TOWARDS PROVABLE SECURITY FOR ROUTE DISCOVERY PROTOCOLS IN MANETS

MIKE BURMESTER, BRENO DE MEDEIROS

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, that are dynamically discovered and maintained through collaboration between the nodes. Discovery of such routes is a major task, both from an efficiency and from a security point of view. Recently, a security model tailored to the specific requirements of MANETs was introduced by Acs, Buttyán, and Vajda. Among the novel characteristics of this security model is that it promises security guarantees under concurrent executions, a feature of crucial practical implication for this type of distributed computation. A novel route discovery algorithm called endairA was also proposed, together with a claimed security proof within the same model.

In this paper we show that the security proof for the route discovery algorithm endairA is flawed, and that moreover this algorithm is vulnerable to a *hidden channel* attack. We also analyze the security framework that was used for route discovery, and argue that composability is an essential feature for ubiquitous applications. We conclude by discussing some of the major security challenges for route discovery in MANETs.

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 as routers, forwarding packets for nodes that are not in transmission range of each other. Several route discovery algorithms have been proposed in the literature (see e.g., [1, 2, 3, 4, 5]). 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 (e.g., [6, 7, 8, 9, 10], for a survey see [11]), but their security is restricted to rather weak 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 to address the security of MANET route discovery more robustly, the most recent one being introduced in a series of papers by L. Buttyàn and I. Vajda [12], and by G. Acs, L. Buttyàn and I. Vajda [13, 14, 15, 16]. In these works, the authors develop a formal idealization and simulation framework that adapts ideas from the *secure reactive systems* approach [17] and the *universally composable security* approach [18] to the realm of MANET applications. One of the advantages of the new approach—which we will refer to as the ABV model—is that it highlights security issues related to concurrent protocol executions. Indeed,

Both authors are with Florida State University, Tallahassee FL 32306.

the ABV authors prove that, within their model, the routing algorithms SRP [3] and Ariadne [15] are insecure and subject to a *hidden channel* attack. A solution is then proposed in the form of a novel route discovery algorithm, named endairA—the name reflects the fact that it applies security primitives in the reverse order of the Ariadne protocol—and a proof is also supplied for the claim that endairA is secure in the ABV model [15].

Our main contribution in this paper is to show that the security proof for endairA given in [15] is flawed and that this routing algorithm is similarly subject to a hidden channel attack. Revisiting the ABV model, we present several reasons why we think that concurrent security for MANET route discovery—i.e., the ABV model's security standard—is insufficient in practice because it requires the absence of channels that are always present in any real-world MANET application. We then argue that a higher security standard—namely, composability—is a fundamental requirement for ubiquitous applications. Subsequently, we make some observations about issues that have to be addressed by any routing protocol that achieves security in a composable model.

The organization of this paper is as follows. In Section 2 we overview route discovery and the Ariadne protocol. In Section 3 we briefly describe the attack on Ariadne given in [15, 12], and the ABV model. 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 significance of concurrency-based 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 challenges for secure route discovery, and in Section 7 we summarize our arguments for provable security in MANETs.

2. Routing Algorithms

Routing is a basic network functionality that supports communication. In MANETSs each node acts as a router forwarding data to other nodes. We distinguish three basic phases in routing: (*i*) *route discovery*, in which one or more routes (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*) *packet forwarding*, in which communication is achieved via established routes.

Route discovery can be *proactive* and *reactive (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. Source-initiated on-demand route discovery is triggered by a node that requests from its neighbors information that can be used to find a route that links it to a target node. The neighbors forward the request to their neighbors, and so on, until a route that links S to T is discovered.

Proactive routing is network-centric, and is appropriate for networks with heavy communication traffic for which security is not critical. Indeed, such routing strategies tend to rely on link-to-link security, which implies trust in intermediate nodes. 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 easier to analyze for its security properties, because the security is end-to-end (managed by the source and target).

2.1. **The Source Routing Protocol (SRP).** SRP [3] is an on-demand source 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 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. Intermediate 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 the target 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 on $(rreq, S, T, id, sn)$ computed by S using a key shared with T. If $S, X_1, \ldots, X_p, T$ is a discovered route, then the route reply of the target T has the following fixed form for all intermediate nodes X_j, $1 \leq j \leq p$:

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

where mac_T is a MAC computed by T with the key shared with S on the message field preceding it. Intermediate nodes should check the route reply header (including its id and sn) and that they are adjacent with two of their neighbors on the route before sending the route reply upstream.

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 node pairs (X_j, X_{j+1}) that are adjacent on the route may not be neighbors, but may divert traffic via other routes. The faulty nodes need not include the details of these routes in the route request. It is similarly possible for a malicious node to pad route requests with the identities of other nodes that are not its neighbors, and impersonate these nodes in the reply phase. The resulting route therefore may not be valid, in the sense that some of its adjacent nodes may not be neighbors.

2.2. **Ariadne.** Ariadne [19] is an on-demand routing algorithm based on the Dynamic Source Routing (DSR) protocol [2]. There are several variants of Ariadne, depending on which mode of authentication is used to protect route requests: one uses digital signatures, one TESLA [20], and one uses Message Authentication Codes (MACs). The MAC version has an optimized variant that uses iterated MAC computations instead of several independent MACs. In addition to being more efficient, the iterated MAC version has superior security characteristics when compared to the non-optimized version, as noted in [15]. We describe this version below.

A typical route request that reaches an intermediate node X_j, $1 \leq j \leq p$, on the route $S = X_0, X_1, \ldots, X_p$, $X_{p+1} = T$ 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 the MAC computed by X_j with a key it shares with T on the route request received from X_{j-1}:

$$(rreq, S, T, id, X_1, \ldots, X_j, mac_{SX_1 \cdots X_{j-1}}).$$

The target T, on receiving the last request from X_p, is able to re-compute all intermediate MAC values since it shares a key with each one of the intermediate nodes, and then iteratively reconstruct that sequence up to the last value that should match the MAC received from X_p. If the verification succeeds, with overwhelming probability (given by the security of the MAC construction) all intermediate MACs were correctly computed by the nodes included in the route. The route reply of T is:

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

where mac_T is a MAC computed by T with a key shared with S on the message field that precedes it: $(rrep, S, T, id, X_1, \ldots, X_p)$. This is unicast upstream to S via the nodes $X_p, X_{p-1}, \ldots,$ X_1. Intermediate nodes must check that their label appears in the route, adjacent to two of their neighbors.

3. Analysis of Ariadne

L. Buttyàn and I. Vajda [12] described a security framework tailored to analyze on-demand source routing algorithms for MANETs. This framework was used to analyze SRP and Ariadne, 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 [15]. A proof of the security claim for endairA is also given in [15].

In this section we first outline the ABV framework and the attending attack on Ariadne. We then describe endairA. This discussion is not original, and closely parallels arguments in [15]. However, it is directly cogent to the novel arguments that follow (Section 4), that show that the security proof for endairA provided in [15] is flawed, and that moreover this route discovery protocol is not secure even in the (somewhat restricted) ABV security model.

3.1. The ABV model.

The security framework used by Acs, Buttyàn, and Vajda [15] is based on the simulation paradigm for protocol security, which was envisioned early by Beaver [21] and Beaver and Haber [22] in the context of information-theoretic security; and that culminated in two standing (and related) approaches in the (standard) complexity-theoretic security model, developed independently as the *secure reactive systems* approach by Pfitzmann and Waidner [17], and Backes, Pfitzmann, and Waidner [23], and as the *universally composable security framework* by Canetti [18].

These approaches 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 [24], 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

[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.

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 [15], 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 use out-of-band channels. Consequently a faulty node may appear to the non-faulty nodes as having multiple identifiers—even though non-faulty nodes have unique identifiers. Therefore, after the neighbor-discovery process, a node will learn a set of identifying labels that are possessed by neighbor nodes—where adversarial nodes can share possession of compromised labels and can use any subset of these during the discovery process [25].

A *configuration* of a MANET is a triple $(G(V, E), V^*, \mathcal{L})$, with $V^* \subset V$ the set of faulty nodes and $\mathcal{L} : V \to 2^L$ is a labeling function that assigns to each node a set of identifiers in such a way that: every non-faulty node $v \in V \backslash V^*$ has a a unique label. A sequence of identifiers $\ell_1, \ldots, \ell_n$, $n > 2$, 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 sequence of nodes v_i assigned to the partitions form a simple path in G. This definition is intended to capture the basic requirements of a route, given that faulty (compromised) nodes may share their private identifying keys and may *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 [15] 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.

The real-world and ideal-world models described in [15] are similar to those used in the generic secure reactive system approach [17, 23], but there are some crucial differences. In the ABV framework: (1) The adversary does not have full control of message delivery schedule, in the sense that the broadcast channel enforces the concept of communication rounds—in particular, the ABV framework does not capture *rushing* attacks (synchrony); (2) The adversary may prompt honest parties to initiate new route discoveries but not dishonest ones, in other words, the ABV

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

security framework does not capture concurrent security in the presence of route discovery sessions that are initiated by adversarial nodes; and (3) The adversarial is non-adaptive, i.e., cannot initiate new route discoveries as a function of previously observed messages—See Section 3.2 of [15] for these restrictions—and; (4) The link configuration $(G(V,E), V^*, \mathcal{L})$ of a MANET is enforced in the security framework by the communication medium functionality (Machine C in the real-world model of ABV [15]).

3.2. **The attack on Ariadne.** We briefly describe the attack against Ariadne described in [15]. Consider an instance 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 $C \neq B$ be 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, which broadcast the corresponding route request. The second adversarial node Y does not respond to either request, while a little later, the first adversarial node X creates a fake route reply in the name of Y:

$$msg_{S,T,rrep} = (rrep, S, T, id, A, X, B, Y, mac_{SAX}), \tag{3.1}$$

(with the wrong MAC) and unicasts it to B, who only checks the id and that X, Y are its neighbors. Since B has processed an earlier request with identifier id it will re-transmit this, intending it for X. Node Y intercepts it and generates the route request:

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

This is accepted by D and continued along to T. Since the iterated MAC is correctly constructed, it will be accepted by the target T, who 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—namely the one provided by the lack of directionality in wireless broadcast—linking it to the second faulty node Y, and sending via this channel the message (3.1) to Y. This message contains mac_{SAX}, a MAC that Y needs in order to compute mac_{SAXY}. There are several other hidden channels that X and Y could use, as we shall see later.

3.3. **The protocol endairA.** This is a variant of Ariadne, designed to address the hidden channel attack described above. In endairA, the route replies of intermediate nodes X_j are protected, rather than the route requests as in Ariadne. A typical route request broadcast by a node X_j, $0 \leq j \leq p$, on route $S = X_0, X_1, \ldots, X_p, X_{p+1} = T$, is of the form:

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

while the route reply unicast by X_j, $1 \leq j \leq p+1$, is:

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

where sig_{X_j} is the digital signature of X_j on the message field preceding it.

4. Analysis of endairA

The protocol endairA is claimed to be proven secure in the ABV security framework [15]. We now revisit the proof of security and identify a flaw. The proof in [15] considers the possibility of an attack against endairA being successful, hoping to achieve a contradiction.

Let $(\ell_{ini}, \ell_1, \ldots, \ell_p, \ell_{tar})$ be some route that is accepted by endairA, where ℓ_{ini} is the label of a non-adversarial initiator node, and ℓ_{tar} is the label of the target. This is assumed (by contradiction) not to correspond to a valid route in the sense that it includes non-neighbor vertices. Since adversarial nodes can share labels, any number of adversarial nodes can be subsumed in a single label. However, Acs, Buttyán, and Vajda exclude such faulty routes (which may appear shorter than actual network routes by collusion of adjacent adversarial nodes) by subsuming *all* adjacent adversarial nodes, and indeed any two adversarial nodes with direct means of communication (e.g., via out-of-band channels) as single nodes—see Section 3.1 or [15]. Consequently, adversarial nodes are, by definition, never adjacent in the ABV model. This is an arbitrary restriction that greatly limits the scope of the security statements in the ABV model in their ability to capture realistic security requirements; However, we do not need to leave this model to identify a problem with the security proof of endairA. So, for the sake of argument, we also assume that adversarial nodes are never adjacent.

This implies that the route can be uniquely partitioned as follows: each partition consists of a single non-compromised identifier (label) or a sequence of consecutive compromised identifiers. A *plausible route* is one whose partitions correspond to that of a real route that physically exists in the network. The security statement of endairA is that it only accepts plausible routes. Note that this statement also does not consider an adversarial lengthening of a route by assignment of multiple labels to a single compromised network node as an attack. Again, this is a strong restriction on the security guarantees that the ABV model can provide, but we also follow this paradigm because we wish to show that endairA fails in the exact model in [15].

For the sake of seeking a contradiction, the proof in [15] lets $P_1, P_2, \ldots, P_k$ be a partition of $(\ell_{ini}, \ell_1, \ldots, \ell_p, \ell_{tar})$, which is a non-plausible route 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 nodes 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 $P_{i+2} = \{\ell_{j+q+1}\}$ 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 nodes 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:

> *Machine*[3] 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 ABV model prohibits direct communication (either via wireless links or through any out-of-band channels) between two adversarial 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 route discovery requests of endairA (which are not authenticated). These route requests need not be initiated by adversarial nodes (in compliance with an ABV model restriction), they just need to be initiated by honest nodes prompted by the adversary (through route discovery requests). Similarly, the 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 channels of a wireless broadcast medium in a neighborhood. However, security is not achieved because other hidden channels remain present.

4.1. **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

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

[3] In the standard complexity-theoretic model for system security the components of system (in our case the nodes of a MANET) are modeled by interactive Turing Machines.

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, which will compute and send back a route reply. Node Y will then receive from D:

$$msg_{S,T,rrep} = (rrep, S, T, id, A, X, Y, D, sig_T, sig_D). \tag{4.1}$$

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. However, suppose that Y had earlier received a request from D to find a route linking it to node A. Then, since the adversary schedules (non-adaptively) all the route discoveries prompted by honest nodes in the ABV model, it can arrange for this to be the case. Say the route request was:

$$msg_{D,A,rreq} = (rreq, D, A, id'),$$

with an identifier id'. Y mangles id' into some id'' that contains (possibly encrypted) information that X can use to re-construct the signatures sig_T, sig_D in message (4.1) (and the signature sig_Y of Y if this is needed), before sending it along to B, and eventually X. Now, the identifier id' will most likely not be long enough for this purpose, so node Y must take advantage of several route discovery requests that should go through Y to reach X, mangling all the identifiers. For example, Y may compute $\sigma(sig_T)||\sigma(sig_D) = id'||id''|| \cdots ||id^{(k)}$, where "$||$" is concatenation and σ is a bit permutation known to both X and Y, and use id', id'', $\ldots$, $id^{(k)}$ as identifiers for route requests. Again, since the adversary can prompt honest nodes to create route discovery requests it can ensure that enough sessions will have reached it ahead of time (and non-adaptively). Eventually, X will be able to reconstruct these signatures, and can then 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. Even for the version of endairA with no identifiers (insecure against replay attacks) the attack still applies because other hidden channels exist. For instance, the list of labels included in route requests may also be used to convey information. In particular, if there are n authorized labels, then there are $\binom{n}{k}$ possible lists of k labels that can be used to hide information. The node Y would mangle the concurrent requests by arbitrary combination of nodes to signal the appropriate information to X.

Digital signatures that use randomness (e.g., the DSA) can also be used to hide information [26]: 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).

4.2. **Hidden channel attacks and concurrency attacks.**

In all the attacks described above, including the attacks in [12, 15], 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. 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. 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) framework [18], and the secure reactive systems model [17, 23] were designed to deal with the composition of concurrent protocol execution attacks, and are therefore more appropriate models 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 adversarial nodes should be bound by the same constraints as non-adversarial nodes, for example have similar communication capabilities [15]. This may be the case for some applications, but it is not realistic. Although it may seem reasonable to assume that the resources of adversarial nodes are (polynomially) bounded, allowing for the constraints on

[4] In such cases one may use one of the *adaptive gossip* protocols in [27].

ubiquitous applications, it is unreasonable to assume that adversarial nodes cannot use more powerful transmitters than non-adversarial nodes, say transmitters that are 50% more powerful than the norm,[5] if with such means they can compromise the system.

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 the UC framework or reactive systems, as demonstrated by the ABV communication model.

5.2. **The communication medium.** There are several rather nasty attacks on MANETs that are hard to prevent. Of these, the Sybil attack [25] and the wormhole attack [28] 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. The ABV model seeks to do an end-run about Sybil attacks by considering only *partitions* of plausible routes. However, as seen above, the multiplicity of identifiers can be used as a hidden channel to perform subtler attacks that the ABV model cannot tolerate. Ultimately, it is important to provide some security against Sybil attacks, possibly using some additional feature of the physical broadcast medium during neighbor discovery at the network layer (e.g., by using radio frequency fingerprinting [29]).

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 routing protocols (as we did in Section 4). Wormhole attacks can be combined with *timing* or *rushing attacks* [30] 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 the 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 (routing messages) and the data plane (proper communication 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.

[5] This would make them "virtual" neighbors of some non-adversarial nodes, who would be in their broadcast range, but they could only receive messages from these nodes via out-of-band channels.

Indeed, interference between security properties at different layers also manifests itself 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 [31]. 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,[6] 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. 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 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 routing discovery) requires at least a concurrent approach, as illustrated by the attacks on Ariadne [15]. We extend this observation by remarking that special care needs to be taken when assuming properties of lower network layers, especially 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

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

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. The Challenges of Secure Route Discovery

In this section, we remark that 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 following this argument, 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 articulate distributed attacks by "encapsulating" and tunneling routing requests. Therefore, it does not seem possible 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 those that route discovery will establish.

Even though it is not possible to discover secure routes in general MANETs, there are several other approaches that could be used to establish secure communication channels. In the following we consider two such approaches: *multipath routes* and *route discovery with traceability*.

6.1. **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 [32]. 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 [32]). Another approach is to use random subgraphs G_{ST} of G that link S, T. Gossip protocols [27] 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.

6.2. **Route discovery with traceability.** In general solutions such as those proposed above are only appropriate for applications in which security is critical. Perhaps a more practical solution would be to use routing algorithms that trace malicious behavior—see e.g., [33]. 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 [33]). This approach supports *self-healing* security: the power of the adversary is diminished with each attack, if we assume that the number of adversarial nodes is bounded over time.

7. Conclusion

A new security framework tailored for on-demand route discovery protocols in MANETs was proposed in [15]. This represents a first effort towards a formal security model that can deal with concurrent attacks, and is successful in mitigating a class of hidden channel attacks—the attacks that are intrinsic to the wireless broadcast medium in a neighborhood. However, as we 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] C. E. Perkins and P. 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.* New York, USA: ACM Press, 1994, pp. 234–244.

[2] D. Johnson and D. Maltz, “Dynamic source routing in ad hoc wireless networks,” *Mobile Computing (ed. T. Imielinski and H. Korth), Kluwer Academic Publishers, Dordrecht, The Netherlands*, 1996.

[3] 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.

[4] C. Perkins, “Ad-hoc on-demand distance vector routing,” in *MILCOM ’97, Panel on Ad Hoc Networks*, 1997.

[5] C. E. Perkins and E. M. Belding-Royer, “Ad-hoc on-demand distance vector routing,” in *2nd Workshop on Mobile Computing Systems and Applications (WMCSA ’99), 1999, New Orleans, USA*, 1999, pp. 90–100.

[6] M. G. Zapata, “Secure ad hoc on-demand distance vector routing,” *Mobile Computing and Communications Review*, vol. 6, no. 3, pp. 106–107, 2002.

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

[8] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer, “A secure routing protocol for ad hoc networks,” in *ICNP*, 2002, pp. 78–89.

[9] Y.-C. Hu, D. B. Johnson, and A. Perrig, “SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks,” *Ad Hoc Networks*, vol. 1, no. 1, pp. 175–192, 2003.

[10] Y.-C. Hu, A. Perrig, and D. B. Johnson, “Packet leashes: A defense against wormhole attacks in wireless networks,” in *INFOCOM*, 2003.

[11] Y.-C. Hu and A. Perrig, “A survey of secure wireless ad hoc routing,” *IEEE Security & Privacy*, vol. 2, no. 3, pp. 28–39, 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] G. Acs, L. Buttyan, and I. Vajda, “Provably secure on-demand source routing in mobile ad hoc networks,” International Association for Cryptologic Research, Tech. Rep. 159, 2004.

[14] G. Ács, L. Buttyán, and I. Vajda, "Provable security of on-demand distance vector routing in wireless ad hoc networks," in *ESAS*, 2005, pp. 113–127.

[15] G. Acs, L. Buttyan, and I. Vajda, "Provably Secure On-Demand Source Routing in Mobile Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 11, pp. 1533–1546, 2006.

[16] G. Ács, L. Buttyán, and I. Vajda, "Modelling adversaries and security objectives for routing protocols in wireless sensor networks," in *SASN*, 2006, pp. 49–58.

[17] B. Pfitzmann and M. Waidner, "Composition and integrity preservation of secure reactive systems," in *ACM Conference on Computer and Communications Security*, 2000, pp. 245–254.

[18] 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)*, 2001, pp. 136–145.

[19] 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.

[20] J. T. A. Perrig, R. Canetti and D. Song, "Efficient authentication and signing of multicast streams over lossy channels," in *+IEEE Symposium on Security and Privacy*, 2000, pp. 56–73.

[21] D. Beaver, "Foundations of secure interactive computing," in *Proceedings of Advances in Cryptology (CRYPTO '91), ser. LNCS, vol. 576, Springer*, 1992, pp. 377–391.

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

[23] B. P. 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, pp. 336–354.

[24] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, pp. 198–207, 1983.

[25] J. R. Douceur, "The Sybil attack," in *Proceedings of the International Workshop on Peer-to-Peer Systems (IPTPS 2002)*, 2002, pp. 252–260.

[26] 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*, 1993, pp. 35–54.

[27] 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*, vol. 5, no. 3, pp. 286–297, 2007.

[28] 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.

[29] J. Hall, M. Barbeau, and E. Kranakis, "Enhancing intrusion detection in wireless networks using radio frequency fingerprinting," in *IASTED International Conference on Communications, Internet, and Information Technology, 2004, St. Thomas, US Virgin Islands*, 2004, pp. 201–206.

[30] Y.-C. Hu, A. Perrig, and D. Johnson, "A survey of secure wireless ad hoc routing protocols," *IEEE Security and Privacy Magazine*, vol. 2, no. 3, pp. 28–39, May/June 2004.

[31] R. Perlman and C. Kaufman, "Key exchange in IPSEC: Analysis of IKE," *IEEE Internet Computing Magazine*, vol. 4, no. 6, pp. 50–56, 2000.

[32] M. Burmester and T. van Le, "Secure multipath communication in mobile ad hoc networks," *ITCC,*, vol. 02, pp. 399–405, 2004.

[33] M. Burmester, T. van Le, and M. Weir, "Tracing Byzantine faults in ad hoc networks," in *Proc. Computer, Network and Information Security 2003, New York*, 2003, pp. 43–46.