Dependable Dynamic Source Routing without a Trusted Third Party

Asad Amir Pirzada **Chris McDonald** **Amitava Datta**

School of Computer Science & Software Engineering
The University of Western Australia,
35 Stirling Highway, Crawley, Western Australia, 6009.
Email: {pirzada,chris,datta}@csse.uwa.edu.au

Abstract

Ad-hoc networks are frequently used to establish communication in improvised environments without requiring any fixed infrastructure. These networks are formed with the help of their constituent wireless nodes, which are expected to forward packets for other nodes according to a pre-agreed upon protocol. The Dynamic Source Routing (DSR) protocol is one such protocol that helps to create and maintain routes in an ad-hoc network in spite of the dynamic topology. The accurate execution of this protocol requires sustained benevolent behaviour by all nodes participating in the network. This behaviour may not always be observed, and a number of known attacks against the standard DSR protocol can lead to its incorrect execution, and even failure. In this paper, we present a novel technique of discovering and maintaining dependable routes in an ad-hoc network even in the presence of malicious nodes. With the results from extensive simulations, we highlight the efficacy of our scheme and accentuate that it outperforms the DSR protocol when as many as 40% of the nodes are acting maliciously.

Keywords: Trust, Security, Ad-hoc, Networks, Protocols

1 Introduction

Mobile ad-hoc wireless networks hold the promise of the future, with the capability to establish networks at anytime, anywhere. These networks don't rely on extraneous hardware which makes them an ideal candidate for rescue and emergency operations. As these networks are devoid of any single traffic concentration point, each node in the network plays the role of a router on the run. The mobility of the nodes increases the routing complexity, as there is now no default gateway with a pre-configured static IP address.

Ad-hoc network routing protocols thus take into account the mobility of the network in order to compute functional routes. These protocols can generally be categorized into two types as Reactive and Proactive (Royer & Toh 1999). In reactive routing protocols, routing is accomplished in an on-demand manner, which generates initial latency during the route discovery process. However, in doing so they conserve the battery power of a node, which is considered a vital resource in ad-hoc networks. In contrast, proactive routing protocols create and update routes in the network at all instants of time but at the cost of high battery usage.

Both types of routing protocols require that the network nodes execute the protocols in a truthful manner in spite of their contemporary commitments and workload. However, such selfless behaviour is often difficult to sustain and so the execution of the routing protocols by the selfish nodes, frequently deviates from defined specifications. Similarly, as the ad-hoc networks are generally established in open and physically insecure environments, these may be targeted by attackers in a number of other ways.

These attacks in general can be divided into two major types: Passive and Active (Hu, Perrig & Johnson 2002). In passive attacks, a node only eavesdrops upon the network traffic in order to extract vital information from the data and control packets. In active attacks, however, a malicious nodes expends its own energy to launch fabrication, modification or impersonation attacks (Dahill, Levine, Royer & Shields 2002).

In order to protect ad-hoc networks against selfish and malicious behaviour, a number of secure routing protocols have been developed. These protocols employ a variety of cryptographic tools to protect the core routing protocol, which in turn secures the routes, thereby protecting the data that flows over them. More emphasis is laid upon the fortification and precise establishment of the routes, as there is least advantage of protecting the data if it never reaches its requisite destination.

A review of these secure routing protocols for mobile ad-hoc wireless networks (Pirzada & McDonald 2004*c*) indicates that most of the protocols presume the existence of a centralized or distributed trusted third party in the network. This assumption is also coupled with pre-configuration of nodes with encryption keys prior to joining the network. As the name suggests, ad-hoc wireless networks are rarely instituted in a planned manner. They have the diverse nature of being impromptu and so inherently oppose the dependence upon prerequisites. Based upon their establishment techniques, they can be roughly divided into two types: managed and pure (Pirzada & McDonald 2004*a*).

Managed ad-hoc networks are those, which have the provision of sustaining a trusted third party in the network. These networks thus require some a priori knowledge concerning the volume and setting of the network. This in turn restricts the size and mode of application of these networks. Managed ad-hoc networks are thus deemed suitable for law enforcement and military set-ups that generally have prior knowledge of forthcoming requirements.

In contrast, pure ad-hoc networks are not based upon any such assumptions, and hence permit rapid establishment of the network without any extraneous requirements. These networks are formed in a

spontaneous and self organized manner without necessitating a physical or virtual infrastructure. Any wireless node can gain or lose the membership of the network at any instant of time, without the need for prior registration (Pirzada & McDonald 2004*b*).

Establishing trust in managed networks is relatively simple to realize, as the absolute trust is placed in the cryptographic algorithms and their mode of implementation. However, this is achieved at the cost of deviating from the unplanned nature of ad-hoc networks to a semi-organized one. In order to sustain the improvised nature of ad-hoc networks, we deviate from the customary strategy of using cryptography and as an alternative use a trust-based system that is influenced by the human behavioural model. According to Denning (1993), "Trust cannot be treated as a property of trusted systems but rather it is an assessment based on experience that is shared through networks of people". As in real life, two entities with no previous mutual experience, put confidence in each other's competence so as to realize their respective goals. These shared experiences lead to trust development that augments and decays with time and frequency of interactions. In addition, reference also plays an imperative role in trust establishment. In the case where the trusted entity is unfamiliar, it assists to institute a preliminary trust level. Whereas in other cases, where the trustee is acquainted with the trusted entity, reference can strengthen or lessen the previous trust levels.

In this paper, we apply a similar trust and reference scheme to the DSR routing protocol. All nodes in the network independently execute a trust model and maintain their own assessment concerning other nodes in the network. Each node, based upon its personal experiences, rewards collaborating nodes for their benevolent behaviour and penalizes malicious nodes for their malevolent conduct. These trust levels are shared with other nodes in the network in a unique and efficient manner, permitting nodes to get a better viewpoint of unfamiliar nodes. Each sending node uses this trust information to compute the most trustworthy path to a particular destination. The routes worked out in this way are neither protected in terms of security nor minimal in terms of hops. However, these routes traverse nodes that have been identified as trust worthier than others and for this reason are more dependable in extemporized circumstances.

In Section 2 of this paper we present some related previous work. In Section 3 we explain our proposed scheme in detail. Simulation results are presented in Section 4. An analysis of the proposed scheme is presented in Section 5 with concluding remarks in Section 6.

2 Previous Work

To protect an ad-hoc network from attacks, a routing protocol must fulfil a set of requirements (Dahill et al. 2002) to ensure that the discovered path from source to destination functions properly in the presence of malicious nodes. These are:

1. Authorized nodes should perform route computation and discovery,
2. Minimal exposure of network topology,
3. Detection of spoofed routing messages,
4. Detection of fabricated routing messages,
5. Detection of altered routing messages,
6. Avoiding formation of routing loops, and
7. Prevent redirection of routes from shortest paths.

A number of secure routing protocols have been developed that conform to most of the above mentioned requirements. These protocols employ a variety of cryptographic tools for protecting the vulnerabilities in different routing protocols. However, these protocols have been developed as a practical response to specific problems that arose due to attacks on ad-hoc network routing protocols. Consequently, these protocols only cover a subset of all possible threats and are not flexible enough to be integrated with each other.

Some of the related previous work that has been carried out in order to make ad-hoc networks more trustworthy is explained in the following sub-sections.

2.1 ARAN

The Authenticated Routing for Ad-hoc Networks (ARAN) (Dahill et al. 2002) secure routing protocol is an on-demand routing protocol that identifies and shields against malevolent actions by malicious nodes in the ad-hoc network environment. ARAN relies on the use of digital certificates and can successfully operate in the managed-open scenario where no network infrastructure is pre-deployed, but a small amount of prior security coordination is expected. ARAN provides authentication, message integrity and non-repudiation in ad-hoc networks by using a preliminary certification process that is followed by a route instantiation process that guarantees end-to-end provisioning of security services.

All nodes are supposed to keep fresh certificates with a trusted server and should know the server's public key. Prior to entering the ad-hoc network, each node has to apply for a certificate that is signed by the certificate server. The certificate contains the IP address of the node, its public key, a timestamp of when the certificate was generated and a time at which the certificate expires, along with the signature by the certificate server. ARAN accomplishes the discovery of routes by a broadcast route discovery message from a source node, which is replied to in a unicast manner by the destination node. All the routing messages are authenticated at every hop from the source to the destination, as well as on the reverse path from destination to source.

ARAN requires the use of a trusted certificate server in the network. This requirement imposes a number of pre-setup restrictions that must be catered to before establishing the ad-hoc network. Also, having a centralized certificate repository in a physically insecure environment creates a single point of compromise and capture.

2.2 ARIADNE

ARIADNE (Hu et al. 2002) is an on-demand secure ad-hoc routing protocol based on the Dynamic Source Routing (DSR) protocol that protects against node compromise and relies only on extremely efficient symmetric cryptography. The security of ARIADNE is based upon the secrecy and authenticity of keys that are kept at the nodes. ARIADNE prevents a large number of Denial-of-Service attacks from malicious or compromised nodes. ARIADNE provides assurance that the target node of a route discovery process can verify the initiator, that the initiator can verify each transitional node that is on the path to the destination present in the `ROUTE REPLY` message and that no intermediate node can reduce the node list in the `ROUTE REQUEST` or `ROUTE REPLY` messages. Route Discovery is performed in two stages: the Initiator floods the network with a `ROUTE REQUEST` that

solicits a ROUTE REPLY from the Target. During route discovery the Target authenticates each node in the node list of the ROUTE REQUEST and the Initiator authenticates each individual node in the node list of the ROUTE REPLY.

For node authentication, ARIADNE has three alternative techniques i.e. TESLA (Timed Efficient Stream Loss tolerant Authentication) (Perrig, Canetti, Tygar & Song 2002), Digital Signatures, or pair-wise shared secret keys. The authentication mechanisms used by ARIADNE require excessive preconfiguration of nodes during the network establishment phase. ARIADNE prefers using the TESLA broadcast authentication scheme with delayed key disclosure both for its speed and efficiency. However TESLA in turn, requires clock synchronization between communicating nodes, which is considered to be an unrealistic constraint upon ad-hoc wireless networks.

2.3 Distributed Trust Model

The Distributed Trust Model (Rahman & Hailes 1997) employs a dedicated protocol for exchanging, withdrawing and reviving recommendations. Each entity maintains its own trust database by using this recommendation protocol. This in turn guarantees that the computed trust is neither absolute nor transitive. A decentralized approach to trust management is exercised. The model uses trust categories and trust values to find different levels of trust. The integral trust values in the model vary from -1 to 4 representing distinct levels of trust from absolute distrust (-1) to absolute trust (4). All entities execute the recommendation protocol either as a recommender or a requestor. The trust levels are computed by means of the recommended trust value of the target and its recommenders. The model also supports multiple recommendations for the same target and averages the same to generate a single recommendation value.

The Distributed Trust Model is most appropriate for scenarios with casual and short-term relationships but cannot be directly adapted for ad-hoc networks. Furthermore, it does not provide measures for safeguarding against false or malicious recommendations about other entities.

2.4 Distributed Public-Key Model

The Distributed Public-Key Model (Zhou & Haas 1999) employs threshold cryptography to distribute the private key of the Certification Authority over a number of servers. A (n, t+1) scheme permits any t+1 servers out of n servers to join their partial keys in order to generate the entire secret key. In the same way, it necessitates that a minimum t+1 servers must be overridden to get hold of the secret key.

The scheme is fairly robust but due to a variety of reasons its application to ad-hoc networks is severely restricted. First and foremost, it demands for the preconfiguration of servers and nodes, secondly the t+1 servers may not always be available to nodes seeking authentication and finally asymmetric cryptographic operations have been found to drain valuable node batteries.

3 Critical Issues

Some of the critical issues that need to be considered before employing any cryptographic or trust-based scheme to ad-hoc networks are:

1. Energy is one of the greatest constraints to a node's capabilities,
2. Symmetric encryption/decryption algorithms and hashing functions consume minimal computational energy in comparison to public key algorithms,
3. Transmission energy consumption is over three orders of magnitude greater than the energy consumption for encryption and hashing (Carman, Kruus & Matt 2000),
4. Requirement of a central trust authority is impractical in ad-hoc networks,
5. Intra-node relationships are usually less formal, temporary and short-term, and
6. There are two types of adversaries that an ad-hoc network may have to deal with: malicious and compromised.

Nodes in a wireless ad-hoc network are extremely dependent upon efficient utilization of their battery packs. Undue usage due to extra transmissions or computing can result in rapid battery draining. A less vivid but understated malicious behaviour is node selfishness in which nodes, in order to save their batteries, may be tempted to not relay packets. An easy solution against such attacks is the establishment of a central trust authority that can facilitate building of trust relationships among communicating nodes. Public Key Infrastructure (PKI) is an effective way of establishing trust but is deemed unsuitable because it makes use of asymmetric cryptographic algorithms that have been proved to be a target of Denial of Service attacks. Also, maintaining a centralized repository of certificates is itself a potential target of attack.

In a mobile ad-hoc network, the nodes are constantly leaving and entering the network. This necessities that the intra-node relationships be short lived and reciprocal in nature. As the time span of these relationships is really acute, consistent benevolent behaviour is expected of all nodes. However, in addition to benevolent nodes, there are malicious and compromised nodes also present in the network (Carter & Yasinsac 2002). A malicious node attempts to eavesdrop, replay, distort and impersonate messages while a compromised node is a benevolent node that has been taken over by an adversary to do the same. Compromised nodes possess valid signatures, identities and certificates and are hence very difficult to detect and isolate.

In contrast to encryption based mechanisms, trust-based schemes are able to identify nodes, which alter their behaviour pattern over a period of time (Marti, Giuli, Lai & Baker 2000). These schemes work in an interactive manner by computing the trust in network nodes based upon their past and present performance. The computed trust values are subsequently used to make critical routing decisions when accompanied by malicious, selfish or compromised nodes. Such schemes also have comparatively less pre set-up requirements and are hence deemed most appropriate for rapid deployment.

4 Trusted Routing in DSR

4.1 Dynamic Source Routing Protocol

The Dynamic Source Routing (DSR) protocol (Johnson, Maltz & Hu 2003) is a reactive routing protocol. As the name suggests it makes use of the strict source routing feature of the Internet Protocol. All data packets that are sent using the DSR protocol contain the complete list of nodes that the packet has to traverse. During route discovery, the source node broadcasts a ROUTE REQUEST packet with a unique

identification number. The ROUTE REQUEST packet contains the address of the target node to which a route is desired. All nodes that have no information regarding the target node, or have not previously seen the same ROUTE REQUEST packet, append their IP addresses to the ROUTE REQUEST packet and rebroadcast it. In order to control the spread of the ROUTE REQUEST packets, the broadcast is done in a non-propagating manner with the IP TTL field being incremented in each route discovery.

The ROUTE REQUEST packets keep spreading in the network until the time they reach the target node or any other node that has a route to the target node. The recipient node creates a ROUTE REPLY packet, which contains the complete list of nodes that the ROUTE REQUEST packet had traversed. Depending upon the implementation, the target node may respond to one or more incoming ROUTE REQUEST packets. Similarly, the source node may accept one or more ROUTE REPLY packets for a single target node.

For optimization reasons, nodes maintain a Path Cache or a Link Cache scheme (Hu & Johnson 2000). The former stores complete paths to a particular destination, while the latter only caches information related to individual links. The advantage of the Link Cache scheme is that it allows alternate paths to a destination even when some of the intermediate links have failed. Each node either forwarding or overhearing data and control packets, adds all useful information to its respective route cache. This information is used to limit the spread of control packets for subsequent route discoveries.

4.2 Vulnerabilities in DSR

The major vulnerabilities (Pirzada & McDonald 2004*d*) present in the DSR protocol are:

4.2.1 Deceptive alteration of IP addresses

During propagation of the ROUTE REQUEST packet, intermediate nodes add their IP addressees to it for route creation. However, any malicious node may modify, delete or add IP addressees to create routes as per its own requirement. Doing so enables malicious nodes to launch a variety of attacks in the network including creation of worm-holes, grey-holes and black-holes. Some of these attacks are described in Section 5.2.

4.2.2 Deceptive alteration of Hop Count

The hop count field of the IP packet usually informs the recipient of the total number of hops that the packet has traversed so far. Malicious nodes may increase this count so as to portray longer routes or decrease it for shorter routes. By doing so a malicious node is able to degrade or upgrade routes, thereby creating a topology that is most favourable to it.

4.3 Trust-based Routing

To compute the direct trust in a node, we have used an effort-return based trust model (Pirzada & McDonald 2004*a*). The accuracy and sincerity of the immediate neighbouring nodes is measured by observing their contribution to the packet forwarding mechanism. Every time a node transmits a data or control packet it immediately brings its receiver into the promiscuous mode, so as to overhear its immediate neighbour forwarding the packet. The sending node verifies the different fields in the forwarded IP packet for requisite modifications through a sequence of integrity checks. If the integrity checks succeed, it confirms that the node has acted in a benevolent manner

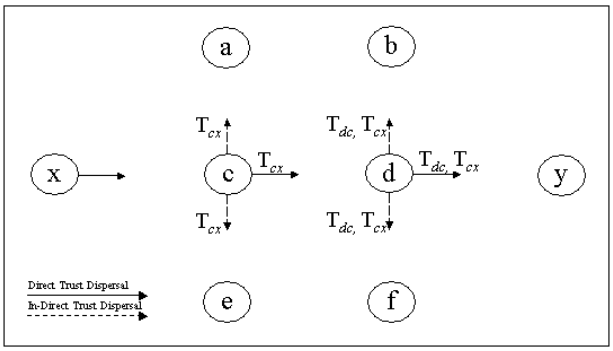

Figure 1: Propagation of Direct Trust along with Data Packets

and so its direct trust counter is incremented. Similarly, if the integrity check fail or the forwarding node does not transmit the packet at all, its corresponding direct trust measure is decremented.

We represent the direct trust in a node y by node x as T_{xy} and is given by the following equation:

$$\mathrm{T}_{xy} = \mathrm{W}(\mathrm{P_A}) \times \mathrm{P_A} + \mathrm{W}(\mathrm{P_P}) \times \mathrm{P_P}$$

where $\mathrm{P_A}$ represents the category Packet Acknowledgements that preserves a count of the number of packets that have been forwarded by a node. $\mathrm{P_P}$ represents the category Packet Precision, which maintains a count of the number of packets forwarded correctly. W reflects the weight or priority assigned to that particular category.

The category $\mathrm{P_P}$ and $\mathrm{P_A}$ are employed in combination to protect the DSR protocol against deceptive alteration of vital protocol fields and for identifying selfish node behaviour respectively.

4.4 Trust Propagation

Nodes in an ad-hoc network have limited transmission range, usually in the order of a few hundred meters. This restriction also limits the number of nodes that can be assessed by means of the direct trust mechanism. In order to disseminate the trust information beyond a single hop, we piggyback the direct trust value of nodes alongside the data packets as shown in Fig. 1.

Each node acting as a forwarder in an ongoing data session, appends the direct trust value to the data packet that was sent by the preceding node. In this manner, when the data packet is sent, it also distributes the trust information of other nodes in the network in a direct or in-direct manner. For example, a data packet from a node 'x' to node 'y', disseminates the trust information of node 'x' beyond a single hop to nodes 'a', 'b', 'd', 'e', 'f' and 'y' respectively.

Trust values can also be shared using an independent higher layer Reputation Exchange Protocol (Pirzada, Datta & McDonald 2004). However, such a protocol not only increases the control packet overhead but is also vulnerable to a number of other types of attacks specific to the exchange protocol.

To compute the total trust value of a target node, we have adopted and modified the probabilistic computing method by Beth *et al.* (1994) , however, any trust computing method (Jøsang 2001, Rahman & Hailes 1997) may be used for combining the trust values.

4.5 Trust Application

In DSR, before initiating a new route discovery, the cache is first scanned for a working route to the desti-

```
t(n)  The trust value assigned to node n by
      the source node s
p     Set of predecessors for each node on
      the most trustworthy path from the
      source
S     Set of nodes whose most trustworthy
      path from the source has been found
Q     Set of nodes whose most trustworthy
      path from the source has yet to be
      determined


initialise(Node s)
for all v nodes in the link-cache
{
      t[v] = 0
      p[v] = NULL
      t(s) = INFINITY
}

extract-most-trustworthy(Q)
{
      find the most trustworthy node in Q
      remove it from Q
      return the node
}

relax-neighbours(Node u, Node v, Tuv)
{
      if t(v) < t(u) + Tuv
      {
            t(v) = t(u) + Tuv
            p(v) = u
      }
}

dijkstra (Node s)
{
      initialise(s)
      S = Q = {}
      add s to Q
      while Q is not empty
      {
            u= extract-most-trustworthy(Q)
            add u to S
            for each node v which is an
            immediate neighbour of u
            {
                  relax-neighbours(u, v, Tuv)
            }
      }
}
```

Figure 2: Modified Dijkstra, from Shortest Path to Most Trustworthy Path Algorithm

nation. In the event of unavailability of a route from the cache, the ROUTE REQUEST packet is propagated. In the Link Cache scheme the default cost of each link is one. So when the search is made for a route in the cache, the shortest path in terms of number of hops is always returned. We modify this rule and associate the trust values, which have either been computed directly or received through the data packets, to the respective links in the cache. So each time a new route is required, the modified Dijkstra (1959) algorithm, as shown in Fig. 2, is executed to find the route with the maximum trust level. This method ensures that malicious nodes with lower trust levels are avoided and bypassed in all subsequent route discoveries.

5 Simulation

5.1 Set-up

To evaluate the effectiveness of the proposed scheme, we simulated the scheme in ns2 (NS 1989). The simulation parameters are listed in Table 1.

Table 1: Simulation Parameters

Examined Protocol		DSR
Simulation time		3600 seconds
Simulation area		1000 x 1000 m
Number of nodes		50
Transmission range		250 m
Movement model		Random waypoint
Maximum speed		20 m/s
Pause time		10 seconds
Traffic type		CBR (UDP)
Maximum Connections		30
Payload size		512 bytes
Packet rate		4 pkt/sec
Maximum malicious nodes		20
Types of attacks		Modification, black and grey hole
Black/Grey hole attacks	$W(P_A)$	0.25
Modification attacks	$W(P_P)$	0.75

We implement the random waypoint movement model for the simulation, in which a node starts at a random position, waits for the pause time, and then moves to another random position with a velocity chosen between 0 m/s to the maximum simulation speed. All benign nodes execute the trust model for the duration of the simulation. The optimal situational weights that are assigned to the P_A and P_P trust categories have been evaluated in past simulations (Pirzada & McDonald 2005).

5.2 Attack Patterns

Malicious nodes simulate the following types of attacks against data and control packets:

Modification Attack. These attacks are carried out by adding, altering or deleting IP addresses from the ROUTE REQUEST, ROUTE REPLY, ROUTE ERROR and Data packets, which pass through the malicious nodes.

Black Hole Attack. In this attack the malicious node drops all packets, which it was supposed to forward.

Grey Hole Attack. In the grey hole attack the malicious node selectively dumps data and control packets at random intervals.

5.3 Metrics

To evaluate the performance of the proposed scheme, we use the following metrics:

Packet Loss Percentage. It is the fraction of packets that were dumped by malicious nodes without any notification.

Throughput. It is the ratio between the number of packets received by the application layer of destination nodes to the number of packets sent by the application layer of source nodes.

Packet Overhead. This is the ratio between the total number of control packets generated to the total number of data packets received during the simulation time.

Byte Overhead. This is the ratio between the total number of control bytes generated to the total number of data bytes received during the simulation time.

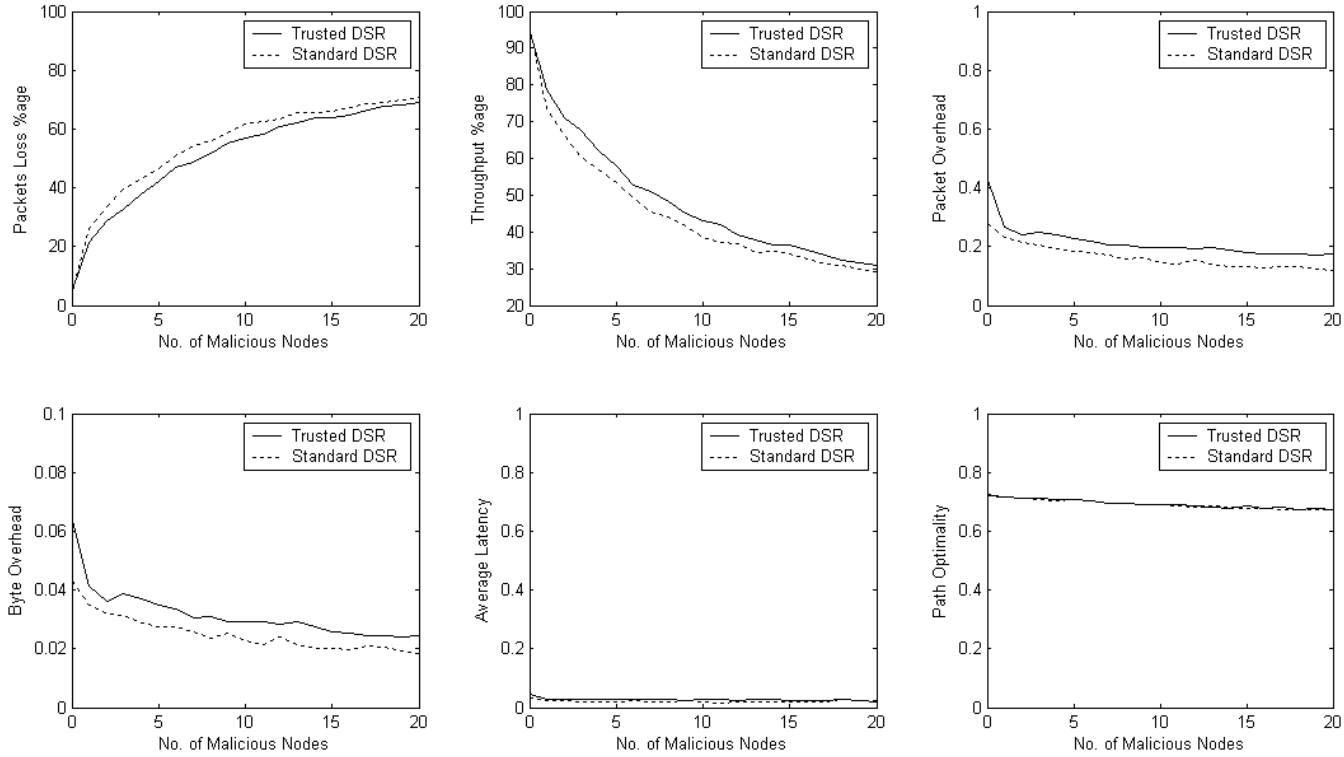

Figure 3: Simulation of Trust-based Routing in the presence of Malicious Nodes

Average Latency. Gives the mean time (in seconds) taken by the packets to reach their respective destinations.

Path Optimality. It is the ratio between the number of hops in the shortest possible path to the the number of hops in the actual path taken by a packet to reach its destination.

6 Analysis

Fig. 3 depicts the performance results for the proposed scheme compared with that of the standard DSR protocol, in the presence of a varying number of malicious nodes. The packet loss in the standard DSR protocol is 5 to 10% higher than that of the trusted DSR protocol. This can be attributed to the fact that the former does not take into account the benevolence levels of the nodes and prefers shorter routes by default. All listed attacks generate no form of notification that inform the other nodes regarding their malevolent activities. So the malicious nodes are constantly selected in the routing process, which leads to an overall lower throughput of the network.

The trusted DSR protocol, on the other hand, constantly monitors the ongoing behaviour of its neighbouring nodes. It selects or deselects en route nodes based upon their trust levels and thus attempts to avoid any malicious nodes. This deviation from the optimal paths leads to an increase in the packet overhead by up to 5%. The byte overhead is negligible and has no significant impact on the overall throughput. The latency and path optimality also indicate variation from that of the standard DSR protocol. This is due to the fact that the routes, which are being retrieved from the cache, have been optimized based upon their trust levels rather than the number of hops. This essentially decreases the path optimality of the routes, which consequently increases the latency of the network.

The method of verifying the sincerity of the immediate neighbour through promiscuous mode is an effective way of detecting selfish and malicious nodes. However, this technique has certain drawbacks, which have been highlighted by Marti *et al.* (2000). One of them is the ambiguous collision problem in which a node A cannot hear the broadcast from neighbouring node B to node C, due to a local collision at A. The other is the receiver collision problem, in which a node A overhears node B broadcast a packet to C but cannot perceive the collision which takes place at node C. In the same way, if the nodes have different transmission power ranges, the mechanism of passive acknowledgments might not function as desired.

It is possible that a malicious node may fallaciously modify the trust levels of other nodes during the transmission of the data packets. However, as the next immediate recipient holds a specific direct trust level for the malicious node, it would append that value to the data packet. If the received trust level for an immediate neighbouring node is below a certain threshold, the recipient node may completely disregard the data packet.

In order to deceive the trust mechanism, any malicious node may initially perform in a benevolent manner so as to gain a higher trust rating in the network. However, as soon as the malicious node starts to modify or dump packets, its corresponding direct trust levels maintained by its immediate neighbours start to decay. Consequently, the malicious node is bypassed in all subsequent route discoveries.

However, the isolation of malicious nodes, which depict complex behaviour like launching attacks in collusion or by operating at the verge of benevolent and malevolent behaviour, is quite intricate. We recommend using Intrusion Detection systems such as those proposed by Zhang *et al.* (2000) and Kachirski *et al.* (2002) for isolating such nodes in ad-hoc networks.

7 Conclusions

Ad-hoc networks are frequently targeted by participating malicious nodes to sabotage the network.

A common mechanism to protect these networks is through the use of encryption and hashing mechanisms. However, the implementation of these mechanisms generally imposes certain unessential requirements, which are considered as restrictive for unplanned environments.

In order to maintain the makeshift nature of ad hoc networks, in this paper we have used an unusual approach of enforcing trust in the network. We have moved from the common mechanism of achieving trust in the network via security to enforcing dependability through collaboration. Each node in the network monitors its surrounding neighbours and maintains a direct trust value for them. These values are propagated through the network along with the data traffic. This permits evaluation of the global trust knowledge by each network node without the need of a trusted third party. These trust values are then associated with the nodes present in the DSR link cache scheme. This permits nodes to retrieve dependable routes from the cache instead of standard shortest paths. Through extensive simulations we have found that the throughput of the trusted DSR protocol remains significantly higher than that of the standard DSR protocol in the presence of malicious nodes.

8 Acknowledgements

This work was supported by an International Postgraduate Research Scholarship and the University Postgraduate Award. We would also like to thank Diana Senn from the ETH Information Security Group (Swiss Federal Institute of Technology, Zurich) for contributing NS2 code in support of the simulations.

References

Beth, T., Borcherding, M. & Klein, B. (1994), 'Valuation of trust in open networks', *Proceedings of the Third European Symposium on Research in Computer Security (ESORICS)* pp. 3–18.

Carman, D., Kruus, P. & Matt, B. (2000), 'Constraints and approaches for distributed sensor network security', *Technical Report #00-010, NAI Labs* .

Carter, S. & Yasinsac, A. (2002), 'Secure position aided ad hoc routing protocol', *Proceedings of the International Conference on Communications and Computer Networks (CCN)* pp. 329–334.

Dahill, B., Levine, B. N., Royer, E. & Shields, C. (2002), 'A secure routing protocol for ad hoc networks', *Proceedings of the International Conference on Network Protocols (ICNP)* pp. 78–87.

Denning, D. (1993), 'A new paradigm for trusted systems', *Proceedings of the ACM New Security Paradigms Workshop* pp. 36–41.

Dijkstra, E. W. (1959), 'A note on two problems in connection with graphs', *Numerische Mathematik* pp. 83–89.

Hu, Y. C. & Johnson, D. B. (2000), 'Caching strategies in on-demand routing protocols for wireless ad hoc networks', *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking* pp. 231–242.

Hu, Y. C., Perrig, A. & Johnson, D. B. (2002), 'Ariadne: A secure on-demand routing protocol for ad hoc networks', *Proceedings of the eighth Annual International Conference on Mobile Computing and Networking* pp. 12–23.

Johnson, D. B., Maltz, D. A. & Hu, Y. (2003), 'The dynamic source routing protocol for mobile ad hoc networks (dsr)', *IETF MANET, Internet Draft (work in progress)* .

Jøsang, A. (2001), 'A logic for uncertain probabilities', *Proceedings of the International Journal of Uncertainity, Fuzziness and Knowledge-Based Systems* **9**(3), 279–311.

Kachirski, O. & Guha, R. (2002), 'Intrusion detection using mobile agents in wireless ad hoc networks', *Proceedings of the IEEE Workshop on Knowledge Media Networking (KMN)* .

Marti, S., Giuli, T., Lai, K. & Baker, M. (2000), 'Mitigating routing misbehavior in mobile ad hoc networks', *Proceedings of the Sixth annual ACM/IEEE International Conference on Mobile Computing and Networking* pp. 255–265.

NS (1989), 'The network simulator', *http://www.isi.edu/nsnam/ns/* .

Perrig, A., Canetti, R., Tygar, D. & Song, D. (2002), 'The tesla broadcast authentication protocol', *RSA CryptoBytes* **5**(2).

Pirzada, A. A., Datta, A. & McDonald, C. (2004), 'Propagating trust in ad-hoc networks for reliable routing', *Proceedings of the International Workshop on Wireless Ad-hoc Networks (IWWAN)* .

Pirzada, A. A. & McDonald, C. (2004*a*), 'Establishing trust in pure ad-hoc networks', *Proceedings of the 27th Australasian Computer Science Conference (ACSC)* **26**(1), 47–54.

Pirzada, A. A. & McDonald, C. (2004*b*), 'Kerberos assisted authentication in mobile ad-hoc networks', *Proceedings of the 27th Australasian Computer Science Conference (ACSC)* **26**(1), 41–46.

Pirzada, A. A. & McDonald, C. (2004*c*), Secure routing protocols for mobile ad-hoc wireless networks, *in* T. A. Wysocki, A. Dadej & B. J. Wysocki, eds, 'Advanced Wired and Wireless Networks', Springer.

Pirzada, A. A. & McDonald, C. (2004*d*), 'Secure routing with the dsr protocol', *Proceedings of the Third International Workshop on Wireless Information Systems (WIS)* pp. 24–33.

Pirzada, A. A. & McDonald, C. (2005), Reliable routing in ad-hoc networks using direct trust mechanisms, *in* D. Z. Du & G. Xue, eds, 'Advances in Wireless Networks and Mobile Computing (to appear)', Springer.

Rahman, A. A. & Hailes, S. (1997), 'A distributed trust model', *Proceedings of the ACM New Security Paradigms Workshop* pp. 48–60.

Royer, E. M. & Toh, C. K. (1999), 'A review of current routing protocols for ad hoc mobile wireless networks', *IEEE Personal Communications Magazine* **6**(2), 46–55.

Zhang, Y. & Lee, W. (2000), 'Intrusion detection in wireless ad hoc networks', *Proceedings of the 6th International Conference on Mobile Computing and Networking* .

Zhou, L. & Haas, Z. J. (1999), 'Securing ad hoc networks', *IEEE Network Magazine* **13**(6), 24–30.