocol, RREQ message trapdoor contains a public key related pseudonym so that the destination is able to verify the integrity of the message. The protocol does not use real identity for routing and data transmission, however adversary can disguise as an intermediate node in the route. This protocol thwarts such type of attacks as the pseudonyms are linked to public key, and the corresponding private key is only known to the node that first announces the pseudonym. The RREP messages are encrypted in onion like structure. An adversary can insert itself in a route; nevertheless, without all keys of the entire route, it is impossible to reveal the real content of the message. In addition, the use of Shamir’s secret sharing technique renders the data secure.

V. CONCLUSION

Conventional routing protocols in wireless network are based on single path. Any event such as link breakage causing invalidation of the path results in failure of the entire routing path, has lesser reliability and consumes a lot of energy since the source node requires rediscovering a path. In order to provide efficient identity, location and route anonymity along with transfer of data securely we use multiple routes and Shamir’s (k, n) methodology to split data and achieve data security. In addition key caching is used and a lower threshold energy value (e_thresh) is defined which help strike a balance between anonymity and energy consumption. Clearly, the protocol improves reliability along with prolongation of lifetime of the network. As future work we will find the efficiency of the proposed protocol in comparison to other secure and anonymous routing protocols using NS-2 simulator.

REFERENCES :

[1] B. Chen, R. Morris et al. Span “An energy-efficient coordination algo for topology maintenance in ad hoc wireless networks.” Wireless Networks, 8(5):481-94, 2002.

[2] Bashir Yahya, Jalel Ben-Othman “Robust and Energy Efficient Multipath Routing Protocol for Wireless Sensor Networks” IEEE "GLOBECOM" 2009 proceedings

[3] Bo Zhu, Zhiguo Wan, Mohan S. Kankanhalli, et al.Anonymous Secure Routing.

[4] J. Kong and X. Hong. ANODR: anonymous on demand routing with untraceable routes for mobile ad-hoc networks. 4th ACM International Symposium on Mobile Ad hoc Networking and Computing (MobiHoc 2003).

[5] Azzedine Boukerche, Li Xu et al. SDAR: A Secure Distributed Anonymous Routing Protocol for Wireless and Mobile Ad Hoc Networks.29th Annual IEEE International Conference on Local Computer Networks (LCN’04)

[6] Yanchao Zhang, Wei Liu, et al. MASK: Anonymous On-Demand Routing in Mobile Ad Hoc Networks. IEEE Transactions on Wirless Communications, VOL.5,Sept. 2006.

[7] C.-K. Toh, "Max Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks," IEEE Comm. Magazine, June, 2001.

[8] Seys, Preneel.ARM: Anonymous Routing Protocol for Mobile Ad hoc Networks.20th International Conference on Advanced Information Networking & applications (AINA’06)

[9] Martin Tompa, Heather Woll. “How to share a secret with Cheaters”. Springer-Verlag,1998.

[10] A. Shamir. “How to share a secret.” Comm. ACM, 22(11):612-613, November 1979.