# Key Distribution in Mobile Ad Hoc Networks based on Message Relaying*

Johann van der Merwe, Dawoud Dawoud, and Stephen McDonald

University of KwaZulu-Natal, School of Electrical, Electronic and Computer Engineering, South Africa; email: {vdmerwe, dawoudd, mcdonalds}@ukzn.ac.za

**Abstract.** Securing wireless mobile ad hoc networks (MANETs) is challenging due to the lack of centralized authority and poor connectivity. A key distribution mechanism is central to any public key management scheme. We propose a novel key distribution scheme for MANETs that exploits the routing infrastructure to effectively chain peer nodes together. Keying material propagates along these virtual chains via a message relaying mechanism. We show that the proposed approach results in a key distribution scheme with low implementation complexity, ideally suited for stationary ad hoc networks and MANETs with low to high mobility. The proposed scheme uses mobility as an aid to fuel the rate of bootstrapping the routing security, but in contrast to existing schemes does not become dependent on mobility. The key dissemination occurs completely on-demand; security associations are only established as needed by the routing protocol. We show through simulations that the scheme's communication and computational overhead has negligible impact on network performance.

**Key words:** Mobile ad hoc networks, wireless network security, key management, network level key distribution, trust establishment, data dissemination

## 1 Introduction

Protecting the network infrastructure in mobile ad hoc networks (MANETs) is an important research topic in wireless security. Key management is central to MANET security [1] [2] [3]; most secure routing schemes ( [4] [5] [6] [7] [8]) neglect the crucial task of secure key management and assume pre-existence and pre-sharing of secret and/or public/private key pairs [1]. One of the primary objectives of any key management scheme is the efficient and secure dissemination of keying material. *Key distribution* in MANETs is more difficult than in conventional wireline networks due to poor connectivity. Furthermore, using conventional methods such as an online key distribution center (KDC), results in a single point of vulnerability. Issuing all the nodes in the network with their own keying material *and* with the keying material of all other potential network

* This work was supported by ARMSCOR, the Armaments Corporation of South Africa.

participants, *prior* to network formation, makes the network nonscalable and introduces a tedious, inefficient, offline initialization phase. This approach may be impractical for a large group of MANET applications [9] [10] and does not allow for 'ad hoc' network formation.

There are two main approaches in the area of key management for MANETs. Most schemes either make use of a distributed trusted authority [1] [11] [12] or take on a fully self-organized nature [3] [9] [10].

Existing self-organized key management schemes, such as [3] [9] [10], allow nodes to generate their own keying material. Each node thus acts as its own authority domain and generates its own public key certificate or establishes symmetric keying material on a peer-to-peer basis. In [3], as an alternative to the fully self-organized setting, an *offline* trusted authority can also issue each node with its own certificate and a universal set of system parameters. Nodes exchange certificates when they come into transmission range. This *authority-based* approach allows for strong access control while eliminating any form of online trusted authority. We look more closely at key distribution in an authority-based setting and therefore do not explicitly consider the fully self-organized case.

The key management scheme in [9] [10] [7] distributes public keys by including them in the routing control packets. A similar approach is taken in [13]. With the large number of route requests sent by on-demand routing protocols, inflating the control packets (specifically route request messages) wastes valuable bandwidth, which is a limited commodity in ad hoc networks. Adding keying material in routing control packets is therefore not an ideal solution.

The key establishment mechanisms proposed in [3] break the *routing-security interdependence cycle* as defined in [13], but rely on node mobility to bring nodes within transmission range (or a "secure range") to set up bi-directional security associations. The dependence on mobility introduces a time delay in bootstrapping of the routing security. Furthermore, the key establishment mechanisms of [3] is not designed for a stationary (or low mobility) network, but are well suited for establishing keying material on the application layer in a fully self-organized setting.

*Informal Problem Statement.* In the light of the above discussion on the existing key management schemes, we identify a new problem within the area of key management; the challenge is to design a straightforward *key distribution* scheme that can issue all the nodes in *authority-based* MANETs with the minimum amount of required keying material (e.g. certificates), while satisfying the following constraints:

- The key distribution mechanism must exploit mobility as originally shown by Capkun et al. [3], but in contrast to existing solutions [3] avoid relying on node mobility in any way; the key dissemination mechanism must therefore be fully functional in a stationary or low mobility ad hoc network and perform even better in a high mobility scenario. If the scheme is dependent on node mobility the key distribution mechanism will fail in low mobility or stationary settings.

- The scheme should be *fully distributed* and therefore equally share the responsibility of setting up security associations between all nodes forming the network. This is to ensure reliable security services that place the same burden on the computational, memory and energy resources of *all* nodes [1] [2].
- The key dissemination mechanism should break the *routing-security interdependence cycle* [13], while ensuring network scalability. Pre-distributing keying material to all the nodes, such that security associations between all nodes will be *guaranteed*, trivially mitigates the routing-security interdependence cycle. This however makes the network nonscalable; the offline trusted third party needs to engage with all nodes before the network can be formed. The key distribution mechanism should thus only require each node to be issued with its own keying material prior to network formation and not with the keying material of other nodes, that is, the key distribution scheme should allow for 'ad hoc' network formation.
- The scheme should avoid introducing any noticeable delay in the set up of security associations; the routing must be secure from the start of network formation, hence leave no window of opportunity for an attacker during security bootstrapping.
- The key distribution scheme should reduce communication and computational overhead to have negligible impact on network performance under realistic traffic and mobility scenarios.
- The scheme should avoid inflating the routing protocol control packets in order not to waste bandwidth.
- The key dissemination mechanism should introduce minimal changes in the underlying secure MANET routing protocol and integrate seamlessly with existing secure routing protocols.
- Certificates must be distributed (on-demand) as needed on the network (routing) layer and be transparent to the network participants, that is, the scheme should require no user involvement. Unnecessary user involvement makes the scheme prone to attacks that exploit human error.

In this paper we contribute a new *key distribution* mechanism in support of secure routing that satisfies all the constraints given above. We will not focus on a complete key management solution, but concentrate our efforts on the described key distribution problem. The proposed scheme is designed specifically to have a low implementation complexity and to allow for easy integration into most secure MANET routing protocols. The proposed scheme, called Certificate Dissemination based on Message Relaying (CertRelay), is derived from the following straightforward procedure, illustrated in Fig. 1:

When a node (RN) receives a routing control packet it checks in its certificate database if it has the certificates of the packet originator (ON) and the previous-hop node (PN) on the forward route. If RN has both the certificates of ON and PN ($Cert_{ON}$ and $Cert_{PN}$), it can process the control packet as normal. If not, it requests both the certificates from PN. If RN does not have the certificate of PN it also sends its own certificate with the request to the previous-hop. Note that if RN is the first-hop on the route, then the previous-hop node and

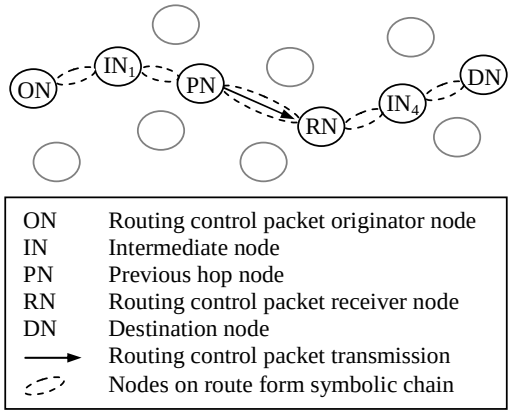


**Fig. 1.** CertRelay certificate distribution main procedure

the control packet originator node will be the same entity. The routing messages thus effectively chain nodes together and allow them to relay all keying material, as required, along the virtual chains.

The paper is organized as follows: in Sect. 2 we propose the new key distribution mechanism, called Certificate Dissemination based on Message Relaying (CertRelay). Section 3 discusses the security, performance and features of the proposed certificate distribution scheme. Some conclusions are provided in Sect. 4.

## 2 Proposed Certificate Distribution Mechanism

The discussion commences by giving an overview of CertRelay's system model followed by an abstract explanation of the proposed scheme.

### 2.1 System Model

Similar to [3], we consider a fully distributed network of wireless nodes with generic medium access control (such as IEEE 802.11) and secure on-demand routing mechanisms (such as *endairA* [8]). Nodes can be stationary or move with low to high mobility speeds ($0m/s - 20m/s$). We assume that there are no pre-existing infrastructure and no form of *online* trusted authority to assist the key distribution mechanism. Since we are considering *authority-based* MANETs as defined in [3], there exists an *offline* authority to bootstrap the system; before users join the network they have to acquire a certificate from the *offline* trusted authority. The trusted authority thus only issues each node with their own certificate and not with the certificates of any other nodes. This requirement is fundamental to ensuring scalability and on-demand network formation. Each node is also issued with the authentic public key of the trusted authority and a

universal set of system parameters. The certificate must contain the offline authority's identity, the node's public key and identity/network address, a unique sequence number, certificate generation date and expiry date.

We are now ready to discuss our key distribution mechanism, called Certificate Dissemination based on Message Relaying (CertRelay).

### 2.2 Proposed Key Distribution Scheme

While reading the explanation of the proposed key relaying mechanism below, it will be useful to keep in mind an existing MANET routing protocol. Being familiar with the operation of, for example, *endairA* [8], one of the latest *provably* secure MANET routing protocols, will help to visualize how the proposed protocol will integrate into an existing routing protocol. We point out that any other secure routing protocol will also suffice. For example, SAODV [14] [7] can also help to place the functionality of CertRelay into context.

**Table 1.** Message exchange decision table for receiver node (RN)

| Case 1: ON IP address = PN IP address | | | |
|---|---|---|---|
| Case# | ON cert stored | PN cert stored | Messages exchanged with PN |
| 1a | no | no | Peer-to-Peer certificate exchange<br>$Cert_{RN} \longrightarrow ON$[a], $Cert_{ON} \longrightarrow RN$ |
| 1b | yes | yes | No action, process routing packet as normal |
| **Case 2: Originator IP address ≠ PN IP address** | | | |
| Case# | ON cert stored | PN cert stored | Messages exchanged with PN |
| 2a | no | no | Peer-to-Peer certificate exchange<br>$[Cert_{RN} \parallel CertQ \longrightarrow PN]$[b],<br>$[Cert_{PN} \parallel Cert_{ON} \longrightarrow RN]$ |
| 2b | yes | no | Peer-to-Peer certificate exchange<br>$Cert_{RN} \longrightarrow PN$, $Cert_{PN} \longrightarrow RN$ |
| 2c | no | yes | $CertQ \longrightarrow PN$, $Cert_{ON} \longrightarrow RN$ |
| 2d | yes | yes | No action, process routing packet as normal |

[a] $RN$ = Receiver node, $ON$ = Originator node, $Cert_X$ = certificate of X.
[b] $PN$ = Previous-hop node, $CertQ$ = certificate query (RN uses this message to request $Cert_{ON}$ from $PN$, $A \parallel B$ = concatenation of messages $A$ and $B$.

The proposed key distribution scheme, CertRelay, is mainly based on the straightforward procedure introduced in Sect. 1. Table 1 explains CertRelay's core procedure in more detail from the routing control packet (RCP) receiver node's perspective (see Fig. 1). Table 1 can alternatively be seen as a summary of the conditions under which the RCP receiver node (RN) will request certificates

from and relay certificates to the previous-hop node (PN) in the virtual chain. We briefly discuss Table-1:

- When any node in the network receives a RCP it first determines if the originator of the message (ON) has the same network address as the previous-hop node (PN) on the forward route, that is, RN has to determine if ON is the first-hop. Assume the addresses of ON and PN are equivalent as shown in Table 1, Case 1. RN consults its certificate repository and searches for the certificate corresponding to ON.
  - In Case 1a the search produces no result and the RN sends the ON its own certificate, $Cert_{RN}$. The ON replies with $Cert_{ON}$. After $Cert_{ON}$ is verified by RN the RCP can be processed as specified by the routing protocol.
  - If the search yields a positive result the routing message can be processed without RN requesting $Cert_{ON}$ (Case 1b).
- If the ON address and the previous-hop node (PN) address are not equal (Case 2, Table 1), the RN will search its certificate repository for $Cert_{ON}$ *and* $Cert_{PN}$.
  - In Case 2a the search yields a negative result. RN concatenates its own certificate $Cert_{RN}$ with a certificate query (CertQ) and relays (unicasts) the message to the previous-hop[1]. PN responds with a concatenation of its own certificate and the certificate of ON ($Cert_{PN} \parallel Cert_{ON}$). Node RN should verify both certificates before continuing to process the RCP as defined by the routing protocol.
  - If RN already has $Cert_{ON}$, but not $Cert_{PN}$, it initiates a peer-to-peer certificate exchange by sending its own certificate to PN (Case 2b). PN will respond with $Cert_{PN}$, which should be verified by RN before proceeding.
  - Case 2c is applicable if RN has $Cert_{PN}$, but not $Cert_{ON}$. This case will be the most probable since PN is within RN's local neighborhood (transmission range). RN sends PN a CertQ message. PN responds with $Cert_{ON}$. Again RN verifies $Cert_{ON}$ before processing the RCP.
  - The routing message can be processed as normal in Case 2d, since $Cert_{ON}$ and $Cert_{PN}$ are already stored in the node's certificate repository.

We have discussed how the proposed key distribution scheme can be integrated into most secure MANET routing protocols. In summary, any routing message that is received by a node acts as a trigger for the node to request from the previous hop, the relaying of required keying material. The conditions that

[1] RN sends its own certificate to PN, since PN may require $Cert_{RN}$ when routing control messages are sent back via the established route. In addition, since RN and PN are neighbors they will most probably require each others certificates during future route discovery procedures. We show in Sect. 3.3 that the success rate of localized peer-to-peer certificates exchanges are high, thus if RN does not have $Cert_{PN}$ then PN will also not have $Cert_{RN}$ with high probability.

warrant the requests and format of the requests are defined by the rules in Table 1. In the following section we will analyze CertRelay in terms of efficiency and security.

## 3 Discussion on the Security and Features of CertRelay

### 3.1 On the security of CertRelay

To ensure the integrity of all messages sent by CertRelay we require that messages are signed using a secure digital signature scheme (for example RSA). Ideally CertRelay should use the same signature scheme as deployed by the underlying routing protocol. A unique sequence number or random number (to guarantee the uniqueness of each message) must also be included in the messages to avoid replay attacks.

In the remainder of the section we will analyze the security of CertRelay in the authenticated-links adversarial model (AM) of Bellare, Canetti, and Krawczyk [15]. Cagalj, Capkun and Hubaux [16] also uses AM to prove the security of their scheme, which supports our use of AM. As formally proven in [15] and further explained in [16], a strong security argument in the AM model (or ideal world model) will also apply in the unauthenticated links model (UM) by correctly applying a signature-based *message transmission* (MT)-authenticator to each message sent. The security of the protocol, if provably secure in an *authenticated* network, can then be conveniently reduced to the security of the digital signature scheme in an *unauthenticated* network [15]. The goal is thus to show that CertRelay is secure in AM, which will imply equivalence in UM. Without losing credence in the security argument we will keep our treatment informal, but firmly rooted in the formal foundations of the AM adversarial model defined by [15].

Consider Case 1a in Table 1, which portrays a generic communication scenario in CertRelay. The discussion also applies with minor modifications to any of the other cases (Case 1b to 2d). Let ON be party $A$ and RN party $B$ [2]. Note that in Case 1a the originator node is the same entity as the previous-hop node (ON = PN). An AM adversary ($\mathcal{M}$) models the authentication protocol executed by party $A$ and party $B$ (from $A$'s perspective) as an oracle $\prod_{A,B}^{s}$ with session ID $s \in \mathbb{N}$ [17]. In the same way, queries sent to $B$ from $\mathcal{M}$ and the corresponding responses are modelled by oracle $\prod_{B,A}^{t}$, where session ID $t \in \mathbb{N}$. Using the notation of [16], the timely messages sent to and received from $\prod_{A,B}^{s}$ are denoted by conversation $conv_A$ and $conv_B$ for $\prod_{B,A}^{t}$. Oracles $\prod_{A,B}^{s}$ and $\prod_{B,A}^{t}$ have *matching conversations* (as defined in [17] and further explained in [16]) if message $m$ sent out by $\prod_{A,B}^{s}$ at time $\tau_i$ is received by $\prod_{B,A}^{t}$ at time $\tau_{i+1}$.

In the AM model the adversary $\mathcal{M}$ has full control, that is, $\mathcal{M}$ can activate or corrupt parties at random, but cannot forge or replay messages to impersonate uncorrupted parties and is also bound to deliver sent messages faithfully [15]. The

[2] We assume that both $A$ and $B$ can be trusted to behave as specified by CertRelay, otherwise there is not much to discuss.

CertRelay protocol commences by $\mathcal{M}$ activating $\prod^{s}_{A,B}$ at time $\tau_0$. The outgoing routing control message $R_{msg}$ of $\prod^{s}_{A,B}$ contains the identity (or network address) of $A$ [3]. The AM adversary cannot modify the network address (identity) in the AM model by definition (see [15]) and has to deliver the message to $\prod^{t}_{B,A}$, modelling an arbitrary party $B$ of $\mathcal{M}$'s choice[4]. Incoming message $R_{msg}$ activates $\prod^{t}_{B,A}$ to respond with $B$'s certificate $Cert_B$ at time $\tau_1$ (any other activation will not comply with CertRelay). $Cert_B$ (containing the identity of $B$) is appended to $A$'s identity and delivered to $\prod^{s}_{A,B}$ as required of $\mathcal{M}$. Up to this point there is not much the adversary can do to attack the protocol; according to the definition of AM, $\mathcal{M}$ can activate any of the oracles (in an appropriate manner in compliance with the CertRelay protocol), but cannot forge messages coming from the oracles that simulate uncorrupted parties ($A$ and $B$) and has to deliver the outgoing messages after activation to the oracles. In the next round the AM adversary has no option but to activate $\prod^{s}_{A,B}$ which will respond to $\prod^{t}_{B,A}$ with $Cert_A$ (containing the identity of $A$, appended with the identity of $B$) at time $\tau_2$. Since $\tau_0 < \tau_1 < \tau_2 < \tau_3$ and $conv_A$ and $conv_B$ are matching conversations, as illustrated below, both oracles will output "Accept" [5].

$$conv_A = (\tau_0, \perp, R_{msg}), (\tau_2, CertB, CertA);$$
$$conv_B = (\tau_1, R_{msg}, Cert_B), (\tau_3, CertA, \perp);$$

As described above the AM adversary cannot attack CertRelay in the AM model without breaking the rules of AM or modifying the oracles not to comply with CertRelay. Considering the communication model of [15] and the security argument above, it is clear that CertRelay is a *message driven protocol* (as defined in [15]) that forces *matching conversations* between parties that engage via CertRelay. CertRelay is therefore a secure mutual authentication protocol (with authenticated data as described by [17]) in the AM model:

As mentioned above CertRelay can be transformed from a secure AM protocol to a secure UM protocol using a signature-based MT-authenticator [15]; each unique message $m$ (containing the identity of the sender) is signed with the private key of the sender. The signatures are verified with the sender's corresponding public key. Each public key is bound to the identity of the corresponding private key holder by an offline authority to form a certificate. As assumed in the system model (see Sect. 2.1) each network participant has the authentic public key of the offline authority readily available to verify the authenticity of the received certificates. Successful verification convinces the receiver of the binding between the public key and the user's identity (network address). Since

[3] Once $\mathcal{M}$ has activated $\prod^{s}_{A,B}$ for an arbitrary party $A$, $\mathcal{M}$ cannot alter the identity of $A$ anymore without violating CertRelay or the rules of AM.

[4] Party $B$ does not necessarily know the identity of $A$ *a priori* and does not need to until $B$ receives $R_{msg}$ from $A$. If $B$ receives the $R_{msg}$, $\mathcal{M}$ cannot alter the identity of $B$ anymore without violating CertRelay or AM.

[5] To remain compatible with [16] we also use $\perp$ to denote that a party receives/sends no message in the corresponding time $\tau_i$.

the certificates are included in the exchanged messages, it is therefore clear that CertRelay is a mutual authentication protocol in the UM model with an exchange of *implicitly* authenticated data. As a final observation we note that the probability of *No-Matching*, as defined in [17], between $conv_A$ and $conv_B$ (in the UM model) is given by the probability that the adversary can break the underlying signature scheme, which should be negligible if the signature scheme is carefully chosen and securely implemented.

### 3.2 On the efficiency of CertRelay

The efficiency analysis of CertRelay on the network layer in an ideal setting, i.e. assuming guaranteed connectivity, is rather easy. Certificate exchanges all take place on a peer-to-peer basis. From Table 1 it can be seen that all the exchanges take at most two asynchronous rounds with one unicast message from each node. Each node pair only exchanges their certificates once on a need-to-know basis.

In the following section we evaluate CertRelay in a more realistic setting.

### 3.3 Performance Evaluation of CertRelay

The performance of CertRelay was evaluated in a simulation study, as commonly done in the validation of MANET protocols, where factors such as poor connectivity and route failures (due to the error-prone wireless channel, node mobility, congestion, packet collisions etc.) have an impact on performance. The ease of coding CertRelay in the ns-2 simulator (release 2.28) [18] confirmed the low implementation complexity of the proposed key distribution scheme.

**Simulation Model** In the simulation of CertRelay we used the IEEE 802.11b physical layer and medium access control (MAC) protocols included in the ns-2 simulator. The radio-model was set to a nominal bit-rate of 11Mb/s and a transmission range of 250m. The network area for all simulations was set to 2000m x 2000m. The ns-2 constant bit-rate (CBR) traffic generator was used to set up the connection patterns. For all simulations a 512byte CBR packet size was used and the traffic loading was varied between 1 CBR packet/sec and 7 CBR packets/sec. The size of certificates was also set to 512bytes. The total of 50 nodes in the network each had one CBR traffic connection with a single unique destination, with an average path length of approximately 4 hops between connected nodes. The traffic sources were started within the first 60sec of each 1000sec simulation. We note that this is unlikely to occur in practice, but it is an effective strategy to force as much certificate distribution activity as possible from the start of network formation.

The choice of an appropriate mobility model is a problem and it is unlikely that everybody will agree with any specific choice. Although mobility models for MANETs have received much attention lately [19], a widely used, realistic mobility model is not available and it is unlikely to appear due to the application specific nature of mobility patterns. To be consistent with most literature

the random waypoint model was chosen to simulate node mobility. The *mobgenss* [20] mobility scenario generator was used to produce random mobility patterns. It is pointed out that the *setdest* mobility generator included in the ns-2 distribution is flawed [20]. The initial probability distribution of *setdest* differs at a later point in time as it converges to a "steady-state distribution" [20]. All simulation results were averaged over 10 random seeds (runs).

We wanted to observe the effectiveness of CertRelay in very low (almost stationary), moderate and high node mobility settings. In the simulations the mean speed was set to 0.1m/sec, 5m/sec and 20m/sec for each traffic scenario. These mobility speeds are widely used in MANET simulations based on the random waypoint model. Since a pause time greater than zero reduces the relative node speed, the pause time was set to zero. The Ad Hoc On-demand Distance Vector (AODV) routing protocol [21] was chosen for the simulations. The implementation of CertRelay in ns-2 *closely* followed the discussions in Sect. 2 and will not be explained here in order to avoid repetition.

**Simulation Results** In this section the simulation results of CertRelay are presented. The aim is to make an assessment of CertRelay's impact on network performance. The following two metrics are observed: 1) Constant bit-rate (CBR) packet delivery ratio (PDR) as a function of mobility and load. 2) CBR packet end-to-end delay as a function of mobility and load.

The primary function of any communication network is to deliver data packets between end points with an acceptable success rate and tolerable delay. It is therefore important to establish if the proposed key distribution mechanism degrades the performance of the network. We limit our scope to the routing and upper layers; to save space we do not consider message overhead occurred on the lower layers.

In Fig. 3, it can be seen that the PDR of the "CBR reference" simulation corresponds closely with that of the "CBR with CertRelay" (CBRwC) simulation[6]. We claim that the impact on network performance is negligible for 0.1m/sec, 5m/sec and 20m/sec mobility. As per design specification, CertRelay exploits mobility; as the mobility increases the CBR and CBRwC simulations become even more correlated (see Fig. 2). The mobility characteristic of MANETs is widely regarded as a limiting factor, as it is a major contribution to route failures. The close relation between the CBR and CBRwC at 0.1m/sec indicates that CertRelay not only turns mobility around as an aid, but in contrast to previous efforts [3] does not rely on mobility. We believe that [3] mainly indicates that mobility can aid security on the application layer. We thus make a novel contribution and show that mobility can aid security in MANETS on the routing layer, but *without* forcing security to become dependent on mobility.

To place the PDR vs. load results into context, the average CBR packet end-to-end delay is shown in Fig. 3. The figure confirms that CertRelay does not add

[6] Note that the "CBR reference" simulation was performed with a standard ns-2 installation with no modifications. The implementation of such a simulation in ns-2 is straightforward and widely accepted as a suitable benchmark.

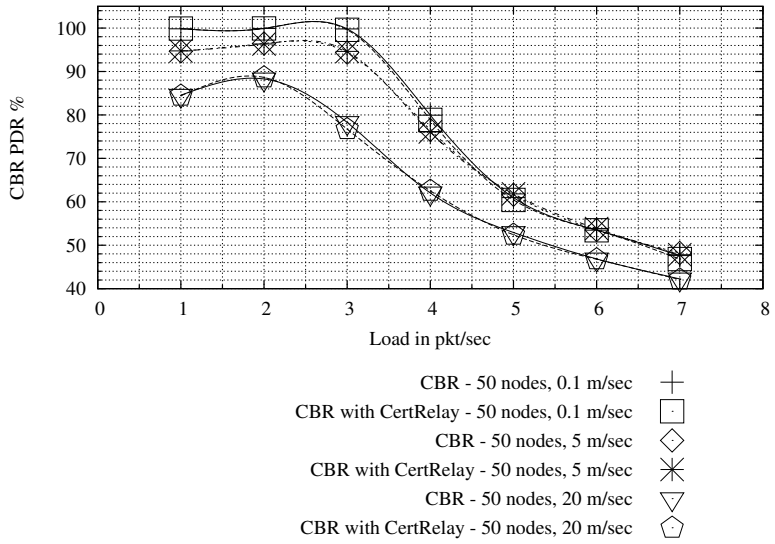


**Fig. 2.** CertRelay's CBR packet delivery ratio % vs. load in pkt/sec for 0.1m/sec, 5m/sec and 20m/sec mobility

any significant delay to the delivery of CBR packets for 0.1m/sec, 5m/sec and 20m/sec mobility.

CertRelay avoids dependence on mobility by using only localized (one-hop) communication. The certificates of nodes not within transmission range are relayed along the virtual chain formed by intermediate nodes. The effectiveness of this mechanism relies on the node's channel access success rate, which is MAC protocol specific. Figure 4 shows that this form of communication with the IEEE 802.11b MAC protocol is very effective. As the load increases one would expect a significant decrease in the certificate delivery ratio. What we can see from Fig. 4 is that the average certificate delivery ratio does decrease with an increase in mobility, but does not deteriorate significantly as the load increases. Between 86% and 97% of the certificates sent between nodes on a one-hop basis are delivered. This explains why RN includes its own certificate $Cert_{RN}$ with the request for the certificate of PN in Case 1a, 2a and 2b defined in Table 1; if RN does not have the certificate of PN, then PN most probably does not have the certificate of RN. PN may also require $Cert_{RN}$ when a packet transverses the reverse route or during a future route discovery process. The proposed certificate scheme exploits the successful localized communication to avoid becoming dependent on the routing infrastructure's performance and thus overcomes one of the main problems of ensuring the availability of the key distribution mechanism in MANETs.

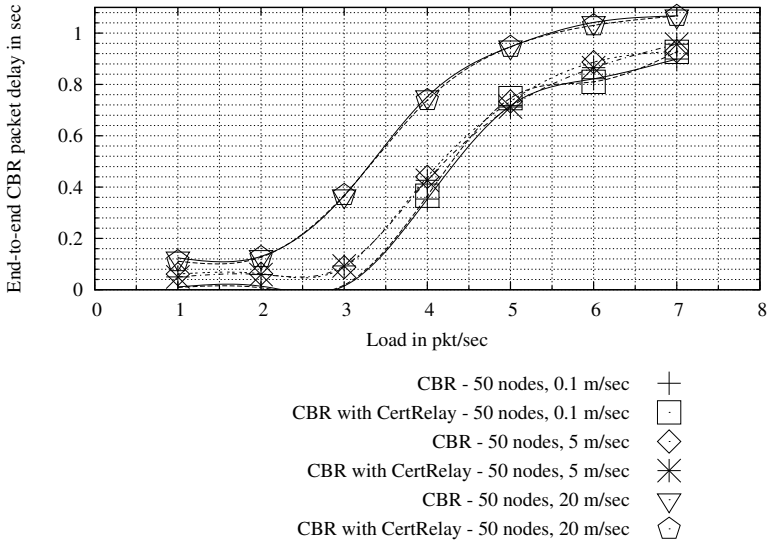


**Fig. 3.** CertRelay's CBR packet end-to-end delay vs. load in pkt/sec for 0.1m/sec, 5m/sec and 20m/sec mobility

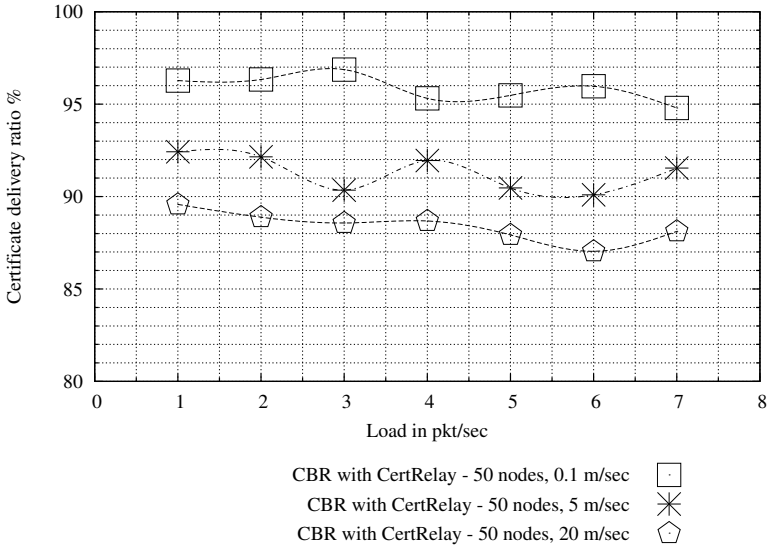


**Fig. 4.** CertRelay's certificate one-hop delivery ratio % vs. load in pkt/sec for 0.1m/sec, 5m/sec and 20m/sec mobility

## 4 Conclusion

The paper identifies a new key distribution problem within the area of key management for MANETs. We propose a novel, key distribution scheme, called Certificate Dissemination based on Message Relaying (CertRelay), as an effective solution to the problem. CertRelay helps nodes to set up security associations in a fully distributed manner without the assistance of an online trusted authority. CertRelay is based on a straightforward procedure to establish security associations in support of the routing infrastructure. The proposed scheme allows nodes to form a virtual chain (by exploiting the routing control messages) along which keying material can be relayed, as required, using only reliable one-hop communication. CertRelay breaks the classic routing-security interdependence cycle and during its entire operation eliminates any explicit dependence on the routing infrastructure for certificate delivery; keying material is relayed along the chain without setting up and maintaining a route. This is an important feature since it implies that CertRelay does not suffer from poor connectivity, aggravated by route failures which are caused by node mobility and error-prone wireless connectivity. In fact, we have shown through simulations that as mobility increases, and the number of route failures increases, the performance of CertRelay improves. The proposed scheme does not introduce any noticeable delay in the set up of security associations, that is, the routing can be secured from the start of network formation leaving no window of opportunity for an attacker.

Capkun et al. [3] have shown that mobility can aid key distribution, but their scheme relies on the temporary proximity of users to exchange certificates. As a result users will experience a delay in the bootstrap of the routing security with evident failure in a stationary setting. Their proposal is however ideal for key establishment on the application layer in a fully self-organized MANET. In this paper we make a novel contribution: to the best of our knowledge, the fact that mobility can be exploited to aid security in MANETs (on the routing layer), *without* depending on mobility, has not been demonstrated prior to this submission.

The simplicity of CertRelay allows for a strong security argument in a widely accepted, formal adversarial model. The nodes of CertRelay exchange only authenticated information on a peer-to-peer basis, which provides provable protection against forgery and undetected modification. The fully distributed scheme preserves the symmetric relationship between the nodes and provides an adversary with no convenient point of attack.

The effectiveness of CertRelay, its low implementation complexity and ease of integration into existing secure routing protocols were verified through coding and simulating the scheme in ns-2. We have shown that CertRelay has negligible impact on the network performance. It was concluded that the message relay mechanism provides an efficient way to distribute keying material. The one-hop certificate exchange success rate varied between 86 % and 97 % which highlighted the effectiveness of localized communication in MANETs.

## References


1. Zhou, L., Haas, Z.J.: Securing Ad Hoc Networks. IEEE Network: Special Issue on Network Security **13**(6) (1999) 24–30
2. Capkun, S., Buttyan, L., Hubaux, J.P.: Self-Organized Public-Key Management for Mobile Ad Hoc Networks. IEEE Trans. on Mobile Computing **2**(1) (2003) 52–64
3. Capkun, S., Hubaux, J., Buttyan, L.: Mobility Helps Peer-to-Peer Security. IEEE Trans. on Mobile Computing **5**(1) (2006) 43–51
4. Hu, Y.C., Johnson, D.B., Perrig, A.: Ariadne: A Secure OnDemand Routing Protocol for Ad Hoc Networks. In: proc. Eighth ACM International Conf. on Mobile Computing and Networking (Mobicom) (2002)
5. Hu, Y.C., Johnson, D.B., Perrig, A.: SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks. IEEE Workshop on Mobile Computing Systems and Applications (2002)
6. Papadimitratos, P., Haas, Z.J.: Secure Routing for Mobile Ad Hoc Networks. In: proc. SCS Communication Network and Distributed System Modeling and Simulation Conf. (2002)
7. Guerrero Zapata, M.: Secure Ad Hoc On-demand Distance Vector (SAODV) Routing (September, 15 2005) INTERNET-DRAFT draft-guerrero-manet-saodv-04.txt
8. Acs, G., Buttyan, L., Vajda, I.: Provably Secure On-demand Source Routing in Mobile Ad Hoc Networks. IEEE Trans. on Mobile Computing **5**(11) (2006) 1533–1546
9. Guerrero Zapata, M.: Key Management and Delayed Verification for Ad Hoc Networks. In: proc. International Conference on High Performance Computing (HiPC): 3rd International Trusted Internet Workshop (TIW) (2004)
10. Guerrero Zapata, M.: Key management and Delayed Verification for Ad hoc networks. Journal of High Speed Networks **15**(1) (2006) 93–109
11. Luo, H., Zerfos, P., Kong, J., Lu, S., Zhang, L.: Self-securing Ad Hoc Wireless Networks. In: proc. Seventh International Symposium on Computers and Communications (ISCC) (2002)
12. Yi, S., Kravets, R.: MOCA: Mobile certificate authority for wireless ad hoc networks. In: proc. of the 2nd Annual PKI Research Workshop (PKI) (2003)
13. Bobba, R.B., Eschenauer, L., Gligor, V.D., Arbaugh, W.: Bootstrapping Security Associations for Routing in Mobile Ad-Hoc Networks. In: proc. IEEE Global Telecommunications Conf. (2003)
14. Guerrero Zapata, M.: Secure Ad Hoc On-demand Distance Vector (SAODV) Routing. ACM Mobile Computing and Communications Review (MC2R) **6**(3) (2002) 106–107
15. Bellare, M., Canetti, R., Krawczyk, H.: A Modular Approach to the Design and Analysis of Authentication and Key Exchange Protocols. In: 30th Annual ACM Symposium on the Theory of Computing. (1998) 419–428
16. Cagalj, M., Capkun, S., Hubaux, J.: Key agreement in peer-to-peer wireless networks. Proceedings of the IEEE (Special Issue on Cryptography and Security) **94**(2) (2005) 467–478
17. Bellare, M., Rogaway, P.: Entity Authentication and Key Distribution. In: Advances in Cryptology - CRYPTO. (1993)
18. The Network Simulator - ns-2, Available at http://www.isi.edu/nsnam/ns or http://nsnam.isi.edu/nsnam/index.php/User_Information

19. Boundec Le, J.Y., Vojnovic, M.: Perfect Simulation and Stationarity of a Class of Mobility Models. In: proc. IEEE INFOCOM (2005)
20. Navidi, W., Camp, T.: Stationary Distributions for the Random Waypoint Mobility Model. IEEE Trans. on Mobile Computing **3**(1) (2004) 99–108
21. Perkins, C.E., Belding-Royer, E.M.: Ad-hoc On-demand Distance Vector Routing. In: proc. The Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA) (1999)