Private Logical Neighborhoods for Ad Hoc Networks

Kulasekaran Sivakumar, Mahalingam Ramkumar
Department of Computer Science and Engineering
Mississippi State University, MS.

***Abstract*— In wired networks routers are aware of all other routers to which they are directly connected. In contrast, in mobile ad hoc networks (MANETs) a mobile router (node) will not be aware of all nodes within its transmission range. While many MANET routing protocols have been proposed in the literature, only few of them mandate proactive neighborhood discovery protocols like the Internet message encapsulation protocol (IMEP), for identifying all neighbors within a reliable delivery neighborhood (RDN). We argue why it is especially important for *secure* routing protocols, which have the additional constraint of the need to "live with" non cooperative nodes, to go beyond simply mandating an RDN, by mandating a *private logical neighborhood* (PLN).**

I. Introduction

Nodes participating in mobile ad hoc networks (MANET) [1] simultaneously act as network hosts and routers and relay packets amongst each other. Apart from many issues addressed by all routing protocols, MANET protocols have to address some additional constraints like 1) the resource constrained nature of mobile devices, 2) rapid changes in topology due to mobility and 3) issues specific to wireless (as opposed to wired) links. It is thus not surprising that most ad hoc routing protocols in their original incarnations [2], ignored other practical considerations like the existence of non cooperative or malicious routers.

Many secure MANET protocols [3] - [5] have been proposed since then which strive to enforce mutual cooperation, by reducing the degrees of freedom of participants to violate rules. However, while original MANET protocols which ignored security issues did explicitly address salient differences between wired and wireless links, several popular *secure* protocols in the literature have unfortunately ignored such differences. Neglecting these important differences leads to many potential security holes in the protocols.

A. Contributions

From a security standpoint, a primary difference between wired and wireless networks is the fact that in wired networks a router is well aware of all routers to which it is *physically* connected. In contrast, in wireless networks, it is not possible for a node A to determine the list of all entities that can hear a packet sent by A. Arguably, the first step towards securing MANET protocols is the inclusion of some proactive security features to eliminate the differences between wired and wireless networks. Such an approach is especially important in many emerging hybrid networks which may consist of nodes interconnected by wired and wireless links.

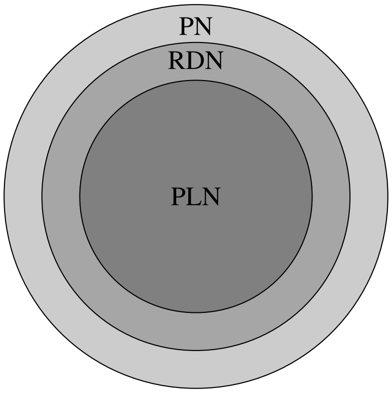

Fig. 1. Relationship between physical neighborhood (PN), reliable delivery neighborhood (RDN), and private logical neighborhood (PLN).

In this paper we argue the need for maintaining a private *logical* neighborhood (PLN) for MANETs (see Figure 1). The reliable deliver neighborhood (RDN) of a node A is a subset of physical neighbors of A with which the existence of bi-directional links has been confirmed. The logical neighborhood of a node A may in general consist of only a subset of nodes in its RDN. Irrespective of the nodes that may actually be within the hearing distance of A's transmissions, the node A can explicitly specify a subset of such nodes that will be *inducted* into the PLN of A. All other nodes in the physical neighborhood will not have access to the packets sent by A.

In Section II we provide a broad overview of MANET protocols, their secure extensions, and cryptographic authentication of routing data. In Sections II-B to II-D we describe three shortcomings common to many secure MANET protocols - all of which arise from issues specific to wireless links.

In Section III we argue that imposing private logical neighborhoods can overcome the three shortcomings. We argue why (contrary to popular belief) key distribution schemes for imposing a PLN are *not* expensive. We further enumerate several compelling advantages that can be accrued by mandating PLNs. Some such advantages include i) more efficient operation in highly mobile and dense neighborhoods; ii) ability to promote self-less behavior; and iii) improved defensive measures against malicious nodes without the risk of susceptibility to denial of service attacks.

II. Limitations of Current Secure Routing Protocols

Any routing protocol will require participants to *assimilate* topology information from neighbors and *advertise* topology information that is consistent with the information assimilated. MANET routing protocols [2] can be classified into proactive and reactive protocols. Proactive approaches like the destination sequenced distance vector (DSDV) protocol strive to maintain a consistent view of the entire network at all times. In reactive protocols like the ad hoc on demand distance vector (AODV), dynamic source routing (DSR), routes are discovered when necessary. In on-demand protocols the source invokes a route request packet (RREQ) which is flooded throughout the network. The destination (or in some cases a node with the knowledge of a path to the destination) raises a route response (RREP) packet which is relayed back to the source.

A. Secure Routing Protocols

Attacks on routing protocols can be classified based on the type of attacks (active, passive and semi-active attacks); the perpetrator (external or internal attacker); and the severity of attacks (for example, a ratio of "damage resulting from the attack" to the "attacker cost" or the "risk faced by the attacker") [6], [7]. Most secure protocols employ cryptographic authentication techniques for verifying the integrity and the source of routing information.

1) Cryptographic Authentication: Cryptographic authentication is facilitated by key distribution schemes (KDS). For example, a key distribution scheme which facilitates pairwise secrets between any two nodes will permit two nodes A and B to compute a secret K_{AB}, using the secrets provided to them by a key distribution center (KDC). The pairwise secret can be used for protecting the privacy of exchanges between A and B and for mutual authentication of a message M exchanged between A and B, where the sender appends an "authentication token" in the form of a hashed message authentication code (HMAC) $h(M, K_{AB})$, which can be verified only by the receiver (which can also compute K_{AB}).

While it is possible for a node A to authenticate a message to multiple verifiers by appending independent HMACs (based on individually shared secrets), this may not be practical in scenarios where the number of verifiers are large. It is not even possible in scenarios where the message source does not know the identities of potential verifiers *a priori*. Key distribution schemes which facilitate source authentication permits a source A to compute an authentication token $T_{A,M}$ for a message M, which can be *verified* by any receiver. Thus source authentication schemes cater for *unlimited* number of verifiers, whose identities may not be known to the message source a priori. An example is a digital signature based on asymmetric cryptographic primitives. The digital signature of A for a message M (or the authentication token $T_{A,M}$) is computed using a private key known only to A. The signature can be verified by any entity which has access to a legitimate copy of A's public key. The legitimate copy of A's public key is usually conveyed through a certificate signed by a trusted certificate authority (whose public key is made available to all nodes).

In the rest of this section we shall address some limitations common to several secure routing protocols.

B. One-Way Links

Some routing protocols like temporally ordered routing algorithm (TORA) [8] rely on an underlying mechanism like Internet message encapsulation protocol (IMEP) [9] to identify a reliable delivery neighborhood (RDN). On the other hand, some protocols like DSR do not rely on IMEP. The original version of DSR did *not* require the assumption of bidirectional links. However, most secure extensions of DSR [3], [10], [11], *rely* on the assumption that all links are bidirectional. Unfortunately, such secure extensions simply assume that all links are bidirectional even while they do not possess any explicit mechanism to *ensure* this requirement.

One common (but incorrect) rationale [11] provided for this justification is that the underlying medium access control (MAC) protocols take care of this requirement by employing a handshake, where the sender sends a short request-to-send (RTS) packet and the receiver responds with a clear-to-send (CTS) packet. However, such exchanges can be used only for *unicast* exchanges that follow the RTS / CTS handshakes. The RTS and CTS packets that explicitly identify the sender and a unique receiver. For route request (RREQ) packets that are meant for all nodes within range, such handshakes cannot be used. Thus, an RREQ packet sent by a node B can reach a neighbor C even if the reverse link $C \rightarrow B$ does not exist. If C forwards such RREQs such RREQ packets (which are bound to fail to establish a path) can *preempt* other good RREQ packets (as each node forwards only one RREQ), and thus prevent discovery of alternate good paths.

A quantitative illustration of the effect of one-way links

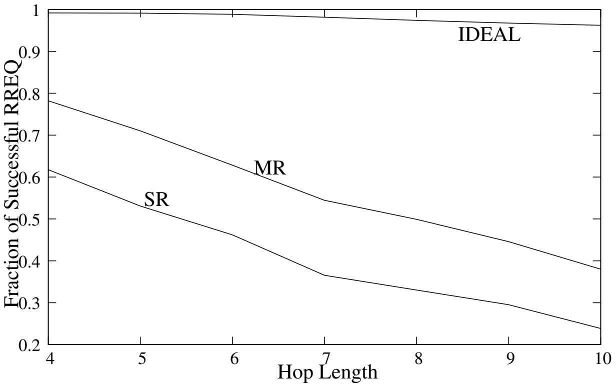

Fig. 2. Plots depicting the effect of one-way links on the efficacy of route discovery process for protocols where multiple RREQs (MR) reach the destination and for protocols where only the first RREQ (SR) is honored. The plot labelled IDEAL depicts a scenario where some proactive measure is enforced for avoiding the use of one-way links.

on on-demand protocols like DSR and AODV is depicted in Figure 2. Simulations where carried out for random realizations of 200 nodes in a square region with unit edges. To simulate one-way links the range of each node was chosen to be uniformly distributed between 0.09 and 0.11 units. RREQ propagation was simulated between every pair of nodes, separated by different hops lengths (X-axis). The plots depict the fraction of successful node pairs (Y-axis) that discover a path free of one-way links.

For protocols like DSR where the source and destination can discover *multiple* paths, the end points are assumed to succeed in their quest (to establish a path) if at least one path is free of one-way links. For protocols like AODV where the destination responds only to the first RREQ received the first RREQ should be clear of one-way links. The plots labeled MR and SR depict the fraction of successful route request attempts for the scenarios where i) at least one RREQ should succeed (MR) and ii) the first RREQ should succeed (SR). The plot labelled IDEAL depicts the scenario where some proactive mechanism is employed to inhibit the propagation of RREQs over one-way links.

C. *Link Level Authentication*

While the danger of unauthorized tapping does exist even in wired networks, this can be easily addressed by establishing a shared secret between the end points. Establishing a shared secret between A and B at two ends of a wired connection is comparatively trivial as key distribution schemes used for such purposes do not need to scale well. Such keys can even be set up manually. Furthermore, even if the overheads for establishing such secrets are high, it is acceptable as 1) the devices are typically not resource constrained; and 2) the established secrets can be used for very long durations as the end-points very rarely change.

In contrast, establishing a shared secret between two neighbors in an ad hoc subnet is more challenging. Firstly, as the neighbors of a node may change rapidly, the key establishment process has to be performed more frequently. Secondly, a node has no *a priori* knowledge of *who* could end up as a neighbor. Thus, node A should be prepared to accept potentially *every* node in the network as its neighbor. This calls for a key distribution schemes that support very large (or even unlimited) network sizes, that facilitate non mediated establishment of shared secrets. The resource constraints inherent to MANET nodes make this requirement more challenging to meet.

Many popular secure routing protocols like Ariadne [3], SRP [10], SAODV [4] either lack mechanisms, or employ ineffective mechanisms for authentication of neighboring nodes. In the DSR-based secure routing protocol (SRP) [10] only the source and destination share a secret. No mechanism exists for verification of the authenticity of intermediate nodes. Ariadne [3] employs TESLA [12] for authentication of intermediate nodes in the path by the RREQ source. A *network-wide* shared secret, used for encrypting / authenticating all packets sent by every node to keep *external* nodes away, is the only form of link level authentication employed.

In Ariadne a malicious internal node C forwarding an RREQ can insert itself as some node C' in the RREQ. The authentication appended by intermediate nodes can be verified only by the destination (for the scheme in [11]) or the source (for Ariadne with TESLA). However such bad RREQs can still preempt other good RREQs from reaching the destination. In SAODV (where no authentication is required to be appended by intermediate nodes) any node (internal or external) can engage in such attacks. What makes the attack more appealing for attackers is that they face absolutely *no risk* in carrying out such attacks. Without deterrents, the attackers can engage in such attacks unabated.

Figure 3 depicts plots illustrating the fraction of node pairs that successfully discover a path free of malicious nodes. Once again the simulations involved random network realizations with 200 nodes, out of which 20 nodes were randomly labelled malicious. For DSR it is assumed that the end points will succeed if at least one path free of the malicious nodes is established (plot labelled MR). For AODV the first RREQ received by the destination needs to be free of malicious nodes (plot labeled SR). To see the extent of suppression of good RREQs, the plot labelled IDEAL depicts the scenario where the 20 nodes do *not* take part in forwarding the RREQ.

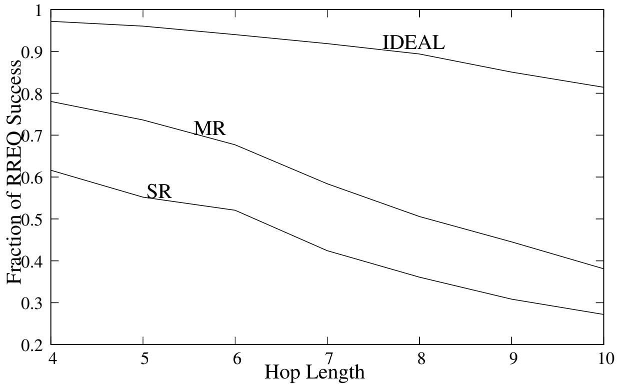

Fig. 3. Plots depicting the effect of risk-free rushing attacks on the route discovery process for protocols where multiple RREQs (MR) reach the destination and for protocols where only the first RREQ (SR) is honored. The plot assumes 20 (randomly chosen) attackers. The plot labelled IDEAL depicts a scenario where the 20 malicious nodes refrain from taking part in RREQ propagation.

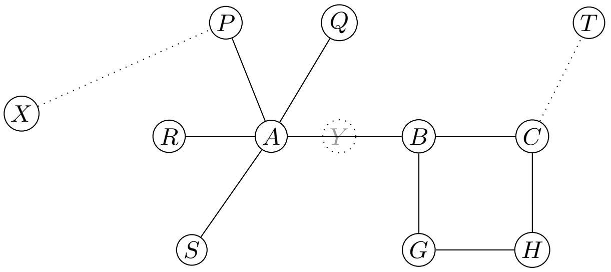

Fig. 4. Network topology used for illustrations.

D. Per-hop Hashing

While the cryptographic authentication appended by a node A is sufficient to convince a verifier that the information does indeed originate from A (assuming no other node has access to secrets of A), it does not provide any assurance that the information provided by A is indeed correct. Some redundant information is required to facilitate nodes to gather the *same* information from multiple sources.

1) Carrying Over Authentication: One of the most common strategies for providing this redundancy is by *carrying over* authentication. To illustrate some issues in carrying over authentication, we shall consider the network topology in Figure 4. In the description below we use the following notations:

1) $[X, 4]_A$ denotes an authenticated message[1] from A indicating that X is four hops away from A.
2) $[A \leftrightarrow B]_A$ denotes an authenticated message from A indicating that B is a neighbor of A; $[A \leftrightarrow B]_B$ represents an authenticated message from B indicating that A is its neighbor.

Assume that at some t, A broadcasts a message $[X, 4]_A$ indicating that X is four hops away from A. At some time $t_1 > t$ a neighbor B of A broadcasts a message $[X, 5]_B$, which is received by a neighbor C of B. If the mesage from B is also accompanied by the message from A, viz., $[X, 4]_A$, it would appear at first sight that C has two "independent confirmations" that X is six hops away from A. However, this is not true as it is possible that a node Y exists between A and B; the node Y could have relayed the information sent by A to B along with its distance to X, $[X, 5]_Y$; a malicious B relays the message $[X, 4]_A$ from A instead of the message $[X, 5]_Y$ from Y, and announces a shortended hop-count (or path length) $[X, 5]_B$.

In relaying the message $[X, 4]_A$, note that an implicit claim of B is that "A is my neighbor." The problem is that this claim is *not* verifiable by B's neighbors. One approach is for A to authenticate this information with a message $[A \leftrightarrow B]_A$, which should also be broadcast along with the message $[X, 4]_A$. More specifically, in a scenario where A has multiple neighbors (P, Q, R, N, and B) at time t, A should create an authenticated list of all its neighbors.

The authentication appended by A is intended for verification by all two-hop neighbors of A. Obviously, as A does not necessarily know the identities of all potential verifiers (all two-hop neighbors) a source authentication scheme (like a digital signature) will need to be used for authenticating the message. Furthermore, as the neighbor-list of A may change rapidly, the appended authentication token should be deemed valid only for a small period, and therefore, will need to be refreshed frequently. Obviously such an approach can introduce substantial overhead, especially in highly dynamic subnets.

2) Per-hop Hashing: An elegant way to address this problem (providing verifiable proof that B is indeed a neighbor of A) with relatively low overhead, is the per-hop hashing strategy. Instead of explicitly naming all its neighbors and periodically providing authenticated neighbor lists, node A provides a "per-hop hash value" *only* to its neighbors. That a node (say B) has access to the value provided by the previous hop (A) is "somehow demonstrated" to downstream nodes. The specifics of *how* this demonstration is performed, and to *whom*, are however protocol dependent.

For example, in Ariadne [3], this proof is demonstrated only to the destination. The per-hop hash is seeded by a value β_0 which is privy to both the source S and and destination T (β_0 is derived as a one-way function of a secret shared K_{ST} between the source and the destination). The source S broadcasts this value to all its neighbors. A neighbor A of S replaces the value β_0 with the value $\beta_1 = h(\beta_0, A)$ and broadcasts β_1 to all

[1] An authenticated message includes the message and a verifiable authentication token like a HMAC or a digital signature.

its neighbors. A neighbor B of A similarly replaces β_1 with $\beta_2 = h(\beta_1, B)$, and so on, at every hop. Thus when the destination T receives a value β_n with n nodes in the path, it can compute β_0, recursively compute β_n, and verify that no nodes have been removed from the path. In a scenario where B is *not* a neighbor of A, note that B cannot claim that A is its neighbor (by removing Y from the path indicated), as B does not have access to the per-hop hash broadcast by A. Variants of the per-hop hashing strategy are also used in secure extensions of AODV [4] and DSDV [5] to prevent attacks involving shortening of paths. Unlike the per-hop hashing scheme in Ariadne, for the schemes in [4] and [5] computing the per-hop hash does not include the identity of nodes.

While per-hop hashing is an efficient strategy, its security rests on the assumption that *only* neighbors of A have access to the per-hop hash value broadcast by A. Sending the per-hop hash value in the clear for the benefit of all nodes within range (with the assumption that nodes that are not neighbors cannot hear the value anyway) is obviously a loop-hole that can be exploited by attackers. As a concrete example, consider a scenario where a malicious node C simply *pretends* to be out of B's range. Assume that B relays a request originating from S indicating a path (A, B) and a per-hop hash value β_2. Now C waits for the RREQ to be relayed along another path (A, B, G, H). However, with access to β_2 transmitted by B, C has the ability to remove its immediate upstream neighbor H, or both G and H, from the path. In this particular instance C would obviously to remove H from the path, as removing both G and H is tantamount to *admitting* that C can hear B.

III. Private Logical Neighborhood

The three major limitations described in Sections II-B to II-D can be effectively addressed by employing a private logical neighborhood (PLN), where a node A explicitly invites all or a subset of its neighbors in its physical neighborhood into its PLN by providing them with a one-hop secret. For example, node A provides a secret K_A to all its PLN nodes. All subsequent transmissions by A will be encrypted using K_A. Thus

1) neighbor authentication is implicitly catered for;
2) a node A can simply cut off a neighbor B if it suspects that the link $A \rightarrow B$ is not reliable, by providing a new secret to all other nodes in its PLN; and
3) nodes that are not *explicitly* invited into the PLN of A will not gain access to the per-hop hash value sent by A.

Eliminating one-way links and attacks that exploit the lack of link level authentication can result in substantial improvement in the success of successful RREQs, as indicated by the plots labelled IDEAL in Figures 1 and 2. In Figure 1 the plot labelled IDEAL represents the scenario where one-way links are pro-actively inhibited (RREQs are not allowed to pass through such links). In Figure 2 the plot labelled IDEAL represents the scenario where attackers do not participate in the process of relaying RREQs.

Note that without link-level authentication attackers can afford to carry out rushing attacks by forwarding ill-constructed RREQ packets without facing any risk of being identified. However, if a PLN is imposed, attackers face the risk of being identified by neighbors. Furthermore, establishing a PLN is also mandatory in scenarios where per-hop hashing strategy is used. More specifically, the per-hop hashing strategy implicitly demands a mechanism to ensure that the privilege (per-hop hash) is privy only to intended neighbors.

A. Key Distribution

Enforcing PLNs calls for a schemes for ad hoc establishment of pairwise secrets between nodes, where any two nodes A and B should be able to independently compute a pairwise secret K_{AB}. It is widely held [3], [11] that "scalable schemes for ad hoc establishment of pairwise secrets (using only symmetric cryptographic primitives) are impractical." However, while bandwidth and computational overhead are expensive for mobile devices (due to the need to preserve battery life), *storage* is a relatively inexpensive resource. Flash storage supporting several GBs are already very common. The low cost of storage can offset the inherent limitations of key predistribution schemes (KPS) for ad hoc establishment of pairwise secrets.

For example, for the "basic" key predistribution scheme, the key distribution center can choose a master secret M. The pairwise secret between two nodes A and B can be computed as $K_{AB} = h(M, A, B) \oplus h(M, B, A)$. For a network of N nodes the KDC provides $\binom{N}{2}$ secrets to every node. More specifically, toa ssign secrets to a node A the KDC computes $K_{AB} = h(M, A, i) \oplus h(M, i, A)$, where i is the identities of the $N - 1$ other nodes in the network.

The reason that the "basic" KPS is nonscalable are two-fold. The first is the $\mathbb{O}(N)$ storage requirement for each node. However, this may not be a practical limitation even for networks with tens of millions of nodes. After all, a million 80-bit secrets require a mere 10 MB of storage. The second, and more important reason that makes the "basic" KPS impractical is that it does not facilitate asynchoronous induction of nodes into the network. For example, id a node A is inducted before node B, while B can be provided with the secret K_{AB}, it is impractical to provide K_{AB} to A. Several novel

KPSs have been proposed recently. All such schemes have been motivated by the need to reduce computational and bandwidth overhead, by leveraging the substantial storage capabilities.

1) MLS: In a recently proposed "nonscalable" KPS, the modified Leighton-Micali scheme (MLS) [14], every node receives one secret from the KDC. In addition, every node receives multiple public values. More specifically, the i^{th}node to be inducted into the network receives one secret and $i-1$ public values. The first node inducted into the network does not need to store any public value. The millionth node inducted into the network will need to store $999,999$ public values (a mere 10 MB of storage if each public value is 80 bits long). The ten millionth node will need about 100 MB for storing its public values. Any two nodes, irrespective of when they were inducted into the network, can compute a pairwise secret by performing one hash function evaluation.

MLS can realistically support a maximum network size of several *tens of millions* [14]. While like the basic KPS MLS has a limit on the maximum number of nodes N, MLS has many desirable properties that are usually associated only with truly scalable schemes: like i) the ability to support asynchronous induction of nodes; and ii) identity based allocation of secrets.

2) Scalable KPSs: For network sizes that cannot be supported by the "nonscalable" MLS, scalable KPSs are viable options. However, unlike MLS, scalable KPSs are susceptible to collusions. An n*-secure* KPS can "resist" an attacker who has pooled together all secrets of n entities. More generally, for an (n,p)-secure KPS, an attacker with access to secrets of n nodes can compute a fraction[2] p of all possible pairwise secrets.

Improving the collusion resistance of scalable KPSs demands increased complexity. The complexity associated with any scalable KPS however has two different facets: i) storage complexity for secret/public values; and ii) computational complexity (for computing the pairwise secret).

For *deterministic* n-secure KPSs [15] *both* facets of complexity are $\mathbb{O}(n)$, which makes them ill-suited for realizing large collusion resistance n. For some recently proposed *probabilistic* (n,p)-secure KPSs [16],[17] the storage complexity is $\mathbb{O}(n\log(1/p))$. The computational overhead is merely $\mathbb{O}(\log(1/p))$, and more importantly, *independent* of the desired collusion resistance n.

As the achievable security is limited *only* by available storage, realizing very high levels of collusion resistance (say n of the order of hundreds of thousands) is very much practical, thus rendering the issue of collusion resistance irrelevant. The cost is a few tens of megabytes of storage for each node. To compute any pairwise secret a node has to fetch a few tens ($\mathbb{O}(n\log(1/p))$) of secrets from bulk-storage (for example a flash card) and perform a mere tens ($\mathbb{O}(n\log(1/p))$) of hash operations to compute the pairwise secret.

Due to their very low computational overheads, the few tens of hash computations with secrets can also be easily performed by a modest SIM card in the mobile device, to further alleviate the issue of exposure of secrets from a large number of nodes. An attacker desiring the exploit the "collusion susceptibility" of such KPSs will have to successfully hack and expose secrets from over one-hundred-thousand SIM cards.

Thus far schemes which employ one hop secrets[3] [18] assume that that secrets are established by exchanging public keys and performing asymmetric computations. Obviously the overheads for such approaches may render establishment of one-hop secrets impractical. Fortunately, the fact *that storage is an inexpensive resource for mobile computers renders scalable light weight schemes for ad hoc establishment of pairwise secrets practical.*

B. Other Advantages of PLNs

Apart from addressing the three issues that plague many secure MANET protocols, imposing a PLN results in several other benefits.

1) Mandating PLNs is useful in scenarios involving highly dynamic nodes. Consider a scenario where a mobile device in a fast moving vehicle sends a RREQ packet. If every neighbor simply floods the RREQ onwards it may result in substantial wastage of bandwidth as the RREQ source is very likely to have moved away from the location from which the RREQ originated by the time the response comes back. If nodes enforce a PLN (and induct nodes in their PLN only after a few exchanges) only nodes moving at roughly the same speed (or relatively stationary to each other) will form an exclusive network. Thus a set of northbound vehicles in Interstate I-95 may form an ad hoc subnet that excludes all southbound vehicles (and vice-versa).

2) There are many valid reasons as to why a node A may desire to cut-off a *specific* node C (which is physically in the neighborhood of A) from its logical neighborhood. For instance 1) A may have observed consistent misbehavior or non participation by node C, or 2) suspect a one-way link between A and C or 3) suspect the presence of a semi-active attacker [6] between A and C. Such a suspicion could be triggered if A hears an echo of its own packet and / or if the time delay between RTS / CTS handshakes between A and C seem above normal.

[2] However, as long as p is low enough (say 2^{-64}) it is computationally infeasible for an attacker to even identity *which* pairwise secrets can be compromised by using the pool of secrets accumulated from n nodes.

[3] The scheme in [18] employs one-hop and two-hop secrets.

Most existing approaches for mitigating participation by malicious nodes involve propagating accusatory messages regarding misbehavior of nodes. Unfortunately, in most scenarios it may be infeasible for the observer to provide incontrovertible proof of misbehavior of some node, verifiable by all nodes in the network. Thus such strategies are themselves susceptible to simple denial of service (DoS) attacks where a node can send false accusations to create unnecessary traffic. The ability to cut-off neighbors in the physical neighborhood from the PLN facilitates DoS-free countermeasures to reduce the ill-effects of malicious nodes. Nodes cut off by all neighbors are effectively cut off from the network.

3) Mandating a PLN can also deter selfish behavior by nodes which would wish to remain silent and not participate in the routing process until there a packet addressed to it. With a logical neighborhood a node will have to be *inducted* into the PLNs of its neighbors before they can monitor traffic. Once inducted, a node C is pressured to participate to the fullest extent due to the fact that it is under constant observation by its neighbors, who may cut C off if they sense selfish participation of C.

4) Furthermore, in a scenario where two nodes A and C are situated very close to each other and have identical views of the network, the nodes gain nothing by adding each other to their respective PLNs (unless A and C are end points in an interaction). In a region where a node A has 100 nodes within range, A may decide to include only 10 of them in its PLN as 10 neighbors may be sufficient to provide A with connectivity to all other nodes.

5) In dense deployments of wireless devices that are expected in the future, and in scenarios where physical layer jamming is an issue, dynamic spread spectrum strategies will need to be used. The shared keys needed CDMA or frequency hopping can also be derived from the one-hop group secret.

IV. Conclusions

We argued the need for private logical neighborhoods for MANETs and enumerated several compelling advantages offered by mandating PLNs. We described three limitations common to several popular secure MANET protocols and argued that such limitations can be removed in one stroke by mandating PLNs. We then enumerated several other advantages of using PLNs.

Maintaining PLNs demands a lightweight key distribution schemes for ad hoc establishment of pairwise secrets. It is perhaps due the wide-spread belief that "scalable lightweight key distribution schemes for ad hoc establishment of pairwise secrets are impractical[4]" that has in turn led to the assumption that enforcing PLNs will be impractical. Even in protocols which employ one-hop secrets it is assumed that such secrets are established using asymmetric primitives. We argued why the low cost of storage for mobile computing applications has substantially improved the appeal of lightweight key predistribution schemes for this purpose, and thus provides a practical approach for establishing PLNs.

[4] For example, this is the rationale provided for the choice of TESLA [12] instead of pairwise secrets in Ariadne [3].

References

[1] Web Link, http://www.ietf.org/html.charters/manet-charter.html.

[2] E.M. Royer,C.K. Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks," *IEEE Personal Communications,* pp. 46-54, 1999.

[3] Y-C Hu, A Perrig, D B.Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks," Journal of Wireless Networks, **11** pp 11–28, 2005.

[4] M G Zapata, N.Asokan ,"Securing Ad Hoc Routing Protocols," Proceedings of the ACM workshop on Wireless security, Atlanta, Georgia, September 2002.

[5] Y-C Hu, D B. Johnson, A Perrig, "SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks," Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications, 2002.

[6] J.Marshall, V.Thakur, A.Yasinsac,"Identifying flaws in the secure routing protocol," Proceedings of the 2003 IEEE International Performance, Computing, and Communications Conference, 2003.

[7] Y-C Hu, A. Perrig, D.B. Johnson, "Rushing Attacks in Wireless Ad Hoc Network Routing Protocols," WiSe 2003, San Diego, CA, September 2003.

[8] V. D. Park, M. S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," pp 1405–1413, INFOCOM 1997.

[9] S. Papademetriou, P. Papadopoulos, V. Park, A. Qayyum, "An Internet MANET Encapsulation Protocol (IMEP) Specification," Internet Draft, August 1999.

[10] P Papadimitratos, Z. J.Haas, "Secure Routing for Mobile Ad Hoc Networks," Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference(CNDS 2002), San Antonio, Texas,2002.

[11] J. Kim, G. Tsudik, "SRDP: Securing Route Discovery in DSR," IEEE Mobiquitous'05, July 2005.

[12] A. Perrig, R. Canetti, D. Song, D. Tygar, "Efficient and Secure Source Authentication for Multicast," in Network and Distributed System Security Symposium, NDSS '01, Feb. 2001.

[13] P. Gupta, P. R. Kumar, "The capacity of wireless networks," IEEE Trans. Inform. Theory, vol. 46, no. 2, pp. 388404, Mar. 2000.

[14] M. Ramkumar, "On the Scalability of a "Nonscalable" Key Distribution Scheme," IEEE SPAWN 2008, Newport Beach, CA, June 2008.

[15] R. Blom, "An Optimal Class of Symmetric Key Generation Systems," *Advances in Cryptology: Proc. of Eurocrypt 84*, Lecture Notes in Computer Science, **209**, Springer-Verlag, Berlin, pp. 335-338, 1984.

[16] M. Ramkumar, "Trustworthy Computing Under Resource Constraints With the DOWN Policy," IEEE Transactions on Secure and Dependable Computing, Jan 2008.

[17] M. Ramkumar, "Efficient Key Distribution Schemes for Large Scale Mobile Computing Applications," Cryptology ePrint Archive, 2008/332, http://eprint.iacr.org/2008/332.pdf.

[18] X.Du, Y.Wang, J.Ge, Y.Wang,"A Method for Security Enhancements in AODV Protocol,"In Proceedings of the 17th International Conference on Advanced Information Networking and Applications, AINA 2003.