Can Ad Hoc Routing Protocols be Shown Provably Secure?

Todd R. Andel *
Computer Science Department
Florida State University

Abstract. Formal security analysis of ad hoc routing protocols has historically been lacking, normally following non-formal inspection to identify flaws and attacks. This leads to a cycle of published development, attack, development, attack, … . Recent formal models and methods attempt to introduce rigor into the analysis approach. We discuss flaws in a previously proposed security framework based on the simulatability model, commonly used in cryptographic proofs. We show how improper application and inappropriate assumptions lead to our discovery of an attack on both the endairA and ARAN protocols, which had been previously shown to be provably secure under this model. Instead of attempting to prove security of ad hoc routing protocols, we propose a framework for analyzing attacks and vulnerabilities. This alleviates the need for a common definition of routing security or the restriction of bounding the problem domain by unattainable assumptions.

Keywords. MANETs, Provable Security, Secure Routing Protocols, Simulatability

1 Introduction

In the late 1990's and early 2000's research into routing protocols for mobile ad hoc networks (MANETs) primarily focused on functionality and efficient operation of wireless multi-hop networks. Royer and Toh [1] provide a survey which discusses many of the proposed protocols for this area.

Security of these protocols, once an afterthought, is now being addressed as many proposed secure ad hoc routing protocols are being researched (see the survey by Hu and Perrig [2]). The majority of these secure protocols are based on the Dynamic Source Routing (DSR) [3] or the Ad Hoc On-Demand Distance Vector (AODV) [4] protocols. Unfortunately, the analysis of ad hoc routing protocol security features is typically informal and tends to consistently result in a stamp of approval from the protocol developer, as they claim to show their protocol has met their definition of security. The effect is a false sense of security, and many of these "secure" protocols are later shown to have flaws. One reason for this is that there does not seem to be any standard definition of secure routing. Each author tends to define security within their own boundaries or assumptions to meet their needs.

There have been recent attempts to develop formal analysis models to prove (or disprove) routing protocol security. Yang and Baras [5] focus on modeling insider attacks against Secure AODV (SAODV) [6] by adding extensions to the strand spaces [7] formal method and use of the Athena [8] model checker. Marshall [9] performed an analysis of

* The views expressed in this article are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

the Secure Routing Protocol (SRP) [10] using a formal method tool package called CPAL-ES [11]. We focus our analysis in this paper on the simulatability model proposed by Buttyán, Vajda, and Ács [12,13], based on classical proofs in cryptographic session key operations. This model is a valiant attempt to prove security properties, but falls short of its intended goal.

Section 2 of this paper provides background on the simulatability model proposed by Buttyán, Vajda, and Ács. In Section 3 we discuss failures in this model and describe an attack on the endairA and ARAN protocols, which had been previously shown to be secure in the model of Section 2. In Section 4 we propose a framework as a basis for security comparison between MANET protocols. Section 5 lists our conclusions and future work.

2 Simulatability Model Background

On-demand source routing protocols, such as DSR [3], depend on a route discovery process, allowing the source to identify an entire path to a given destination. This path is a significant security issue, in that if it is corrupted the ensuing data transmission attempting to use the path will fail. Secure routing protocols attempt to deal with protection of these predetermined paths, however obscure attacks usually render such protocols insecure after they have been published. Buttyán and Vajda [12] contend these undiscovered flaws primarily result from the lack of a common definition of secure routing, and that there is not a formal or mathematical process to analyze such routing protocols for security flaws. The formal model they propose attempts to *"…make the first steps towards a formal model in which one can precisely define what secure routing means and prove (or fail to prove) that a given protocol indeed satisfies that definition."* [12]

2.1 Model Description

Their proposed model [12] follows a common simulatability technique historically used to analyze cryptographic proof systems. The model consists of two separate yet nearly identical models referred to as a real-world model and an ideal-world model. Both models consist of basic Turing machine components to simulate an on-demand source routing protocol for an instance in time (i.e. mobility is not accounted for).

The composition of the ad hoc network is represented by the undirected graph $G = (V, E, L)$, where V is the set of nodes, E is the set of edges, and L is a labeling function which assigns each node a unique identifier (e.g. node A will typically be identified by the label A). For each node $v \in V$, the neighborhood determines which nodes are within transmission distance of the sender and is formally defined as $N_G(v) = \{v' : (v, v') \in E\}$. The configuration, formally defined as $conf = (G, \tilde{v})$, represents the location of the adversarial node $\tilde{v}$. The work in [13] extends the model to allow multiple, possibly colluding, adversarial nodes.

Figure 1, based on [13], depicts the various Turing machines used to simulate the given protocol. Machine H models the upper protocol layers, initiates routing requests, and collects identified routes. Machines M_1 to M_n model the operation of the non-corrupted nodes according to the given protocol. Machine C models the communication

channels by taking the output of each node and delivering it to the respective neighbors as defined by $\mathcal{N}_G(v)$. Machines $A_1 - A_m$ describes the adversarial nodes. The adversaries have the same capabilities as non-corrupted nodes, but are not required to follow the routing protocol. This allows adversaries to change the embedded path in the route request or route reply messages, drop messages, or fake messages. The machines in the shaded area are controlled by a higher machine T in the ideal-world model. Machine T works as an oracle which analyzes returned routes and invalidates routes that are not possible, according to the complete knowledge of possible paths defined by graph *G*. The real-world model utilizes the individual machines, without the use of the oracle T, thus invalid routes may exist.

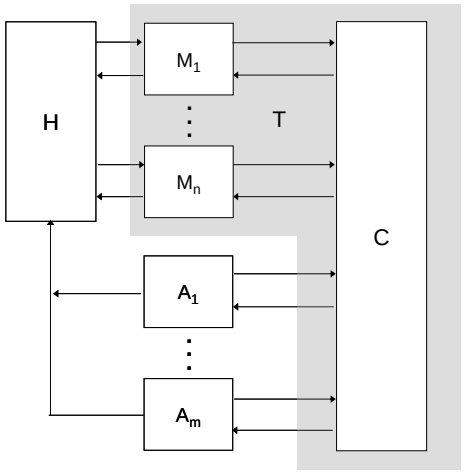

Fig. 1. Simulatability Model adapted from [13]. The shaded area T models the ideal-world as an oracle.

2.1.1 Model Assumptions

The model presented in [12,13] is dependent on numerous assumptions. The adversary nodes are assumed to have equal communication capabilities as non-corrupted nodes. While this constrains communications within normal transmission distances, out-of-band channels, or tunnels, may be used when multiple adversary nodes are present. Cryptographic material, such as signature keys, is assumed to be compromised on corrupted nodes. In scenarios with multiple colluding nodes, the compromised cryptographic material may be shared among the nodes. This allows the model to view any adversarial nodes within direct communication distance to be modeled as a single vertex in the network graph. We show in Section 3 how this combined view can lead a node into a false belief of network connectivity and uncover an attack on the endairA protocol, which is shown to be provably secure in this simulatability model.

Additional assumptions include the exclusion of the Sybil attack [14] and network wormholes [15]. The authors claim the Sybil [1] attack cannot occur since nodes are linked to their unique identifiers during neighborhood discovery. Similarly, they claim that

[1] Sybil attacks allow an attacker to appear as more than one node.

wormhole [2] attacks are eliminated though the use of temporal or geographical packet leashes.

2.1.2 Model Operation

During operation, each Turing machine of the simulatability model follows the process of reading the input buffer, updating states, and writing to the output buffer. The machines follow a round robin schedule of H, M_1, …, M_n, A, C. The final simulation output consists of the returned routes (i.e. Out_{real} or Out_{ideal}). Additional functionality in the ideal-world model consists of checking a corruption flag. The ideal-world model analyzes returned routes against its complete knowledge of graph *G*. If the route is not valid, the corruption flag is set and the route is not added to the routes reported in Out_{ideal}.

Route security, as defined in this model [12,13], requires that all returned routes must exist as valid paths in *G*. This does not protect against having an adversary in the path, since compromised nodes may operate completely normal until they decide to start dropping packets. It does though eliminate the ability of a compromised node, or colluding nodes, to return a path that does not exists in the first place. Since Out_{ideal} will never output non-existing routes, [12,13] considers a routing protocol to be provably secure if $Out_{real} = Out_{ideal}$.

2.2 Uncovering Attacks

The single adversary model in [12] contends that use of the model has discovered unknown attacks against the Secure Routing Protocol (SRP) and Ariadne.

SRP is presented in [10] as an attempted security extension to on-demand source routing protocols, such as DSR. SRP depends on a preexisting security association between a source and destination pair. The source uses the security association to generate a keyed message authentication code (MAC) to sign the source-destination pair. The destination verfies the MAC with the share secret and produces a new MAC of the complete path as delivered in the route request and returns in the route reply to the destination. The attack in [12] shows that an appropriately positioned adversary can change the path (e.g. add non-existent nodes) during route discovery and relay the corrupted signed path to be delivered to the source.

Ariadne [16] is another attempted security extension to DSR by authenticating each intermediate node within the path, as opposed to the single source-destination authentication used in SRP. In Araidne, each intermediate node digitally signs or appends a MAC on a per-hop basis during the forward path of route discovery. Once the destination receives the route request, the target digitally signs or MACs the received path and returns this to the sender. The attacks in [12,13] show that an appropriately positioned adversary node (or nodes) can corrupt the route path and calculate new verifiable signatures or MACs that are accepted and validated by the destination. Since the intermediate nodes do not individually validate the returned route reply the inappropriate path will not be de-

[2] Wormhole attacks occur when two or more colluding nodes tunnel packets or a single node relays packets to trick a source and destination pair into believing a shorter path exists.

tected, allowing intermediate adversaries to adapt and spoof the return path as needed to reach the source. The adversaries ensure that the path embedded in the final message delivered to the source will contain the validated signed path sent by the destination. In this case, non-existent routes are returned to the source.

In the updated model presented in [13] the authors concede that the model itself is not directly useable to detect unknown attacks in protocols, but attacks may be discovered as a side effect of attempting to prove security in their model. Additional informally developed (e.g. through visual inspection) attacks on Ariande are also presented, consisting of multiple attackers with and without multiple corrupted cryptographic keys.

In all cases, the attacks, once discovered, show SRP and Ariadne are not secure in the simulatability model, since the ideal-world model will reject invalid routes and the real-world model will not. That is, $Out_{real} \neq Out_{ideal}$.

2.3 endairA Protocol

The protocol endairA was proposed to provide a provably secure routing protocol in the context of the described simulatability model [12,13]. endairA is an adaptation of Ariadne, in which the intermediate nodes digitally sign the route reply instead of the route request. In effect, it is Ariandne in reverse, as its name implies. As the route reply is returned, each node checks if it is in the path, checks that the predecessor and successor nodes are neighbors, digitally signs the message (signature is over the entire message to include path and previous signatures), and forwards to the next node on the path back to the source node. Upon receipt of the route reply, the source node ensures it was delivered by a neighbor and validates all signatures against the returned path. Figure 2 represents an example route discovery as presented in [13].

$S \rightarrow *$	:	(rreq, *S*, *T*, *id* ())
$A \rightarrow *$	:	(rreq, *S*, *T*, *id* (*A*))
$B \rightarrow *$	:	(rreq, *S*, *T*, *id* (*A*, *B*))
$T \rightarrow B$	:	(rrep, *S*, *T*, (*A*, *B*), (sig_T))
$B \rightarrow A$	:	(rrep, *S*, *T*, (*A*, *B*), ($sig_{T,}$ sig_B))
$A \rightarrow S$	:	(rrep, *S*, *T*, (*A*, *B*), ($sig_{T,}$ $sig_{B,}$ sig_A))

Fig 2. endairA protocol provided by [13]. rreq is a route request, rrep is a route reply. Each node appends a signature *sig* over the entire received message of the route reply.

The authors contend that endairA is provably secure in their simulatability model since the corruption flag will never be set in the ideal-world model. Thus for all iterations, $Out_{real} = Out_{ideal}$, which meets their definition of a provably secure routing protocol. In Section 3 we show an attack on endairA is possible, thus it is not provably secure in this sense.

2.4 Extending the Model

The model in [12,13] is extended in [17] to support on-demand distance vector protocols, such as AODV [4]. Unlike source routing protocols, where packets contain an entire source-to-destination path, on-demand distance vector protocol packets contain only the source and destination portions of the path. Each node is independently responsible to forward packets to the next node in the path according to local routing table information (e.g. next hop).

2.4.1 Model Changes

The model now focuses on the correctness of each non-corrupted node's routing tables, as opposed to correctly reported route paths. A simplified routing table consists of entries containing the triple (destination, next hop, cost), where cost is typically the number of hops left to the destination. The entire system state of all routing tables (in non-corrupted hosts) is tracked as entries according to the 4-tuple (v, destination, next hop, cost), for each route entry for every $v \in V$. This is reported as Out_{real} for the real-world model. The ideal-world model uses the full knowledge of G to eliminate incorrect entries and report $\text{Out}_{\text{ideal}}$ appropriately. For a state entry to be correct, a path must exist for each entry $v \in V$, that goes to the intended target, through the intended hop, with a cost $\leq$ the identified cost.

2.4.2 Attacking SAODV

SAODV [6] is an adaptation of AODV, attempting to provide security of local routing table information in ad hoc networks. SAODV packets have both a mutable (changeable) and non-mutable (unchanging) portions. The non-mutable data (e.g. source, destination, payload) is protected via digital signatures. During route discovery and packet transmission the hop-count embedded within the packet must continually change, therefore it remains a mutable element. A secure hash chain is used to protect the hop-count, based on an initial hash value of a random seed and the maximum hop-count value. Each intermediate node checks the current hash, updates the hop-count, and computes a new hash based on the difference between the initial hash and the current hop count.

Two attacks on SAODV are presented in [17]. The first attack is based on the fact that a node may forward a route request without increasing the hop count, thus subsequent nodes will set up reverse routes to the adversaries upstream neighbor with too low of a hop-count. In the second attack, the hop-count remains correct, but an adversary can forward a route reply in the name of the another node (via spoofing). This results in a routing table entry for a next-hop node that is not part of the path.

In both cases, the model representation shows that the real-world state of route tables contain invalid routes, while the ideal-world state drops the corrupt results according to the full knowledge of graph G. The result is an insecure protocol since $\text{Out}_{\text{real}} \neq \text{Out}_{\text{ideal}}$.

2.4.3 Proving ARAN Secure

Authenticated Routing for Ad hoc Networks (ARAN) is another attempted secured on-demand distance vector routing protocol [18]. ARAN does not use the mutable hop-count for routing decisions, therefore each node produces a signature on the entire received packet during route request and reply packets. During route discovery each node checks the signature of the last transmission, if valid it adds a reverse path entry in the local routing table, computes a local signature over the entire received packet, and rebroadcasts (unicasts if a route reply) the discovery packet.

For the ideal-world ARAN representation, the authors in [17] claim the only possible reasons for an incorrect state in any table entry of (*v*, destination, next hop, cost) are: a route does not exist to the destination in graph *G*, there are no routes from *v* to the destination via the listed next hop, and routes that exist with a higher cost than listed. Note that ARAN doesn't directly use a cost metric such as a hop, however the quickest route is returned which can be viewed as a transmission delay cost in the model. They contend that with digital signatures, there is negligible probability (e.g. someone guesses a key) that the ideal-world model would encounter a corrupt table entry to drop. This results in their model showing provable security since $Out_{real} = Out_{ideal}$. In Section 3 we show an attack on ARAN is possible, thus it is not provably secure in this sense.

3. Model Failures

We observe that the original assumptions disregarding wormhole and Sybil attacks are inadequate when colluding nodes are present, which impacts a nodes defined network view. We show how this tainted network view leads to attacks on the endairA and ARAN protocols that were previously shown provably secure with the simulatability model of [12,13,17].

Unfortunately, the simulatability model presented in section II cannot provide provable security. As [13] points out, the model cannot be used to directly discover attacks. However, they claim use of the model in proof attempts may find flaws as a side-effect. We observe that the only way this model could be used in operation is to exhaustively simulate all possible configurations. More significantly, flaws in the model itself render it invalid for use in analyzing and proving the security of routing protocols.

3.1 Improper Assumptions

Typical on-demand source routing protocols, such as DSR, SRP, or Ariadne, do not gather or use local neighbor information. That is one of their attractive features, in that no periodic neighbor updates (or *hello*) messages are required. This reduction in network transmissions reduces network congestion and saves precious battery power in wireless nodes. The endairA protocol requires the use of local neighborhood information to decide if a packet could have been sent by the node which claims to have sent it, and checks if the next node the packet is to be forwarded to is also a neighbor (recall each intermediate node checks the path in the route reply).

The mechanism endairA used to gather this neighbor information was not specified in [12,13]. We assume that source *S* learns of a neighbor promiscuously when the neighbor forwards a route request. Unfortunately, contention or reception errors at promiscuously listening *S* may drop one of these packets. Since these packets were not directed (i.e. unicast) to *S*, link-layer retransmission schemes would not ensure the route-request (intended to be passed toward the destination) would be retransmitted to *S*, thus *S* would not become aware of this neighbor.

For the purposes of this discussion, we will assume that *S* does have knowledge of its local neighbors as anticipated in [13]. Unfortunately, we do not agree that Sybil and wormhole attacks cannot occur as contended. Once multiple colluding adversaries share cryptographic materials, neighborhood discovery mechanisms can be forged or spoofed, since any adversary can sign and authenticate as any node for which it holds authentication keys. Colluding nodes may share compromised keys via out-of-band transmissions or by encrypting ordinary communication channels between corrupted nodes. Thus with colluding adversaries, a node can claim to be multiple nodes (i.e. the Sybil attack). Additionally, wormhole attacks are possible when authentication keys are shared. Packet leashes embed either time (temporal) or positional (geographical) information to detect a wormhole between two nodes or a simple node relaying as an invisible host. Without node authentication, the information embedded into the packet leash can be corrupted [15].

3.2 Attacking endairA

To accommodate the assumptions of [13], the authors combine all adversary nodes that can build a direct, adversarial only, communication path into a single adversary. Figure 3 shows an actual network view as viewed in their model. Figure 3a depicts actual network connectivity based on communication range. Figure 3b incorporates *X* and *Z* (neighboring adversaries) into the single adversarial node *u**. The adversarial node Y (referred to as *v**) is disjoint from *u**, since it is not a direct neighbor of any node in *u**. Also, nodes *u** and *v** have previously shared authentication keys from prior connectivity.

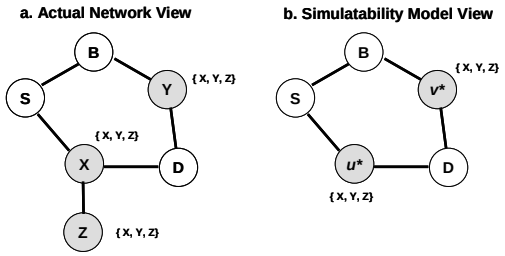

Fig. 3. Network connectivity views. Shaded nodes are corrupt. {*X*, *Y*, *Z*} indicates shared authentication keys (labels). View b. is a simplified adaptation of fig. 3 in [17].

Unfortunately, the view of figure 3b. benefits the adversary only. The adversary can utilize the out-of-band network to allow *Z* to observe the network with *Y*'s view. This view does not benefit the non-corrupted node in any way. The adversary *X* can use the

shared corrupted keys during any neighbor discovery (e.g. a Sybil attack) to lead both *S* and *D* into believing corrupted views of their connectivity as shown in figure 4.

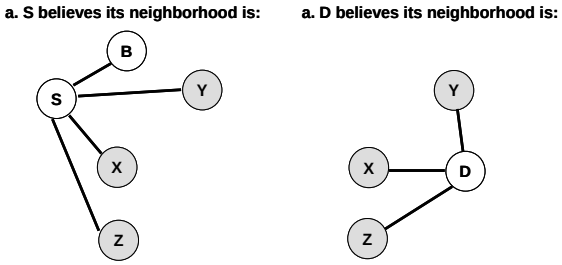

Fig. 4. Corrupted neighborhood views. Shaded nodes are corrupt.

The ability to masquerade, or spoof, as another node allows dishonest nodes to corrupt route paths. Figure 5 provides an attack on endairA in the context of the true network configuration of figure 3a and the corrupted views of figure 4. The example shows how node *X* can utilize the shared corrupted key from node *Y* to supply *S* with a non-existent route (since it can authenticate messages as *Y)*. *S* has been tricked into believing the path is *S-Y-D*. Even though *X* acts as *Y* during the route discovery, *X* is not obliged to act as *Y* during data transmission, but can act as the valid node *X* in other route paths to guard itself from being implicated as corrupt.

$S \rightarrow *$	:	(rreq, *S*, *D*, *id* ())
$X \rightarrow *$	:	(rreq, *S*, *D*, *id* (*Y*))
$D \rightarrow Y$	:	(rreq, *S*, *D*, (*Y*), (sig_D))
$X \rightarrow S$	:	(rreq, *S*, *D*, (*Y*), ($sig_{D,}$ sig_Y))

Fig 5. An attack on endairA. *X* and *Y* are colluding adversaries,. *sig* is the respective signature. rreq is a route request, rrep is a route reply. *X* tricks *S* into accepting the non-existent path *S-Y-D*.

Figure 6 presents the corrupted real-world view that generates Out_{real} vs. the actual network connectivity view that generates Out_{ideal}. The virtual connectivity in the real-world model (due to spoofing) results in $Out_{real} \neq Out_{ideal}$.

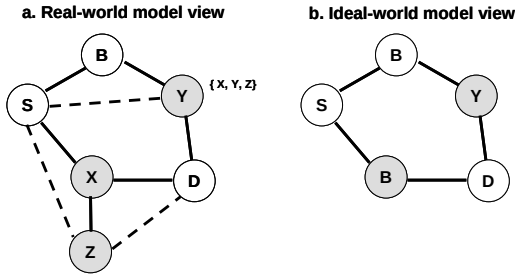

Fig. 6. Real vs. Ideal Network Connectivity. Shaded nodes are corrupt. Solid lines are actual links, dashed lines are virtual links.

3.3 Attacking ARAN

Our attack on ARAN follows the same reasoning developed in our endairA attack. Since the colluding nodes share authentication keys, a corrupted intermediate node can sign and authenticate as another adversary. Figure 7 shows our ARAN attack in the context of the network depicted in figure 3a. We assume that *Y* allows a valid route request to be set up with destination *D*. On the route reply *Y* tricks *B* into accepting the route entry (*D, Z,* 2), indicating B believes it has a path to *D* through node *Z* with a cost of 2. *Y* is not obliged to act as *Z* during data transmission but can act as the valid node *Y* in other route paths to guard itself from being implicated as corrupt.

$S \rightarrow *$	:	(rreq, D, $cert_S$, N_S, t, sig_S)
$B \rightarrow *$	:	(rreq, D, $cert_S$, N_S, t, sig_S) sig_B, $cert_B$
$Y \rightarrow *$	:	((rreq, D, $cert_S$, N_S, t, sig_S) sig_B) sig_Y, $cert_Y$
$D \rightarrow Y$	:	(rrep, S, $cert_D$, N_S, t, sig_D)
$Y \rightarrow B$	:	(rrep, S, $cert_D$, N_S, t, sig_D) sig_Z, $cert_Z$
$B \rightarrow S$	:	((rrep, S, $cert_D$, N_S, t, sig_D) sig_Z) sig_B, $cert_B$

Fig 7. An attack on ARAN. rreq is a route request, rrep is a route reply. cert and sig are the respective certificates and signatures. *Y* tricks *B* (at the shaded entry) into accepting the non-existent route entry (*D*, *Z*, 2).

3.4 Improper Application

The simulatability model of section II does not meet the authors' intentions in [12,13, 17]. One of their main purposes was to alleviate the problem of overlooking subtle flaws in the current non-formal methods used during protocol security analysis. The attacks we presented on the endairA and ARAN protocols, both shown to be provably secure in this model, are subtle attacks due to improper application of the model's network view.

The fundamental flaw in the presented approach is that the real-world and ideal-world model are identical, with the exception of the analysis of the corruption flag in the ideal case. If an ideal-world model representation is resistant to the corruption flag being set, the result is that $Out_{real} = Out_{ideal}$ will always be true. This simply transfers the problem of finding subtle flaws in the real-world model to finding subtle flaws in the ideal-world model. This can be done only with an exhaustive search on all possible configurations. Therefore, the model is only useful to formally describe previously discovered flaws in the context of the model. That is, any known flaw always results in $Out_{real} \neq Out_{ideal}$.

4. Comparison Framework

Proving security of ad hoc routing protocols is a challenging task. Typically, security protocols under development are deemed secure by the designers, later to be shown insecure in subsequent publications. Many times, as we have shown here, improperly applied assumptions mask very significant attacks.

We contend there is no method to ensure a protocol is completely secure, we can never know if an unknown attack will render unknown vulnerabilities in a protocol. Take for

instance an attacker that has compromised a node. The attacker can operate normally until it finally decides to start dropping or corrupting packets. Some may argue that we can't defend against this so we should not consider this part of security analysis. This again reflects the problem in multiple definitions of protocol security. We should not be asking if a routing protocol is provably secure. The real question should be if any such protocol is physically implemented and used in operation, in what environments will it be secure and to what vulnerabilities do we know a protocol will fail?

Instead of attempting to prove routing protocol security, we propose a framework to rate protocols against what vulnerabilities we know they fail against. While this approach can not claim a protocol is vulnerability free, it does allow protocol comparison for attacks we have found. This is similar to the approach taken on the security of the Advanced Encryption Standard (AES). It is assumed to be secure since a successful attack has not yet been found, however it is not known if an AES is vulnerability free.

Rather than discussing specific attacks and assumptions, our framework uses attacker capabilities (i.e. the impact) and is not based on assumptions that bound operational environments. It is based on the active attacker hierarchy proposed in [16]. They define an attacker as active-n-m, where n is the number of compromised insider (Byzantine) nodes and m is the number of outsider nodes. We do not consider passive attackers since they this is a data privacy issue does not pose a threat to routing security.

Table 1. Routing Vulnerability rating Framework

Attack Classification	Protocol A	Protocol B
Ideal (no attackers – baseline)	S	S
External attackers (non-trusted)		
- Single attacker		
-- Add self to route	V	S
-- Corrupt/change route	V	S
-- Return shorter route (relay)	V	V
- Multiple attackers		
-- Corrupt/change route	V	S
-- Return shorter route (relay)	V	S
Internal attackers (compromised/trusted)		
- Single attacker		
-- Corrupt/change route	V	V
-- Return shorter route (relay)	V	V
- Multiple attackers		
-- Corrupt/change route	V	V
-- Return shorter route (relay)	V	V
- Identify thresholds		
-- Determine the threshold of corrupt nodes under which a protocol can find a non-corrupted route (i.e. mitigation)	UNK	n/3

Table 1 presents our proposed framework to rate ad hoc routing protocols. An entry marked as S indicates that the representative attack has not been shown for that protocol, V represents the a known vulnerability, and UNK indicates that the analysis has not yet been performed. Protocol A and B are non-existent protocols used for demonstration proposes only. Since internal attackers can act as valid nodes and discontinue routing operations at the time of their choosing, protocol analysis at this junction should look at

mitigation and threshold strategies to alleviate routing misbehavior and the ability to identify adversaries and re-establishment of adversarial free routes. This technique is currently being address in various protocol extensions such as watchdog-pathrater [19] and the On-Demand Secure Byzantine Routing (ODSBR) protocol [20].

While following this framework may seem tedious, it enforces the necessity for research into formal methods that may be automated to analyze against specific attack classes, using such tools as Athena [5,8] or CPAL-ES [9,11].

5. Conclusion

We have shown the need for formal models to analyze the security of ad hoc routing protocols. We took a critical look at the proposed simulatability model of [12,13,17] for use in proving the security of ad hoc routing protocols, concluding that improper assumptions during analysis can lead to omission of attacks. The improper application of the simulatability technique and combined inappropriate assumptions led to our discovery of attacks on both the endairA and ARAN protocol, that had been shown to be provably secure in this model.

Can ad hoc routing protocols be shown provably secure? We contend no. The provability of any routing security property is directly related to the context (i.e. security definition) or bounds implemented by any associated assumptions. We should be more interested in how proposed secure routing protocols operate against various risks they may encounter. We propose a comparative framework to formally document and rate security of ad hoc routing protocols. This allows the selection of a given protocol to meets the needs of the indented real-world operation environment.

Our future work includes research into formal models and automated mechanisms allowing comparative security analysis of proposed ad hoc routing security protocols.

6. Acknowledgements

The author thanks Mike Burmester and the Spring 2006 CIS 5930-MANETs class for comments and helpful discussions.

References

1 Royer, E.M., Chai-Keong, T.: A review of current routing protocols for ad hoc mobile wireless networks. IEEE Personal Communications **6** (1999) 46-55

2 Hu, Y.C., Perrig, A.: A survey of secure wireless ad hoc routing. IEEE Security & Privacy **02** (2004) 28-39

3 Johnson, D., Maltz, D.: Dynamic source routing in ad hoc wireless networks. In: Imielinski, T., Korth, H. (eds.): Mobile Computing. Kluwer (1996) 153-181

4 Perkins, C.E., Royer, E.M.: Ad-hoc on-demand distance vector routing. 2nd IEEE Workshop on Mobile Computing Systems and Applications (1999) 90-100

5 Yang, S., Baras, J.S.: Modeling vulnerabilities of ad hoc routing protocols. 1st ACM Workshop on Security of Ad hoc and Sensor Networks (2003) 12-20

6 Zapata, M.G., Asokan, N.: Securing ad hoc routing protocols. 3rd ACM Workshop on Wireless Security (2002) 1-10
7 Thayer Fábrega, F.J., Herzog, J.C., Guttman, J.D.: Strand spaces: Proving security protocols correct. Journal of Computer Security **7** (1999) 191-230
8 Xiaodong Song, D., Berezin, S., Perrig, A.: Athena: A novel approach to efficient automatic security protocol analysis. Journal of Computer Security **9** (2001) 47-74
9 Marshall, J.: An Analysis of the Secure Routing Protocol for mobile ad hoc network route discovery: using intuitive reasoning and formal verification to identify flaws. Department of Computer Science. Florida State University, Tallahassee, FL (2003)
10 Papadimitratos, P., Haas, Z.J.: Secure routing for mobile ad hoc networks. SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002), San Antonio, TX (2002)
11 Yasinsac, A., Wulf, W.A.: A Framework for A Cryptographic Protocol Evaluation Workbench. The International Journal of Reliability, Quality and Safety Engineering (IJRQSE) **8** (2001) 373-389
12 Buttyán, L., Vajda, I.: Towards provable security for ad hoc routing protocols. 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks, Washington DC, USA (2004) 94-105
13. Ács, G., Buttyán, L., Vajda, I.: Provably secure on-demand source routing in mobile ad hoc networks. IEEE Transactions on Mobile Computing (to be published)
14 Douceur, J.R.: The Sybil attack. 1st Intl. Workshop on Peer-to-Peer Sys. (IPTPS 2002) (2002)
15 Hu, Y.C., Perrig, A., Johnson, D.B.: Packet leashes: a defense against wormhole attacks in wireless networks. INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (2003) 1976-1986
16 Hu, Y.C., Perrig, A., Johnson, D.B.: Ariadne: a secure on-demand routing protocol for ad hoc networks. 8th Annual International Conference on Mobile Computing and Networking (MobiCom '02) ACM Press, Atlanta, Georgia, USA (2002) 12-23
17 Ács, G., Buttyán, L., Vajda, I.: Provable security of on-demand distance vector routing in ad hoc networks. European Workshop on Security and Privacy, LCNS Vol. 3813. Springer-Verlag, New York (2005) 113-127
18 Sanzgiri, K., Dahill, B., Levine, B.N., Shields, C., Belding-Royer, E.M.: A secure routing protocol for ad hoc networks. 10th IEEE International Conference on Network Protocols (2002) 78-87
19 Marti, S., Giuli, T.J., Lai, K., Baker, M.: Mitigating routing misbehavior in mobile ad hoc networks. 6th Annual International Conference on Mobile Computing and Networking (2000) 255-265
20 Awerbuch, B., Holmer, D., Nita-Rotaru, C., Rubens, H.: An on-demand secure routing protocol resilient to byzantine failures. 3rd ACM workshop on Wireless security, Atlanta, GA, USA (2002) 21-30