Ad Hoc Wireless Networking
X. Cheng, X. Huang and D.-Z. Du (Eds.) pp. – – –

Secure Communication in Adverse Mobile Ad Hoc Networks[1]

Panagiotis Papadimitratos
School of Electrical and Computer Engineering
Cornell University, Ithaca, NY 14853
E-mail: `papadp@ece.cornell.edu`

Zygmunt J. Haas
School of Electrical and Computer Engineering
Cornell University, Ithaca, NY 14853
E-mail: `haas@ece.cornell.edu`

Contents

[1]Reprinted with permission from "Handbook of Ad Hoc Wireless Networks," chapter 31, CRC Press 2003. Copyright Zygmunt J. Haas.

1 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 strongly depends 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. Moreover, 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 particular, practically any node in the open, collaborative MANET environment can abuse the network operation and disrupt or deny communication. In this paper we introduce our approach to this multifaceted and intriguing problem: a set of protocols that secure the fundamental networking operations of routing and data transmission. Moreover, we survey solutions that address the management of trust in ad hoc networks.

2 Introduction

Mobile ad hoc networks comprise freely roaming wireless nodes that cooperatively make up for the absence of fixed infrastructure, with the nodes themselves supporting the network functionality. Nodes form transient associations 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 the communicating parties. The MANET technology targets networks that can be rapidly deployed or formed in an arbitrary environment to enable communications or to serve, in some cases, 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 an especially 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 peer-to-peer node interaction opens MANET protocols to abuse. The MANET paradigm seeks to enable communication across networks whose topology and membership may change very frequently, based on the cooperative support of the network functionality. Malicious nodes can disrupt or even deny the communications of potentially any node within their ad hoc networking domain. This is so, exactly because each and every node is not only entitled, but is, in fact, required to assist the network operation.

With migrating nodes joining and leaving MANET domains and transient associations between nodes constantly established and torn down, it is particularly difficult to distinguish which nodes are trustworthy and supportive. First, the practically invisible or non-existent administrative boundaries encumber the a priori classification of a subset of nodes as trusted. Second, it is impractical, in such a volatile communication environment, to determine which nodes can be trusted based on the network interaction - the overhead and especially the delay to make such an inference would be prohibitive, with additional overhead and complexity imposed if such inferences were to propagate in the form of recommendations or accusations. In most cases, transiently associated nodes will assist each other with the provision of basic networking services, such as route discovery and data forwarding. As a result, the nodes, or, practically the users of the devices, may have no means to establish a trust relationship, since not all mobile nodes would necessarily pursue collectively a common mission.

In other words, in mobile ad hoc networks, the particular challenge is to safeguard the correct operation of the network layer protocols. Nodes may be designated as trusted or non-trusted at the application layer - for example, access to a service or participation to its collaborative support would be allowed only to nodes that present the necessary credentials. However, only closed, mission-oriented networks could satisfy such an assumption of full trust. Thus, the reliance on trusted nodes solely would drastically narrow the scope and limit the potential of ad hoc networking.

We first outline the primary goals of security enhancements for MANET and shed light on the commensurate challenges. In Section 5 we present the Secure Routing Protocol (SRP) and the Secure Link State Protocol (SLSP), which safeguard the discovery of routing information assisted by the Neighbor Lookup Protocol (NLP). The Secure Message Transmission (SMT) protocol, which enhances the security and robustness of the data transmission, follows in Section 6. Next, we discuss approaches for managing trust in the MANET environment in Section 7, briefly survey related work in Section 8 and conclude with a discussion.

3 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 [43], 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.

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 disable the routing protocol operation, by disrupting the operation of the route discovery procedure. Moreover, an 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 an in-transit message 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. Integrity viewed in the context of a specific 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, possibly 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, A can use this message to accuse B 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 security 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.

4 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 shared 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. 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 assist in 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 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; i.e., nodes that are 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 attackers 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. Incorrect routing information can be injected by malicious nodes that respond with or advertise inexistent or stale routes and links. In addition, 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.[2] 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

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

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 [7, 14]. Such 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.

5 Secure Routing

The secure operation of the MANET routing protocol is of central importance, primarily because of the absence of a fixed infrastructure. Attackers can "effectively" obstruct the flow of data by systematically disrupting the discovery of routing information and thus distorting or even dictating the topology knowledge of benign nodes.

Protocols proposed to secure Internet routing share goals with MANET secure routing protocols, seeking to safeguard the correct operation of the topology discovery. However, they cannot be readily transplanted into the MANET context, since they are designed to operate in a fundamentally different networking environment. They establish a line of defense, separating the fixed routing infrastructure from all other network entities. Routers are equipped with credentials (public keys, certificates) that signify the router's authority to act within the limits of the employed protocol (e.g., advertise certain routes), and allow all routing traffic to be authenticated, not repudiated and protected from tampering [33]. Clearly, the volatility and the salient features of the MANET environment, as discussed in Section 4, impede such an approach.

Our design seeks to overcome such limitations and furthermore provide protocols applicable to a wide range of MANET instances. More specifically, we do not require that each node be able to present and validate credentials, such as public keys, for all other network nodes. In addition, the operation of our protocols does not rely on any assumption on the network membership, which may change frequently. At the same time, we do not make any assumption on the node mobility, node equipment (such as Global Positioning System (GPS)), and network size. Finally, our protocols do not rely on intrusion detection or monitoring techniques and do not assume any regularity or patterns of malicious behavior in order to identify and isolate adversarial nodes. Instead, they are capable of operating in the presence of adversaries that actively disrupt the discovery and distribution of routing information.

The scope of the proposed protocols is broadened by their ability to operate under the least restrictive assumptions. Nevertheless, differing operational conditions may call for different topology discovery approaches. First, we propose a reactive routing protocol, the Secure Routing Protocol (SRP). Additionally, a proactive protocol, the Secure Link State Protocol (SLSP), is presented. Finally, we introduce the Neighbor Lookup Protocol (NLP), which can complement and strengthen both SRP and SLSP.

NLP addresses the correctness of communication with the node's immediate neighbors, that is nodes within its transceiver range. In essence, NLP is responsible for countering attackers that exploit the wireless communication over the shared medium. Its purpose is to identify traffic that violates specific criteria, and notify the conceptually overlying routing protocol. As a result, the routing protocol discards such traffic.

The goal of both SRP and SLSP is to safeguard the acquisition of topological information by countering attacks that disrupt or exploit the route discovery operation to deny communication. In particular, they provide correct, i.e., factual, up-to-date and authentic topology information, and they robust against individual Byzantine adversaries.

5.1 The Neighbor Lookup Protocol

The Neighbor Lookup Protocol (NLP), which can be an integral part of the routing protocol, is responsible for the following tasks: (*i*) it maintains a mapping of MAC and IP layer addresses of the node's neighbors, (*ii*) it identifies potential discrepancies, such as the use of multiple IP addresses by a single data-link interface, and (*iii*) it measuring the rates at which control packets are received from each neighbor, by differentiating the traffic primarily based on MAC addresses. The measured rates of incoming control packets are provided to the routing protocol. This way, control traffic originating from nodes that selfishly or maliciously attempt to overload the network can be discarded.

Basically, NLP extracts and retains the 48-bit hardware source address for each received (overheard) frame, along with the encapsulated IP address. This requires a simple modification of the device driver [50], so that the data link address is "passed up" to the routing protocol along with each packet. With nodes operating in promiscuous mode, the extraction of such pairs of addresses from all overheard packets leads to a significant reduction in the use of the neighbor discovery and query/reply mechanisms for medium access control address resolution.

Each node updates its neighbor table by retaining both, the data-link and the network interface addresses. The mappings between the two addresses are retained in the table as long as transmissions from the corresponding neighboring nodes are overheard; a lost neighbor timeout period [3] is associated with each table entry.

NLP issues a notification to the routing protocol, according to the content of a received packet, in the event that: (*i*) a neighbor used an IP address different from the address currently recorded in the neighbor table, (*ii*) two neighbors used the same IP address (that is, a packet appears to originate from a node that may have "spoofed" an IP address), (*iii*) a node uses the same medium access control address as the detecting node (in that case, the data link address may be "spoofed"). Upon reception of the notification, the routing protocol discards the packet bearing the address that violated the aforementioned policies.

Each notification is used by the routing protocol to discard the corresponding transmission originating from the node suspected to be misbehaving. This is the primary goal of NLP, which can be secondarily used to identify the attacker itself. The unambiguous identification of the attacker and the later use of such information to protect the network is a particularly difficult task for MANET. Its feasibility depends on the underlying trust between nodes, among other factors. On the other hand, the prompt detection of traffic that can harm the network operation individually at each node avoids such limitations and can be readily beneficial.

[3] The lost neighbor timeout should be longer than the timeout periods associated with the flushing of routing information (link state, routing table entries), related to the particular neighbor.

The operation of NLP can be adjusted to the requirements of the routing protocol, by employing cryptography or not. This depends on the requirements and operation of the routing protocol. For example, in the context of SRP that assumes solely an end-to-end security association, NLP would not be expected to utilize cryptography. Nevertheless, the cryptographic operation of NLP can be orthogonal to end-to-end security associations. Such an example is SLSP, which requires a portion of its traffic to be cryptographically validated by immediate neighbors and thus strengthen NLP.

5.2 The Secure Routing Protocol

The Secure Routing Protocol (SRP) [31] for mobile ad hoc networks provides correct end-to-end routing information over an unknown, frequently changing network, in the presence of malicious nodes. We require 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, with no need for cryptographic operations on control traffic at intermediate nodes, two factors that render the scheme efficient and scalable. SRP places the overhead on the end nodes, an appropriate choice for a highly decentralized environment, and contributes to the robustness and flexibility of the scheme.

The novelty of SRP lies in the verification of the correctness of the discovered route(s) from the route "geometry" itself. False or corrupted control traffic is discarded in parts by the end nodes, thanks to the end-to-end security association, and in parts by the intermediate benign nodes, without cryptographic processing in the latter case. Basically, route requests propagate verifiably to the sought trusted destination and route replies are returned strictly over the reversed route, as accumulated in the route request packet. Moreover, intermediate nodes do not relay route replies unless their downstream node had previously relayed the corresponding query. In order to guarantee this crucially important functionality, the interaction of the protocol with the IP-related functionality is explicitly defined. An intact reply implies that (*i*) the reported path is the one placed in the reply packet by the destination, and (*ii*) the corresponding connectivity information is correct, since the reply was relayed along the reverse of the discovered route and consists of all nodes that participated in both phases of the route discovery.

The securing of the route discovery deprives the adversarial nodes of an "effective" means to systematically disrupt the communications of their peers. Despite our minimal trust assumptions, attackers cannot impersonate the destination and redirect data traffic, cannot respond with stale or corrupted routing information, are prevented from broadcasting forged control packets to obstruct the later propagation of legitimate queries, and are unable to influence the topological knowledge of benign nodes. To that extent, SRP provides very strong assurances on the correctness of the link-level connectivity information as well. It precludes adversarial nodes from controlling multiple potential routes per source-destination pair, and from forming "dumb" relays, that is, from not placing themselves in a route whose discovery they assisted.

The security features of SRP do not undermine its efficiency, that is, the ability of nodes to quickly respond to topological changes and discover correct routes. In addition, the protocol retains its ability to operate when under attack, with adversaries actively disrupting the route discovery. Furthermore, its low cryptographic processing overhead renders SRP applicable for nodes with very limited computational resources. Finally, the reliance on the basic and widely accepted reactive route discovery mechanism (broadcasted route query packets traverse the network as the relaying intermediate nodes append their identifier (IP address)) allows SRP to naturally extend a number of existing protocols. In particular, the IERP [16] of the Zone Routing Protocol (ZRP) [17] framework, the Dynamic Source Routing (DSR) [20], and ABR [48] are protocols that can incorporate the features of SRP with minimal or limited modifications.

5.2.1 The Generation of Route Requests

A source node S maintains a query sequence number Q_{SEQ} for each destination it securely communicates with. The 32-bit Q_{SEQ} increases monotonically, for each request generated by S, and allows T to detect outdated route requests. The sequence number is initialized at the establishment of the SA and although it is not allowed to wrap around, it provides approximately a space of four billion query requests per destination. If the entire space is used, a new security association has to be established.

For each outgoing $ROUTE\ REQUEST$, S generates a 32-bit random Query Identifier Q_{ID}, which is used by intermediate nodes as a means to identify the request. Q_{ID} is the output of a secure pseudo-random number generator [28]; its output is statistically indistinguishable from a truly random one and is unpredictable by an adversary with limited computational power. Since intermediate nodes have limited memory of past queries, uniqueness and randomness can be efficiently achieved, by using a one-way function (e.g., SHA-1 [42] or MD5 [41]) and a small random seed as input.

This renders the prediction of the query identifiers practically impossible, and combats the following attack: malicious nodes simply broadcast fabricated requests only to cause subsequent legitimate queries to be dropped.

Along with Q_{ID} and Q_{SEQ}, the *ROUTE REQUEST* header includes a Message Authentication Code (MAC). The MAC is a 96-bit long field, generated by a keyed hash algorithm [9], which calculates the truncated output of a one-way or hash function. The one-way function input is the entire IP header, the route request packet and most importantly, $K_{S,T}$ the key shared by the two communicating nodes. However, the packet fields where the addresses of the intermediate nodes are accumulated as the packet propagates towards the destination are excluded.

The querying node regulates the rate of queries it generates in order to avoid overloading the network. It can also indicate the number of route replies per query the destination should return by including a field protected the request MAC. This value may be increased in case of a failed route discovery or when the node needs to enrich its view of the network topology.

5.2.2 The Processing of Route Requests

Nodes parse received *ROUTE REQUEST* packets to determine whether an SRP header is present. If the SRP header is not present the packet is dropped. Intermediate nodes extract the Q_{ID} value to determine if they have already relayed a packet corresponding to the same request. If not, they compare the last entry in the accumulated route to the IP datagram source address, which belongs to the neighboring node that relayed the request. Request packets are dropped in the case of a mismatch or if an NLP provides a notification that the relaying neighbor violated one of the enforced policies. Otherwise, the packet is relayed (re-broadcasted), with the intermediate node inserting its IP address.

The Q_{ID}, the source and the destination address field values are placed in the query table. In addition, intermediate nodes retain the IP addresses of their neighbors overheard forwarding (re-broadcasting) the query, in a *FORWARD LIST* associated with the query table entry. As it will be explained below, this information ensures that intermediate nodes cannot hide themselves from a discovered route.

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.

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 their 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.

When requests arrive at the sought destination, they are validated thanks to the security binding with the querying node. First, Q_{SEQ} is compared to $S_{MAX}(S)$, the latest (highest) query sequence number received from S, within the lifetime of the S-T SA. If $Q_{SEQ} < S_{MAX}(S)$, the request is discarded as outdated or replayed. If $Q_{SEQ} \geq S_{MAX}(S)$, T verifies the integrity and authenticity of the origin of the request packet. It generates a route reply, up to the point it has responded to the number of valid requests indicated by the query packets.

5.2.3 The Generation and Processing of Route Replies

The $ROUTE\ REPLY$ is identified by the Q_{SEQ} and Q_{ID} of the corresponding $ROUTE\ REQUEST$. The reverse of the route accumulated in the request packet is used as the source route of the reply packet. The destination calculates, using KS,T, and appends a MAC covering the entire SRP header, and the source route of the reply packet. The reply is routed strictly along the reverse of the discovered route. This way, the source is provided with evidence that the request had reached the destination and that the reply was indeed returned along the reverse of the discovered route.

As the reply propagates along the reverse route, each intermediate relaying node check whether the source address of the $ROUTE\ REPLY$ datagram is the same as the address of its downstream node, as reported in the reply. If not, or if and NLP notification has been received, the reply packet is discarded. Clearly, replies are discarded if the corresponding request is not previously received and relayed, that is, there is no entry in the node's query table.

Additionally, the reply packet is discarded if it originates from a node that is not listed in $FORWARD\ LIST$. This last control practically eliminates the possibility of a "dumb" or "Byzantine" relay, if a malicious node relayed both the route request and route reply while 'hiding' itself from the discovered route. This further strengthens the defense provided by NLP, which would issue a notification upon receiving the packet that the malicious node would relay attempting to mount the attack. If NLP is employed alone, a collision at the receiver (benign node) could prevent the detection of the first stage of the attack. On the other hand, such a collision could lead to discarding a benign reply, but it ensures that the Byzantine link attack cannot be mounted. The choice of either or both countermeasures is clearly a design decision, dependent on the envisioned networking environment.

Ultimately, the source validates the reply: it first checks whether it corresponds to a pending query. Then, it suffices to validate the MAC, and extract the route from the IP source route of the $ROUTE\ REPLY$, which already provides the (reversed) discovered route.

5.2.4 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 or feedback on the status of utilized routes. This may be possible if a subset of nodes share a common objective, belong to the same group G and mutually trust all the group members.

In that case, the mutual trust could be instantiated by all group members sharing a secret key K_G. In that case, a querying node can append to each query an additional MAC calculated with the group key K_G, 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.

Nodes other than the sought destination respond to a validated request, if they have knowledge of a route to the destination in question. The replies are generated as above, except for the MAC calculation that uses KG. 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. When the reply is generated by the destination, an additional $MAC(K_G, ROUTE\ REPLY)$ is appended, apart from the end-to-end $MAC(K_{S,T}, ROUTE\ REPLY)$. This would allow an intermediate node V that is part of the route and a member of the group G to utilize the discovered route suffix (i.e., the V to T part).

The INRT 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. Finally, the shared KG can be utilized for purposes that are beyond the discovery of routes. One example is the authentication of *ROUTE ERROR* messages, as explained below.

5.2.5 The Route Maintenance Procedure

Intermediate node that fails to deliver a data packet to the next hop generate a *ROUTE ERROR* packet which is strictly source-routed back to the source node S along the prefix of the route being reported as broken. The 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. If there is no NLP notification that the relaying neighbor violated one of the enforced policies, the packet is relayed towards the source.

The source node 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, this allows it to harm only the route it belongs to, something that was possible in the first place, if it simply dropped or corrupted in-transit data packets.

If the reporting intermediate node does not have a security association with the source node, *ROUTE ERROR* messages do not include a MAC. This allows an adversary that can spoof a data link address and lies within one hop of an end-to-end data flow (route) to inject a *ROUTE ERROR*. This would be possible if it impersonated a node that is part of the route. Although the NLP of the victim would issue a notification, the forged error packet would be in-transit towards the source.

Consequently, *ROUTE ERROR* messages can be used in the following cases: (i) an end-to-end secure mechanism is present and thus the source node can infer the status of the utilized route(s), and (ii) the intermediate issuing node has a security association with the source node. In case (i) holds but (ii) does not, route errors can be used in a complementary manner. For example, the Secure Message Transmission (SMT) protocol which we describe Section 6, can utilize unauthenticated *ROUTE ERROR* messages to update the 'rating' of the utilized route(s) only when the end-to-end secure feedback reports a failed transmission. In case (*ii*), an intermediate node, which is for example member of the same group as the source of the broken route, can use the group key to generate a MAC that covers the entire route error packet and its IP source route.

5.3 The Secure Link State Protocol

The Secure Link State Protocol (SLSP) for mobile ad hoc networks [34] is responsible for securing the discovery and distribution of link state information. The scope of SLSP may range from a secure neighborhood discovery to a network-wide secure link state protocol. SLSP nodes disseminate their link state updates and maintain topological information for the subset of network nodes within R hops, which is termed as their zone [38]. Nevertheless, SLSP is a self-contained link state discovery protocol, even though it draws from, and naturally fits within, the concept of hybrid routing.

Each node is equipped with a public/private key pair, namely E_V and D_V, and with a single network interface per node within a MANET domain.[4] Key certification can be provided in a number of ways, as we explain in detail in Section 7.

Nodes are identified by their IP addresses, which may be assigned by a variety of schemes, e.g., dynamically or even randomly [12, 15]. Although E_V does not need to be tied to the node's IP address, it could be beneficial to use IP addresses derived from the nodes' public keys [29]. Nodes are equipped with a one-way or hash function H and a public key cryptosystem (e.g., [40]).

To counter adversaries, SLSP protects link state update (*LSU*) packets from malicious alteration, as they propagate across the network. It disallows advertisements of non-existent, fabricated links, stops nodes from masquerading their peers, strengthens the robustness of neighbor discovery, and thwarts deliberate floods of control traffic that exhausts network and node resources.

To operate efficiently in the absence of a central key management, SLSP provides for each node to distribute its public key to nodes within its zone. Nodes periodically broadcast their certified key, so that the receiving nodes validate their subsequent link state updates. As the network topology

[4] To support operation with multiple interfaces, one key pair should be assigned to each interface.

changes, nodes learn the keys of nodes that move into their zone, thus keeping track of a relatively limited number of keys at every instance.

SLSP defines a secure neighbor discovery that binds each node V to its Medium Access Control (MAC) address and its IP address, and allows all other nodes within transmission range to identify V unambiguously, given that they already have E_V. Each node commits its Medium Access Control (MAC) address and its IP address, the $(MAC_V,\ IP_V)$ pair, to its neighbors by broadcasting signed hello messages. Receiving nodes validate the signature and retain the information; in the case of SUCV addresses [29] the confirmation for the IP address can be done in a memory-less manner.

5.3.1 The Link State Updates

Nodes advertise the state of their incident links by broadcasting periodically signed link state updates. SLSP restricts the propagation of the LSU packets to within the zone of their origin node. Receiving nodes validate the updates, suppress duplicates, and relay previously unseen updates that have not already propagated R hops. Link state information acquired from validated LSU packets is accepted only if both nodes incident on each link advertise the same state of the link.

Link state updates are identified by the IP address of their originator and a 32-bit sequence number, which provides an ample space of approximately four billion updates. To ensure that the LSU's propagate only within the zone of its origin, i.e., R hops away, the node selects a random number X and calculates a hash chain: $X_i = H^i(X)$, $i = 1, ..., R$, $H^0(X) = X$. It places X_R and X_1 in the *zone radius* and the *hops traversed* fields of the LSU header,[5] respectively, and sets TTL equal to $R-1$, with R placed in the R_{LSU} field. Finally, a signature is appended.

Receiving nodes check if they have the public key of the originating node, unless the key is attached to the LSU (see Section 5.3.2 below). For an LSU that has already travelled over i hops ($i = R - TTL$), if i is less than the radius of the originating node, the packet is not relayed unless $H^{R-i}(hops\ traversed)$ equals *zone radius*. Each relaying node sets *hops traversed* equal to $H(hops\ traversed)$, decrements TTL, and rebroadcasts the LSU.

The provided information is discarded after a $confirm_{LS}$ timeout, unless both nodes incident on a link report the same state. Finally, NLP notifications result in discarding an update relayed by a misbehaved node. The

[5]Hash chains have a wide range of applications; in the MANET context, they have been used to assist in hop count authentication [52].

flooding of the LSU packets renders the protocol resilient against malicious failures (e.g., packet dropping, alteration, or modification of the packet's *hops traversed* field). Meanwhile, the localized flooding keeps the transmission and processing overhead low.

5.3.2 The Public Key Distribution

Nodes use Public Key Distribution (PKD) packets, or attach their certified keys to LSU packets. PKD packets are flooded throughout the zone, or they may be distributed less frequently throughout an extended zone.

Alternatively, the keys can be distributed attached to LSU packets. This approach can provide for timely acquisition of the key and thus validation of routing information to nodes that move into a new zone. It can also reduce to a great extent the transmission of PKD packets, thus reducing the message complexity. On the other hand, the distribution within an extended zone can reduce the delay of validating new keys when nodes outside a zone eventually enter the zone.

Key broadcasts are timed according to the network conditions and the device characteristics. For example, a node can rebroadcast its key when it detects a substantial change of the topology of its zone; that is, if at least some percentage of nodes has departed from the node's neighborhood since the last key broadcast.

The node's certificate "vouches" for the public key. Additionally, the authenticity and freshness of the PKD packet are verified by a signature from the node that possesses and distributes the key. The PKD sequence number is set to the next available value, following the increasing values used for LSU packets. When the LSU-based key broadcast is used, no additional PKD signature is required.

Nodes validate PKD packets only if they are not already aware of the originator's public key. Upon validation, E_V and the corresponding source IP address are stored locally, along with the corresponding sequence number.[6] Each node can autonomously decide whether to validate a key broadcast or not. For example, if it communicates with a nearby destination, it might have no incentive to validate a PKD that originates from a node a large distance away. Similarly, a validation could be avoided if the node

[6]This information is maintained in a FIFO manner. If the entire sequence is covered, a new key is generated and distributed, after the node voluntarily remains "disconnected" for a period equal to NLP's *neighbor lost*. This temporary disconnection ensures that the possible change of the node's IP address does not cause neighbors to perceive this as a possible attack (i.e., spoofing of an IP address).

considers its topology view broad enough, or sufficient to support its communication. This could happen for a dense network or zone, when not all physically present links are necessary.

Malicious floods of spurious PKD packets are countered by several mechanisms: (*i*) NLP imposes a bottleneck thanks to the lost neighbor timeout, (*ii*) PKD packets will not propagate more than R hops, unless they are "carried" farther by adversaries (e.g., when they don't update the *hops traversed* field), (*iii*) nodes can autonomously decide whether to validate a public key or not (e.g., for an very high R), and (*iv*) PKD packets are also subject to restrictions imposed by the above-mentioned penalizing priority mechanism.

6 Secure Data Forwarding

No secure routing protocol, including SRP and SLSP, can 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, initially provide correct routing information during the route discovery stage, and later forge or corrupt data packets during the data forwarding stage.

Clearly, upper layer mechanisms, such as reliable transport protocols, or mechanisms currently assumed by MANET routing protocols, such as reliable data link or acknowledged routing, cannot cope with malicious disruptions. In fact, the communicating nodes may be easily deceived for long periods of time that the data flow is undisrupted, with no actual communication taking place.[7]

To cope with such attacks, the integrity of the exchanged traffic can be cryptographically protected. However, cryptographic protection of the data cannot shield the communication against denial of service (DoS). Compromised routes, i.e., routes that are not free of malicious nodes, may be repeatedly chosen, and to communicate nodes may have to rely on long cycles of disconnection detection and new route discovery, with the successive query broadcasts imposing additional overhead.

Below, we present a simple, efficient and effective protocol, the Secure Message Transmission (SMT) 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. We emphasize that the goal of SMT is

[7]For example, the adversary can corrupt the data and their checksum, forge network or transport layer acknowledgments, or, it may drop a route error message and "hide" a route breakage.

not to securely discover routes in the network - this is achieved by protocols such as the SRP and SLSP.

6.1 The Secure Message Transmission Protocol

SMT is a secure end-to-end data forwarding protocol that safeguards the communication across an unknown frequently changing network, in the presence of adversaries [32]. SMT can counter attacks against the data transmission without network monitoring and misbehavior detection, which would impose complexity and excessive overhead to the network operation. At the same time, such an improvement is achieved without restrictive assumptions on the trust, membership and size of the network, or the types of misbehavior. As a result, SMT is a practical, broadly applicable protocol.

Furthermore, SMT is capable of supporting real-time traffic, while adapting to the frequently changing network conditions. SMT can be continuously reconfigured to provide either enhanced security and resilience, or highly efficient operation in a relatively safer environment. Finally, the proposed SMT, a network layer protocol, does not rely on restrictive assumptions on cross-layer interactions, providing a self-contained solution tailored to the MANET characteristics.

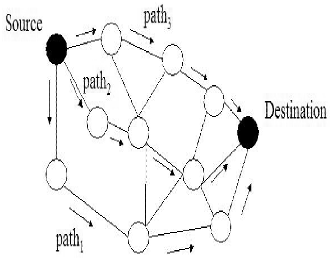

Figure 1: The Secure Message Transmission Protocol makes use of multiple diverse paths connecting the source and the destination. In particular, the Active Path Set (APS) contains paths that have not been detected as failed, either due to path breakage or because of the presence of an adversary on the path.

The basic idea behind the SMT protocol is to combine efficient end-to-end security services and a robust feedback mechanism, with dispersion of transmitted data and simultaneous usage of multiple paths. At the same time, continuous reconfiguration driven by an easy-to-implement method allows the adaptation of SMT to the requirements of the networking environment.

6.2 The SMT Operation

Our protocol determines a set of diverse routes, as shown in Figure 1. It disperses each outgoing message by introducing limited redundancy and dividing the data with the redundant information into a number of pieces. The information dispersal is based on the algorithm proposed in [39], which is in essence an error correction code.

Due to the message dispersion, the reception of a sufficient number of pieces allows successful reconstruction at the receiver's side. A low-cost cryptographic header is appended to each piece and the dispersed message is transmitted across a set of diverse, preferably node-disjoint paths. Diversity is welcome, so that a malicious node cannot harm more than one piece.

The receiver validates the incoming packets and acknowledges the successfully received packets, with the feedback cryptographically protected as well. If a sufficient number of pieces were received, the receiver reconstructs the message. Otherwise, it awaits the additional needed packets to be retransmitted by the sender. Once the message is successfully reconstructed, it is passed to the upper protocol layers.

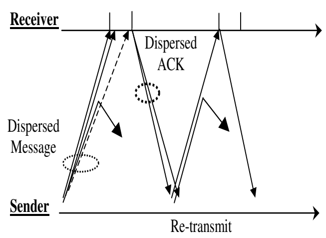

Figure 2: Simple example of the SMT protocol operation.

An illustrative example of a single message transmission is shown in Figure 2. The sender disperses the message, so that any three out of four packets are sufficient for successful reconstruction. The four packets are routed over four disjoint paths and two of them arrive intact at the receiver. The remaining two packets are compromised by malicious nodes lying on the corresponding paths; for example, one packet is dropped, and one (dashed arrow) is modified.

The receiver extracts the information from the first incoming validated packet and waits for subsequent packets, while setting a reception timer. When the fourth packet arrives, the cryptographic integrity check reveals the data tampering and the packet is rejected. At the expiration of the timer, the receiver generates an acknowledgement reporting the two successfully received packets and transmits it across the two operational paths.

It is sufficient for the sender to receive and cryptographically validate only one acknowledgement, ignoring duplicates. The two missing pieces are then retransmitted; however, one of the two packets is lost, for example, because of intermittent malicious behavior, or a benign path breakage. The receiver acknowledges the successful reception immediately, before the timer expiration, since an adequate number of packets have been received. In all cases, the sender sets a retransmission timer, so that total loss of all the message pieces or of all the acknowledgments is detected.

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 feedback transmission, but if links are not bi-directional, the destination will have to determine its own "reverse" APS.

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.

The dispersion of messages is coupled to the APS characteristics, and with the appropriate selection of the dispersion algorithm parameters discussed below. Once dispersed, the message pieces are transmitted across APS in cryptographically protected packets. If the message cannot be reconstructed at the destination, the source re-transmits the pieces that were not received, according to the feedback that is verifiably 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, a number of re-transmissions by SMT enhance its efficiency, 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 compromised transmissions, while adapting its operation to provide secure data forwarding with low delays.

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. Moreover, the simultaneous routing over a number of paths, if not the entire APS, provides the opportunity for low-cost probing of the paths. The source can easily tolerate the loss of a piece that was transmitted over a low-rated path, and the benefit from doing so can be two-fold: either the piece will be lost but the rating of a failing path will be further decreased and removed from the APS, or, the piece will be successfully received and contribute to the re-construction of the message, if an adversary lying on the path misbehaves intermittently.

6.3 The SMT Adapation

The protocol can continuously adapt its configuration in order to achieve both enhanced robustness in a highly adverse environment and efficiency in low-risk conditions. The adaptation of SMT takes into consideration the network state and the requirements of the supported application.

Intuitively, path diversity is the primary goal to meet in order to provide increased protection by disallowing any single malicious node to compromise more than one data flow. In general, the sender needs to determine a sufficiently high number of paths in order for the dispersed message to be successfully received. Although this is the most obvious solution, one cannot expect that in every occasion a high number of paths will be found. In low connectivity conditions (small number of disjoint paths), the sender could increase the transmission redundancy in order to provide increased assurance.

More specifically, the protocol adaptation is the result of the interplay among the following parameters: (*i*) K, the (sought) cardinality of the 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 information dispersal, and (*iv*) x, the maximum number of malicious nodes.

If we assume that no more than $X\%$ of the nodes may act maliciously at any time instance, then $x = X \times a$, with the number of network nodes denoted by a. In particular, nodes may either estimate or be given an estimate or prediction of the percentage of malicious nodes, which can be viewed as the probability that any single node is malicious. Instead of a, a node can use the number of nodes in its topology view.

If M out of N transmitted packets are required for successful transmission, $r = N/M$, and, for an allocation of one piece per path, K should be at least N. The larger K is, the higher the number of faults that can be tolerated. Equivalently, the higher x is, the larger K should be for a fixed r. For an APS of K paths, the required number of packets is K/r. The relationship among the interacting parameter values is shown by the condition for successful reception, which is $x \leq \lceil K \times (1 - r^{-1}) \rceil$.

Our protocol adapts its operation as follows: K, the required number of paths, is determined as a function of r, so that the probability of successful transmission is maximized. In order to do so, the source starts by constructing an APS of k node-disjoint paths, depending on the actual node connectivity of its topology view. This can be done by constructing k node-disjoint paths connecting the two end nodes, using an algorithm such as [46] with the number of hops as cost, so that the shortest k-path set has the minimum sum of the path lengths. Alternatively, a minimum-cost maximum-flow algorithm [1] with unit node capacities and a fixed goal of k paths can yield the same result. We should note that other cost measures could be used as well.

Then, let P_{GOAL} be the target probability of successful reconstruction of a dispersed message. P_{GOAL} can be provided from the application layer and correspond to the features of the supported application for example. The source determines the sought number of paths and redundancy factor to achieve a secure transmission. The calculation of the probability of success as a function of r and K can use the approximation provided in [43]. Given P_{GOAL} and k, the node calculates the corresponding redundancy factor, r_{GOAL}. Then, outgoing messages are dispersed with the redundancy value closest or equal to r_{GOAL}. Note that the source may achieve similar results with different values of M and N, a flexibility that is proven valuable.

If $N < k$, the node selects the N paths of the APS with the highest rating. Similarly, the few first most highly rated paths are selected for retransmissions, that is, transmission of fewer than M pieces. As this process continues, paths will be deemed failed, thus reducing k. Then, the node repeats the above-mentioned algorithm.

On the other hand, if $k \ll K$, then the sender can enhance the resilience of the communication by determining additional, partially disjoint paths.

Given a set of k node-disjoint paths, additional $K - k$ paths can be calculated, partially overlapping with the node-disjoint ones. If less than k malicious nodes lie on the selected paths, at least one or more packets will reach the destination. For any additional non-disjoint path, the number of faulty paths that can be tolerated increases in practice, but no guarantee can be provided for the worst case, without knowing the actual overlapping information. If the adversarial nodes constitute a cut of cardinality C_X, the result would be either a partitioned network ($C_X \geq k$) as seen by S and T, or a mere failure to reconstruct the message at the receiver ($C_X \geq k - M$).

While transmitting across the APS, the source updates the rating 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, which implies that its constituent links are discarded as well. This last procedure implies that the determination of the APS by SMT is performed in parallel and it can contribute to the update of the topology view of the node. The reverse interaction is also possible, if for example route error messages are taken into consideration, in a complementary manner, to update the path rating. Furthermore, an alternative implementation could reduce a metric for each of the path's constituent links, when it is removed from APS, and discard links only when their metric drops below a threshold. We should note that in all cases it is desirable to promptly discard connectivity that corresponds to non-operational paths.

7 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 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. 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 along with a set of rules that determine the allowed interaction with and between devices [43]. 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 [6], 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 [44]. However, if the control over a node 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 [13]. 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 off-line secure channel or perform a "pre-authentication" step through a localized channel [3]. Then, they can execute a password-based authenticated key exchange over the insecure wireless medium. Schemes that derive a shared symmetric key could use a two- or a multi-party version of the password authenticated Diffie-Hellman key-exchange algorithm [4, 11].

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, even though 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 do 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.

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, possibly multi-hop, 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 would 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 K such certificates are collected, the node acquires the complete certificate [53, 24]. 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 [19].

The solution of a key management facility that meets the requirements of the MANET environment has been proposed in [53]. To do so, the proposed instantiation of the public key infrastructure provides increased availability and fault-tolerance. The distributed 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.

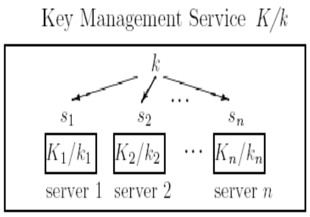

Figure 3: The configuration of a key management service comprising n servers. The service, as a whole, has a public/private key pair K/k. The public key is known to all nodes in the network, whereas the private key k is divided into n shares $s_1, s_2, \ldots s_n$, with one share for each server. Moreover, each server has a public/private key pair K_i/k_i and knows the public keys of all nodes. Reprinted with permission from [L. Zhou and Z.J. Haas. "Securing Ad Hoc Networks," *IEEE Network Magazine,* vol. 13, no. 6, November/ December 1999] ©1999 IEEE.

The CA is instantiated by a set of nodes (servers), as shown in Figure 3, for enhanced availability. However, this is not done through nave 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 3. 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, 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 older ones and cannot be combined with the old shares 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.

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 membership 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

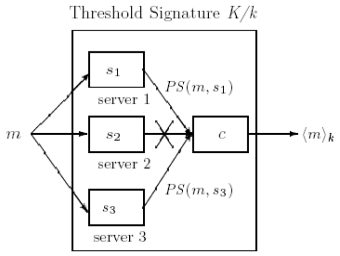

Figure 4: The calculation of a threshold signature. As an example, the service consists of three servers a $(3, 2)$ threshold cryptography scheme. K/k is the public/private key pair of the service and each server has a share s_i of the service private key. To calculate the threshold signature on a message m, each server generates a partial signature $PS(m, s_i)$ and correct servers 1 and 3 forward their 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. Reprinted with permission from [L. Zhou and Z.J. Haas. "Securing Ad Hoc Networks," *IEEE Network Magazine,* vol. 13, no. 6, November/ December 1999] ©1999 IEEE.

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. Such protocols could, in fact, approximate the assumption of reliable links between servers in [47] even in the presence of adversaries. In particular, the above-discussed protocols SRP, SLSP and SMT, lend themselves naturally to this model. Any two servers [8] can discover and maintain routes to each other, and forward service-related traffic, regardless of whether intermediate nodes are trusted or not.

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

8 Related Work

A number of secure routing protocols for MANET have appeared in the literature and are related to our protocols. They fall mainly into two categories: solutions that target to secure the route discovery, or solutions to mitigate malicious or selfish behavior regarding the forwarding of data.

In the former category, it has been proposed to tackle the protection of the route discovery by classifying nodes into different trust and privilege levels [51]. At each trust level, nodes share symmetric encryption and decryption keys to provide protection (e.g., integrity) of the routing protocol traffic against adversaries outside a specific trust level.

A number of works proposed the enhancement of existing MANET routing protocols with cryptographic primitives. Such schemes require that each node has obtained and trusts the credentials of all other nodes in the network.

It has been proposed to extend the Ad Hoc On-demand Distance Vector (AODV) routing protocol [36] with public key signatures. [52] authenticates control traffic at each relaying and utilizes a hash chain mechanism to protect the path length (hop count). [10] proposes an onion-like digital signature scheme to protect control traffic and a two-phase discovery of possibly the shortest path to the destination.

The authentication of all nodes that relay control traffic has also been proposed to secure the route discovery of the DSR protocol [18]. The scheme utilizes a broadcast authentication technique, which was initially introduced for the protection of multicast traffic flows, and requires that all nodes are equipped with synchronized clocks.

Secure link state routing protocols proposed for the "wired" Internet bear resemblance to SLSP but have additional requirements or features pertinent to the fixed-infrastructure routing. For example, [36] utilizes a robust flooding protocol and a central entity to distribute all keys throughout the network, along with the reliable flooding of link state updates throughout the entire network. [30] enhances the security of OSPF and seeks to synchronize the topology maps across all nodes or to support the full exchange of link state databases. [35] provides nodes with credentials to prove their authorization to advertise specific routing information due to the continuously changing network connectivity and membership. Finally, a number of non-link state protocols, which we review in [33], authorize the participation of nodes in routing based on their possession of credentials.

The Internet security architecture (IPsec) [21] provides authentication and integrity [22], confidentiality [23], or their combination, in addition to a framework [27] for the establishment of keys for the participating entities. The IPsec protocols that assume the existence of a fixed routing infrastructure may not be applicable to MANET. [9] Nevertheless, the goals such as the end-to-end protection of packets, and in particular, authentication, integrity and replay protection apply equally to the MANET context as well.

Transport layer protocols such as [45] relies on the services of IPsec and bear some remote resemblance to SMT. Another transport protocol [5], which utilizes the Information Dispersal Algorithm (IDA) [39] to introduce redundancy, protects against dropped ATM (Asynchronous Transfer Mode) cells to avoid TCP segments to be dropped.

As for security solutions targeting MANET, it has been proposed to detect misbehaving nodes and report such events to the rest of the network [26]. [8] takes the same direction, but, additionally, all nodes are assumed capable to authenticate traffic from all other network nodes. Both of the two previously mentioned schemes seek to detect the misbehaving nodes, i.e., nodes that do not forward packets. Similarly, [2] proposes to test each utilized path, after a threshold rate of failures has been observed, to determine where the failure occurred, without exchanging alerts. A different approach [9] provides incentive to nodes, so that they comply with protocol rules and properly relay user data.

9 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 caf, 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 per-

[9] The "Router Implementation" of IPsec does not make sense within a MANET domain. Similarly, the "Tunnel Mode" will not be applicable, unless a master/slave association exists (e.g., Bluetooth [6]), even though the dependent devices would be practically invisible at the routing layer.

form 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.

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.

10 Acknowledgements

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, the DoD Multidisciplinary University Research Initiative (MURI) program administered by the Air Force Office of Scientific Research under the grant number F49620-02-1-0233, and by the National Science Foundation grant number ANI-9980521.

References

[1] R.K. Ahuja, T.L. Magnati, and J.B. Olin, "Network Flows," *Prentice Hall,* Upper Saddle River, NJ, 1993

[2] B. Awerbuch, D. Holmer, C. Nita-Rotaru and H. Rubens, "An On-Demand Secure Routing Protocol Resilent to Byzantine Failures," in Proceedings of the *ACM Workshop on Wireless Security,* Atlanta, GA, Sept. 2002

[3] D. Balfanz, D.K. Smetters, P. Stuart, and H.C. Wang, "Talking to Strangers: Authentication in Ad Hoc Networks," in Proceedings of the *Network and Distributed System Security Symposium (NDSS),* San Diego, CA, Feb. 2002

[4] S.M. Bellovin and M. Merritt, "Encrypted Key Exchange: Password-based protocols secure against dictionary attacks," in Proceedings of the *IEEE Symposium on Security and Privacy,* Oakland, CA, May 1992

[5] A. Bestavros and G. Kim, "TCP-Boston: A Fragmentation-Tolerant TCP Protocol for ATM networks," in Proceedings of the *IEEE Infocom 1997,* Kobe, Japan, Apr. 1997

[6] Bluetooth Special Interest Group, "Specifications of the Bluetooth System," *http://www.bluetooth.com*

[7] M. Brown, D. Cheung, D. Hankerson, J.L. Hernadez, M. Kirkup and A. Menezes, "PGP in Constrained Wireless Devices," in Proceedings of the *9th USENIX Symposium,* Denver, CO, Aug. 2000

[8] S. Buchegger and J.Y. LeBoudec, "Performance Evaluation of the CONFIDANT protocol," in Proceedings of the *Third ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc),* Lausanne, Switzerland, Jun. 2002

[9] L. Buttyan and J.P. Hubaux, "Enforcing Service Availability in Mobile Ad Hoc WANs," in Proceedings of the *First ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc),* Boston, MA, Aug. 2000

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

[11] W. Diffie and M.E. Hellman, "New directions in cryptography," *IEEE Transactions in Information Theory,* Vol IT-22, No 6, p. 644-654, 1976

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

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

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

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

[16] Z.J. Haas, M.R. Pearlman, P. Samar, "The Interzone Routing Protocol (IERP) for Ad Hoc Networks," *draft-ietf-manet-zone-ierp-02.txt,* IETF MANET Working Group, Jul. 2002

[17] Z.J. Haas, M.R. Pearlman, P. Samar, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks," *draft-ietf-manet-zrp-02.txt,* IETF MANET Working Group, Jul. 2002

[18] Y.C. Hu, A. Perrig and D. Johnson, "Ariadne: A Secure on demand routing protocol," in Proceedings of the *The Eighth Annual International Conference on Mobile Computing and Networking (Mobicom),* Atlanta, GA, Sept. 2002

[19] J.P. Hubaux, L. Buttyan and S. Capkun, "The quest for security in mobile ad hoc networks," in Proceeding of the *Second ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc),* San Diego, CA, Oct. 2001

[20] D. Johnson et al, "The Dynamic Source Routing Protocol," *draft-ietf-manet-dsr-07.txt,* IETF MANET Working Group, Jul. 2002

[21] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," *IETF RFC 2401,* Nov. 1998

[22] S. Kent and R. Atkinson, "IP Authentication Header," *IETF RFC 2402,* Nov. 1998

[23] S. Kent and R. Atkinson, "IP Encapsulating Security Payload," *IETF RFC 2406,* Nov. 1998

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

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

[26] S. Marti, T.J. Giuli, K. Lai and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," in Procedings of the *The Sixth Annual International Conference on Mobile Computing and Networking (Mobicom)* Boston, MA, Aug. 2000

[27] D. Maughan, M. Schertler, M. Schneider and J. Turner, "Internet Security Association and Key Management Protocol," *IETF RFC 2408,* Nov. 1998

[28] A. Menezes, P.V. Oorschot and S. Vanstone, "Handbook of Applied Cryptography," *CRC Press,* Oct. 1996 (5th reprinting Aug. 2001)

[29] G. Montenegro and C. Canstellucia, "SUCV Identifiers and Addresses," *draft-montenegro-sucv-02.txt,* Internet Engineering Task Force (IETF), Jul. 2002

[30] S. Murphy et al, "Retrofitting Security into Internet Infrastructure Protocols," in Proceedings of the *DARPA Information Survivability Conference and Exposition (DISCEX'00),* 2000

[31] P. Papadimitratos and Z.J. Haas, "Secure Routing for Mobile Ad Hoc Networks," in Proceedings of the *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002),* San Antonio, TX, Jan. 2002

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

[33] P. Papadimitratos and Z.J. Haas, "Securing the Internet Routing Infrastructure" *IEEE Communications Magazine,* Vol. 40, No. 10, Oct. 2002

[34] P. Papadimitratos and Z.J. Haas, "Secure Link State Routing for Mobile Ad Hoc Networks" in Proceedings of the *IEEE CS Workshop on Security and Assurance in Ad hoc Networks, (in conjunction with the 2003 International Symposium on Applications and the Internet),* Orlando, FL, Jan. 2003

[35] C. Partridge et al, "FIRE: flexible Intra-AS routing environment," *ACM SIGCOMM Computer Comm. Review,* Vol. 30, Issue 4, Aug. 2000

[36] C.E. Perkins, E.M. Royer and S.R. Das, "Ad hoc On-Demand Distance Vector Routing," *draft-ietf-manet-aodv-08.txt,* IETF MANET Working Group, Jun. 2001

[37] R. Perlman, "Interconnections: Bridges, Router, Switches and Internetworking Protocols," *Addisson-Wesley*, 2000

[38] M.R. Pearlman and Z.J. Haas, "Determining the Optimal Configuration of for the Zone Routing Protocol," *IEEE JSAC, special issue on Ad-Hoc Networks,* vol. 17, no.8, Aug. 1999

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

[40] R. Rivest, A. Shamir and L. Adleman, "A method for obtaining Digital Signatures and Public Key Cryptosystems," *Communications of the ACM,* Vol. 21, No 2, pp. 120-126, Feb. 1978

[41] R. Rivest, "The MD5 Message-Digest Algorithm," *IETF RFC 1321,* Apr. 1992

[42] "Secure Hash Standard," *Federal Information Processing Standard,* FIPS 180-2, Aug. 2002

[43] F. Stajano and R. Anderson, "The Resurrecting Duckling: Security Issues for Ad Hoc Wireless Networks," in Proceedings of the *7th International Workshop on Security Protocols,* LNCS, Springer-Verlag, 1999

[44] F. Stajano, "The Resurrecting Duckling - What next?" in Proceedings of the *8th International Workshop on Security Protocols,* LNCS, Springer-Verlag, 2000

[45] R. Stewart et al, "Stream Control Transmission Protocol," *IETF RFC 2960,* Oct. 2000

[46] J.W. Suurballe, "Disjoint Paths in a Network," *Networks,* vol. 4, p. 125-145, 1974

[47] S. Thomson and T. Narten, "IPv6 Stateless Address Autoconfiguration," *IETF RFC 2462,* Dec. 1998

[48] C.K. Toh, "Associativity-Based Routing for Ad-Hoc Mobile Networks," *Wireless Personal Communications,* Vol. 4, No. 2, pp. 1-36, Mar. 1997

[49] A. Tsirigos and Z.J. Haas, "Multipath Routing in the Presence of Frequent Topological Changes," *IEEE Communications Magazine,* p. 132-138, Nov. 2001

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

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

[52] M.G. Zapata and N. Asokan, "Securing Ad hoc Routing Protocols," in Proceedings of the *ACM Workshop on Wireless Security,* Atlanta, GA, Sept. 2002

[53] L. Zhou and Z.J. Haas, "Securing Ad Hoc Networks," *IEEE Network Magazine,* vol. 13, no.6, Nov./Dec. 1999