Secure Key Distribution for Mobile Ad Hoc Networks

Dawoud S DAWOUD
dshenouda@nur.ac.rw
National University of Rwanda
Butare, Rwanda

and

Jesudoss AUXEELIYA
jauxeeliya@nur.ac.rw
National University of Rwanda
Butare, Rwanda

ABSTRACT

The central focus of this paper is mobile ad hoc networks (MANETs) and their security. Key distribution is a major challenge in this dynamic type of network. A novel mechanism called Direct Indirect Trust Distribution (DITD) is proposed to secure key distribution for an on-demand ad hoc routing protocol. The solution includes a trust evaluation and representation mechanism. A security evaluation metric is proposed that aggregates trust along a path based on a security metric and the path distance. The proposed solution is implemented on a modified AODV routing protocol, and simulated on the ns2 Network Simulator. Simulations are conducted in order to compare the performance of the AODV and DITD protocols. The simulation results show that the DITD model provides secure key distribution in the presence of adversary black hole nodes.

Keywords: Key distribution, Key management, Mobile ad hoc networks, Security metric, Routing protocol.

1. INTRODUCTION

Mobile ad hoc networks have distinct characteristics that 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; physical vulnerability; and node mobility. Users cannot rely on an outside central authority, such as a trusted third party (TTP) or certificate authority (CA), to perform security and network tasks [1]. The responsibility of networking and security is distributed among the network participants. Users have no prior relationship with each other and do not share a common encryption key. Therefore, only after the network has been formed, do the users establish trust and networking links. The establishment of networking links is identified as being vulnerable to security attacks. Trust establishment should allow protection for the network layer and ensure that honest links are created. The network layer provides a critical service to the mobile ad hoc network and the routing protocol. In the context of trust and security, the provision of secure routes is one of the most vital elements for trust establishment.

Adversaries will launch impersonation, eavesdropping and denial of service attacks on the network. Certain attacks are specific to the network layer and difficult to prevent. We present different types of attacks which endanger the routing layer of the mobile ad hoc network.

1) Wormhole attack
In a wormhole attack a compromised node receives packets at one place in the network. The attacker tunnels the packets to another destination (i.e. an external attacker) in the network where the packets are re-sent back into the network [2]. The tunnel created by the adversary is known as a wormhole. If the routing mechanism is not protected against such an attack, mobile ad hoc routing protocols may fail to find valid routes.

2) Black hole attack
Another attack is the black hole attack. During route discovery, a malicious node may falsely advertise itself as possessing the optimal route to the requested destination. Consequently, the adversary attracts all routing messages. The attacker then creates a black hole attack by dropping all routing packets, and disrupting the routing protocol and discovery phase.

3) Byzantine attack
Byzantine attacks are when a malicious node or a group of malicious nodes will launch attacks on the routing protocol. The aim is to direct routing packets to follow non-optimal routes, routing loops, and selective dropping of packets [3]. Byzantine behavior is difficult to detect. A network could be operating with Byzantine failures and be unaware of the attack on its routing mechanism.

4) Routing Table Poisoning
Malicious nodes will target the routing table in an attempt to sabotage the establishment of routes. One such attack is the routing table poisoning attack where malicious nodes send counterfeit routing updates or modify existing routing updates. This results in conflicting link information, unnecessary traffic congestion or denial of service.

5) Rushing attack
Mobile ad hoc networks that use on-demand routing protocols are vulnerable to rushing attacks [4]. On-demand routing protocols, such as AODV [5] and DSDV [6], use route request messages to discover the optimal route to a destination node. The network is flooded with route request messages. These messages are forwarded until the optimal route is found between the source and destination nodes. An adversary that receives a route request performs a rush attack by hurriedly flooding the network with that route request before other nodes, receiving the same route request, can respond. When other nodes receive the legitimate routing request, it is assumed to be a duplicate of the request that is distributed by the adversary,
and the legitimate routing request is dropped. Consequently, the adversary will become part of the route that is discovered and this will result in a generally insecure route.

Key management systems are responsible for the secure distribution of keys to their intended destinations. Cryptographic techniques use these keys to provide confidentiality, authenticity and integrity. Achieving secure key management in mobile ad hoc networks is a challenge due to the unique attackers. Many secure routing protocols for mobile ad hoc networks are published, e.g. SAODV [7], SEAD [8], ARIADNE [9], and endaira [10]. Most of these assume pre-existing and pre-sharing keying relationships. Key management proposed in [7, 11-13] operates on the routing layer to achieve key distribution. We aim to secure these key distribution mechanisms by providing secure trust path evaluation.

### 2. PROPOSED SCHEME

The security of a key distribution mechanism can be complemented by implementing an intelligently secure way of distributing keys. We propose a conduct-based trust enhancing mechanism which complements our existing key distribution proposal [13] and can operate in conjunction with other non-specific on-demand key distribution mechanism. The proposal is called DITD (Direct Indirect Trust Distribution).

The provision of conduct-based trust enhances the trust decision made by nodes and thereby affects 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” [14]. 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.

#### A. 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 variable that is a function of the available confidence and trustworthiness evidence.

\[
\text{trust}(t_{\text{evidence}}, c_{\text{evidence}}) = t_i \in [0,5]
\]

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 that 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 \((t_{\text{evidence}}, c_{\text{evidence}})\) 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 \( t_i \)

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 neighboring 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 a weighted trust graph \( G(V,E) \) in Figure 1.

#### Figure 1: Weighted Direct Trust Graph

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

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 route maintenance is to maintain the routes and to share neighborhood 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, a trust accusation may be made against the offender.

The DITD model inherits aspects of the semiring mathematical trust representation following semiring properties that are used for aggregating trust opinions along and across paths. The distance semiring operators \( \odot \) and \( \oplus \) are applied to optimize trust accumulation. The \( \odot \) operator is used to add trust values along a trusted certificate chain. The \( \oplus \) operator allows for a
The trust value decreases 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 that states that a chain is as strong as its weakest link. Figure 1 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.

B. Trust Evaluation

The conduct-based proposal complements the certificate exchange and verification mechanism forming a hybrid security model that embraces certificate trust establishment as well as conduct-based trust. Ideas from the modified proactive generic-single-source-shortest-distance algorithm [15, 16] are inherited, and this semiring mathematical formula can be applied 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. DTID will have a hand in the selection of the multi-hop routes. Hence, its operation will lie in the network layer. The DTID model modifies and optimizes the shortest path algorithm on the network layer. The following modifications are made to the trust path discovery phase in order 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 distance semiring mathematic operator \( \otimes \) [16] is used to allow for trust to be calculated for a route from source node \( S \) through intermediate nodes \( I \) to \( n \) toward destination node \( D \). Trust for this path is a function of the participating nodes.

\[
\text{trust}_{SD}(t_S, t_{I_1} \otimes t_{I_2} \otimes t_{I_3} \otimes \ldots \otimes t_{I_n} \otimes t_D) = \frac{1}{t_S + t_{I_1} + t_{I_2} + t_{I_3} + \ldots + t_{I_n} + t_D}
\]

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 respect to the level of trust of the reverse path to the source.

2) Trust is aggregated along the RREP path: As was the modus operandi with the RREQ path, the trust from the destination to the source is aggregated using the distance semiring formula, as follows:

\[
\text{trust}_{DS}(t_D, t_{n-1} \otimes t_{n-2} \otimes t_{n-3} \otimes \ldots \otimes t_1 \otimes t_S) = t_D \otimes t_{n-1} \otimes t_{n-2} \otimes t_{n-3} \otimes \ldots \otimes t_1 \otimes t_S = t_{DS}
\]

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 2 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 \( t_{RREQ} \), \( t_{RREP} \) and finally \( t_{DS} \) which is the trust for the route between \( S \) and \( D \). Figure 2 also follows the RREP message calculating the trust associated with the forward routes stored in the routing table. Figure 2 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 eliminates untrusted nodes from participating in the multi-hop routes, and 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 \( t_{\text{thresh}} \). If the trust is lower than the \( t_{\text{thresh}} \) then the RREQ message is discarded.

\[
t_{\text{RREQ}} < t_{\text{thresh}}
\]

This procedure will act as an implicit revocation mechanism for DTID. 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.

4) Filter the most trusted path: When the RREP is propagated back to the source node, it is very possible that multiple routes are found. Hence, 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 realize 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.

We have discussed 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-hop 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.

3. DISCUSSION

A. 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 and number of adversary nodes. A modified “random waypoint” mobility model was used to prevent mobility concerns highlighted in [17]. 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 with 3 different mobility patterns (0.1, 5 and 30m/s) representing a stationary, slow moving and highly mobile network respectively. These speeds were simulated 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.

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 thereby protecting the network from black hole type attacks. A black hole adversary model was designed on the ns-2.31 link layer (LL) that 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 that included 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 that 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.

The same simulation scenario and traffic model was used to analyze 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 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 to 10. Figure 3 shows the ns simulation file for a simulation scenario with 10 adversary nodes. Each scenario was averaged over 10 speeds, resulting in 720 iterations for the security evaluation scheme analysis.

![Figure 3: Sample ns simulation of black hole network simulation](image)

B. Simulation Results

The black hole attack aims to drop data packets and reduce the networks throughput. The effects of a black hole and rushing attack are analyzed using the packet delivery ratio performance metric.

A black hole type problem is implemented to simulate the success of DITD’s security evaluation scheme. The scenario assumes that weighted nodes carry a security metric that 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 [18]. 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 4 and Figure 5. 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 4; and it is observed that even in presence of 10 adversary nodes, the packet delivery ratio is not less than 90%. Figure 5 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 4. 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 5 correlate with the packet delivery ratio at 20m/s in Figure 6-3.

A rushing attack was included for the simulations presented in Figure 6 and Figure 7. An adversary node equipped with a rushing type attack will participate in more routes thereby maximizing the effect of its attack. Figure 6 and Figure 7 show that when adversary nodes employ a rushing attack, the effects of the black hole attack are maximized. 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.

DITD provides a security scheme that excludes malicious nodes from participating in trusted routes, thereby 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.
Figure 7: Packet Delivery Ratio for fast moving network under black hole rush attack

5. REFERENCES


