Towards provable security for route discovery protocols in mobile ad hoc networks

Mike Burmester, Breno de Medeiros
Department of Computer Science, Florida State University, Florida, USA
E-mail: {burmester, breno}@cs.fsu.edu

Abstract

Mobile ad hoc networks (MANETs) are collections of wireless mobile devices with restricted broadcast range and resources, and no fixed infrastructure. Communication is achieved by relaying data along appropriate routes. Discovering such routes however is a major task, both from an efficiency point of view and from a security point of view. In particular, it is important that the route discovered is not controlled by the adversary. Several route discovery protocols have been proposed in the literature that address the particular requirements of a MANET, but as we demonstrate in this paper their security is still analyzed in weak models and cannot tolerate certain classes of attack.

Recently, a security framework tailored to the specific requirements of MANETs was presented and a route discovery algorithm, endairA, was proposed that was "proven" secure in this framework. In this paper we show that the security proof for endairA is flawed, and that the proposed route discovery algorithm is vulnerable to a *hidden channel* attack. We then analyze the security framework used for route discovery and argue that composability is an essential feature for ubiquitous applications. We conclude by discussing some of the main issues that must be addressed for secure route discovery.

Keywords: Secure routing, MANET security, concurrent security, subliminal channels, universal composability, provably secure protocols.

1 Introduction

Routing is a basic functionality for multihop mobile ad hoc networks (MANETs). These networks are decentralized, with nodes acting both as hosts and routers, forwarding packets for nodes that are not in transmission range of each other. Several route discovery algorithms have been proposed in the literature [26, 24, 13, 18, 17]. These focus mainly on efficiency issues, such as scalability with respect to network size, traffic load, mobility, and on the adaptability to network conditions, such as link quality and power requirements. Some of the proposed routing algorithms also address security issues [5, 2, 4, 3, 17, 31, 30, 25, 21, 28], but their security is analyzed in artificially constrained adversary models. There are several reasons for this, the most important one being that it is hard to model a formal security framework that captures all the basic security aspects of a MANET.

Several attempts have been made, the most recent one being by G. Acs, L. Buttyàn and I. Vajda [12, 5], in which the universal composability security framework [14, 27] is adapted for MANET applications. This security framework is used to prove that the route discovery algorithms SRP [24] and Ariadne [5] are insecure and subject to a *hidden channel* attack. Acs *et al.* then propose a new route discovery algorithm, endairA, and "prove" that it is secure in this security framework.

In this paper we first show that the security proof for endairA given in [5] is flawed and that indeed this route discovery algorithm is subject to a hidden channel attack. We then analyze the security framework for MANETs proposed in [5], and argue that universal composability is an essential feature

for ubiquitous applications. We conclude by discussing the main issues that have to be addressed for secure route discovery.

The organization of this paper is as follows. In Section 2 we overview SRP and Ariadne. In Section 3 we briefly describe the attack on Ariadne given in [5, 12], and the security framework proposed for analyzing MANETs. In Section 4 we show that the security proof for endairA is flawed and that this algorithm is subject to a hidden channel attack. We then discuss the importance of concurrency attacks. This is followed in Section 5 by a general discussion on the requirements for a formal security framework for MANETs. In Section 6 we discuss impossibility results on secure routing, and in Section 7 we discuss some possibility results for the routing problem.

2 Routing Algorithms

We distinguish three basic phases in routing: (i) *route discovery*, in which one or more routes (consisting of adjacent nodes) that link a source S to a target T are sought, (ii) *route maintenance*, in which broken links of established routes are fixed, and (iii) *data communication*, in which data is forwarded via established routes. Route discovery is initiated by a source node S that requests from its neighbors information that can be used to find a route that links it to a target node T. The neighbors of S forward the request to their neighbors, who in turn forward it to their neighbors, and so on, until eventually a route that links S to T is discovered. All nodes on a route other than S, T are called *intermediate* nodes.

There are two general types of route discovery: *proactive* and *reactive* or *on-demand*. Proactive routing is usually table driven: nodes maintain routing tables with routing information to potential target nodes. The tables are updated at regular intervals, and are used by intermediate nodes for route discovery. With reactive algorithms, routes are discovered only when needed.

Proactive routing is network-centric, and is appropriate for networks with heavy communication traffic for which security is not critical. Reactive routing is source-centric: intermediate nodes are restricted to forwarding and possibly verifying route requests or route responses. From a security point of view, reactive (on-demand) routing is preferable because the security is to a large extent centralized (managed by the source).

2.1 The Source Routing Protocol (SRP)

SRP [24] is an on-demand routing protocol that captures the basic features of reactive routing. In SRP, route requests generated by a source S are protected by MACs (Message Authentication Codes) computed using a private key shared with the target T. Requests are broadcast to all the neighbors of S. Each neighbor that receives a request for the first time appends its identifier to the request and re-broadcasts it. Neighbors of neighbor nodes do the same. The MAC in the request is not checked because only S and T know the key used to compute it. When this request reaches the target T, its MAC is checked by T. If it is valid then it is assumed by T that all adjacent pairs of nodes on the path of the route request are neighbors. Such paths are called *valid* or *plausible routes*. The target T replaces the MAC of a valid route request, by a MAC computed with the same key that authenticates the route. This is then send back (upstream) to S using the reverse route. For example, a route request that reaches an intermediate node X_j is of the form:

$$msg_{S,T,rreq} = (rreq, S, T, id, sn, X_1, \ldots, X_j, mac_S),$$

with id a randomly generated route identifier, sn a session number, and mac_S a MAC computed by S with a key shared with T on $(rreq, S, T, id, sn)$. A route reply of the target T is of the form:

$$msg_{S,T,rrep} = (rrep, S, T, id, sn, X_1, \ldots, X_p, mac_T),$$

with mac_T a MAC computed by T with the key shared with S on all the message fields that precede it.

Observe that even though the upstream route from T to S is authenticated by the target, the downstream route (S to T) is not. Consequently faulty nodes pairs (X_{j-1}, X_j) that are adjacent on this route may not be neighbors, but divert traffic via other routes. The faulty nodes need not include the details of these routes in the route request. Therefore the discovered route may not be a valid route, in the sense that some of its adjacent nodes may not be neighbors.

2.2 Ariadne

Ariadne [18] is an on-demand routing algorithm whose route requests are authenticated. There are three versions of Ariadne depending on the mode of authentication: one uses MACs, one TESLA [1], and one digital signatures. In this paper we consider an optimized MAC version. For this version, a typical route request of an intermediate node X_j on route $S, X_1, \ldots, X_p, T$ is of the form:

$$msg_{S,T,rreq} = (rreq, S, T, id, X_1, \ldots, X_j, mac_{SX_1 \cdots X_j}),$$

where $mac_{SX_1 \cdots X_j}$ is a MAC computed by X_j with a private key it shares with T on the route request received from X_{j-1}: $(rreq, S, T, id, X_1, \ldots, X_{j-1}, mac_{SX_1 \cdots X_{j-1}})$. The route reply of T is:

$$msg_{S,T,rrep} = (rrep, S, T, id, X_1, \ldots, X_p, mac_T),$$

with mac_T a MAC computed by T with the key shared with S on all the message fields that precede it. This is broadcast upstream to S via the nodes $X_p, X_{p-1}, \ldots, X_1$.

3 Analysis of Ariadne

L. Buttyàn and I. Vajda proposed a security

L. Buttyàn and I. Vajda described a security framework tailored to analyze on-demand source routing algorithms for MANETs. This framework was used to analyze SRP and Ariadne [12], finding them insecure against hidden-channel attacks, and led to the design of endairA, an on-demand route-discovery protocol that the authors claim to be provably secure. Later, G. Acs, L. Buttyàn and I. Vajda refined the security framework, which we refer to as the ABV model [5]. A proof of the security claim for endairA is also given in [5].

In this section we first describe this particular security framework and the attack on Ariadne. We then describe endairA, a variant of Ariadne. We conclude the section by showing that the security proof for endairA given in [12] is flawed, and that indeed this route discovery protocol is not secure even in the restricted security framework of Buttyàn and I. Vajda.

3.1 The ABV security model

The security framework used by Acs, Buttyàn, and Vajda [5] is based on the simulation paradigm for protocol security, which was envisioned early by Beaver [6] and Beaver and Haber [7] in the context of information-theoretic security; and that culminated into two standing (and related) approaches in the (standard) complexity-theoretic security model, developed independently as the secure reactive systems approach by Pfitzmann and Waidner [27], and Backes, Pfitzmann and Waidner [22]; and as the universally composable security framework by Canetti [14].

These approaches (here we use the treminology in [5]) compare executions of a protocol π in a *real-world model* to its executions in an *ideal-world model* that is controlled by the functionality $\mathcal{F}_\pi$, that captures formally the goals that π is supposed to achieve. In the real-world, the adversary is modeled as a traditional Byzantine adversary of the Dolev-Yao model [15], i.e., it is able to schedule

and tamper with all communication channels, to provide inputs to honest parties and observe their outputs,[1] and to coordinate the actions of all corrupted parties. Additionally, the adversary is capable of interacting with other sessions of the protocol that may be executing concurrently.[2] The ideal-world adversary mimics the behavior of the real-world one to allow for simulations of real-world protocol executions in the ideal-world. In order that π be secure in this framework, the effects on the execution of π in the real-world model by *any* real-world adversary $\mathcal{A}$ should be indistinguishable from those of an appropriately chosen ideal-world adversary $\mathcal{A}'$ in the ideal-world model.

In the model described in [5], a MANET is represented by a graph $G(V, E)$, with node set V and edge set E. Each node v is assigned an identifier $\ell \in L$. It is assumed that the identifiers are authenticated during a neighbor discovery process, so the links in E represent true wireless links. This model allows faulty nodes to share possession of compromised identifiers and use any subset of these during the discovery process. Therefore, after the neighbor-discovery process, a node may learn a set of identifying labels that are possessed by a faulty neighbor node. Consequently a faulty node may appear to the non-faulty nodes as having multiple identifiers—even though non-faulty nodes have unique identifiers.

A *configuration* [5] of a MANET is a triple $(G(V, E), V^*, \mathcal{L})$, with $V^* \subset V$ the set of corrupted nodes and $\mathcal{L} : V \to 2^L$, a labeling function that assigns to each node a set of identifiers in such a way that non-corrupted nodes $v \in V \backslash V^*$ have unique identifiers. A sequence of distinct identifiers $\ell_1, \ell_2, \ldots, \ell_n$, $n \geq 2$, is called a *plausible route*, if it can be partitioned into successive subsequences such that: (i) the identifiers of each partition are assigned to a single node $v_i \in V$, (ii) the corresponding sequence of nodes $v_1, v_2, \ldots, v_k$, $2 \leq k \leq n$, forms a simple path in G.

This definition is intended to capture the basic requirements of a route, given that faulty nodes may share their private identifying keys and can *extend* a route by using any sequence of corrupted identifiers. Note that this implies that some of the edges of the path of a plausible route may be virtual and not correspond to wireless links. The particular case when corrupted neighbor nodes *remove* themselves from routes must also be addressed. To deal with such attacks the authors of [5] propose to merge faulty neighbor nodes into a single node whose neighbors are those of the merged nodes. As a result, the neighbors of a faulty node on a plausible route are not faulty. This modification of the definition results in some of the edges of a plausible route corresponding to multi-hop paths that link faulty nodes to a non-faulty node. Consequently the adjacent nodes of a plausible route are either: (i) neighbors in G, or (ii) linked by a path with at least one edge in G and possibly some virtual edges. Plausible routes however do not have adjacent nodes that are faulty.

Our ultimate goal is to show that this definition is artificial and that no route discovery algorithm can find such routes in the ABV security framework.

3.2 The attack on Ariadne

We briefly describe the Buttyàn-Vajda attack [5, 12]. Consider an instance of Ariadne with source node S and let (S, A, X, B, Y, D, T) be a sequence of identifiers of pairwise neighbor nodes, in which only X, Y are faulty. Let C is another neighbor of both X and Y. In the attack, when the first adversarial node X receives the route request

$$msg_{S,T,rreq} = (rreq, S, T, id, A, mac_{SA}),$$

it broadcasts

$$msg_{S,T,rreq} = (rreq, S, T, id, A, X, mac_{SAX}).$$

This is received by both B and C, that broadcast the corresponding route requests. The second adversarial node Y does not respond to either request, while a little later, the first adversarial node

[1] In the universal composability model, the ability to assign inputs and observe outputs rests with a separate party called the *environment* that interacts with the adversary in an arbitrary fashion.

[2] Again, the external interaction is captured by the *environment* in the case of the universal composability model.

X creates a fake route reply in the name of Y:

$$msg_{S,T,rrep} = (rrep, S, T, id, A, X, B, Y, mac_{SAX}), \tag{1}$$

and sends it to B. B only checks the id and that Y is its neighbor (but not mac_{SAX}). Since it has processed an earlier request with identifier id it will re-broadcast this, intending it for X. Node Y intercepts it and generates the route request:

$$msg_{S,T,rreq} = (rreq, S, T, id, A, X, Y, mac_{SAXY}).$$

Since the iterated MAC is correctly constructed, it will accepted by the target T which creates and sends back the route reply:

$$msg_{S,T,rrep} = (rrep, S, T, id, A, X, Y, D, mac_T).$$

When this reaches Y, the label for node C is added to the listing, so that C will re-broadcast it. When X gets it, this label is discarded, and the message send back to the source S, where it will get validated.

In this attack the adversarial node X has succeeded in shortening an existing route by using a hidden channel linking it to the second faulty node Y, and sending via this channel the message (1) to Y. This message contains mac_{SAX}, a MAC that Y needs in order to compute mac_{SAXY}. The hidden channel exploits a particular feature of wireless communication: when a node transmits a message, *all* its neighbors will receive it. There are several other hidden channels that X, Y could use, as we shall see later.

3.3 The protocol endairA

This variant of Ariadne was proposed in [12] to address the hidden channel attack described above. In endairA [12], route replies of intermediate nodes X_j are protected, rather than the route requests as in Ariadne. A typical route request of X_j on route $S = X_0, X_1, \ldots, X_p, X_{p+1} = T$, $j = 0, 1, \ldots, p$, is of the form:

$$msg_{S,T,rreq} = (rreq, S, T, id, X_1, \ldots, X_j),$$

while the route reply of X_j, $j = 1, \ldots, p+1$, is:

$$msg_{S,T,rrep} = (rrep, S, T, id, X_1, \ldots, X_p, sig_T, \ldots, sig_{X_j}),$$

where $sig_T, \ldots, sig_{X_j}$ are digital signatures of $T, \ldots, X_j$ on the message fields that precede them.

4 Analysis of endairA

In [5], it is "proven" that endairA is secure in the ABV security model, provided the signature scheme used is secure against chosen message attacks. In this section we first show that this proof is incomplete. We then show that it is wrong and that endairA is subject to an interleaving attack. We conclude with a discussion on concurrency attacks.

4.1 A flaw in the security proof of endairA

The proof in [5] considers the possibility of an attack against endairA being successful, and proceeds to derive a contradiction. Let $R \equiv (\ell_{ini}, \ell_1, \ldots, \ell_p, \ell_{tar})$ be a route that is accepted by endairA, with ℓ_{ini} the label of a non-adversarial initiator node and ℓ_{tar} the label of the target. R is assumed, by contradiction, not to be plausible, that is to have at least one pair of adjacent nodes that are not

neighbors (in G). Because faulty nodes on routes are merged in the ABV model, at least one of these nodes must be non-faulty.

At this point the authors in [5] make an additional assumption for the ABV model that prohibits faulty nodes on plausible routes having a wireless link (that is, being neighbors in G), or from having some other out-of-band channel. This is a strong restriction on the security guarantees that the ABV model can provide, but we follow this paradigm because we wish to show that endairA fails in the exact model in [5].

For the sake of seeking a contradiction, the proof in [5] lets $P_1, P_2, \ldots, P_k$ be a partition of the non-plausible route R that has been accepted by endairA. This implies one of two cases: Either (1) there exist two partitions $P_i = \{\ell_j\}$ and $P_{i+1} = \{\ell_{j+1}\}$ such that both ℓ_j and ℓ_{j+1} are identifiers that correspond to non-adversarial vertices that are not neighbors or; (2) There exist three partitions $P_i = \{\ell_j\}$, $P_{i+1} = \{\ell_{j+1}, \ldots, \ell_{j+q}\}$, and $P_{i+1} = \{\ell_{j+q+1}\}$ such that ℓ_j and ℓ_{j+q+1} are non-compromised identifiers and $\ell_{j+1}, \ldots, \ell_{j+q}$ are compromised identifiers, but the vertices corresponding to ℓ_j and ℓ_{j+q+1} do not share a common adversarial neighbor. The flaw in the proof is the argument against the possibility of case (2). Quoting [5]:

> *Machine ℓ_j must have received*
>
> $$msg' = (rrep, \ell_{ini}, \ell_{tar}, (\ell_1, \ldots, \ell_p), (sig_{\ell_{tar}}, sig_{\ell_p}, \ldots, sig_{\ell_{j+1}})$$
>
> *from an adversarial neighbor, say, A, since ℓ_{j+1} is compromised.*

...

> *Machine ℓ_j must have received*
>
> $$msg' = (rrep, \ell_{ini}, \ell_{tar}, (\ell_1, \ldots, \ell_p), (sig_{\ell_{tar}}, sig_{\ell_p}, \ldots, sig_{\ell_{j+1}})$$
>
> *from an adversarial neighbor, say, A, since ℓ_{j+1} is compromised.*

...

> *In order to generate msg', machine A must have received*
>
> $$msg'' = (rrep, \ell_{ini}, \ell_{tar}, (\ell_1, \ldots, \ell_p), sig_{\ell_{tar}}, sig_{\ell_p}, \ldots, sig_{\ell_{j+q+1}})$$
>
> *because, by assumption, the adversary has not forged the signature of ℓ_{j+q+1}, which is non-compromised. Since A has no adversarial neighbor, it could have received msg'' only from a non-adversarial machine ...*

The fallacy with the above reasoning is contained in the last sentence: There is no such necessity for the adversarial node A to get information from a non-adversarial node. It is true that the security of the ABV model prohibits direct communication (either via wireless links or through any out-of-band channels) between two faulty nodes. However, there exist hidden channels available for compromised nodes to exploit and send communication through. For instance, compromised nodes can arbitrarily tamper with concurrent endairA route discovery requests (which are not authenticated). These route discovery requests need not be initiated by adversarial nodes (in compliance with restriction of the ABV model), they just need to be present due to honest nodes having been prompted to request route discovery by the adversary. Similarly, these requests do not need to be initiated dynamically (as the ABV model also restricts this), only to be under way concurrently and have their messages corrupted dynamically (in accordance with the ABV model).

We conclude that the proof makes the unwarranted assumption that no direct channels implies no direct bandwidth between adversarial nodes; the proof is therefore incomplete. It could be possible that the security claims remained valid even as their proof is incorrectly argued. However, we show

that this is not the case. Indeed, we give concrete examples of how to exploit hidden channels in the next section.

Fundamentally, endairA (and the ABV model) was developed to deal with a class of hidden channels (the intrinsic hidden channel of a wireless broadcast medium in a neighborhood). However, security is not achieved because other hidden channels remain present.

4.2 An attack on endairA

This is a hidden channel attack that does not require out-of-band resources. Consider an instance of endairA with source node S and let (S, A, X, B, Y, D, T) be a sequence of identifiers of pairwise neighbor nodes, in which only X, Y are faulty. In the attack, when the second faulty node Y receives

$$msg_{S,T,rreq} = (rreq, S, T, id, A, X, B),$$

it drops node B from the listing and transmits:

$$msg_{S,T,rreq} = (rreq, S, T, id, A, X, Y).$$

Eventually, the route request will reach the target T, that will compute and send back a route reply. Node Y will then receive from D:

$$msg_{S,T,rrep} = (rrep, S, T, id, A, X, Y, D, sig_T, sig_D). \tag{2}$$

Now, Y can obviously attach its label and signature to this reply and transmit to B the extended reply, but B will not re-transmit it because B is not included in the listing. So Y initiates a new route discovery session with source Y and target X, and sends to B a route request:

$$msg_{Y,X,rreq} = (rreq, Y, X, id'),$$

with an identifier id' that contains the information required to construct the signatures sig_T, sig_D in message (2), the identifier D, and the signature sig_Y of Y (if this is needed). The identifier id' will most likely not be long enough for this purpose, so node Y has to initiate several route discovery sessions using identifiers id'', id''', etc, to get all the bits required. Eventually, X will be able to reconstruct the signatures, and generate the route reply:

$$msg_{S,T,rrep} = (rrep, S, T, id, A, X, Y, D, sig_T, sig_D, sig_Y, sig_X),$$

which is send back to the source S and validated.

Note that the route discovery sessions that were mangled by Y as part of the above attack will eventually be discarded by their respective initiators. Still, one route was accepted that is not plausible, violating the stated concurrent security of endairA. Moreover, the attack will succeed with overwhelming probability in those network topologies that contain a sufficient number of non-adversarial nodes (suitable for initiator and target of concurrent route discovery sessions).

The hidden channel used in this attack exploits the fact that there is enough redundancy in the protocol identifier id to hide signature information. Information can also be hidden in the list of labels included in route requests. For example, if there are n authorized labels, then there are $\binom{n}{k}$ possible lists of k labels that can be used to hide information. Digital signatures that use randomness (e.g., the DSA) can also be used to hide information [29]: the adversarial signer, instead of using a random string, uses the information to be transmitted. This information can then be extracted by any other adversarial node that knows the secret signing key (in our case, X must know the signing key of Y).

Our attack is essentially an *interleaving* attack: different instantiations of the same route discovery algorithm endairA are combined by the adversary to force the route discovery protocol to generate a non-plausible route. This argument leads us to the next level of attacks: *concurrency* attacks.

4.3 Hidden channel attacks and concurrency attacks

In all the attacks described above, including the attacks in [12, 5], adversarial nodes succeed in shortening plausible routes by removing intermediate nodes. The adversarial nodes use hidden channels to communicate and transfer the necessary data (signatures, etc). The hidden channels that we considered above do not use out-of-band resources, although this is an obvious alternative.

However there are other channels that in many respects are much more natural. Indeed the main objective of a route discovery algorithm is to find a route that is a suitable communication channel. Route discovery per se makes little sense. It would therefore be natural for nodes to use for their communication a route that was discovered earlier, whatever their intention.[3] Therefore it is unreasonable to restrict nodes from using hidden channels. Note that privacy is a legitimate goal for secure communication, so intermediate nodes should expect to re-transmit encrypted data.

Let us now pursue our earlier discussion on interleaving protocol instances further. In a networking environment one should expect that several instantiations of a routing protocol are executed. Some may involve route discovery, while others route maintenance, data communication, or general network applications. It makes no sense to require that route communication can only start when all the other route discovery instantiations (and network applications) have been completed. Indeed this argument should be carried to its logical extension: the security of *any* protocol should not be considered in isolation, but in the presence of concurrent executions whether these involve the same protocol or other protocols. Consequently in our adversarial model we should allow the adversary to interleave instantiations of several protocols, all running concurrently. This is a natural requirement for security.

5 The Universal Composability framework for Routing Algorithms

It is well known that attacks on ad hoc routing protocols can be very subtle. Attacks may exploit the nature of the wireless medium, the mobility of the system, power constraints, and more generally the fact that the adversary is not necessarily bounded by the constraints on non-faulty nodes (the system). It is important that such issues be taken into account when designing security models for wireless systems and more generally, models for ubiquitous applications. The universal composability (UC) security framework [14, 27] is designed to deal with the composition of concurrent protocol execution attacks, and is clearly the most appropriate model for ubiquitous applications.

Obviously, one has to make allowances for the constraints imposed on ad hoc network systems and for the fact that their mobility may make conventional route discovery infeasible (*e.g.*, when routes becomes disconnected by the time they are discovered[4]). Below we list some important aspects that are often neglected in order to make security issues more manageable.

5.1 The adversary

It is sometimes suggested that faulty nodes should be bound by the same constraints as non-faulty nodes (see *e.g.*, [12, 5]). This may be the case in some applications, but is not realistic. What prevents an adversary from using a more powerful transceiver, or out-of-band channels, if with such means he can achieve his goal? Furthermore, although it may seem reasonable to assume that the resources of the adversary are (polynomially) bounded, allowing for the constraints on ubiquitous applications, it is unreasonable to assume that the adversary cannot use a transceiver that is, say, 50% more powerful than the norm. That being said, it is technically possible and may be convenient in some cases to restrict the communication capability of nodes in a simulation-based security model such as UC or reactive systems, as demonstrated by the ABV communication model.

[3] The adversary need not be adaptive to mount such an attack: nodes normally store routing information.

[4] In such cases one may use one of the *adaptive gossip* protocols in [9].

5.2 The communication medium

There are several rather nasty attacks on MANETs that are hard to prevent. Of these, the Sybil attack [16] and the wormhole attack [19] are possibly the worst. The Sybil attack deals with problems caused by sharing secret identifying keys: although a non-faulty node is uniquely identified by its public keys, a faulty node may present itself as one of several nodes. In particular, a faulty node may present itself as several nodes *during the neighbor discovery protocol*. Unless there is some way of physically detecting the source of an identifying call, it is hard to detect such attacks. An argument that is usually used to prevent Sybil attacks is to assign a unique identifier to each node (see *e.g.*, [5]), and then use these to identify the nodes during neighbor discovery. This argument is based on an erroneous understanding of public key cryptography. A more convincing argument involves the use of some additional feature of the broadcast medium during neighbor discovery at the network layer.

In a wormhole attack the adversary establishes an out-of-band channel, or a system channel, to subvert the normal functioning of an ad hoc network. In the context of routing, this attack can be used to corrupt route-dicovery (as we did in Section 4). Wormhole attacks can be combined with *timing* or *rushing attacks* [20] in which the attacker succeeds in forwarding packets faster by using appropriate mechanisms or channels (possibly out-of-band). As with the Sybil attacks, these attacks are usually discounted as preventable at the network layer.

It should be pointed out that claiming that an attack is easily preventable at the network layer is in many respects equivalent to claiming that the security of a wireless system can be achieved at the physical layer. Although this may be the case for some restricted applications it fails to take into account the malicious nature of some attacks. Note that route discovery is a distributed (global) computation, whereas neighbor discovery is a local process. Therefore route discovery is better suited to identification of threats such as Sybil and wormhole attacks, which only become detectable when global information is collated.

5.3 Composability issues

We argue that composability is an essential requirement for secure routing in MANETs. Indeed, MANETs can be distinctly characterized from fixed-infrastructure networks by the fact that both the control plane (for routing messages) and the data plane (for communicating messages) are highly subject to a variety of attacks. It becomes essential to understand how the security requirements of each layer interfere with each other.

Indeed, interference between security properties at different layers also manifest themselves in the fixed-infrastructure setting. We illustrate this point with a real-world example, the well-known *rogue packet attack against SSL*, described for instance in [23]. In this active attack, a rushing node injects an SSL packet in an existing TCP connection, re-computing the TCP checksums to ensure acceptance of the inserted packet at the transport layer. When the SSL protocol daemon, residing at the session layer,[5] receives the SSL packet (TCP payload), it determines that the packet has been tampered with by failing to verify the message authentication code (that the attacker is unable to forge for lacking knowledge of the shared authentication keys).

The packet is therefore discarded at the SSL layer. However, since it was already accepted at the TCP layer, and moreover has arrived earlier than the legitimate packet from the original sender, it will prevent TCP from accepting the later (legitimate) packet. This is because the TCP daemon has recorded that packet's sequence number as already received, and will acknowledge it. The SSL session layer fails to recover the missing data, and therefore SSL+TCP does not provide *availability* guarantees.

In this scheme, TCP provides availability but not integrity. SSL provides integrity but relies on the availability properties of TCP. This reliance proves unfounded, as the availability guarantees of

[5] According to the OSI 7-layer network model; or application layer according to the 4-layer TCP-IP network model.

TCP are only provided under the weaker integrity notion corresponding to verifiability of the TCP checksums. Composability fails accordingly.

MANET routing security presents very similar problems. Indeed, as has been demonstrated by the designers of the endairA protocol, even the provision of a single property (safety of route discovery) requires a composable approach [12]. We extend this observation by remarking that special care needs to be taken when assuming properties of lower network layers, specially when such properties are achieved under restrictions. If such restrictions are incompatible with requirements at other layers, a solution may be nominally composable but incomplete because no comprehensive solution is achieved (or achievable) in composition. For an example of such a shortcoming, we re-examine the endairA protocol.

In that protocol, safety-type properties (such as integrity) at the MANET control plane are achieved by assuming restricted availability of transmission channels. However, such restrictions may be fundamentally incompatible with liveness guarantees (such as availability) at the data (user) plane. For instance, a MANET could enforce that other forms of data transmission are interrupted while routing computations are ongoing, realizing the required restriction and supporting safety at the control plane. However, this strategy puts the liveness requirements of the control and data plane in direct conflict. Denial-of-service attacks against data transmission could be initiated by frequent triggering of new routing computations. Limiting the frequency of new routing computations might prevent such attacks at the expense of reducing the network capability to deal with frequent topology changes.

To summarize, in contrast with the situation for fixed-infrastructure networks, where infrequency of topology changes can be assumed and therefore it may be acceptable to deny data services to destinations during any period where routing information to that destination is being (re-)computed, in MANETs it is not acceptable to assume temporal disjointness of the routing discovery and data communication phases, and security under composability of different protocols is necessary. It is insufficient to consider only the simpler (and yet hard to achieve!) requirement of security under concurrent executions of the route discovery protocol.

6 Impossibility Results for Secure Route Discovery

From our discussion above it follows that in general it is not possible to achieve secure route discovery in a MANET within a composable security framework that does not incorporate additional global and physical information, if the route sought is a simple path (as in Section 3.1). However, before pursuing this argument further, it is important to note that there is no way of checking that a discovered route is not under the control of the adversary, because adversarial behavior is unpredictable. So our argument is not about the impossibility of finding secure routes, but the impossibility of finding paths that correspond to physical routes in the network.

Our argument about the impossibility of secure discovery of routes is simple and has been articulated throughout this paper. We base it on the fact that every route discovery algorithm is in practice vulnerable to attacks that exploit alternative communication channels to mount distributed attacks by "encapsulating" and tunneling routing requests. Therefore, it is fundamentally impossible to capture or "model out" Sybil and wormhole attacks from pure-protocol-based security models. The purpose of routing being to establish a communication infrastructure, it is always reasonable to assume the existence of alternative communication channels, namely the same channels whose goal is for the routing discovery to (re-)establish.

7 Positive Result for Secure Route Discovery

Even though it is impossible to discover secure routes in general MANETs, there are several other approaches that could be used to establish secure communication channels. In this section we consider

two such approaches: *multipath routes* and *route discovery with traceability*.

7.1 Multipaths and subgraphs

Routes need not be restricted to paths in the network graph G: any subgraph G_{ST} of G that links the source S to the target T can be used for communication. Of particular interest, from a security point of view, are subgraphs G_{ST} with multiple connectivity between S, T. For example, multipaths [11, 10]. Such routes may have sufficient redundancy to guarantee communication, i.e., may contain at least one secure path (with no adversarial nodes). Obviously such routes will have additional communication overhead. However there are ways to partly mitigate this. For example, the source can select communication paths in G_{ST} on a rotation basis (adaptive multipath routing [11]). Another approach is to use random subgraphs G_{ST} of G that link S, T. Gossip protocols [9] use this approach: this guarantees packet propagation while minimizing the number of nodes that forward packets. This latter approach completely blurs all separation of the routing discovery, maintenance, and data communication phases. Paradoxically, this approach's meshing of functionalities may facilitate showing the composability of its security properties.

7.2 Route discovery with traceability

In general, solutions such as those proposed above are only appropriate for applications in which the security is critical. Perhaps a more practical solution would be to use route discovery algorithms that trace malicious behavior—see e.g., [8]. It is possible to do this in such a way that there is practically no additional cost when the adversary is passive, while the extra cost is only for tracing adversarial nodes ("optimistic" tracing). This approach supports *self-healing* security: if we assume that the number of adversarial nodes is bounded over time then the power of the adversary is diminished with each adversarial attack. As in Section 7.1 this feature provides hope for eventually secure route discovery in the universal composability framework.

8 Conclusion

A new security framework tailored for on-demand route discovery protocols in MANETs is proposed in [5]. This represents a first effort towards a formal security model that can deal with concurrent attacks, and succeeds in mitigating a class of hidden channel attacks—the attacks that are intrinsic to the wireless broadcast medium in a neighborhood. However, as we have observed above, there are a plethora of other hidden channels that become available through concurrent execution of route discovery protocols. Additionally, in the context of mobility, which requires that route discovery take place simultaneously with data communication, large additional bandwidth is naturally generated and available to adversarial nodes. Consequently, in the proposed formal model, it is impossible to prevent that adversarial nodes break up routes by inserting non-existing links. To address this shortcoming, either more flexible definitions of routes must be employed (e.g., redundant routing) or it becomes necessary to address global threats directly, such as those posed by Sybil, wormhole, and more generally, man-in-the-middle attacks.

References

[1] J. Tyger A. Perrig, R. Canetti and D.X. Song. Efficient authentication and signing of multicast streams over lossy channels. In *Proccedings of the IEEE Symposium on Se4curity and Privacy*, 2000.

[2] G. Acs, L. Buttyan, and I. Vajda. Provably secure on-demand source routing in mobile ad hoc networks. Technical Report 159, International Association for Cryptologic Research, 2004.

[3] Gergely Ács, Levente Buttyán, and István Vajda. Provable security of on-demand distance vector routing in wireless ad hoc networks. In *ESAS*, pages 113–127, 2005.

[4] Gergely Ács, Levente Buttyán, and István Vajda. Modelling adversaries and security objectives for routing protocols in wireless sensor networks. In *SASN*, pages 49–58, 2006.

[5] Gergely Acs, Levente Buttyan, and Istvan Vajda. Provably secure on-demand source routing in mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 05(11):1533–1546, 2006.

[6] D. Beaver. Foundations of secure interactive computing. In *Proceedings of Advances in Cryptology (CRYPTO '91), ser. LNCS, vol. 576. Springer*, 1992.

[7] D. Beaver and S. Haber. Cryptographic protocols provably secure against dynamic adversaries. In *Proceedings of Advances in Cryptology (EUROCRYPT '92), ser. LNCS. Springer*, pages 307–323, 1992.

[8] M. Burmester, T. van Le, and Matt Weir. Tracing byzantine faults in ad hoc networks. In *Proc. Computer, Network and Information Security 2003, New York*, pages 43–46, 2003.

[9] M. Burmester, T. van Le, and A. Yasinsac. Adaptive gossip protocols: Managing security and redundancy in dense ad hoc networks. *Journal of Ad hoc Networks*, 5(3):286–297, 2007.

[10] Mike Burmester and Tri Van Le. Secure communications in ad hoc networks. In *Proceedings of the 5th Annual IEEE Information Assurance Workshop' 10 - 11 June 2004 United States Military Academy West Point, New York*, 2004.

[11] Mike Burmester and Tri Van Le. Secure multipath communication in mobile ad hoc networks. *ITCC*, 02:399–405, 2004.

[12] L. Buttyan and I. Vajda. Towards provable security for ad hoc routing protocols. In *Proceedings of the ACM Workshop on Ad Hoc and Sensor Networks (SASN 2004)*, 2004.

[13] C. Perkins. Ad-hoc on-demand distance vector routing, In MILCOM '97 panel on Ad Hoc Networks, 1997.

[14] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings of the IEEE Annual Symposium on Foundations of Computer Science (FOCS 2001)*, pages 136–145, 2001.

[15] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29, 2004.

[16] J. R. Douceur. The sybil attack. In *Proceedings of the International Workshop on Peer-to-Peer Systems (IPTPS 2002)*, 2002.

[17] Y. Hu, A. Perrig, and D. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks, 2005.

[18] Y.-C. Hu, A. Perrig, and D. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. In *Proceedings of the ACM Conference on Mobile Computing and Networking (MOBICOM 2002)*, 2002.

[19] Y.-C. Hu, A. Perrig, and D. Johnson. Packet leashes: A defense against wormhole attacks in wireless ad hoc networks. In *Proceedings of the IEEE Annual Conference on Computer Communications (INFOCOM 2003)*, 2003.

[20] Y.-C. Hu, A. Perrig, and D. Johnson. A survey of secure wireless ad hoc routing protocols. *IEEE Security and Privacy Magazine*, 2(3), May/June 2004.

[21] Yih-Chun Hu, David B. Johnson, and Adrian Perrig. Sead: secure efficient distance vector routing for mobile wireless ad hoc networks. *Ad Hoc Networks*, 1(1):175–192, 2003.

[22] B. Pfitzmann M. Backes and M. Waidner. A general composition theorem for secure reactive systems. In *Proceedings of the Theory of Cryptography Conference (TCC 2004), ser. LNCS, vol. 2951. Springer*, 2004.

[23] C. Meadows. Analysis of the Internet Key Exchange Protocol Using the NRL Protocol Analyzer. *SP*, 13:0214–0216, 1999.

[24] P. Papadimitratos and Z. Haas. Secure routing for mobile ad hoc networks. In *Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, 2002.

[25] P. Papadimitratos and Z. Haas. Securing mobile ad hoc networks. *In, Ilyas, M, Handbook of Ad Hoc Wireless Networks, CRC Press.*, 2002.

[26] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *SIGCOMM '94: Proceedings of the conference on Communications architectures, protocols and applications*, pages 234–244, New York, NY, USA, 1994. ACM Press.

[27] Birgit Pfitzmann and Michael Waidner. Composition and integrity preservation of secure reactive systems. In *ACM Conference on Computer and Communications Security*, pages 245–254, 2000.

[28] Kimaya Sanzgiri, Bridget Dahill, Brian Neil Levine, Clay Shields, and Elizabeth M. Belding-Royer. A secure routing protocol for ad hoc networks. In *ICNP*, pages 78–89, 2002.

[29] G. Simmons. The subliminal channels of the us digital signature algorithm (dsa). In *Proceedings of the 3rd Symposium on: State and Progress of research in Cryptography*, pages 35–54, 1993.

[30] Manel Guerrero Zapata. Secure ad hoc on-demand distance vector routing. *Mobile Computing and Communications Review*, 6(3):106–107, 2002.

[31] Manel Guerrero Zapata and N. Asokan. Securing ad hoc routing protocols. In *WiSE '02: Proceedings of the 3rd ACM workshop on Wireless security*, pages 1–10, New York, NY, USA, 2002. ACM Press.