Securing Mobile Ad Hoc Networks

Panagiotis Papadimitratos
School of Electrical and Computer Engineering
Cornell University
395 Rhodes Hall, Ithaca NY 14853
papadp@ece.cornell.edu
http://www.people.cornell.edu/pages/pp59/

Zygmunt J. Haas
School of Electrical and Computer Engineering
Cornell University
323 Rhodes Hall, Ithaca NY 14853
haas@ece.cornell.edu
http://wnl.ece.cornell.edu

Abstract

The vision of nomadic computing with its ubiquitous access has stimulated much interest in the Mobile Ad Hoc Networking (MANET) technology. Those infrastructure-less, self-organized networks that either operate autonomously or as an extension to the wired networking infrastructure, are expected to support new MANET-based applications. However, the proliferation of this networking paradigm is strongly dependent on the availability of security provisions, among other factors. The absence of infrastructure, the nature of the envisioned applications, and the resource-constrained environment pose some new challenges in securing the protocols in the ad hoc networking environments. In particular, the security requirements can differ significantly from those for infrastructure-based networks, while the provision of security enhancements may take completely different directions as well. In this paper, we study the schemes proposed to secure mobile ad hoc networks. We expose the primary goals of security enhancements, shedding light on the commensurate challenges, and survey the up-to-date literature on this topic. Then, we introduce our approach to such a multifaceted and intriguing problem. Finally, we identify some open problems and plausible approaches.

1. Introduction

Mobile ad hoc networks comprise freely roaming wireless nodes that cooperatively make up for the absence of fixed infrastructure; that is, the nodes themselves support the network functionality. Nodes transiently associate with their peers that are within the radio connectivity range of their transceiver and implicitly agree to assist in provision of the basic network services. These associations are dynamically created and torn down, often without prior notice or the consent of all parties. MANET technology targets networks that can be rapidly deployed or formed in an arbitrary environment to enable or facilitate communications or to serve a common objective dictated by the supported application. Such networks can be highly heterogeneous, with various types of equipment, usage, transmission, and mobility patterns. Secure communication, being an important aspect of any networking environment, becomes especially a significant challenge in ad hoc networks. This is due to the particular characteristics of this new networking paradigm and due to the fact that traditional security mechanisms may be inapplicable.

The absence of a central authority deprives the network of the administrative and management services that would otherwise greatly facilitate its operation. Instead, MANET has to rely on continuous self-configuration, especially because of the highly dynamic nature of the network. Problems such as scheduling, address assignment, provision of naming services, or formation of network hierarchy, cannot be solved by traditional centralized protocols. Instead, the distributed operation is necessary in all aspects of network control, including basic security-related operations, such as the validation of node credentials. In the fully distributed and open environment of ad hoc networking, the provision of such services not only may incur a high overhead, but also give additional opportunity for misbehaving nodes to harm the network operation.

Moreover, it is assumed that nodes participate in the protocol execution as peers, which implies that potentially any network node can abuse the protocol operation. In other words, it is fairly difficult to make the distinction of trustworthy and supportive nodes based on the network interaction. As a result, it is far less straightforward to determine the protocol or network components that have to be safeguarded, and even more difficult to design adequate security countermeasures.

Meanwhile, the practically invisible (or nonexistent) administrative or domain boundaries make the enforcement of any security measures an even more complex problem. Migrating nodes may face varying 'rules' even when they run the same application, as they move through different network areas and become associated with different groups of nodes. Moreover, they may lack the ground for the establishment of trust associations, that is, the establishment of some type of secret, so that cryptographic mechanisms can be employed.

Below, we will discuss in further detail the vulnerability of mobile ad hoc networks, clarify how security goals may have to be modified, and which types of solutions are

This work has been supported in part by the DoD Multidisciplinary University Research Initiative (MURI) program administered by the Office of Naval Research under the Grant number N00014-00-1-0564 and by the National Science Foundation grant number ANI-9980521.

plausible for different network instances. Although the discussion throughout a great part of the paper lends itself to all types of ad hoc networks, it is important to realize that not all solutions can be applied in all ad hoc networking environments. Moreover, it is necessary to emphasize the relative importance of addressing certain security issues, which can be considered, to some extent, as prerequisites for solutions to other security problems. In the following sections, we will present the challenges posed by the MANET environment, survey the relevant literature, identify the limitations of the proposed approaches, and suggest directions for future solutions.

2. Security Goals

The overall problem of securing a distributed system comprises the security of the networked environment, and the security of each individual network node. The latter issue is important due to the pervasive nature of MANET, which does not allow us to assume that networked devices will always be under the continuous control of their owner. As a result, the physical security of the node becomes an important issue, leading to the requirement of tamper-resistant nodes [24], if comprehensive security is to be provided. However, security problems manifest themselves in a more emphatic manner in a networked environment, and especially in mobile ad hoc networks. This is why in this work we focus on the network-related security issues.

Security encompasses a number of attributes that have to be addressed: *availability, integrity, authentication, confidentiality*, *non-repudiation* and *authorization.* These goals, which are not MANET-specific only, call for approaches that have to be adapted to the particular features of MANET. First, we provide a generic definition of each goal, and, then, we expose in detail the challenges posed by this new networking paradigm.

Availability ensures the survivability of network services despite misbehavior of network nodes; for instance, when nodes exhibit *selfish* behavior or when denial-of-service (DoS) attacks are mounted. DoS attacks can be launched at any layer of an ad hoc network. For example, an adversary could use jamming to interfere with communication at the physical layer, or, at the network layer, it could disrupt the routing protocol operation, disabling the operation of the route discovery procedure. Moreover, the adversary could bring down high-level services. One such target is the key management service, an essential service for an implementation of any security framework.

Integrity guarantees that a message being transferred is not altered. A message could be altered because of benign failures, such as radio propagation impairments, or because of malicious attacks on the network. In addition, integrity viewed in the specific context of a connection, that is, the communication of two or more nodes, can provide the assurance that no messages are removed, replayed, re-ordered (if re-ordering would cause loss of information), or unlawfully inserted.

Authentication enables a node to ensure the identity of the peer node that it is communicating with. Without authentication, an adversary could masquerade a node, potentially gain unauthorized access to resources and sensitive information, and interfere with the operation of other nodes.

Confidentiality ensures that certain information is never disclosed to unauthorized entities. Confidentiality is required for the protection of sensitive information, such as strategic or tactical military information. However, confidentially is not restricted to user information only; routing information may also need to remain confidential in certain cases. For example, routing information might be valuable for an enemy to identify and to locate targets in a battlefield.

Non-repudiation ensures that the origin of a message cannot deny having sent the message. Non-repudiation is useful for detection and isolation of compromised nodes. When a node *A* receives an erroneous message from a node *B*, non-repudiation allows *A* to accuse *B* using this message and to convince other nodes that *B* is compromised.

Finally, *authorization* establishes rules that define what each network node is or is not allowed to do. In many cases, it is required to determine which resources or information across the network a node can access. This requirement can be the result of the network organization, or the supported application, when, for instance, a group of nodes or a service provider wishes to regulate the interaction with the rest of the network. Another example could be when specific roles are attributed to nodes in order to facilitate the network operation.

The *trustworthiness* of mobile ad hoc networks has additional dimensions, such as *privacy*, *correctness*, *reliability*, and *fault-tolerance*. In particular, the resilience to failures, which in our context can be the result of malicious acts, and the protection of the correct operation of the employed protocols are of critical importance and should be considered in conjunction with the security of the mobile ad hoc network.

3. Threats and Challenges

Mobile ad hoc networks are vulnerable to a wide range of active and passive attacks that can be launched relatively easily, since all communications take place over the wireless medium. In particular, wireless communication facilitates eavesdropping, especially because continuous monitoring of the medium, referred to as *promiscuous mode,* is required by many MANET protocols. Impersonation is another attack that becomes more feasible in the wireless environment. Physical access to the network is gained simply by transmitting with adequate power to reach one or more nodes in proximity, which may have no means to distinguish the transmission of an adversary from that of a legitimate source. Finally, wireless transmissions can be intercepted, and an adversary with sufficient transmission power and knowledge of the physical and medium access control layer mechanisms can obstruct its neighbors from gaining access to the wireless medium.

Assisted by these "opportunities" the wireless communication offers, malicious nodes can meaningfully alter, discard, forge, inject and replay control and data traffic, generate floods of spurious messages, and, in general, avoid complying with the employed protocols. The impact of such malicious behavior can be severe, especially because the cooperation of all network nodes provides for the functionality of the absent fixed infrastructure. In particular, as part of the normal operation of the network, nodes are transiently associated with a dynamically changing, over time, subset of their peers; that is, the nodes within the range of their transceiver, or the ones that provide routing information and implicitly agree to relay their data packets. Due to this transient association, it is more difficult to identify malicious nodes and detect their misbehavior. As a result, a malicious node can obstruct the communications of potentially any node in the network, exactly because it is entitled, or, even, expected to participate in assisting the network operation.

In addition, freely roaming nodes join and leave *MANET* sub-domains independently, possibly frequently, and without notice, making it difficult in most cases to have a clear picture of the ad hoc network membership. In other words, there may be no ground for an *a priori* classification of a subset of nodes as trusted to support the network functionality. *Trust* may only be developed over time, while trust relationships among nodes may also change, when, for example, nodes in an ad hoc network dynamically become affiliated with administrative domains. This is in contrast to other mobile networking paradigms, such as Mobile IP or cellular telephony, where nodes continue to belong to their administrative domain, in spite of mobility. Consequently, security solutions with static configuration would not suffice, and the assumption that all nodes can be bootstrapped with the credentials of all, or substantial fraction of, other nodes would be unrealistic for a wide range of *MANET* instances.

From a slightly different point of view, it becomes apparent that nodes cannot be easily classified as 'internal' or 'external,' that is, nodes that belong to the network or not, or nodes expected to participate and be dedicated to supporting a certain network operation and those that are not. In other words, the absence of an infrastructure impedes the usual practice of establishing a line of defense, separating nodes into trusted and non-trusted. As a result, attacks cannot be classified as internal or external either, especially at the network layer. Of course, such a distinction could be made at the application layer, where access to a service, or participation to its collaborative support, may be allowed only to authorized nodes. In the latter example, an attack from a compromised node within the group, that is, a group node under the control of an adversary would be considered as an internal one.

The absence of a central entity makes the detection of attacks a very difficult problem, since highly dynamic large networks cannot be easily monitored. Benign failures, such as transmission impairments, path breakages, and dropped packets, are naturally a fairly common occurrence in mobile ad hoc networks, and, consequently, malicious failures will be more difficult to distinguish. This will be especially true for adversaries that vary their attack pattern and misbehave intermittently against a set of their peers that also changes over time. As a result, short-lived observations will not allow detection of the adversaries. Moreover, abnormal situations may occur frequently, because nodes behave in a selfish manner and do not always assist the network functionality. It is noteworthy that such behavior may not be malicious, but only necessary when, for example, the node shuts its transceiver down in order to preserve its battery.

Most of the currently considered MANET protocols were not originally designed to deal with malicious behavior or other security threats. Thus they are easy to abuse. Compromised routes, i.e., routes that are not free of malicious nodes, may be repeatedly chosen with the "encouragement" provided by the malicious nodes themselves.[1] The result being that the pair of the communicating end-nodes will experience DoS, and they may have to rely on cycles of time-out and new route discovery to find operational routes, with successive query broadcasts imposing additional overhead. Or, even worse, the end nodes may be easily deceived for some period of time that the data flow is undisrupted, while no actual communication takes place. For example, the adversary may drop a route error message, "hiding" a route breakage, or it can corrupt both the data and their checksum, or forge network and transport layer acknowledgments.

Finally, mobile or nomadic hosts have limited computational capabilities, due to constraints stemming from the nature of the envisioned MANET applications. Expensive cryptographic operations, especially if they have to be performed for each packet and over each link of the traversed path, make such schemes implausible for the vast majority of mobile devices. Cryptographic algorithms may require computation delays ranging from one to several seconds [5, 11]. These delays, imposed, for example, by the generation or verification of a single digital signature, affect the data rate of secure communication. But, more importantly, mobile devices become ideal targets of DoS attacks due to their limited computational resources. An adversary would generate bogus packets, forcing the device to consume substantial portion of its resources. Worse even, a malicious node with valid credentials would generate control traffic, such as route queries, at a high rate not only to consume bandwidth, but also to impose cumbersome cryptographic operations on sizable portion of the network nodes.

4. Trust management

The use of cryptographic techniques is necessary for the provision of any type of security services, and mobile ad hoc networks are not an exception to this rule. The definition and the mechanisms for security policies, credentials, and trust relationships, i.e., the components of what is collectively identified as trust management, are a prerequisite for any security scheme. A large number of solutions have been presented in the literature for

[1] For instance, by the malicious nodes claiming that they possess an inexpensive (short) route to the destination

distributed systems, but they cannot be readily transplanted into the MANET context, since they rely on the existence of network hierarchy and on the existence of a central entity. In fact, MANET lacks exactly these two features. Envisioned applications for the ad hoc networking environment may require a completely different notion of establishing a trust relationship, while the network operation may impose additional obstacles to the effective implementation of such solutions.

For small-scale networks, of the size of a personal or home network, trust can be established in a truly ad hoc manner, with relationships being static and sporadically reconfigured manually. In such an environment, the owner of a number of devices or appliances can *imprint* them, that is, distribute their credentials and a set of rules that determine the allowed interaction with and between devices [24]. The proposed security policy follows a master-slave model, with the master device being responsible for reconfiguring slave devices, issuing commands or retrieving data. The return to the initial state can be done only by the master device, or by some trusted *key escrow* service.

This model naturally lends itself to represent personal area networking, in particular network instances such as Bluetooth [4], in the sense that within a Piconet the interactions between nodes can be determined by the security policy. The model can be extended by allowing partial control or access rights to be delegated, so that the secure interaction of devices becomes more flexible [25]. However, if the control over a device can be delegated, the new master should be prevented from eradicating prior associations and assuming full control of the node.

A more flexible configuration, independent of initial bindings, can be useful when a group of people wish to form a collaborative computing environment [9]. In such a scenario, the problem of establishing a trust relationship can be solved by a secure *key agreement*, so that any two or more devices are able to communicate securely. The mutual trust among users allows them to share or establish a password using an offline secure channel, and then execute a password-based authenticated key exchange over the insecure wireless medium. Schemes that derive a shared symmetric key can use a multi-party version of the password authenticated Diffie-Hellman key-exchange algorithm [3]. For instance, after an initial ordering of the nodes and a leader election, in each round, nodes choose a partner and do a two-party exchange, being able to proceed independently in an asynchronous manner, under the assumption that the adversary cannot remove or modify messages [1].

The human judgment and intervention can greatly facilitate the establishment of spontaneous connectivity among devices. Users can select a shared password or manually configure the security bindings between devices, as seen above. Furthermore, they could assess subjectively the ‘security’ of their physical and networking environment and then proceed accordingly. However, human assistance may be impossible for the envisioned MANET environment with nodes acting as mobile routers, although the distinction between an end device and a router may be only logical, with nodes assuming both roles. Frequently, the sole requirement for two transiently associated devices will be to mutually assist each other in the provision of basic networking services, such as route discovery and data forwarding. This could be so since mobile nodes will not necessarily pursue collectively a common goal. As a result, the users of the devices may have no means to establish a trust relationship in the absence of a prior context.

However, there is no reason to believe that a more general trust model would not be required in the MANET context. For instance, a node joining a domain may have to present its credentials in order to access an available service, and at the same time authenticate the service itself. Similarly, two network nodes may wish to employ a secure mode of multihop communication and verify each other’s identity. Clearly, support for such types of secure interaction, either at the network or at the application layer, will be needed.

A public key cryptosystem can be a solution, with each node bound to a pair of keys, one publicly known and one private. However, the deployment of a *public key infrastructure* (PKI) requires the existence of a *certification authority* (CA), a trusted third party responsible for certifying the binding between nodes and public keys. The use of a single point of service for key management can be a problem in the MANET context, especially because such a service should always remain available. It is possible that network partitions or congested links close to the CA server, although they may be transient, cause significant delays in getting a response. Moreover, in the presence of adversaries, access to the CA may be obstructed, or the resources of the CA node may be exhausted by a DoS attack. One approach is not to rely on a CA and thus abolish all the advantages of such a facility. Another approach is to instantiate the CA in a way that answers the particular challenges of the MANET environment.

The former approach can be based on the bootstrapping of all network nodes with the credentials of every other node. However, such an assumption will dramatically narrow the scope of ad hoc networking, since it can be applied only to short-lived mission-oriented and thus closed networks. An additional limitation may stem from the need to ensure a sufficient level of security, which implies that certificates should be refreshed from time to time, requiring, again, the presence of a CA.

Alternatively, it has been suggested that users certify the public keys of other users. One such scheme proposes that any group of K nodes may provide a certificate to a requesting node. Such a node broadcasts the request to its one-hop neighborhood, each neighbor provides a partial certificate, and if sufficient, that is, K such certificates are collected, the node acquires the complete certificate [14]. Another scheme proposes that each node selects a number of certificates to store, so that, when a node wants the public key of one of its peers, the two certificate repositories are merged, and if a chain of certificates is discovered, the public key is obtained [13].

An approach that can be applicable in a general setting and can effectively instantiate a key management facility has been proposed in [29]. The solution of a public key infrastructure is adapted to meet the requirements of the MANET environment, by providing increased availability and fault-tolerance. The certification authority (CA) is equipped with a private/public key pair. All network nodes know the public key of the CA, and trust all certificates signed by the CA's private key. Nodes that wish to establish secure communication with a destination, query the CA and retrieve the required certificate, thus being able to authenticate the other end, and establish a secret shared key for improved efficiency. Similarly, nodes can request an update from the CA, that is, change their own public key and acquire a certificate for the new key.

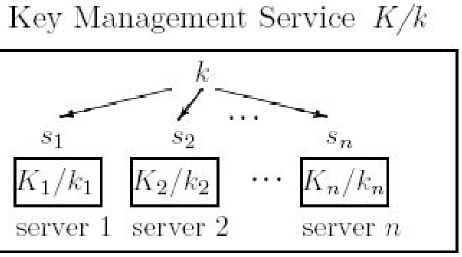

Figure 1. The configuration of a key management service: the key management service consists of *n* servers. The service, as a whole, has a public/private key pair *K/k*. The public key *K* is known to all nodes in the network, whereas the private key *k* is divided into *n* shares *s*1, *s*2, . . . , *sn*, one share for each server. Each server *i* also has a public/private key pair *Ki/ki* and knows the public keys of all nodes.

The CA is instantiated by a set of nodes (servers), as shown in Figure 1, for enhanced *availability*. However, this is not done through naïve replication, which would increase the vulnerability of the system, since the compromise of a single replica would be sufficient for the adversary to control the CA. Instead, the trust is distributed among a set of nodes, which share the key management responsibility. In particular, each of the *n* servers has its own pair of public/private key, and they collectively share the ability to sign certificates. This is achieved with the use of threshold cryptography, which allows any *t+1* out of *n* parties to perform a cryptographic operation, while *t* parties cannot do so. To accomplish this, the private key of the service, as a whole, is divided into *n* shares, with each of the servers holding one share. When a signature has to be computed, each server uses its share and generates a partial signature. All partial signatures are submitted to a *combiner*, a server with the special role to generate the certificate signature out of the collected partial signatures, as shown in the example of Figure 2. This is possible only with at least *t+1* valid partial signatures.

The application of threshold cryptography provides protection from compromised servers, since more than *t* servers have to be compromised before it assumes control of the service. If less than *t+1* servers are under the control of an adversary, the operation of the CA can continue efficiently, since purposefully invalid partial signatures, 'contributed' by rogue servers, will be detected. Moreover, the service provides the assurance that the adversary will not be able to compromise enough servers over a long period of time. This is done with the help of *share refreshing*, a technique that allows the servers to calculate new shares from the old ones without disclosing the private key of the service. The new shares are independent from the old ones and cannot be combined in an attempt to recover the private key of the CA. As a result, to compromise the system, all *t+1* shares have to be compromised within one refresh period, which can be chosen appropriately short in order to decrease vulnerability. The vulnerability can be decreased even further, when a quorum of correct servers detects compromised or unavailable servers and *re-configures* the service, that is, generates and distributes a new set of *n'* shares, *t'+1* of which need be combined now to calculate a valid signature. It is noteworthy that the public/private key pair of the service is not affected by share refreshing and re-configuration operations, which are transparent to all clients.

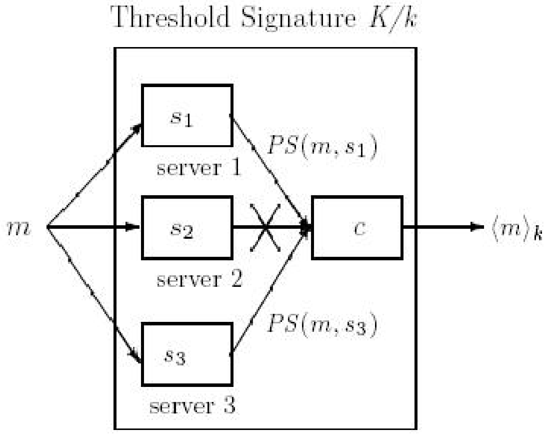

Figure 2. Threshold signature: given a service consisting of 3 servers. Let *K/k* be the public/private key pair of the service. Using a (3, 2) threshold cryptography scheme, each server *i* gets a share *si* of the private key *k*. For a message *m*, server *i* can generate a partial signature *PS*(*m*, *si*) using its share *si*. Correct servers 1 and 3 both generate partial signatures and forward the signatures to a combiner *c*. Even though server 2 fails to submit a partial signature, *c* is able to generate the signature $<m>_k$ of *m* signed by service private key *k*.

The threshold cryptography key management scheme can be adapted further by selecting different configurations of the key management service for different network instances. For example, the numbers of servers can be selected according to the size or the rate of changes of the network; for a large number of nodes within a large coverage area, the number of servers should also be large, so that the responsiveness of the service can be high. Nodes will tend to interact with the closest server, which can be only a few hops away, or with the server that responds with the least delay. Another possibility is to alternate among the servers within easy reach of the client, something that can happen naturally in a dynamically changing topology. This way, the load from queries and updates will be balanced among different servers, and the chances of congestion near one of the servers will be reduced. At the same time, the storage requirements can be traded off for inter-server communication, by storing at each server a fraction of the entire database.

Additionally, the efficient operation of the CA can be enhanced, when it is combined with secure route discovery and data forwarding protocols, which, in fact, can emulate the assumption of reliable links between servers in [29] even in the presence of adversaries. In particular, two of the protocols that will be discussed below, SRP and SMT, lend themselves naturally to this model. Any two servers[2] can discover and maintain routes to each other, and forward service-related traffic, regardless of whether intermediate nodes are trusted or not.

5. Secure Routing

The secure operation of the MANET routing protocol is of central importance, because of the absence of a fixed infrastructure. Instead, nodes are transiently associated and cooperate with virtually any node that could potentially disrupt the route discovery and data forwarding operations. In particular, the disruption of the route discovery may be an "effective" means to systematically obstruct the flow of data. Adversaries can respond with stale or corrupted route replies, or broadcast forged control packets in order to obstruct the propagation of legitimate queries and routing updates.

However, the usual practice for securing the Internet routing protocols [19] cannot be applied in the MANET context. The schemes proposed to secure Internet routing rely mainly on the existence of a line of defense, separating the fixed routing infrastructure from all other network entities. This is achieved by distributing a set of public keys/certificates, which signify the authority of the router to act within the limits of the employed protocol (e.g., advertise certain routes), and allow all routing data exchanges to be authenticated, not repudiated and protected from tampering, However, such approaches cannot combat a single malicious router disseminating incorrect topological information. More importantly, they are not applicable in the MANET context, because of impediments such as the absence of a fixed infrastructure and a central entity.

Although the appropriate design could provide increased assurances of the availability of an online certification authority (CA), the use of digital signatures and the hop-by-hop validation of control traffic may not be practical. First, mobile nodes lack sufficient computational power, as discussed above, and second, the interaction with the CA could become a limiting factor. In order to verify the correctness of the discovered routes, a node will have to acquire and validate the credentials of the responding nodes. Clearly, at least one route to the server has to be discovered before the node can contact the node instantiating the CA server. But the problem is that, in the presence of adversaries, forged replies would still require the server's response to be validated. A primary limitation arises from the frequently changing topology and network membership, which would incur frequent queries addressed to the CA. In addition, congested links close to the server, although they may be transient or intermittent, could result in significant delays or even total failure to provide the certification services. Even relatively small delays may render the validation process obsolete.

The protection of the route discovery process has been regarded as an additional Quality-of-Service (*QoS*) issue [28], by choosing routes that satisfy certain quantifiable security criteria. In particular, nodes in a *MANET* subnet are classified into different trust and privilege levels. A node initiating a route discovery sets the sought 'security' for the route, that is, the required minimum trust level for nodes participating in the query/reply propagation. Nodes at each trust level share symmetric encryption and decryption keys. Intermediate nodes of different levels that cannot determine whether the required *QoS* parameter can be satisfied or decrypt in-transit routing packets drop them. This scheme provides protection (e.g., integrity) of the routing protocol traffic against adversaries outside a specific trust level.

An extension of the *Ad Hoc On-demand Distance Vector* (*AODV*) [20] routing protocol has been proposed [10] in order to protect the routing protocol messages. The *Secure-AODV* scheme assumes that each node have the certified public keys of all network nodes, since intermediate nodes validate all in-transit routing packets. The basic idea is that the originator of a control message appends an *RSA signature* [23] and the last element of a *hash chain* [15], i.e., the result of *n* consecutive hash calculations of a random number. As the message traverses the network, intermediate nodes cryptographically validate the signature and the hash value, generate the *k-th* element of the hash chain, with *k* being the number of traversed hops, and place it in the packet. The route replies are provided either by the destination or by intermediate nodes that have an active route to the sought destination, with the latter mode of operation enabled by a different type of control packets.

A second proposal to secure AODV makes use of public key cryptography as well and operates in two stages, an end-to-end authentication, and an optional secure shortest path discovery [7]. First, a signed route request propagates to the sought destination, which returns a signed response to the querying node. At each hop, for either direction, the receiving node validates the received control packet and forwards it after signing it. At the second stage, a 'shortest path confirmation' packet is sent towards the destination, while now intermediate nodes sign the message in an onion-like manner in order to disallow changes of the path length.

5.1. The Secure Routing Protocol

The Secure Routing Protocol (SRP) [17] for mobile ad hoc networks provides correct end-to-end routing

[2] Any two servers of the key management service have a mutual security binding

information over an unknown, frequently changing network, in the presence of malicious nodes. It is assumed that any two nodes that wish to employ SRP have a Security Association (SA) instantiated by a symmetric shared secret key. Communication takes place over a broadcast medium, and it is assumed that malicious nodes, which may concurrently corrupt the route discovery, cannot collude during a single route discovery. Moreover, we assume that nodes have a single data link interface, with a one-to-one correspondence between data link and IP addresses. Under these assumptions, the protocol is proven robust.

SRP provides one or more route replies, whose correctness is verified by the route "geometry" itself, while compromised and invalid routing information is discarded. The route request packets verifiably propagate to the destination and route replies are returned to S strictly over the reversed route, as accumulated in the route request packet. In order to guarantee this crucially important functionality, SRP employs explicit interaction with the network layer; i.e., the IP-related functionality. Moreover, a number of novel features allow SRP to safeguard the route discovery operation, as explained below.

5.1.1. The Neighbor Lookup Protocol

An integral part of SRP, the Neighbor Lookup Protocol (NLP), is responsible for maintaining a valid mapping of Medium Access Control and IP layer addresses of the node's neighbors. It also detects discrepancies, such as the use of multiple IP addresses by a single data-link interface, or the use of a node's own Medium Access Control address by an adversary within the node's transmission range. To cope with a DoS attack, NLP also measures the rate at which SRP packets are received from each neighbor, primarily by differentiating the traffic based on its Medium Access Control address.

The basic idea is to retain the 48-bit hardware source address for each received (overheard) frame. This requires a simple modification of the device driver [27], so that the data link address is "passed up" with each packet. It is noteworthy that this approach leads to a reduction in the use of the neighbor discovery and query/reply mechanisms for medium access control address resolution.

Potential misbehavior incidents are detected and logged, explicit notification is provided to SRP so that the traffic from the corresponding node is discarded, and the inconsistencies are resolved. For example, NLP will detect the use of the same IP addresses by two different data-link interfaces within its transmission range, or the spoofing of a node's IP and Medium Access Control address by an adversary within the transmission range of the victim. Each notification is used by SRP to discard the corresponding transmission originating from the node suspected to be misbehaving.

NLP does not use cryptographic validation and cannot stop malicious nodes from attempting to spoof any address; nevertheless, it can detect such attempts, and can reduce their impact. In the worst case, at any time, an attacker will be able to spoof the address of a single node, but only as long as the victim is not within its transmission range. This is sufficient for SRP to provide correct connectivity information, since it disallows an adversary from presenting itself as more than one node within a single query/response phase. If this were attempted, an inconsistency would be detected, as discussed above, although its resolution might temporarily block a legitimate node as well. In all other cases, the spoofing of an address cannot affect the operation of SRP, which provides correct connectivity information without attempting to verify the 'identity' of nodes other than the destination. In other words, it is not of interest whether a relay node indeed presented itself with its 'actual' IP address, but whether the node participated in the discovery of the route, which is correct, thanks to SRP.[3]

5.1.2. The Basic Secure Route Discovery Procedure

The querying node maintains a Query Sequence number, $Q_{seq,}$ for each destination it securely communicates with. The monotonically increasing sequence number allows the destination to detect outdated route requests. At the same time, route requests are assigned a pseudorandom Query Identifier, Q_{ID}, which is used by intermediate nodes. Q_{ID} is statistically indistinguishable from a random number, and thus unpredictable by an adversary with limited computational power. As a result, broadcasted fabricated requests will fail to cause subsequent legitimate queries to be dropped.

Both Q_{ID} and Q_{seq} are placed in the SRP header, along with a Message Authentication Code (MAC) that covers the shared key, $K_{S,T}$, and the protocol header. Fields that are updated as the packet propagates towards the destination, such as the accumulated addresses, are excluded from the MAC calculation.

Nodes compare the last entry in the accumulated route to the IP datagram source address, which belongs to the neighboring node that relayed the request. If there is a mismatch, or NLP provides a notification that the relaying neighbor violated one of the enforced policies, the query is dropped. Otherwise the Q_{ID} and the source and destination addresses are placed in the query table, so that previously seen queries are discarded. "Fresh" route requests are re-broadcasted, with intermediate nodes inserting their IP address in the request packet.

The destination validates the integrity and freshness of queries originating from nodes it is securely associated with. It generates a number of replies that does not exceed the number of its neighbors, so that a malicious neighbor does not control more than one route. The reversed accumulated route serves as the source route of the reply packet, which is identified by Q_{ID} and Q_{seq}. The appended MAC covers the SRP header, including the source route. This way the source can be provided with evidence that the request had reached the destination and, in conjunction with

[3] The special case of using the address of a node already in the path is equivalent to any other malicious alteration of the control traffic, which the adversary could do in the first place. Of course, such a duplicate address will cause a loop and the route to be readily discarded.

the source route, that the reply was indeed returned along the reverse of the discovered route.

As the reply propagates along the reverse route, each intermediate node simply checks if the source address of the route reply datagram is the same as the one of its downstream node, as determined by the route reply; if not, the reply is discarded. Ultimately, the source validates the reply, by first checking whether it corresponds to a pending query. Then, it is sufficient to validate the MAC, since the IP source-route already provides the (reversed) route itself.

5.1.3. The Priority-Based Query Handling

In order to guarantee the responsiveness of the routing protocol, nodes maintain a priority ranking of their neighbors according to the rate of queries observed by NLP. The highest priority is assigned to the nodes generating (or relaying) requests with the lowest rate and vice versa. Quanta are allocated proportionally to the priorities and not serviced low-priority queries are eventually discarded. Within each class, queries are serviced in a round-robin manner. On the one hand, selfish or malicious nodes that broadcast requests at a very high rate are throttled back, first by their immediate neighbors and then by nodes farther from the source of potential misbehavior. On the other hand, non-malicious queries, that is, queries originating from benign nodes that regulate in a non-selfish manner the rate of query generation, will be affected only for a period equal to the time it takes to update the priority (weight) assigned to a misbehaving neighbor. In the mean time, the round robin servicing of requests provides the assurance that benign requests will be relayed even amidst a “storm” of malicious or extraneous requests.

5.1.4. The Route Maintenance Procedure

The route-error packets are source-routed to either of the two communicating ends along the prefix of the route that is being reported as broken. The intermediate upstream nodes, with respect to the point of breakage, check if the source address of the route error datagram is the same as the one of their downstream node as reported in the broken route. Then, if there is no notification from NLP that the relaying neighbor violated one of the enforced policies, they relay the packet towards the source. In this case, NLP prevents an adversary that does not belong to, but lies at a one-hop distance from the route, to generate an error message, since an inconsistency with the addresses already used (during the route discovery) by the actual downstream neighbor will be detected.

The notified source compares the source-route of the error message to the prefix of the corresponding active route. This way, it verifies that the provided route error message refers to the actual route, and that it is not generated by a node that is not part of the route. The correctness of the feedback (i.e., whether it reports an actual failure to forward a packet) cannot be verified though. As a result, a malicious node lying on a route can mislead the source by corrupting error messages generated by another node, or by masking a dropped packet as a link failure. However, it can harm only the route it belongs to, something that was possible in the first place, if it simply dropped or corrupted the data packets.

5.1.5. The SRP Extension

The basic operation of SRP can be extended in order to allow for nodes, other than the destination, to provide route replies. This would be possible only under additional trust assumptions, when, for example, nodes sharing a common objective belong to the same group and mutually trust all the group members. In particular, this could be instantiated by all group members sharing a secret key.

Under this assumption, a querying node appends to each query an additional MAC calculated with the group key, which we call Intermediate Node Reply Token (INRT). The functionality of SRP remains as described above, with the following addition: each group member maintains the latest query identifier seen from each of its peers, and can thus validate both the freshness and origin authenticity of queries generated from other group nodes.

If a node other than the sought destination receives such a valid query, then, it can respond to the request, if it has knowledge of a route to the destination in question. However, the correctness of such a route is conditional upon the correctness of the information provided by the intermediate node, regarding the second portion of the route.

This functionality can be provided independently from and in parallel with the one relying solely on the end-to-end security associations. For example, it could be useful for frequent intra-group communication; any two members can benefit from the assistance of their trusted peers, which may already have useful routes.

6. Secure data forwarding

The frequent interaction with a CA and the frequent use of computationally expensive cryptographic tools are restrictive assumptions, especially true for secure data-forwarding schemes. Such protocols must also take into account the inherent limitations of the MANET paradigm, exploit its features, and incorporate widely accepted and evaluated techniques, in order to be efficient and effective. Moreover, a secure routing protocol is a prerequisite for an effective secure data-forwarding scheme. The above Secure Routing Protocol (SRP) for mobile ad hoc networks satisfies the above-stated goals.

However, SRP or any other underlying routing protocol cannot guarantee that the nodes along a correctly discovered route will indeed relay the data as expected. An adversary may misbehave in an intermittent manner, that is, provide correct routing information during the route discovery stage, and later forge or corrupt data packets during the data forwarding stage. This is exactly the function that is required by any secure data forwarding protocol; to secure the flow of data traffic in the presence of malicious nodes, after the routes between the source and the destination have been discovered.

One of the solutions targeting the MANET environment proposes two mechanisms that (i) detect misbehaving nodes and report such events and (ii) maintain a set of metrics reflecting the past behavior of other nodes [16]. This scheme has been proposed to alleviate the detrimental effects of packet dropping. Each node may choose the 'best' route, comprised of relatively well-behaved nodes; i.e., nodes that do not have history of avoiding forwarding packets along established routes. Among the assumptions in the above-mentioned work are a shared medium, bi-directional links, use of source routing (i.e., packets carry the entire route that becomes known to all intermediate nodes), and no colluding malicious nodes. Nodes operating in promiscuous mode overhear the transmissions of their successors and may verify whether the packet was forwarded to the downstream node and check the integrity of the forwarded packet. Upon detection of a misbehaving node, a report is generated and nodes update the rating of the reported misbehaving node. The ratings of nodes along a well-behaved route are periodically incremented, while reception of a misbehavior alert dramatically decreases the node rating. When a new route is required, the source node calculates a path metric equal to the average of the ratings of the nodes in each of the route replies, and selects the route with the highest metric.

A different approach is to provide incentive to nodes, so that they comply with protocol rules, i.e., properly relay user data. The concept of fictitious currency is introduced in [6], in order to endogenize the behavior of the assumed greedy nodes, which would forward packets in exchange for currency. Each intermediate node purchases from its predecessor the received data packet and sells it to its successor along the path to the destination. Eventually, the destination pays for the received packet.[4] This scheme assumes the existence of an overlaid geographic routing infrastructure and a Public Key Infrastructure (*PKI*). All nodes are pre-loaded with an amount of currency, have unique identifiers and are associated with a pair of private/public keys. Finally, the cryptographic operations related to the currency transfers are performed by a physically tamper-resistant module.

Another approach appropriate for MANET, which departs significantly from the two above-mentioned schemes, is presented below. Low-cost cryptography is used to protect the integrity and origin authenticity of exchanged data, without placing any overhead at intermediate nodes. Moreover, the feedback that determines the 'security' of the chosen paths originates only from trusted destinations, thus allowing "safe" inferences on the quality of the paths. Finally, the secure data-forwarding protocol retains the flexibility of an integral part of a *MANET* routing protocol, but at the same time it enhances significantly the reliability and fault tolerance of data transmissions.

6.1. Secure Message Transmission Protocol

The Secure Message Transmission *(SMT)* protocol [18] is a *network-layer* secure and fault tolerant data-forwarding scheme, tailored to the *MANET* characteristics. In short, SMT determines a set of diverse paths connecting the source and the destination, as shown in the example of Figure 3. Then, it introduces limited transmission redundancy across the paths, by dispersing a message into *N* pieces, so that successful reception of any *M*-out-of-*N* pieces allows the reconstruction of the original message at the destination. Each piece, transmitted over one path, is equipped with a cryptographic header that provides origin authentication, integrity, and replay protection. Upon reception of a number of pieces, the destination informs the source of which pieces, and thus routes, were intact. In order to enhance the robustness of the feedback mechanism, the small-sized acknowledgments, also protected by a cryptographic header, are maximally dispersed, so that successful reception of one piece is sufficient. If less than *M* pieces were received, the source re-transmits the remaining pieces over the intact routes, or in general the ones deemed as more 'secure'. If too few pieces were acknowledged or too many messages remain outstanding, the protocol adapts its operation, by determining a different path set, re-encoding undelivered messages, and re-allocating pieces over the path set. Otherwise, it proceeds with subsequent message transmissions.

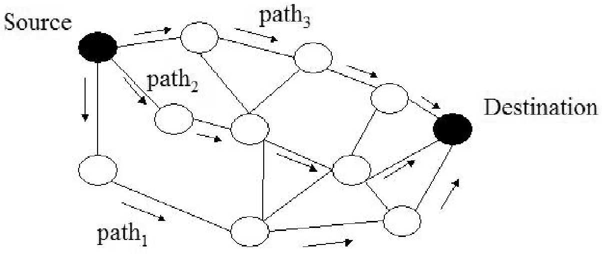

Figure 3. The Secure Message Transmission Protocol makes use of multiple diverse paths connecting the source and the destination. In particular, the Active Path Set (APS) contains paths that have not been detected as failed, either due to path breakage or because of the presence of an adversary on the path.

SMT exploits *MANET* features such as the topological redundancies, interoperates widely with accepted techniques such as on-demand route discovery and source routing, relies on a security association *only* between the source and the destination, and makes use of highly efficient symmetric-key cryptography. Moreover, the routing decisions are made by the querying node, based on the feedback that the destination and the underlying secure routing protocol provide. At the same time, no additional processing overhead is imposed on intermediate nodes, which do not perform any cryptographic operation but simply relay the message pieces. However, the use of multiple paths and the resultant greater number of nodes involved in the forwarding of a single message can be admittedly considered as the price to pay in order to achieve the sought fault tolerance.

On the one hand, *SMT*'s fault tolerance can be enhanced by the adaptation of parameters such as the number of

[4] An alternative implementation, with each packet carrying a purse of fictitious currency from which nodes remove their reward, is proposed as well.

paths, and the ratio of the numbers of transmitted to required pieces, termed as the redundancy or dispersion factor. On the other hand, in a low-risk environment with limited malicious failures, the same parameters can be adjusted, so that the imposed transmission overhead is reduced to a level close to that of a single-path scheme. An additional element that contributes to the flexibility of *SMT* is that different algorithms can be implemented for the selection of the path set based on different metrics and interpretations of the network feedback. *SMT* can yield *100%* successful message reception even in a highly adverse environment, when, for example, *20* percent of the network nodes are malicious, while keeping the message and computation overhead low.

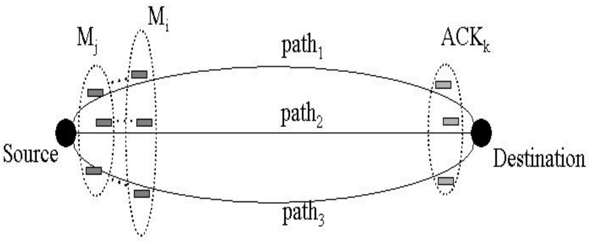

Figure 4. For an APS with three paths, the source can disperse each message into three pieces and transmit them across APS. The destination responds to each message M_k with an acknowledgement ACK_k notifying explicitly which pieces were received. This feedback allows the source to update fast the rating of the APS paths, and re-transmit lost pieces across the operational paths, if the message cannot be reconstructed at the destination.

The two communicating end nodes make use of the *Active Path Set (APS)*, comprising diverse paths that are not deemed failed. The sender invokes the underlying route discovery protocol, updates its network topology view, and then determines the *APS* for a specific destination. This model can be extended to multiple destinations, with one *APS* per destination. At the receiver's side, the *APS* is used for the transmission of the feedback, but if links are not bi-directional, the destination will have to determine its own "reverse" *APS*.

The dispersion of messages, which is performed by the information dispersal algorithm (IDA) [22], is coupled to the *APS* characteristics through an appropriate selection of the dispersion algorithm parameters. For example, in low connectivity conditions (small number of disjoint paths), the sender may increase the redundancy factor in order to provide increased assurance and possibly low transmission delay. The adaptation of the protocol is the result of the interplay among the following parameters: (i) K, the (sought) cardinality of APS, (ii) k, the *S,T-connectivity*, i.e., the maximum number of $S \rightarrow T$ node-disjoint paths from the source (S) to the destination (T), (iii) r, the redundancy factor of the IDA encoding, and (iv) x, the maximum number of malicious nodes. Clearly, the condition for successful reception is $x \leq \lceil K \times (1 - r^{-1}) \rceil$, which demonstrates the coupling among choices of parameters.

In particular, K can be determined as a function of r, so that the probability of successful transmission is maximized. In order to do so, the source starts by determining an *APS* of the k shortest, in number of hops, node disjoint paths. Then, let P_{GOAL} be the sought probability of successful reconstruction of a dispersed message. P_{GOAL} can be provided from the application layer and may correspond to the features of the supported application for example. Given P_{GOAL} and k, the node calculates the corresponding redundancy factor, r_{GOAL}, and disperses outgoing messages with the redundancy value closest to r_{GOAL}. Note that the source may achieve similar results with different values of M and N, and, more importantly, without knowledge of x or information on the percentage of adversaries in the network.

Once dispersed, the message pieces are temporarily buffered by the source, and they are transmitted across *APS*. If $N < k$, the node selects the N paths of the *APS* with the highest rating. If the receiver cannot reconstruct the message, the source re-transmits the pieces that were not received, according to the feedback provided by the destination. Message pieces are re-transmitted by *SMT* a *maximum* number of times, $Retry_{MAX}$, which is a protocol selectable parameter. If all re-transmissions fail, the message is discarded. This way, limited re-transmissions enhance the efficiency of *SMT*, by alleviating the overhead from re-transmitting the entire amount of data. On the other hand, *SMT* does not assume the role of a transport or application layer protocol; its goal is to promptly detect and tolerate failures, and thus adapt its operation to remain effective and efficient.

The transmission of data is continuous over the *APS*, with re-transmissions placed at the head of the queue, upon reception of the feedback. The continuous usage of the *APS* allows *SMT* to update fast its assessment on the quality of the paths. For each successful or failed piece, the rating of the corresponding path is increased or decreased respectively. When the rating drops below a threshold, the path is discarded. The path rating is also decreased slowly as time goes by, in order to reduce the chance of using a stale path. Moreover, the simultaneous routing over a number of paths, if not the entire *APS*, provides the opportunity for low-cost probing of the paths. In particular, the source can easily tolerate the loss of a piece that was transmitted over a low-rated path.

7. Discussion

The fast development of the mobile ad hoc networking technology over the last few years, with satisfactory solutions to a number of technical problems, supports the vision of widely deployed mobile ad hoc networks with self-organizing features and without the necessity of a pre-existing infrastructure. In this context, the secure operation of such infrastructure-less networks becomes a primary concern. Nevertheless, the provision of security services is dependent on the characteristics of the supported application and the networked environment, which may vary significantly. At one extreme, we can think of a library or an Internet café, which provide short-range wireless connectivity to patrons, without any access constraint other than the location of the mobile device. At the other extreme, a military or law enforcement unit can make use of powerful mobile devices, capable to perform expensive cryptographic operations. Such devices would communicate only with the rest of the other trusted devices.

Between these two ends of the spectrum, a multitude of MANET instances will provide different services, assume different modes of interaction and trust models, and admit solutions such as the ones surveyed above. However, it is probable that instead of a clear-cut distinction among network instances, devices and users with various security requirements will coexist in a large, open, frequently changing ubiquitous network.

In this context, an important related issue is the IP addressing scheme employed in the MANET environment. The common assumption that node credentials, e.g., certificates, are bound to *IP* addresses may need to be revisited, since one can imagine that roaming nodes will join *MANET* sub-domains and *IP* addresses will be assigned dynamically (e.g., *DHCP* [8], or IPv6 auto-configuration [26]) or even randomly (e.g., *Zero-Configuration* [12]).

A type of ad hoc network with particular requirements is a sensor network, which requires multihop communication throughout a network of hundreds or even thousands of nodes, with relatively infrequent topological changes. It is expected that a single organization will undertake the deployment and administration of these networks. Moreover, sensing devices have very limited computational capabilities, network transmission rates are relatively low, and communications are mostly data-driven. These requirements may affect in different ways the design of security measures for sensor networks, as demonstrated by the schemes proposed in the literature.

One of the proposals to secure sensor networks provides a protocol for data authentication, integrity and freshness, and a lightweight implementation of an authenticated broadcast protocol [21]. The scheme targets a restricted, infrastructure-oriented environment, with a trusted central entity instantiated by a set of base stations. Sensor nodes communicate only with a base station, which broadcasts messages towards the sensors. The base station and all nodes initially possess a symmetric encryption and authentication key, which secures the exchanged traffic, while, later, the base station periodically broadcasts the key that was used to authenticate transmissions during the last period.

An approach that has similarities but targets a more general setting proposes a key management scheme for sensor networks [2]. The focus is on resource-constrained large sensor networks, comprising nodes that are assumed tamper-resistant and equipped with a secret group key. Similarly to the previous scheme, the use of symmetric key cryptography is proposed as the only feasible, low-cost solution. However, frequent re-keying, that is, periodic re-generation of the single key that is used to encrypt all data transmitted by sensors, is proposed to protect it from possible compromise. In order to make this reconfiguration operation efficient, the sensors are organized into clusters with a two-hop diameter, while cluster heads are elected and form a backbone. Then, from a subset of the backbone, a randomly elected node generates the new key.

The simplified trust models of the sensor networks, which, nonetheless, lead to efficient solutions, may not necessarily be usable in other ad hoc networking instances. The circumstantial co-existence of disparate nodes, or the requirement of fine-grained trust relationships call for solutions that can adapt to specific context and support the corresponding application. However, although the requirements of the application are expected to dictate the characteristics of the required security mechanisms, some aspects of security, such as confidentiality, may not be different at all in the MANET context. Instead, the greatest challenge is to safeguard the basic network operation.

In particular, the securing of the network topology discovery and data forwarding is a prerequisite for the secure operation of mobile ad hoc networks in any adverse environment. Additionally, the protection of the functionality of the networking protocols will be in many cases orthogonal to the security requirements and the security services provided at the application layer. For example, a transaction can be secured when the two communicating end nodes execute a cryptographic protocol based on established mutual trust, with the adversary being practically unable to attack the protocol. But this does not imply that the nodes are secure against denial of service attacks; the adversary can still abuse the network protocols, and in fact, do it with little effort compared to the effort needed to compromise the cryptographic protocol.

The self-organizing networking infrastructure has to be protected against misbehaving nodes, with the use of low-cost cryptographic tools, under the least restrictive trust assumptions. Moreover, the overhead stemming from such security measures should be imposed mostly, if not entirely, on nodes that communicate in a secure manner and that directly benefit from these security measures. Furthermore, we believe that the salient MANET features and the unique operational requirements of these networks call for security mechanisms that are primarily present at, and closely interwoven with, the network-layer operation, in order to realize the full potential of this promising new technology.

REFERENCES

[1] N. Asokan, P. Ginzboorg. “Key Agreement in Ad Hoc Networks.” Computer Communications 23 (17): 1627-1637 Nov 1 2000.

[2] S. Basagni, K. Herrin, E. Rosti, D. Bruschi. “Secure Pebblenets.” 2nd MobiHoc, CA, Oct. 2001.

[3] S.M. Bellovin and M. Merritt. “Encrypted Key Exchange: Password-based protocols secure against dictionary attacks.” Proceedings of the IEEE Symposium on Security and Privacy, May 1992.

[4] Bluetooth Special Interest Group. “Specifications of the Bluetooth System.” http://www.bluetooth.com.

[5] M. Brown, D. Cheung, D. Hankerson, J. L. Hernadez, M. Kirkup, and A. Menezes. “PGP in Constrained

Wireless Devices." Proceedings of 9th USENIX Symposium, Denver, Colorado, August 2000.

[6] L. Buttyan and J.P. Hubaux. "Enforcing Service Availability in Mobile Ad Hoc WANs." 1st MobiHoc, BA Massachusetts, Aug. 2000.

[7] B. Dahill, B.N. Levine, E. Royer, C. Shields. "A Secure Routing Protocol for Ad Hoc Networks." Technical Report UM-CS-2001-037, EE&CS, Univ. of Michigan, August 2001.

[8] R. Droms. "Dynamic Host Configuration Protocol." IETF RFC 2131, Mar. 1997.

[9] L.M. Feeney, B. Ahlgren, A. Westerlund. "Spontaneous Networking: An Application-Oriented Approach to Ad Hoc Networking." IEEE Communications Magazine, vol. 39, No. 6, p. 176-181, Jun. 2001.

[10] M. Guerrero. "Secure AODV." Draft sent to the *manet@itd.nrl.navy.mil* mailing list.

[11] V. Gupta, S. Gupta. "Securing the Wireless Internet." IEEE Communications Magazine, p. 68-74, December 2001.

[12] M. Hattig, Editor. "Zero-conf IP Host Requirements." Draft-ietf-zeroconf-reqts-09.txt, IETF MANET Working Group, Aug. 31st, 2001.

[13] J.P. Hubaux, L. Buttyan, and S. Capkun. "The quest for security in mobile ad hoc networks." 2nd MobiHoc, CA, Oct. 2001.

[14] J. Kong, P. Zerfos, H. Luo, S. Lu and L. Zhang. "Providing Robust and Ubiquitous Security Support for Mobile Ad-Hoc Networks." *IEEE ICNP (International Conference on Network Protocols) 2001, Riverside, CA*, Nov. 2001.

[15] L. Lamport. "Password Authentication with Insecure Communication." Comm. of ACM, 24 (11), pp. 770-772, Nov. 1981.

[16] S. Marti, T.J. Giuli, K. Lai, M. Baker. "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks." 6th MobiCom, BA Massachusetts, Aug. 2000.

[17] P. Papadimitratos and Z.J. Haas. "Secure Routing for Mobile Ad Hoc Networks." SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002), San Antonio, TX, January 27-31, 2002.

[18] P. Papadimitratos and Z.J. Haas. "Secure Message Transmission in Mobile Ad Hoc Networks." Submitted for publication.

[19] P. Papadimitratos and Z.J. Haas. "Securing the Internet Routing Infrastructure." Submitted for publication.

[20] C.E. Perkins, E.M. Royer, S.R. Das. "Ad hoc On-Demand Distance Vector Routing". Draft-ietf-manet-aodv-08.txt, IETF MANET Working Group, June 1st, 2001.

[21] A. Perrig, R. Szewczyk, V. Wen, D. Culler, J.D. Tygar. "SPINS: Security Protocols for Sensor Networks." Proc. 7th Ann. Int'l Conf. Mobile Computing and Networks (Mobicom 2001), ACM Press, pp. 189-199, NY, 2001.

[22] M.O. Rabin. "Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance." Journal of ACM, Vol. 36, No. 2, pp. 335-348, April 1989.

[23] R. Rivest, A. Shamir, L. Adleman. "A method for obtaining Digital Signatures and Public Key Cryptosystems." Comm. of ACM, 21 (2), pp. 120-126, Feb. 1978.

[24] F. Stajano and R. Anderson. "The Resurrecting Duckling: Security Issues for Ad Hoc Wireless Networks." Security Protocols, 7th International Workshop, LNCS, Springer-Verlag, 1999.

[25] F. Stajano. "The Resurrecting Duckling – What next?" Security Protocols, 8th International Workshop, LNCS, Springer-Verlag, 2000.

[26] S. Thomson and T. Narten. "IPv6 Stateless Address Autoconfiguration." IETF RFC 1971, www.ietf.org.

[27] G.R. Wright, W. Stevens. "TCP/IP Illustrated, vol.2, the implementation." Addison-Wesley, Feb. 1997.

[28] S. Yi, P. Naldurg, R. Kravets. "Security-Aware Ad-Hoc Routing for Wireless Networks." UIUCDCS-R-2001-2241 Technical Report, Aug. 2001.

[29] L. Zhou and Z.J. Haas. "Securing Ad Hoc Networks." IEEE Network Magazine, vol. 13, no.6, November/ December 1999.