A Cluster-Based Security Architecture for Ad Hoc Networks

M. Bechler*, H.-J. Hof†, D. Kraft†, F. Pählke†, L. Wolf*

*Institut für Betriebssysteme und Rechnerverbund, TU Braunschweig, Germany
[bechler|wolf]@ibr.cs.tu-bs.de

†Institut für Telematik, Universität Karlsruhe (TH), Germany
[hof|dkraft|paehlke]@tm.uni-karlsruhe.de

***Abstract*— Secure communication is very important in computer networks and authentication is one of the most eminent preconditions. However, common authentication schemes are not applicable in ad hoc networks because public key infrastructures with a centralized certification authority are hard to deploy there. We propose and evaluate a security concept based on a distributed certification facility. A network is divided into clusters with one special head node each. These cluster head nodes execute administrative functions and hold shares of a network key used for certification. New nodes start to participate in the network as guests; they can only become full members with a network-signed certificate after their authenticity has been warranted by some other members. The feasibility of this concept was verified by simulation. Three different models for node mobility were used in order to include realistic scenarios as well as make the results comparable to other work. The simulation results include an evaluation of the log-on times, availability, and communication overhead.**

I. Introduction

Ad hoc networks are subject to various kinds of attacks. Wireless communication links can be eavesdropped on without noticeable effort and communication protocols on all layers are vulnerable to specific attacks. In contrast to wire-line networks, known attacks like masquerading, man-in-the-middle, and replaying of messages can easily be carried out. Moreover, deploying security mechanisms is difficult due to inherent properties of ad hoc networks, such as the high dynamics of their topology (due to mobility and joining/leaving devices), limited resources of end systems, or bandwidth-restricted and possibly asymmetrical communication links.

A central issue concerning the design of any service in ad hoc networks is not to rely on any centralized entities, because such entities would obviously be easy to attack, and their reachability could not be guaranteed at all times for all participants of the network. Therefore, it is not possible to implement a centralized, trusted entity for managing public keys of the participants as performed in local area networks or the Internet. Instead, a distributed solution must be found.

In this paper, we propose and evaluate an architecture for securing communication in mobile ad hoc networks. Our approach divides the network into clusters and implements a decentralized certification authority. Decentralization is achieved using threshold cryptography and a network secret that is distributed over a number of nodes. While this basic idea has been proposed earlier [1], its application on a clustered network is a novelty of our work. Our architecture addresses issues of authorization and access control, and a multi-level security model helps to adapt the complexity to the capabilities of mobile end systems. Moreover, an extensive evaluation is given.

In the following, we first give a brief overview of security goals, common techniques for authentication and secret sharing, as well as related work for securing ad hoc networks. In section III, our security concept is described in detail. An important contribution of our work is the evaluation of the security architecture in section IV. We simulated ad hoc networks that use our architecture in order to demonstrate its feasibility and to measure performance and overhead. Those measurements are based upon different mobility models, which are described in this section as well. We also discuss the results and provide information on the configuration of variable parameters. Finally, section V concludes the paper and gives an outlook to further research.

II. Security in Ad hoc Networks

In a security concept, typically striving for goals like authenticity, integrity, confidentiality, non-repudiation and availability, authentication of communicating entities is of particular importance as it forms the basis for achieving the other security goals: e.g., encryption is worthless if the communication partners have not verified their identities before. Authentication of entities and messages can be realized in different ways using either symmetric (3DES, AES) or asymmetric (ElGamal, RSA) cryptographic algorithms (see e.g. [2] for details).

While symmetric algorithms depend on the existence of a preshared key (which does not exist in the general case), authentication by asymmetric cryptography requires a secure mapping of public keys to the owners' identities which is often realized by public key infrastructures (PKI). PKIs use digitally signed certificates to verify a key owner's identity. Each user has to prove her identity to a certification authority (CA) and in turn receives a digitally signed certificate proving the ownership of her public key.

In contrast to fixed networks, a centralized PKI or even a centralized certification authority is not feasible in ad hoc networks, as has been pointed out in the previous section. Distributing the signing key and the functionality of a CA over a number of different nodes by the means of secret sharing and threshold cryptography is a possible solution to this problem, as we will study here.

A. Secret Sharing

Secret sharing schemes realize confidentiality of a cryptographic secret by spreading it across different entities. As secret sharing schemes need no central authorities, they are predestined for ad hoc networks. One secret sharing scheme is threshold cryptography: A trusted dealer divides a secret D into n parts so that the knowledge of k parts ($k \leq n$) allows the reconstruction of the secret, which is not possible with the knowledge of $k-1$ or fewer parts. This is called a (k, n) threshold scheme [3]. In general, a trusted dealer is a central authority and thus another central target for attacks. To avoid this, the participants have to construct the secret without any central authority. The construction algorithm has to ensure that participants can only transmit correct values and that each participant can verify both secret and shares, which is called verifiable secret sharing [4].

In order to protect the secret from attackers that move around and compromise multiple share holders over a long period of time, a proactive secret sharing (PSS) scheme should be used in ad hoc networks. In PSS schemes, secret shares are changed periodically without changing the secret itself, so an attacker cannot use a secret's whole lifetime to compromise k participants. All information an attacker collected about the secret becomes worthless after refreshing the shares [5]. Threshold shared secret schemes can be transformed into PSS schemes using discrete logarithms [5]. Proactive digital signatures, which are used in our work, are an implementation of PSS schemes [6], [7].

Due to the movement of mobile nodes, the topology of ad hoc networks changes frequently, and moreover, nodes can join or leave the network at any time. Hence, an algorithm for distributing the same key to a different set of participants is required. Such a refresh algorithm [8] can be triggered periodically, event-based, or both.

B. Related Work on Securing Ad hoc Networks

The idea to use a distributed certification authority based on a shared certification key and threshold cryptography for securing ad hoc networks was first presented by Zhou and Haas [1]. It was further developed in the COCA system [9], a general distributed authentication service.

Our approach is based on the same general idea, but introduces several new concepts like a cluster-based network structure, a process for admitting new participants and a framework for access control within the network.

Luo et al. [10], [11] chose a different way to distribute the certification process. They use a specially crafted key sharing algorithm distributing the key amongst all network nodes instead of a subset only. Upon this, Luo et al. build an access control system based on signed tickets issued (using threshold cryptography) by neighbors of the node seeking access. Misbehaving nodes are excluded from service after they have been detected.

Another different solution was proposed by Hubaux et al. [12]. In order to avoid any distributed certification mechanism, the authors instead rely on every participant to issue certificates for other nodes in a web-of-trust manner. Each participant has to store a number of certificates, and two nodes can only communicate securely when the union of their local stores contains a certificate path between them.

III. A Cluster-based Concept for Securing Ad hoc Networks

The security concept described in this section was designed with the main aim of providing a basis for secure communication and access control in ad hoc networks. Providing for secure authentication without relying on single centralized entities is the most important issue; methods for ensuring integrity, confidentiality or non-repudiation for end-to-end communication were not considered in detail, as these can easily be realized using well-known techniques if secure authentication is possible.

Other requirements for the design of the concept were that it should support open networks, allowing new nodes to join without any mutual a-priori knowledge, it should allow fine-grained access control for services and resources in the network, and it should be scalable to support hardly predictable network sizes and react quickly to dynamic changes.

A. Clustering

In order to make our concept scalable, to avoid expensive long-range traffic, and to enhance availability by providing service locally, we partition an ad hoc network into a number of clusters. In each cluster, exactly one distinguished node – the cluster head (CH) – is responsible for establishing and organizing the cluster. Gateways (GWs, which need not necessarily be CHs) manage communication with adjacent clusters. The CHs are responsible for sending CH beacons in their clusters, containing administrative information for the cluster members, e.g., lists of nodes and GWs in the cluster. Also, GWs periodically transmit GW beacons to inform their respective clusters about adjacent clusters.

Clustering is also used in some routing protocols for ad hoc networks. Routing is then typically divided into two parts: routing within a cluster (intra-cluster) and routing between different clusters (inter-cluster). One solution for such a scenario is the zone routing protocol, a combination of proactive intra-cluster and reactive inter-cluster routing; communication between two clusters is always routed via GWs [13].

If a cluster-based routing protocol is used, the clusters established by the routing protocol can also be used for our security concept, and some additional advantages are to be expected. However, as we do not want to limit the applicability of the security concept to ad hoc networks with particular

routing protocols, we do not require that clustering is provided by a routing scheme. In case no clusters are given from outside the security part, they are formed as needed: Nodes finding no existing clusters create some themselves, and existing clusters are merged and split on demand. The techniques used for this are described in section III-C.

B. Conceptual Building Blocks

In our concept, a network-wide distributed certification infrastructure forms a basis for securing end-to-end communication by public key cryptography. Additional security of communication links within single clusters is provided by symmetric encryption. For controlling access to resources and services, authorization certificates are used. These building blocks will be described in more detail now.

1) Network-Wide Certification Infrastructure: The basis for our security concept is the use of public key cryptography for ensuring authentication, integrity and confidentiality. Every node participating in the network holds a self-generated key pair, which is used for providing end-to-end security between arbitrary nodes.

Public keys are distributed in the ad hoc network using certificates issued by a trustworthy CA. In contrast to PKIs common in fixed networks, the CA is distributed: It is formed by a subset of all network nodes. For issuing certificates, a certain share (e.g. a majority) of these nodes must actively take part. This concept has two advantages: Firstly, availability is enhanced, because certificates can be issued even if some certification nodes are not reachable. Secondly, the security infrastructure becomes more resistant against intruders, as it can tolerate the compromise of single nodes without the CA as a whole becoming compromised.

In this work, we assign the role of the distributed CA to the cluster heads of the network. Regarding the protocols used for generation, management and usage of the common certification key and for organizing the whole ad hoc network, the CHs therefore form a logical network, the so-called CH network. The private key of the CA is distributed over the CHs, i.e., every CH holds a fragment of the whole key.

The extent of an ad hoc network in respect to our architecture is determined by the extent of a CH network sharing a single private key, i.e. forming a single distributed CA; the shared key is also called the *network key*. More than one network, each having a different network key, can be neighbors in the same area (or even overlap, if clustering is independent of routing). They may or may not be merged into a single network (cf. section III-C.6) at a later point in time.

2) Intra-Cluster Security: Independent of end-to-end security measures that can be built upon the asymmetric key of every node, we use a cluster-wide symmetric key that is known to the cluster's nodes. This key can be used, e.g., to protect all traffic on the links between the nodes. This may be useful for cluster internal traffic that is not protected by other means, and also to hide information like source and destination addresses of transmitted packets from eavesdroppers not belonging to the cluster. The benefits of such a link-wise encryption are similar to those of the encryption in IEEE 802.11 or Bluetooth; it can replace such mechanisms where they are too weak, or it can be integrated with them to provide key management functions.

3) Node Status and Authorization: A new node that joins a cluster has an initial status of a guest with practically no access rights. Only when its public key is signed by the CH network (after authentication has completed successfully), it becomes a full member and can acquire additional access rights by having authorization certificates issued to it. In contrast to identity-based key certificates, authorization certificates can be issued by any network node managing a particular service or resource, like a printer or Internet access. Such services or resources can then be used by the subject of the certificate, who can also transitively grant access to other nodes if the certificate allows it.

For the initial authentication of new nodes and for judging their trustworthiness when granting them access rights, additional trust relationships are needed that do not yet exist for nodes unknown to the network. Therefore, new nodes are obliged to first acquire a certain number of warranty certificates from other network nodes. Warranting nodes can be, e.g., immediate neighbors to the new node, where personal contact between human users is possible and allows for authentication on other than technical levels. In this respect, warranting nodes are similar to registration authorities in conventional PKIs. The more warranty certificates a node collects the more certain is its authentication. Considering this, the CH network can give more or higher level access rights to new nodes holding more than the minimum number of warranty certificates.

C. Details and Protocols

In the following, some procedures and mechanisms are elaborated in more detail, and some of the used protocols are described in an informal manner.

1) Key Distribution and Key Refreshment: The network key, which is shared amongst the CHs of an ad hoc network, is created using proactive secret sharing according to the Digital Signature Scheme [14] (cf. section II-A). As the composition of the CH network changes dynamically when CHs join or leave the network, the secret shares also have to be renewed regularly, because the number of shares needs to be adapted to the number of CHs. Apart from that, it has to be made sure that the key shares are renewed after a certain period of time in order to make it hard for a moving attacker to compromise a number of k CHs over time. In our approach, we always combine joining or leaving of CHs with a key share renewal and only schedule additional renewals if the CH network remains unchanged for some time.

The public key of the CH network must be known to all nodes in the ad hoc network. It is propagated via the CH beacons, which are broadcasted periodically in every cluster. Besides of the public network key, a CH beacon also contains the CHs own public key, a list of nodes of the current cluster including their status, and a list of gateways connecting to adjacent clusters.

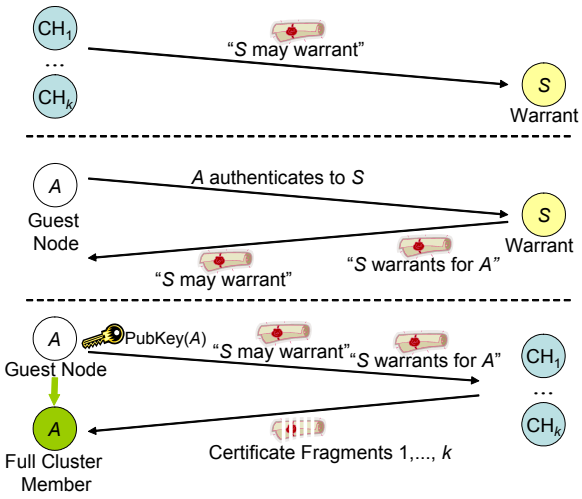

Fig. 1. Authentication process

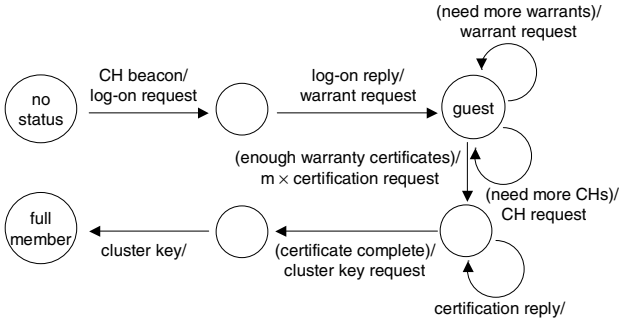

Fig. 2. States of a new node during log-on

2) Log-On Procedure: The log-on procedure described in the following is the means for a new node to join a network by becoming a guest node first and a full member later.

In order to log on, a new node first has to find a cluster. If it receives CH beacons, it sends its log-on request to the cluster's CH. The new node and the CH negotiate some parameters (like the number of warranty certificates required and how the symmetric cluster key is to be used later on), and the new node becomes a guest. If, instead, a nodes does not receive any CH beacons, it establishes its own cluster and acts as a CH of this cluster. For this, it generates a secret symmetric cluster key and starts to transmit CH beacons.

For authenticating themselves to the network, new nodes need warranty certificates (WarrantCert). Such certificates can be acquired from warrants, i.e. from full members that have been granted the privilege of warranting by the network because they are believed to be trustworthy. A new node A may request a signature of the CH network if it possesses a (previously negotiated) number of WarrantCerts. Each of these certificates is signed by a warrant S to guarantee its authenticity, and also includes a period of validity:

WarrantCert(A) := Node(A), PubKey(A), Validity(t),
Fct("S warrants for A"), Sign(S)

A warrant may only vouch for a node if it has verified the node's identity. One method of securing the message exchange necessary for this is to use a location-limited side channel [15], [16], i.e. a channel where the users can control which devices are communicating. How this is done in detail depends of the deployment scenario: In the simplest case – e.g. on conferences where network nodes are personal devices – this could be visual contact and voice communication between users and physical contact (wired, or infrared) between devices. In less "intimate" scenarios – like vehicle communication on motorways – other methods like directed short-range radio or number plate recognition are needed. Another possibility, which may be useful in some cases and has the advantage of being remotely applicable, is using some certificate of a trusted external root CA the public key of which the warrant happens to know.

When CHs are being asked for certificate shares by a new node, they first have to make sure that the issuers of the given WarrantCerts are really authorized to vouch for a guest. This is verified using warranty authorization certificates (WarrantAutCert). Each warrant S sends a copy of its certificate to A:

WarrantAutCert(S) := Node(S), PubKey(S),
Fct("S may warrant"), Sign(CH-Network)

Both certificates together can be used to request a signature for A's public key from the CH network. The CHs check the WarrantAutCert and the WarrantCert presented by A and send their shares of an identity certificate if all the certificates are valid. After A collected enough certificate shares, it can complete its identity certificate:

IdCert(A) := Node(A), PubKey(A), Validity(t),
Sign(CH-Network)

Now having its key signed, A is a full member. The CH sends the symmetric cluster key to A (encrypted with A's public key).

Fig. 1 illustrates the message exchange during a successful log-on: The top fraction shows a WarrantAutCert being issued to a warrant (at some earlier time), in the middle a new node asks a warrant for a WarrantCert and receives it together with the warrant's WarrantAutCert, and at the bottom the new node sends WarrantCerts and WarrantAutCerts to a number of CHs and receives IdCert fragments. Fig. 2 summarizes the states a new node goes through during this process.

In order to ask for identity certificate shares, A has to know about at least k CHs in a (k, n) threshold scheme. If A does not already know enough CHs, it can send a query for further CHs to its own CH. As the CH is in regular contact to other CHs in the CH network, it is able to provide a list of the network's CHs to the requesting node. Alternatively, A can extract information on further clusters and their CHs from received GW beacons.

The procedure of warranting and key certification utilizes the restricted resources sparingly, as only few messages are necessary to get a key certificate. If a side channel is used for authentication, a (k, n) threshold cryptography system needs $2k$ messages for requesting and receiving certificate shares. If the requests to and replies from w warrants are transmitted

over the ad hoc network, another $2w$ messages are needed.

Of course, how easy or difficult it is for a new node to find some warrants depends of the distribution of the warrants and the degree of mobility. However, there is no fixed time limit for the process of finding warrants, and the new node can already use the network as a guest while searching for warrants. Besides, it is assumed that most full members are granted the warranting privilege after some time, so there should be no lack of potential warrants.

3) Interaction with Routing: In section III-A, we mentioned the possibility of reusing cluster structures of the underlying routing protocol of the ad hoc network for the security concept as well. This also allows to offer "secure routing" in the sense of restricting the set of nodes that are considered for forwarding packets.

In general, routing in cluster-based ad hoc networks is different for intra-cluster and inter-cluster communication. If proactive routing is used for intra-cluster routing, a sender may specify (by setting a flag in the routing header) if either only full members or all nodes in the cluster are allowed to forward a packet. As a result, each node possibly has to manage two routing tables, one for routing via full members only, and one considering all nodes in the cluster. In case of a reactive intra-cluster routing strategy, the sender has to find a route before transmitting a packet. As the CH beacons also propagate the status of the involved nodes along the route, a sender is able to specify its security requirements in the route request packet, which limits the route replies to, e.g., full members. Note that each CH individually defines the cluster's security guidelines for intra-cluster routing. Those guidelines specify, e.g., encryption of link state updates for proactive routing.

Inter-cluster routing must be adapted as well: For both reactive and proactive routing, a sender has to notify the GWs about whether each node or only full members may forward packets. This is necessary because only GWs know the status of nodes in adjacent clusters.

4) Gateways: Each node N that gets in contact with a foreign cluster can potentially act as a gateway. Optionally, the permission to act as a gateway can be controlled using gateway authorization certificates (GwAutCert) signed by the CH network:

GwAutCert := Node(N), PubKey(N), Fct("Gateway"),
Sign(CH-Network)

A potential gateway notifies both its CH and the CH in the discovered cluster about the contact. The address of the new CH can be requested from the foreign node the gateway first got in contact to. In turn, both CHs send the information about the new gateway in their CH beacons, and the new gateway itself starts to transmit GW beacons containing its public key and its status in the corresponding clusters (guest node or full member, possession of a GwAutCert). If the discovered cluster was not associated with the network previously, the gateway will initially be a guest node there although it is a full member in its original network.

5) Delegation of Cluster Heads: If a node is no longer able to act as a CH, it delegates this role to another trusted node within the cluster. This avoids an expensive re-configuration of the cluster and possibly of the whole network. If a CH is looking for a successor, it queries for a node that will continue the CH functionality further on. Once a trustworthy successor is determined, the old CH securely migrates its state to the successor and sends a signed broadcast message containing the new CH's identity, so all nodes in the cluster are able to adapt themselves to the new CH and to its CH beacons they will receive. Nodes that do not receive this broadcast message will consider the CH beacons they receive after the change as foreign. However, they are still full members as their network certificates are still valid.

Apart from the nodes in the cluster, the CH also has to notify the members of the CH network about the CH delegation; this is realized by separate encrypted messages to each other CH. As the old CH transfers his share of the private network key to the new CH, the sharing of the network key will be unaffected. During the next refresh of the key shares, the new CH will be updated instead of the old one.

Without explicit delegation of the CH function, a failing CH results in the break-up of the cluster. Former cluster members have to join neighboring clusters or form a new cluster after a new CH has been found. Because of the changes in the CH network and in the cluster membership of the abandoned nodes, this is a rather costly process and should therefore be avoided.

6) Merging Networks: The merging of two complete networks into a single network is one of the most difficult and expensive operations to occur. As the two network keys cannot be mixed, one of them must be dropped and the other distributed over the whole network. All the certificates that had been signed with the dropped key have to be reissued in the long run, although it is possible to keep the dropped key for some period of time to facilitate this process. Possibly, it might become necessary to adapt the (k, n) threshold for the changed number of nodes and CHs in the new network.

Before merging starts, a decision must be made on which of the networks is to remain. In the simplest case, one of the networks consists of just one cluster. Its single CH can then be integrated into the other network quite simply, or, if the single cluster has only few members, it can be dissolved, leaving its members to join the other network on their own. In contrast, merging two bigger networks is rather difficult. The decision about the remaining network depends on parameters like the number of CHs and the number of nodes that would have to apply for new certificates.

A requirement for the incorporation of a new CH into an existing network is that a particular number of nodes from the existing network have expressed their trust into the new CH by issuing warranty certificates to it. If the CH has collected enough certificates, it receives its share of the private network key with a following key share refreshment. Otherwise, it has to delegate the CH role to another node in its cluster that has collected enough warranty certificates. The integration will not

TABLE I
LEVELS OF CONTROL OVER ADMITTED USERS

User or provider group	*Credential*
all nodes	none
all full members	secret cluster key or certified public node key
specific nodes	authorization certificate
directly trusted nodes	any of the above, or a preshared key

be possible if no node in the new cluster possesses enough certificates. In that case, each node has to join the remaining ad hoc network separately. For the merging of whole networks, mutual consent can be found by an explicit decision of a majority of the CHs of both networks. This is necessary because the networks in effect have to trust each other. Various possible decision mechanisms are issues for further study.

7) Access Control: Access to services and resources can be controlled using authorization certificates. Entities that are responsible for controlling access to a particular service or resource, or the service provider or owner of the resource itself, can give authorization certificates to the users they wish to admit. These certificates include the public key of the subject and some authorization information. Nodes may pass those rights transitively to other nodes if they also hold a permission for doing so.

Apart from using authorization certificates, simpler methods of access control can be applied as well. Altogether, for the providing as well as for the using side, four different levels of control over the group of admitted users, respectively trusted providers, can be distinguished; Table I shows them together with the credentials used for access control.

8) Adaptable complexity: Different types of keys (symmetric cluster key, asymmetric public key) and certificates can be used for communication. Nodes decide in each case which security level is needed and use the appropriate encryption. In order of increasing complexity these levels are:

1) No encryption
2) Secret cluster key (intra-cluster only)
3) Public node keys, directly exchanged
4) Public node keys, certified by the CH network

Allowing an adaptable complexity is an advantage for nodes with low resources, which can choose a suitable security level. However, if nodes cannot agree to a common level of security, communication may be impossible.

IV. EVALUATION

In order to evaluate our concept, we developed a simulation prototype using the OMNeT++ simulation framework [17]. OMNeT++ is an object-oriented, discrete event simulation system developed at Budapest University. The goals of our simulation were twofold: One goal was the proof of concept, i.e. to demonstrate that our security architecture can be deployed in ad hoc networks. The second goal was to determine the performance of our approach. Our intention was to evaluate the performance of the security protocols independently of the communication characteristics. Hence, we assumed an abstract link layer with a maximum transmission range of 180 m and a delay of 100 ms per hop, which is similar to the values published by Compaq, Cisco, and Siemens for their IEEE 802.11b products (see respective web pages of the companies). Furthermore, we assumed no bandwidth restrictions, no transmission collisions, no packet drops, and no bit errors in our simulation in order to measure the undisturbed performance of the security mechanisms.

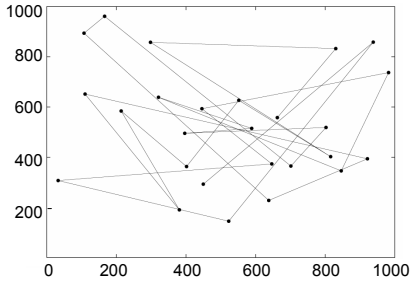

Fig. 3. Mobility pattern for random waypoint

A. Evaluation Scenarios

The mobility patterns of mobile nodes in an ad hoc network are a very fundamental factor for its evaluation. Hence, we examined our security architecture with three different scenarios: random waypoint, a motorway scenario, and a conferencing scenario. Due to space constraints, we only present the results of the random waypoint and the motorway scenario in this paper.

1) Random Waypoint: The random waypoint mobility pattern [18] divides a participant's movement into two phases. First, a participant waits for a random idle time period. In the second phase, the participant chooses a random destination within the simulation area and moves towards this destination at a randomly chosen speed. If the participant reaches this destination, the process continues with the first phase. Waiting time, destination within the simulation area, and speed are chosen with a uniform distribution. Fig. 3 illustrates an exemplary trace of a mobile node moving according to random waypoint in a square of $1000\,\mathrm{m} \times 1000\,\mathrm{m}$.

Although random waypoint is a rather unrealistic mobility model, it is widely used for the performance evaluation of various ad hoc networking aspects. Hence, we consider this mobility model in our evaluation as well in order to allow a comparison of our work with the research of others.

2) Motorway Scenario: Future vehicular communication scenarios will be a very important application field of ad hoc networks. In such scenarios, vehicles traveling along a motorway organize themselves in ad hoc networks for exchanging local data. Due to the high mobility of vehicles, the network topology is very dynamic and, thus, challenging for ad hoc networking protocols. Several research projects deal with vehicular ad hoc networking, e.g., FleetNet [19] or CarNet [20].

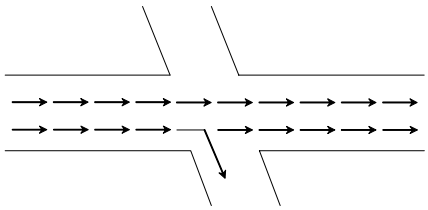

Fig. 4. Movement in the motorway scenario

In order to model the motorway scenario, we used a modified city selection mobility pattern [21], [22]. The simulation area consists of a road network with specific characteristics of each road (e.g., a speed limit). Each vehicle starts at a predefined point and seeks for an arbitrary destination. The mobility algorithm calculates the shortest path to this point and the vehicle travels along this path to the destination. Once a vehicle arrives at the destination, it waits a predefined period of time and repeats the algorithm from the beginning. Note that vehicles have to follow predefined paths on their way to the destination. Compared to other mobility patterns, the movement of a vehicle is possible along the roadway only. Within these regions, we use speed vectors depicting the direction of the vehicles' movements. Their absolute value simulates road conditions, speed limits etc. According to their speed v, we modeled three types of vehicles:

- trucks with 40 km/h $\leq v \leq$ 80 km/h,
- slow cars with 60 km/h $\leq v \leq$ 120 km/h, and
- fast cars with 100 km/h $\leq v \leq$ 220 km/h.

We simulated two motorway sections of 2 km length each, connected by a motorway interchange as illustrated in Fig. 4. The speed and direction of each vehicle depends on the road speed vector at the current position and the current vehicle's speed. At the motorway interchange, a vehicle leaves the motorway with a probability of 0.1.

B. Simulation Results

For our evaluations, we implemented a simulation of our security architecture using OMNeT++. We first measured the performance with the random waypoint mobility model. The simulation area spanned a square of 600 m $\times$ 600 m in which 15 nodes moved around (if not stated otherwise; there were also runs with 30 and 60 nodes). As described previously, a node's transmission range was 180 m with a delay of 100 ms per hop. Cluster heads broadcasted their beacons over 2 hops every 20 s. In order to achieve full membership, a guest node required three warrants to receive its identity certificate. The lifetime of this certificate was chosen randomly between 200 s and 300 s. For the evaluation, we varied some of the parameters to determine their overall performance impact. Each measurement comprised 50 simulation runs with a simulated duration of 240 s each. The time for seeking warrants was ignored because it includes social factors that are impossible to simulate; warrants answered to warrant requests immediately and with a positive ratio of 85 %. The consideration of social interaction would result in higher log-on times, and the number of unsuccessful log-ons due to timeouts would possibly increase as well. In order to route the IP packets through the ad hoc network, we used an OMNeT++ implementation of the Fish-eye State Routing protocol [23]. Although routing has an obvious impact on the timing values measured, this influence has shown to be very small: The differences to some later experiments with "direct" routing on the shortest path without any routing protocol overhead were rather marginal.

In order to determine the impact of the scenario chosen, we compared the random waypoint measurements with the motorway scenario. For the motorway scenario, we used a simulation area of 2 km $\times$ 2 km with the model described in section IV-A.2. In this area, 60 vehicles traveled along the predefined road. For some measurements, we also tried to examine the impact of different parameters (e.g., the number of nodes) on the performance. We evaluated the following performance criteria:

- *Log-On time:* the period of time between the receipt of a CH beacon and the full membership of a mobile node (for which a guest node had to find three warrants).
- *Availability:* the percentage of all nodes within the ad hoc network that are able to communicate securely.
- *Overhead:* the additional network traffic caused by our security architecture for the different types of mobile nodes (cluster heads, gateways, and full members).

1) Log-On Time: In order to determine a mobile node's log-on time, we measured the time period between the first receipt of a CH beacon and the achievement of a full membership. Within this log-on time, the mobile node has to find three warrants and collect three shares of the identity-based key certificate. Finally, it combines the shares and requests the symmetrical cluster key from the cluster head. Ideally, the log-on procedure should be short to guarantee the mobile nodes a quick admission to participate at the ad hoc network.

A simulation run for the log-on time period was performed as follows: In the first step, we assigned each of the 15 nodes a randomly chosen type (cluster head, gateway, or guest node). Cluster heads transmitted their CH beacons every 20 s via 2 hops. In the second step, all nodes began to move randomly and the guest nodes tried to obtain a full membership.

Fig. 5 shows the measured distribution of log-on times. We aggregated the results in 5 s steps; we also aggregated the probabilities for log-on times larger than 100 s. In this configuration, the log-on of a guest node requires 24.9 s on average and approximately 25 % of the guest nodes achieve full membership within the first 10 s. The distribution has peaks at all multiples of the CH beacon interval (20 s), i.e. at 25 s and 45 s respectively. Note that the simulation also contains unsuccessful log-ons, which are restarted after a predefined period of 30 s. The log-on procedure of a guest node can fail for one of the following reasons:

- The guest is not able to collect enough warranty certificates within the predefined period of time.
- After having received a CH beacon, the guest cannot

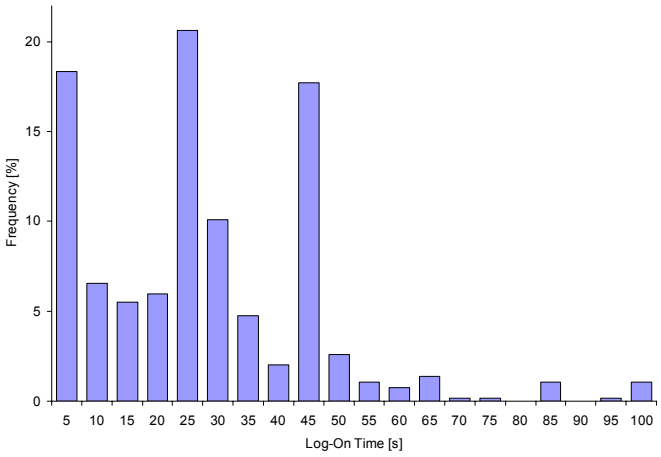

Fig. 5. Log-on (15 nodes, random waypoint)

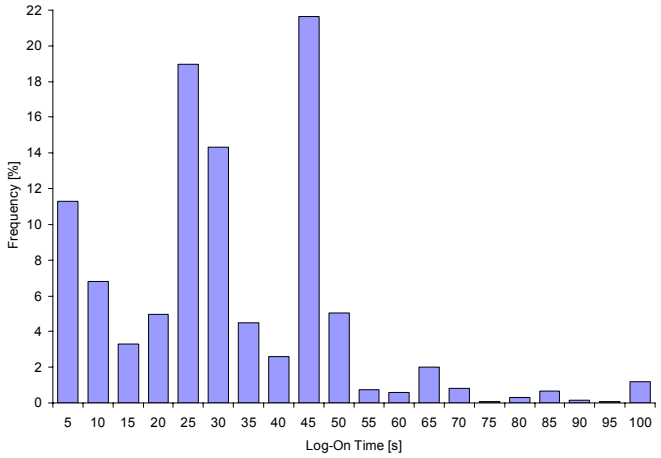

Fig. 6. Log-on (30 nodes, random waypoint)

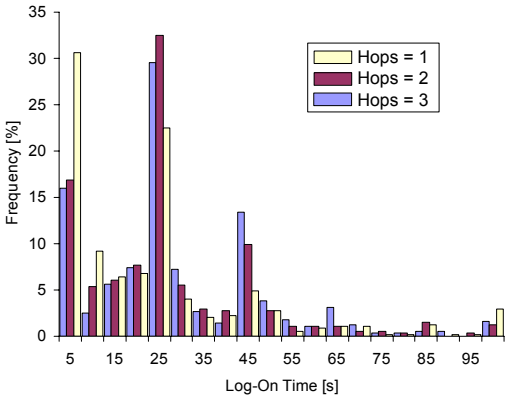

Fig. 7. Log-on times for different cluster sizes

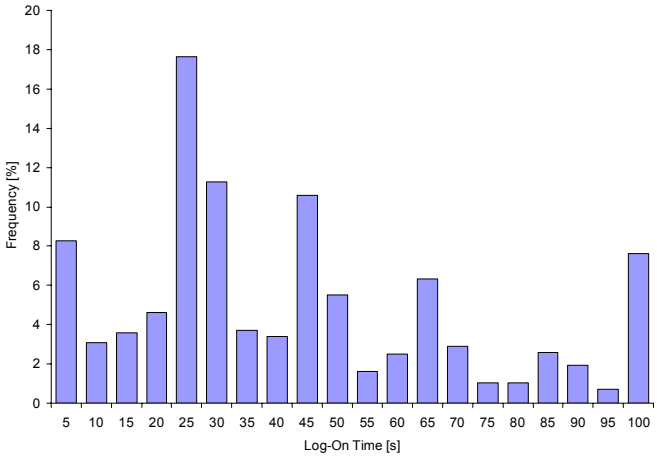

Fig. 8. Log-on (60 nodes, motorway)

communicate with the CH because the network topology has changed in the meantime.

- A merge of CH networks occurs during the log-on procedure.

Separate examinations showed that the typical log-on time for successful log-ons only was approximately 2 s, and about 50 % of the guests nodes are able to become full members within the first second.

In order to determine the effect of the nodes' density on the log-on time, we repeated the random waypoint simulation with 30 mobile nodes. The remaining parameters were not modified. As illustrated in Fig. 6, the log-on times increased with the number of nodes; the average log-on time increased to 28.7 s. Notice again the peaks at 25 s and 45 s, which are correlated to the CH beacon frequency. The increase of the log-on times is caused by the increased probability for the merging of CH networks in the setup phase since due to the higher density of nodes, each node has contact to more neighbors. As described previously, this effect is costly (cf. section III-C.6) and influences the log-on of the nodes.

Another important configuration parameter is the cluster size: The probability of CH network merges increases with increasing cluster size. In order to determine the effects of the cluster size, we repeated the measurements from Fig. 5 with different cluster sizes, determined by the number of hops a CH beacon is forwarded. Fig. 7 shows the results with cluster sizes of 1, 2, and 3 hops respectively. As expected, small cluster sizes resulted in very quick log-on times; in this measurement, the average log-on time was 21.1 s for one hop, 24.5 s for 2 hops, and 27.1 s for 3 hops. Further measurements (not shown here) approved that the log-on times of successful log-ons changed only marginally.

A different situation occurs in the motorway scenario, as illustrated in Fig. 8, for which we used 60 mobile nodes. The performance was by far lower compared to the previous scenarios. The average log-on time was 41.5 s, whereas only 11 % of the guest nodes were able to log-on within the first 10 s.

2) Availability: The availability is another important parameter of a security architecture for ad hoc networks. The following fraction defines the availability:

$$\text{availability} = \frac{\text{number of full members}}{\text{total number of nodes}}$$

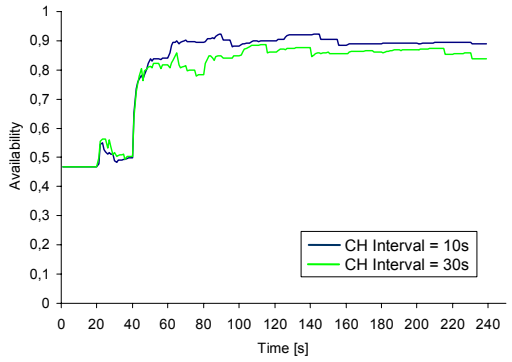

Fig. 9. Availability (15 nodes, random waypoint)

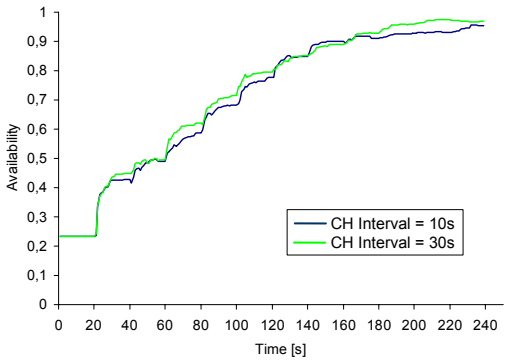

Fig. 10. Availability (60 nodes, motorway)

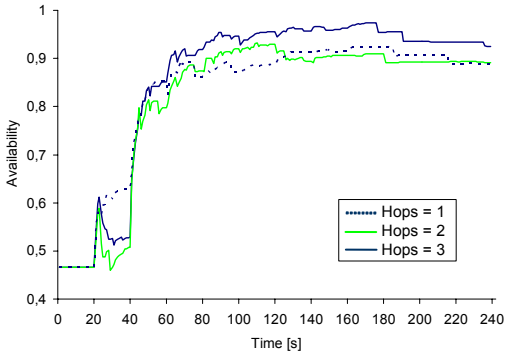

Fig. 11. Availability depending on the cluster size (random waypoint)

After a successful log-on with the cluster head, secure communication is possible until a mobile node's certificate expires. The validity of the certificate was chosen randomly between 200 s and 300 s. The merge of two CH networks plays an important role for the availability. In this case, the availability will decrease as a new secret network key must be generated. Hence, all mobile nodes have to obtain a new certificate, which must be signed with the new network key.

First, we studied the impact of the CH beacon frequency on the availability. For our measurements with the random waypoint mobility model, we used the same parameters as specified in the previous section, apart from the varied CH beacon interval. Fig. 9 illustrates the results with CH beacon intervals of 10 s and 30 s. We can clearly identify two phases: In the first phase (from the beginning to approximately 55 s), the structure of the security architecture establishes slowly. During this phase, approximately 58 % of the nodes are able to communicate securely. The low average availability results from several CH networks being merged in the beginning. In the second phase (starting at 55 s), the cluster topology and the security infrastructure are well established. About 90 % of the mobile nodes are able to communicate securely. In general, our measurements showed that the CH beacon interval does not affect the availability significantly.

In order to investigate the impact of the mobility scenario on the availability, we repeated the simulation using the motorway model. We used the same parameters as described in the previous section, and varied the frequency of CH beacons. Fig. 10 shows the results: An interesting observation is that the availability increased much slower compared to the random waypoint model; it takes about 125 s until the availability reaches an average of more than 90 %. These characteristics result from the relatively high number of cluster merges that occurred throughout the simulation time. Like in the random waypoint simulation, the impact of the frequency of CH beacons was very small.

Finally, we examined the impact of the cluster size on the availability. For this, we used the random waypoint model with the cluster sizes of 1, 2, and 3 hops respectively. Fig. 11 shows the results of the three measurements. The larger the clusters are, the longer a mobile node is associated to its cluster. As a result, it is easier for a guest node to find a sufficient number of warrants in order to achieve full membership.

3) Communication Overhead: Obviously, security protocols always cause additional overhead, which burdens both network and end systems. In this section, we consider the "costs" incurred for the establishment and maintenance of our security architecture. We therefore measured the number of packets/s that were transmitted in the simulation using the same parameters as in the previous measurements. Note that this measurement is independent of the radio technology; we also do not consider the overhead caused by the routing protocol. Fig. 12 shows the overhead (in packets/s) for a CH beacon interval of 10 s and 30 s using random waypoint. Especially in the setup phase, the overhead is very high. The significant increase of overhead after 20 s coincides with the increase of availability illustrated in Fig. 9. When the security infrastructure is established (after about 55 s), the overhead decreases slowly to less than 50 packets/s. As observed previously, the CH beacon interval seems to have very little effect on the overhead in the random waypoint scenario. Note that the overhead refers to the overall ad hoc network, i.e. the aggregated traffic of all communication links.

A comparison with the motorway scenario illustrates the influence of different mobility patterns. Fig. 13 illustrates the results of the measurements using the motorway scenario with the previous configuration. In contrast to random waypoint, the CH beacon interval has a significant impact on the overhead. Comparing a CH interval of 10 s with a CH interval of 30 s, we

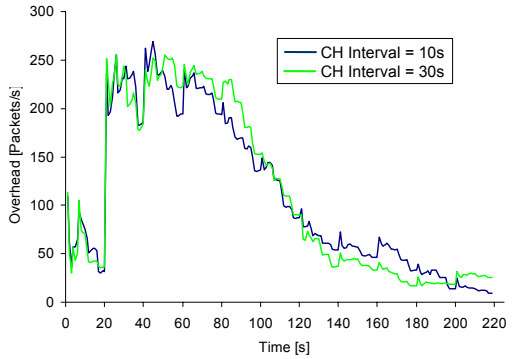

Fig. 12. Overhead (random waypoint)

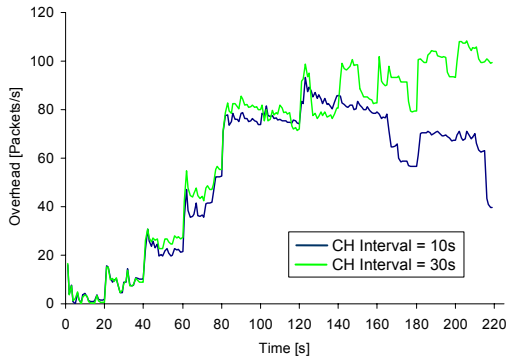

Fig. 13. Overhead (motorway)

found that starting at about 120 s, the average overhead further increases for a CH interval of 30 s, whereas it decreases for a CH interval of 10 s. Moreover, the peaks in the motorway scenario are by far less significant compared to the random waypoint model (100 packets/s compared to 250 packets/s). Notice again the correspondence to Fig. 10: with an increasing availability, the overhead also increases.

We also studied the influence of the cluster size on the overhead. In Fig. 14, we examined the overhead for the random waypoint mobility model with cluster sizes of 1, 2, and 3 hops. This simulation shows that the overhead remains small if the CH beacons are only broadcasted across one hop. The overhead increases with increasing cluster size. The reason for this effect is that the probability of merging two CH networks increases with an increasing cluster size, because separated clusters will discover each other earlier. In this case, the merging of two cluster head networks requires a new log-on of some nodes, which causes additional overhead.

Finally, we examined the distribution of node types and the overhead caused by each type. We assumed a cluster radius of 2 hops and considered the overhead caused by the Fish-eye State Routing. In our measurements of the random waypoint model with 15 nodes, we had an average of 20.4 % cluster heads, whereas 61.8 % of the nodes (on average) were full members, and 17.8 % of the nodes acted as gateways. A comparison of the overhead traffic caused by the different types of nodes showed the following results:

- cluster heads caused 47.5 % of the traffic,

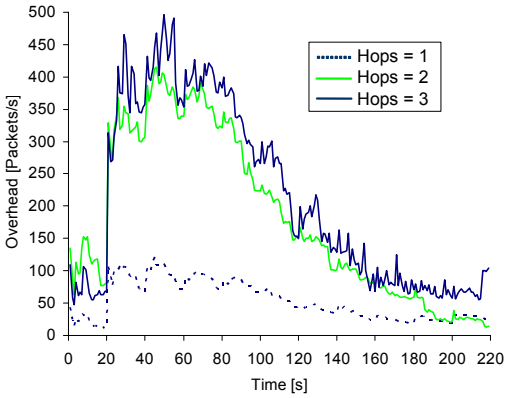

Fig. 14. Overhead depending on the cluster size (random waypoint)

- full members caused 18.3 % of the traffic, and
- gateways caused 34.2 % of the traffic.

C. Discussion

From the evaluation, several interesting performance aspects of our security architecture can be observed. One important result is that the mobility patterns of mobile nodes have a highly significant impact on performance. All measurements revealed remarkable differences between the random waypoint and the motorway scenario.

Our measurements also show the impact of different parameters. Whereas the frequency of CH beacons seems to have little impact (motorway scenario) or almost no impact at all (random waypoint) on the availability, the overhead differs noticeably for different CH beacon intervals in the motorway scenario. The cluster size plays an important role for the overhead caused by our security architecture. One drawback is the considerable additional load for cluster heads and gateways: Together, they generate about 80 % of the additional network load.

Besides the examined aspects, further parameters are also relevant for the performance of the described security architecture. One parameter is the time period a mobile node waits for incoming CH beacons. If this timer expires (without receiving a CH beacon), the mobile node nominates itself as a cluster head. This timeout parameter is of interest in the bootstrapping phase. For our simulations, we set the timer to the CH beacon interval plus a randomly chosen time (up to 10 s). In other measurements, this configuration turned out to be suitable for the scenarios deployed.

Another tuning parameter is the minimum number of required warrants. This value should be chosen carefully, as a high value results in low availability, whereas a low value might violate the trustworthiness of our security architecture. We suggest a value of about 40 % of the number of full members. This value proved to be a good choice. In this context, the validity of the received certificate needs to be configured according to the requirements of the given scenario. As stated previously, we used a validity between 200 s and 300 s for a mobile node's identity certificate.

Concerning the security aspects of our protocols, the parameter k of the (k, n) threshold scheme needs to be determined. If k is configured too low, an attacker might be able to compromise the security architecture before the network key is refreshed. In case of a high value for k, a guest node has to contact more cluster heads. Hence, its log-on time increases, or it even fails if the guest node cannot find enough cluster heads. Of course, it must be ensured that k will be always lower than n, even if the number of cluster heads varies. In our simulations, a value of about 50 % of the total number of cluster heads seemed to be a good choice.

Of course, additional fine tuning of those and other protocol parameters may still result in further optimizations. However, depending on the actual deployment scenario, optimizations will always require a tradeoff between required security and availability, acceptable overhead, and expected performance.

V. Conclusion

In this article, we introduced a cluster-based architecture for a distributed public key infrastructure that is highly adapted to the characteristics of ad hoc networks. In order to adapt to the highly dynamic topology and varying link qualities in ad hoc networks, we consequently avoided any central instances that would form single points of attack and failure. Instead, the ad hoc network was divided into clusters, and the cluster heads jointly perform the tasks of a certification authority. Our concept uses a proactive secret sharing scheme, which distributes the private network key to the cluster heads in the ad hoc network. Instead of a registration authority, arbitrary nodes with respective warranty certificates may warrant for a new node's identity. Based upon this authentication infrastructure, we provide a multi level security model ensuring authentication, integrity, and confidentiality. Authentication itself is realized in two stages. First, a node gets the status of a guest node. After sufficient authentication, the node will become a full member. An additional important feature is the possibility to delegate the cluster head functionality to another node. We also pointed out how the security concept can be integrated with routing protocols in order to achieve routing on secure paths.

In order to evaluate our approach, we used two different scenarios in our simulations: random waypoint and motorway. Based upon these mobility models, the evaluation of the mobile nodes' log-on times, the availability of the security infrastructure, and the overhead shows that it is possible to deploy a security architecture with an acceptable performance and overhead. We also showed how different parameter variations affect the performance. A very interesting observation was that the mobility model highly impacts the behavior of such an security architecture.

Future work should address the impact and limitations of the communication technology deployed. These investigations allow an evaluation of our security architecture in a real-world scenario. Furthermore, the configuration of further parameters (e.g., timeouts for the log-on procedure) still needs to be evaluated. We are also going to evaluate our approach with additional ad hoc routing protocols and application scenarios.

References

[1] L. Zhou and Z. J. Haas, "Securing ad hoc networks," *IEEE Network*, vol. 13, no. 6, pp. 24–30, 1999.

[2] B. Schneier, *Applied Cryptography*. John Wiley, 1996.

[3] A. Shamir, "How to share a secret," *ACM Comm.*, vol. 22, no. 11, 1979.

[4] T. Pedersen, "A threshold cryptosystem without a trusted party," in *Advances in Cryptology, Proc. Eurocrypt'91*, ser. LNCS, vol. 547. Springer-Verlag, 1991.

[5] A. Herzberg, M. Jakobsson, S. Jarecki, H. Krawczyk, and M. Yung, "Proactive public key and signature systems," in *ACM Conf. on Computer and Comm. Security*, Zürich, 1997.

[6] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, "Proactive secret sharing, or: How to cope with perpetual leakage," in *Advances in Cryptology, Proc. CRYPTO'95*, ser. LNCS, vol. 936. Santa Barbara, California: Springer-Verlag, Aug. 1995, pp. 339–352.

[7] K. Takaragi, K. Miyazaki, and M. Takahashi, "A threshold digital signature issuing scheme without secret communication," IEEE P1363 Study, Nov. 2000.

[8] Y. Desmedt and S. Jajodia, "Redistributing secret shares to new access structures and its applications," George Mason Univ., Tech. Rep., 1997.

[9] L. Zhou, F. B. Schneider, and R. van Renesse, "COCA: A secure distributed on-line certification authority," *ACM Trans. Computer Systems*, vol. 20, no. 4, pp. 329–368, Nov. 2002.

[10] H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang, "Self-securing ad hoc wireless networks," in *Proc. 7th IEEE Symp. on Comp. and Communications (ISCC)*, Taormina, 2002.

[11] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "Providing robust and ubiquitous security support for mobile ad-hoc networks," in *Proc. 9th International Conference on Network Protocols (ICNP)*. Riverside, California: IEEE, Nov. 2001, pp. 251–260.

[12] J. Hubaux, L. Buttyan, and S. Capkun, "The quest for security in mobile ad hoc networks," in *Proc. ACM Symp. on Mobile Ad Hoc Networking and Computing (MobiHOC)*, Long Beach, Oct. 2001.

[13] C. Perkins, *Ad Hoc Networking*. Addison-Wesley, 2001.

[14] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Robust threshold DSS signatures," in *Advances in Cryptology, Proc. Eurocrypt'96*, ser. LNCS, vol. 1070. Saragossa: Springer-Verlag, 1996, pp. 354–371.

[15] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong, "Talking to strangers: Authentication in ad-hoc wireless networks," in *Proc. Symp. on Network and Distributed System Security (NDSS)*, San Diego, Feb. 2002.

[16] F. Stajano and R. Anderson, "The resurrecting duckling: Security issues for ad-hoc wireless networks," in *Proc. 7th International Workshop on Security Protocols*, ser. LNCS, vol. 1796. Springer-Verlag, 1999, pp. 172–194.

[17] A. Varga, "The OMNeT++ discrete event simulation system," in *Proc. European Simulation Multiconference (ESM)*, Prague, Czech Republic, June 6–9, 2001.

[18] D. B. Johnsson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 1996, vol. 353.

[19] W. Franz, R. Eberhardt, and T. Luckenbach, "Fleetnet – Internet on the road," in *Proc. 8th World Congress on Intelligent Transport Systems*, Sydney, Oct. 2001.

[20] R. Morris, J. Jannotti, F. Kaashoek, J. Li, and D. Decouto, "Carnet: A scalable ad hoc wireless network System," in *Proc. 9th ACM SIGOPS European Workshop*, Kolding, Sept. 2000.

[21] J. Markoulidakis, G. Lyberopoulos, D. Tsirkas, and E. Sykas, "Mobility modeling in third-generation mobile telecommunications systems," *IEEE Personal Commun. Mag.*, vol. 4, no. 4, 1997.

[22] V. Davies, "Evaluating mobility models within an ad hoc network," Master's thesis, Colorado School of Mines, 2000.

[23] M. Gerla, X. Hong, and G. Pei, "Fisheye state routing for ad hoc networks," Internet Draft, IETF, June 2002, draft-ietf-manet-fsr-03.txt.