25

Trust Establishment in Mobile Ad Hoc Networks: Direct Trust Distribution-Performance and Simulation

Dawoud D.S.[1], Richard L. Gordon[2],
Ashraph Suliman[1] and Kasmir Raja S.V.[3]
[1]National University of Rwanda
[2]University of KwaZulu Natal
[3]SRM University, Chennai,
[1]Rwanda
[2]South Africa
[3]India

1. Introduction

In the previous chapter, we discussed the distinct characteristics of ad hoc networks, which make them very difficult to secure. Such characteristics include: the lack of network infrastructure; no pre-existing relationships; unreliable multi-hop communication channels; resource limitation; and node mobility. We provided a theoretical background to mobile ad hoc networks and the security issues that are related to such networks. We defined the ad hoc networks and their characteristics in terms of trust establishment. As the focus of the two chapters is on the network layer, attacks specific to this layer are identified and explained in Chapter 1. We also presented a survey of the existing key management solutions for mobile ad hoc networks.

The current chapter is a continuation for the previous one. This is why we start this chapter by Section-2 that offers a survey of the existing secure routing protocols for mobile ad hoc networks. This section makes a pertinent observation that most secure routing protocols assume some kind of key management authority exists. Mobile ad hoc networks have little fixed network architecture and it is unlikely that there is a centralised authority member. Section-2 of this chapter together with last section of the previous one identify the problem that the two chapters together are addressing. There exists secure routing mechanisms to address the unique characteristics of mobile ad hoc networks, however, these solutions assume that key management is addressed prior to network establishment. A novel, on-demand solution to the key management problem for mobile ad hoc networks is then described. Section-3 details the functionality and operation of the proposed model: "Direct Indirect Trust Distribution" (DITD). The DITD model focuses on the task of distributing keying information. The DITD model also includes a verification optimization protocol and trust evaluation metric, which maximises the security of distribution.

The implementation and simulation of the DITD model is examined in Section-4. There are various packages used to compare existing and proposed routing protocols. One such package is the ns2 Network Simulator, which is commonly used in the relating literature. A comparative ns2 simulation study between the DITD and the AODV protocols is presented. The DITD model is based on the Ad Hoc On-demand Distance Vector (AODV) routing protocol. Simulations show the performance overhead of including key management functionality and the performance of DITD in the presence of malicious attacking nodes.

Section-5 summarises the contribution of the Chapter to the field of trust establishment in mobile ad hoc networks. Section-5 also provides future direction for research.

2. Secure routing in mobile ad hoc networks

A mobile ad hoc network's routing protocol has unique challenges due to the dynamic nature of ad hoc network. Mobile ad hoc networks do not have the same privileges that fixed, wired networks have. The routing mechanisms are uniquely designed to deal with the lack of infrastructure and unreliable wireless multi-hop communication channels.

This section investigates the procedure of securing of these routing protocols. The routing solutions are briefly visited and an extensive survey is presented for the existing security mechanisms that are used to secure these routing protocols.

Routing in mobile ad hoc networks is divided into two categories: table driven methods and on-demand methods. Table driven methods are also known as proactive routing. They maintain routing tables that contain routes to all the nodes in the network. Theses tables are periodically updated which allows routing information to be available at all times. Examples of table methods include Destination Sequence Distance Vector Routing (DSDV) [Perkins & Bhagwat, 1994] and Optimized Link State Routing (OLSR). Source initiated on-demand routing methods establishes routes in a reactive manner. Routes are established through a route discovery phase. During a route discovery phase node *S* will broadcast a request message *RREQ* into the network. This request message is forwarded until it reaches its target destination node *D*. Node *D* then replies with a reply message *RREP* which is unicast along the reverse route, until it reaches the source and the route is established. Routes are maintained as long as they are required. Examples of on-demand methods include Ad Hoc On-Demand Distance Vector (AODV) [Perkins et al, 2003] and Dynamic Source Routing protocol (DSR) [Johnson et al, 2001]. The reactive on-demand approach is less computationally expensive, in comparison with the proactive table driven approach.

In the previous chapter, it is identified that most security attacks target the network layer, and more specifically the routing protocol. These attacks include: black-hole attacks; wormhole attacks; eavesdropping attacks; byzantine attacks; resource consumption attacks; and routing table poisoning. The routing protocol is found on the network layer and is a significant service for mobile ad hoc network. Adversaries, specifically, target the routing protocol. Thus, a secure routing solution is needed for ad hoc networks to be securely implemented.

This section gives a survey and an analysis of the existing secure routing protocols. Each protocol is presented and investigated based on: functionality; operational assumptions; and security. A summary and discussion is formulated at the close of this section.

2.1 Secure Efficient Ad hoc Distance vector routing protocol (SEAD)

Secure efficient ad hoc distance vector (SEAD) [Hu et al, 2002] is a secure routing protocol which is used in conjunction with the table driven destination-sequenced distance vector (DSDV) routing protocol [Perkins & Bhagwat, 1994]. The DSDV routing protocol uses a distributed version of the Bellman-Ford algorithm to discover the shortest path between two nodes. The SEAD protocol uses symmetric key cryptography and one-way hash functions to protect against security attacks like denial of service and resource attacks.

a. System Overview

The DSDV routing protocol discovers the shortest path based on a route's hop count. Routing packets are assigned sequence numbers to ensure the most recent route is processed. The hop count and sequence number variables are stored in the routing packets. Attackers can create an incorrect routing state in nodes resulting in a denial of service attack (DoS) where the attacker attempts to make other nodes consume excess bandwidth and processing time. SEAD makes the routing process robust against multiple uncoordinated attackers by authenticating the hop count and sequence number of routing packets with a one-way hash function h. Hash chaining is used so that only nodes that are in possession of the previous routing update (identified by a sequence number) can broadcast a new routing update. Authenticated routing updates are computed to prevent against malicious routing updates broadcast by attackers.

b. One-Way Hash Function

SEAD uses a one-way hash function to authenticate routing updates and minimize resource consumption attacks. A formal definition of the hash function H is provided in [Stalling, 2003]. The most commonly used hash functions are MD-5 [Rivest, 1992] and SHA-1 [Publications F IPS, 2008].

A one-way hash function H is used to generate a one-way hash chain h. The one-way hash function H has an input of any bit length * and outputs a variable of fixed bit length p. The one-way hash function H must be computationally impossible to invert.

$$H: (0,1)^* \rightarrow (0,1)^p$$

A hash chain h_i is created when a node selects a random number $x \in (0,1)^p$ and uses it to generate a list of variable which make up a hash chain $h_0, h_1, h_2, h_3, \ldots, h_n$. Here $h_0 = x$ and h_i is calculated using the irreversible one-way hash function H such that:

$$h_i = H(h_{i-1}) \text{ where } 0 \leq i \leq n$$

Assuming there is an existing authenticated element, a node can verify elements later in the chain's sequence. For example if an authenticated element h_i exists, a node can authenticate h_{i-4} by checking that $h_i = H(H(H(H(h_{i-4}))))$. SEAD assumes the existence of an authentication and key distribution mechanism to distribute an authenticated element like h_n allowing for authentication by hash chaining [Hu et al, 2002].

c. Authenticating routing updates

SEAD uses the elements of the hash chain to provide authentication and secure the routing updates in DSDV. SEAD assumes an upper bound on the variable to be authenticated, for example if it were the hop count then SEAD would assume a maximum route distance n in the network (the maximum hop count between two nodes allowed). This also eliminates any routes with a length greater than m to exist, eliminating possible routing loops or the routing infinite problem.

The sequence values that make up the hash chain are calculated from the H function such that $h_1, h_2, \ldots, h_n$ where n is divisible by m. For a routing table entry with sequence number i

let $k = {}^{n}/_{m} - i$. An element from h_{km}, h_{km+1}, …, h_{km+m-1} is used to authenticate the routing entry with sequence number i. If the hop count is j where $0 \leq j < m$, then h_{km+j} is used to authenticate the routing entry found with sequence number i and hop count j [Hu et al, 2002].

Routing updates are sent with the appropriate routing information and a hash chain value is used to authenticate the update. If the authentication value appended is h_{km+j} then only attackers with h_{km+j-1} can modify the authentication value. Nodes receiving a routing update, check the authentication value h_{km+j} by calculating the new hash chain value. Receiving nodes can calculate the new hash chain value by using the earlier hash chain value h_{km+j-1} and the received sequence number i and hop count j. If the new calculated hash value is equal to h_{km+j} then the routing update is verified.

SEAD proposes two methods for routing update authentication. One method uses clock synchronization and a broadcast authentication mechanism like TESLA [Perrig et al, 2001]. The second method requires a shared secret between each communicating node pair. The secret can be used to implement a message authentication code (MAC) between nodes authenticating routing update messages.

d. Analysis

The SEAD protocol protects the ad hoc network from routing attacks that target resource consumption. The SEAD protocol does protect against multiple uncooperative attacks, preventing routing loops but routing loop prevention cannot be guaranteed in the presence of co-operating attackers. The SEAD protocol is vulnerable to intelligent attackers that use the same sequence number and same hop count of the most recent update to corrupt routing information. The SEAD protocol provides protection against denial of service attacks [Perrig et al, 2001], replay attacks and routing table poisoning by authenticating routing updates so malicious nodes cannot corrupt the routing procedure.

2.2 A secure on-demand routing protocol for ad hoc networks (Ariadne)

Ariadne [Hu et al, 2005] is a secure on-demand routing protocol which uses symmetric cryptography. Ariadne is based on the on-demand DSR [Johnson et al, 2001] routing protocol and is developed by the same authors as the SEAD protocol [Hu et al, 2002]. Ariadne provides end-to-end authentication on the routing layer.

a. System Overview

Ariadne assumes a shared secret key between communicating node pairs and uses message authentication code (MAC) to authenticate end-to-end packets between the communication pair. Broadcast authentication is employed, with loose time synchronization, to authenticate route request and other broadcast packets. The TESLA [Perrig et al, 2001] broadcast authentication scheme is used. In TESLA the source generates a one-way key chain and a schedule is made which defines at which time keys of the chain are revealed. This mechanism limits Ardiadne’s operation to ad hoc networks which have time synchronization. Ardiane provides end-to-end authentication in an on-demand manner over the DSR routing protocol [Hu et al, 2005].

b. End-to-end Authentication

For communication from a source node *S* to a destination node *D*, the source *S* will broadcast a route request into the network and expect a reply from *D*. Ariadne assumes a shared secret between *S* and *D*, K_{SD} and K_{DS}, which enables message authentication for each respective direction.

Nodes *S* wanting to start a route discovery for node *D* will first generate an initial hash chain h_0 consisting of: a packet identifier identifying the type of packet (a request packet *RREQ* in this case); the source's address (ID_S); the destinations address (ID_D); a broadcast identity (*bi*) identifying the current route discovery; and a TESLA time interval (*tes*) identifying the expected time of arrival at the destination.

$$h_0 = MAC_{K_{SD}}(RREQ|ID_S|ID_D|bi|tes)$$

Node *S* will broadcast a route request packet which includes: a packet identifier, the hash chain h_0; the source's address (ID_S); the destinations address (ID_D); the broadcast identity (*bi*); the TESLA time interval (*tes*); a node list *N()* and a MAC list *M()*. The packet broadcast is as follows:

$$S \rightarrow broadcast \;:\; RREQ|h_0|ID_S|ID_D|bi|tes|N()|M()$$

A neighbouring node that receives the route request checks the validity of the TESLA time interval, *tes*. The TESLA time interval is valid if the corresponding key that it points to has not been revealed yet and the time interval does not point too far in the future. The neighbouring node *A* will then compute a new hash chain h_1 using the previous hash chain h_0. A message authentication code of the packet to be broadcast is created (MAC_A). MAC_A is calculated using the TESLA key (K_{Ates}). Before forwarding the packet the neighbour node *A* includes: the hash chain h_1; itself in the node list *N*; and the MAC_A calculated in the MAC list *M*. The hash function and broadcast packet are as follows:

$$h_1 = H(A|h_0)$$

$$A \rightarrow broadcast \;:\; RREQ|h_1|ID_S|ID_D|bi|tes|N(A)|M(MAC_A)$$

Intermediate node *P* receiving a forwarded route request first calculates a new message authentication code MAC_P and a new hash chain $h_i = H(P-1|h_{i-1})$ where *P*-1 is the previous node and h_{i-1} is the previous hash chain value. Secondly it includes this information and forwards the route request as follows:

$$P \rightarrow broadcast \;:\; RREQ|h_i|ID_S|ID_D|bi|tes|N(A, \dots, P)|M(MAC_A, \dots, MAC_P)$$

The route request is propagated to the destination node *D*. When *D* receives the route request it validates the authenticity of the route request by checking that the TESLA time intervals indicate no keys have been released as of yet and that the hash chain is valid. *D* then generates a message authentication code MAC_D. $\mathrm{MAC_D}$ and an empty key list *K()* are included in the packet and sent back along the reverse path indicated by the node list and DSR protocol. The $\mathrm{MAC_D}$ and reply message are as follows:

$$MAC_D = MAC_{K_{DS}}(RREP|ID_D|ID_S|bi|tes|N(\dots)|, M(\dots)$$

$$D \rightarrow P \;:\; RREP|ID_D|ID_S|bi|tes|N(\dots)|M(\dots)|MAC_D|K()$$

Intermediate node that receive a reply message will wait for the *tes* time interval to lapse so the corresponding key can be revealed an included in the key list *K()*. The reply message is forwarded until it contains all the TESLA keys of the intermediate nodes and it finally reaches the source node *S*. The source then verifies the validity of all the keys, MAC_D, and the message authentication code contains.

c. Maintenance

Ariadne achieves secure route maintenance by authenticating the DSR error messages. Ariadne authenticates error messages preventing malicious nodes from broadcasting false broken links and causing denial of service type attacks. When an error message is generated TESLA authentication information is included. If authentication is delayed as a result of the TESLA time intervals, the intermediate nodes buffer the error message until the appropriate keys are revealed and the message can be authenticated and action taken [Hu et al, 2005].

d. Analysis

The authors of Ariadne are the same authors of SEAD [Hu et al, 2002] protocol. Ariadne employs an end-to-end approach to authentication while SEAD uses a hop-by-hop approach because of the DSDV routing procedure. The Ariadne proposal is based on the on-demand DSR routing protocol. Ariadne implements TESLA broadcast authentication and message authentication code to provide authentication for routing packets in an ad hoc network environment. The Ariadne proposal assumes that there exists some shared secret between a communication pair, therefore assuming the existence of an authentication and key distribution mechanism. Ariadne relies on TESLA authentication which requires time synchronization in the ad hoc network, synchronization is difficult to achieve without the presence of an outside authorized member or TTP.
Ariadne implements end-to-end authentication to prevent unauthorized nodes from sending error messages and incorrect routing packets in the form of repays attacks. However this proposal does not consider the case where attackers do not cooperate with the routing protocol and drop routing packets which are suppose to be forwarded. An extension is proposed in [Hu et al, 2003] which uses packet leashing to solve this problem.

2.3 Authenticated Routing for Ad hoc Networks

The authenticated routing for ad hoc networks (ARAN) protocol [Sanzgiri et al, 2002] is a securing routing solution which uses cryptographic certificates. ARAN is designed for an on-demand ad hoc routing protocol and achieves authentication, integrity and non-repudiation on the network layer but assumes prior shared secrets at initialization.

a. System Overview

The ARAN secure routing protocol establishes trust in three stages:

1. Issuing of certificates
2. Route Discover process
3. Shortest path Optimization

Initially ARAN assumes the presence of a trusted third party (TTP) which issues valid certificates, and a shared public key for all participating nodes. The route discovery process of ARAN provides end-to-end authentication for communicating nodes. The source node broadcasts a route request which carries the source’s certificate. The route request is propagated to the destination node by an end-to-end authentication process. The destination node responds by unicasting a reply message back along the found route using the end–to-end authentication protocol.

b. Issuing of Certificates

This section describes how the certificates are issued and distributed to the participating nodes. The assumption is made that an authenticated trusted third party (TTP) member exists which plays the roles of an initial certificate authority (CA). This TTP CA is known to all the nodes in the network. The ARAN protocol assumes that certificates are generated by the TTP CA and distributed to nodes before they officially join the wireless ad hoc network. No specific key distribution mechanism is described for the ARAN protocol. Node *i* entering the network will receive a certificate $cert_i$ from the TTP CA that has the following contents:

$$TTP - CA \rightarrow i \ \ : \ \ cert_i = E_{k_{TTP-CA}}(ID_i|K_i|t|et)$$

The certificate $cert_i$ is signed by the private key of the TTP-CA ($k_{CA\text{-}TTP}$) and has the following contents: ID_i representing the identification of node *i* for example a specific IP address; K_i the public key of node *i*; *t* the timestamp for the $cert_i$; and *et* the expiry time of the certificate.

c. Route Discovery Process

The route discovery process provides end-to-end authentication which ensures that the packets sent from a source node *S* reach their intended destination node *D*. Each node maintains a routing table which contains the active communication routes between the different source and destination pairs. The route discovery process begins by a source *S* broadcasting a route request. The route request is signed by the source node's private key k_S and contains: the certificate of the source node ($cert_S$); the identification of the destination node (ID_D); a nonce (N_S); a timestamp (*t*); and a packet identifier identifying that the packet is a route request packet (*RREQ*). The authenticated route request broadcast by node *S* is:

$$S \rightarrow broadcast \ \ : \ \ E_{k_S}(cert_S|ID_D|N_S|t|RREQ)$$

The nonce value is incremented every time the source sends a route request. The nonce value acts like a sequence number ensuring the most recent route request is dealt with. Each node that receives the route request will process it if it has a higher value of the source's nonce than previously received route requests from the same source node. Each intermediate node *P* receiving the route request will validate the signature with the certificate, update the routing table with the neighbour from whom it received the route request, sign the route request and broadcast it to its neighbours. Node *P* will remove the signature and certificate of the previous node if the previous node was not the source itself. Therefore each forwarded route request is authenticated by the source and the intermediated node and will contain two certificates $cert_S$ and $cert_P$:

$$P \rightarrow broadcast \ \ : \ \ cert_P|E_{k_P}(E_{k_S}(cert_S|ID_D|N_S|t|RREQ)$$

The route request is propagated to the destination node *D* which will reply with a reply message *RREP*. The reply packet is signed by the destination node's private key k_D and the packet contains: the identity of the source node ($\mathrm{ID_S}$); the destination's certificate ($cert_D$); a nonce of validity (N_D); a timestamp (*t*); and a packet identifier (*RREP*). The reply packet is unicast along the reverse path toward the source node with a similar authentication procedure to the forwarding of the route request.

$$D \rightarrow reverse\ path \ \ : \ \ E_{k_D}(cert_D|ID_S|N_D|t|RREP)$$

$$P \rightarrow reverse\ path\ \ :\ \ cert_P|E_{k_P}(E_{k_D}(cert_D|ID_S|N_D|t|RREP)$$

The source node will receive the reply packet *RREP* and check the signature and nonce (*N_D*) to verify that the packet was sent by the destination node and not a malicious attacker. If the nonce or certificate fails an error message is broadcast and the route request process restarted.

d. Shortest Path Confirmation

This is an optional procedure employed by ARAN to ensure that the shortest path is found between source and destination. Path confirmation has a high computational cost. After a route has been found between *S* and *D* the shortest path confirmation process begins. The source will broadcast a packet signed by the public key of *D* (K_D) containing: the certificate of the source; the identity of the destination node; a nonce (N_S); timestamp (*t*); and packet identifier identifying that this is a shortest path confirmation packet (*SPC*).

$$S \rightarrow braodcast\ \ :\ \ E_{K_D}(cert_S|ID_D|N_S|t|SPC)$$

Each intermediate node that receive the *SPC* packet updates its routing table, signs the packet, includes its certificate and signs it with the public key of the destination node.

$$P_1 \rightarrow braodcast\ \ :\ \ E_{K_D}(cert_{P_1}|E_{k_{P_1}}(E_{K_D}\left(cert_S|ID_D|N_S|t|SPC\right)$$

The destination node will verify all the signatures and reply to the first and subsequent *SPC* packets with a recorded shortest path packet *RSP*. The *RSP* is propagated to the source which confirms the shortest path by verifying the nonce N_S sent with the *SPC* packet.

e. Maintenance

The ARAN solution uses error messages and implicit revocation of certificates to maintain routes. Error message packets (*ERR*) are broadcast by any node *P* that discovers a broken route. An *ERR* packet is signed by its originator and includes the certificate of the originator, the source and destination pair describing the broken route, a nonce, a timestamp, and a packet identifier. Each node receiving an ERR packet will check its routing table if it contains the accused route. If it does then the *ERR* packet is rebroadcast unchanged.

$$P \rightarrow braodcast\ \ :\ \ E_{k_P}(cert_P|ID_S|ID_D|N_P|t|ERR)$$

The expiration (*et*) attribute included in each certificate allows for implicit revocation of certificates. Certificates are implicitly checked during the route discovery process. Explicit revocation is achieved by the TTP CA broadcasting a certificate revocation message to nodes which then can forward it. Routes are re-calculated as a result of certificate revocation.

Analysis

The ARAN solution uses asymmetric key cryptography to provide authentication, integrity and non-reputation. Asymmetric cryptography will result in high complexity and computational cost. A trusted certificate authority (TTP CA) is required so that authentication can be made available. In the route discovery process unlike AODV, ARAN disallows intermediate nodes which have a path to the destination to reply with a *RREP* message. This creates addition routing overheads but ensures authentication [Sanzgiri et al, 2002].

2.4 Secure Ad hoc On-demand Distance Vector (SAODV)

Zapata et al [Zapata, 2002] proposes the Secure Ad hoc On-demand Distance Vector (SAODV) protocol as a security extension to the AODV protocol. SAODV secures the AODV protocol by using a hybrid cryptographic approach involving asymmetric cryptography in the form of digital signatures and symmetric cryptography in the form of hash chains.

a. System Overview

SAODV defines the fields in a routing packet into two categories mutable and non-mutable. The non-mutable fields are authenticated using asymmetric cryptographic signatures. The only mutable field in an AODV routing packet is the hop count. A new hash chain is created for each route discovery phase which is used to secure the hop count. SAODV requires that the AODV routing packet is extended to include the security information like digital certificates. The implementation of digital signatures and hash chains provides end-to-end authentication for the AODV routing protocol.

SAODV uses asymmetric cryptography and assumes the presence of a key management scheme to distribute keys in a secure manner [Zapata, 2002]. It also assumes that it is possible to verify the relationship between a public key and an IP address or identity.

b. Packet Extension

SAODV proposes an extension to the standard AODV message format so that security mechanism can be implemented. The SAODV extension contains the following fields as described in Table - 1 and Figure 1 [Zapata, 2002].

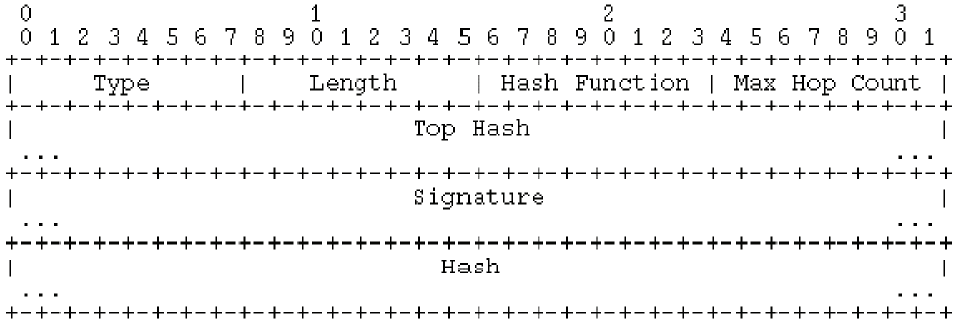
The standard AODV protocol uses packet sizes of 512 bytes but the SAODV extension requires the AODV packet size to be extended to use packets of size 1024 bytes.

c. Route Discovery Process

A source node *S* initiates a route request to a destination node *D* by performing the following steps:

Field	Description
Type	This is a packet identifier where the value 64 identifies a request packet *RREQ* and the value 65 identifies a reply packet *RREP*.
Length	The length of the packet data.
Hash Function	This describes the hash function used for example MD5 [Rivest, 1992] or SHA-1 [Publications F IPS, 208].
Max Hop Count	The maximum hop count (*mhc*) is used for hop count authentication. It defines the maximum number of nodes a packet is allowed to pass through.
Top Hash	The top hash is the result of the hash function *H* applied *mhc* times to a random generated number *x* such that: *top hash* = $H^{mhc}(x)$. Top Hash is vital in the hop count authentication process.
Signature	This field is 32-bit aligned and contains the signature used to authenticate all the non-mutable fields in the AODV packet.
Hash	This field is 32-bit aligned and contains the hash value h_i used to authenticate the mutable hop count variable.

Table 1. RREQ and RREP Signature Extension Fields



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     | Hash Function | Max Hop Count |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Top Hash                            |
 ...                                                         ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Signature                           |
 ...                                                         ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Hash                              |
 ...                                                         ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fig. 1. RREQ Single Signature Extension

1. *S* sets the max hop count (*mhc*) variable equal to the *TTL* (time to live) variable found in the IP header.
2. *S* generates a random number *x* and sets it as the value in the hash field such that $h_0 = x$.
3. The top hash is then generated by applying the hash function *H*, max hop count (*mdc*) times to h_0 such that: *top hash*= $h_{mhc} = H^{mhc}(h_0)$. The hash function *H* is defined in the hash function field in the packet header.
4. *S* digitally signs all the fields in the packet except hop count and hash field and stores the digital signature in the 32-bit signature field.
5. *S* then broadcasts the route request packet to its neighbours.

When an intermediate node receives a route request it verifies the authenticity of the hop count and the integrity of the digital signature. The digital signature is verified using asymmetric cryptography. The hop count is verified by checking $h_{mhc} = H^{mhc-i}(h_i)$, where h_{mhc} is the top hash; h_i is the hash field of the route request; and H^{mhc-i} is the application of the hash function max hop count minus the hop count (*i*) times. The one-way hash chain approach used to authenticate the hop-count is similar to the approach used in the SEAD protocol [Hu et al, 2002]. After the intermediate node verifies the digital signature and hop count, it replaces the packet's hash field with a new hash value computed by applying the hash function to the existing hash value. The intermediate node then rebroadcasts the route request and propagates it until it reaches the destination *D*.

When the route request *RREQ* reaches the destination *D*, *D* checks the validity of the packet and will reply with a reply packet *RREP* if the route request is valid. The *RREP* packet is forward along the reverse route to the source following the same authentication and integrity procedure that the *RREQ* message experienced.

AODV allows for intermediate nodes to also reply to route requests if they have a valid route to the destination node. SAODV proposes two solutions to this security problem. The first is the simplest disallowing intermediate nodes to reply ensuring that the destination node sends the reply message *RREP* and guaranteeing authentication. The second approach uses a double signature extension which allows an intermediate node *P* to reply to a route request from *S* for *D* $RREQ_{SD}$. Intermediate node *P* will reply with a double signature *RREP* message. One signature will sign the intermediate node *P*'s standard reply and the second signature will sign the original *RREP* packet received by *P* for its route to *D*. Both reply message headers are included and sent to the source *S* to establish a secure route to *D*.

d. Maintenance

AODV uses error messages *RERR* to report broken links. SAODV secures these messages using digital signatures. The originator of the error message signs the entire message except the destination sequence number. Each forwarding node also signs the message to prevent unauthorized error messages being broadcast. Nodes using SAODV do not change their destination sequence number after receiving an error message because the error message's sequence number is not authenticated.

Analysis

SAODV authenticates the AODV routing packets preventing certain impersonation attacks. The assumption is made that a node's identity and address can be securely bound to a public key. Such an assumption leaves SAODV vulnerable to Sybil attacks.

SAODV employs asymmetric cryptography which is computationally taxing. The packet size has to be extended to incorporate the security mechanism resulting in a serve communication overhead. For every route discovery a new one-way hash chain is computed resulting in further computational overheads.

The SAODV protocol assumes a key management entity is available in the ad hoc network which can successfully distribute public keys among participants. Such infrastructure is difficult to execute in mobile ad hoc networks and before SAODV is to be implemented either an off-line TTP or distributive key management scheme must be employed.

The SAODV solution uses hash chaining and digital signatures providing security against impersonation routing attacks. It also helps toward preventing denial of service attacks and eavesdropping attacks where malicious users may re-direct a route through a malicious node where eavesdropping may occur.

2.5 Secure Link-State routing (SLSP)

A hybrid scheme is proposed in [Perrig et al, 2001] called Secure Link State Routing Protocol (SLSP). It is a proactive security solution which uses digital signatures and one-way hash chains to provide security to link-state updates and neighbour discovery. SLSP secures link state information in localized manner and can operate alone or be combined with a reactive ad hoc protocol routing protocol like ZRP where SLSP would be the intra-zone routing protocol for the Zone Routing Protocol (ZRP) [Haas & Pearlman, 2001].

a. System Overview

SLSP provides secure neighbour discovery for nodes in a limited hop radius called a zone. Link state updates are authenticated using a hybrid cryptographic method and flooding attacks prevented by a priority ranking mechanism. The main assumption of SLSP is that nodes have existing asymmetric key pairs. SLSP assumes that a key management scheme is present to certify the public keys in the network.

SLSP uses four components: key distribution, secure neighbour discovery, link state update management, and a priority ranking scheme. SLSP's priority ranking scheme prevents denial of service type attacks.

b. Key distribution

SLSP assumes that each node has an existing signed public key before it joins the network; and the certification of keys is performed by an assumed key management method. Public keys are bound to the IP addresses of the network nodes. Nodes then broadcast their public keys into their neighbourhood zone, for example a two hop radius. The received public keys are used to authenticate future packets from the source node.

c. Secure Neighbour Discovery

SLSP uses a Neighbour Location Protocol (NLP) to proactively check that neighbouring nodes do not perform impersonation attacks. Link state information is periodically broadcast by nodes in the form of a NLP hello message. These messages are signed by the source and contain the source's MAC address (a unique hardware address) and IP address (a distinctive network address). A NLP hello message broadcast by source S is described here:

$$S \rightarrow broadcast \;:\; E_{k_S}(IP_S|MAC_S)$$

Notification messages are generated when conflicting link state information is broadcast. An inconsistent mapping of the IP and MAC addresses is considered as conflicting link state information. For example when two nodes with different IP addresses have the same physical MAC address.

d. Link State Update Management

Link state update packets are periodically broadcast to a limited hop radius of nodes. A link state update packet (LSU) contains the IP address of the source, a 32-bit sequence number used for updating and a hop count variable. The LSU hop count variable is authenticated using hash chains as discussed in SEAD [Hu et al, 2002] and SAODV [Zapata, 2002] and the rest of the packet content is authenticated using digital signatures.

When a LSU is received the digital signature is verified using the previously distributed public key. The hash chain is verified and the time to live (TTL) variable is decremented. The hop count authentication protects the LSU packet from travelling too many hops or from link state updates not to be received by nodes.

e. Priority Ranking Scheme

SLSP uses a lightweight flooding prevention scheme which gives priority ranking to nodes. A priority list is maintained for neighbouring nodes which ranks node's priority based on the number of link state updates a node broadcast. Malicious nodes will flood the network with link state update packets to cause resource and denial of service attacks. SLSP gives high priority to nodes that send less link state updates limiting the affects of flooding attacks.

Analysis

The Secure Link State Routing Protocol is a hybrid cryptographic scheme using digital signatures to provide authentication for its NLP hello messages and link state update (LSU) packets. Hash chains are used to authenticate the limited hop broadcast of LSU. Impersonation type attacks are prevented by monitoring the IP and MAC address bindings of neighbours through a neighbour location protocol (NLP). Link state updates are authenticated using digital signatures and hash chains. Flooding type attacks are prevented using priority ranking.

The SLSP protocol provides security to topology discovery but cannot act as a standalone security mechanism as it lacks a data transmission protection agent. Nodes that securely join the network can misbehave during data transmission without being detected or prevented.

SLSP lacks a disciplining agent like a revocation mechanism. For example a malicious node *B* can impersonate another node *A* and flood *A*'s neighbours with LSU packets. SLSP's priority mechanism will limit the effectiveness of the flooding attack. The NLP protocol will detect the impersonation attack but node *A* has no mechanism to correct to the attack and *A* will remain with a low priority.

2.6 On-Demand Secure routing Byzantine Resilient routing protocol (ODSBR)

The on-demand secure routing byzantine resilient routing protocol (ODSBR) is proposed in [Hu et al, 2003b]. Byzantine behaviour is defined by the authors as any action taken by an authenticated node to disrupt the routing procedure. ODSBR is a secure reputation based routing protocol that prevents the effects of byzantine failures on successful routing.

a. System Overview

ODSBR uses weighted paths to select routes in the route discovery process. Paths are assigned weights based on a fault path detection method. A high weight is assigned to an unreliable path. ODSBR is divided into three components: route discovery process, fault detection, and weight path management. ODSBR assumes that a public key infrastructure is present to manage public key authentication.

b. Route Discovery Process

Routes are discovered in an on-demand manner like in DSR. ODSBR extends the standard route request *RREQ* by adding a weight list instead of a node list like in DSR. A weight list includes the list of chained nodes with their associated weights. These weights are defined by link failures detection mechanism. The *RREQ* is signed by the source and broadcast into the network updating its weighted list after each hop until it reaches the destination. The destination then verifies the signature and broadcasts the reply message *RREP*. Each node that receives a *RREP* will then calculate the total weight of the path by summing the weights of the specific path to the current node. The *RREP* forwarded if the total weight is less than the total weight of any previously forwarded *RREP*. Before an intermediate node *P* forwards a suitable *RREP* all the signatures are verified and node *P* appends its signature. The source node receives the *RREP* and calculates the total weight and verifies all the signatures.

c. Fault Detection Method

Each node *i* has a list of probe nodes. Each probe node sends node *i* an acknowledgement message for each data packet *i* sends. If a threshold *t* of acknowledgements is not received a fault accusation is logged against a specific path. Using a binary search technique ODSBR identifies a path as faulty after *log(n)* fault accusations, where *n* is the length of the accused path.

d. Weight Path Management

A low weight is associated with a secure path. The weights associated to paths are influenced by two factors: time and fault detection. When ODSBR identifies a path as faulty based on the fault detection method then the weight for that path is doubled. Path weights are halved after a counter reaches zero, each path has an associated counter.

e. Analysis

The on-demand secure routing protocol (ODSBR) provides ad hoc on-demand routing with byzantine failure prevention. Weights are assigned to paths by a fault detection method and paths are selected based on the weights. The Secure Routing Protocol (SRP) proposed in [Papadimitratos & Hass, 2002] introduces the metric specific path selection method but this proposal is not a standalone secure routing protocol like ODSBR [Awerbuch et al, 2008].

ODSBR assumes the existence of a public key management system. ODSBR further assumes that a shared key exists between source and destination nodes to ensure authenticity and integrity of acknowledgement messages sent by probe nodes. This helps avoid expensive asymmetric per packet computations for acknowledgement messages.

The route discovery process of ODSBR broadcast reply messages instead of unicasting them which results in a computationally expensive operation. This method will result in $2^{\frac{n}{2}+1}-1$ reply packet transmissions where *n+2* is the number of nodes in a path from node *A* to *B* including nodes *A* and *B* [Awerbuch et al, 2008]. Furthermore the cost monitoring of data packet transmission is computationally high because the fault detection method requires a threshold of *t* probe nodes to reply with an acknowledgement for every data packet sent.

ODSBR is identified by authors [Awerbuch et al, 2008] to be vulnerable to wormhole attacks. The wormhole attack may be avoided in the case where the wormhole link node exercises byzantine behaviour.

2.7 Reputation based CONFIDANT

The CONFIDANT protocol representing the ‘Cooperation Of Nodes: Fairness In Dynamic Ad Hoc Networks’ [Buchegger & Boudec, 2002] is a reputation based solution which operates over the DSR routing protocol.

a. System Overview

The CONFIDANT solution does not use any cryptographic techniques to achieve secure routing. The system is solely reputation-based and operates in an on-demand ad hoc network environment as an extension of the DSR routing protocol. Each node in the ad hoc network is required to be involved in the four components of CONFIDANT: monitoring, trust management, reputation system and path management.

b. Monitoring

Each node is a monitor and is responsible for the packets that it sends or forwards. For every packet that a node forwards it watches that the next hop node forwards the packet properly. The monitor looks for inconsistent behaviour and triggers an alarm to the reputation system if misbehaviour is discovered.

c. Trust Management

The trust management system manages the alarm messages. The alarm messages are generated by each node’s monitoring system and exchanged between other nodes as to build and maintain a local rating list. The trust management manages the input of alarm messages and assigns more influence to alarm messages that come from trusted nodes and less influence from other nodes. CONFIDANT assumes pre-existing relationships between a selection of nodes called friends [Buchegger & Boudec, 2002], friend nodes are highly trusted nodes. Local rating lists are exchanged as well and their influence managed by the trust management system.

d. Reputation System

The reputation system manages and maintains the local rating list. This list contains node identities and corresponding rating. A rating will correspond to the amount of misbehaviour a node has displayed. The ratings will be updated from alarm messages and direct observations.

e. Path Management

Paths are selected based on a rating threshold, local rating lists and a blacklist containing all untrusted nodes. A node is blacklisted when its rating is below the rating threshold *t*. The path manager removes the paths in the network which contain the blacklisted node. The path manager manages the route discovery process by reacting to route requests from blacklisted nodes or route requests that have passed through a blacklisted node.

f. Analysis

CONFIDANT is an exclusively reputation based routing protocol. Local rating lists of node's behaviour is recorded and used during the route discovery process. The authors note that only negative evidence is gathered against nodes so nodes can only be identified as less trusted as the network continues. Like most reputation based schemes a counter system is employed where each rating list entry has an associated counter. When the counter reaches zero the rating is reset to the default state of null misbehaving accusations. The CONFIDANT protocol assumes the existence of prior trust relationships between a selected number of nodes called friends [Buchegger & Boudec, 2002].

2.8 Discussion

Several different secure routing protocols are presented in Section- 2.2 they differ in the areas of security and operational requirements. The diversity of their design makes it difficult to compare their success but this section outlines the diverse characteristics of the presented protocols.

a. Security Analysis

The proposals can be categorized by the security techniques which include the asymmetric, symmetric and hybrid cryptographic security approaches. The last category is the reputation based solutions. The security mechanism of each protocol is presented and the attacks which the protocol protects against are highlighted. A summary of the evaluation is presented in Table -2.

Symmetric Cryptography

The symmetric cryptographic approaches include the Secure Efficient Ad Hoc Distance Vector Routing protocol (SEAD) and the Ariadne protocol. Hash functions and hash chains like SHA-1 [Publications F IPS, 2008] and MD5 [Rivest, 1992] are used for authentication purposes usually for the hop count variable. The hash function is lightweight compared to asymmetric security techniques.

The SEAD approach uses a hop-by-hop authentication technique. SEAD authenticates the sequence number and hop count of routing packets protecting the routing procedure from resource consumption attacks for example denial of service attacks, route table poisoning, and replay attacks.

The Ariadne protocol is proposed by the same authors of SEAD. Ariadne uses message authentication code (MAC) to provide end-to-end authentication between communication nodes. Ariadne protects against similar attacks to SEAD but uses end-to-end authentication.

Protocol	Security Approach	Techniques	Attack Prevention
SEAD [Hu et al, 2002]	Symmetric Cryptography	• Hop-by-hop authentication • Hash chains	• Resource consumption • Denial of Service • Route table attack • Replay
Ariadne [Hu et al, 2005]	Symmetric Cryptography	• End-to-end authentication • Hash chains	• Resource consumption • Denial of Service • Route table attack • Replay
ARAN [Sanzgiri et al, 2002]	Asymmetric Cryptography	• End-to-end authentication • Certificate Authority	• Route table attack • Replay attacks
SAODV [Zapata, 2002]	Hybrid Cryptography	• End-to-end authentication • Hash chains • Digital Signatures	• Denial of Service • Route table attack • Replay
SLSP [Papadimitratos & Hass, 2003]	Hybrid Cryptography	• Secure neighbour discovery • Authenticated link state updates	• Denial of Service • Route table attack • Replay
ODSRP [Awerbuch et al, 2008]	Reputation Based	• Path specific reputation lists • Digital Signatures	• Denial of Service • Route table attack • Replay • Byzantine Failures
CONFIDANT [Buchegger & Boudec, 2002]	Reputation Based	• Node specific reputation lists	• Black Hole • Replay

Table 2. Summary of security analysis for secure routing in ad hoc networks

Asymmetric Cryptography

The only solely asymmetric cryptographic approach investigated is the Authenticated Routing for Ad hoc Networks protocol (ARAN). Asymmetric cryptographic is computationally costly compared to symmetric cryptography and it requires the existence of a trusted third party or self organized key management system.

ARAN provides end-to-end authentication for an on-demand mobile ad hoc network. ARAN provides authentication and protecting from replay attacks and unauthorized routing table attacks. ARAN is vulnerable to flooding attacks. A malicious node can flood nodes with fake routing packets signed with illegitimate keys this will result in many unsuccessful verifications and ultimately denial of service and resource consumption.

Hybrid Cryptography

The SAODV and SLSP protocols are hybrid solutions which employ both asymmetric cryptography and symmetric cryptography. The common approach is for all the mutual fields to be digitally signed and the immutable fields, like the hop count, to be protected using hash chains. The Secure Ad Hoc On-demand Distance Vector protocol (SAODV) employs this tactic to provide end-to-end authentication but at the cost of extending the routing packet header. SAODV protects against impersonation attacks on the routing protocol. It also helps prevent replay and denial of service attacks.

Secure Link State Routing Protocol (SLSP) provides secure neighbour discovery and authenticated link state updates but lack a secure data transmission protocol. The Neighbour Location Protocol of SLSP protects against impersonation type attacks where malicious nodes adopting conflicting IP and MAC addresses would want to corrupt the routing table. Furthermore flooding attacks, which result in resource consumption and a denial of services, are prevented by a priority ranking scheme.

Reputation Based

Reputation based or conduct based systems allow for a nodes behaviour in the network to affect its assigned security or trustworthiness. Reputation based protocol can be computationally costly because they usually require packet monitoring systems and the proactive exchange or behavioural evidence between nodes. The On-demand Secure Routing Protocol Resilient to Byzantine Failures (ODSBR) and the CONFIDANT protocol are reputation based systems.

ODSBR uses reputation based system to select the most secure routes. A fault detection method monitors the success of each packet transmission and faults are logged against specific paths. Reputation is path specific in ODSBR. ODSBR couples with asymmetric cryptographic approach to provide end-to-end authentication along the selected secure path. The CONFIDANT uses exclusively reputation based techniques to provide security. Similarly to ODSBR only negative evidence is considered. Nodes monitor every packet which they forward and maintain a local rating list. Reputation or ratings are node specific unlike ODSBR. Both CONFIDANT and ODSBR monitor packet forwarding this will protect the system from black hole attacks. Replay attacks which use the method flooding are prevented using path reputation and node reputation in ODSBR and CONFIDANT respectively. A disadvantage of the negative reputation approach for ODSBR and CONFIDANT is that black list nodes or faulty path entries have an expiration time after which their confidence is reinstated. This allows malicious nodes to continue disrupting the network until they are caught again.

b. Operational Requirements

The presented secure ad hoc routing protocols have certain assumptions that each makes to realize its design. Furthermore protocols are designed specifically for operation in specific routing environments. This section summarizes the operational requirements of the presented secure ad hoc routing protocols. Table -3 summarizes this discussion.

The symmetric cryptographic approaches do not rely on a public key infrastructure but still require some kind of key management in the ad hoc network. The SEAD protocol is designed for a table-driven routing protocol and is based on the DSVD routing protocol. SEAD requires a key management mechanism to distribute an authenticated initial hash element. Ariadne is a DSR based on-demand protocol which assumes there are shared secrets between each communication pair. The shared keys are used in TESLA authentication and a key management system is assumed present to distribute the keys. TESLA authentication also requires time synchronization between each node. This is difficult without the presence of an online TTP.

ARAN, SAODV, SLSP and ODSBR use asymmetric cryptography and key management is simply assumed for each of these protocols. ARAN assumes that an online TTP is present that acts as a certificate authority (CA). Prior shared secrets are assumed between all participating nodes and the CA. ARAN is an on-demand protocol. SAODV protocol is based on the AODV on-demand routing protocol and assumes the presence of a key management system to distribute keys. SLSP assumes that nodes enter the network with asymmetric key pairs and a key management scheme is present to certify the keys in the network. SLSP is a table based routing solution. ODSBR is based on DSR routing and authenticates its routing packets with digital signatures and a public key infrastructure is assumed to manage the keys. Shared keys are also assumed to allow for authentic acknowledgement message communication between source and probe nodes.

Protocol	Routing	Assumption
SEAD [Hu et al, 2002]	Table Driven DSDV based	• Key management system to distribute an authenticated element for hash chaining
Ariadne [Hu et al, 2005]	On-Demand DSR based	• Time synchronized network • Shared secret key between each node pairs for MAC • Key management system to manage TESLA keys
ARAN [Sanzgiri et al, 2002]	On-Demand Not protocol specific	• TTP acting as a certificate authority (CA) • Prior shared CA public key
SAODV [Zapata, 2002]	On-Demand AODV based	• Key management scheme • Secure IP public key binding
SLSP [Papadimitratos & Hass, 2003]	Table Driven Not protocol specific	• Nodes have existing asymmetric key pair • Key management system
ODSRP [Awerbuch et al, 2008]	On-Demand DSR based	• Key management system • Shared keys between source and probe nodes
CONFIDANT [Buchegger & Boudec, 2002]	On-Demand Not protocol specific	• Pre-existing relationships between a selection of nodes called friends

Table 3. Operational requirements of the present secure routing protocols

CONFIDANT does not use cryptographic techniques and does not require the existence of a key management scheme. CONFIDANT does assume pre-existing relationships between a small number of nodes called friends. The CONFIDANT solution is designed for on demand routing.

From this discussion, the conclusion is made that most secure ad hoc routing protocols assume the existence of a key management system to certify, authenticate, and distribute keying information. Mobile ad hoc networks cannot assume the existence of a TTP and must address the problem of key management.

3. Proposed security scheme: Direct Indirect Trust Distribution (DITD)

A security establishment scheme is proposed for a mobile ad hoc network. Key management is central to the establishment of trust in these networks. The proposal focuses on key management. The proposal is for a mobile ad hoc network, which operates in a self-organized and fully distributive network environment. These networks allow for nodes to join and exit the network, unrestricted. These networks find application in the military and commercial filed. For example application can be found in, tactical positional networks for military based communication or personal area networks for secure peer-to-peer data and file sharing. The proposed protocol is planned for these self-organized distributive networks. It is noted that these networks do not allow for rigorous access control. The proposed protocol can be extended to allow access control services.

In a self-organized mobile ad hoc network, there is no presence of a separate authority member, such as a trusted third part or certificate authority. Instead each node that enters the network is considered the security authority of its own domain. Security is established by nodes creating and issuing certificates which bind nodal identities to their respective public keys. These certificates are issued and distributed in order to realize secure communication. A bi-directional security association is made between nodes *A* and *B,* when node *A* holds a certificate binding the public key of *B* and *B*'s identity; and *B* holds the certificate binding the public key of *A* and *A*'s identity. Malicious adversary nodes that wish to disrupt communication will target the network layer, and more specifically the routing mechanism, as identified in the previous chapter. The network layer is identified as the sphere of design.

The problem is then to provide secure communication, which is implemented on the network layer. Secure communication is achieved when node *A* is able to set up a secure communication channel, where no other entity can interrupt or eavesdrop on its communication with node *B*. The question as to whether node *B* is worthy of trust, is not the concern. That question must be decided by the nodes themselves, based on available trust evidence. The proposal made on the network layer aims to provide the most secure route between *A* and *B*, preventing malicious adversaries from sabotaging communication.

The term trust is defined as the "belief by a trustor with respects to the competence, honesty, security and dependability of a trustee within a specific context." [Grandison, 2003]. There are two trust variables: direct trust and indirect trust. Direct trust is a result of independent or local trust evaluation between two immediate nodes. Indirect trust is evaluated using the advice from other nodes. In the context of certificate base trust, direct trust is defined as trust between local neighbours. Indirect trust is created by certificate chaining.

A hybrid trust model is proposed, uniting certificate and conduct based trust to provide a more secure communication. A key management model is also proposed. This model supports an existing routing protocol. The proposed scheme is called Direct, Indirect Trust Distribution (DITD) and it follows the following procedure: direct trust is established by requesting that all nodes involved in the route discovery stage, share their self certificates with each others' one-hop neighbours involved in the route discovery phase. Indirect trust is further established by requesting that the sender's self certificate propagates with the route request towards the destination. The routing messages trigger certificate distribution allowing direct trust relationships between one-hop neighbours. These trust relationships are then chained together providing a trusted route to the destination node. Keying material is allowed to be propagated along these chains of trust. A disadvantage of the self-organized nature of these networks is that the established security of trust chains will rely on transitive trust [Capkun et al, 2003]. The DITD model proposes coupling the security provided by the certificates with a conduct based trust analysis model. Conduct trust is affix, which allows for more secure communication. This is achieved by calculating the trust of routes, based on the conduct of the nodes involved, and selecting the most trusted route for communication.

3.1 Related work

A detailed survey was presented on key management schemes for mobile ad hoc networks in the previous chapter. Section-2 focused on the network layer and presented a survey of existing secure routing protocols. This section provides work directly related to the DITD model.

The authors of [Capkun et al, 2003] propose a completely self organized public key system for mobile ad hoc wireless networks. This is a PGP based solution which provides key management in ad hoc networks without the presence of an off-line or on-line authority, like a CA, TTP or server. Each node distributes its self certificates and maintains its own certificate repositories. Nodes participating in the network share their certificate repositories and repository updates are preformed in a proactive manner. Certificates are reciprocally authenticated and trust chains formed linking remote nodes to each other. Security is realized on the application layer.

Zapata [Zapata, 2006] addresses the issue of verification delays in secure mobile ad hoc networks. Zapata proposes a protocol to optimize the number of verifications made in a single secure route discovery phase. Once a route is established only then are the shared certificates verified. This helps in reducing the computational overhead of verifications on multi-hop paths. By reducing the total number of verifications made in a network's life time there is a resultant end-to-end delay upon the delivery of routes.

Theodorakopoulos *et al.* [Theodorakopoulos & Baras, 2006] proposes a fully distributive conduct based trust model which has PGP characteristics. This model operates on the application layer and allows for trust to be established without the presence of a central authority member. PGP models share certificates to establish trust while the work proposed in [Theodorakopoulos & Baras, 2006] allows for other trust evidence, like conduct and location, to influence the trust establishment. Trust is fully distributed in a proactive manner allowing all nodes to give trust opinions about other nodes.

Semiring mathematics presented in [Kscischang et al, 2001] has more recently been used to model trust calculations in [Theodorakopoulos & Baras, 2006]. Trust opinions are mathematically aggregated along a path and trust decisions are mathematically represented. The work in [Theodorakopoulos & Baras, 2006] uses Dijsktra's extended algorithm proposed by Mohri [Mohri, 2002] to include trust. This finds the most trusted path between two remote nodes in a proactive manner.

The majority of literature mentioned function in a proactive manner for application layer solutions. The DITD model is designed on the network layer for a reactive, fully distributive, self organized, mobile ad hoc network environment. The ideas of some of these protocols have inspired the creation of the DITD model and the impact of these protocols is discussed in Section-4.

3.2 Proposed security scheme

The aim is to design and investigate a security mechanism to specifically provide: *public key certificate distribution*, *optimal verification*, and a *conduct trust model* to optimize trust decisions.

The security mechanism is to provide secure communication in a mobile ad hoc network environment while satisfying the following requirements based on environment and functionality.

a. Design Requirements

Environment

- **Network layer design:** The security mechanism is to be implemented on the network layer protecting these dynamic networks from attacks and avoiding multi-layer design.
- **Self organised:** Nodes are responsible for their own security services, including the distribution of keying information.
- **Fully distributive:** The certificate distribution scheme is to be designed in a fully distributive manner where all nodes participate in the operation and implementation.
- **On-demand:** The DITD model is to be design in an on-demand environment optimizing the limited resources of ad hoc networks. On-demand models provide security to nodes upon request. The proactive approach provides security to an entire network at once and requires computationally taxing periodic updates.

Functionality

- **Distribution of keying material:** DITD is to provide direct and indirect trust relationships between local and remote nodes by efficiently distributing self certificates between nodes.
- **Minimize the overhead:** DITD aims to minimize the overheads upon the network routing performance while still providing trust establishment. DITD aims to avoid alterations to the routing control packets and strives for independence between routing and trust establishment.
- **Provide secure communication from the start:** Secure communication is requested from the start to the end of the network lifetime unlike the model proposed in [Tanabe & Aida, 2007] which is flawed by an initial setup phase with weak security.
- **Trust evaluation mechanism:** Security should be supported by a trust evaluation mechanism allowing for more secure routes to be established and ensuring the secure distribution of certificates.
- **Robust in the presence of topology change:** The DITD model should be robust to poor connectivity and routing failure due to changing mobility which is an inherent characteristic of a mobile ad hoc network.

b. System Model

To fulfil the constraints given in above, we assume the following system model. There is no pre-existing infrastructure and no online trusted third party present during communication. The model is a fully distributive network of wireless nodes using an ad hoc on-demand routing mechanism. It is assumed that nodes have their own keying material before joining the network generated by a fully self organized mobile ad hoc network [Capkun et al, 2003], or by an off-line authority issuing keying material before a node enters the network for example in [Capkun et al, 2006]. Each node is assumed to have a public and private key pair; a self certificate binding the public key and user identification of the node; and a set of network security parameters common to all nodes in the network. Secure communication is requested from the start to the end of the network's lifetime. Users can join and leave the network without any restrictions. Any user with the correct keying material may participate in the network. It is assumed that conduct information is available to each node from node monitors [Tseng et al, 2003].

The DITD model uses certificate based trust coupled with conduct based trust to develop a hybrid trust protocol maximizing trust in the network. The DITD model addresses the issues of *key exchange*, *verification protocol* and *conduct trust evaluation*. It is designed on the network layer accompanying an on-demand routing protocol. Figure 2 describes the high-level system model.

The DITD scheme performs the task of *key exchange*, exchanging self certificates on the network layer following an on-demand routing mechanism. Direct trust is established by self certificate exchanges among one-hop neighbours, triggered by the route discovery process. This allows for a bi-directional security association to be made between immediate nodes, we refer to this relationship as direct trust. Direct trust associations are chained together creating trusted paths and allowing for two nodes, out of each other's communication range, to exchange self certificates. This describes the *key exchange* element of the DITD model. It is divided into two parts: the exchange of direct and indirect trust relations. Figure 3 illustrates the direct, indirect trust relations established by DITD.

An on-demand route discovery phase will flood the network with route requests in search of a destination. This will couple with the *key exchange* mechanism of DITD. The result of this is a flood of self-certificate exchanges. Verification of these certificates is optimized by the DITD model. Trust chains will have an accumulative certificate verification delay because possibly each direct trust association will need to be verified. The DITD model proposes a *verification protocol* which optimizes and manages the verification process.

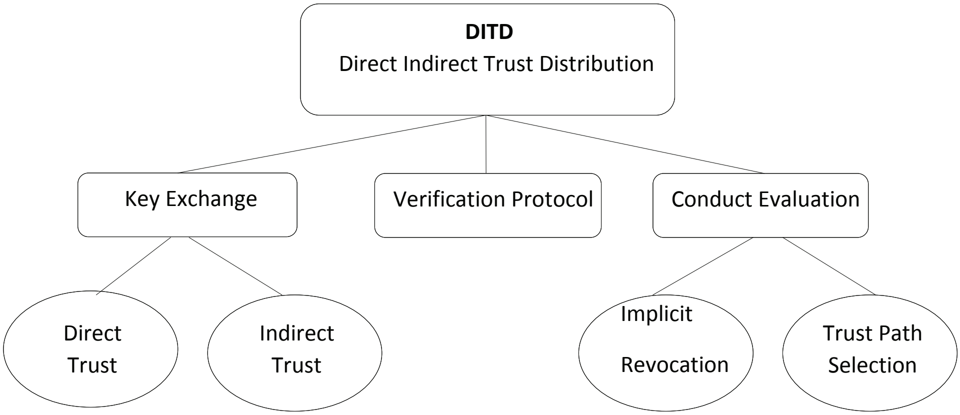


Fig. 2. High level system model

DITD uses the existence of conduct trust evidence to maximize the security provided by public key certificates. Trust is aggregated in a reactive manner using semi-ring mathematics and a reactive shortest path algorithm. The most trusted routes are selected by a *conduct evaluation protocol* which includes an implicit revocation mechanism and trust evaluation metric. Direct and indirect trust establishment is strengthened by DITD's conduct trust evaluation.

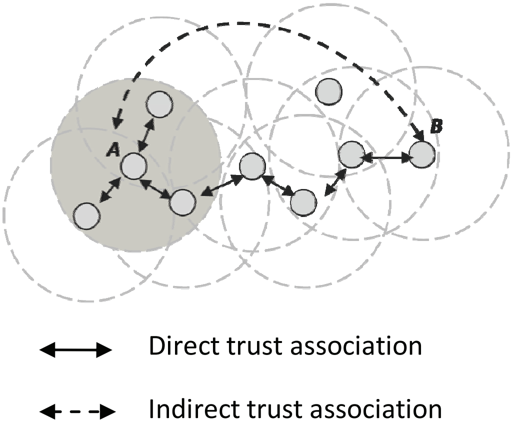


Fig. 3. Direct and Indirect trust establishment in DITD model

c. Key exchange

A certificate trust model is used to create trust between nodes. As an example, a secure bidirectional communication path can be setup between node *A* and *B* if both nodes have exchanged each other's self certificates as seen in Figure 4. The self certificate binds the user *ID* and the public key. So the exchange of self certificates allows keys to be exchanged authentically. Node *A* can trust node *B* if node *A* has a certificate verifying the identity of node *B*, and similarly concerning node *B* with respects to communication with node *A*.

The DITD certificate trust model appends an existing on-demand mobile ad hoc routing protocol. Its principals can be applied to any on-demand routing scheme. Knowledge of the operation of an ad hoc routing mechanism will help visualise the explanation of DITD.

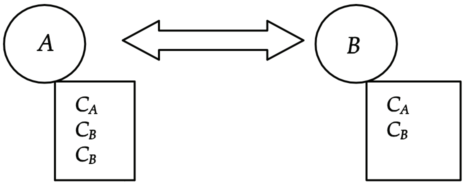


Fig. 4. Secure bidirectional communication between node A and B

The application of an ad hoc on demand routing procedure is briefly revisited to aid the DITD explanation. The AODV [Perkins et al, 2003] routing procedure follows three stages during route discovery:

1. Sending of request message (*RREQ*)
2. Receiving of request message (*RREQ*)
2. Sending reply message (*RREP*)

In stage one, the source node *A* request communication with destination node *B* by broadcasting a routing request $RREQ_{AB}$ into the network. This request is forwarded and propagated through the network to *B*. The *RREQ* message may have been sent by *A* or forwarded by an intermediate node P_i. When the *RREQ* is received by an intermediate node

P_i, stage two begins. At stage two a reverse route to *A* is then set up and P_i checks if it is the destination *B* or has a fresh route to the destination node *B*. If not then the *RREQ* is further broadcast by P_i and propagates until the destination is found. When the destination or a fresh route to the destination is found stage three commences. At stage three a reply message *RREP* is propagated along the reverse route until it reaches the source node *A* establishing the communication route.

When a node receives a routing control packet, certificate requests are triggered and sent using separate unicast messages. The certificate distribution is added at stage two and stage three, the receiving of a route request and the sending of a reply message respectfully. In Table-4 we define the symbols we used next in our explanation.

P_i	intermediate node *i* receiving the *RREQ*
P_{i-1}	previous node P_{i-1} who forwards *RREQ* to its neighbour P_i
A	originator node of RREQ message
B	destination node of RREQ message
$Cert_i$	certificate of node P_i

Table 4. Definition of symbols

At stage two upon receiving a route request packet, before this packet is processed and the routing table updated, direct trust and indirect trust establishment is set up.

Direct Trust

At stage two, direct trust relationships are made by sharing neighbouring nodes self certificates. When intermediate node P_i receives a route request *RREQ* it first checks its certificate repository for the certificate of the neighbour who forwarded the request, P_{i-1}. If it does not possess such a certificate, $Cert_{i-1}$, a local self certificate exchange is done between node P_i and its previous hop neighbour P_{i-1} as follows: a *unicast* message is sent from P_i to P_{i-1} with P_i's self certificate $Cert_i$ appended; P_{i-1} receives the message, updates its certificate repository and replies with a unicast message to P_i containing P_{i-1}'s certificate. If the certificate $Cert_{i-1}$ is found in P_i*'s* certificate repository there is no need for a self certificate exchange. This procedure follows the *RREQ* as it floods the network in search of a route to the destination node. Direct trust establishment is illustrated in Figure 5. This, as it is expected, causes an increase in control packet overhead as the DITD model transmits additional certificate packets into the network.

A second direct trust establishment approach is proposed, which exploits the *HELLO* packets of the AODV routing protocol. The AODV protocol uses periodic one-hop broadcasts packet to maintain and establish communication between neighbouring nodes. The DITD model proposes that direct trust can be established independent of the route discovery process by including a certificate query in the *HELLO* packets. When a $HELLO_A$ packet is broadcast by node *A* and received by node *B*, the receiver *B* checks its certificate repository for the certificate $cert_A$. In a similar way to the first approach if no certificate is found a localized certificate exchange is performed. The certificate exchange messages are independent from the *HELLO* packets. The second approach allows for direct trust establishment with the least amount of dependence on the routing procedure.

Indirect Trust

Still at the stage two (receiving the routing request) indirect trust is established between remote nodes *A* and *B* by the exchange of remote self certificates. Similarly to the direct trust set up, before node P_i processes the received routing request *RREQ* a certificate search and exchange is made. Node P_i searches for the source *A*'s certificate, $Cert_A$. If the certificate is not found, P_i sends a separate unicast certificate request for $Cert_A$ to the previous node P_{i-1}, whose address can be found at the next hop on the reverse route in the routing table. This addition allows for the source's certificate $Cert_A$ to be propagated towards the destination *B*. It is noted that by not appending the certificate to the route requests this reduces dependency between the route establishment and certificate trust establishment.

For indirect trust to be complete between nodes *A* and *B*, the source *A* is required to possess the destination's certificate, $Cert_B$. Further additions to stage three, sending the reply message, are required to complete the indirect trust establishment. A reply is sent is sent under two conditions. Firstly when the destination node is found and secondly when a fresh route to the destination node is found.

For the first condition, the reverse route to the source *A* is already setup with localized direct trust existing between nodes on the route. Therefore a trusted chain of nodes is available from *B* toward the originator node *A*. All that is required is for the certificate of the destination node, $Cert_B$, to be piggy backed on the routing reply message *RREP* toward *B*. Each intermediate node stores $Cert_B$ and updates its certificate repository and the forward route to *B*.

For the second condition, if a fresh route to *B* is found at P_i, there exists a route from the intermediate node P_i to the destination *B* and a route from P_i to the source *A*. Both routes have localized direct trust existing already. Two *RREP* messages are then propagated, one toward *B* with $Cert_A$ appended and one toward *A* with the $Cert_B$ appended. Indirect trust is therefore set up by certificate chaining as illustrated in Figure 5.

c. Verification Protocol

For trust to be established between two entities they must not only share the certificates but the certificates must be verified for the users to be authenticated. Ideally verification will take place immediately after a certificate is received but a single verification can take up to 1ms delay [Stephan Eichler, 2006] on 1024-bit RSA key, and even more for a ECC key. These verifications can accumulate across multi hop routes. For application specific networks that are time dependent like audio applications and military automation networks a delay of milliseconds is critical. A requirement of DITD is for the security additions not to cripple or delay the existing routing mechanism.

Verification for direct trust establishment can be done immediately without incurring a delay upon the routing mechanism. This is because the localised certificate messages are separate and independent from the request messages. Furthermore during route discovery, request messages (*RREQ*) can be forwarded without waiting for verification to be processed [Zapata, 2006] as verification can be confirmed on the reply route. Delayed confirmation of verification is not possible for the reply message (*RREP*) because the exchange of the destination node's certificate, $Cert_B$, follows the *RREP* message. The $Cert_B$ certificate must be verified before the *RREP* message can be securely forwarded and trusted routes established. This means that a certificate trust chain will have an accumulative processing delay due to verifications. Therefore the problem is that the verification of the destination certificate $Cert_B$ may cause a delay in route establishment because $Cert_B$ is distributed with the *RREP* message.

A solution to this is that if any intermediate node has $Cert_B$, it can distribute $Cert_B$ to the reverse route, during *RREQ* message propagation. When a *RREQ* message is forwarded a

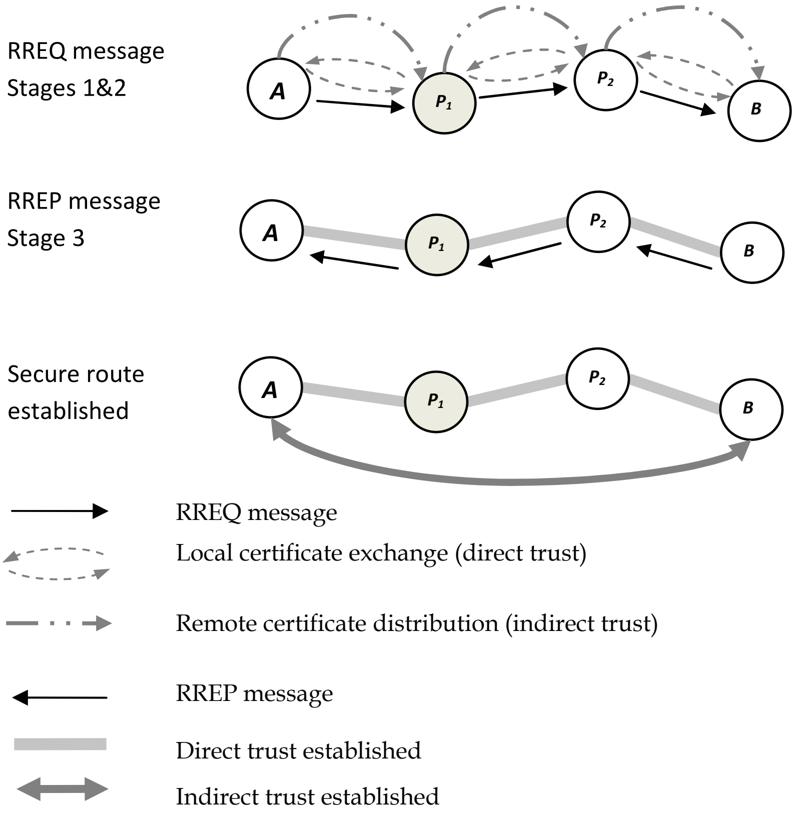


Fig. 5. Illustrating the certificate exchange protocol

flag is appended identifying if the forwarder has the destination certificate $Cert_B$. Intermediate node *P* receives the *RREQ* message and updates the reverse route entry with $flag_{cert}$ indicating if the previous hop has $Cert_B$. *P* checks if it has $Cert_B$ in its certificate repository and assigns an appropriate value to $flag_{cert}$ before forwarding the *RREQ* message. If *P* has $Cert_B$ and the reverse route variable $flag_{cert}$ indicates that the previous hop does not have $Cert_B$ then *P* sends a unicast certificate message containing $Cert_B$ to the previous hop, whose identity can be found from the reverse route in the routing table. The $Cert_B$ is propagated along the reverse route by checking the routing table entry $flag_{cert}$ and responding in a similar fashion. This allows the destination certificate $Cert_B$ to be distributed during the route discovery phase independent from route establishment. The verification protocol is illustrated in Figure 6 where source *A* sends a *RREQ* for destination *B* and intermediate node P_2 has $Cert_B$ but no route to *B* itself. In this case while route discovery continues $Cert_B$ is transmitted to the nodes in the reverse route which do not have $Cert_B$, these nodes are indicated by the $flag_{Cert}$ variable. Certificate verification is done concurrently with route discovery therefore minimising the amount of verifications that delay the route discovery. Outstanding verifications are done following the *RREP*. Verification checks are preformed with the *RREP* message. Figure 6's example allows for three less verification delays at P_2, P_1 and *A* because of back tracked verification.

The condition under which the above will be most effective is when a node has received $Cert_B$ during a previous route establishment but it no longer has an existing route in its routing table for *B*. Such an occurrence is a result of a node previously involved in a route to *B* but due to route expiry, loss of connectivity or node mobility the node is no longer part of such a route. Therefore the benefits of the verification protocol are most evident in ad hoc networks with moderate to high speeds. The DITD model implements verification optimization to reduce the delay incurred on the routing mechanism by verification.

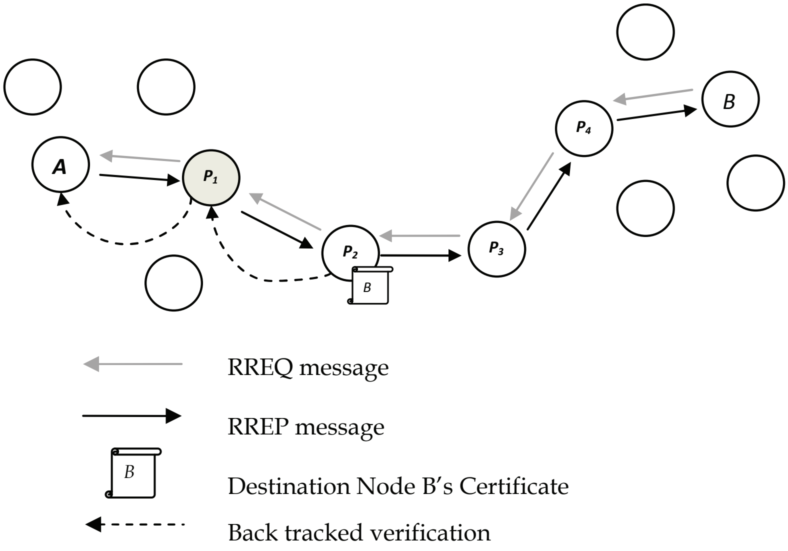


Fig. 6. Illustration of verification protocol

The authentication protocols ARAN, SAODV, SEAD or Ariadne discourage intermediate nodes with a route to the destination to reply to route requests. If the DITD model would be used in conjunction with such a protocol then although intermediate node with a validate destination certificate are unauthorized to reply DITD maximizes the availability of the destination certificate. The verification protocol would be used to distribute the destination's certificate it along the reverse path and perform verification checks so lesser time delay is incurred from the route reply message.

Direct and indirect trust establishment is realised through the route establishment phase of the ad hoc routing scheme. During the initial stage of route establishment the network is flooded with routing requests and in turn certificate exchange messages. It can be expected that there will be a large packet overhead as a result of additional certificate packets.

Mobility produces erratic connectivity problems and unexpected routing failure. Multi-hop routes are vulnerable to failure under increased mobility while localised one-hop route connections are less vulnerable. If the proposed solution was dependent upon such multi-hop routes, like [Capkun et al, 2006] is, it would suffer severely from inherit link breakages common to highly mobile networks. The proposed solution prevents the certificate exchange procedure from using multi-hop routes by exchanging certificates in a strictly localized manner. This allows the DITD certificate distribution scheme to operate in ad hoc networks with varied mobility's and changing connectivity without the worry of routing failure interfering with security.

d. Conduct Trust Evaluation

Providing conduct based trust enhances the trust decision made by nodes and therefore effect keying decisions: "Conduct trust influences decisions like access control, choice of public keys, etc. It could be useful as a complement to a public key infrastructure (PKI), where an entity would accept or reject a public key according to the trustworthiness of the entities that vouch for it; this is the idea behind PGP web of trust [Abdul-Rahman, 1997]. It also provides trust influence at the network layer allowing for routes to be selected based on trust. Trust Establishment incorporates the following functions: specification of evidence, generation, distribution, discovery and evaluation of trust evidence. The scope of the work focuses upon trust evaluation rather than the collecting of trust evidence from the network and the semantics of such trust evidence. These issues are still important, and need to be addressed in a complete system.

Trust Representation

The DITD model represents trust on a weighted trust graph *G(V,E)* by a trust opinion. The trust opinion is a numeric trust variable which is a function of the available confidence and trustworthiness evidence.

$$trust_i(t_{evidence}, c_{evidence}) = t_i \in [0,5] \tag{1}$$

A high trust opinion means that the node is a good node, or that the node provides highly accurate location information, or that the certificate issued by the node is highly trusted. Trust is further influenced by network operation confidence. This includes the duration of a node's participation in the network, or the lack of negative evidence against the node.

The trust function, *trust*, computes the available evidence ($t_{evidence}$ and $c_{evidence}$) into a semantic numeric representation of trust. Trust is represented at each node or vertex of the trust graph. The work focuses upon the evaluation of routes and the assignment of trust to individual nodes is assumed to be taken care of by a network monitoring system.

A trust variable, t_i, will be assigned and stored at each node or vertex of the trust graph. Each node entering the network with a valid self-certificate is provided with a default trust value t_d. The DITD model can be extended to include access control and allow for a trusted outside member to assign trust values to nodes entering the network. This would allow for a more secure system with limited or specified users and still maintain the self organized nature of the network.

Trust is assigned to the established routes including both one-hop neighbouring routes and multi-hop routes. In an on-demand routing environment, nodes maintain a routing table storing the routes to each known node. A trust variable t_{AB} will be assigned to each of these routes representing the aggregated trust from the source node *A* to node *B*. The duration of time for which these routes are maintained securely will influence the weight of trust assigned to the edges of the trust graph. The trust of nodes (t_i) and trust of routes (t_{AB}) will change as the network progresses and new trust evidence is made available. The representation of trust is illustrated on weighted trust graph *G(V,E)* in Figure 7.

Certificate based trust provides the user with binary trust, i.e. when two nodes share certificates they trust each other; otherwise they don't trust each other. DITD represents trust with a range from 0 to 5. Where 0 represents a malicious node or a node not worthy of trust and 5 represent a full confidence in the certificate and trustworthiness of the node. This gives the trust graph system some flexibility.

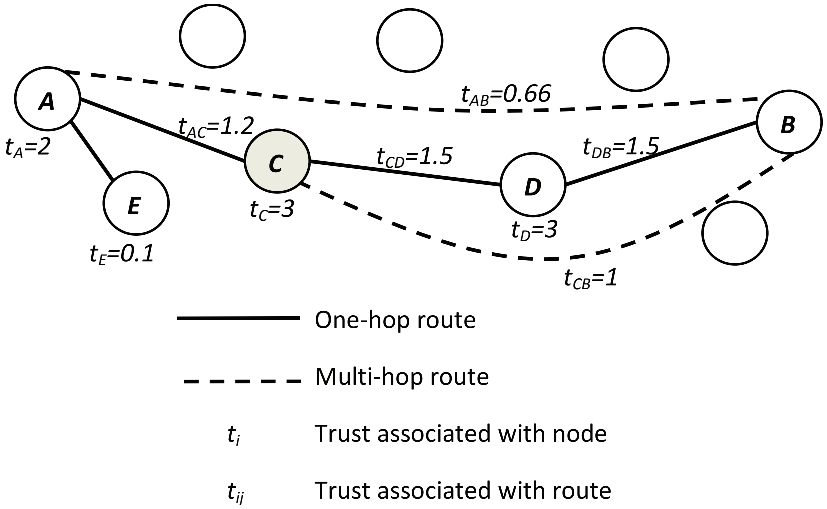


Fig. 7. Weighted Direct Trust Graph

When inconsistent data is shared then a trust accusation may be made against offenders reducing the trust of the node and the routes in which it participates. A proposal is made to use the route maintenance mechanism implemented by the on-demand routing protocol, to help establish the confidence and trust variables. The purpose of the route maintenance is to maintain the routes and to share neighbourhood information. This allows for provided trust evidence to be shared in a localized manner. This maintenance protocol allows nodes to "monitor" their neighbours and when inconsistent data is shared then a trust accusation may be made against the offender.

The DITD model inherits aspects of the semiring mathematical trust representation following semiring properties which are used for aggregating trust opinions along and across paths. The distance semiring operators $\oplus$ and $\otimes$ are applied to optimize trust accumulation. The $\otimes$ operator is used to add trust values along a trusted certificate chain. The $\otimes$ operator allows for a final trust value to be calculated representing a chain of nodes with different trust values. Trust values will be aggregated along a path like parallel resistors would be summed i.e. $\frac{1}{R_T} = \frac{1}{R_1} + \frac{1}{R_2} + ... + \frac{1}{R_n}$. The trust will decrease along the path and the final trust of the path can be no larger than the lowest trust value. This aligns with the description of a trust chain which states that a chain is as strong as its weakest link. The distant semiring approach is based on Eiser proposal [Hu et al, 2002]. Figure 5 illustrates how trust is aggregated along a path and stored representing the trust of a specific route between node *A* and *B*.

In summary, methods are proposed to allow for a trust semantic but this is not the focus of the work. The assumption is made that trust evidence is available and that each weighted vertex has been assigned a trust value.

Trust Evaluation

The conduct based proposal compliments the certificate exchange and verification mechanism forming a hybrid security model which embraces certificate trust establishment as well as conduct based trust. Ideas from the modified proactive generic-single-source-shortest-distance algorithm [Theodorakopoulos & Baras, 2006] [Mohri, 2002] are inherited and we propose to apply this semiring mathematical formulae to the reactive on-demand trust path discovery phase of the routing protocol. The generic-single-source-shortest-distance algorithm calculates the shortest path from a source node to all nodes in the network, working in a proactive manner. The DTID proposal is a reactive path specific model. DITD will have a hand in the selection of the multi-hop routes therefore its operation will lie in the network layer. The DITD model modifies and optimizes the shortest path algorithm on the network layer. The following modifications are made to the trust path discovery phase as to compliment the certificate based model with a conduct based model evaluating trust along a path.

1. Trust is aggregated along the *RREQ* path

The distance semiring mathematic operator ⊗ [Mohri, 2002] is used to allow for trust to be calculated for a route from source node *S* through intermediate nodes *1* to *n* toward destination node *D*. Trust for this path is a function of the participating nodes.

$$\begin{aligned} trust_{SD}(t_S, t_1, t_2, t_3 \ldots, t_n, t_D) &= t_S \otimes t_1 \otimes t_2 \otimes t_3 \otimes \ldots \otimes t_n \otimes t_D = t_{SD} \\ &= \left(\frac{1}{\frac{1}{t_S} + \frac{1}{t_1} + \frac{1}{t_2} + \frac{1}{t_3} + \ldots + \frac{1}{t_n} + \frac{1}{t_D}} \right) \end{aligned} \quad (2)$$

Trust is aggregated along the path that the *RREQ* propagates. The trust of the route is updated at every hop and the trust value is stored in the routing table of the intermediate nodes with respects to the level of trust of the reverse path to the source.

2. Trust is aggregated along the *RREP* path

Similarly to 1 the trust from the destination to the source is aggregated using the distance semiring formulae.

$$trust_{DS}(t_D, t_{n-1}, t_{n-2}, t_{n-3} \ldots, t_1, t_S) = t_D \otimes t_{n-1} \otimes t_{n-2} \otimes t_{n-3} \otimes \ldots \otimes t_1 \otimes t_S = t_{DS} \quad (3)$$

Although the total trust between the source and destination is already calculated after *RREQ*'s propagation, this step is necessary to provide appropriate trust values for the forward path recorded in the intermediate node's routing table.

The aggregation of trust is illustrated in Figure 8 where source node, *S*, sends a *RREQ* message to destination node *D*, and at each hop of the *RREQ* message the trust is calculated and stored as a trust value associated with the reverse route to *S*. The trust associations are t_{SP1} , t_{SP2} and finally t_{SD} which is the trust for the route between *S* and *D*. Figure 8 also follows the *RREP* message calculating the trust associated with the forward routes stored in the routing table. Figure 8 shows that trust is route specific and trust must be aggregated along both the *RREP* and *RREQ* paths.

3. Implicit revocation: Filter trust path discovery participation

The nodes that participate in the trust path discovery process must all have an acceptable value of trust. This therefore eliminates untrusted nodes from participating in the multi-hop routes; this also eliminates low level trust paths from being discovered. Before a *RREQ* message is processed and forwarded, the aggregated trust of the propagating route request is compared to a trust threshold t_{thresh}. If the trust is lower than the t_{thresh} then the *RREQ* message is discarded.

$$t_{RREQ} < t_{thresh} \tag{4}$$

This procedure will act as an implicit revocation mechanism for DITD. A trust chain is as weak as its weakest link, therefore if the weakest links are not considered then their corresponding weak trust chains are not considered either. This modification helps find the most trusted path and reduces unnecessary network computation and message propagation.

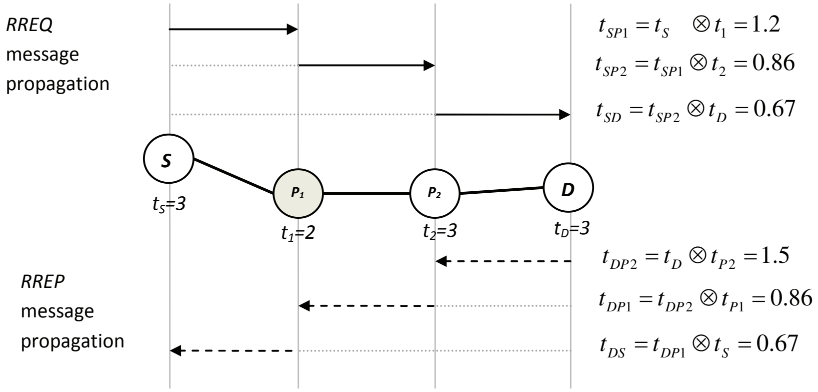

RREQ message propagation
$t_{SP1} = t_S \otimes t_1 = 1.2$
$t_{SP2} = t_{SP1} \otimes t_2 = 0.86$
$t_{SD} = t_{SP2} \otimes t_D = 0.67$
S
P₁
P₂
D
t_S=3
t_1=2
t_2=3
t_D=3
$t_{DP2} = t_D \otimes t_{P2} = 1.5$
$t_{DP1} = t_{DP2} \otimes t_{P1} = 0.86$
$t_{DS} = t_{DP1} \otimes t_S = 0.67$
RREP message propagation

Fig. 8. Illustration of trust aggregated along RREQ and RREP path

4. Filter the most trusted path

Figure 9 illustrates this step. When the *RREP* is propagated back to the source node, it is very possible that multiple routes are found therefore an intermediate node may receive more than one *RREP* message. In this case the first *RREP* is forwarded and successive *RREP*'s are only forwarded based on their sequence number or total trust value, effectively filtering the most recent and most trusted routes to the destination. The generic-single-source-shortest-distance algorithm would unicast *RREQ* messages in order of trust to their neighbours. By doing this, cyclic paths are avoided and the procedure of discovering the most trust path is maximised. The possibility of unicasting *RREQ* messages instead of broadcasting them is unfeasible for mobile ad hoc networks due to resource limitations. Instead DITD's proposal of filtering the routes by sequence number and trust will effectively realizes the relaxation process of the Dijkstra's shortest path algorithm in a reactive rather than a proactive manner.

The four additives to the trust path discovery phase allow for conduct trust evaluation to be added to the on-demand routing protocol of the ad hoc network increasing the security of trust chains created during indirect trust establishment. The conduct model is explained with an example.

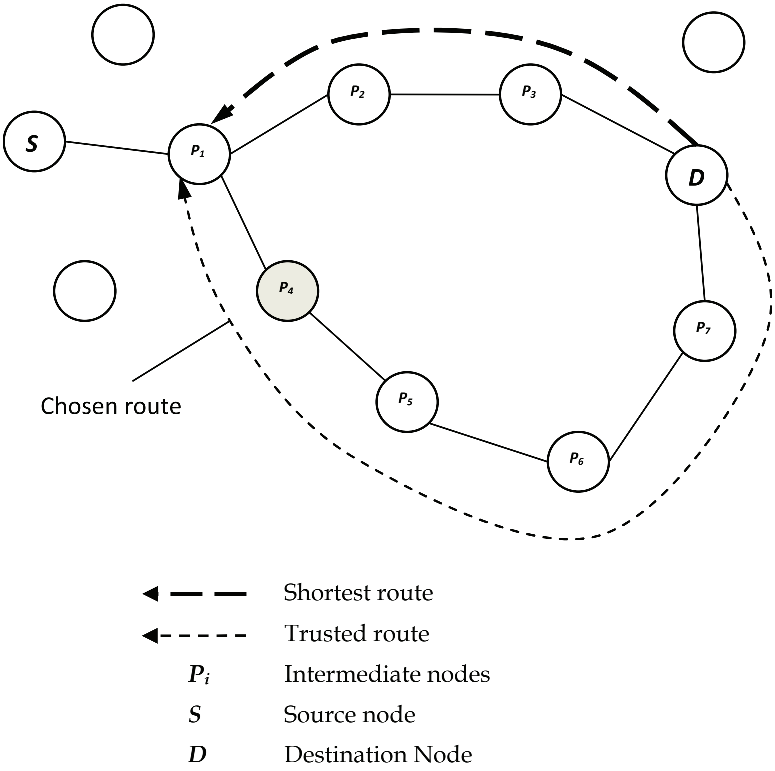


Fig. 9. Illustration of the filtering the most trusted route

To summarise this section, we can say that the section showed how the proposed hybrid trust scheme is incorporated into an ad hoc on-demand routing scheme with a low level complexity. Direct and indirect trust is established by localized one-hope certificate exchanges in a reactive manner and conduct trust is appended by aggregating trust along paths. The following section discusses the performance of the proposed scheme with use of simulations.

4. Performance and simulation study of the proposed DITD Model

There are two main approaches to evaluate routing applications for mobile ad hoc networks: simulations and real test beds [Kiess & Mauve, 2007] [Ke et al, 2000]. Real test beds can provide realistic results. However, they are impractical to set up. A real test bed, for a large network of nodes would requires 50 nodes in operation which is considerably costly. It is also difficult to compare different protocols because of the difficulty in repeating test conditions, such as mobility and erratic wireless connectivity. Therefore, real test beds are logistical unfeasibly. Currently, simulations are widely used to compare proposed routing protocols. Simulation packages like ns2 [http://,2007] and GloMoSim [Zeng et al, 1998] provide an environment to design and compare proposed and existing protocols. The majority of literature on this subject use ns2 as its enhanced functionality is suitable for wireless scenarios. The ns2 network simulator was selected to perform a simulation study for the DITD model.

This section presents the effects of adding the security functionality, proposed by the DITD model, to the AODV routing protocol. This functionality includes a certificate distribution mechanism and a trust evaluation mechanism. The environment investigated is a large mobile ad hoc network which uses an on-demand routing algorithm.

We use subsection 4.1 to: describe the simulation environment, discuss the simulation scenario, and introduce the traffic and mobility models. Subsection 4.2 describes the performance metrics used to analyze the simulated routing protocols. The focus of this section is found in Subsection 4.3 where a comprehensive simulation study is presented. This is done by comparing the proposed DITD model with the AODV routing protocol. Results are presented in simple line graphs and discussed accordingly.

4.1 Simulation setup

The goal of the simulation experiments is to measure the proposed routing protocol's performance to a changing network topology and network conditions. To measure this, protocols are simulated at varied mobility conditions. A comprehensive simulation study is presented of the proposed security scheme for mobile ad hoc networks implemented on the network layer. A summary of the simulation set used in our study is given in Table-5.

a. Simulation Scenario

The network was set up with 50 wireless nodes allowing data communication to occur in a peer-to-peer manner. Nodes are mobile in a rectangular space of 1500m x 300m and the simulation is run for 900 seconds. A rectangular area is preferred to a square area as longer routes can be expected. Nodes were configured to use the 802.11b standard communicating over wireless channels with a two-ray ground radio propagation model with a bandwidth of 2Mbps and a nominal transmission range of 250m.

Simulation Scenario	
Physical and MAC model	IEEE 802.11b standard
Nominal bit rate	2Mbps
Transmission Range	250m
Number of nodes	50 nodes
Simulation duration	900 seconds
Simulation area	1500m x 300m
Traffic Model	
Traffic type	CBR
Data packet size	64 byte
Traffic rate	4 packets per second
Traffic started	0 – 180 seconds
Number of connections and sources	30 and 20
Mobility Model	
Model	Random Waypoint
Max speed	0.1 , 1, 5, 10, 20, 30 m/s
Pause time	0 and 250 seconds

Table 5. Simulation Setup for varied topology

b. Traffic Model

Traffic was simulated using a constant bit rate (CBR) traffic generator which models UDP traffic. TCP traffic was not used because it uses its own flow control mechanism which schedules data packets based on the network's ability to carry them. CBR traffic is more useful for a routing protocol analysis as it allows the routing protocol to manage the flow of traffic. All traffic is started within the first 180 seconds of the simulation. Simulations were performed with data packets sizes of 64, 256, 512 and 1024 bytes. At higher data packet sizes traffic congestion causes a few nodes to drop most of their received packets, this was observed from test simulation runs. A data packet size of 64 bytes was selected for the simulation analysis. The focus of the simulation study is to compare the performance of routing protocols against changing topology and as no load balancing is employed in any simulated protocol, congestion is factored out by selecting a lower data packet size. The traffic analysis model is consistent with routing protocol analysis in [Broch, 1998].

For topology analysis the traffic load is fixed with a rate of 4 packets per second. The maximum number of connections is set to 30 connections with a traffic model with 20 sources.

c. Mobility Model

A modified "random waypoint" mobility model was used to prevent mobility concerns highlighted in [Navidi, 2004]. The modified random waypoint model improves upon the standard model by selecting a speed which is between 10% and 90% of the given maximum speed. This addition provides a more balanced mobility and prevents extreme drops in speed during simulation.

Changing network topology is simulated based on network participant speed. The maximum speed was varied from 0 to 30m/s with 6 different mobility patterns (0.1, 1, 5, 10, 20 and 30m/s) for two different pause time scenarios, 0 and 250 seconds, representing a network with continuous motion and a partially stable network.

4.2 Performance metric

The following quantitative metrics are used to analyze the performance of the routing protocols in mobile ad hoc networks.

a. Packet Delivery Ratio

The packet delivery ratio (PDR) represents the percentage of data packets that are successfully received by their intended destination. This metric is also known as throughput and is considered a measurement of the effectiveness of a routing protocol. The equation for PDR is:

$$PDR\% = \frac{\sum_1^n CBRrec}{\sum_1^n CBRsent} \times 100$$

where $\sum_1^n CBRrec$ and $\sum_1^n CBRrec$ are the number of CBR data packets received and sent respectively.

b. Routing Overhead

A routing protocol uses control packets to establish routes on which data packets are transmitted. Control packets are separate from data packets but share the same communication channel. Due to the lack of channel capacity in mobile ad hoc networks a large number of control packets can result in poor network performance. Key management would require additional control packets to achieve key management functionality this will be reflected in the simulations. The routing overhead is also known as a routing protocol's internal efficiency and will represent the number of control packets used for a given protocol.

c. Average End-to-End Delay

This is a qualitative measurement of the delay of data packets. The average end-to-end delay of a data packet is the time from which it is created at the source and when it arrives at the intended destination. The delay includes propagation and queuing delay. Delay can be caused by a high number of control packets propagating in the network or a high computational overhead for the given protocol. The average end-to-end delay is calculated as follows,

$$End\ to\ End\ Delay = \frac{\sum_1^n (CBRsendtime - CBRrecvtime)}{\sum_1^n CBRrec}$$

where *CBRsendtime* and *CBRrecvtime* represent the record times that a CBR data packet was sent and received.

4.3 DITD simulation

a. Implementation

A linux based server was set up to run the Network Simulator ns-2.31 [http://,2007]. A routing protocol was designed in C++ based on the AODV routing protocol available in the ns-2.31 package. The routing protocol DITD is programmed as a routing agent class. The routing agent handles the establishment of routes, certificate distribution and trust evaluation. Modifications are made to the AODV routing agent at the *RecvRequest, SendRequest, RecvReply*, and *SendReply* functions. These modifications allow for the distribution of separate certificate packets, triggered by the routing packets. The routing agent's packet header was modified to include a certificate control packet *CertS*. The size of the certificate included is 450 bytes which correlates with experiments in [Zapata, 2006]. The size of the certificate control packets is increased resulting in an effective delay in communication simulating the transfer of actual certificates. The authors of [Awerbuch et al, 2008] use a similar approach to simulate the effect of security processing. A certificate table is included at each node *CertTable* which is updated by certificate control packets. The certificate table is linked to the routing table and each node is responsible for managing its own certificate table.

The trust evaluation scheme assumes that monitoring trust evidence is available. Routing control packets are modified to include an associated trust variable. As each routing packet propagates through the network, the trust of the specific route is calculated and stored in the routing table of each node. Implicit trust revocation and trust path selection is performed at *RecvRequest* and *RecvReply* functions respectively.

A simulation *tcl* file is written to setup the mobile ad hoc network's desired simulation scenario, traffic and mobility model. The trace support files in ns-2.31 were modified to support the DITD routing agent allowing the inclusion of certificate control packets and trust information. As a result the output trace and *nam* files reflect the operation of the DITD routing agent. Figure 10 shows a sample output of the *nam* simulation file and Figure

11 shows a sample trace file output. AWK, an extremely versatile programming language for unix based systems, was used to write script files to analyze the trace data and provide the measured performance metrics. Finally unix based shell script files were written to allow for multiple iterations and simulation scenarios to be run simultaneously resulting in over 1000 simulation runs and 430 Gb of data analyzed and presented in simple line graphs.

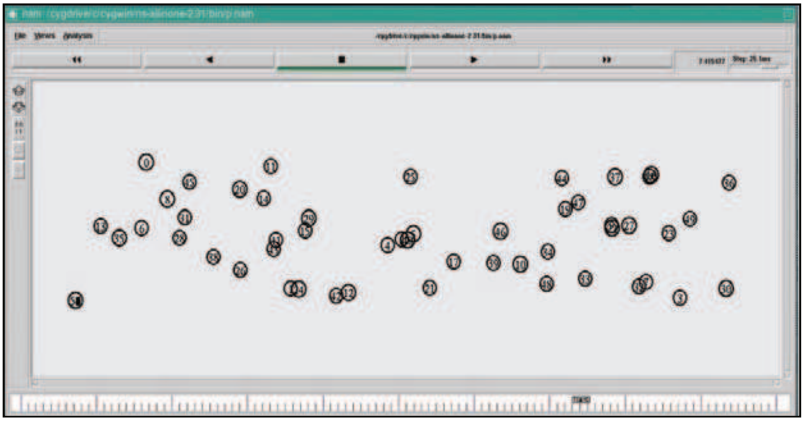


Fig. 10. Sample *nam* simulation file illustrating typical network topology

```
r 19.867 _9_ RTR  --- 0 AODV 60 [0 ffffffff 16 800] --- [22:255 -1:255 26 0] [0x2 5 1 [19 0] [17
20]] 20 (REQUEST)
r 19.867 _43_ RTR  --- 0 AODV 60 [0 ffffffff 16 800] --- [22:255 -1:255 26 0] [0x2 5 1 [19 0] [17
20]] 20 (REQUEST)
r 19.868 _25_ RTR  --- 0 AODV 508 [13a 19 27 800] --- [39:255 25:255 1 25] [0x14 [25 32] [39] 10
[0 0] 4 0 1] (CERT_R)
r 19.871 _13_ RTR  --- 125 cbr 84 [13a d 11 800] --- [17:3 19:0 30 13] [0] 1 3
f 19.8711 _13_ RTR  --- 125 cbr 84 [13a d 11 800] --- [17:3 19:0 29 39] [0] 1 3
r 19.8724 _26_ RTR  --- 0 AODV 508 [13a 1a 28 800] --- [40:255 26:255 1 26] [0x14 [26 28] [40] 10
[0 0] 4 0 1] (CERT_R)
r 19.8733 _39_ RTR  --- 125 cbr 84 [13a 27 d 800] ---[17:3 19:0 29 39] [0] 2 3
f 19.8733 _39_ RTR  --- 125 cbr 84 [13a 27 d 800] --- [17:3 19:0 28 19] [0] 2 3
r 19.877 _48_ RTR  --- 0 AODV 508 [13a 30 12 800] --- [18:255 48:255 1 48] [0x12 [48 30] [18]
10.00 [18 17] 1] (CERT_S)
```

Fig. 11. Sample trace file output for DITD simulation run

b. DITD Performance Results

The DITD model is compared with the AODV routing protocol. Further comparisons are presented against a conventional approach to key distribution. The simulation scenario used is described in Section 2.4.2 which is used throughout the simulation study. The traffic model simulates a moderate traffic load at a rate of 4 packets per second. The effects of changing topology are investigated by varying the node speed for a continuously moving network and a partially stable network. The simulation results were averaged over 10 speeds per scenario, resulting in a total of 360 iterations for the speed analysis.

Packet delivery

The packet delivery results for the AODV and DITD routing protocols are presented in Figure 12 and Figure 13. Figure 12 represents a simulation environment with a pause time of 0 seconds. This represents a network of nodes that are continually moving, while Figure 13 represents a partially stable network. The observation is made that as the speed increases both protocols throughput decreases. At high speeds the network topology changes rapidly causing breakages in routing links. The reduction in packet delivery at high speeds is because both protocols will drop data packets as a result of increased routing breakages. The curves for the AODV and DITD packet delivery ratio have similar shapes. This is expected because the DITD model is based on the AODV model. In Figure 12 the DITD model shows a 0–10% reduction gap in packet delivery when compared to the AODV model. The gap increases uniformly as the speed increases leveling at 10% for speeds of 20 m/s and higher. Similarly for the more stable network, presented in Figure 13, there is a reduction in packet delivery ratio of 0-5% when compared to the AODV model. The stable network in Figure 13 shows better performance at higher speeds because the number of route link breakages is reduced as a result of a larger pause time. A large pause time represents a network that will move at a given speed then pause in a fixed location for a set amount of time. During this time routing link breakages are not expected until movement commences again. The reduction in packet delivery ratio of DITD, when compared to AODV, can be attributed to the additional certificate packets being distributed and handled by the routing agent. The packet queue for the routing protocol has a limited capacity and when it is overloaded, packets are dropped. This will cause a resultant drop in throughput. The DITD model optimizes its throughput by processing the routing and certificate control packets independently of each other.

A certificate distribution scheme would expect a severe reduction in performance due to an excessive number of packets being transmitted in the network or the additional size of the control packet. A conventional certificate distribution scheme, suggested as a possible solution in [Buchegger & Boudec, 2002], simply includes the source certificate in the request packets *RREQ* and the destinations certificate in the reply packets *RREP*. This method was implemented as a separate routing agent *AODVcert* in ns2. A similar method is suggested in [Papadimitratos & Hass, 2002]. Implementation includes increasing the packet size of the

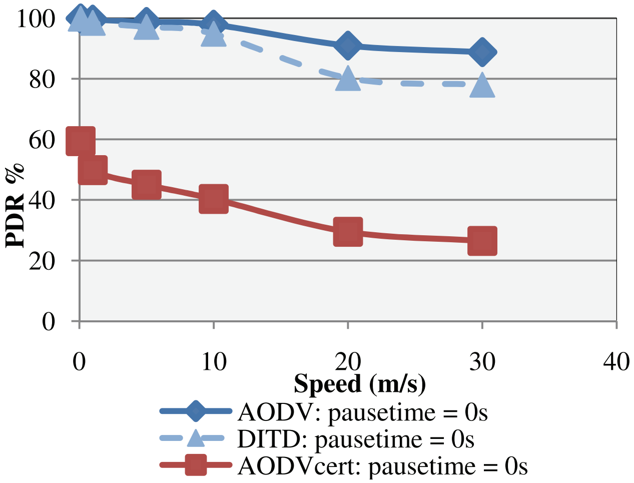


Fig. 12. Packet Delivery Ratio for highly mobile network (0 second pause time)

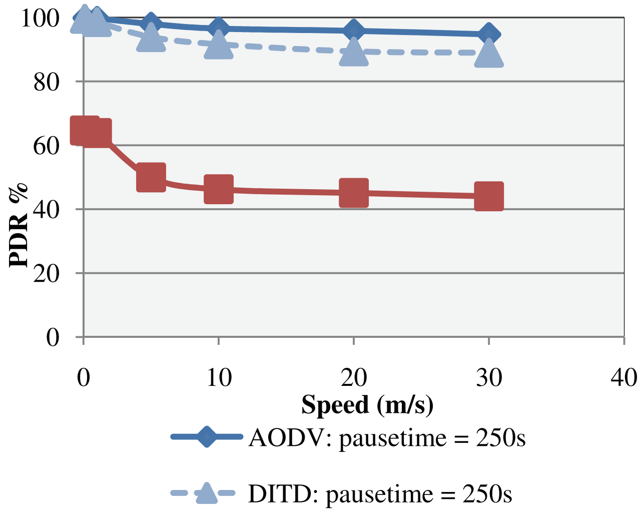


Fig. 13. Packet Delivery Ratio for partially stable network (250 second pause time)

routing control packets to include a 450 byte certificate. This effectively increased the regular 56 byte AODV route control packets to 506 bytes. Such an approach would result in the simplest method of certificate distribution but transmitting 450 bytes more data per control packet would severely reduce the network performance.

The *AODVcert* routing agent was simulated under the same simulation conditions as AODV and DITD, and the packet delivery ratio is presented in Figure 30 and Figure 31. It can be observed that the packet delivery ratio is severely less than both the AODV and DITD model. For a pause time of 0 seconds, there is an average gap of 55% between *AODVcert* and AODV and an average gap of 49% between *AODVcert* and DITD. Similar results are observed for the stable network in Figure 31. This simulation shows that DITD optimizes the distribution of certificates by sending them as separate certificate control packets independent of the route control packets. The certificate control packets are processed independently of the routing packets, allowing concurrent processing in a fully distributive system. The operation of DITD allows for certificate distribution with minimal effect upon the routing procedure.

During this time routing link breakages are not expected until movement commences again. The reduction in packet delivery ratio of DITD, when compared to AODV, can be attributed to the additional certificate packets being distributed and handled by the routing agent. The packet queue for the routing protocol has a limited capacity and when it is overloaded packets are dropped. This will cause resultant drop in throughput. The DITD model optimizes its throughput by processing the routing and certificate control packets independent of each other.

A certificate distribution scheme would expect a severe reduction in performance due to an excessive number of packets being transmitted in the network or the additional size of control packet. A conventional certificate distribution scheme, suggested as a possible solution in [Zapata, 2002], simply includes the source's certificate in the request packets *RREQ* and includes the destination's certificate in the reply packets *RREP*. This method was implemented as a separate routing agent *AODVcert* in ns2. A similar method is suggested in [Sanzgiri et al, 2002]. Implementation includes increasing the packet size of the routing control packets to include a 450 byte certificate. This effectively increased the regular 56 byte AODV route control packets to 506 bytes. Such an approach would result in the simplest method of certificate distribution but the result of transmitting 450 bytes more data per control packet would severely reduce the network performance. The *AODVcert* routing agent was simulated under the same simulation conditions as AODV and DITD and the packet delivery ratio is presented in Figure 12 and Figure 13. It can be observed that the packet delivery ratio is severely less than

Figure 14 shows that the DITD model has a 10% reduction in throughput for high speed mobile ad hoc networks. A high speed network is described by a maximum node speed of 20 and 30 m/s. This simulates mobile units travelling at a maximum speed of 70–100km/h which is typical of mobile military vehicles. Mobility aids the distribution of certificates as nodes come in close contact with each other and are able to establish direct trust relations reducing end-to-end certificate distribution. These benefits are similar to Capkun's solution which relies upon mobility to establish trust in a localized manner [Capkun et al, 2006]. Capkun's solution is aided by mobility but is also dependent upon mobility for trust relations to be established. Because of this dependency, a period of weakened security is expected as nodes exchange certificates. DITD does not only distribute certificates in a localized manner but Figure 30 shows that the DITD model has a 0 - 3% reduction in throughput for low speed mobile ad hoc networks where nodes move at a maximum speed of 0–10 m/s. This type of networks is typical of infantry units or a *nam* the ground scenario. DITD allows for mobility to aid the distribution of certificates but not relying upon mobility for throughput success. This allows DITD to operate successfully in slow moving and stationary type networks. The packet delivery ratio results show that DITD provides certificate distribution at a low performance cost for high speed networks and for low speed networks.

Control Packet Overhead

The control packet overhead presents a comparison between the AODV and DITD models. The overhead is presented in terms of the number control packets. The AODV model will have only routing control packets while the DITD model will have both routing and certificate packets. The results are presented in Figure 32 and Figure 15 for a highly mobile network with pause time of 0 seconds and a partially stable network with pause time of 250 seconds. The DITD model aims to distribute certificates while routes are discovered and a resultant packet overhead is expect. AODV and DITD are similar in shape and it is observed that the number of control packets increases as the speed increases. As the speed increases the topology of the network changes more rapidly causing routing link breakages and forcing nodes requesting communication to re-establish routes by send new route request messages. For a partially stable network presented in Figure 15 the effects of speed are reduced. This confirms that a larger pause time provides a more stable network. Figure 14 and Figure 15 show a consistent control packet overhead for the DITD model. It is observed that the gradient of DITD's packet overhead decreases as speed increases. This is because mobility aids certificate distribution and as the speed increases less certificate control packets are required. For example in Figure 14 at the low speed of 1 m/s there is a 132% increase in the number packets when compared to the AODV protocol. This overhead decreases for higher speeds showing a comparative 38% and 33% packet overhead for speeds of 20 m/s and 30 m/s respectively. This confirms that mobility aids certificate distribution.

A standard AODV request message is 48 bytes and a reply message is 44 bytes. The DITD model uses request message of 60 bytes and reply messages of 56 bytes. Therefore, DITD increases the routing control packet size by 12 bytes. DITD's routing control packets contain trust associated variables and flags to trigger back-tracked certificate distribution. The DITD certificate control packets are 508 bytes in size as they included a 450 byte certificate. It is noted that making the routing and certificate control packets separate and independent from each other has a greater impact on reducing the per byte packet overhead. This independency allows for concurrent processing of packets which is optimal in a fully distributive ad hoc network.

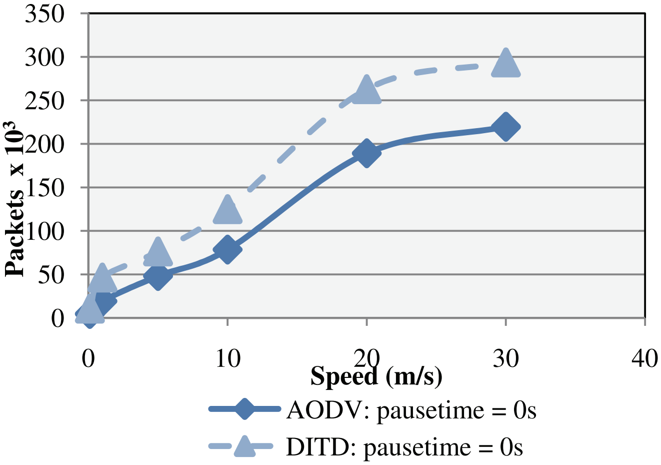


Fig. 14. Control packet overhead for highly mobile network (0 second pause time)

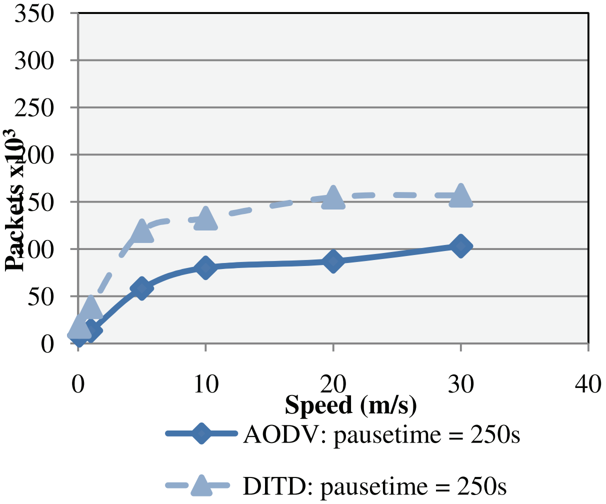


Fig.15. Control packet overhead for partially stable network (250 second pause time)

End-to-End Delay

The average end-to-end delay results are presented in Figure 16 and Figure 17. It is observed that the DITD model delivers packets with more delay than AODV. The additional delay is attributed to the transmission delay, the packet queuing delay, and the processing delay of additional certificate control packets. The processing delay includes verification. A conventional certificate distribution scheme that follows the route discovery process would require that certificates be verified before the routing packets are forwarded. DITD performs verifications independent of the routing procedure. The request route is established following the route request message *RREQ* to the destination and DITD performs verifications independently without hindering the propagation of the *RREQ* message.

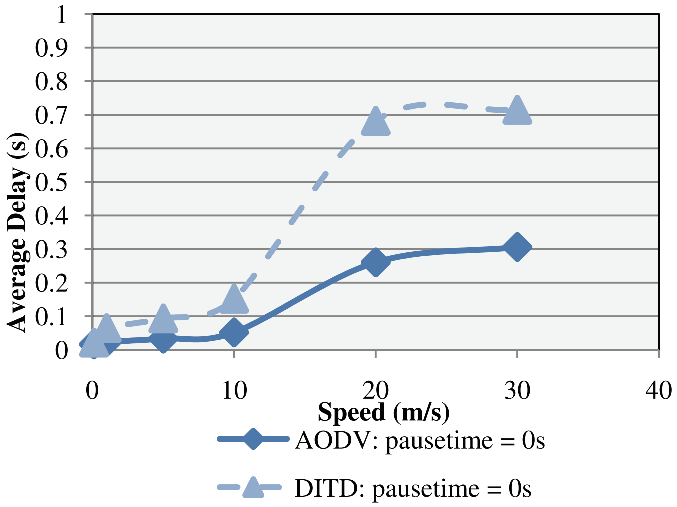


Fig. 16. Average end-to-end delay for highly mobile network (0 second pause time)

DITD uses back-track verification to minimize the number of verifications performed on the reply route which follows the reply message *RREP* toward the source. Hass and Pearlman [Haas & Pearlman, 2001] propose a solution which performs all verifications on the reply route. This method minimizes the nuns performed in a networks lifetime but results in delayed establishment of routes. If ECC (elliptic curve cryptography) type keys are used the verification process could take up to 16 ms per verification [Zapata, 2006] such a delay is unrealistic for multi hop routes requiring verification. DITD's approach attempts to minimize the delay incurred.

c. Trust Evaluation Results

In order to test the performance of the security evaluation scheme, a black hole attack was simulated to show that DITD's security evaluation scheme excludes malicious nodes from trust and route establishment protecting the network from black hole type attacks. A black hole adversary model was designed on the ns-2.31 link layer (LL) which lies below the routing layer. Modifications were made to the link layer agent *ll.cc* to simulate a black hole attack. Each packet sent by the routing layer is checked at the link layer, the adversary model silently drops all data packets while still allowing routing packets to be passed. This creates the affect of a black hole attack. A second black hole adversary model was implemented which includes a rushing type attack. The rushing attack was implemented by allowing adversary nodes to forward routing packets immediately, removing the small jitter delay that AODV implements. AODV uses this small delay to reduce the number of collisions and ensure the shortest path is selected. The rushing attack gives an adversary node a time advantage over normal nodes resulting in the adversary node becoming part of considerably more routes.

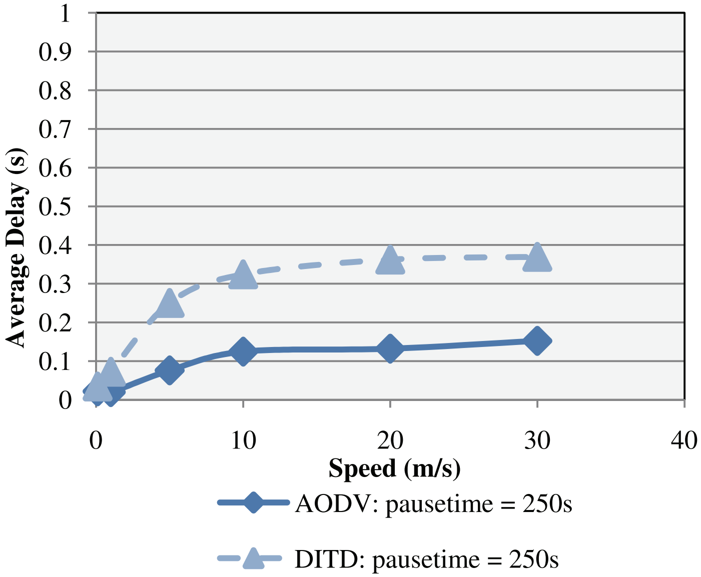


Fig. 17. Average end-to-end delay for partially stable network (250 second pause time)

The same simulation scenario and traffic model was used to analyse the black hole attack. The mobility was fixed with a pause time of 0 seconds and three speeds were investigated (0.1m/s, 5m/s and 20m/s). A 50 node network was simulated with 6 different attack scenarios. The attack scenarios were created by varying the number of black hole adversary nodes added by 0 to10. Figure 18 shows the *nam* simulation file for a simulation scenario with 10 adversary nodes. Each scenario was averaged over 10 seeds resulting in 720 iterations for the security evaluation scheme analysis. The black hole attack aims to drop data packets and reduce the networks throughput. The effects of a black hole and rushing attack are analysed using the packet delivery ratio performance metric.

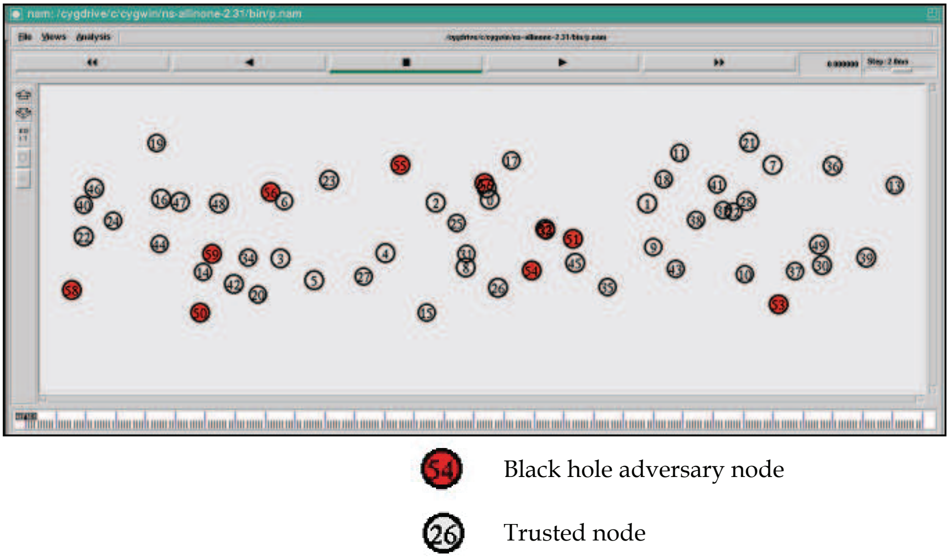

nam: /cygdrive/c/cygwin/ns-allinone-2.31/bin/p.nam
File
Views
Analysis
54
Black hole adversary node
26
Trusted node

Fig. 18. Sample *nam* simulation of black hole network simulation

Packet delivery

A black hole type problem is implemented to simulate the success of DITD's security evaluation scheme. The scenario assumes weighted nodes carry a security metric which identifies fault detection or data transmission errors carried out by a monitoring system at each node. An example of such a system is found in [Buchegger & Boudec, 2002]. The weighted nodes are used to establish a weighted trust graph where each edge or route carries a trust calculated by DITD's security evaluation scheme. The effects of the black hole attack upon AODV and DITD are compared in Figure 37 and Figure 38. It is observed that as the number of adversary nodes increases the packet delivery ratio for the AODV model decreases. The AODV model is vulnerable to black hole attacks and in the presence of 10 adversary nodes the packet delivery ratio is below 65%. The reduction in throughput is expected as more data packets will be dropped by the presence of many adversary nodes. DITD avoids the adversary nodes by implicitly excluding these nodes during route establishment. The success of the protocol at low speeds is presented in Figure 19 and it is observed that even in presence of 10 adversary nodes the packet delivery ratio is not less than 90%. Figure 38 presents the success of the DITD model at a higher mobility of 20m/s. The DITD model prevents the severe effects of black hole attacks showing better results when 4 and greater than 4 adversary nodes are present. There is approximately a 10% decrease in packet delivery ratio when compared to the low mobility scenario in Figure 19. This reduction in packet delivery ratio is attributed to the increase in link breakages apparent at higher speeds and the overhead incurred from the certificate exchange protocol. The results of DITD in Figure 20 correlate to the packet delivery ratio at 20m/s in Figure 12.

A rushing attack was included for the simulations presented in Figure 21 and Figure 22. An adversary node equipped with a rushing type attack will participate in more routes maximising the effect of its attack. Figure 21 and Figure 22 show that when adversary nodes employ a rushing attack the effects of the black hole attack are maximised. The packet delivery ratio of the AODV protocol is dropped to 40% when 10 adversary nodes are present. This is considerably less when compared to the 60-65% packet delivery ratio that AODV experiences under the same conditions with a standalone black hole attack. The results of DITD under rushing attacks are unnoticeable when compared to DITD with no rushing attacks. For low speeds, DITD provides a throughput rate of above 90% even in the presence of 10 adversary nodes.

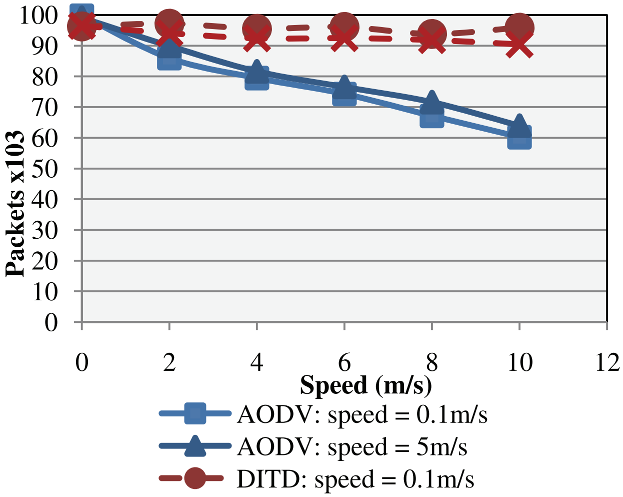


Figure 19: Packet Delivery Ratio for slow moving network under black hole attack

DITD provides a security scheme that excludes malicious nodes from participating in trusted routes, therefore preventing black hole attacks and a number of other attacks targeting the network layer. The inclusion of this trust evaluation scheme allows the distribution of certificates to operate in the most trusted routing environment.

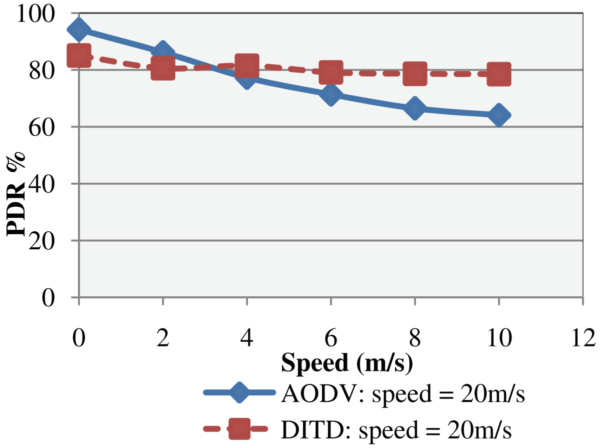


Fig. 20. Packet Delivery Ratio for fast moving network under black hole attack

4.4 Design verification

The DITD model, in relation to the design requirements stated in Section-2.3, will now be discussed. These requirements are based on the environment and functionality. The design requirements are briefly revisited throughout the discussion that follows.

a. Environment

The DITD model is required to operate on the network layer in an on-demand, fully distributive, self-organized manner. Implementation was performed on the network layer, which avoided multi-layer design problems. The simulation environment is set-up with no TTP member. This is similar to the way in which a certificate authority and network nodes are responsible for their own routing and trust establishment. The successful operation of DITD in the given environment is proven through simulation results, as presented in Section-6.

DITD is self-organized in nature. However, it is noted that DITD assumes the nodes are able to create their own keying material prior to joining the network. Self-certificates provide a strong binding between a user's key and a unique identity. The generation of keying material without the presence of a TTP is a complex problem. Solutions exist based on identity-based key generation [Shamir, 1984] [Weimerkirch & Westhoff, 2003]. The author suggests that further research in this area is carried out.

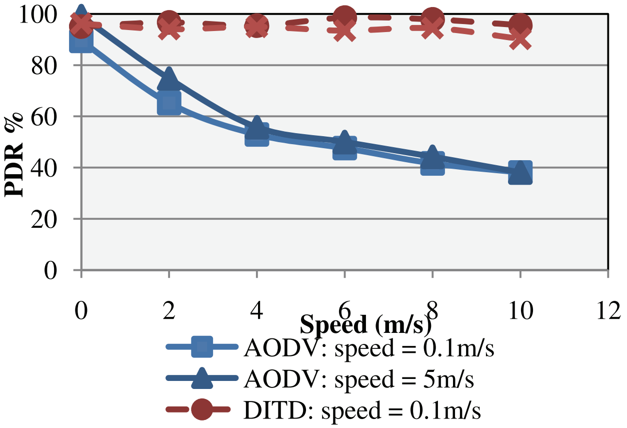


Fig. 21. Packet Delivery Ratio for slow moving network under black hole rush attack

b. Functionality

Certificate distribution is a requirement of the DITD model. DITD provides the distribution of keying material in the form of self-certificates. Local certificate exchanges are made between one-hop neighbors, which create direct trust relations. These direct trust relations are chained together to share certificates across multi-hop channels.

The DITD model assumes the existence of a weighted conduct value at each node. This allows the initial direct trust relations to have meaning. If this information is not available, direct trust relationship need to be established over a location-limited channel to ensure security, similar to infrared. Proximity based solutions are used in [Capkun et al, 2006] [Scannell et al, 2009]. DITD's simulation model assumes the availability of conduct information. Certificates are observed in the trace table as they are successfully transmitted to their desired destinations.

A second design requirement is that DITD must minimize the network overhead. The DITD model distributes certificates which use separate unicast certificate control packets. The certificates are triggered by the routing control packets. In comparison to AODV, DITD has an approximate 38% increase in control packets for highly mobile, high speed networks. The routing control packet size is increased by 12 bytes to include trust information and certificate control packets are 508 bytes in size. These packets result in a serve control packet overhead. The effects upon performance are reduced by: independency; concurrent processing; and back-track verification. Despite the significant control packet overhead, DITD merely reduces the packet delivery ratio by a 0-10% gap when compared to AODV. This reduction is notable if compared to a convention certificate distribution method, which increases the routing control packets by 450 bytes and results in over 50% reduction in packet delivery ratio. The performance of DITD is improved with more stable networks which have a higher pause time.

Simulations show that as the speed of nodes increase, the network performance decrease, as a result of a rapidly changing topology and increased link breakages. Simulations also show that mobility aids certificate distribution. However, DITD is not reliant on mobility and can still successfully operate in low speed and stationary type networks. This allows DITD to meet the requirement to provide secure communication at the start of the network lifetime. Solutions in [Capkun et al, 2006] [Tanabe & Aida, 2007] depend on mobility to establish trust and expect an initial time delay before trust is established. DITD provides secure communication in a reactive manner without a significant time delay. DITD is not limited by mobility, as it shows high throughput rates for low speed and stationary network environments.

DITD is required to be robust in spite of changing topologies. The simulations presented in Section- 6 were performed under varied pause times and speeds. This helped the investigation of the performance of DITD under varying topology environments. The simulation results show that DITD is robust in the presence of changing mobility, which will inherently have frequent routing failures. As mentioned above, DITD only reduces the throughput by a 0-10% gap across for changing topologies. It was observed that the DITD

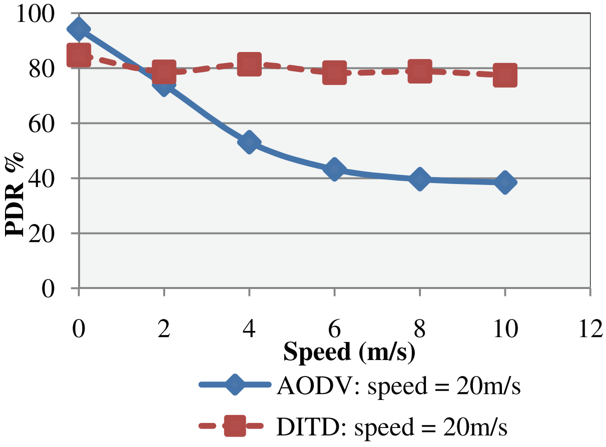


Fig. 22. Packet Delivery Ratio for fast moving network under black hole rush attack

model has an approximate 0.7 second end-to-end delay (0.4 seconds greater than AODV) for high speed, highly mobile networks. This indicates that DITD is not feasible to use for audio application, in highly mobile network environments. DITD's average end-to-end delay is reduced to 0.35 seconds (0.2 more than AODV) in a more stable network environment, which is within acceptable limits for audio application.

The last functional requirement was the inclusion of trust evaluation scheme. The trust evaluation scheme allows for the most trusted route to be selected and for malicious nodes to be excluded from route participation. The success of the scheme is present in its prevention against black hole attacks. Simulations show that a black hole attack of 10 adversary nodes causes a 35-40% reduction in packet delivery for the AODV routing protocol. DITD avoids black hole and rushing attacks by excluding malicious nodes. In low speed networks DITD achieves a 90-95% throughput rate in the presence of 10 adversary nodes.

5. Contribution and future work

5.1 Summary of contribution

Mobile ad hoc networks allow for a new set of applications that benefit from the dynamic, autonomous, and spontaneous mobile nature, inherent to these networks. However, the very qualities that make these networks so attractive also provide designers with new security challenges.

The focus of this work is upon trust establishment in mobile ad hoc network. This work contributes to the body of work in the following ways:

- Background knowledge on mobile ad hoc networks is presented. Their application in the military and commercial arena is investigated. A review of security attacks is present. Such attacks include: black hole attacks; wormhole attacks; eavesdropping attacks; byzantine attacks; resource consumption attacks; and routing table poisoning. The author identifies that mobile ad hoc networks are most vulnerable to network layer attacks and focus is placed on trust establishment on the network layer.
- Providing a comprehensive survey on the existing key management solutions for mobile ad hoc networks. The solutions are intended for different types of ad hoc networks and therefore their comparison is difficult. The solutions that are investigated are:
    - Off-line Trusted Third Party Models
    - Partially Distributed Certificate Authority
    - Fully Distributed Certificate Authority
    - Cluster based Model
    - Proximity-based Identification
    - Self Issued Certificate Chaining

  A discussion of the functionality and characteristics of each approach is presented. The self-issued certificate model is identified as providing the lowest level of pre-configuration and off-line trusted third party (TTP) involvement.
- A secure ad hoc routing survey. This work is vital to understanding trust establishment on the network layer. The following solutions are presented:
    - SEAD: Secure Efficient Ad Hoc Distance Vector Routing Protocol
    - Ariadne: A secure on-demand routing protocol for ad hoc networks

- ARAN: Authenticated Routing for Ad Hoc Networks
- SAODV: Secure Ad hoc On-demand Distance Vector (SAODV)
- SLSP: Secure Link-state routing
- ODSBR: On-Demand Secure Routing Byzantine Resilient Routing Protocol
- CONFIDANT: Reputation based solution

A comparative summary is presented focusing upon the security analysis and operational requirements of each solution. The Ariadne, ARAN, SAODV, OSRP and CONFIDANT are designed for on-demand ad hoc routing. All the protocols investigated, except the CONFIDANT protocol, assumption pre-existing key relationships or the presence of a key management system to perform the tasks of key distribution and maintenance. The CONFIDANT protocol avoids key management by establishing trust based solely on conduct. This part of the dissertation identifies an open research field in area of key management on the routing layer of mobile ad hoc networks.

- Presenting a novel security solution for mobile ad hoc networks. The solution is called Direct Indirect Trust Distribution (DITD) and is designed for an on-demand, fully distributive, self-organized, mobile ad hoc network. The scheme provides key distribution in the form of separate unicast certificate exchanges. The certificate exchange packets are independent from the routing control packets allow route establishment to operate concurrently but independently from trust establishment. A trust evaluation scheme is proposed that allows conduct based trust to influence to selection of routes and implicitly exclude malicious attacking nodes. This scheme allows the keying information to be distributed in a more secure manner.
- A comprehensive simulation study compares the performance of DITD and AODV, the protocol on which DITD is based. Simulation results show that under changing topologies DITD provides successful certificate distribution and trust evaluation with a minimal throughput reduction of 0-10%. Simulations show that DITD does not rely on mobility to distribute certificates and still performs in low speed communication networks. A black hole and rushing attack adversary model is designed on the link layer. Simulations show that DITD is successful in excluding malicious nodes from participating in route and trust establishment. The work simulation results and the discussions show that the proposed model can be implemented with low complexity and provides the functionality of key distribution and security evaluation with trivial effects on the network performance.

5.2 Future work

Future development will be made to enhance the DITD protocol, to further minimise the performance overhead. Future work includes the implementation of a load balancing agent to compliment and optimize the efficiency of DITD's key management.

The proposed model is not a standalone security solution. Future work includes the integration of the DITD scheme with a secure ad hoc routing protocol to realize a complete security system.

The key management tasks are key distribution, key generation, key maintenance and key revocation [Menezes et al, 1996b]. The DITD model addresses key distribution assuming that keys are generated by participating nodes. The generation of a secure certificate binding between a node and its public key is difficult without the presence of a trusted third party.

Furthermore, the effects adversary nodes with multiple identities performing Sybil attacks is a problem that is difficult to solve.

Trust evaluation schemes require that trust evidence be made available. Trust establishment is made up of the following services: gathering, generation, discovery and evaluation of trust evidence. This dissertation focuses upon the trust evaluation. Future work includes the gathering and interpreting of trust evidence by using local network monitors.

Mobile ad hoc cluster based networks has found increasing application in the military sector. Efficient and secure cluster based key management is a open research area to be investigated in the future.

6. References

[Abdul-Rahman, 1997] A. Abdul-Rahman, "The PGP trust model," *EDI-Forum: The Journal of Electronic Commerce,* vol. 10, pp. 27-31, 1997.

[Aram et al, 2003] K. Aram, K. Jonathan, and A. A. William, "Toward Secure Key Distribution in Truly Ad-Hoc Networks," in *Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT'03 Workshops)*: IEEE Computer Society, 2003.

[Awerbuch et al, 2002] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens, "An on-demand secure routing protocol resilient to byzantine failures," in *Proceedings of the 1st ACM workshop on Wireless security* Atlanta, GA, USA: ACM, 2002.

[Awerbuch et al, 2008 B. Awerbuch, R. Curtmola, D. Holmer, C. Nita-Rotaru, and H. Rubens, "ODSBR: An on-demand secure Byzantine resilient routing protocol for wireless ad hoc networks," *ACM Trans. Inf. Syst. Secur.,* vol. 10, pp. 1-35, 2008.

[Basagni et al, 2001] S. Basagni, K. Herrin, D. Bruschi, and E. Rosti, "Secure pebblenets," in *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking \&amp; computing* Long Beach, CA, USA: ACM, 2001.

[Broch, 1998] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking* Dallas, Texas, United States: ACM, 1998.

[Bruce, 2003] S. Bruce, *Beyond Fear: Thinking Sensibly about Security in an Uncertain World*: Springer-Verlag New York, Inc., 2003.

[Buchegger & Boudec, 2002] S. Buchegger and J.-Y. L. Boudec, "Performance analysis of the CONFIDANT protocol," in *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking \&amp; computing* Lausanne, Switzerland: ACM, 2002.

[Capkun et al., 2003] S. Capkun, L. Butty, and J.-P. Hubaux, "Self-Organized Public-Key Management for Mobile Ad Hoc Networks," *IEEE Transactions on Mobile Computing,* vol. 2, pp. 52-64, 2003.

[Capkun et al, 2006] S. Capkun, L. Buttyan, and J.-P. Hubaux, "Mobility Helps Peer-to-Peer Security," *IEEE Transactions on Mobile Computing,* vol. 5, pp. 43-51, 2006.

[Chor et al, 1985] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch, "Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract)," *proc. 26th IEEE Annual Symposium on Foundations of Computer Science,* October, 21-23 1985.

[Davis, 2004] C. R. Davis, "A localized trust management scheme for ad hoc networks. ," *In: 3rd International Conference on Networking (ICN'04)*, pp. 671–675, 2004.

[Desmendt & Jajodia, 1997] Y. Desmedt and S. Jajodia, "Redistributing Secret Shares to New Access Structures and Its Applications," Department of Information and Software Engineering, School of Information Technology and Engineering, George Mason University, Technical ReportJuly 1997.

[Douceur, 2002] J. R. Douceur, "The Sybil Attack," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*: Springer-Verlag, 2002.

[Eschenauer & Gligor, 2002] L. Eschenauer and V. D. Gligor, "A Key-Management Scheme for Distributed Sensor Networks," *proc. 9th ACM Conf. on Computer and Communication Security (ACM CCS'02)*, November, 17-21 2002.

[Frankel et al, 1997] Y. Frankel, P. Gemmell, D. MacKenzie, and M. Yung, "Optimal resilience proactive public key cryptosystems," *proc. 38th Annual Symposium on Foundations of Computer Science (FOCS '97)*, October, 19-22 1997.

[Grandison, 2003] T. Grandison, "Trust Management for Internet Applications," Imperial College London, 2003.

[Haas & Pearlman, 2001] Haas Z.J. and M. R. Pearlman, "The performance of query control schemes for the zone routing protocol," *IEEE/ACM Trans. Netw.*, vol. 9, pp. 427-438, 2001.

[Hu et al, 2002] Hu Y.C., D. B. Johnson, and A. Perrig, "SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks," in *Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications*: IEEE Computer Society, 2002.

[Hu et al, 2003b] Hu Y.C., A. Perrig, and D. B. Johnson, "Packet leashes: a defense against wormhole attacks in wireless networks," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*. vol. 3, 2003, pp. 1976-1986 vol.3.

[Hu et al, 2005] Hu Y.C., A. Perrig, and D. B. Johnson, "Ariadne: a secure on-demand routing protocol for ad hoc networks," *Wirel. Netw.*, vol. 11, pp. 21-38, 2005.

[http://2007] "The Network Simulator," *ver 2.31, Available at http://isi.edu/nsnam/ns/,* 2007.

[Johnson et al, 2001] Johnson D.B., D. A. Maltz, and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks," in *In Ad Hoc Networking, edited by Charles E. Perkins, Chapter 5*, 2001, pp. 139-172.

[Ke et al, 2000] Ke Q., I. David, D. Maltz, and D. B. Johnson, "Emulation of Multi-Hop Wireless Ad Hoc Networks," in *in The 7th International Workshop on Mobile Multimedia Communications (MoMuC*, 2000.

[Kiess&Mauve, 2007] Kiess W. and M. Mauve, "A survey on real-world implementations of mobile ad-hoc networks," *Ad Hoc Netw.*, vol. 5, pp. 324-339, 2007.

[Kscischang et al, 2001] Kschischang F.R., B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory,* vol. 47, pp. 498-519, 2001.

[Menezes et al, 1996a] Menezes A., P. van Oorschot, and S. Vanstone, *Handbook in Applied Cryptography*: CRC Press, 1996.

[Menezes et al, 1996b] Menezes A.J., S. A. Vanstone, and P. C. V. Oorschot, *Handbook of Applied Cryptography*: CRC Press, Inc., 1996.

[Mohri, 2002] Mohri M., "Semiring frameworks and algorithms for shortest-distance problems," *J. Autom. Lang. Comb.*, vol. 7, pp. 321-350, 2002

[Navidi, 2004] Navidi W., "Stationary Distributions for the Random Waypoint Mobility Model," *IEEE Transactions on Mobile Computing,* vol. 3, pp. 99-108, 2004.

[Papadimitratos & Hass, 2002] Papadimitratos P. and Z. J. Haas, "Secure Routing for Mobile Ad Hoc Networks," in *proc. SCS Communication Network and Distributed System Modeling and Simulation Conf. (CNDS'02)*, 2002.

[Papadimitratos & Hass, 2003] Papadimitratos P. and Z. J. Haas, "Secure Link State Routing for Mobile Ad Hoc Networks," in *Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT'03 Workshops)*: IEEE Computer Society, 2003.

[Perkins & Bhagwat, 1994] Perkins C.E. and P. Bhagwat, "Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers," *SIGCOMM Comput. Commun. Rev.*, vol. 24, pp. 234-244, 1994.

[Perkins et al, 2003] Perkins C., E. Belding-Royer, and S. Das, *Ad hoc On-Demand Distance Vector (AODV) Routing*: RFC Editor, 2003.

[Perrig et al, 2001] Perrig A., R. Canetti, D. Song, and D. Tygar, "Efficient and Secure Source Authentication for Multicast," Network and Distributed System Security Symposium (NDSS'01), 2001.

[Publications FIP, 2008] F. I. P. S. Publications, "Secure Hash Standard (SHS)," National Institute of Standards and TechnologyOctober 2008.

[Rivest, 1992] Rivest R., *The MD5 Message-Digest Algorithm*: RFC Editor, 1992

[Sanzgiri et al, 2002] Sanzgiri K., B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer, "A Secure Routing Protocol for Ad Hoc Networks," in *Proceedings of the 10th IEEE International Conference on Network Protocols*: IEEE Computer Society, 2002.

[Scannell et al, 2009] Scannell A., A. Varshavsky, A. LaMarca, and E. D. Lara, "Proximity-based authentication of mobile devices," *Int. J. Secur. Netw.*, vol. 4, pp. 4-16, 2009.

[Shamir, 1984] Shamir A., "Identity-Based Cryptosystems and Signature Schemes," in *proc. Advances in Cryptology: Crypto'84*, 1984.

[Stalling, 2003] Stallings W., *Cryptography and Network Security: Principles and Practices*: Prentice Hall, 2003.

[Stephan Eichler, 2006] Stephan Eichler C.R., "Challenges of Secure Routing in MANETs: A Simulative Approach using AODV-SEC," in *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, 2006.

[Tanabe & Aida, 2007] Tanabe M. and M. Aida, "Secure communication method in mobile wireless networks," in *Proceedings of the 1st international conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications* Innsbruck, Austria: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007.

[Theodorakopoulos & Baras, 2006] Theodorakopoulos G. and J. S. Baras, "On Trust Models and Trust Evaluation Metrics for Ad-Hoc Networks," *IEEE Journal on Selected Areas in Communications,* vol. 24, pp. 318-328, 2006 2006.

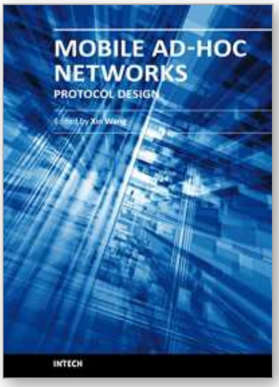
[Tseng et al, 2003] Tseng C.Y., P. Balasubramanyam, C. Ko, R. Limprasittiporn, J. Rowe, and K. Levitt, "A specification-based intrusion detection system for AODV," in *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks* Fairfax, Virginia: ACM, 2003.

[Weimerkirch & Westhoff, 2003] Weimerskirch A. and D. Westhoff, "Identity Certified Authentication for Ad-hoc Networks," in *proc. 1st ACM workshop on Security of ad hoc and sensor networks*, 2003.

[Zapata, 2002] Zapata M.G., "Secure ad hoc on-demand distance vector routing," *SIGMOBILE Mob. Comput. Commun. Rev.,* vol. 6, pp. 106-107, 2002.

[Zapata, 2006] Zapata M.G., "Key management and delayed verification for ad hoc networks," *J. High Speed Netw.,* vol. 15, pp. 93-109, 2006.

[Zeng et al, 1998] Zeng X., R. Bagrodia, and M. Gerla, "GloMoSim: a Library for Parallel Simulation for Large-scale Wireless Networks," in *proc. 12th Workshop on Parallel and Distributed Simulations (PADS '98)*, 1998.



Mobile Ad-Hoc Networks: Protocol Design
Edited by Prof. Xin Wang

ISBN 978-953-307-402-3
Hard cover, 656 pages
Publisher InTech
Published online 30, January, 2011
Published in print edition January, 2011

Being infrastructure-less and without central administration control, wireless ad-hoc networking is playing a more and more important role in extending the coverage of traditional wireless infrastructure (cellular networks, wireless LAN, etc). This book includes state-of-the-art techniques and solutions for wireless ad-hoc networks. It focuses on the following topics in ad-hoc networks: quality-of-service and video communication, routing protocol and cross-layer design. A few interesting problems about security and delay-tolerant networks are also discussed. This book is targeted to provide network engineers and researchers with design guidelines for large scale wireless ad hoc networks.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821