Proceedings of the 1st IEEE Consumer Communications and Networking Conference (CCNC'2004), January 2004, Las Vegas, Nevada, USA.

Distributed Access Control for Consumer Operated Mobile Ad-hoc Networks

Daniel Kraft
Institute of Telematics
University of Karlsruhe, Germany
daniel.kraft@tm.uka.de

Günter Schäfer
Department of Electrical Engineering, TKN-Group
Technical University of Berlin, Germany
schaefer@ee.tu-berlin.de

***Abstract*— We propose a concept for realizing access control in mobile ad-hoc networks to exclude nodes that do not contribute to the provision of network services from using them. Each node observes the behavior of its neighbors in order to build up opinions about their willingness to take part in different network activities. In turn, service is provided only to nodes that have positive opinions about them. A precondition for assigning opinions to nodes is having a means for authentication; we rely on a web-of-trust structure for this, where all nodes can issue certificates for others after they have verified their identities. A proactive certificate management system makes it possible – as the evaluation results presented in this paper show – to find optimal certificate chains between given keys and to quickly retrieve the needed certificates.**

I. Introduction

Mobile ad-hoc networks enable groups of mobile users to acquire the benefit of having network services anywhere, without depending on any pre-installed infrastructure. Apart from giving a certain amount of freedom in whether to accept the terms of the third party providing such infrastructure – if it exists –, this becomes especially important in situations were there is nobody to provide infrastructure or where the third party that could do so has no interest in doing it.

Because mobile ad-hoc networks are based solely on components owned by the participating users and are only provided by their joint efforts, these participants have a legitimate interest in protecting their components and resources (like their battery power) from unauthorized use by parasitic individuals who wish to benefit from using network services but are not willing to contribute to their provision. Shutting out such unwanted participants – analogous to infrastructure networks, where a provider wants to shut out users that do not pay him for the service he provides –, is the task of an *access control scheme*. This paper proposes such a scheme for ad-hoc networks which needs only minimal user interaction and thus is suitable for use in consumer operated networks.

Central components for access control and related functions in infrastructure networks normally have a special security level with higher privileges, are well-protected and always available. Such special fixed components can not be realized in ad-hoc networks, because neither could their all-time availability be guaranteed nor would it be clear which node should be legitimated to have such special privileges. In contrast, all of these tasks have to be accomplished in a distributed manner. Furthermore, there is nobody to define a general security policy which describes who should have access to which services and how an applicant should be checked. Such questions must effectively be answered by single participants in ad-hoc networks, possibly applying rules that all participants or a sufficiently sized subset of them agreed upon. In our approach, every node bases its decision about admitting another node on its *opinion* about the other node's willingness to cooperate and bring in its own resources for the good of the whole network. The opinion about a node is derived from *observation* of the other node's behavior, or, if no observations of its own have been made, a trusted third node can be asked about its opinion.

A precondition for access control (as well as for other security services) is a means for authenticating entities and messages, because firstly, observations must be securely associated with the participants that caused them and secondly, measures that exclude misbehaving participants from the network (i.e., not forwarding their packets, in the first place) also must only affect the actual culprit. As a basis for authentication, we rely on asymmetric cryptography: Every node needs to have a (normally self-generated) private and public key pair and be able to produce and verify signatures; this is inevitable in an open environment, where the existence of pre-shared keys between entities cannot be assumed. Apart from its use for access control, the key pair can also be employed for realizing end-to-end security between nodes.

Authentication by asymmetric cryptography requires a secure mapping of public keys to the owners' identities, which is often realized by public key infrastructures in fixed networks: Each user has to prove her identity to a certification authority and in turn receives a digitally signed certificate proving the ownership of her public key. As a centralized public key infrastructure, like any centralized component, is not feasible in ad-hoc networks, we rely on a web-of-trust approach (like PGP [1] does) for certification: Every participant can issue certificates for others. As a replacement for a directory service for storing issued certificates and delivering them to anyone who needs them for authenticating keys in public key infrastructures, we realize a fully distributed key and certificate management system specifically designed for answering frequent requests for existing certificates quickly and causing only a minimum of network traffic.

From the great variety of possible scenarios for the deployment of mobile ad-hoc networks we chose a reference scenario that can be described as "conference-like": The network nodes are personal devices of individual human users who are also available for making decisions if needed, and the total number of nodes is limited to about 100 to 1000.

The rest of this paper is organized as follows: Section II gives an overview of related work. Sections III and IV present the proposed concept in more detail, the former concentrating on authentication, key management and certificate distribution, and the latter covering trust management and access control. Section V gives preliminary results of our evaluation of the concept. In Section VI we draw some conclusions.

II. Related Work

Controlling access to mobile ad-hoc networks can, from a somewhat different point of view, also be seen as trying to bring the participants into providing service for the good of the whole network. Work in this area normally falls into one of these alternative categories:

- Detection based: Nodes that misbehave are being detected and then excluded from further service. Disadvantages of this method include relatively severe effects of erroneous observations due to mobility or malicious intervention and the possibility for detected nodes to change their identities and start over with clear records. Examples for such approaches, which are typically highly dependent of the used routing protocol, are [2], [3], [4], [5].
- Motivation based: Correct behavior is being rewarded. Often, a kind of virtual currency is used, amounts of which are dealt out by nodes using services and received by those providing them (like in [6]), or they are transferred between a kind of bank accounts after the respective events have been reported to the bank. The bank needed in the latter variant must be trusted by all the participants and furthermore be highly available; in [7] it is therefore required to be located in an infrastructure network which is at least intermittently reachable from the ad hoc network.

In our approach, we tried to combine mostly the advantages of both of these alternatives by recording positive as well as negative observations while building opinions about other nodes. In particular, erroneous observations are handled gracefully and the changing of identities is rendered quite useless, because it always means to throw away positive records.

Concerning authentication and key management, there are fundamentally different approaches as well. One idea is using a distributed certification authority based on a shared certification key and threshold cryptography for securing ad hoc networks. It was first presented in [8] and further developed into a general distributed authentication service [9], later. Having groups of nodes with privileged functions (like the subset of nodes carrying shares of the certification keys) tends to throw up difficult questions like how to legitimate the nodes that take extra privileges or what to do when independently formed groups need to be merged.

An other method of distributing certification is chosen in [4]: Using a specially crafted key sharing algorithm, the key is distributed amongst all network nodes instead of only a subset. Upon this, the authors build an access control system based on signed tickets issued (using threshold cryptography) by neighbors of the node seeking access.

The idea of coping without any distributed certification mechanism, like we do in our concept, was also used earlier in [10]. There, each participant is required 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. Algorithms for finding a good selection of certificates to store are evaluated, but no general service for retrieving any needed certificate is provided.

III. Authentication and Key Management

A. Certification

Our concept uses a web-of-trust approach for certification, where each participating node can issue certificates for other nodes. When issuing a new certificate, the issuer has to be sure, that the certified information – the binding of a key to an identity – is correct. This implies that communication between subject and issuer during the certification request must be safe from manipulation, e.g. by an attacker in the middle. Wireless transmission alone is clearly not suited for this purpose; it must be complemented by a kind of location-limited side channel [11], [12] like a direct physical or an infrared link or at least by voice communication between human users. The issuing of certificates is the one and only operation where users are explicitly required to be involved, as this is the only way to fix a key to an identity that cannot be easily replaced.

When verifying certificates, the verifier needs an authentic key of the issuer as well as some trust into the issuer's sincerity and competence concerning certification. The former is provided by a chain of certificates, each having the issuer of the next chain-link as subject, ending in the target key and starting from a known authentic key (like one owned by the verifying node). Such a chain is also called a *certificate path* in the following. The latter is provided by a certification policy in typical public key infrastructures, to which all certification authorities adhere. As we cannot rely on a common certification policy to be observed by all nodes in open ad-hoc networks, every certificate here contains a component describing its quality, which is derived from the issuer's estimation of the subject's certification sincerity and competence.

B. Certificate Storage

Certificates are always stored during their whole validity period by their subject node (unlike in [10]) in order to both make sure that they cannot be accidently lost (their issuing being a quite expensive operation) and provide for a canonical way to find them: When any node needs a particular certificate, it can always request a copy from the subject node. Certificates that have been retrieved in this way (or that have been seen being retrieved by others) are candidate for being stored in a local cache. A criterion for which certificates to keep may be

their usefulness for authenticating the caching node: They may be considered the more valuable (namely for authenticating the caching node's identity to others in the future) the smaller their distance to the caching node in any certificate chain ending in the caching node is.

Note that the mentioned way of finding certificates still has a weakness: A certificate could still be useful in the middle of certificate paths even though the subject has left the network in the meantime. The method described in the next section handles this problem as well.

Supplying a certificate that another node was looking for is a service rendered and is awarded by a positive rating by the requesting node and all observing nodes.

As a single human user tends to meet others with similar interests or working areas repeatedly, it may be useful to issue long-lived certificates and keep them much longer than the time of the staying in a single ad-hoc network, thus reducing the number of new certificates that need to be issued in a new network that the same nodes happen to be part of.

C. Finding Certificate Paths

One task during authentication is to find a certificate path from the verifier's key to that of the node to be authenticated. In fixed networks, the search process for such a path is often accomplished using a directory service that holds all certificates. Assuming that all certificates can be found at their subject node, a "naive" approach to conduct the search in an ad-hoc network would be to start from the target key and ask the respective node for all keys that issued certificates for it. When recursively repeating this step for all issuer keys found, the result will either contain the key of the verifier at some point, or there will be no more issuers left, meaning that no path exists between the considered keys. The cost of this (breadth-first search) process is at least $O(n)$ message transmissions for a network with n nodes, depending on whether the recursive requests are handled by the asked nodes themselves or by the verifying node; in the latter case, the requests and replies to the verifier add another factor of about $\sqrt{n}$, but give more control over the process to the verifier, e.g. allowing her to exclude certain keys from the path.

When the operation of finding certificate paths is frequently used – and this is the case in our access control approach described in section IV – a more resource preserving method becomes highly desirable. There are three different kinds of information about certificates that a node must know in order to find a certificate path:

- Which certificates do exist? This information is needed for finding paths, and therefore it is important to be as complete as possible. The knowledge about every single certificate can be decisive in whether a path exists at all or for a paths quality. Certificates are created relatively seldomly (requiring active user participation), and so this kind of information changes only slowly.
- Where are cached copies of a certificate stored? When a path has been identified, the composing certificates have to be fetched, and for this, it is advantageous to know locations where they are cached. With that, it becomes unnecessary to rely on only the possibly hard-to-reach subject node for delivering the certificate, and furthermore, a certificate can also be used after the subject node has left the network, but cached copies still exist. This kind of information may change often, but its accuracy is not vital.
- How far away are the cached copies? This kind of information is obviously useful to decide which copy can be fetched with the smallest cost. Strictly speaking, it belongs to the routing protocol instance, but for one thing, it can be useful for the certificate fetching service to be mostly independent of routing, and secondly, distance information is also useful to decide how far the information about a cache needs to be distributed.

In our certificate management scheme, we use a special protocol to distribute certificate information inside of the ad-hoc network proactively, i.e. before any specific certificate requests are being handled. Every node builds a local certificate index, which holds the nearest caching location and its distance for every certificate inside the network. Initially, each node's index contains only those certificates that have been issued by other nodes for itself. After each change in the local index or in the node's neighborship (the start of operation counting as a change as well), a node broadcasts all changes (or the whole index, if new neighbors appeared) to its neighbors, who incorporate them into their local indices as follows:

- Certificates that were unknown are added, together with the nodes where the information about them came from.
- Changes in caching location and distance of already known certificates are taken in if they come from the same node that caused the insertion of the certificate.
- Information about nearer copies of already known certificates replaces existing information, even if it comes from different nodes.

Note that by this protocol, existence information is effectively flooded throughout the network, while location information is only distributed in environments of nodes holding cached copies of certificates. Using the information about all existing certificates, each node searches for paths locally and afterwards fetches the nearest copies of the certificates it needs.

IV. Trust Management and Access Control

A. Opinion Values

Decisions about admitting other nodes are based on opinions about the other nodes' willingness to cooperate, in our concept. Such opinions are derived from positive and negative observations of neighbor behavior using a special metric [13] that can account for uncertainty in absence of observations.

The opinion about an other node's willingness to cooperate is one that is important for access control, but there are others, e.g. about the capability and willingness to issue correct certificates, or about the consistence between an other node's and one's own opinions. More varieties can be added for every additional service in the network. Therefore, each node

maintains for each other node a list of opinions concerning various properties of that node.

B. Observation of Neighbor Behavior

Observations of neighbor behavior, from which opinions can be derived, are made by listening to all network traffic of neighbors, taking advantage of the promiscuous nature of wireless transmission. In general, if a neighbor reacts to a request as is expected, this is counted as a positive observation, and if it obviously reacts in a deliberately wrong way or misuses network resources, this is counted negatively.

When sending packets, a node waits (holding some state for the packets) for the next hop router to forward it. If it fails to do so inside of a given time limit, this is counted as misbehavior. Most wireless transmission protocols (like those of IEEE 802.11, e.g.) include an acknowledgement for transmitted packets, so the sender can be sure that the next hop router has really received it. Furthermore, if an acknowledgement packet can be received, the next hop router also seems to be in transmission range, so the forwarded packet probably will be heard as well. But of course there is no absolute certainty – the next hop router could have moved out of range or be shielded from the sender during between acknowledging and forwarding –, and because of this, one failed forwarding only slightly decreases the sender's opinion about the next hop router.

For to be able to associate observations with the respective neighbors, it is necessary to verify the network packets' senders authenticity. Therefore, each packet is signed by its source node. For verifying the signature, each forwarding node needs an authentic copy of the public key of the source node. The methods for finding certificate paths and locating certificate copies described in section III-C have been designed with this in mind, i.e. they should be able to handle numerous requests without causing too much network overhead.

C. Using Opinions

Access control to the network service as a whole is implemented by not forwarding packets from sources that are not positively trusted to be cooperative. This means that every node before forwarding any packet always checks its opinion about the source of the packet. If this opinion is not positive, the packet is not forwarded. Note that here, as with observation, every forwarding node needs to have the authentic key of the source node of each packet, so we depend once more on the fast key and certificate management described above.

Opinions can also be used as criteria for access to other services and resources. The relevant categories of opinions of course depend of the service or resource under consideration.

D. Exchanging Opinions

For the case where a particular opinion about a particular node is needed for making an access decision, but no such opinion has yet been formed by local observations, we allow to ask other nodes, e.g. one's neighbors, about their opinion; of course, the asking node must have some trust into the asked nodes for this (ref. [13] gives methods for applying opinions about others to the opinions they expressed). The answer to such an "opinion request" is signed by the node who's opinion it represents, and therefore resembles a certificate, i.e. it can safely be given to and stored by any other node, including the subject node itself, for later use. However, the life time of such an "opinion certificate" must not be very long in order to restrict misuse for keeping opinions high after having changed one's behavior to the worse.

E. Warrants

With only the mechanisms described above, nodes that are completely new to a network suffer from the fact that no opinions about them exist inside the network (even after they found someone to certify their public keys, which is always the first step in joining a network). As a consequence, they would be denied access to any service. One more building block allows them to change this situation: They may send "warranty requests" to established nodes inside the network, e.g. to the nodes they want to communicate with in the first place. We're assuming that the users of nodes that are contacted in this way are often willing to help the new users, either because they know them personally or because they already agreed to communicate using some out-of-band means.

Given that the node receiving a warranty request is in fact willing to help, it issues a signed "warranty certificate" and sends it back to the new node. On its way through the network, every node forwarding this certificate shifts a particular share of positive opinion (provided some such exists) from the issuer's account to that of the new node. This way, the new node gains some trust inside the network and can start to communicate. It can give the "borrowed" trust back to the warranting node later when it has gained some positive opinions itself. The warranting node bears the risk of losing the trust it lent out, and is therefore motivated not to do this carelessly, which would undermine the access control scheme.

V. Evaluation

A. Security Issues

Our concept introduces special protocol messages (certificate information and requests as well as opinion and warranty requests) that are being exchanged between instances of the access control system on different nodes. To enable smooth operation, these messages must be transported through the network independent of the opinions that forwarding nodes have about their source. Such specially treated messages are therefore at risk of being used for covert channels by attackers that try to circumvent access control. One unspecific countermeasure is to restrict the forwarding rate for these messages. A more specific analysis of actual vulnerabilities in this or other respects and methods to protect the protocols is currently being worked upon.

B. Performance Issues

The need to be able to verify each packet's sender in each forwarding node (for associating an opinion about the sender

with it as well as for making sure that no previous "forwarder" has forged the packet to produce positive observation) motivated the decision to put an individual asymmetric signature in each network packet. Obviously, this is quite expensive, and for very small devices it might too slow to be practicable. The signature can however be reused for end-to-end integrity protection. We are also still investigating a method that avoids the asymmetric source signature and replaces it by a hop-by-hop symmetric one; trust in the authenticity of the source would then be based upon trusting the previous forwarder to have already verified the authenticity.

Apart from the computation power requirement just mentioned, another critical resource is storage space. The need to store keys of and opinions about potentially any other node in the network obviously makes the feasibility of the approach depend on the network size. A detailed analysis of the effects of storage size, grade of mobility and traffic locality upon the amount of overhead traffic caused for verifying keys and ask about opinions that could not be stored locally is planned. However, we believe that smaller networks in the order of magnitude of 100 to 1000 nodes should be manageable for current mobile laptop computers.

C. Simulation

In order to obtain some proof of concept and to find the conditions and parameter settings under which the concept can be used beneficially, it is currently being implemented and analyzed in a simulation environment. As a first step, we evaluated the proactive key and certificate management: As mentioned above, a high performance of this essential component is crucial for the functioning of the whole access control system. Some interesting results of this evaluation will be presented in the following.

The simulation uses the OMNeT++ discrete event simulation system [14] developed by András Varga. We tried to keep any factors out of the simulation that are not in the target of the evaluation and would only influence the results in an unnecessary and hard-to-calculate manner. Therefore no particular transmission technology or routing protocol properties were implemented beyond the bare minimum.

The network for the simulation runs described in the following consisted of 36 nodes in a 600 m x 600 m square area, moving using the random way-point mobility model with uniformly chosen way-points inside the area, speeds between 1 and 10 m/s, and waiting times between 1 and 30 s. The nodes had an average of 10.11 neighbors inside their transmission range of 180 m, with a standard deviation of 4.09.

Packets were routed using optimal paths of minimal spacial distance. Application layer data was generated according to the following model: Each node randomly chose a peer node from the network, and exchanged a randomly chosen number of packets with it, each being answered by the peer. After this, the initiator node waited for some time before starting over. Overall, this resulted in an average rate of about 1.24 packets per node and second.

TABLE I
FIGURES ABOUT 10 RUNS OF 600 S EACH IN A 36-NODE NETWORK

Certificates	issued		1874
	discovered		65592
	cached		14508
Updates	sent		11280
	received		109353
Certificate requests	requests		12636
	retries		48
	delivered		12634
	failed		2
Payload packets	sent		268165
Key lookups	requests		580646
	answered	locally	604
		from cache	570197
		non-locally	9300
	failed	key unknown	169
		path not found	80
		no path exists	247
		key not trusted	49

Public key certificates were generated with randomly chosen issuers and subjects (but paired, i.e. they always "both certified each other's key") at the beginning of each run, resulting in an average of 5.2 (and a minimum of 1) certificates for each node's key. For the time being, no new certificates were issued during simulation runs.

Table I displays some figures cumulated about 10 independent simulation runs of 600 seconds each. The numbers in the first block show that all existing 1874 certificates have been discovered at some point in time by all of the 35 nodes each that didn't have them from the beginning ($1874 * 35 = 65590$; the two extra discoveries happened because the information about two certificates had been deleted when the attempt to fetch them had failed repeatedly, because the holding nodes were disconnected from the rest of the network for a while). Although all the certificates were known to all nodes, only an average of 40.3 certificates (of about 187.4 existing in each run) have actually been fetched and used by each node.

Fig. 1 shows the time distribution of the certificate discovery and later fetching activity. While the bulk of the discoveries happens within the first 7 seconds of the run and a few more follow within the first 25 seconds (and the two exceptions mentioned above at $t = 123.31\ s$), the actual fetching activity decreases more slowly (and roughly exponentially) and only subsides after about 315 seconds. No more certificates needed to be fetched after this point in any of the simulation runs, which indicates firstly that the (comfortably low) average number of 40.3 certificates per node was really sufficient to verify all the keys, and secondly that really most of the certificates (nearly 95 %, to be more exact) were known to the whole network after only 7 seconds, thereby allowing the nodes to find nearly optimal certificate paths quickly.

Key lookup activity, in comparison, was sustained at about the same level of an average of 2.69 lookups per node and second during the whole simulation time interval; most of

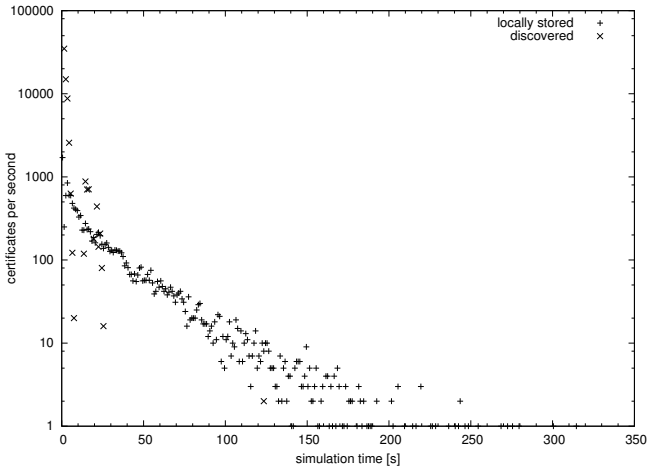

Fig. 1. Certificate Discovery and Caching

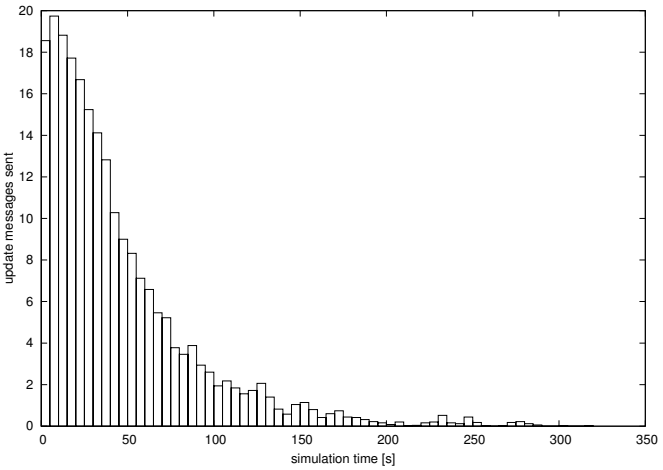

Fig. 2. Update messages sent

these lookups were being answered from the local key cache, as the figures in the "key lookups" block of Table I illustrate. Only few key lookups failed, and more detailed scrutiny showed that most of these failures happened during the initial 47 seconds. Overall, this shows that the provisioning of key and certificate management and, based on this, authentication, for the rest of the system worked very well.

Figure 2 shows the time distribution of certificate information update messages sent inside the network, and reflects the subsiding of updating activity after about 317 s. This is due to a lack of changes in existence information, because the sending of updates for pure topology changes was not implemented yet. Even though this will add a certain low level of permanent activity, there is also still room for optimization: As location changes are less important than existence changes, there will be a longer delay for the respective update messages, and we expect this to reduce the over-all rate of update messages considerably.

VI. Conclusion

We presented an access control scheme using observation based opinions that avoids some major advantages of other approaches by recording positive as well as negative behavior of other nodes.

Automatically observing and analyzing the behavior of other network nodes depends on quickly being able to find their authentic keys, which we provided for by adding a proactive certificate management scheme that makes existing certificates quickly known all over the network and also maintains information about the nearest copies of the certificates. Issuing certificates is the only activity that requires aid from the consumer operating the device.

Evaluation has been done for the proactive key and certificate subsystem, which proved to reach a very good view of the whole certificate graph in a short time and was able to retrieve all keys of other nodes by fetching only a small subset of the existing certificates. Further evaluation will concentrate on the other part of the access control system: observation and opinion assessment. There, we hope to answer important questions like whether opinion forming works well enough and reliably recognizes misbehaving nodes, and whether our (or any) access control scheme for ad-hoc networks is actually worthwhile to deploy considering the overhead it causes compared to the expected losses due to unwanted use.

References

[1] P. R. Zimmermann, *The Official PGP User's Guide*. MIT Press, June 1995.

[2] S. Marti, T. J. Giuli, K. Lai, and M.Baker, "Mitigating routing misbehaviour in mobile ad hoc networks," in *Proc. 6th International Conf. on Mob. Comp. and Networking (MOBICOM)*, Aug. 2000, pp. 255–265.

[3] K. Paul and D. Westhoff, "Context aware detection of selfish nodes in DSR based ad-hoc networks," in *Proc. IEEE Global Telecommunications Conference (Globecom)*, Taipei, Taiwan, Nov. 2002.

[4] 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.

[5] S. Buchegger and J.-Y. L. Boudec, "Performance analysis of the confidant protocol," in *Proc. ACM Symp. on Mobile Ad Hoc Networking & Computing (MobiHoc)*, June 2002.

[6] L. Blaževic, L. Buttyán, S. Čapkun, S. Giordano, J. Hubaux, and J. L. Boudec, "Self-organization in mobile ad-hoc networks: the approach of terminodes," *IEEE Commun. Mag.*, June 2001.

[7] B. Lamparter, K. Paul, and D. Westhoff, "Charging support for ad hoc stub networks," *Elsevier Journal of Computer Communications*, vol. 26, no. 13, Aug. 2003.

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

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

[11] 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 Syst. Sec. (NDSS)*, San Diego, Feb. 2002.

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

[13] A. Jøsang, "A subjective metric of authentication," in *Proc. European Symp. on Research in Computer Security (ESORICS)*, ser. LNCS. Springer-Verlag, 1998.

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