A Trust Model for Secure and QoS Routing in MANETS

Shilpa S G [#1] , Mrs. N.R. Sunitha [#2], B.B. Amberker [#3]

[#1] *Department of Computer Science and Engg, Siddaganga Institute of Technology, Tumkur, Karnataka, India*

[#2] *Department of Computer Science and Engg, Siddaganga Institute of Technology, Tumkur, Karnataka, India*

[#3] Department of Computer Science and Engg, National Institute of Technology, Warangal, Andhra Pradesh, India

Abstract— **Due to the dynamic topology, limited and shared bandwidth, limited battery power of the mobile ad hoc network (MANET), providing Quality of Service (QoS) routing is a challenging task in MANET. The presence of malicious nodes in the network cause an internal threat that disobey the standard and degrades the performance of well-behaved nodes significantly. However, little work has been done on quantifying the impact of internal attack on the performance of ad hoc routing protocols using dynamic key mechanism. In this paper, we focus on the impact of Byzantine attack implemented by malicious nodes on AODV routing protocol as an extension of the previous work. Here, we propose a trust model in which the trustworthiness of each node is evaluated based on trust value and remaining energy of each node. Association level of each node is estimated based on the trust value calculated. Route selection is done using the trustworthiness and performance requirement of each route which is calculated based on both link capacity and traffic requirement to achieve QoS.**

Keywords: MANET, Byzantine attack, Trustworthiness, Authentication, QoS.

I. INTRODUCTION

MANET is vulnerable to various types of attacks because of open infrastructure, dynamic network topology, lack of central administration and limited battery-based energy of mobile nodes. These attacks can be classified into external attacks and internal attacks. Several schemes had been proposed previously that solely aimed on detection and prevention of external attacks [1]. But most of these schemes become worthless when the malicious nodes already entered the network or some nodes in the network are compromised by attacker. Such attacks are more dangerous as these are initiated from inside the network and because of this the first defense line of network become ineffective. Since internal attacks [1] are performed by participating malicious nodes which behave well before they are compromised therefore it becomes very difficult to detect. Routing protocols are generally necessary for maintaining effective communication between distinct nodes. Routing protocol not only discovers network topology but also built the route for forwarding data packets and dynamically maintains routes between any pair of communicating nodes. Routing protocols are designed to adapt frequent changes in the network due to mobility of nodes. Several ad hoc routing protocols have been proposed in literature and can be classified into proactive, reactive and hybrids protocols.

Due to several issues, routing protocol design has become a challenging task. The basic problem with most of the routing protocols is that they trust all nodes of network and based on the assumption that nodes will behave or cooperate properly but there might be a situation where some nodes are not behaving properly. Most adhoc network routing protocols becomes inefficient and shows dropped performance while dealing with large number of misbehaving nodes. Such misbehaving nodes support the flow of route discovery traffic but interrupt the data flow, causing the routing protocol to restart the route-discovery process or to select an

alternative route if one is available. The newly selected routes may still include some of misbehaving nodes, and hence the new route will also fail. This process will continue until the source concludes that data cannot be further transferred. Thus, the routing algorithm must react quickly to topological changes as per the degree of trust of a node or a complete path between a source and a destination pair. Nodes in MANETs communicate over wireless links. Therefore efficient calculation of trust is a major issue because an ad hoc network depends on cooperative and trusting nature of its nodes. As the nodes are dynamic the number of nodes in route selection is always changing, thus the degree of trust also keep changing.

Another challenging issue is energy efficient routing. Especially energy efficient routing is most important because all the nodes are battery powered. Failure of one node may affect the entire network. If a node runs out of energy the probability of network partitioning will be increased. Since every mobile node has limited power supply, energy depletion is become one of the main threats to the lifetime of the ad hoc network. So routing in MANET should be reliable in such a way that it will use the remaining battery power in an efficient way to increase the life time of the network.

In this paper, we propose a trust model for quality of service (QoS) routing in Manets, called Trust and Energy-based Quality of Service (TE-QOS) routing, which includes secure route discovery, secure route setup, and Trustworthiness based QoS routing metrics to mitigate against malicious nodes which selectively drop or modify packets they agreed to forward. The routing control messages are secured by using both public and shared keys, which can be generated on-demand and maintained dynamically. The message exchanging mechanism also provides a way to detect attacks against routing protocols, particularly the most difficult internal attacks. The routing metrics are obtained by combing the requirements on the trustworthiness value of the nodes in the network and the QoS of the links along a route.

The paper is organized as follows. In Section 2, we outline some relevant previous work. In section 3 we discuss a dynamic key management mechanism. In section 4 we discuss our trust model in MANET in detail and optimal routing are developed. In Section 5, we conclude this paper.

II. RELATED WORK

A) The following list of papers shows the relative work carried out for different types of attacks in MANETS and possible solutions given.

1) A Distributed Security Scheme for Ad Hoc Networks discuss the dos attack like flooding using AODV protocol and concludes with an immediate enhancement to make the limit-parameters adaptive in nature. This can be done by making calculations based on parameters like memory, processing capability, battery power, and average number of requests per second in the network and so on in [2].

2) A study of different types of attacks on multicast in mobile ad hoc networks: considers only rushing attack, black hole attack, neighbour attack and jellyfish attack in [3].

3) Mitigating denial-of-service attacks in MANET by incentive-based packet filtering: A game-theoretic approach in [4].

4) A survey of routing attacks in mobile ad hoc network which considers only routing attacks, such as link spoofing and colluding misrelay attacks in [5].

5) A Secure Routing Protocol against Byzantine Attacks for Manets in Adversarial Environments which considers an integrated protocol called secure routing against collusion (SRAC), in which a node makes a routing decision based on its trust of its neighbouring nodes and the performance provided by them in [6].

6) Enhanced Intrusion Detection System for Discovering Malicious Nodes in Mobile Ad Hoc Networks: The main feature of the proposed system is its ability to discover malicious nodes which can partition the network by falsely reporting other nodes as misbehaving and then proceeds to protect the network in [7].

7) WAP: Wormhole Attack Prevention Algorithm in Mobile Ad Hoc Networks without using any specialized hardware wormholes can be detected and isolated within the route discovery phase in [8].

8) A Reliable and Secure Framework for Detection and Isolation of Malicious Nodes in MANET: This

security framework involves detection of malicious nodes by the destination node, isolation of malicious nodes by discarding the path and prevention data packets by using dispersion techniques in [9].

9) A Cooperative Black hole Node Detection Mechanism for ADHOC Networks [10].

10) Rushing Attack Prevention (RAP) [11] for rushing attacks.

B) This section gives the overview of some proposed protocols that are related to energy balance and trust evaluation in reactive protocols.

In [12], Gupta Nishant and Das Samir had proposed a technique to make the protocols energy aware by using a new routing cost metric which is the function of the remaining battery level in each node on a route and number of neighbours of the node. This protocol gives significant benefits at high traffic but at low mobility scenarios.

In [13], Rekha Patil etal, has proposed an approach in which the intermediate nodes calculate cost based on battery capacity. The intermediate node judges its ability to forward the RREQ packets or drop it. This protocol improves packet delivery ratio and throughput and reduces nodes energy consumption.

M.Tamailarasi etal, in [14] has discussed the mechanism that integrates load balancing approach and transmission power control approach to maximize the life span of MANET. The results of this proposal reduce the average required transmission energy per packet compared to the standard AODV.

Bhalaji et al. in [15] have proposed an approach based on the relationship between the nodes to make them to cooperate in an ad hoc environment. The trust values of each node in the network are calculated by the trust units. The relationship estimator has determined the relationship status of the nodes by using the calculated trust values.

Kamal Deep Meka et al. in [16] have proposed a trust based framework to improve the security and robustness of adhoc network routing protocols. For constructing their trust framework they have selected the Ad hoc on demand Distance Vector (AODV) which is popular and used widely. Making minimum changes for implementing AODV and attaining increased level of security and reliability is their goal. Their schemes are based on incentives & penalties depending on the behaviour of network nodes. Their schemes incur minimal additional overhead and preserve the lightweight nature of AODV.

Huafeng Wu & Chaojian Shi1 in [17] has proposed the trust management model to get the trust rating in peer to peer systems, and aggregation mechanism is used to indirectly combine and obtain other node's trust rating. The result shows that the trust management model can quickly detect the misbehaviour nodes and limit the impacts of them in a peer to peer file sharing system.

The above papers have dealt the parameters battery power or trust of a node individually. Our proposal combines these two parameters to discover a reliable route between the source and destination.

III. DYNAMIC KEY MANAGEMENT SCHEME

A. Dynamic Key Mechanism

There are two basic key management approaches, i.e., public and secret key-based schemes [6]. The public key-based scheme uses a pair of public/private keys and an asymmetric algorithm such as RSA to establish session keys and authenticate nodes. In the latter scheme, a secret key is a symmetric key shared by two nodes, which is used to verify the data integrity. The initial construction starts by issuing public key certificates based on a users' own knowledge about other users' public keys. Initially, there is a PKI or CA to distribute the knowledge among users. Clearly, we need to assume that there is some kind of initial trusts among the nodes.

We first define a network, as shown in Fig. 1, and then describe a framework of dynamic key management.

Let $G = (V; E)$ be a network whose vertices in V are nodes and whose edges in E are direct wireless links among nodes. We define for each node x, the set $N_1(x)$,which contains the vertices in the network G that are hop-1 or direct neighbors of x, i.e.

$$N_1(x) = \{y: (x, y) \in E \text{ and } y \neq x\} \quad \text{--- (1)}$$

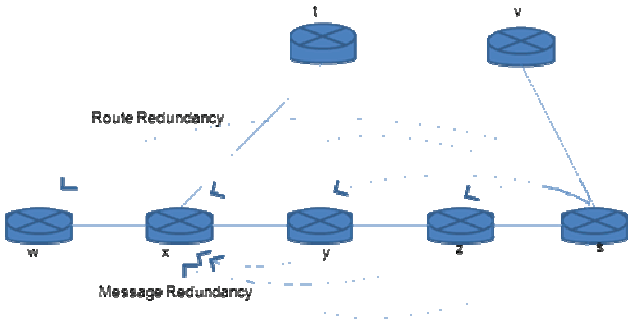

Fig.1 Route Discovery

Similarly, we define the hop-2 neighbours of a node as follows. For each node *x*, $N_2(x)$ contains the vertices in the network *G* that are hop-2 neighbours of *x*, which include neither vertices in $N_1(x)$ nor *x* itself, i.e.,

$$N_2(x) = \{z: (y; z) \in E \text{ and } y \in N_1(x), z \neq x\} \text{ --- (2)}$$

Similarly, we can define the hop-n neighbours of *x* $[N_n(x)]$ in terms of $N_{n-1}(x)$ if the flooding path from the source to destination has n links.

Initially, a node *x* has a public key $K_{x,pub}$ that is distributed to $N_1(x)$ by using PKI or CA. Similarly, a node *y* has public key $K_{y,pub}$ distributed to $N_1(y)$. Thus, for example, if $y \in N_1(x)$ and $x \in N_1(y)$, i.e., *x* and *y* are hop-1 neighbours, then *x* can authenticate *y* by issuing a certificate that is signed by *x* with *x's* private key. Those who hold *x's* public key can now read the certificate and trust the binding of *y* and its public key. Based on the available certificate and key information, two hop-1neighboring nodes can easily establish a secret key between them.

B. Key Distribution and Node Authentication

Whenever there is a need for a node to initiate a route discovery process, it creates pairwise shared keys with intermediate nodes, hop by hop, until it reaches the destination. First, it picks random number num. Then, it signs num with its private key by using a public key algorithm like RSA. After that, the route discovery message is protected by a keyed hash MAC algorithm such as *MD5*.Finally, the hash value and signature can now be attached to the route discovery message and sent out to its neighbours [6]. The complete route request (RREQ) packet sent by the node can be summarized as

$$m + h(m + num) + E(num, K_{s,pri}) \text{ -- (3)}$$

where $m = M + \{IDf\} + SN$, *M* is the original message; *IDf* denotes the ID of f , which is the node that forwards the message *m*; *SN* is the sequence number of the message; and $h(m + k)$ denotes the keyed hash algorithm with a key *k* on message *m*, where + denotes, the concatenation of strings, *E(m, K)* denotes the public key encryption algorithm.

Suppose that $z \in N_1(s)$ is one of s's hop-1 neighbours. Whenever there is a need for *s* to initiate a route discovery process, it picks a key k_1 at random, which will serve as the shared secret key between *s* and *z*. Then, *s* encrypts the key k_1 by using its neighbour's public key $K_{z,pub}$. After that, it encrypts the above encrypted key by using its own private key $K_{s,pri}$. The result serves as a signature for the route discovery message, which is protected by a keyed hash MAC algorithm .The complete RREQ sent by *s* can be summarized as

$$m_q + h(m_q + k_l) + E(E(k_l, K_{z,pb}), K_{s,pri}), \quad \text{for } r \in N_1(s) \text{ ---- (4)}$$

where m_q stands for the message used in RREQ. Then, *z* sends back s a route reply (RREP) packet in a similar format

$$m_p + h(m_p + k_1) + E(E(k_1, K_{s,pub}), K_{z,pri}), \text{ for } z \in N_1(s) \text{--- (5)}$$

where m_p stands for the message used in RREP.

By decrypting the message and comparing the key, *s* can authenticate *z* and distribute a shared key to *z*. Similarly, *s* establishes a shared key with each of its hop-1 neighbours.

Suppose that $y \in N_1(z)$. *z* can also similarly find out its hop-1 neighbours and also establishes a shared key with each of them. For *s* to send messages to its hop-2 neighbours, i.e., $N_2(s)$, for example, *y, s* requests *z* to forward the message to *y*. In *z's*

handshaking with *y, z* can pick s's public key instead of a random key and send it to *y*. This way, s's public key can be delivered to its hop-2 neighbours. Similarly, *s* can obtain the public keys of its hop-2 neighbours.

By checking the acknowledgement message back from *y* via *z*, *s* can find out all of its hop-2 neighbours $N_2(s)$. Therefore, *s* can send a message to $r \in N_2(s)$ via $z \in N_1(s)$ in the following format:

$$m_2 + h(m_2 + k_1) \text{ ----(6)}$$

where

$$m_2 = m + h(m + k_2) + E(E(k_2, K_{r,pub}), K_s, pri) \text{ for } r \in N_2(s), \text{ -- (7)}$$

where k_2 is the shared key between *s* and its hop-2 neighbour *r*.

C. Route Discovery and Attack Detection

Once the security associations between a source and destination have been established, and trustworthy routes have been identified from source to destination, the source can simply use the shared key to protect the

data traffic sent to the destination:

$m + h(m + k_{sd})$, where k_{sd} *is* the key shared between the source node (s) and destination node (d).

To detect internal attacks, including Byzantine attacks, we assume the following.

1) Each node has a pair of public/private keys and a unique ID. A compromised node participates in routing until detected.

2) The source and destination nodes are secured by external security agents. There is a shared key between the source and destination nodes.

3) Each of the intermediate nodes between the source and destination has established a shared key with the source node by using the key management scheme described in Section III-B.

4) There are enough uncompromised nodes in the network so that a message can arrive at the destination via different routes.

In this section, algorithm is extended in detecting collusion to Byzantine attacks, in which two or more nodes collude to drop, fabricate, modify, or misroute packets, and these nodes are consecutively located on a path [6].

1) Detection of a Single Malicious Node: To be more specific, we assume that *z* (in Fig. 1) is a compromised node during the route discovery phase, although it is initially authenticated. Clearly, *z* could not tamper the message from *s* to *y* because the message is protected with a key between *s* and *y*. Of course, *z* may simply drop the message when it needs to forward the message to *y*. However, there are at least two copies of the same message y expects to receive.

By comparing these copies from other neighbours, *y* is still able to detect that *z* is faulty or compromised. Similarly, *y* can also detect other internal attacks, such as message fabrication caused by *z*. Therefore, the attacks initiated by a single inside node can be detected.

2) Detection of Two Colluding Nodes: A more challenging case is the Byzantine attack. In our design of key management schemes, a source has directly established a shared key with each of its hop-n neighbours.

Suppose that both *z* and *y* are compromised and colluding. In addition, s shares a hop-1 key with *z* (i.e., $k_{1,sz}$), a hop-2 key with *y* (i.e., $k_{2,sy}$), and a hop-3 key with *x* (i.e., $k_{3,sx}$). During route discovery, *x* may receive three copies of a message *m* from *s* and via different intermediate nodes *y* and *z*, respectively, in the following formats:

$$C_1 = m + h(m + k_{3,sx})$$

$$C_2 = m + h(m + k_{2,sy}) + h(m + h(m + k_{2,sy}) + k_{1,yx})$$

$$C_3 = m + h(m + k_{1,sz}) + h(m + h(m + k_{1,sz}) + k_{1,zy}) + h(m + h(m + k_{1,sz}) + k_{1,yx}) \text{ ----- [8]}$$

When x receives the messages C_1, C_2, and C_3 which are from *s, y, z* respectively, it compares and find discrepancies among messages. C_1 directly comes from *s* and thus can be trusted; *y* cannot change the message without being detected, and thus, C_2 must match C_1. Therefore, C_3 has been modified, and *x* finds that there may be some compromised or faulty nodes among the nodes that forward the message, e.g., *z* and/or *y*. has modified the message but *y* does not tell during its forwarding. If *y* reports the discrepancies of the two copies, then *z* must be a compromised node. Otherwise, both *y* and *x* are

Compromised and colluding nodes, although *y* does not change the message.

In summary, the internal attacks initiated by a single compromised node and the Byzantine attacks can be detected without using expensive aggregated signatures as in [18], which are used to protect a route from end to end.

IV.TRUST MODELING AND OPTIMAL ROUTING

Trust modelling is a technical approach to represent trust for digital processing. The trust values are estimated considering various attributes related to behaviour of the node for a certain time. For each node in the network, a trust value will be stored that represent the value of the trustiness to each of its neighbour nodes. This trust value will be adjusted based on the experiences that the node has with its neighbour nodes.

Based on the above parameters trust level of each node can be of the following types:

STRANGER

- Node x have never sent/received any messages to/from node y
- Trust levels between them are very low.
- Probability of malicious behavior is very high.
- Newly arrived nodes are grouped in to this category.

KNOWN

- Node x have sent/received some messages to/from node y
- Trust levels between them are neither low nor too high.
- Probability of malicious behavior is to be observed.

FRIEND

- Node x have sent/received plenty of messages to/from node y
- Trust levels between them are very high.
- Probability of malicious behaviour is very less.

A. Association Evaluator Technique:

The Association status depends up on the trust value. The trust values are calculated based on the following parameters of the nodes

For that, a very simple equation for the calculation of trust value:

$$TV = \tanh(R1+R2+A) \quad \ldots \quad (9)$$

where, TV= Trust Value

$$R1 = \frac{No.of\ pkts\ forwarded\ succesfully\ from\ neighbor\ node}{Total\ no.of\ pkts\ to\ be\ forwarded\ by\ neighbor\ node}$$

If the denominator is not zero and R1 is less than the chosen threshold (R1 <1) & not zero then it can cause selective packet drop attack.

$$R2 = \frac{No.of\ pkts\ recieved\ from\ neighbor\ node\ but\ originated\ by\ other\ node}{Total\ no.of\ pkts\ recieved\ from\ that\ neighbor\ node}$$

A = Acknowledgement bit. (0 or 1) if the acknowledgment is received for data transmission from the destination then nodes in that path are assigned value 1 else value 0 is assigned.

Based on the trust value (TV) calculated for each node, the trust levels can be estimated as shown in table 1.

TABLE I:

TRUST ESTIMATION OF A NODE

Threshold Trust Value	Trust Level
0.7-1.0	F
0.4-0.6	K
0.0-0.3	S

Also, the Association between nodes is asymmetric, i.e. node x may not have trust on node y the same way as node y has trust on node x or vice versa.

Each node in an adhoc network would have identified its neighbourhood friends over a certain period of time by evaluating their trust levels. Some of the neighbourhood friends may suddenly turn malicious and non co-operative due to node capturing. To detect this, each node before starting

the data transfer may invoke the trust evaluator for a specific interval of time and can re-establish the trust levels.

If the threshold trust value is not satisfied, the friend is degraded to known and their packets are not forwarded. This is the penalty the node pay for not being cooperative. If however, the node turns out to be a repenting offender that is no longer malicious and that has behaved normally for a certain amount of time, re-socialization or re-integration in to the network is possible if the threshold trust level for a friend is satisfied. In this case, the concerned node will have to work its way up to raise its trust level to the threshold set for a friend.

B. *Power Consumption*

Every node in the MANET calculates its power consumption and finds the remaining energy periodically. Each node may operate in any of the following modes:

1) Transmission mode: The power consumed for transmitting a packet is given by the Eq (1)

Consumed energy = Pt * T (10)

where *Pt* is the transmitting power and *T* is transmission time.

2) Reception mode: The power consumed for receiving a packet is given by Eq (2)

Consumed energy = Pr * T (11)

where *Pr* is the reception power and *T* is the reception time.

The value *T* can be calculated as

T= Data size / Data rate (12)

Hence, the remaining energy of each node can be calculated using Eq (1) or Eq (2)

Rem energy = Current energy–Consumed energy (13)

Initially every node has full battery capacity say 100% which is assigned to current energy .On each transmission or reception of a data packet the remaining energy is found using the Eq(4). If the remaining energy falls below 50%, that node will not act as a router to forward the packets.

C. *Reliability Relation*

In Table II, the relationship of trust level and the remaining energy of each node are given which determines its Trustworthiness.

TABLE II:

TRUSTWORTHINESS OF EACH NODE

Trust Level	Remaining Energy %	Reliability(R)	Trustworthiness (T)
F	80-100	Very Very high	1.0
K	80-100	Very High	0.8
F	50-79	High	0.6
K	50-79	Medium	0.4
F	00-49	Low	0.3
K	00-49	Low	0.2
S	50-100	Low	0.2
S	00-49	Low	0.1
M	00-40	Very low	0.0

By using the trustworthiness value of each node we calculate the path trust.

Calculating Path Trustworthiness:

Consider a path $p \in P_{s \to x}$, where $P_{s \to x}$ is the set of paths that start from a source node s to a destination node x, i.e., $P_{s \to x}$= *{all paths from s to x}.*

The nodes on the path *p* calculate its trustworthiness using Table II by checking its trust level and the remaining energy and take the following decision:

If the reliability is very low the node discards the route request else the reliability is acceptable, cumulative reliability is found by adding the predecessor trustworthiness with its trustworthiness.

Cumulative Reliability,

$$CR_p = \sum_{n \in p} T \qquad (14)$$

If the node has already received the route request with same source address and same broad cast id and if the cumulative reliability is less than the cumulative reliability of current route request, the previous route request path is rejected and the current route request path is recorded.

Path Trustworthiness,

$$PT_p = \frac{CR_p}{No.of\ Hops} \qquad (15)$$

D. Mathematical Formulation of Optimal Routing

The routing metrics can be quantified as follows. First, we need to consider the trustworthiness of each candidate route. We assume that each node has locally built up a trustworthiness repository for the nodes it knows based on its CR and current behaviour observed in the topology discovery phase. A destination node has also calculated a path trustworthiness value for each possible route from the source node, as shown in above section. The repository is updated every time the topology is rediscovered.

Second, the performance requirement must be considered in making routing decisions. Assume that the transmission capacity of the wireless link that originates from a node n is B_n. The traffic requirement for the link is F_n, which is measured in the same units as B_n. For delay-sensitive traffic, we also use τ_n to represent the processing and propagation delay when being delivered by n to its next hop. A frequently used objective function [6] is,

$$Q_x(p) = \sum_{n \in p} \left(\frac{F_n}{B_n - F_n} + \tau_n F_n \right), \quad for\ p \in P_{s \to x} \qquad (16)$$

which is the average number of packets in the network based on the hypothesis that each queue behaves as an M/M/1 queue of packets. Note that for a link on path p, a smaller value of $Q_x(p)$ is preferred, either because of a smaller delay or a relatively larger link capacity.

Rewriting (16) into a recursive format, we have

$$Q_x(p) = Q_y(p_1) + \frac{F_x}{B_x - F_x} + \tau_x F_x, \quad y \in N_1(x), y \in p_1 \in p \qquad (17)$$

Combining both requirements on the trustworthiness and performance, a path that is less reliable and does not meet the desired performance must be penalized in our objective; thus, a combined cost function can be designed as

$$D(p) = \beta(1 - PT_p) Q_x(p), for\ p \in P_{s \to x} \qquad (18)$$

where β>0 is a constant used to scale the value of the cost function.

Now the routing problem can be written as

$$\text{minimize } D(p), \text{ for } p \in P_{s \to x} \qquad (19)$$

and subject to the constraints

$$B_n - F_n \geq 0;\ \tau_n \geq 0;\ R_{min} \leq R_x(n; j) \leq 1;\ \text{for } n \in p \qquad (20)$$

where R_{min} is the minimum trustworthiness required for a node to be allowed to join in a route. In (18)–(20), we explicitly integrate the security and performance requirements into a routing problem.

Note that if the intermediate nodes between s and x have the same levels of trustworthiness, then all the routes between s and x can equivalently be measured by the traditional hop counts. The routes of the same hop counts will have the same levels of trustworthiness. In this case, the optimal route is only determined by the performance requirements, as shown in (18), provided that the requirement on the minimum trustworthiness is met. By solving the optimization problem, we can develop a routing algorithm.

Here, the source node selects multiple routes as candidates. Each intermediate node along the candidate routes computes a cost from the source and passes it the next node on the way to the destination. The cost contains two parts, i.e., PT_p and Q_x, in terms of (15) and (17), respectively, which can be calculated based on the nodes along the route. Therefore, each route can be assigned an index called Trustworthiness–QoS index (TQI), i.e., a number to represent the combined trustworthiness and performance cost along the route, by each intermediate node along the route. The destination chooses a final route among the candidates in terms of the accumulated TQI value.

E. Routing Algorithm

The heuristic algorithm can be summarized as follows.

1) During route discovery, a source node sends RREQ packets to its neighbouring nodes. In these packets, along with the regular information, the node also sends its security related information, such as key information

2) Once an RREQ packet is received by an intermediate node, it calculates the TQI by using (18). The node places the link trustworthiness and QoS information in the RREQ packet and forwards it to its next hop. This process is repeated until it reaches the final destination.

3) At the destination, the node waits for a fixed number of RREQs before it makes a decision. Or else, a particular time can be set for which the destination or intermediate node needs to wait before making a routing decision. Once the various RREQs are received, the destination node compares the various TQI index values and selects the index with the least cost. It then unicasts the RREP back to the source node. When the source node receives the RREP, it starts data communication by using the route.

4) Once the route is established, the intermediate nodes monitor the link status of the next hops in the active routes. Those that do not meet the performance and trustworthiness requirements, as shown in (20), will be eliminated from the route.

5) When a link breakage in an active route is detected, a route error (RERR) packet is used to notify the other nodes that the loss of that link has occurred. Some maintenance procedures are needed as in AODV.

V. CONCLUSION

Security of mobile ad hoc networks has recently gained momentum in the research community. Due to the open medium of ad hoc networks, and their inherent lack of infrastructure, security exposures can be an obstacle to basic network operation. It is impossible to find a general idea that can work efficiently against all kinds of attack, since every attack has its own distinct characteristics. There are only fewer works on detecting and defending against internal attacks in the field of MANET's routing protocol.

This paper develops an optimal routing algorithm by combining both trustworthiness and performance. To derive trustworthiness, we used a dynamic trust based approach through which association between nodes are used to resist adhoc networks in byzantine environment. Fairness mechanism is included by the notion of friends, knowns and strangers. As far as number of alternative routes exists this protocol works well by choosing the optimal paths. This novel protocol can work in critical environment like military scenarios. Proposed approach is the first secure routing that quantitatively considers not only the security, network performance but also network lifetime by considering energy as well.

REFERENCES

[1] P. Papadimitratos, and Z.J. Haas, *"Securing the Internet Routing Infrastructure,"* IEEE Communications, vol. 10, no. 40, October 2002, pp. 60-68. Digital Object Identifier 10.1109/MCOM.2002.1039858

[2] Dhaval Gada, Rajat Gogri, Punit Rathod, Zalak Dedhia and Nirali Mody Sugata Sanyal, Ajith Abraham, "*A Distributed Security Scheme for Ad Hoc Networks",* ACM Publications, Vol-11, Issue 1, 2004, pp. 5 – 5.

[3] Hoang Lan Nguyen and UyenTrang Nguyen, "*A study of different types of attacks on multicast in mobile ad hoc networks"*, IEEE proceedings in International Conference on Networking (ICN 2006), 2006.

[4] Xiaoxin Wu, David K.Y, "*Mitigating denial-of-service attacks in MANET by incentive-based packet filtering: A game-theoretic approach"*, 3rd International conference on secure Communications, September 2007, pp. 310-319.

5] A , "A survey of routing attacks in mobile ad hoc networks", IEEE Journal on Wireless Communication, Vol-14, Issue 5, December 2007, ISSN: 1536-1284, pp.85-91.

[6] Ming Yu; Mengchu Zhou; Wei Su, "A Secure Routing Protocol Against Byzantine Attacks for MANETs in Adversarial Environments", IEEE Transactions on Vehicular Technology Vol-58, Issue 1, Jan. 2009 , pp.449 – 460.

[7] Nasser, N.; Yunfeng Chen, "Enhanced Intrusion Detection System for Discovering Malicious Nodes in Mobile Ad Hoc Networks", IEEE International Conference on Communications, ICC apos; Vol-07 , Issue 24-28 June 2007 , pp.1154 – 1159.

[8] Sun choi, Doo-young Kim, Do-hyeon Lee, Jae-il Jung, "WAP: Wormhole Attack Prevention Algorithm in Mobile Ad Hoc Networks", International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, Vol-0 , ISBN = {978-0- 7695-3158-8}, 2008, pp.343-348 .

[9] S.Dhanalakshmi, Dr.M.Rajaram, "A Reliable and Secure Framework for Detection and Isolation of Malicious Nodes in MANET", IJCSNS International Journal of Computer Science and Network Security, vol-8 No.10, October, 2008.

[10] Moumita Deb, "A Cooperative Black hole Node Detection Mechanism for ADHOC Networks", Proceedings of the World Congress on Engineering and Computer Science, 2008.

[11] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Rushing attacks and defense in wireless ad hoc network routing protocols," in Proc. ACM WiSe, Sep. 2003, pp. 30–40.

[12] Gupta Nishant, Das Samir, "Energy-aware on-demand routing for mobile Ad Hoc networks," Lecture notes in computer science ISSN: 0302-743, Springer, International workshop in Distributed Computing, 2002.

[13] Rekha Patil , A.Damodaram, "Cost Based Power Aware Cross Layer Routing Protocol For Manet", IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.12, December 2008.

[14] M.Tamilarasi, T.G Palani Velu, "Integrated Energy-Aware Mechanism for Manets using On-demand Routing", International Journal of Computer, Information, and Systems Science, and Engineering 2;3 © www.waset.org Summer 2008.

[15] N.Bhalaji, Dr.A.Shanmugam "Reliable Routing against selective packet drop attack in DSR based MANET" in Journal of Software, Vol. 4, No. 6, August 2009. pp 536-543

[16] Meka, Virendra, and Upadhyaya, "Trust based routing decisions in mobile ad-hoc networks" In Proceedings of the Workshop on Secure Knowledge Management, 2006.

[17] Huafeng Wu1, Chaojian Shi1," A Trust Management Model for P2P File Sharing System", International Conference on Multimedia and Ubiquitous Engineering, IEEE Explore 978-0-7695-3134-2/08, 2008.

[18] B. Awerbuch, R. Curtmola, D. Holmer, and C. Nita-Rotaru, ODSBR:An On-Demand Secure Byzantine Routing Protocol, Oct. 15, 2003,JHU CS Tech. Rep., Ver. 1.