Trust Establishment in Ad Hoc Networks by Certificate Distribution and Postponed Verification

Richard GORDON
School of Electrical, Electronic and Computer Engineering, University of KwaZulu Natal
Durban, KwaZulu Natal, South Africa

and

Dawoud DAWOUD
School of Electrical, Electronic and Computer Engineering, University of KwaZulu Natal
Durban, KwaZulu Natal, South Africa

ABSTRACT

Trust establishment in wireless ad hoc networks is a challenge because of its unique characteristics. These include the lack of a central authority and the autonomous, dynamic nature of these networks which result in poor connectivity and routing failure. Security can be provided by a certificate based model but key management is a difficulty in wireless ad hoc networks. A key management scheme is proposed which realizes certificate distribution and verification. The key management scheme occurs in an on-demand, fully distributive, wireless ad hoc network environment, establishing trust on the routing layer exclusively. Trust and route establishment are achieved simultaneously with reduced dependency between the security and routing mechanisms. Distribution and verification of keying material places delays upon the delivery of secure communication routes. Simulations show the overhead of the proposed scheme and that it has negligible impact on network performance while providing trust establishment for the network.

Keywords: Trust Establishment, Mobile Ad Hoc Networks, Key Distribution, Key Management.

1. INTRODUCTION

Wireless ad hoc networks are complex networks which have little or no existing network infrastructure. This lack of fixed network architecture creates complex security problems. The term secure 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."[1]. There are two main approaches for trust establishment: certificate based trust models [2, 3] and conduct based trust models [4, 5]. Certificate based trust models use vital keying material to provide trust. 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 and indirect trust is created by certificate chaining. Key management is central to certificate based trust establishment [2, 3, 6]. One primary task of key management is the distribution of the keying material for example self-certificates. In a fixed network an on-line trusted authority is present to perform key management tasks. This is not possible in an ad hoc network which lacks a central trusted authority or fixed network infrastructure. Literature shows that there are two main approaches to solve this problem. A partial distributive scheme [3, 7] which distributes the functionality of the trusted network authority amongst a limited number of nodes. The second approach is a fully distributive scheme [2, 8] which distributing the security responsibilities across the entire network. In a fully distributive scheme each node is considered to be the centre of its own world, and is responsible for its own secure communication [2].

Achieving key management in mobile ad hoc networks is a challenge due to the lack of a central authority and the autonomous, dynamic nature of these networks which result in poor connectivity and routing failure. Many secure routing protocols for mobile ad hoc networks are published, e.g. *SAODV* [9], *SEAD* [10], *ARIADNE* [11], and *endairA* [12]. Most of these assume pre-existing and pre-sharing keying relationships. Key management proposed in [7-9] operates on the routing layer to achieve key distribution. The required certificates are appended to all routing request in an effort to distribute keying material during the route establishment phase. This approach is not ideal for an on-demand ad hoc network environment because it results in flooding the network with route request during its route discovery phase.

Secure protocols exist that provide key management tasks such as key distribution [6] but these schemes lack to consider the delay incurred from the key management task of verification, assuming it to be negligible. Existing models have such delayed bootstrapping security phases that security is only delivered after an initial time of setup. This creates a window period of weaken security or a window period of restricted communication [4, 6].

The aim of our paper is to design a key management scheme that can be used to distribute and verify certificates in a wireless on-demand ad hoc network, with negligible affects upon routing performance. The proposed scheme establishes trust by distribute and verify certificates for all the nodes in a network with the following constraints:

- The key management scheme is to operate in an on-demand environment, exclusively on the routing layer.
- The key management scheme is to distribute and verify certificates between local and remote nodes providing direct and indirect trust relationships respectively.
- Each node in an indirect trust chain must verify its neighbour, the originator and destination node's certificates, before the trust chain is secure.

- The key management scheme aims to minimize the security overheads which affect the network routing performance. These overheads include certificate verification and distribution delays.
- The key management scheme should avoid altering the routing mechanism, and strive for independence between routing and trust establishment. Routing packet size is not to be extended to incorporate security information.
- The certificate scheme is to be designed in a fully distributive manner with no existence of an on-line trust authority or prior trust relationships.
- Security should be available as a node enters a network with a seemingly timeless bootstrapping phase for security.
- The key management scheme should be robust to poor connectivity and routing failure due to shifting mobility, error-prone wireless channels and traffic congestion which are natural characteristics of wireless ad hoc networks.

The proposed scheme is called Direct, Indirect Trust Distribution (DITD) and it follows the procedure outlined in Section 2. The paper is structured in the following manner: In Section 2 the Direct, Indirect Distribution scheme is proposed, describing the distribution and post verification mechanisms. Section 3 includes the implementation, simulation and evaluation of DITD's performance. Section 4 provides closing conclusions.

2. PROPOSED SCHEME

System Model

To fulfil the constraints given in Section I, 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 [2], or by an off-line authority issuing keying material before a node enters the network for example in [6]. Each node is assumed to have a public and private key pair, a 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 lifetime, unlike [4, 6] which is flawed by its initial setup phase with weak security.

Proposed DITD Model

The proposed Direct, Indirect Trust Distribution Model (DITD) aims to distribute and verify self certificates to create direct and indirect trust relationships between nodes. DITD is a certificate based trust model which works with existing mobile ad hoc routing schemes. It is not specific for a single routing protocol but its principals can be applied to any routing scheme. In the following we introduce the proposed scheme in AODV environment.

AODV [13] routing procedure has three stages: sending the request message; receiving the request message; and sending the reply message. In the first stage, the originator node *A* requests communication with destination node *B* by broadcasting a routing request *RREQ* into the network. This request is forwarded by intermediate nodes and propagated through the network to *B*. When the *RREQ* message is received by an intermediate node *P*, it may have been sent by *A* or forwarded by a neighbouring node *NP*. Upon receiving the *RREQ* message stage two begins. At stage two a reverse route to *A* is then set up and *P* 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* and propagates until the destination is found. When the destination or a fresh route to the destination is found, stage three commences. A reply message *RREP* is propagated along the reverse route until it reaches the originator node *A* establishing the communication route.

When a node receives a routing control packet, and before that packet is processed, DITD sends certificate requests using separate unicast messages. The self certificate distribution is added at stage two and stage three; the receiving of the route request and the sending of the reply message stages.

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. The proposed scheme is subdivided into three parts: Direct trust establishment, indirect trust establishment and the post verification optimization.

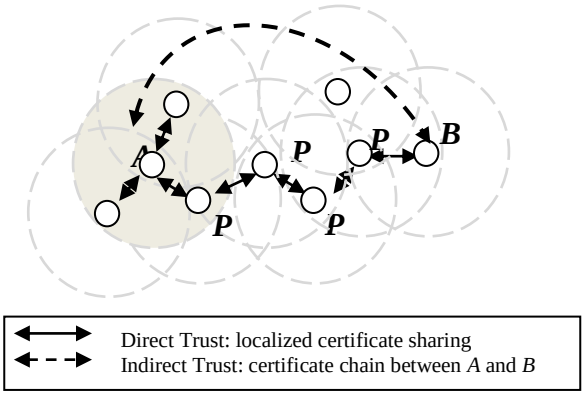

Figure 1: Direct and Indirect trust establishment

1) Direct Trust

At stage two, direct trust relationships are made by allowing the neighbouring nodes to exchange certificates. When intermediate node *P* receives a route request *RREQ* it first checks its certificate repository for the certificate of the neighbour, *NP*, who forwarded the request. If it does not possess such a certificate, $Cert_{NP}$, a local self certificate exchange is done between node *P* and its neighbour *NP* using two unicast messages. The exchange of certificates follows the *RREQ*. This will floods the network in search of a route to the destination node. Direct trust establishment is illustrated in Figure 1. What can be expected is an increased initial packet overhead.

2) Indirect Trust

Similar to direct trust establishment, at stage two node *P* searches for the originator's certificate, $Cert_A$. If it is not found, node *P* sends a unicast certificate request for $Cert_A$ to *NP* whose address can be found at the next hop on the reverse route. This propagates $Cert_A$ to the destination *B*. For indirect certificate trust to be established originator *A* is required to possess the destination's certificate, $Cert_B$, as well. By not appending the certificate to the route requests dependency is reduced between the route establishment and trust establishment.

At stage three, sending the reply message, the indirect trust establishment is completed. Sending a reply is guided by 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 *A* is already setup with localized direct trust existing between nodes on the route; therefore a trusted certificate chain of nodes is available towards

the originator node *A*. It is required only that 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. For the second condition, if a fresh route to *B* is found, there exists a route from intermediate node *P* to destination *B* and a route from *P* to *A*. Both routes have localized direct trust existing already, so the two routes can be view as certificate chains. Two *RREP* messages are then propagated, one toward *B* with the $Cert_A$ appended and one toward *A* with the $Cert_B$ appended. Indirect trust is therefore set up by certificate chaining as illustrated in Figure 1.

3) Verification Protocol

Verification upon an indirect trust chain requires each node to verify its chain neighbour, the originator and destination node's certificate. Ideally verification will take place immediately after a certificate is received but the processing of a single verification results in a computational delay. For application specific networks that are time dependent like military automation networks a delay of even milliseconds is critical. DITD provides optimized verification by allowing routing messages to be forwarded pending verification confirmation.

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, *RREQ* message can be forwarded without waiting for verification to be processed [8] as verification can be confirmed on the reply route. Such delayed confirmation of verification is not possible for the *RREP* message and certificates must be verified before the *RREP* message can be securely forwarded and trusted routes established. 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 the use of back tracked verification. Figure 2 illustrates the verification protocol. 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 *flag* is appended identifying if the forwarder has the destination certificate $Cert_B$. Node *P* receives the *RREQ* message and updates the reverse route entry with $flag_{cert}$ indicating if the previous hop has $Cert_B$. Node *P* checks if it has $Cert_B$ in its certificate repository and assigns an appropriate value to *flag* before forwarding the *RREQ*. If node *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 $Cert_B$. 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 route discovery phase independent from route establishment.

Therefore the neighbour and the originator certificate ($Cert_A$) are verified without causing any delay upon the route discovery. The destination certificate ($Cert_B$) is verified on the *RREP* message, and this delay is elevated by a prior distribution of $Cert_B$ on the reverse route where this is possible.

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 during the initial trust establishment stage.

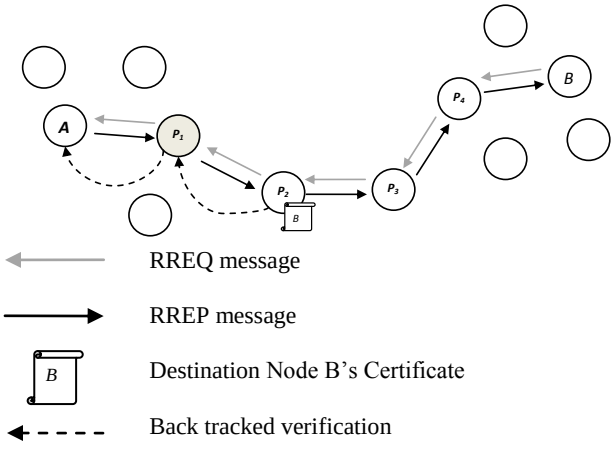

Figure 2: Illustration of verification protocol

3. DISCUSSION

Performance Evaluation

The performance of the proposed DITD protocol was analysed in a simulation study, to identify the effects of the certificate exchange and verification protocol upon the network layer and network performance. DITD is designed in C++ based on the AODV routing protocol available in the *ns-2* (release 2.31) [14] package. DITD allows certificate distribution and trust establishment in an on-demand manner. The size of the certificate included is 450 bytes which correlates with experiments in [8]. The size of the control packets is increased to allow certificate distribution resulting in an effective delay in communication simulating the transfer of actual certificates.

The DITD model is compared with the AODV routing protocol. Packet delivery, control packet overhead and end-to-end delay are the metrics which are used to measure DITD's performance.

Simulation Model

A wireless ad hoc network is simulated using the *ns-2* designed IEEE 802.11b physical layer and medium access control (MAC) protocols. The transmission range of each node was set to 250m. The network was setup with 50 nodes 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. Traffic is simulated using a constant bit rate (CBR) traffic generator that models UDP traffic. TCP traffic is not used because it uses its own flow control mechanism which schedules data packets based on the network's ability to carry them. The data packet size is set to 64 bytes and a traffic load of 4 packets per second is used. All traffic is started within the first 180 seconds of the simulation. The maximum number of connections is set to 30 connections with a traffic model with 20 sources.

The focus of the simulation study is to compare the performance of routing protocols against changing topology. A modified "random waypoint" mobility model was used to prevent mobility concerns highlighted in [15]. The modified random waypoint model improves upon the standard model by selecting a speed that 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.
The effect of changing topology is investigated by varying the node speed for a continuously moving network and a partially stable network. The simulation results were averaged over 10 seeds per scenario, resulting in a total of 360 iterations.

Simulation Results

The simulation results for the DITD protocol are presented and discussed aiming to assess the impact of the DITD protocol on the network performance.

1) Packet Delivery Ratio

The packet delivery ratio (PDR) represents the percentage of data packets that are successfully received by their intended destination. The PDR results for the AODV and DITD routing protocols are presented in Figure 3 and Figure 4. Figure 3 represents a simulation environment with a pause time of 0 seconds. This represents a network of nodes that are continually moving, while Figure 4 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 3, 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. Likewise for the more stable network presented in Figure 4, there is a reduction in packet delivery ratio of 0-5% when compared to the AODV model. The stable network in Figure 4 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 [16], 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 [17]. 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 3 and Figure 4. 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 4. 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.
Figure 3 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 thereby 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 [6]. 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 3 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 network is typical of infantry units or a man-on- the-ground scenario. DITD allows for mobility to aid the distribution of certificates but does not rely on 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.

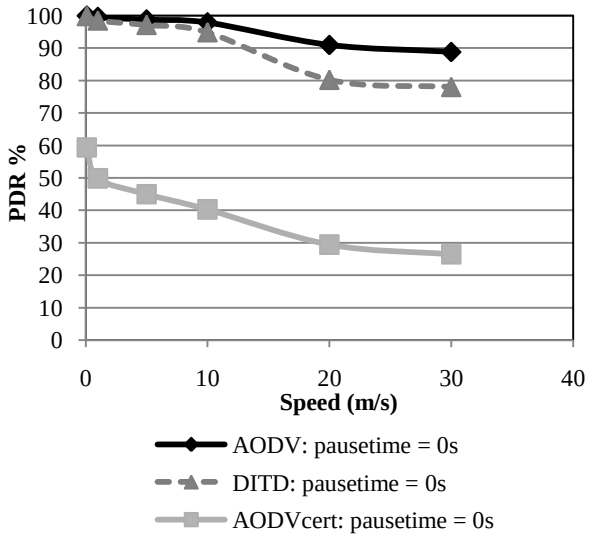

Figure 3: Packet Delivery Ratio for highly mobile network (0 second pause time)

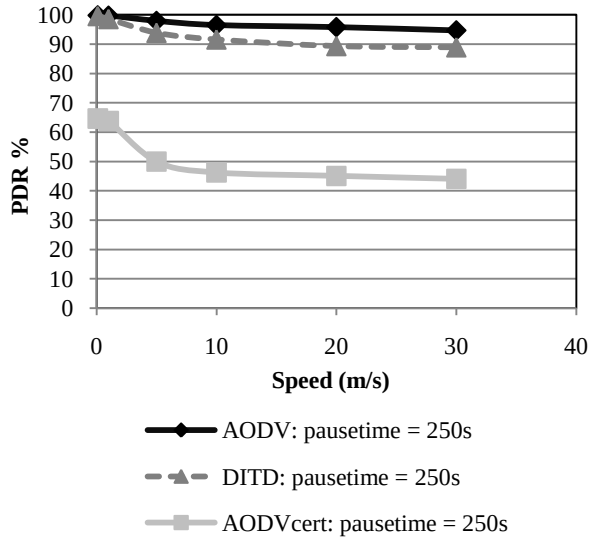

Figure 4: Packet Delivery Ratio for partially stable network (250 second pause time)

2) 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 5 and Figure 6 for a highly mobile network with a pause time of 0 seconds and a partially stable network with a pause time of 250 seconds. The DITD model aims to distribute certificates while routes are discovered and a resultant packet overhead is expected. 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 6, the effects of speed are reduced. This confirms that a larger pause time provides a more stable network. Figure 5 and Figure 6 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 5 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 messages 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 include 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 than reducing the per byte packet overhead. This independency allows for concurrent processing of packets which is optimal in a fully distributive ad hoc network.

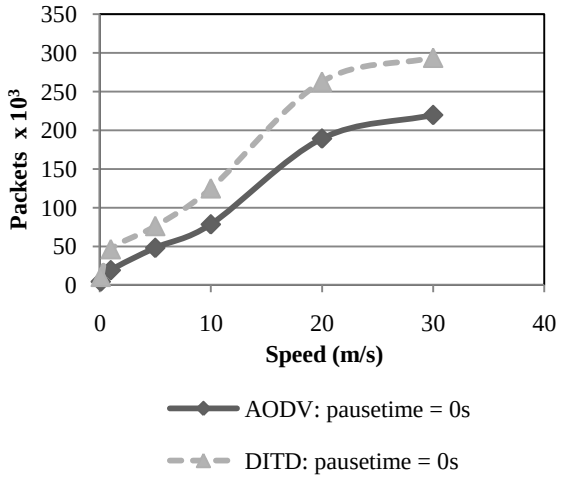

Figure 5: Control packet overhead for highly mobile network (0 second pause time)

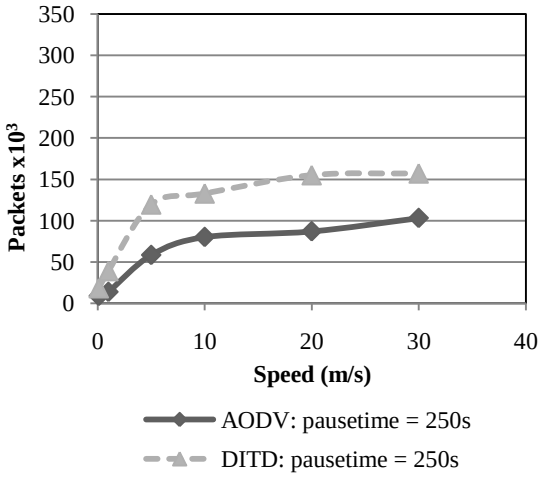

Figure 6: Control packet overhead for partially stable network (250 second pause time)

3) End-to-End Delay

Average end-to-end delay is a qualitative measurement of the delay of data packets. The average end-to-end delay of a data packet is the duration of the time from which it is created at the source until it arrives at the intended destination. The average end-to-end delay results are presented in Figure 7 and Figure 8. 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. DITD uses back-track verification to minimize the number of verifications performed on the reply route

that follows the reply message *RREP* toward the source. The authors of [8] propose a solution that performs all verifications on the reply route. This method minimizes the number of verifications 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 16ms per verification [8]. Such a delay is unrealistic for multi-hop routes requiring verification. DITD's approach attempts to minimize the delay incurred.

The average end-to-end delay of DITD follows a similar shape to the AODV model. It is observed that DITD has a lesser delay for low speeds. This is because the network topology is not rapidly changing and fewer packets are required to be transferred and processed. It is observed that the end-to-end delay and control packet overhead are closely related. This confirms the effect that additional control packets have on the network performance. For a high speed network with a pause time of 0 seconds and maximum speeds of greater than 20m/s, the average delay is a consistent 0.4 seconds more than AODV. For more stable networks with a pause time of 250 seconds, the average delay is reduced to an average of 0.2 seconds for speeds of 10m/s and higher. The additional delay of DITD is expected as the protocol performs certificate distribution and security evaluation which requires additional control packets to be transmitted and processed. Since packet queues are implemented in a first-in-first-out structure, the additional certificate control packets would result in data packets being queued for a longer time.

4. CONCLUSION

In conclusion the Direct, Indirect Trust Distribution (DITD) model provides a novel key management solution to realize certificate distribution and optimum certificate verification on the routing layer of a mobile ad hoc network. Routing packets are exploited by localized certificate exchanges providing direct trust and indirect trust by certificating chaining. Localising certificate exchange messages, remove trust dependency upon multi-hop routes which are vulnerable to collapse due to the dynamic nature of mobile ad hoc networks. DITD operates in an on-demand manner allowing secure communication from the start of the network formation. DITD's postponed and back track verification mechanism helps minimize delays caused by computationally costly verification.

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 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. It is concluded that DITD uses local certificate exchanges and delayed certificate verification as an efficient way to provide trust establishment and key distribution in a mobile ad hoc network.

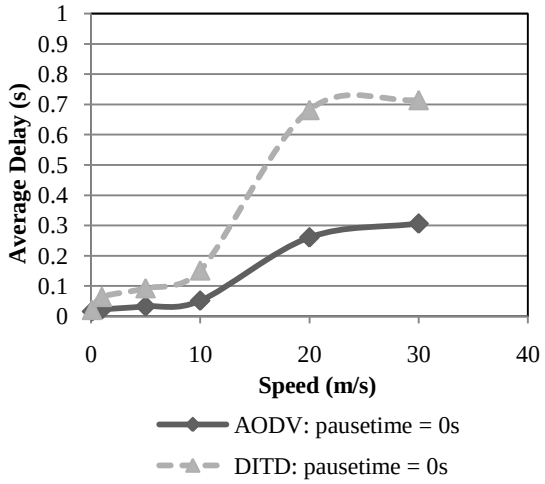

Figure 7: Average end-to-end delay for highly mobile network (0 second pause time)

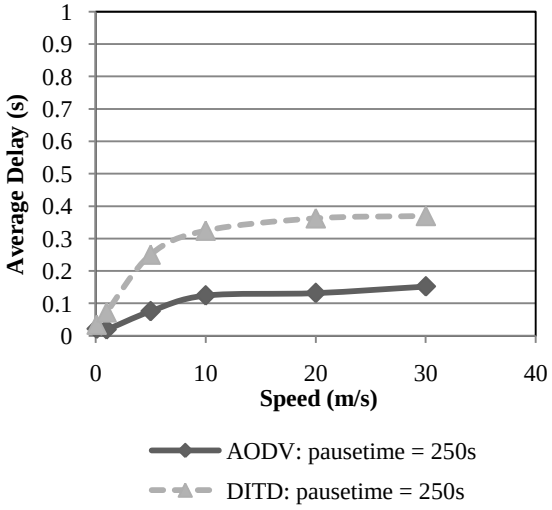

Figure 8: Average end-to-end delay for partially stable network (250 second pause time)

5. REFERENCES

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

[2] 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.

[3] L. Zhou and Z. J. Haas, "Securing Ad Hoc Networks," *IEEE Network: special issue on network security*, vol. 13, pp. 24-30, 1999.

[4] M. Tanabe 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.

[5] G. Theodorakopoulos 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.

[6] 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.

[7] 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.

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

[9] C. E. Perkins, E. Belding-Royer, and S. R. Das, "Secure Ad Hoc On-demand Distance Vector (SAODV) Routing," 2003.

[10] Y.-C. Hu, 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.

[11] Y.-C. Hu, 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.

[12] G. Acs, L. Buttyan, and I. Vajda, "Provably Secure On-Demand Source Routing in Mobile Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 5, pp. 1533-1546, 2006.

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

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

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

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

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