Secure On-Demand Distance Vector Routing in Ad Hoc Networks

Panagiotis Papadimitratos
Electrical and Computer Engineering
Virginia Tech
papadp@vt.edu

Zygmunt J. Haas
Electrical and Computer Engineering
Cornell University
haas@ece.cornell.edu

***Abstract*— We address the problem of securing the route discovery in mobile ad hoc networks, proposing a light-weight yet robust routing protocol, the Distance-Vector Secure Routing Protocol (DV-SRP). DV-SRP discovers on-demand multiple routes, which are established across the network, without providing explicitly the network connectivity. DV-SRP combines the advantages of the type of route discovery first introduced by AODV, with security and thus resilience to adversaries that disrupt the route discovery. Compared to previous proposals in the literature to secure the AODV-like route discovery, DV-SRP is either more robust, or more efficient, or more general.**

I. Introduction

Independently of security considerations, a multitude of routing protocols has been proposed, with a number of those heading towards standardization through the *MANET WG* [2]. Different protocols are shown to be appropriate for different network settings, and, clearly, there may not be a single protocol that outperforms all alternative ones in all settings.

Our focus in this paper is the reactive route discovery that resembles the *Ad hoc On-demand Distance Vector* (*AODV*) protocol [1], whose operation can be advantageous: nodes maintain information only for destinations they actively communicate with, and the network overhead per route request and reply packet does not increase with the number of nodes along a discovered route. From a different point of view, another advantage is the improved concealment of topological knowledge, as the control and data packets do not carry the links of the discovered or utilized routes.

What we are after here is to ensure the correctness of the route discovery. Two proposals in the literature secure the *AODV*-like route discovery. The *Secure AODV* (*S-AODV*) [5] is a secure version of *AODV*, proposing a combination of public key cryptography and hash chains. The *Authenticated Routing for Ad hoc Networks* (*ARAN*) [4] is a secure protocol that resembles somewhat *AODV*, but it does not seek to provide *AODV*'s features.

In this paper, we propose the *Distance-Vector Secure Routing Protocol* (*DV-SRP*), which is either more robust, or more efficient, or more general than previous proposals to secure this type of route discovery. *DV-SRP* prevents adversaries from manipulating the length (hop count) of the discovered routes, uses primarily symmetric key and thus low cost cryptographic primitives, and discovers multiple routes. Next, we discuss the network and adversary models. Then we present the operation of *DV-SRP*, followed by an analysis and discussion.

II. Network and Adversary Model

A network *node* is a process with a unique identity *V*, a public/private key pair E_V, D_V, a module implementing the networking protocols, such as *DV-SRP*, and a module providing communication across a wireless network interface, e.g., based on the widely adopted *IEEE 802.11* [7]. We are concerned with pair-wise communication across multiple wireless links between a *source S* and a *destination T*. We denote *S* and *T* as the *end nodes*, and nodes that assist the *S, T* communication as *intermediate nodes*. We assume that each end-node knows the identity and the public key of its peer end-node, and all nodes know the identities and the public keys of their neighbors. These, as well as the establishment of symmetric shared keys by the end nodes or two neighbors, can be achieved by protocols such as the *Neighbor Lookup Protocol* (*NLP*) or mechanisms that are part of the *Secure Routing Protocol* (*SRP*) [13], [8], [3].

We consider two models of *active* adversaries, *independent* adversaries and *arbitrary* adversaries [13]. *Independent adversaries* can modify, forge, or replay routing or data packets, but ignore received traffic that does not comply with the operation of the networking protocols, and thus do not generate any message due to the receipt of such traffic. Any message that does not follow the expected, protocol-specific format or fails one of the protocol checks is deemed as non-compliant. We emphasize that traffic is non-compliant if and only if the receiving node can detect that a message does not comply with the protocol; otherwise, messages that appear to be compliant, but actually are not, are processed as compliant. If non-compliant traffic is attributed to misbehavior, independence implies that adversaries do not process and relay traffic that appears to originate from or have been previously relayed by an adversary. In other words, independent adversaries do not attempt to assist other adversaries mounting an attack, either by ignoring the attack and further relaying traffic or by not responding to received non-compliant traffic.

This model of failures allows for a range of malicious behaviors and it is more general than *crash*, *omission* failures, and *timing* failures [10], [11]. Even though the malicious behavior of independent adversaries is constrained, the model does not prevent adversaries from simultaneously launching their attacks, which may have a compound effect. As it will become clear, the model of independent adversaries serves as a necessary condition to achieve stronger protocol properties than those achieved without the model's constraint on the adversarial behavior.

In general, adversarial nodes are allowed to deviate from the protocol execution in an arbitrary manner [12]. *Arbitrary adversaries* can be more sophisticated and powerful than independent adversaries, having, for example, knowledge of the identities of other adversaries in the network, devoting resources (e.g., route discovery) to establish direct and possibly private communication with other adversaries, and exchanging traffic and information about their local execution of the protocol.

III. DISTANCE VECTOR SECURE ROUTING PROTOCOL

The *source node* (S) generates a *route query* or *route request* packet (*RREQ*). The route request fields include S, T, a *query identifier* Q that was not previously used, and an authenticator A_{IN}^{S} to authenticate S as the origin of the *RREQ* at each intermediate node. Moreover, the maximum node count field, *MNC*, is set to a protocol-selectable positive integer value, and the maximum hop count field is set to $MHC=h^{MNC}(x_0)$. [1] An authenticator $A=f_K(S, T, Q, A_{IN}^{S}, MHC)$ is calculated as a function of the route query fields and a key K.[2] Finally, the node count field is set to $NC=1$, and the hop count field to $HC=h(x_0)$.

S transmits the route request, i.e., broadcasts it, and initializes an empty *ForwardList* for each *RREQ* it generates, and retains *NC* and *HC*. S adds to the *ForwardList* each neighbor V it overhears relaying *RREQ* with $NC=NC+1$, and $HC=h(HC)$. If either of the previous two equalities does not hold, S ignores the *RREQ*. Each node receiving a *RREQ* determines if its own identity matches the sought destination.

Each intermediate node V_k invokes the *PreviouslySeen(RREQ)* routine[3] to specify if *RREQ* must be processed. If not, the *RREQ* is discarded. Then, V_k checks if $NC \geq MNC$, and if $h^{MNC-NC}(HC)=MHC$. If either test fails, or if $Validate(A_{IN}^{S})$ returns false, it discards *RREQ*. Otherwise, V_k retains the address of its *precursor* V_{k-1},[4] it updates $NC=NC+1$, and $HC=h(HC)$, and relays *RREQ* further. Finally, V_k initializes an empty *ForwardList* for each *RREQ* it relays. It then adds to the *ForwardList* only each neighbor V it overhears relaying *RREQ* with $NC=NC+1$ and $HC=h(HC)$.

Once the *RREQ* reaches the destination, T invokes the *PreviouslySeen*(*RREQ*) routine to check if *RREQ* has been previously processed. If so, the *RREQ* is discarded. Then, T checks if $NC \geq MNC$, and if so, whether $h^{MNC-NC}(HC)=MHC$. It then invokes $Validate(A_{IN}^{S})$, it calculates $f_K(S, T, Q, A_{IN}, MHC)$ and compares it to A; if any of the checks fails, T discards *RREQ*. Otherwise, T retains its predecessor, generates and returns a route reply to S.

The *route reply* (*RREP*) packet fields comprise S, T, Q, and the route length is set to $RL=NC+1$, with *NC* the value of the *RREQ* field. Moreover, values assigned to: (*i*) a (reverse) hop count field, $RHC=h(y_0)$, (*ii*) a maximum hop count field, $MHC=h^{RL}(y_0)$, and (*iii*) a (reverse) node count field, $RNC=1$. Finally, *RREP* includes a (random not previously used) route identifier, $route_{ID}$, and an authenticator A_{IN}^{T} to authenticate T as the origin of the *RREP* at each intermediate node, and an authenticator $A' = f_K(S, T, Q, RL, A_{IN}^{T}, MHC)$. The destination transmits the *RREP* to its neighbor, i.e., the first entry of the *Route* list V_{n-1}.

Each V_k, including S, verifies that its *successor*[5] $V_{k+1} \in ForwardList$. It checks if $RNC \geq RL$, if $h^{RL-RNC}(RHC)=MHC$, and if $RL-RNC=NC_k$, with NC_k the value V_k used when the node relayed the corresponding *RREQ*. If any of the checks fails, or if $Validate(A_{IN}^{T})$ returns false, V_k discards *RREP*. Otherwise, it retains its successor V_{k+1} and $route_{ID}$, it updates $RHC=h(RHC)$ and $RNC=RNC+1$, and relays *RREP* to V_{k-1}. Once *RREP* reaches the source, S calculates $f_K(S, T, Q, RL, A_{IN}^{T}, MHC)$ and compares it to A'. If there is not a match, S rejects the reply. Otherwise, it accepts the reply.

[1] h is a one-way function and x_0 is a random number.

[2] The function f and the key depend on the cryptographic primitives. Similarly to *SRP* [3], K can be a symmetric key shared by S and T.

[3] The *PreviouslySeen*( ) routine can be implemented in different ways, trading off robustness for lower routing overhead, defining different query propagation mechanisms. For *DV-SRP*, only a single copy of each query identified by the S, T, Q triplet is relayed, as long all protocol-specific checks succeed, by each intermediate node. We denote this as the $QueryPropagation_1$ mechanism. When invoked at the destination, *PreviouslySeen*( ) returns true if the *RREQ* is received from a different neighbor.

[4] The node that previously relayed the *RREQ* that is now processed.

[5] The node that relayed the *RREP* that is now processed.

Finally, we discuss the calculation of the A_{IN}^{S} and A_{IN}^{T} authenticators. Those could be digital signatures, even though this would be computationally expensive. To mitigate this cost, a digital signature can be used only during the first route discovery, along with a commitment to a hash chain, $h^{x}(z_0)$, with z_0 a random number. Then, for the *i-th RREQ*, *S* appends $h^{x-i}(z_0)$ to the *RREQ* packet, and all intermediate nodes verify that $h^{i}(h^{x-i}(z_0))$ equals the initial commitment. Similarly, *T* commits to a hash and uses its elements for intermediate nodes to authenticate the origin of the *RREP*. To ensure that all network nodes received the hash chain commitment, *S* can periodically rebroadcast a signed *RREQ* with a commitment to some $h^{x-j}(z_0)$ element; this way, nodes that previously received the commitment can perform a low cost verification (check, in the worst case, if $h^{j}(h^{x-j}(z_0))$ equals $h^{x}(z_0)$), and those that have not, can validate the signature. The hash chain solution, unlike the digital signatures, does not provide integrity of the *RREP* and *RREQ*, with the intermediate nodes possibly relaying corrupted packets. However, such corruption can is eventually detected by the end nodes.

IV. Protocol Analysis And Discussion

DV-SRP ensures that in the presence of arbitrary adversaries no loop includes correct intermediate nodes, and that the route reply is generated by the destination within a (t_1, t_2) interval, where t_1 is the time at which *S* generated the route query and t_2 the point in time at which *S* received the route reply. This implies that a route along which the *RREQ* and *RREP* propagated existed in the network. If not, *S* would have never received *RREP*. However, two or more arbitrary adversaries M_1 and M_2 can 'tunnel' *RREQ*, *RREP* packets to each other across multiple hops. Moreover, with M_1, M_2,…, M_k, $k \geq 2$, arbitrary adversaries forming a path, any of the M_2,…, M_{k-1}, (in general, $k>2$) may deviate from the protocol, and relay protocol packets while not performing the required checks. As a result, in the first case, the number of hops between M_1 and M_2, and, in the latter case, *k-2* hops between M_1 and M_k can be 'hidden' or 'inflated' by the adversaries.

In the presence of independent adversaries, *DV-SRP* provides stronger properties. The number of hops of the discovered route is correct at the source node: an adversary *M* cannot decrease or avoid increasing the route length, and it cannot increase the hop length.

A decrease or no-increase of the length is prevented, because one of *M*'s predecessors will discard the corresponding *RREP*. The increase of the route length is prevented because *M* has to relay *RREQ* with the correct *HC* and *NC* fields. To further illustrate this, *M* could first relay a $RREQ'$ with 'inflated' *HC* and *NC*, so that $RREQ'$ is relayed by *M*'s successor (Recall that *QureyPropagation*$_1$, defined in Footnote 3, is used by *DV-SRP*). Then, *M* relays a *RREQ* with field values its predecessor deems compliant and then, and later *M* relays the incorrect $RREP'$. However, *M*'s predecessor can 'blacklist' *M* because of $RREQ'$, and ignore (discard) any *RREP* from *M* bearing the same query identifier with $RREQ'$.

Regarding the two types of attacks a group (of two or more) arbitrary adversaries could mount, for independent adversaries, M_3, for example, ignores any non-compliant *RREQ* it receives from M_2, and similarly for the tunneling attack, M_2 (M_1) ignores traffic received from M_1 (M_2) as non compliant. In general, in the presence of independent adversaries,

If A_{IN}^{T} prevents modification of any field other than *RHC* and *RNC* (which is the case for A'), the intermediate nodes can obtain the correct hop count to the destination. However, an intermediate node V_k cannot have the correct length of the route prefix, because any V_{k-i} can manipulate the *NC*, *HC*, and *MHC* fields, causing, nevertheless, either *T* to discard *RREQ* or, later, V_{k-i-1} to discard *RREP*.

The A_{IN}^{S} and A_{IN}^{T} authenticators prevent two (arbitrary) adversaries to forge *RREQ* and *RREP* respectively and mislead intermediate nodes to establish 'forward' and 'backward' routes, that is, routes towards the adversary that relayed the *RREQ* and *RREP* respectively. With such an attack, adversaries could simply overflow the routing tables of correct nodes, and prevent additional routes to be established across such nodes. To fully thwart such attacks, each intermediate node relays at most *n RREP*'s per neighbor per query, with *n* a protocol-selectable parameter, given that all protocol checks for such *RREP*'s are successful. At the same time, intermediate nodes service *RREP* packets from different neighbors in a round-robin manner to ensure that even if one adversary transmits replies at a fast rate, legitimate ones will also be relayed.

Finally, consider the following case: an adversary *M* fully complies with the *RREQ* propagation rules, and later receives a *RREP* from *W* to *V*, with *V*, in turn, bound to relay *RREP* to its predecessor *U*. Then, if *M* is a neighbor to both *W* and *U*, it might relay the *RREP* before *V*. This would result in a structurally intact route, which, nevertheless, includes an adversary. To prevent such an attack, nodes relaying a *RREP* can encrypt the *route*$_{ID}$ using neighbor-to-neighbor symmetric keys [8]. In general, to counter adversaries complying with the secure route discovery to place themselves on a route, protocols such as the *Secure Message Transmission* (*SMT*) and the or the *Secure Single Path* (*SSP*) protocols are necessary [6].

Compared to *S-AODV*, *DV-SRP* prevents an independent adversary *M* from increasing the hop count, and also prevents such an adversary from 'rushing' an overheard legitimate *RREP*, that is, relay it to some node *U* before *U*'s successor *V*. Compared to *ARAN*, which discovers only one route based on the first-received route reply, *DV-SRP* is more general, as it provides the correct length of the route in the presence of independent adversaries, and it discovers multiple routes. Compared to both protocols, *DV-SRP* is more efficient, as it makes limited use of public key cryptography, but rather relies primarily on symmetric key primitives that incur roughly three to four orders of magnitude lower processing overhead. We note that *DV-SRP*, as well as *S-AODV* and *ARAN*, require authentication of the end nodes at all intermediate nodes handling *RREQ* and *RREP* packets, unlike *SRP* [3]. Moreover, we note that neither *DV-SRP* nor *S-AODV* or *ARAN* can ensure that routes are link- or node- disjoint, a useful feature for multi-path data forwarding [9].

References

[1] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," *IETF RFC 3561*, Jul. 2002

[2] IETF MANET Charter, http://www.ietf.org/html.charters/manet-charter.html

[3] P. Papadimitratos and Z.J. Haas, "Secure Routing for Mobile Ad Hoc Networks," in proceedings of the *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference* (*CNDS 2002*), San Antonio, TX, Jan. 27-31, 2002

[4] K. Sanzgiri, B. Dahill, B.N. Levine, E. Royer, C. Shields. "A Secure Routing Protocol for Ad Hoc Networks," in proceedings of *ICNP 2002*, Nov. 2002

[5] M. G. Zapata and N. Asokan, "Securing Ad hoc Routing Protocols," in proceedings of the *ACM WiSe 2002*, Atlanta GA, Sept. 2002

[6] P. Papadimitratos and Z.J. Haas, "Secure Message Transmission in Mobile Ad Hoc Networks," in proceedings of the *ACM WiSe 2003*, San Diego CA, Sept. 2003

[7] IEEE Std. 802.11, "Wireless LAN Media Access Control (MAC) and Physical Layer (PHY) Specifications," 1999

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

[9] P. Papadimitratos, Z.J. Haas, and E.G. Sirer, "Path Set Selection in Mobile Ad Hoc Networks," in proceedings of the *Third ACM Symposium on Mobile Ad Hoc Networking & Computing* (*MobiHoc 2002*), Lausanne, Switzerland, Jun. 2002

[10] V. Hadzilacos and J. Y. Halpern, "Message-optimal protocols for Byzantine agreement," *Mathematical Systems Theory*, 26, pp. 41-102, 1993

[11] F. Cristian, H. Aghili, R. Strong, and D. Dolev, "Atomic broadcast: From simple message diffusion to Byzantine agreement," in proceedings of the *Fifteenth International Symposium on Fault-Tolerant Computing*, p. 200-206, June 1985. Also, IBM Research Laboratory Technical Report RJ5244, Apr. 1989

[12] M. Pease, R. Shostak, and L. Lamport, "Reaching agreement in the presence of faults," *Journal of the ACM*, 27(2):228-234, Apr. 1980

[13] P. Papadimitratos, "Secure and Fault-tolerant Communication in Mobile Ad Hoc Networks," PhD Dissertation, Cornell University, 2004.