THIRD QUARTER 2005, VOLUME 7, NO. 3

IEEE COMMUNICATIONS SURVEYS
The Electronic Magazine of Original Peer-Reviewed Survey Articles

www.comsoc.org/pubs/surveys

SECURE ROUTING FOR MOBILE AD HOC NETWORKS

PATROKLOS G. ARGYROUDIS AND DONAL O'MAHONY, UNIVERSITY OF DUBLIN, TRINITY COLLEGE

ABSTRACT

In this article we present a survey of secure ad hoc routing protocols for mobile wireless networks. A mobile ad hoc network is a collection of nodes that is connected through a wireless medium forming rapidly changing topologies. The widely accepted existing routing protocols designed to accommodate the needs of such self-organized networks do not address possible threats aiming at the disruption of the protocol itself. The assumption of a trusted environment is not one that can be realistically expected; hence, several efforts have been made toward the design of a secure and robust routing protocol for ad hoc networks. We briefly present the most popular protocols that follow the table-driven and the source-initiated on-demand approaches. Based on this discussion we then formulate the threat model for ad hoc routing and present several specific attacks that can target the operation of a protocol. In order to analyze the proposed secure ad hoc routing protocols in a structured way we have classified them into five categories: solutions based on asymmetric cryptography; solutions based on symmetric cryptography; hybrid solutions; reputation-based solutions; and a category of add-on mechanisms that satisfy specific security requirements. A comparison between these solutions can provide the basis for future research in this rapidly evolving area.

During the last few years we have all witnessed steadily increasing growth in the deployment of wireless and mobile communication networks. Mobile ad hoc networks consist of nodes that are able to communicate through the use of wireless mediums and form dynamic topologies. The basic characteristic of these networks is the complete lack of any kind of infrastructure, and therefore the absence of dedicated nodes that provide network management operations as do the traditional routers in fixed networks. In order to maintain connectivity in a mobile ad hoc network all participating nodes have to perform routing of network traffic. The cooperation of nodes cannot be enforced by a centralized administration authority since one does not exist. Therefore, a network-layer protocol designed for such self-organized networks must enforce connectivity and security requirements in order to guarantee the undisrupted operation of the higher-layer protocols.

Unfortunately all of the widely used ad hoc routing protocols have no security considerations and trust all the participants to correctly forward routing and data traffic. This assumption can prove to be disastrous for an ad hoc network that relies on intermediate nodes for packet forwarding. Simulations have shown that if 10 percent to 40 percent of the nodes that participate in an ad hoc network perform malicious operations, then the average throughput degradation reaches 16 percent to 32 percent [1]. Earlier surveys and review papers presenting comparisons of ad hoc routing protocols completely ignored security problems [2–4]. This article presents a survey of the solutions that address the problem of secure and robust routing in mobile ad hoc networks. We are not concerned with solutions that address the protection of the wireless physical layer against denial of service attacks since such problems lie outside the scope of this survey.

The following section presents a brief introduction to the general problem of ad hoc routing, which is required since several of the surveyed proposed solutions secure existing protocols. We present the possible attacks that a malicious node can use for disrupting the operation of a routing protocol in a self-organized network. We analyze the already proposed secure ad hoc routing protocols that exist in the literature and present their operational principles. An important part of our work focuses on the comparison of these protocols and the identification of possible research directions. We then conclude the article.

1553-877X/05/$20.00 © 2005 IEEE

Routing in Mobile Ad Hoc Networks

Routing in mobile ad hoc networks faces additional problems and challenges when compared to routing in traditional wired networks with fixed infrastructure. There are several well-known protocols in the literature that have been specifically developed to cope with the limitations imposed by ad hoc networking environments. The problem of routing in such environments is aggravated by limiting factors such as rapidly changing topologies, high power consumption, low bandwidth, and high error rates [2]. Most of the existing routing protocols follow two different design approaches to confront the inherent characteristics of ad hoc networks: the *table-driven* and the *source-initiated on-demand* approaches. The following sections analyze in more detail these two design approaches, and briefly present example protocols that are based on them. Such an introduction is necessary since most of the secure protocols presented later are built on top of existing ad hoc routing protocols.

Table-driven Ad Hoc Routing Protocols

Table-driven ad hoc routing protocols maintain at all times routing information regarding the connectivity of every node to all other nodes that participate in the network. Also known as *proactive*, these protocols allow every node to have a clear and consistent view of the network topology by propagating periodic updates [2]. Therefore, all nodes are able to make immediate decisions regarding the forwarding of a specific packet. On the other hand, the use of periodic routing messages has the effect of having a constant amount of signaling traffic in the network, totally independent of the actual data traffic and the topology changes. As an example of two protocols that follow the table-driven design approach, we will briefly present the Destination-Sequenced Distance-Vector (DSDV) protocol [5] and the Optimized Link State Routing (OLSR) protocol [6].

Destination-Sequenced Distance-Vector Routing — DSDV is a table-driven routing protocol based on the Bellman-Ford algorithm [7]. The DSDV protocol can be used in mobile ad hoc networking environments by assuming that each participating node acts as a router. Each node must maintain a table that consists of all the possible destinations. In more detail, an entry of the table contains the address identifier of a destination, the shortest known distance metric to that destination measured in hop counts, and the address identifier of the node that is the first hop on the shortest path to the destination [5]. Furthermore, the DSDV protocol adds a sequence number to each table entry assigned by the destination node, preventing the formation of routing loops caused by stale routes. The routing tables are maintained by periodically transmitted updates by each router to all the neighboring routers. For a more detailed description the interested reader can see [5].

Optimized Link State Routing (OLSR) — The Optimized Link State Routing (OLSR) protocol is a proactive link state routing protocol based on the Open Shortest Path First (OSPF) protocol [8]. OLSR has been specifically developed to support mobile ad hoc networks and the constraints they impose on routing. The OLSR protocol can be conceptually divided into three different operations: *neighbor sensing*, *distribution of signaling traffic,* and *distribution of topological information* [6]. Neighbor sensing in OLSR is accomplished by transmitting periodic hello messages that contain the generating node's address identifier, a list of its neighboring nodes, and the type of link it has with each neighbor (e.g. symmetric or asymmetric). For the distribution of signaling traffic, OLSR adopts a flooding mechanism whereby every node forwards a flooded message that it has not forwarded previously. Finally, the distribution of topological information function is realized with the use of periodic *topology control* messages that result in each node knowing a partial topology graph of the network which is then used for the computation of optimal routes [4, 6].

Source-Initiated On-demand Ad hoc Routing Protocols

An alternative approach to that followed by table-driven protocols is the source-initiated on-demand routing. According to this approach, a route is created only when the source node requires a route to a specific destination. A route is acquired by the initiation of a *route discovery* function by the source node. The data packets transmitted while a route discovery is in process are buffered and are sent when the path is established. An established route is maintained as long as it is required through a *route maintenance* procedure. The Ad hoc On-demand Distance Vector (AODV) routing protocol [9] and the Dynamic Source Routing protocol [10] are examples of this category of protocols, also known as reactive.

Ad hoc On-demand Distance Vector Routing (AODV) — The AODV protocol uses route request (RREQ) messages flooded through the network in order to discover the paths required by a source node. An intermediate node that receives a RREQ replies to it using a *route reply* message only if it has a route to the destination whose corresponding destination sequence number is greater or equal to the one contained in the RREQ [9]. This effectively means that an intermediate node replies to a RREQ only if it has a *fresh* enough route to the destination. Otherwise, an intermediate node broadcasts the RREQ packet to its neighbors until it reaches the destination. The destination unicasts a RREP back to the node that initiated the route discovery by transmitting it to the neighbor from which it received the RREQ. As the RREP is propagated back to the source, all intermediate nodes set up forward route entries in their tables. The route maintenance process utilizes link-layer notifications, which are intercepted by nodes neighboring the one that caused the error. These nodes generate and forward route error (RERR) messages to their neighbors that have been using routes that include the broken link. Following the reception of a RERR message a node initiates a route discovery to replace the failed paths.

Dynamic Source Routing (DSR) — The Dynamic Source Routing (DSR) protocol is based on a method known as *source routing* [10]. The route discovery process in DSR is similar to the one used by AODV, except that each intermediate node that broadcasts a route request packet adds its own address identifier to a list carried in the packet. The destination node generates a route reply message that includes the list of addresses received in the route request and transmits it back along this path to the source. Route maintenance in DSR is accomplished through the confirmations that nodes generate when they can verify that the next node successfully received a packet. These confirmations can be link-layer Acknowledgments, passive Acknowledgments, or network-layer Acknowledgments specified by the DSR protocol. When a node is not able to verify the successful reception of a packet it tries to retransmit it. When a finite number of retransmissions fail, the node generates a route error message that specifies the problematic link, transmitting it to the source node.

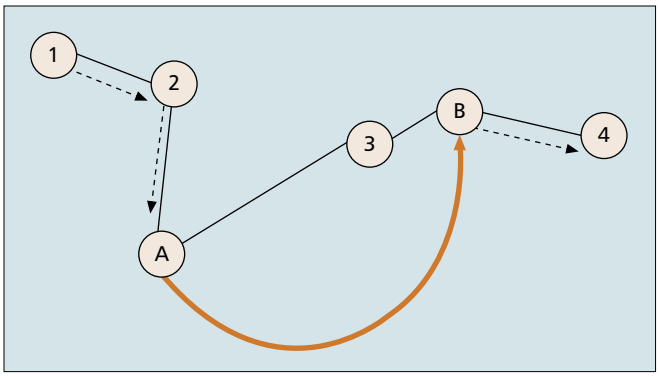

Figure 1. *A wormhole attack performed by colluding malicious nodes A and B.*

Security Problems with Existing Ad hoc Routing Protocols

The main assumption of the previously presented ad hoc routing protocols is that all participating nodes do so in good faith and without maliciously disrupting the operation of the protocol [11, 12]. However, the existence of malicious entities cannot be disregarded in any system, especially in open systems such as ad hoc networks. The RPSEC IETF working group has performed a threat analysis that is applicable to routing protocols employed in a wide range of application scenarios [13]. According to this work, the routing function can be disrupted by *internal* or *external* attackers. An internal attacker can be any legitimate participant of the routing protocol. An external attacker is defined as any other entity. As we have previously noted, we consider denial-of-service attacks that target the utilized wireless medium, such as frequency jamming, outside the scope of our threat model. Two commonly used countermeasures against jamming are frequency hopping spread spectrum (FHSS) and direct sequence spread spectrum (DSSS) [14]. Furthermore, outside the scope of our threat model are transport layer attacks, such as session hijacking, and application layer attacks, such as repudiation-based attacks and user information disclosure.

The strongest assumption for an external attacker is that it is able to eavesdrop the communication between two legitimate network participants, inject fabricated messages, and delete, alter, or replay captured packets. Weaker assumptions of external attackers include the ability to inject messages but not read them, or read and replay messages but not inject new messages, or only the ability to read messages. Cryptographic solutions can be employed to prevent the impact of external attackers by mutual authentication of the participating nodes through digital signature schemes [15]. However, the underlying protocols should also be considered since an attacker could manipulate a lower-level protocol to interrupt a security mechanism in a higher level. Although these attacks are a significant part of a complete threat assessment, our analysis focuses only on network-layer threats and countermeasures.

Internal attackers have the capabilities of the strongest outside attacker, as they are legitimate participants of the routing process. Having complete access to the communication link, they are able to advertise false routing information at will and force arbitrary routing decisions on their peers [16]. One of the most difficult problems to detect in routing is that of *byzantine failures*. These failures are the result of nodes that behave in a way that does not comply with the protocol. The reasons for the erroneous behavior could be software or hardware faults, mistakes in the configuration, or malicious compromises. Attempts to solve the problem of byzantine failures have been proposed for both infrastructure [17] and infrastructureless networks [18].

Based on this threat analysis and the identified capabilities of the potential attackers, we will now discuss several specific attacks that can target the operation of a routing protocol in an ad hoc network.

Location Disclosure [19]: Location disclosure is an attack that targets the privacy requirements of an ad hoc network. Through the use of traffic analysis techniques [20], or with simpler probing and monitoring approaches, an attacker is able to discover the location of a node, or even the structure of the entire network.

Black Hole [16]: In a black hole attack a malicious node injects false route replies to the route requests it receives, advertising itself as having the shortest path to a destination. These fake replies can be fabricated to divert network traffic through the malicious node for eavesdropping, or simply to attract all traffic to it in order to perform a denial of service attack by dropping the received packets.

Replay [13]: An attacker that performs a replay attack injects into the network routing traffic that has been captured previously. This attack usually targets the freshness of routes, but can also be used to undermine poorly designed security solutions.

Wormhole [21]: The wormhole attack is one of the most powerful presented here since it involves the cooperation between two malicious nodes that participate in the network. One attacker, e.g. node A, captures routing traffic at one point of the network and tunnels them to another point in the network, to node B, for example, that shares a private communication link with A. Node B then selectively injects tunneled traffic back into the network (Fig. 1). The connectivity of the nodes that have established routes over the wormhole link is completely under the control of the two colluding attackers.

Blackmail [22]: This attack is relevant against routing protocols that use mechanisms for the identification of malicious nodes and propagate messages that try to blacklist the offender. An attacker may fabricate such reporting messages and try to isolate legitimate nodes from the network. The security property of non-repudiation can prove to be useful in such cases since it binds a node to the messages it generated [23].

Denial of Service: Denial of service attacks aim at the complete disruption of the routing function and therefore the entire operation of the ad hoc network. Specific instances of denial of service attacks include the *routing table overflow* [19] and the *sleep deprivation torture* [24]. In a routing table overflow attack the malicious node floods the network with bogus route creation packets in order to consume the resources of the participating nodes and disrupt the establishment of legitimate routes. The sleep deprivation torture attack aims at the consumption of batteries of a specific node by constantly keeping it engaged in routing decisions.

Routing Table Poisoning: Routing protocols maintain tables that hold information regarding routes of the network. In poisoning attacks the malicious nodes generate and send fabricated signaling traffic, or modify legitimate messages from other nodes, in order to create false entries in the tables of the participating nodes. For example, an attacker can send routing updates that do not correspond to actual changes in the topology of the ad hoc network. Routing table poisoning attacks can result in the selection of non-optimal routes, the creation of routing loops, bottlenecks, and even partitioning certain parts of the network.

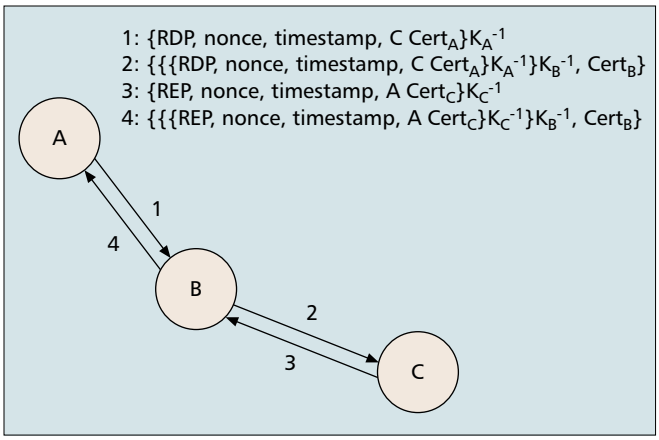

Figure 2. *Route discovery in the ARAN protocol. Messages 1 and 2 are broadcast, while 3 and 4 are unicast ($\{M\}K_N^{-1}$ denotes a signature on message M generated by private key K_N^{-1} of node N).*

Secure Ad Hoc Routing

There exist several proposals that attempt to architect a secure routing protocol for ad hoc networks, in order to offer protection against the attacks mentioned in the previous section. These proposed solutions are either completely new stand-alone protocols, or in some cases incorporations of security mechanisms into existing protocols (e.g. DSR and AODV). As we will see, the design of these solutions focuses on providing countermeasures against specific attacks, or sets of attacks. Furthermore, a common design principle in all the examined proposals is the performance-security trade-off balance. Since routing is an essential function of ad hoc networks, the integrated security procedures should not hinder its operation. Another important part of the analysis is the examination of the assumptions and the requirements on which each solution depends. Although a protocol might be able to satisfy certain security constraints, its operational requirements might thwart its successful employment.

In order to analyze the proposed solutions in a structured manner we have classified them into five categories: solutions based on asymmetric cryptography; solutions based on symmetric cryptography; hybrid solutions; reputation-based solutions; and a category of mechanisms that provide security for ad hoc routing. However, this classification is only indicative since many solutions can be classified into more than one category. As we will see in the rest of this article, most proposals follow similar approaches to solve the problems of insecure ad hoc routing protocols, hindering extensive classification attempts.

Asymmetric Cryptography Solutions

Protocols that use asymmetric cryptography to secure routing in mobile ad hoc networks require the existence of a universally trusted third party (TTP). The TTP issues certificates that bind a node's public key with a node's persistent identifier. Furthermore, the TTP can be either online or offline. In approaches that use an online TTP, revocation of the issued certificates is accomplished by broadcasting certificate revocation lists (CRLs) in the network. In offline systems revocation becomes a particularly complicated problem and usually involves the exchange of recommendations between the participating nodes. Although this category presents only one protocol, ARAN, many of the other protocols presented in other categories that use asymmetric cryptography operate in a similar manner and have similar requirements and limitations.

Authenticated Routing for Ad hoc Networks (ARAN) — The Authenticated Routing for Ad hoc Networks (ARAN) protocol, proposed in [25], is a stand-alone solution for securing routing in ad hoc networking environments. ARAN utilizes cryptographic certificates in order to achieve the security goals of authentication and non-repudiation.

ARAN, an on-demand secure ad hoc routing protocol, consists of three distinct operational stages, of which the first two are compulsory and the third is optional. The first stage is, in essence, a *preliminary certification* process that requires the existence of a trusted certification authority (CA). Each node, before attempting to connect to the ad hoc network, must contact the certification authority and request a certificate for its address and public key. The protocol assumes that each node knows a priori the public key of the certification authority. The second operational stage of the protocol is the route discovery process that provides end-to-end authentication. This ensures that the intended destination was indeed reached. Each node must maintain a routing table with entries that correspond to the source-destination pairs that are currently active. The route discovery of the ARAN protocol begins with a node broadcasting a route discovery packet (RDP) to its neighbors. The RDP includes the certificate of the initiating node, a nonce, a timestamp, and the address of the destination node. Furthermore, the initiating node signs the RDP. Each node validates the signature with the certificate, updates its routing table with the neighbor from which it received the RDP, signs it, and forwards it to its neighbors after removing the certificate and the signature of the previous node (but not the initiator's signature and certificate). The signature prevents malicious nodes from injecting arbitrary route discovery packets that alter routes or form loops [25]. The destination node eventually receives the RDP and replies with a reply packet (REP). The REP contains the address of the source node, the destination's certificate, a nonce, and the associated timestamp. The destination node signs the REP before transmitting it. The REP is forwarded back to the initiating node by a process similar to the process described for the route discovery, except that the REP is unicasted along the reverse path. The source node is able to verify that the destination node sent the REP by checking the nonce and the signature. Figure 2 illustrates the process of route discovery in ARAN.

The ARAN protocol does not allow intermediate nodes that have paths to a destination to reply to a route discovery packet. This guarantees that only the destination can answer a route discovery, thus ensuring loop freedom but at the cost of high latency [25].

The third operational stage of the ARAN protocol is optional and ensures that the *shortest paths* are discovered. However, this optimization comes at a high cost. After the source has a route to the destination, it broadcasts a signed shortest path confirmation (SPC) message to its neighbors, which includes the destination address, a nonce, a timestamp, and its certificate. The message is also encrypted with the destination's public key. Each node that receives the message signs it, incorporates its own certificate, and encrypts the message again with the public key of the destination. As in the previous stage, each receiving node updates its routing table in order to avoid forwarding duplicate packets, and to route the reply packet from the destination back to the source. The destination verifies the validity of all the signatures and replies to the first SPC, as well as any later SPCs with a shorter path, with a recorded shortest path (RSP) message. Upon receiving the RSP, the source node verifies the nonce it sent with the SPC.

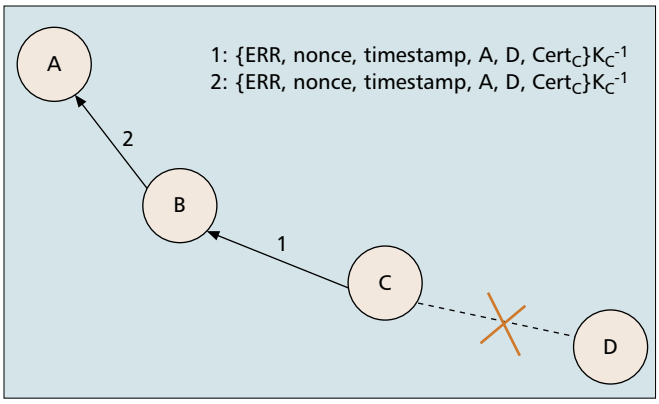

Figure 3. *Route maintenance in the ARAN protocol. Error messages are broadcast without any modification or additional signing by nodes that use the reporting node as a next hop.*

Route maintenance in the ARAN protocol is achieved with broadcasted error (ERR) messages signed by the nodes that generate them in order to report broken links. The signed ERR messages provide non-repudiation prohibiting malicious nodes from generating false broken link reports. The ERR messages include a nonce and a timestamp in order to ensure protection against replay attacks. Nodes that have paths with the node that reports the broken link in their tables rebroadcast the ERR packet exactly as they receive it (Fig. 3).

ARAN uses limited-time certificates. The certification server broadcasts a revocation message to the network when a certificate must be revoked. All receiving nodes forward this broadcast to their neighbors, and recompute routing in order to avoid transmission through the node with the revoked certificate. As mentioned by the authors, this revocation process is not safe since a revocation message might not be forwarded by the malicious node creating a partition in the network.

The ARAN protocol requires a trusted certification authority to exist in the ad hoc network in order to authenticate routing traffic. Authentication in ARAN is provided through public key cryptography. Routing traffic messages, such as route discoveries and route replies, must be signed by the node that generates or forwards them.

Symmetric Cryptography Solutions

This category presents solutions that rely solely on symmetric cryptography to secure the function of routing in wireless ad hoc networks. The most commonly utilized mechanisms are *hash functions* and *hash chains*. A one-way hash function is a function that takes an input of arbitrary length and returns an output of fixed length [26]. Hash functions have the property of being computationally expensive to reverse, i.e. if $h = f(m)$, it is hard to compute m such as $f(m) = h$. There are several well-known hash functions that possess these properties, such as SHA-1 [27] and MD5 [28]. A hash chain can be generated by applying repeatedly a given hash function to a random number known as the *root* of the chain. Simply stated, in order to generate a hash chain of length n a hash function is applied n times to a random value p, and the final hash q that is obtained is called the *anchor* of the chain. In order to use a hash chain for authentication purposes an initial authenticated element of the chain is assumed, usually the anchor. Given this, it is possible to verify the authenticity of the elements that come later in the sequence. Since hash functions are especially lightweight when compared to other symmetric and asymmetric cryptographic operations, they have been extensively used in the context of securing ad hoc routing, and specifically in hop count authentication.

Secure Routing Protocol (SRP) — The Secure Routing Protocol (SRP) is a set of security extensions that can be applied to any ad hoc routing protocol that utilizes broadcasting as its route querying method [29]. The authors specifically mention DSR as a particularly appropriate protocol for incorporating their proposed security extensions. The operation of SRP requires the existence of a security association (SA) between the source node initiating a route query and the destination node. This security association can be utilized in order to establish a shared secret key between the two, which is used by SRP.

The SRP protocol appends a header (SRP header) to the packet of the basis routing protocol. The source node sends a route request with a query sequence (QSEQ) number that is used by the destination in order to identify outdated requests, a random query identifier (QID) that is used to identify the specific request, and the output of a keyed hash function, as shown in Fig. 4. The input to the function is the IP header, the header of the basis protocol, and the shared secret between the two nodes.

The mutable fields of the request, like the accumulated addresses of the intermediate nodes, are transmitted in the clear. The intermediate nodes broadcast the query to their neighbors after updating their routing tables. The query is dropped in case it has the same QID with an entry in an intermediate node's routing table. Furthermore, all nodes maintain a priority ranking of their neighbors according to the rate of the generated route queries. Nodes that generate a low rate of queries have a higher priority. This guarantees that the routing protocol is responsive [29]. The destination confirms that the query is not outdated or replayed through the QSEQ, and verifies its integrity and authenticity through the calculation of the keyed hash. In response to a valid route query the destination node generates a number of replies with different routes, at most as many as its number of neighbors. This mechanism is an additional protection against malicious nodes that attempt to modify route replies. A route reply consists of the path from the source to the destination, the QSEQ and QID numbers. The integrity and authenticity of the reply is ensured through the same method as the route request, namely with a message authentication code (MAC). The source node checks the QSEQ and QID numbers of the reply in order to verify that they correspond to the active query, compares the IP source route with the reverse of the route in the payload of the reply, and if they match it calculates the MAC. Although the authors do not encourage the optimization of intermediate node replies to a route query as a severe vulnerability, they propose an extension to SRP that implements this functionality. They accomplish this by defining groups of nodes with shared secrets. For more details see [29].

Route maintenance is realized in SRP by route error messages that are source-routed along the prefix of the path that they report as broken. When the notified node receives a route error packet, it compares the route taken by the packet with the prefix of the corresponding route. However, this approach cannot guarantee that a malicious node did not fabricate the route error packets.

SRP consists of several security extensions that can be applied to existing ad hoc routing protocols providing end-to-end authentication. The operational requirement of SRP is the existence of a security association between every source and destination node. The security association is used to establish a shared secret between the two nodes, and the nonmutable fields of the exchanged routing messages are protected by this shared secret.

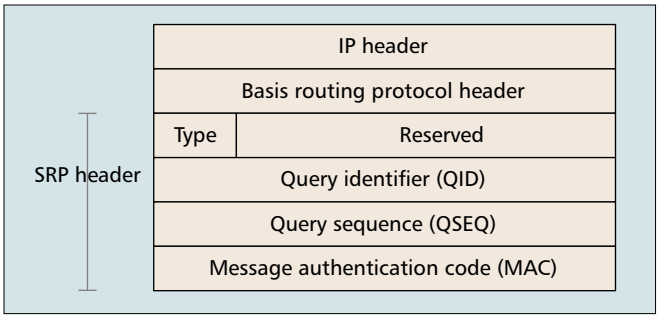

Figure 4. *SRP Packet Header. The input to the keyed hash function is the IP header, the header of the basis protocol, and the shared secret.*

Secure Efficient Ad hoc Distance Vector Routing — The Secure Efficient Ad hoc Distance vector (SEAD) is a secure ad hoc network routing protocol based on the design of the Destination-Sequenced Distance-Vector (DSDV) algorithm [22]. In order to find the shortest path between two nodes, the distance vector routing protocols utilize a distributed version of the Bellman-Ford algorithm [5]. The SEAD routing protocol employs the use of hash chains to authenticate hop counts and sequence numbers.

Applying repeatedly a one-way hash function to a random value creates a hash chain. The elements of such a chain are used to secure the updates of the routing protocol. SEAD requires the existence of an authentication and key distribution scheme in order to authenticate one element of a hash chain between two nodes. Given this authenticated element, a node is able to verify later elements in the chain [22]. When a node transmits a routing update it includes one value from the hash chain for each entry in the update message. Moreover, it includes the address of the destination node (or its own address if the update concerns itself), the metric and the sequence number of the destination (from its routing table), and a hash value equal to the hash of the hash value received when it learned the route to the destination. This hash value can be authenticated by the nodes that receive this routing update since they have an already authenticated element of the same hash chain. As noted by the authors of the protocol, this mechanism allows other nodes to only *increase* the metric in a routing update, but not to *decrease* it. In order to avoid denial of service attacks, a receiving node can specify the exact number of hashes it is willing to perform for each authentication. A node that receives a routing update verifies the authentication of each entry of the message. The hash value of each entry is hashed the correct number of times and it is compared to the previously authenticated value. Depending on this comparison the routing update is either accepted as authenticated or discarded.

The SEAD routing protocol proposes two different methods in order to authenticate the source of each routing update. The first method requires clock synchronization between the nodes that participate in the ad hoc network, and employs broadcast authentication mechanisms such as TESLA [30]. The second method requires the existence of a shared secret between each pair of nodes. This secret can be utilized in order to use a message authentication code (MAC) between the nodes that must authenticate a routing update message.

In SEAD every node that participates in the ad hoc network has a hash chain. The elements of the hash chain are used in succession to authenticate the entries in the transmitted routing messages, given that an initial authenticated element exists. The hash chains have a finite size and must be generated again when all their elements have been used.

Ariadne — Ariadne is a secure on-demand ad hoc routing protocol based on DSR and developed by the authors of the SEAD protocol presented in the previous section. Security in Ariadne follows an end-to-end approach, while the SEAD protocol employs hop-by-hop security mechanisms due to the distance vector routing philosophy it adopts. Ariadne assumes the existence of a shared secret key between two nodes, and uses a message authentication code (MAC) in order to authenticate point-to-point messages between these nodes [31]. Additionally, Ariadne employs the TESLA broadcast authentication protocol to authenticate broadcast messages, such as route requests. In TESLA a sender generates a one-way key chain and defines a schedule according to which it discloses the keys of the chain in reverse order from generation [30]. Therefore, time synchronization is an absolute requirement of ad hoc networks that use Ariadne.

When a node transmits a route request it includes its own address, the address of the destination node, a number (ID) that identifies the current route discovery, a TESLA *time interval* that denotes the expected arrival time of the request to the destination, a hash chain consisting of its address, the destination address, the ID, and the time interval, as well as two empty lists, a *node list* and a *MAC list*. A neighboring node that receives the route request checks the validity of the TESLA time interval. A valid time interval is one that it is not too far in the future and its corresponding key must not have been disclosed yet. A packet with an invalid time interval is discarded. Otherwise, the current node inserts its address in the node list, replaces the hash chain with a new one consisting of its address plus the old one, and appends a MAC of the entire packet to the MAC list. The MAC is calculated using the TESLA key that corresponds to the time interval of the request. Then the neighboring node broadcasts the route request to its own neighbors.

The destination node checks the validity of the route request upon receiving it. A route request is considered valid if the keys from the specified time interval have not been disclosed yet, and if the included hash chain can be verified. The destination generates and broadcasts a route reply packet for every valid route request it receives. A route reply contains the same fields with the corresponding route request, and additionally it contains a *target MAC* field and an empty *key list*. The target MAC field is set to the calculated MAC of the preceding fields of the route reply and the key that the destination shares with the initiator. The reply is forwarded back to the initiator by following the reverse of the route included in the node list, as specified by the DSR protocol. An intermediate node that receives the route reply waits until the specified time interval allows it to disclose its key, which it appends to the key list and forwards the message to the next node. Upon receiving a route reply, the initiator verifies the validity of every key in the key list, of the target MAC, and of every MAC in the MAC list.

The Ariadne protocol also specifies a mechanism for securing route maintenance, which ensures the validity of route error messages concerning broken links in the ad hoc network. A node that generates a route error includes TESLA authentication details in the message. Therefore, every node that forwards the route error toward the destination of the message is able to authenticate it. The intermediate nodes buffer the route error message and its authentication does not take place until the node that generated it discloses the key [31].

Ariadne is based on DSR and provides end-to-end security mechanisms for ad hoc routing. Ariadne utilizes a message authentication code in order to authenticate routing table entries. The most important requirement of Ariadne is the

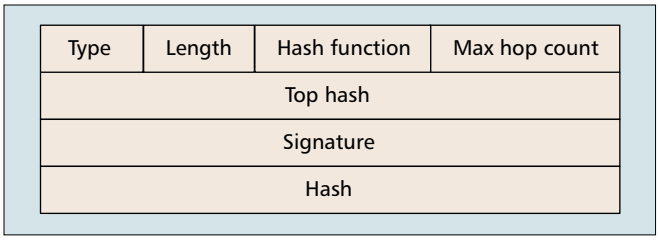

Figure 5. *SAODV Protocol Header. The top hash field is the result of the application of the hash function max hop count times to the hash field, a randomly generated number.*

existence of clock synchronization in the ad hoc network. The basic Ariadne protocol can be disrupted by wormhole attacks, but an extension developed by the authors can be utilized to secure against it [32].

Hybrid Solutions

In this category we have included the secure routing protocols that employ both symmetric and asymmetric cryptographic operations. The most common approach is to digitally sign the immutable fields of routing messages in order to provide integrity and authentication, and to use hash chains to protect the hop count metric.

Secure Ad hoc On-demand Distance Vector Routing (SAODV) — Secure Ad hoc On-demand Distance Vector (SAODV) is a proposal for security extensions to the AODV protocol [33]. The proposed extensions utilize digital signatures and hash chains in order to secure AODV packets. In particular, cryptographic signatures are used for authenticating the non-mutable fields of the messages, while a new one-way hash chain is created for every route discovery process to secure the hop-count field, which is the only mutable field of an AODV message. Since the protocol uses asymmetric cryptography for digital signatures it requires the existence of a key management mechanism that enables a node to acquire and verify the public key of other nodes that participate in the ad hoc network.

In order to facilitate the transmission of the information required for the security mechanisms, SAODV defines extensions to the standard AODV message format. These SAODV extensions consist of the following fields. The *hash function* field identifies the one-way hash function that is used. The field *max hop count* is a counter that specifies the maximum number of nodes a packet is allowed to go through. The *top hash* field is the result of the application of the hash function max hop count times to a randomly generated number. Finally, the field *hash* is this random number, as shown in Fig. 5 [33].

When a node transmits a route request or a route reply AODV packet it sets the max hop count field equal to the time to live (TTL) field from the IP header, generates a random number and sets the hash field equal to it, and applies the hash function specified by the corresponding field max hop count times to the random number, storing the calculated result to the top hash field. Moreover, the node digitally signs all fields of the message, except the hop count field from the AODV header and the hash field from the SAODV extension header. An intermediate node that receives a route request or a route reply must verify the integrity of the message and the hop count AODV field. The integrity requirement is accomplished by verifying the digital signature. The *hop count* field is verified by comparing the result of the application of the hash function max hop count minus hop count times to the hash field with the value of the top hash field. Before the packet is re-broadcast by the intermediate node, the value of the hash field is replaced by the result of the calculation of the one-way hash of the field itself in order to account for the new hop.

As in previous protocols we have seen, the authors mention that the main problem with securing an on-demand protocol such as AODV is that it allows intermediate nodes with *fresh* routes to reply to a route query since the reply has to be signed on behalf of the destination node. In order to overcome this problem the authors suggest two solutions. The first solution is to forbid intermediate nodes to respond to route request messages since they cannot sign the message on behalf of the final destination. The second solution involves the addition of the signature that can be used by intermediate nodes to reply to a route request by the node that originally created the route request.

In SAODV, route error messages (RERR) that are generated by nodes that inform their neighbors that they are not going to be able to route messages to specific destinations are secured using digital signatures. A node that generates or forwards a route error message cryptographically signs the whole message, except the destination sequence numbers (Fig. 6).

Since the destination does not authenticate the destination sequence number, the authors suggest that a node should never update the destination sequence numbers of the entries in its routing table based on route error messages [33]. Route error messages are still useful in SAODV in order to allow a node to decide whether it should completely remove a route from its routing table or not.

SAODV is a set of security extensions to the AODV protocol. In SAODV every route discovery that is initiated by a node corresponds to a new one-way hash chain. The elements of the chain are used in order to secure the metric field in the route request packets.

Secure Link State Routing Protocol (SLSP) — The Secure Link State Routing Protocol (SLSP) [34] has been proposed to provide secure proactive routing for mobile ad hoc networks. It secures the discovery and the distribution of link state information both for locally and network-wide scoped topologies. SLSP can be employed as a stand-alone solution for proactive link-state routing, or combined with a reactive ad hoc routing protocol creating a hybrid framework. The main operational requirement of SLSP is the existence of an asymmetric key pair for every network interface of a node. Participating nodes are identified by the IP addresses of their interfaces. The specific mechanism for the certification of public keys is not addressed by the protocol, as previously proposed key management solutions are assumed to be in operation. Furthermore, SLSP limits its scope to secure only

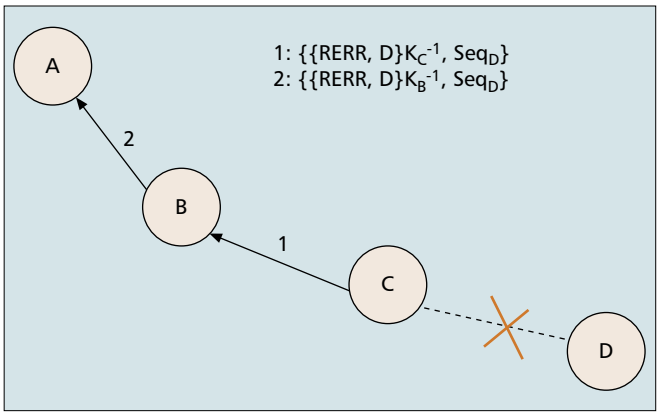

Figure 6. *Route maintenance in the SAODV protocol.*

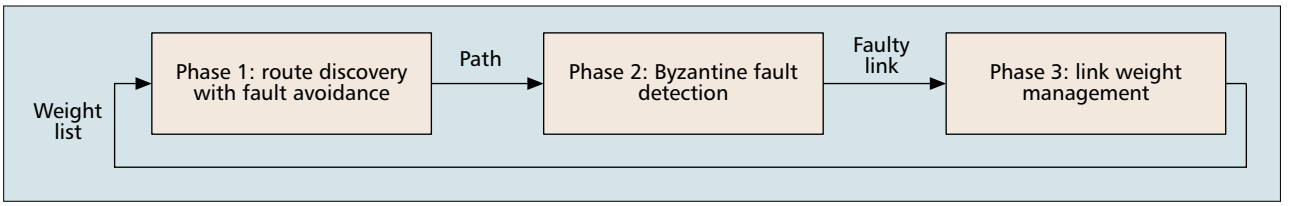

Figure 7. *The three phases of the protocol operate in sequence and each one receives the output of the previous as input.*

the process of topology discovery; parties that participate in it and decide to misbehave during data transmission are not detected or penalized.

SLSP can be logically divided into three components: *public key distribution*, *neighbor discovery*, and *link state updates*. To avoid the need for a central key management server, nodes broadcast their public key certificates within their zone using signed public key distribution (PKD) packets. Receiving nodes are then able to verify subsequent SLSP packets from the source node. Link state information is also broadcast periodically using the Neighbor Lookup Protocol (NLP), an internal part of SLSP. NLP *hello* messages are also signed and include the sending node's MAC address and IP address for the current network interface. This allows a node's neighbors to maintain a mapping of MAC and IP addresses. By generating *notification* messages, NLP can inform SLSP when suspicious discrepancies are observed, such as two different IP addresses having the same MAC, or a node trying to claim the MAC of the current node, etc. Such notifications are used to inform SLSP to discard the suspicious packets. Link state update (LSU) packets are identified by the IP address of the initiating node and include a 32-bit sequence number for providing updates [34]. The hop count included in the packet is authenticated using hash chains, as we have previously seen in the SAODV and other protocols. The authentication of the hash chain itself is performed through the anchor that is included in the digitally signed part of an LSU message. Nodes that receive an LSU verify the attached signature using a public key they have previously cached in the pubic key distribution phase of the protocol. The *hops_traversed* field of the LSU is set to hashed hops_traversed, the TTL is decremented, and finally the packet is broadcast again. To protect against denial of service attacks, SLSP nodes maintain a priority ranking of their neighboring nodes based on the rate of control traffic they have observed. High priority is given to nodes that generate LSU packets with the lowest rate. This functionality enables the neighbors of malicious nodes that flood control packets at very high rates to limit the effectiveness of the attack.

SLSP provides a proactive secure link state routing solution for ad hoc networks. By securing the neighbor discovery process and using NLP as a method to detect discrepancies between IP and MAC addresses, SLSP offers protection against individual malicious nodes. As mentioned by the authors, SLSP is vulnerable to colluding attackers that fabricate non-existing links between themselves and flood this information to their neighboring nodes.

Reputation-Based Solutions

Several reputation mechanisms have been proposed to address the problem of selfish behavior and disruption of the routing process in ad hoc networks. The main goal of reputation systems is to make decisions regarding trustworthy entities and to encourage behavior that leads to increasing trust [35]. Their operation usually relies on passive monitoring of transactions and exchange of recommendation or alarm messages between entities that participate in a system. In this section we present three systems that use reputation mechanisms to mitigate malicious behavior in ad hoc routing. The first system, OSRP, employs both reputation and cryptographic operations, while CONFIDANT and the Watchdog and Pathrater schemes avoid any use of cryprography.

On-demand Secure Routing Protocol Resilient to Byzantine Failures (OSRP) — The problem of malicious nodes in an ad hoc network performing byzantine attacks in order to disrupt the routing function is studied in [18]. The authors propose an on-demand secure routing protocol that is able to function in the presence of colluding nodes introducing byzantine failures in the process of routing. Their approach is based on the detection of faulty links after *log n* faults have occurred, where *n* is the length of the route. The protocol bases on-demand route discovery on weight values of paths, and the paths that are identified as malicious are assigned increased weights. The authors define the term *byzantine behavior* as any action taken by an authenticated node that disrupts the routing process. The utilized detection method avoids the identification of nodes as malicious, but instead tries to attribute a flaw to a link between two nodes.

The protocol is separated into three different phases: *route discovery with fault avoidance*, *byzantine fault detection,* and *link weight management*. The phases operate in sequence and each one receives the output of the previous as input (Fig. 7).

The metric upon which path selection is based consists of link weights, where high weights represent an unreliable path. Every node that participates in the network is required to maintain a *weight list* and update it according to the results of the fault detection phase.

The first phase of the protocol is responsible for establishing a route between the initiating and the destination node. The initiating node signs with its private key a route request message that is broadcast to all of its neighbors. The message includes the address of the initiator, the address of the destination, a sequence number, and a weight list. When an intermediate node receives a route request it checks if a request with the same identifiers has been seen before. If such a request does not exist in its list, it verifies the signature of the initiator, adds the request to its list, and rebroadcasts it. Upon receiving a request the destination node checks the validity of the signature and creates a signed route response message. The response contains the source and destination addresses, a sequence number, and the weight list from the request message. The destination node broadcasts the response to its neighboring nodes. Intermediate nodes compute the total weight of the path by summing the weight of all the links on the specified path to the current node [18]. If the total computed weight is less than that of any previous response message with the same identifiers, the current node verifies all the signatures, appends its own identifier, signs it, and broadcasts it. The initiating node performs the same process as the intermediate nodes upon receiving a route response. The initiator updates the route to the destination if a received path is better than the one already used.

The second phase of the protocol, byzantine fault detection, requires specific nodes on a discovered path to return Acknowledgments to the source node. Data packets originating from the source contain a list of nodes, known as *probe*

nodes, which are required to send Acknowledgments for every received packet. If the number of unacknowledged packets violates an acceptable threshold, a fault is registered on the path. Thus, a malicious node is not able to drop packets without actually dropping the list of the probe nodes. The list contains non-overlapping intervals that cover a route, where each interval covers the sub-path between two consecutive nodes [18]. Using binary search, the fault detection algorithm is able to locate a faulty link after *log n* faults have been detected, where *n* is the length of the route where a fault was registered. In order to avoid expensive asymmetric operations on a per-packet basis, the protocol requires the existence of shared keys between the source node and each probe node for ensuring the authenticity and integrity of the Acknowledgments.

The third and final phase of the protocol manages the weights of the links that were identified as faulty by the previous phase. When a link is identified as faulty the corresponding weight value is doubled. The protocol maintains counters associated with each link and when this counter reaches zero the weight of the associated link is halved.

The main goal of the protocol is to provide a robust on-demand ad hoc routing service that is resilient to byzantine failures. The operation of the protocol requires the existence of public-key infrastructure in the ad hoc network to certify the authenticity of the participating nodes' public-keys. Based on this assumption, the protocol manages to discover a fault-free path if one exists even in an environment with colluding malicious nodes. As the authors note, a limitation rests in the inability of the protocol to prevent wormhole attacks. However, if the wormhole link demonstrates byzantine behavior then the protocol will detect it and avoid it [18].

Watchdog and Pathrater — The *watchdog* and *pathrater* scheme consists of two extensions to the DSR routing protocol that attempt to detect and mitigate the effects of nodes that do not forward packets although they have agreed to do so [1]. This misbehavior may be due to malicious or selfish intent, or simply the result of resource overload. Although the specific methods proposed build on top of DSR, the authors suggest that the basic concepts can be applied to other source routing protocols for ad hoc networks. The watchdog extension is responsible for monitoring that the next node in the path forwards data packets by listening in promiscuous mode. It identifies as misbehavior nodes those nodes that fail to do so. The pathrater assesses the results of the watchdog and selects the most reliable path for packet delivery. One of the base assumptions of this scheme is that malicious nodes do not collude in order to circumvent it and perform sophisticated attacks against the routing protocol.

Every node that participates in the ad hoc network employs the watchdog functionality in order to verify that its neighbors correctly forward packets. When a node transmits a packet to the next node in the path, it tries to promiscuously listen if the next node will also transmit it. Furthermore, if there is no link encryption utilized in the network, the listening node can also verify that the next node did not modify the packet before transmitting it [1]. The watchdog of a node maintains copies of recently forwarded packets and compares them with the packet transmissions overheard by the neighboring nodes. Positive comparisons result in the deletion of the buffered packet and the freeing of the related memory. If a node that was supposed to forward a packet fails to do so within a certain timeout period, the watchdog of an overhearing node increments a failure rating for the specific node. This effectively means that every node in the ad hoc network maintains a rating assessing the reliability of every other node from which it can overhear packet transmissions. A node is identified as misbehaving when the failure rating exceeds a certain threshold bandwidth [1]. The source node of the route that contains the offending node is notified by a message sent by the identifying watchdog. As the authors of the scheme note, the main problem with this approach is its vulnerability to blackmail attacks.

The pathrater extension to DSR selects routes for packet forwarding based on the reliability rating assigned by the watchdog mechanism. Specifically, a metric for each path is calculated by the pathrater by averaging the reliability ratings of the nodes that participate in the path. This path metric allows the pathrater to compare the reliability of the available paths, or to emulate the shortest path algorithm when no reliability ratings have been collected [1]. The pathrater selects the path with the highest metric when there are multiple paths for the same destination node. The algorithm followed by the pathrater mechanism initially assigns a rating of 1.0 to itself and 0.5 to each node that it knows through the route discovery function. The nodes that participate on the active paths have their ratings increased by 0.01 at periodic intervals of 200 milliseconds to a maximum rating of 0.8. A rating is decremented by 0.05 when a link breakage is detected during the packet forwarding process to a minimum of 0.0. The rating of –100 is assigned by the watchdog to nodes that have been identified as misbehaving. When the pathrater calculates a path value as negative this means that the specific path has a participating misbehaving node. The authors suggest that negative node ratings should be slowly incremented in order to avoid permanent isolation of nodes that suffer from malfunctions or overloads, but such a mechanism has not been implemented.

The watchdog and pathrater extensions facilitate the identification and avoidance of misbehaving nodes that participate in the routing function. The identification is based on overheard transmissions, and the selection of reliable routes is based on the calculated reliability of the paths. Based on simulations performed by the authors, the system is able to increase the throughput by 17 percent in the presence of 40 percent misbehaving nodes, by increasing the percentage of the overhead transmissions from the standard 9 percent of DSR to 17 percent [1]. The main operational assumption, besides the support of promiscuous mode by the participating nodes, is that there is no collusion between active attackers in the network. Since the system avoids the utilization of cryptographic methods for securing exchanged messages, it suffers from the possibility of blackmail attacks.

CONFIDANT — The CONFIDANT (Cooperation Of Nodes: Fairness In Dynamic Ad hoc NeTworks) protocol consists of a set of extensions to DSR that include the following components: the *monitor*, the *reputation system*, the *path manager,* and the *trust manager* [36]. A node that participates in the protocol must operate all four components. Routing paths are chosen based on ratings assigned through directly observed or reported routing and forwarding behavior.

The monitor component of a CONFIDANT node is responsible for monitoring *passive acknowledgments* for each packet it forwards. This is similar to the watchdog functionality that we discussed in the previous paragraph. When a node forwards a packet it monitors the transmissions of its next hop neighbors trying to detect deviations from the expected normal behavior. The trust manager component deals with the sending and receiving of *alarm* messages [36]. These messages are generated and sent when the local node concludes that another node is misbehaving. Such messages are exchanged between nodes that are pre-defined as *friends*. Alarms from other nodes are given substantially less weight. The conclusion

is reached based on the passive Acknowledgments mechanism of the monitor component, or a received alarm message from another node. The reputation system component maintains a table of node identities and the associated ratings. Ratings are modified according to a *rate function* that uses small weights for reported alarms of malicious behavior and greater weights for direct observations. If a rating falls under a certain threshold the path manager component is called in order to remove the path containing the identified malicious node. Furthermore, the path manager ignores routing packets from the attacker and alerts (or ignores, this is a configuration setting) legitimate nodes when they request a route that uses a compromised path.

It is important to note that the CONFIDANT protocol only supports the building of negative experiences associated with a node identity. Each entry in the list of identified attackers maintained by a node is associated with a timer. When this expires the entry is purged and the node is again considered to be a legitimate participant of the ad hoc network.

Add-ons to Existing Protocols

This section presents add-on mechanisms that address specific security problems in ad hoc routing or techniques and extensions to existing approaches. Although most do not constitute complete protocols, in several cases their authors have used them to build secure versions of existing protocols such as AODV and DSR in order to demonstrate their suggestions. Moreover, we have included in this section an analysis of IPsec that has been suggested in the literature as a possibility for securing ad hoc routing.

Security-aware Ad hoc Routing (SAR) — Security-aware Ad hoc Routing (SAR), described in [37], is an approach to ad hoc routing that introduces a security metric in the route discovery and maintenance operations, treating secure routing as a quality of service (QoS) issue. While traditional non-secure routing protocols utilize distance (measured in hop counts), location, power, and other metrics for routing decisions, SAR uses security attributes (such as trust values and trust relationships) in order to define a routing metric. Its operation is applicable in situations where a route that satisfies certain security requirements is more important than a route that satisfies any other requirement.

SAR extends on-demand ad hoc routing protocols (such as AODV or DSR) in order to incorporate the security metric into the route request messages. The authors present an implementation of SAR based on AODV, which they call SAODV (Security-aware AODV). The initiator broadcasts a route request (RREQ) with an additional field (RQ_SEC_REQUIREMENT) that indicates the required security level of the route that she wishes to discover [37]. A neighboring node that receives the packet checks whether it can satisfy the security requirement. If the node can provide the required security then it can participate in the requested route and re-broadcasts the packet to its own neighbors, setting a new field called RQ_SEC_GURANTEE to indicate the maximum level of security it can provide. If a node is not secure enough to participate in the requested route, it simply drops the RREQ. Therefore, when the destination node receives the RREQ it can be sure that a route to the source node exists and that this route satisfies the security requirements defined by the initiator. The destination sends a route reply (RREP) packet with an additional field (RP_SEC_GUARANTEE) that indicates the maximum level of security of the found route. The RREP message travels back along the reverse path of the intermediate nodes that were allowed to participate in the routing, and each node updates its routing table according to the AODV specification, including the RP_SEC_GUARANTEE value. This value is used in order to allow intermediate nodes with cached routes to reply to a request of a route with a specific security requirement.

The security metric of SAR can be specified by hierarchies of trust levels or by desirable security properties. In order to define trust levels, a key distribution or secret sharing mechanism is required. By utilizing this mechanism all the nodes that belong to a particular trust level can share a key. Therefore, nodes of different security levels cannot decrypt or process routing packets and are forced to drop them. Furthermore, the security metric can be specified by standard security properties such as timeliness, ordering, and authenticity, to name a few [37]. These properties can be implemented in the SAR protocol by utilizing techniques such as timestamps, sequence numbers, and certificates, respectively. However, each of these properties has a related cost and adds performance overhead to the routing process. Participating nodes can specify their exact security requirements based on security-performance trade-off decisions.

The main idea behind SAR is the utilization of a security metric in place of the standard metrics, such as hop count, for the route discovery and maintenance functions. The security routing metric is defined through attributes that reflect certain security properties, such as authentication, non-repudiation, and others. Therefore, the discovered and maintained routes satisfy the requirements of the security metric.

Techniques for Intrusion Resistant Ad hoc Routing Algorithms (TIARA) — Techniques for Intrusion Resistant Ad hoc Routing Algorithms (TIARA) is a set of design techniques that can be applied on ad hoc routing protocols to mitigate the impact of malicious nodes and allow the acceptable operation of the network under denial of service attacks [38]. The design principles defined by TIARA can be incorporated more easily into on-demand routing protocols, such as DSR and AODV, and are enumerated here: flow-based route access control (FRAC), *multi-path routing, source-initiated flow routing*, *flow monitoring*, *fast authentication*, *the use of sequence numbers and referral-based resource allocation*.

A flow is defined by TIARA as a sequence of packets that travel from a source node to a destination node. The flow-based route access control technique facilitates an access control list that contains authorized flows at each node that participates in the ad hoc network. Based on this list the node drops packets that belong to unauthorized flows or forwards packets from an authorized flow. In order to incorporate the FRAC mechanism to existing protocols, the routing tables maintained by each node must store flow identifiers, and the forwarding decisions must be based on these identifiers. *Multi-path routing* is the second design technique proposed by TIARA and enforces the discovery and maintenance of all routing paths for a specific flow. Existing on-demand ad hoc routing protocols can be modified more easily than table-driven protocols in order to integrate multi-path routing. This technique assures that an ad hoc network will be able to tolerate failures induced by intrusions on specific paths. *Source-initiated flow routing* is a design technique that complements the existence of multiple routing paths between two nodes. The source node that wishes to send data packets through a specific path adds a label to every packet that indicates this path. Intermediate nodes forward packets to their neighbors based on the information included in the path label. Another design principle of TIARA is the mechanism of *flow monitoring*. The source node periodically transmits *flow status* packets to the

destination node of a particular flow. These messages include sequence numbers to counter replay attacks, are encrypted to ensure confidentiality, and are signed to protect integrity. The destination node monitors the active flows in which it participates, and keeps track of the packets received between flow status messages. A path failure is signaled if the destination node does not receive a flow status message for a specified time interval, or if the number of packets it received is above or below a threshold fraction of the packets sent by the source node. In order to authenticate packets TIARA defines a lightweight authentication mechanism called *fast authentication*. A node that utilizes this mechanism places the path label at a secret location within each packet it transmits. This secret location, different for each node, is made known to the participating nodes with the route establishment function of the employed routing algorithm [38]. To successfully protect against replay attacks TIARA uses sequence numbers. Using a technique similar to fast authentication, the source node places a sequence number in a specific secret location of every data packet it transmits according to the intermediate nodes between itself and the destination. Finally, the referral-based resource allocation mechanism defines the maximum amount of network resources that each routing node will allocate for a particular flow. A node allows the use of additional resources if the source of a flow can present valid recommendations from trusted nodes that guarantee the authenticity of the request.

TIARA provides general design principles and techniques that can be applied to existing ad hoc routing protocols to develop solutions resistant to denial of service attacks. The techniques provided by TIARA are protocol independent, but they require extensive changes to existing protocols in order to be successfully incorporated.

Building Secure Routing out of an Incomplete Set of Security Associations (BISS) — The protocols we have analyzed up to this point assume that a security association already exists between the initiator and the destination node, as with SRP, or that both the initiator and the destination must have established security associations with all the intermediate nodes on the routing path, as with Ariadne. The BISS protocol (Building Secure Routing out of an Incomplete Set of Security Associations) [39] is a set of optimizations to existing ad hoc routing protocols that have been designed with the assumption that participating nodes have established an *incomplete* set of security associations between themselves. The authentication of the intermediate nodes along a route discovery path is not performed only on the basis of pre-established associations, but also by exchanging public key certificates with these nodes. BISS assumes that the target node of a route discovery process has an existing security association with the intermediate nodes and that an off-line trusted authority has certified the public keys of all the participating nodes.

Although the general ideas introduced by BISS can be applied to on-demand routing protocols, the authors have applied them to the DSR protocol. Route request packets are signed by the initiator and also include its public key and certificate. The certificate is signed by the trusted authority and binds the initiator's public key with an identifier, such as the node's address. Intermediate nodes that receive route request packets verify the initiator's signature and authenticate the destination through the pre-established security association. The message is broadcast further if both the initiator and the destination are authenticated correctly and the intermediate node has not seen this particular route request packet before. Similarly to Ariadne, the re-broadcast packet includes a keyed hash calculated over the packet and the security association that the intermediate node shares with the target. When the request reaches the target node the authenticity of the included routes is verified using the security associations and a route is chosen. Using this route, a route reply packet is sent that includes the selected nodes. If the target shares a secret with the initiator, the reply is protected by calculating and attaching a keyed hash. Otherwise, the target signs the reply to allow the initiator to authenticate it. Nodes that are on the route reply path behave in a similar way. Specifically, if an intermediate node has a security association established with the initiator of the request, it calculates and attaches a keyed hash of the packet along with its identifier. If it does not share a secret key with the initiator, it signs the reply and attaches the signature and its public key certificate. The initiator authenticates the route reply by verifying the hashes and the signatures it includes. If all of them are verified correctly a route is established, otherwise the reply is discarded. Route maintenance in BISS also follows the same method. A node that is not able to forward a packet along a specific route sends a route error message to the source of the packet. The route error is authenticated by a keyed hash if the nodes share a secret key or by a digital signature and the corresponding public key certificate if they do not.

The approach followed by BISS has the beneficial side-effect of increasing the number of security associations in an ad hoc network. The keys and certificates of previously unknown nodes are distributed in the network during the route discovery and allow nodes to establish symmetric shared secrets for using the keyed hash authentication method for future message verifications. The simulations performed by the authors have shown that all participating nodes in an ad hoc network that uses a routing protocol with the BISS extensions can route securely, assuming as little as 30 percent of the security associations are pre-established and provided that the node density is sufficiently high [40].

Packet Leashes — Packet leashes [32] are not a complete protocol but a specific solution than can be used in an existing protocol to protect against wormhole attacks. The main idea of the solution is to add some extra information to each packet sent in order to allow a receiving node to determine if a packet has traversed an unrealistic distance. The authors have proposed two kinds of leashes: *temporal* and *geographical*.

According to the temporal leashes scheme, a node adds an extremely precise timestamp to each outgoing packet. The receiver is then able to authenticate the traveled distance given the time taken and the fact that this distance is bounded by the speed of light. As is obvious, the temporal leashes solution requires extremely precise clock synchronization, in the order of hundreds of nanoseconds, between all participating nodes. In order to deal with the uncertainty associated with the transmission times of highly congested nodes, the authors propose the use of a threshold time synchronization error.

The second method of constructing packet leashes is with the use of geographical location information, provided by systems such as the Global Positioning System (GPS) [40], and loosely synchronized clocks. A timestamp and the location information of the sender are added to each outgoing packet. The receiver is then able to verify the distance traveled by the packet during the last hop. All the nodes of the ad hoc network must have appropriate hardware to track their location according to a unified scheme. Clock synchronization in this method does not need to be as precise as with the temporal method since the location information is also used in the calculation of the distance between the sender and receiver.

In general, packet leashes provide a complete solution to

the problem of wormholes in mobile ad hoc networks. Their operational requirement is either extremely precise clock synchronization, or less rigidly synchronized clocks and the knowledge of geographical location.

IP-level Security (IPsec) — Several authors have proposed the use of IPsec as the underlying security mechanism for providing authentication, integrity, and confidentiality in mobile ad hoc networks [12, 19, 41]. According to this approach the operation of the routing protocol relies for protection on the security infrastructure provided by the IPsec suite.

IPsec consists of a set of protocols that provide security services at the Internet Protocol (IP) level. These protocols guarantee the secure transmission of data between two systems anywhere in a networked environment. The goal of IPsec is to provide integrity, confidentiality, and authenticity. Moreover, it should be as resistant as possible to traffic analysis, replay, and man-in-the-middle attacks. The IPsec protocol suite consists of three different protocols [42]. First, the encapsulating security payload (ESP) is added to an IP datagram and provides confidentiality, integrity, and authenticity of the transferred data. The authentication header (AH) is also added to an IP datagram and provides integrity and authenticity of the transmitted packets. AH does not provide confidentiality for the data of network packets since this is the service explicitly provided by ESP. The third protocol is the internet key exchange (IKE), which is the protocol that negotiates the security association between the two endpoints that need to communicate, exchanges the necessary cryptographic keys, and sets up the connection configuration parameters. A security installation based on IPsec requires either the existence of prearranged common secrets between each pair of systems that need to communicate, or an online trusted third party, e.g. a certification authority, in order to certify the validity of the signed Diffie-Hellman key exchange messages and guarantee the identity of the communicating end points.

Unfortunately, neither of the above requirements can be realistically assumed in an ad hoc network. Furthermore, the approach of using IPsec as an underlying security solution has been criticized for producing additional configuration overhead [43]. Another consideration is that when a security solution is not designed concurrently with the basic protocol, but is applied afterward, it may leave unpredictable and undetectable vulnerabilities in the system. This is especially true in the case of IPsec as a retrofitted security solution, whose high level of complexity and lack of documentation hinders attempts at in-depth analysis [44]. Furthermore, even if IPsec can be employed to protect a routing protocol from external fabricated unauthorized traffic, it cannot guarantee correct operation under internal attacks [15].

COMPARISON

This section attempts a comparison of the previously presented secure ad hoc routing protocols. Each protocol has a different set of operational requirements and provides protection against different attacks by utilizing particular approaches. Therefore, a detailed comparison can provide insight regarding the applicability of a particular protocol for a specific application domain. We present the assumptions and operational requirements of the analyzed protocols, and compare them based on their utilized ad hoc routing approaches. A security analysis is attempted focusing on the applicability of the previously described solutions.

REQUIREMENTS AND ASSUMPTIONS

The surveyed protocols base their proposed solutions to the problem of secure ad hoc routing on certain assumptions and operational requirements. Table 1 summarizes the results of the comparison regarding this aspect and forms a basis for the discussion in this section.

As is obvious from the comparison, most of the protocols require the existence of an online trusted third party, e.g. a certification authority, in order to facilitate the acquisition and verification of the public keys of the nodes that participate in the ad hoc network. The protocols that fall into this category are the ARAN, SAR, SEAD, SAODV, the set of design principles specified by TIARA, the On-demand Secure Routing Protocol Resilient to Byzantine Failures, and the protective solutions based on IPsec. Moreover, the On-demand Secure Routing Protocol Resilient to Byzantine Failures requires shared keys between the source node for a route discovery and each probe node on the path used for acknowledging received packets. Alternatively, SAR and IPsec can utilize the existence of prearranged shared secrets between each pair of nodes. The operational requirement of SRP is similar since it needs a pre-established security association between every source and destination node. The SEAD protocol requires the existence of a key distribution scheme for the authentication of one element of a hash chain between two nodes. This can be realized with a broadcast authentication mechanism such as TESLA, which requires the nodes of the network to have synchronized clocks. Ariadne requires both shared secret keys between each pair of nodes to authenticate point-to-point messages, and time synchronization in order to use TESLA as a method for authenticating broadcast messages. Finally, the successful operation of the Watchdog and Pathrater protocol extensions require that no two or more malicious nodes collude in order to perform routing attacks.

AD HOC ROUTING PARAMETERS

This section summarizes the routing approaches utilized by the presented protocols. Most of the security solutions for ad hoc routing are based on existing ad hoc routing protocols. These underlying protocols introduce parameters that must be taken into account. The complete set of these parameters is presented in Table 2.

Most of the surveyed protocols employ the on-demand approach to the ad hoc routing problem. This choice is primarily based on the performance behavior of the on-demand approach in high mobility scenarios. It has been shown that on-demand ad hoc routing protocols, such as AODV and DSR, have a higher delivery rate than proactive solutions since they transmit routing messages only when data packets need to be sent or when there are topology changes in the network [4]. Loop freedom specifies whether the protocol is able to prevent or detect the formation of routing loops; this is usually achieved with the use of sequence numbers. The next comparison parameter is the routing metric used by each protocol. Distance (usually measured in hop counts) is the most commonly used routing metric, since most of the presented security solutions rely on existing ad hoc routing protocols. Two notable exceptions are the protocols SAR and OSRP. SAR utilizes a security requirement, e.g. trust levels, as the metric for establishing routing paths. OSRP utilizes a link weight as the metric for selecting routes. The link weight represents the reliability of the corresponding link. A similar approach is taken by the Watchdog and Pathrater protocol that selects paths according to a calculated path reliability metric. However, if the Pathrater has no available reliability information for the selection of the path, it tries to emulate a

Proposed solution	Requirements
ARAN	Online trusted certification authority. Each node knows a priori the public key of the CA.
SAR	Key distribution or secret sharing mechanism.
SRP	Existence of a security association between each source and destination node. Malicious nodes do not collude within one step of the protocol process.
SEAD	Clock synchronization, or a shared secret between each pair of nodes.
Ariadne	Clock synchronization and the existence of a shared secret between each pair of nodes. Also, an authentic TESLA key for each node in the network and an authentic route discovery chain element for each node for which this node will forward route requests. TESLA keys are distributed to the participating nodes via an online key distribution center.
SAODV	Online key management scheme for the acquisition and verification of public keys.
TIARA	Online public key infrastructure.
On-demand Secure Routing Protocol Resilient to Byzantine Failures	Online public key infrastructure and shared symmetric keys between source and probe nodes.
SLSP	Nodes must have their public keys certified by a TTP. No collusion between malicious nodes.
BISS	The target node of a route discovery must share secret keys with all the intermediate nodes. An off-line trusted authority has certified the public keys of all the participating nodes.
Watchdog and Pathrater	No collusion between malicious nodes.
CONFIDANT	Nodes cannot change their identifier to get rid of their reputation rating. Pre-defined lists of friendly nodes.
Packet leashes: temporal	Extremely precise clock synchronization.
Packet leashes: geographical	Geographical location information and loosely synchronized clocks.
IPsec	Prearranged common secrets between each pair of nodes, or an online trusted third party.

■ Table 1. *Operational requirements of the surveyed secure ad hoc routing solutions.*

shortest path selection algorithm. The last two comparison parameters demonstrate the trade-off between performance and security that the investigated security solutions considered. An important aspect of any routing protocol lies in its ability to identify the shortest path between the two endpoints that try to communicate. However, the identification of the shortest path is not always possible when there are security considerations involved in the route establishment process. For example, the ARAN protocol provides an optional mechanism for finding the shortest path between two nodes using computationally expensive asymmetric operations that add even more to the already high computation overhead of the solution. Furthermore, the optimization of allowing intermediate nodes with fresh routes to reply to route discoveries can greatly benefit the performance overhead of an ad hoc routing protocol, but constitutes a security vulnerability if it is designed poorly. The SRP and SAODV protocols provide optional mechanisms for securing this process.

Security Analysis

In this section we present a security analysis regarding the behavior of the surveyed protocols and their applicability in mobile ad hoc environments. Ideally, a secure ad hoc routing protocol should be able to provide protection against all the categories of attacks previously mentioned. In reality, given the highly dynamic nature of ad hoc networks and the different scenarios of their application, for example utilizing some infrastructure or being completely infrastructureless, it is difficult to design a general solution that can provide adequate protection against all kinds of attacks in all possible application scenarios, with acceptable requirements and overhead. Table 3 provides a comparison of the surveyed secure routing solutions with respect to the different attacks.

Route Discovery — The problem of securing the process of route discovery has been approached differently by the studied protocols. The basic requirement for secure route discovery in on-demand protocols is that the destination of a *route request* packet must be able to authenticate the path, or paths, included in the packet in order to utilize legitimate paths and not those that are fabricated by malicious nodes for sending a *route reply*. Accordingly, the initiator must be able to authenticate all the nodes that are included in the received reply.

Ariadne uses *per-hop hashing* to verify that no node was removed from a request by using one-way hash functions. As we have seen, the authentication is performed by using the released TESLA key. ARAN, which also works in an on-demand mode, assumes that each node has a certificate issued by a universally trusted third party (TTP) that binds its IP address with its public key. Route discovery packets are broadcast and each node checks the signature of all previous

Proposed solution	Routing approach	Loop freedom	Routing metric	Shortest path Identification	Intermediate nodes allowed to reply to route requests
ARAN	On-demand	Yes	None	Optional	No
SAR	On-demand	Depends on the selected security requirement	A security requirement	No	No
SRP	On-demand	Yes	Distance	No	Optional
SEAD	Table-driven	Yes	Distance	No	No
Ariadne	On-demand	Yes	Distance	No	No
SAODV	On-demand	Yes	Distance	No	Optional
TIARA	On-demand[1]	Depends on the basis protocol	Distance	Depends on the basis protocol	Depends on the basis protocol
OSRP[2]	On-demand	Yes	Path reliability	No	No
SLSP	Table-driven	Yes	Distance	No	No
BISS	On-demand	Yes	Distance	No	No
Watchdog and Pathrater	On-demand	Yes	Path reliability or distance[3]	Depends[4]	Yes
CONFIDANT	On-demand	Yes	Path reliability	Depends[4]	Yes
Packet leashes	NA[5]	NA[5]	NA[5]	NA[5]	NA[5]
IPsec	NA[5]	NA[5]	NA[5]	NA[5]	NA[5]

[1] Can also be applied on table-driven protocols, but this requires extensive modifications.
[2] On-demand Secure Routing Protocol Resilient to Byzantine Failures.
[3] The routing metric is distance if no reliability information has been collected.
[4] On whether reliability information has been collected for the path in question.
[5] Depends on the utilized underlying ad hoc routing protocol.

■ Table 2. *Ad hoc routing parameters.*

nodes, removes the last forwarder's signature and certificate, signs it with its own private key, and attaches its own certificate. The target node replies with a route reply packet that is unicast back to the initiator using the same method. We have identified two problems with ARAN. The first concerns mobility and address reconfiguration. As the node's owner moves across different authority domains and networks the node's address changes.

Moreover, frequent partitions are common in ad hoc networks, and address reconfiguration schemes, e.g. the one presented in [45], change the address associated with a node in order to handle these. Each time a node's address changes a new certificate must be issued by the TTP, which therefore needs to be not only constantly online but also reachable. If we assume that a new certificate can be issued, all the previous established routes of the node are invalidated since the new certificate contains a new address, complicating the process of handoff. The other problem we have identified with ARAN concerns possible denial of service attacks. An attacker can flood the network with fabricated route request or route reply packets signed with a bogus key. Legitimate nodes will try to verify the signatures of all these packets, spending valuable computational resources and discarding real routing traffic in case they cannot perform public key operations fast enough.

A hybrid approach of securing route discovery is followed by SAODV. The hop count metric is authenticated using a hash chain, a method also used by SEAD, and the other fields of the route request and route reply packets are authenticated by employing a digital signature scheme. Thus, the same denial of service attack that can be performed against ARAN can also disrupt the operation of SEAD.

An important design decision that all secure on-demand protocols have to make in the process of route discovery concerns replies to route requests by intermediate nodes. A protocol's performance is greatly enhanced if intermediate nodes with fresh routes are allowed to reply to a route query. However, this can lead to an opportunity for an attacker to disrupt the operation of the protocol. Therefore, if such a design choice is followed it has to be carefully secured.

Route Maintenance — One of the most interesting aspects for comparing the surveyed secure ad hoc routing protocols is the process of route maintenance. This function plays an important role in all ad hoc routing solutions since it is responsible for detecting topology changes and informing the

	Protocols						
Attacks	**ARAN**	**SRP**	**SEAD**	**Ariadne**	**SAODV**	**SLSP**	**OSRP**
Location disclosure	No	No	No	No	No	No	No
Black hole	No	No	No	No	No	No	Yes[4]
Replay	Yes	Yes	Yes	Yes[5]	Yes	Yes	Yes
Wormhole	No	No	No	No	No	No	No[7]
Blackmail	NA	NA	NA	NA	NA	NA	NA
Denial-of-service	No	Yes	Yes	Yes	No	Yes	No
Routing table poisoning	Yes	Yes	Yes	Yes	Yes	Yes	Yes
To be continued...							

Table 3. *Defense against attacks*

corresponding nodes, so that they can update the established routes. Usually route maintenance is accomplished with the use of route error messages generated by nodes that detect a broken link, and forwarded to the nodes that utilize the broken link as part of their routes. Therefore, route error messages are an especially attractive target for a malicious node. By fabricating and forwarding route error messages an attacker can try to disrupt the operation of existing routes, not only breaking connectivity but also creating additional routing overhead in the network as a result of legitimate nodes trying to establish alternative paths. In Table 4 we present the route maintenance characteristics of the protocols we have analyzed.

The solution adopted by most of the analyzed protocols requires the signing of the complete error message by the node that generates or forwards it. Given the non-repudiation property of digitally signed messages, each node can be held responsible for the route error packets it generated. A protocol that specifies such an approach is SAODV. However, SAODV avoids the signing of destination sequence numbers in route error messages used to specify the freshness of a route and avoid loops, since that would introduce an important additional overhead. Consequently, the authors of SAODV do not allow the update of destination sequence numbers based on route error packets [33]. Another approach to the same problem is taken by SRP. Route error packets are source-routed along the prefix of the path that is being reported as broken and the receiver compares the traversed path to the prefix of the corresponding route [29]. However, since SRP avoids the use of asymmetric cryptography for signing route signaling packets due to the high associated overhead, it cannot verify the legitimacy of route error messages. This allows malicious nodes to disrupt existing routes by generating fake route error packets.

Clock Synchronization — Clock synchronization is a common requirement among several of the protocols we have investigated. Although the precision constraints differ from extremely strict to loosely coordinated, they are nonetheless essential for the operation of the SEAD, Ariadne, and Packet Leashes proposals. Mobile ad hoc networks are by definition heterogeneous computing environments. Users join and leave the network using many different hardware platforms, such as laptops, handhelds, PDAs, etc. Highly accurate clock synchronization requirements can only be satisfied through the use of specialized hardware, such as GPS. However, the heterogeneity of ad hoc networks makes the adoption of a single technology unlikely.

Another problem with solutions relying on timestamps is the calculation of appropriate thresholds in the highly volatile environment of ad hoc networks. Traffic congestion and frequent disconnections due to mobility introduce a certain amount of uncertainty in the computation of transmission times. Even slightly inappropriate thresholds can lead to the discarding of packets from legitimate nodes and to attack schemes where timestamps are carefully fabricated to accept spoofed packets.

Reputation-based Approaches — The on-demand secure routing protocol proposed by Awerbuch *et al.* employs Acknowledgments that must be sent by specific nodes for every packet they receive. As we have seen, a performance threshold is set and if the number of unacknowledged packets violates this, a binary search is initiated to find the offending link [18]. A possible attack against this scheme is for an attacker to find the threshold through passive analysis and selectively drop packets without violating it. If there are legitimate packet drops along the path, possibly due to node mobility, the binary search fault detection algorithm will identify them as offending the wrong link.

One of the most interesting surveyed protocol extensions is the Watchdog and Pathrater approach. These extensions to the DSR protocol attempt to provide a mechanism for secure packet forwarding in an ad hoc network with misbehaving nodes without employing any kind of cryptographic guarantees. The innovative idea of utilizing the inherent ability to listen in promiscuous mode provided by almost all network interfaces allows the authors to construct a rating system measuring the reliability of the discovered paths. However, as mentioned earlier the scheme is vulnerable to blackmail attacks, where an attacker tries to blacklist legitimate nodes from the network by fabricating misbehaving detection reports. Moreover, the Watchdog mechanism does not choose to maintain state information regarding the monitored nodes and the transmitted packets, as this would add a great deal of memory overhead. As the authors note, this efficiency decision leads to the inability of the system to detect and offer protection against replay attacks [1].

The main problem with the negative reputation system used by CONFIDANT is that in order to avoid the permanent

	Protocols						
Attacks	Watchdog and Pathrater	CONFIDANT	SAR	TIARA[1]	BISS	Packet Leashes[2]	IPsec[3]
Location disclosure	No	No	No	No	No	NA	NA
Black hole	Yes	Yes	No	Yes	No	NA	NA
Replay	No[6]	Yes	Yes	Yes	Yes	NA	NA
Wormhole	No	No	No	No	No	Yes	NA
Blackmail	No	No	NA	NA	NA	NA	NA
Denial-of-service	No	No	No	Yes	No	NA	NA
Routing table poisoning	No	No	Yes	Yes	Yes	NA	NA

[1] The reader should keep in mind that the TIARA techniques are specified as general guidelines without any technical or protocol engineering details.

[2] Packet leashes do not constitute a complete routing protocol, but a specific extension to protect against wormhole attacks.

[3] Since IPsec is not a secure ad hoc routing protocol, its defensive capabilities largely depend on the underlying routing protocol and the engineering details.

[4] An attacker can find the utilized fixed threshold through passive analysis and selectively drop packets without violating it.

[5] However, an attacker can instantly replay a received message, behaving as a repeater.

[6] If a larger memory overhead is acceptable, the scheme can offer protection against replay attacks.

[7] However, if the wormhole link demonstrates byzantine behavior then the protocol will detect it and avoid it.

Table 3. *Defense against attacks (continued).*

isolation of nodes that may have been wrongly identified as malicious, blacklist entries expire and are purged. This allows an attacking node to continue disrupting the routing protocol after its entry expires. Moreover, CONFIDANT relies on persistent node identifiers in order to associate to them reputation ratings. When a new node joins the network it is assigned a neutral rating. This allows an attacker to perform an attack, leave the network when he gets blacklisted, and rejoin it immediately. The dynamic address configuration scheme (which is an essential part of every ad hoc network) will assign him a new address without any associated negative ratings. On the other hand, positive reputation schemes also suffer from problems. Attackers may fully and unselfishly participate in the routing process for a time building up positive reputation before performing an attack. Another side effect of positive reputation schemes employed in ad hoc routing is that well-behaved nodes are chosen more frequently to participate in a route, thereby turning them into *routing bottlenecks*.

Mobility and the Establishment of Security Associations — The establishment of security associations is fundamental to the operation of all the cryptographically-based secure routing protocols we have studied. Traditional key management solutions suggest that these security associations can either be previously shared symmetric keys, pre-exchanged authentic public keys, or public key certificates issued by a universally trusted third party. However, mobile ad hoc networks present new challenges due to their non-hierarchical nature, lack of infrastructure, and mobility of the participating entities. Therefore, traditional key management and trust establishment solutions cannot be easily reused in the context of ad hoc networking. In this section we will present several key management solutions that have been specifically proposed to address the challenges of mobile ad hoc networks and discuss their behavior with respect to mobility patterns and operational requirements. Table 4 summarizes the results of our analysis.

One solution to the problem of having a single trusted third party in an ad hoc network certifying the public keys of participating nodes was presented by Zhou and Haas [23]. Instead of relying on a single certification authority (CA) or a replicated CA at different nodes, which aggravates the single point of failure/attack problem, the authors proposed the use of *threshold cryptography* to divide the private key of the CA service between n arbitrarily chosen nodes of the ad hoc network. Any $t + 1$ of these n nodes can jointly perform a signature generation operation in order to produce a public key certificate for a new node. The system is able to tolerate up to t compromised CA nodes since $t + 1$ partial signatures are required in order to produce a full valid signature. *Share refreshing* techniques that periodically create new sets of private key shares are also used. Thus, a mobile attacker has to compromise at least $t + 1$ CA nodes within the time period of two consecutive share refreshing processes in order to compromise the private key. Although this approach facilitates more flexible key management than traditional public key infrastructure (PKI) systems, it assumes that somehow certain nodes are chosen to serve the special purpose of CA share nodes. This assumption cannot always be realistically satisfied in all ad hoc networking applications. For example, in application scenarios that involve common activities instead of military or emergency activities, the selection of the CA share nodes becomes an obstacle.

A similar solution was proposed in [46] by Kong *et al.* Again the private key of the CA service is divided into a num-

Key management scheme	Application scenarios	Mobility support	Requirements and assumptions
Zhou *et al.* [23]	Military and/or emergency	Medium	Trusted authority to select the nodes that hold the CA private key shares. A mobile attacker cannot compromise $t + 1$ CA share nodes within two consecutive share refreshing processes.
Kong *et al.* [46]	Non-hierarchical	High	Trusted authority to initialize $t + 1$ nodes with private key shares. Nodes cannot change identifiers collecting private key shares.
Capkun *et al.* [47]	Non-hierarchical	High	Transitive trust between all participating nodes.
Asokan *et al.* [48]	Room meetings	Low	Human intervention is required for using a common password, given on a blackboard, to facilitate group or two-party key exchange.
Perrig *et al.* [30]	Source authentication of broadcast messages (e.g. route requests)	High	Initial TESLA keys distributed to participating nodes via an online key distribution center. Key synchronization between each sender and receiver ahead of time.
Stajano *et al.* [24, 49]	Master-slave relationships	Low	Human intervention for the establishment of out-of-band secure channels. Symmetric shared keys need to be established between all nodes that need to communicate.
Balfanz *et al.* [50]	All participants being present in the same physical space	Low	Human intervention for the establishment of out-of-band secure channels. The public keys of nodes that need to communicate have to be exchanged.
Capkun *et al.* [51]	Any that allows frequent encounters between the participants	High	Human intervention for the establishment of out-of-band secure channels. Symmetric shared keys or public keys have to be exchanged between all nodes that need to communicate.

■ Table 4. *Key management schemes for mobile ad hoc networks.*

ber of shares. The main difference with the previous scheme is that *any* participant of the ad hoc network can have a share of the private key. A node is a given a public key certificate binding its node identifier to its public key by using $t + 1$ partial signatures from the nodes that hold shares of the private key. Although this scheme still requires an authority to prime the initial $t + 1$ nodes with private key shares, it copes better than the previous scheme with high mobility scenarios since any node can be a potential CA share node. The shares are rearranged whenever a new node that wants to acquire a share of the private key obtains one by $t + 1$ nodes that already have shares. However, the scheme can be attacked by a malicious node that repeatedly changes its identifier and acquires all the necessary number of shares to reconstruct the private key of the CA service.

A PKI-based key management approach for mobile ad hoc networks has been proposed by Capkun *et al.* [47]. Their proposed solution completely avoids a universally trust third party, or a CA, by using self-signed certificates in a manner similar to the PGP web-of-trust. Based on personal real-world trust considerations, the users issue public key certificates to each other. All users have to maintain a personal repository of certificates that they have issued to others and of certificates that others have issued to them. Certificate revocation is handled *explicitly*, meaning the user informs communicating peers about the status of revoked certificates, or *implicitly* by using short expiration times and the renewal of previously issued certificates. The goal of the scheme is to enable a source node that wishes to communicate with another node to obtain the destination's authentic public key. As an example, consider the case in which source node A wishes to communicate with destination node B. Nodes A and B merge their certificate repositories and A tries to find a public key certificate chain that connects it to B. The chain has to be constructed in a way that the first certificate can be verified using A's public key, and each next certificate can be verified with the key included in the previous certificate of the chain. The last certificate of the chain must include the public key of node B. The main problem with this approach is that users are assumed to issue valid certificates. To deal with malicious participants issuing false certificates, the authors introduce confidence metrics to measure the extent that a certificate can be trusted. Furthermore, the authors assume that trust between participating entities is transitive, that is if A trusts B then B must trust A, which is not always a valid assumption [52].

Another key management approach designed to address scenarios of room meetings has been proposed by Asokan and Ginzboorg [48]. As users come into the meeting room they are given, or read from a blackboard as the authors suggest, one commonly shared password. To enhance the security of the scheme against brute force attacks, the password is not used directly, but a strong shared key is derived from it. The shared key derived from the password can either be the same for every participant (*group key exchange*) or a different one

for every exchange between two participants (*two-party key exchange*). Since this scheme has been designed to address a specific application scenario, it cannot be easily applied to different situations. In ad hoc networks where mobility is high and membership changes frequently, this key management approach becomes cumbersome.

The TESLA authentication mechanism is another approach to provide authentication of data sent by a node when all other participating nodes are not trusted. TESLA relies only on symmetric cryptography, specifically message authentication codes (MACs), and is based on delayed disclosure of keys by the sender node [50]. The main idea behind this approach is to have a sender node attach to each packet it sends a MAC calculated using a key k known only to itself. The receiver node keeps the received packet without authenticating it for a specific amount of time. Furthermore, if the packet is received too late it is also discarded by the receiver. The sender discloses the key k after a specific amount of time and the receiver is then able to authenticate the buffered packet. As a result, a single MAC operation per outgoing packet suffices to provide authentication of the sender. The main requirements are that the receiver and the sender have synchronized their clocks ahead of time, and that initial TESLA keys are distributed to participating nodes via an online key distribution center. In a previous section we saw how the Ariadne protocol uses TESLA to authenticate broadcast messages such as route requests.

The Resurrecting Duckling model was initially proposed to cover master-slave types of relationships between two nodes [24]. The relationship is established when the *master* node exchanges over an out-of-band secure channel a secret piece of information with the *slave* node. Stajano and Anderson call this procedure *imprinting*. The imprinted node is then able to authenticate the master through the common secret. The idea of bootstrapping trust relationships in adversarial environments over secure out-of-band channels was examined further by Balfanz *et al.* [50] and their concept of *location-limited channels*, which have the property that human operators can precisely control which devices are communicating with each other, thus ensuring the authenticity requirement. They propose the use of contact and infrared as a way of exchanging pre-authentication information, such as the public keys of the peers, before the actual key exchange protocol. The extended Resurrecting Duckling security model proposed the imprinting of devices with policies that define the type of relationships the slave device is allowed by its master to have with others in order to address peer-to-peer interactions [49]. However, even the extended Resurrecting Duckling system defines a static association model between the master and the imprinted nodes, limiting its direct application in situations where associations are established in an ad hoc manner.

The effect of mobility on the establishment of security associations has been specifically studied in [51]. The authors assume that security associations are formed via an out-of-band secure channel when two mobile nodes encounter each other, in a way that is similar to the Resurrecting Duckling model. The associations can be based on either the establishment of a shared symmetric key between the two nodes, or the exchange of their public keys when asymmetric cryptography is used. Various mobility models have been studied, and their effect on the rate of the establishment of security associations has been observed analytically and by simulations. The conclusions were that high mobility patterns can increase the chance that two nodes that need to communicate securely have previously established an association [51].

Use of Cryptography — All the analyzed secure ad hoc routing protocols that utilize asymmetric cryptography form the foundation of their proposed solutions on pre-existing key management schemes. The main purpose of such a scheme is to provide a reliable and direct mechanism of establishing a mapping between a node and the public key of that node. This mapping allows all the other nodes that participate in the network to verify the integrity of a transmitted message. Digital certificates have been extensively used in the security literature to provide such a mapping. An underlying assumption related to this requirement is that the address of a node is static, and therefore can be used to uniquely identify it. However, by definition a node may dynamically join or leave an ad hoc network without retaining the same address identifier. Hence, the assumption of nodes carrying certificates that verify the address to public key mappings are unrealistic in the highly dynamic application environment of mobile ad hoc networks. Furthermore, this requires constant network connectivity to a trusted certification authority, a requirement that cannot be easily satisfied in an infrastructureless network.

Another problem that exists in any system that utilizes digital certificates as a part of its security architecture is that of revocation. The issuer of certificates may deem it necessary to revoke the authority that is related to particular certificates. The highly dynamic nature and the locality of mobile ad hoc networks aggravate the problem of efficient certificate revocation. As the authors of the ARAN protocol mention, the node that has its certificate revoked may not forward a broadcast message that announces the revocation, creating a partition in the network [25].

Furthermore, we question the applicability of identity-based certificates in open and dynamic environments. Even if a node manages to access the appropriate certification authority to verify the identity certificate of another node, there is no guarantee regarding the behavior of the identified node. Mobile ad hoc networks provide a medium for communication between strangers, therefore the identity of the participants cannot, and should not, be used as a basis for security solutions.

Symmetric cryptography is also used by several of the surveyed approaches in order to guarantee the authenticity and the integrity of the exchanged messages, usually by employing a keyed MAC method and a pre-shared key. This solution also suffers from problems such as increased storage requirements for the shared keys and increased packet sizes. Specifically, in a network of n nodes the total number of different symmetric keys needs to be at least

$$\frac{n(n-1)}{2}$$

assuming that each pair of nodes share a common key. Each node that forwards an incoming packet must calculate a keyed MAC of the packet, attach the result to the packet and transmit it. After a few hops the size of the packet becomes inappropriately large. Solutions that attempt to alleviate this by sharing the same symmetric key between more than two nodes lose the ability to properly authenticate peers. Moreover, they are particularly vulnerable to compromises and suffer from increased costs of key updates. Another higher-level problem with protocols that rely on symmetric cryptography approaches lies in the establishment of the utilized keys. Designers must take into account application scenarios between users and how their real-world interaction patterns facilitate or not the existence of such secure associations.

CONCLUSION

This survey has presented the best known protocols for securing the routing function in mobile ad hoc networks. The analysis of the different proposals has demonstrated that the inherent characteristics of ad hoc networks, such as lack of infrastructure and rapidly changing topologies, introduce additional difficulties to the already complicated problem of secure routing. The comparison we have completed between the surveyed protocols indicates that the design of a secure ad hoc routing protocol constitutes a challenging research problem since already existing generic solutions, such as IPsec, cannot be successfully applied. Additionally, the flexibility of ad hoc networks enables them to be deployed in diverse application scenarios. Each different scenario has its own set of security requirements and places unique demands on the underlying routing protocol. Hence, an additional difficulty in designing a secure protocol lies in the application scenario that is going to be protected, and how well the protocol can handle scenarios different than the scenario for which it has been designed.

Military applications of ad hoc networks are probably the area that requires the most highly secure routing functionality. In this case the applications that run on top of the network are of critical importance, therefore the underlying routing process should provide a high level of protection, while possibly having less strict performance requirements. On the other hand, commercial application scenarios of ad hoc networking may place higher demands on the underlying routing protocol. However, security still plays an important role since even in commercial or domestic ad hoc environments the exchanged information is usually confidential, e.g. credit card numbers, or of a private nature. Therefore, a flexible secure ad hoc routing solution should take into account the performance-security trade-off associated with an application and dynamically achieve the required equilibrium.

An example of the routing challenges currently faced by mobile ad hoc networks is outlined in a previous review paper [2]. Although the authors mention challenges such as quality of service support and location-aided and power-aware routing approaches, there is no mention of security considerations. We believe that security should be an integral part of any ad hoc and wireless networking routing solution. Retrofitted generic solutions such as IPsec are not able to provide adequate protection in ad hoc environments where all the participants are internal entities of the network [16]. There is a need for a solution that offers a lightweight approach to the problem of secure ad hoc routing, while taking into account the security prerequisites of different application scenarios, offering a flexible approach to the required security-performance balance. In order to guarantee successful deployment a solution should have realistic operational requirements based on the application domain in which it is applied. For example, the establishment of security associations can be greatly simplified by integrating human intervention in the key management protocols [24]. However, such approaches may not be applicable in ad hoc networking environments with high mobility patterns.

Finally, we believe that more work is needed toward a formal model based on solid mathematical grounds that can precisely give a definition for secure ad hoc routing. This will allow researchers to formally prove whether a proposed protocol satisfies the definition under certain assumptions and will make the comparison between the properties of each proposal an easier and well-structured process.

ACKNOWLEDGMENTS

The authors would like to thank Martin Reisslein and the anonymous reviewers for their constructive comments. The first author is supported by the Irish Research Council for Science, Engineering and Technology (IRCSET), as part of the Embark Initiative, under contract number RS/2002/599-2. This material is based in part upon works supported by Science Foundation Ireland under grant number 03/CE3/I405.

REFERENCES

[1] S. Marti *et al.*, "Mitigating Routing Misbehavior in Mobile Ad hoc Networks," *Proc. 6th Annual ACM/IEEE Int'l. Conf. Mobile Comp. and Net. (Mobicom'00)*, Boston, Massachusetts, Aug. 2000, pp. 255–65.

[2] E. M. Royer and C.-K. Toh, "A Review of Current Routing Protocols for Ad hoc Mobile Wireless Networks," *IEEE Pers. Commun.*, vol. 2, no. 6, Apr. 1999, pp. 46–55.

[3] S. Ramanathan and M. Steenstrup, "A Survey of Routing Techniques for Mobile Communications Networks," *Mobile Networks And Applications*, vol. 2, no. 1, Oct. 1996, pp. 89–104.

[4] T. Clausen, P. Jacquet, and L. Viennot, "Comparative Study of Routing Protocols for Mobile Ad hoc Networks," *Med-Hoc-Net'02*, Sardegna, Italy, Sept. 2002, p. 10.

[5] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector (DSDV) for Mobile Computers," *Proc. ACM Conf. Commun. Architectures and Protocols (SIGCOMM'94)*, London, UK, Aug. 1994, pp. 234–44.

[6] T. Clausen *et al.*, "The Optimized Link State Routing Protocol: Evaluation Through Experiments and Simulation," *Proc. 4th Int'l. Symp. Wireless Pers. Multimedia Commun.*, Aalborg, Denmark, Sept. 2001, 6 pp.

[7] R. Perlman, *Interconnections: Bridges, Routers, Switches, and Internetworking Protocols*, Addison-Wesley, Reading, MA, 2000.

[8] J. Moy, "Open Shortest Path First (OSPF) Version 2," RFC 2328, Apr. 1998.

[9] C. E. Perkins, and E. M. Royer, "Ad hoc On-Demand Distance Vector Routing," *Proc. 2nd IEEE Wksp. Mobile Comp. Sys. and Applications*, New Orleans, LA, Feb. 1999, pp. 90–100.

[10] D. B. Johnson, D. A. Maltz, and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad hoc Networks," *Ad Hoc Net.*, C. E. Perkins, ed., Addison-Wesley, 2001, pp. 139–72.

[11] D. B. Johnson *et al.*, "The Dynamic Source Routing Protocol for Mobile Ad hoc Networks (DSR)," Internet Draft, draft-ietf-manet-dsr-07.txt, Feb. 2002.

[12] C. E Perkins, E. M. Royer, and S. Das, "Ad hoc On-demand Distance Vector (AODV)," RFC 3561, July 2003.

[13] S. Murphy, "Routing Protocol Threat Analysis," Internet Draft, draft-murphy-threat-00.txt, Oct. 2002.

[14] K. Zhang, "Efficient Protocols for Signing Routing Messages," *Proc. Symp. Network and Distributed Systems Security (NDSS'98)*, San Diego, CA, Mar. 1998, pp. 29–35.

[15] P. Papadimitratos and Z. J. Haas, "Securing the Internet Routing Infrastructure," *IEEE Commun. Mag.*, vol. 10, no. 40, Oct. 2002, pp. 60–68.

[16] R. Perlman, "Network Layer Protocols with Byzantine Robustness," Ph.D. Dissertation, MIT/LCS/TR-429, MIT, Oct. 1988.

[17] B. Awerbuch *et al.*, "An On-demand Secure Routing Protocol Resilient to Byzantine Failures," *WISE'02*, Atlanta, Georgia, Sept. 2002, pp. 21–30.

[18] J. Lundberg, "Routing Security in Ad Hoc Networks," http://citeseer.nj.nec.com/400961.html.

[19] J.-F. Raymond, "Traffic Analysis: Protocols, Attacks, Design Issues and Open Problems," *Proc. Wksp. Design Issues in Anonymity and Unobservability*, Berkeley, CA, July 2000, pp. 7–26.

[20] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Packet Leashes: A Defense Against Wormhole Attacks in Wireless Ad hoc Networks," *Proc. 22nd Annual Joint Conf. IEEE Comp. and Commun. Societies (Infocom'03)*, San Francisco, CA, Apr. 2003.

[21] Y.-C. Hu, D. B. Johnson, and A. Perrig, "SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad hoc Net-

works," Proc. 4th IEEE Wksp. Mobile Comp. Sys. and Applications, Callicoon, NY, June 2002, pp. 3–13.
[22] L. Zhou and Z. J. Haas, "Securing Ad hoc Networks," *IEEE Net. Mag.*, vol. 6, no. 13, Nov./Dec. 1999, pp. 24–30.
[23] F. Stajano and R. Anderson, "The Resurrecting Duckling: Security Issues for Ad hoc Wireless Networks," *Proc. 7th Int'l. Wksp. Security Protocols*, Cambridge, UK, Apr. 1999, pp. 172–94.
[24] K. Sanzgiri *et al.*, "A Secure Routing Protocol for Ad hoc Networks," *Proc. 10th IEEE Int'l. Conf. Network Protocols (ICNP'02)*, IEEE Press, 2002, pp. 78–87.
[25] S. Yi, P. Naldurg, and R. Kravets, "Security-Aware Ad hoc Routing for Wireless Networks," *Proc. 2nd ACM Symp. Mobile Ad Hoc Net. and Comp. (Mobihoc'01)*, Long Beach, CA, Oct. 2001, pp. 299–302.
[26] P. Papadimitratos and Z. J. Haas, "Secure Routing for Mobile Ad hoc Networks," *Proc. Communication Networks and Distributed Systems, Modeling and Simulation Conf. (CNDS'02)*, San Antonio, Texas, Jan. 2002, pp. 27–31.
[27] A. Perrig *et al.*, "Efficient and Secure Source Authentication for Multicast," *Proc. Symp. Network and Distributed Systems Security (NDSS'01)*, San Diego, California, Feb. 2001, pp. 35–46.
[28] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad hoc Networks," *Proc. 8th ACM Int'l. Conf. Mobile Comp. and Net. (Mobicom''02)*, Atlanta, Georgia, Sept. 2002, pp. 12–23.
[29] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad hoc Networks," *Proc. 22nd Ann. Joint Conf. IEEE Comp. and Commun. Societies (INFOCOM 2003)*, IEEE Press, 2003, pp. 1976–86.
[30] M. G. Zapata, and N. Asokan, "Secure Ad hoc On-demand Distance Vector Routing," *ACM Mobile Comp. and Commun. Review*, vol. 3, no. 6, July 2002, pp. 106–07.
[31] R. Ramanujan, A. Ahamad, and K. Thurber, "Techniques for Intrusion Resistant Ad hoc Routing Algorithms (TIARA)," *Proc. Military Commun. Conf. (MILCOM 2000)*, Los Angeles, CA, Oct. 2000, pp. 660–64.
[32] R. Ogier, M. Lewis, and F. Templin, "Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)," Internet Draft, draft-ietf-manet-tbrpf-08.txt, Apr. 2003.
[33] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," RFC 2401, Nov. 1998.
[34] K. Sufatrio and K.-Y. Lam, "Scalable Authentication Framework for Mobile IP (SAFE-MIP)," Internet Draft, draft-rio-mobileip-safe-mip-00.txt, Nov. 1999.
[35] N. Ferguson and B. Schneier, "A Cryptographic Evaluation of IPsec," http://citeseer.nj.nec.com/ferguson00cryptographic.html
[36] T. Imielinski and J. C. Navas, "GPS-based Geographic Addressing, Routing, and Resource Discovery," *Commun. ACM*, vol. 42, no. 4, Apr. 1999, pp. 86–92.
[37] S. Toner and D. O'Mahony, "Self-Organizing Node Address Management in Ad hoc Networks," *Pers. Wireless Commun., IFIP-TC6 8th Int'l. Conf. (PWC 2003)*, 2003, pp. 476–83.
[38] S. Buchegger and J.-Y. Le Boudec, "Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes: Fairness In Dynamic Ad hoc NeTworks)," *Proc. 3rd Symp. Mobile Ad Hoc Net. and Comp. (MobiHoc 2002)*, ACM Press, 2002, pp. 226–36.
[39] P. Papadimitratos and Z. J. Haas, "Secure Link State Routing for Mobile Ad hoc Networks," *Proc. IEEE Wksp. Security and Assurance in Ad hoc Networks*, IEEE Press, 2003, pp. 27–31.
[40] S. Capkun and J.-P. Hubaux, "BISS: Building Secure Routing out of an Incomplete Set of Security Associations," *Proc. ACM Wksp. Wireless Security*, ACM Press, 2003, pp. 21–29.
[41] B. Schneier, *Applied Cryptography — Protocols, Algorithms and Source Code in C, 2nd Ed.*, John Wiley & Sons, Inc., 1996.
[42] D. Eastlake and P. Jones, "US Secure Hash Algorithm 1 (SHA1)," RFC 3174, Sept. 2001.
[43] R. Rivest, "The MD5 Message-Digest Algorithm," RFC 1321, Apr. 1992.
[44] R. Resnick and R. Zeckhauser, "Trust Among Strangers in Internet Transactions: Empirical Analysis of Ebay's Reputation System," *Advances in Applied Microeconomics: The Economics of the Internet and E-Commerce*, vol. 11, Elsevier Science Ltd., 2002, pp. 127–57.
[45] A. D. Wood and J. A. Stankovic, "Denial of Service in Sensor Networks," *IEEE Comp.*, vol. 35, no. 10, Oct. 2002, pp. 54–62.
[46] N. Asokan and P. Ginzboorg, "Key Agreement in Ad hoc Networks," *Comp. Commun.*, vol. 23, 2000, pp. 1627–37.
[47] S. Capkun, L. Buttyan, and J.-P. Hubaux, "Self-organized Public-key Management for Mobile Ad hoc Networks," *IEEE Trans. Mobile Comp.*, vol. 1, no. 2, Jan.-Mar. 2003.
[48] B. Christianson and W.S. Harbison, "Why Isn't Trust Transitive?" *Proc. Int'l. Wksp. Security Protocols*, Cambridge, UK, 1996, pp. 171–76.
[49] J. Kong *et al.*, "Providing Robust and Ubiquitous Security Support for Mobile Ad hoc Networks," *Proc. 9th Int'l. Conf. Network Protocols (ICNP)*, 2001.
[50] D. Balfanz *et al.*, "Talking to Strangers: Authentication in Ad hoc Wireless Networks," *Proc. 9th Network and Distributed System Security Symp.*, 2002.
[51] F. Stajano, "The Resurrecting Duckling — What Next?," *Proc. 8th Int'l. Wksp. Security Protocols*, LNCS 2133, 2001, pp. 204–14.
[52] S. Capkun, J.-P. Hubaux, and L. Buttyan, "Mobility Helps Security in Ad hoc Networks," *Proc. 4th ACM Int'l. Symp. Mobile Ad hoc Net. and Comp. (MobiHoc 2003)*, June 2003, pp. 46–56.

BIOGRAPHIES

PATROKLOS G. ARGYROUDIS [StM] (argp@cs.tcd.ie) received his B.Sc. (honors) in computer science from the University of Sheffield. He is currently a doctoral student in the Department of Computer Science at the University of Dublin, Trinity College. His research interests include network security and applied cryptography, especially in the areas of decentralized authorization and scalable access control for pervasive computing. He is a student member of the ACM.

DONAL O'MAHONY (omahony@cs.tcd.ie) is an associate professor at the University of Dublin, Trinity College, where he leads a research group in Networks and Telecommunications. This group has ongoing projects in a wide range of areas, including electronic commerce, network security, and mobile communications technology, and has been very influential in developing the concept of 4th Generation mobile systems. He is a Fulbright fellow and a fellow of Trinity College Dublin. He is the author of two books, including his most recent work on electronic payment systems, which is in its 2nd edition. In July 2004 he led a team to establish CTVR, a major multi-university research center established in association with Bell Labs Ireland. He is now the full-time director of this center.