Quantifying Trust in Mobile Ad-Hoc Networks

Mohit Virendra, Murtuza Jadliwala, Madhusudhanan Chandrasekaran, Shambhu Upadhyaya
Dept. of Computer Science and Engineering, SUNY at Buffalo, Buffalo, NY 14260
{virendra, msj3, mc79, shambhu}@cse.buffalo.edu

Abstract— *This paper introduces a Trust-Domain based security architecture for mobile ad-hoc networks (MANETs). The aim of this architecture is twofold: to use trust as a basis to establish keys between nodes in a MANET, and to utilize trust as a metric for establishing secure distributed control in infrastructure-less MANETs. We define metrics for nodes to establish and manage trust, and use this mutual trust to make decisions on establishing group and pair-wise keys in the network. The impact of mobility of the nodes on trust establishment is considered and further its use as a means of propagating trust through the network is investigated. We introduce the concept of self-organizing trust-based Physical-Logical Domains (PLDs) as a means of grouping nodes for distributed control in the network.*

Keywords: Distributed Control, Key Establishment, Mobile Ad-hoc Networks, Mobility, Security, Trust

1. Introduction

Having a metric for making informed decisions is important in ad-hoc networks deployable in the military environment as well as in disaster management applications. For example, consider the scenario where a terrorist attack has taken place. The First Responder System has been rapidly deployed using ad-hoc networks, and coordination between the constituents of the responder and rescue systems has been initiated. Trust is very important here because an attack has already taken place, and now the adversary may try to destroy the relief operations by compromising the first responder system. As another example, consider the scenario of a multi-national military force deployed in a war zone. The different constituents of the force should be able to effectively communicate with each other without the risk of information compromise [1]. Trust is the most important factor in such situations to make decisions regarding whom to communicate with.

Existing key management and generation schemes for ad-hoc networks do not specify any constraints on establishing pair-wise keys between pairs of nodes, and group keys in an entire cluster or group. There are several well known schemes for key generation and management [2], [3], that are either based on secret sharing, threshold cryptography, or assume that nodes are preloaded with some keying material that helps them establish keys as required. These schemes only specify mechanisms to prevent false key generation and compromise of other nodes' keys, if a node is eventually compromised. Thus these schemes inherently assume nodes to be non-malicious at the time of key establishment. To the best of our knowledge, there are no known formal schemes for functional verification or judgment on key establishment requests from other nodes. Specifically, if a node is unknown, keys would still be established with it as long as the infrastructure for establishing keys is present. Schemes involving a trusted third party for key establishment are deemed impractical for ad-hoc networks due to the limitations on finding such an authority, and therefore, are not considered as a practical solution.

Additionally, self organization of ad-hoc network nodes into clusters has been studied in the literature [4] to induce distributed control in such networks which are otherwise infrastructure-less. Present ad-hoc network clustering schemes use physical location as a metric to cluster the nodes [5]. Any node can elect to become a cluster head and can propagate cluster joining requests to its *k*-hop neighbors through various flooding mechanisms. This choice of cluster formation is arbitrary and does not take security into account. A node that is malicious could initiate a cluster formation announcement and could potentially compromise all nodes that elect to join its cluster.

The problems that we want to address in this paper are the following: (a) To define a metric that the nodes can use to make decisions on whether to establish keys with other nodes in an ad-hoc network, given that the infrastructure for establishing such keys exists, and (b) To define a basis on which nodes in an ad-hoc network can securely group together, so that some kind of distributed control can be introduced in an otherwise infrastructure-less network. We propose to use trust between the nodes, as a mechanism to solve both these problems. We present an architecture that uses trust as a metric for nodes to (a) Make decisions on establishing keys with other nodes in the network, and (b) Group together into trust-based domains.

1.1 Related Work and Paper Organization

The idea of using trust to mitigate security threats has been an important area of research [6]. Trust establishment and management between entities (nodes or agents) can be done through a central trusted authority or in a distributed fashion by nodes [7], or a combination of both. Related work in this area [8, 9, 10, 11, 12], employ both these techniques. For example, Zhou et al. [13] propose the idea of utilizing

KIMAS 2005, April 18-21, 2005, Waltham, MA, USA.
0-7803-9013-X/05/$20.00 © 2005 IEEE.

threshold cryptography to distribute trust in ad-hoc networks, Davis [14] proposes the use of certificates based on hierarchical trust model to manage trust, and Eschenauer et al. [1] contrast between trust establishment in ad-hoc networks and the Internet. Our approach is new and different from the existing ones in that no known schemes use trust as a metric for the problems that this paper attempts to address.

This paper is organized as follows: Section 2 formalizes the notion of trust between two nodes as a combination of self trust and group trust. Section 3 quantifies trust between two nodes and describes schemes for trust management. Section 4 describes the notion of Trust Domains and organization of nodes in ad-hoc networks into domains based on trust values. We finally conclude the paper in Section 5 with a summary of its contributions, limitations and proposed future work.

2. Trust Formalization

This section describes the trust formalization. Our schemes draw ideas from the Watchdog and Pathrater schemes [15], utilized for cooperation of nodes in ad-hoc networks. We define a node *n*'s trust on another node *m*:

$$T_{n,m} = \alpha_1 \, {}_nT^m{}_S + \alpha_2 \, {}_nT^m{}_O \quad (1)$$

In the above equation, $T_{n,m}$ is evaluated as a function of two parameters:
(a) ${}_nT^m{}_S$: Node *n*'s self evaluated trust on *m*; *n* computes this by directly monitoring *m*.
(b) ${}_nT^m{}_O$: Weighted sum of other nodes' trust on *m* evaluated by *n*. In eq. (1), $\alpha 1$ and $\alpha 2$ are weighting factors such that $\alpha_1 + \alpha_2 = 1$. Thus, by varying $\alpha 1$ and $\alpha 2$, *n* can vary the weight of self evaluated vs. others trust in calculating its total trust on *m*. Here, $0 \leq \{ T_{n,m} , {}_nT^m{}_S, {}_nT^m{}_O \} \leq 1$, and thus eq. (1) is normalized.

2.1. Evaluating $_n\mathrm{T}^m{}_S$

Node *n* computes this value by directly monitoring *m* when *m* is in its radio range. We define ${}_nT^m{}_S$ as

$${}_nT^m{}_S = f(\Phi, \Omega) \quad (2)$$

Node *n*'s self trust on *m* is a function (f) of traffic statistic functions Φ and Ω computed by monitoring *m*. Precise definition of f can be implementation dependent. We assume f to be a weighted sum of Φ and Ω. Here, Φ is a function of monitored traffic statistics pertaining purely to traffic volume and Ω is a function of monitored traffic statistics pertaining to information integrity. Lee et al. [16] compile node monitoring statistics for one hop neighbors in ad-hoc networks. Thus, node *n* can monitor the following statistics for a one-hop neighbor *m*: Incoming packets on *m*, outgoing packets from *m*, outgoing packets of which *m* is the source, incoming packets of which *m* is the destination, incoming packets on *m* from *n*, etc. Based on these monitored statistics we define Φ and Ω as:

$$\Phi = g\,(\Upsilon_1, \Upsilon_2, \Upsilon_3, \Upsilon_4, \Upsilon_5, \Upsilon_6) \quad (3)$$

$$\Omega = h\,(\lambda_1, \lambda_2) \quad (4)$$

Here g and h can again be defined based on the implementation. Like$f$, we assume them to be weighted summation of their constituent parameters. These parameters are defined below:

Υ_1: packets sent by m to n that are dropped by m
Υ_2: total packets dropped by m
Υ_3: packets dropped by m due to congestion
Υ_4: packets dropped by m due to unknown reasons
Υ_5: n's assessment of m's priority to m's self packets vs. all other nodes' packets
Υ_6: packet forwarding delay by m
λ_1: packets misrouted by m
λ_2: packets falsely injected by m
Based on implementation, other parameters can also be defined.

2.2. Evaluating $_n\mathrm{T}^m{}_O$

In the representation ${}_nT^m{}_O$, *O* is the set of other nodes whose trust on *m* is utilized by *n* in evaluating its own trust on *m*. *O* is defined as:

$O = \{\forall$ node $o \in O \Rightarrow$ o is in the range of both *m* and *n*, and $\exists\, T_{no},$ s.t. $T_{no} \geq$ "good"}.

"good" is a threshold value for demarcating *Unknown* and *Good* trust-regions and this is further explained in Sec. 3.

In this section we present four schemes for computing the value of ${}_nT^m{}_O$, where *n*, *m* and *O* have their usual meanings and ${}_nT^m{}_O$ is defined as above.

1. *Optimistic or Greedy approach:* This is the simplest approach. ${}_nT^m{}_O$ is computed by selecting the largest value of the product $T_{i,m}\ x\ T_{n,i}$ for all values of *i* in the set O. In other words, node *n* uses the highest value of trust that nodes in the set *O* assign to the node *m*, weighting it with its own trust on the nodes in the set *O*.

$${}_nT^m{}_O = \mathbf{max}_{\,i \in O}\{ (T_{i,m}\, x\ T_{n,i}) \} \quad (5)$$

2. *Simple Average of Weighted Products:* The value of ${}_nT^m{}_O$ is the simple average of the product of $T_{i,m}$ and $T_{n,i}$ over all the nodes *i* in the set *O*.

$${}_nT^m{}_O = \frac{\sum_{i \in O} (T_{i,m}\ x\ T_{n,i})}{|O|} \quad (6)$$

where | *O* | is the cardinality of the set *O*.

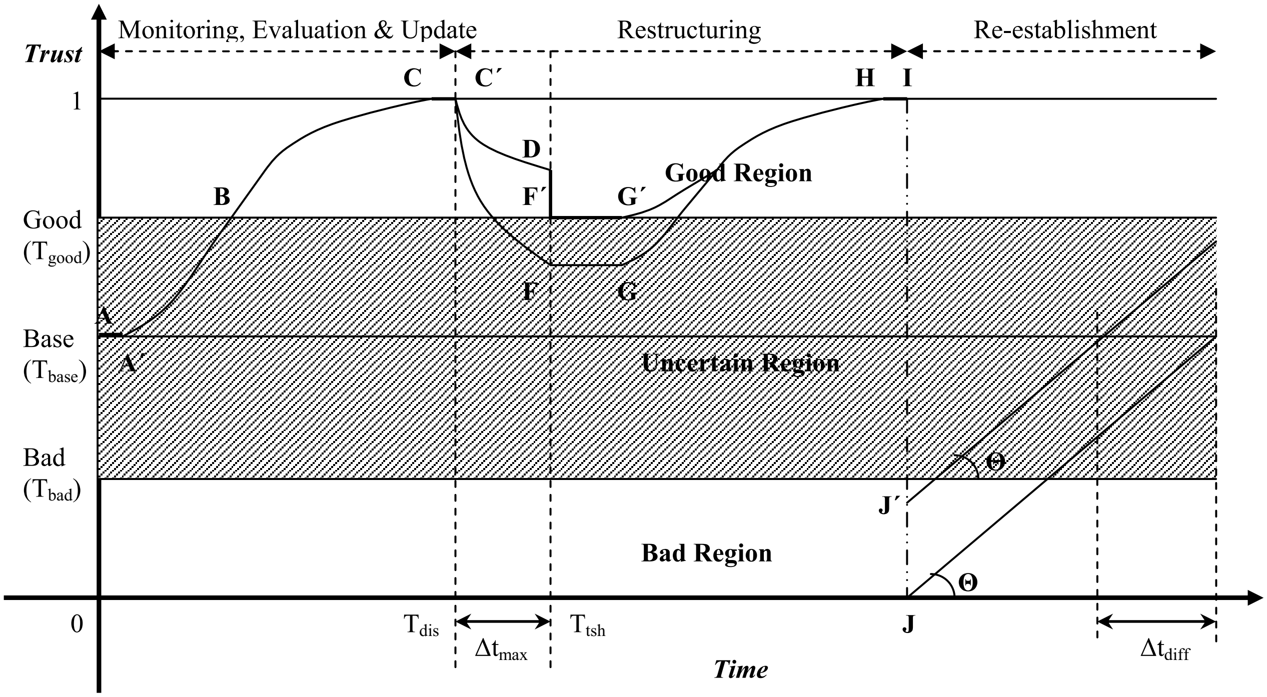

Figure 1: Trust Evaluation

3. *Weighted Average:* The value of ${}_nT^m_O$ is the weighted average of $T_{i,m}$ over all the nodes in the set O. The weight associated with each $T_{i,m}$ is $T_{n,i}$.

$$ {}_nT^m_O = \frac{\sum_{i \in O} (T_{i,m} \times T_{n,i})}{\sum_{i \in O} T_{n,i}} \qquad (7) $$

Alternatively, if we normalize with respect to $T_{i,m}$ we get,

$$ {}_nT^m_O = \frac{\sum_{i \in O} (T_{n,i} \times T_{i,m})}{\sum_{i \in O} T_{i,m}} \qquad (8) $$

Thus, in the weighted average approach the weighted products of $T_{i,m}$ and $T_{n,i}$ are normalized either with $T_{i,m}$ or $T_{n,i}$ for all values of i in the set O.

4. *Double Weighted Approach:* In this approach, we further try to improve on the value of ${}_nT^m_O$ computed from the weighted average approach by normalizing the product of $T_{i,m}$ and $T_{n,i}$ with respect to both $T_{i,m}$ and $T_{n,i}$.

$$ {}_nT^m_O = \frac{\sum_{i \in O} (T_{i,m} / \sum_{j \in O} T_{j,m}) \times T_{n,i}}{\sum_{i \in O} T_{n,i}} \qquad (9) $$

Alternatively,

$$ {}_nT^m_O = \frac{\sum_{i \in O,\, j \in O} (T_{n,i} / \sum T_{n,j}) \times T_{i,m}}{\sum_{i \in O} T_{i,m}} \qquad (10) $$

Note that the first scheme is the simplest, but it is based on the accuracy of trust on *one* node. This scheme would be the most vulnerable to misdecisions and failure due to malicious misrepresentation of trust by a single node, or a collusion of nodes. Schemes 2, 3, and 4 increase in complexity of evaluation, but should also provide a corresponding enhanced accuracy in the evaluation of trust. We are currently performing simulations to verify the validity of this assumption.

3. Trust Evaluation

This section describes trust evaluation (Fig. 1), i.e., trust establishment and management between nodes. We explain the initial trust establishment procedure between a pair of nodes that are one-hop neighbors and explain the consequences of node mobility on the existing trust between a pair of nodes.

We define trust to be non-transitive. Thus $T_{n,m} \neq T_{m,n}$. Both m and n independently evaluate $T_{m,n}$ and $T_{n,m}$ respectively, through the schemes described above. The associability of trust will be addressed in Sec. 4. Trust evaluation between the nodes is defined as a four phase process: Initiation and

Monitoring, Query and Evaluation, Updating, and the Restructuring phase. There is an optional fifth phase: Re-establishment after declared malicious. The five trust phases of the trust evaluation process are outlined in Fig. 1 by tracing a sample trust value of a node *m* continuously evaluated by another node *n* ($T_{n,\ m}$) over a period of time. Fig. 1 depicts three trust regions: *Good*, *Uncertain*, and *Bad*. Nodes above the good trust threshold, i.e., in the *Good* region are highly trusted one-hop neighbors of *n*, and if they are one-hop neighbors of *m* also, then their trust will be utilized by *n* in evaluating ${}_nT^m{}_O$. The nodes in *Uncertain* region are those with intermediate trust values, and their trust is not utilized in evaluating ${}_nT^m{}_O$. Nodes in the *Bad* region are those marked as malicious by *n*. This is further explained in the subsections below.

3.1. Initiation and Monitoring Phase

This is the phase when the network is newly deployed, or a new node joins the network. In a newly deployed network, nodes have no traffic statistics history about the network and their neighbors, and this is akin to the training or learning phase for the network. The scenario of a new node joining the network is similar, as the node does not have any trust information about its neighbors and vice-versa. This is the period when the new nodes have not established keys with their one-hop neighbors (or any other nodes). During this phase, the new node(s) monitors its one-hop neighbors. The monitoring node (*n*) switches over to promiscuous mode and listens for all packets transmitted by the monitored node (*m*). It collects the statistics mentioned in Sec. 2.1. For example, node *m* joins the network at time T = 0, and becomes a one-hop neighbor of n. This phase is represented by the section AA' of the curve in Fig. 1. Thus a new or unknown node is given a "bare" trust value.

During this phase, nodes will not send any sensitive data to their neighbors, unless timely delivery is absolutely essential (e.g., in disaster management scenarios it might be critical to exchange information immediately after network deployment). Time critical data is transmitted immediately utilizing flooding or any other techniques. All other sensitive information is buffered by the node till trust has been evaluated and keys have been established. Thus, our technique is a cautious combination of optimistic and pessimistic approaches: we strictly limit the nature and volume of critical data transmitted during this phase with the optimistic assumption that when the network is in start-up phase, the probability of a malicious node assimilating enough information to compromise the network is very low. This is due to the relatively small amount of such data in the network vs. the large volume of set-up time control messages. It is also important to mention that this initiation phase lasts for a very short period of time and as soon as nodes have collected some information about their one-hop neighbors, they move to the evaluation phase.

3.2. Query and Evaluation Phase

During this phase, the nodes evaluate their self trust on their one-hop neighbors (e.g., ${}_nT^m{}_S$) through a challenge response system. The nodes query their neighbors regarding the statistics they have already assimilated (defined in Sec. 2.1). This is akin to truth verification, as the monitoring nodes already know the correct answers to their queries. The neighbors' trust evaluation is based on the accuracy of their responses. This has been explained in Sec. 2.1. Evaluation of ${}_nT^m{}_O$ is done as explained in Sec. 2.2. The Query and Evaluation phase is represented by A'B in Fig. 1.

3.3. Updating Trust

As long as a node remains in the radio range, its trust is continuously evaluated and updated. Thus, monitoring and querying is performed even after trust has been established between the nodes. However, the periodicity of querying and monitoring is decreased with time if the trust value stabilizes and is maintained at a certain level (e.g., the ceiling value of *Good* Region in Fig. 1). In Fig. 1, these are represented by BC and CC'.

3.4. Restructuring Phase

This phase takes into account two different scenarios and their effect on inter-node trust values:

(a) Trusted one-hop neighbors move out of radio range due to node mobility: A node, say *m*, which was previously in the radio range of a node *n*, now moves out of its radio range due to node mobility (at time T_{dis} in Fig. 1). The value of α_1 (the proportion of self trust in overall trust) now decays exponentially as:

$$\alpha_1 = c.e^{-\lambda.t} \qquad (12)$$

Parameter λ is the decay factor which is determined by the infrastructure and mobility constraints of the network, and c is some constant. Node *n* now fixes ${}_nT^m{}_S$, to the value at time T_{dis} (just before m moved away). But since α_1 exponentially decays, *n*'s importance on ${}_nT^m{}_S$ in calculating $T_{n,m}$ decreases with time. If the node *m* is outside *n*'s radio range for a time period Δt_{max}, and if $T_{n,m} > T_{good}$, then at $T_{tsh} = (T_{dis} + \Delta t_{max})$, α_1 is forced to 0 (i.e., $\alpha_1 = 0$), and $T_{n,m}$ is reduced to T_{good} (i.e., $T_{n,m} = T_{good}$). This is shown in Fig. 1 by the curve C'D and the line DF′. If the value of $T_{n,m}$ is below the *Good* Region then it is left unchanged. This scenario is shown in Fig. 1 by the curve C'F. This value is kept constant as the history information of node *n* (shown by F'G and FG in Fig. 1) for the scenario that *m* and *n* eventually return to each other's radio range.

(b) Trusted one-hop neighbors that had previously moved out of radio range are now back in radio range: the node *m*, after moving out of *n*'s radio range, eventually returns back in the range of *n* (i.e., again becomes a one-hop neighbor of n). Re-evaluation of $T_{n,m}$ by *n* is now required for potentially restoring $T_{n,m}$ to the highest trust value, as *m* becomes directly monitored again. This re-evaluation of $T_{n,m}$ is similar to the Initiation and Monitoring phase (Sec. 3.1). The only difference is that this re-evaluation does not begin from the bare trust value, but starts from the value of $T_{n,m}$ previously fixed by *n* (after m had moved out of its radio range). This is shown in Fig. 1 by the sequence of points G′HI and GHI. Similar computations are done by *m*.

3.5. Re-establishment Phase

This phase explains the scenario for a node *m* that was declared malicious previously by a node *n* and now it wants to re-associate with n. Consider the scenario when *n*'s trust on *m* is good, say at point H in Fig. 1. Assume that after monitoring *m* for some time node *n* discovers to have become malicious. This can be deduced by the challenge response scheme as describe in Sec. 3.1. The point of time when node *n* makes this conclusion is depicted by point I in Fig. 1. Now, depending upon the nature of malicious operations performed by *m*, node *n* drops the trust of node *m* into the bad region. The two levels J and J′ show this drop in trust in the figure. For serious malicious activities having a critical impact on the functioning of *n* itself, the trust of *m* is dropped to zero as shown by the point J. Zero represents the absolute minimum trust value possible (or highest distrust) in the network. For a malicious operation with a less severe effect, the trust can be dropped to any point in the *Bad* region as shown by the point J′. The two levels shown are just for illustration, but the new trust value of the malicious node can lie anywhere in the *Bad* region or it can be quantified into a number of levels based on the seriousness of offense. Trust re-establishment is based on the discretion of the node evaluating the trust (in this case node *n*). If node *n* does not want to immediately re-establish trust with node *m*, then the value of $T_{n,m}$ is unchanged till *n* decides to reconsider the trust establishment process. Trust increase after reevaluation, if at all initiated by the node *n*, is linear, provided that *m* does not perform any more malicious operations. The angle Θ in Fig. 1 represents the constant linear function for restoration of $T_{n,m}$. One important observation here is that a node for which the re-establishment begins at point J′ reaches the *Good* region earlier in comparison to a node for which the re-establishment process begins at point J.

4. Trust Model and Trust Domains

So far we have described the trust establishment and trust evaluation between pairs of nodes that have been one-hop neighbors at some point of time. In this section we extend our pair-wise trust model to include other nodes in the network which are not one-hop neighbors. We define a model through which non- neighbor nodes can establish and manage trust utilizing the five-phased trust evaluation procedure described in Sec. 3, thus providing a basis for establishing pair-wise keys between any pair of nodes, and also establishing group keys in the network. This model organizes nodes into trust-based clusters called Physical-Logical Trust Domains (PLTDs), thus securely grouping nodes to induce distributed control in the otherwise infrastructure-less network. Member nodes in a PLTD can establish and share a domain (group) key. We use node mobility to propagate trust throughout the network. In our scheme, nodes can belong to multiple PLTDs and there can be several overlapping PLTDs in a physical region.

PLTD formation can be initiated by any node. A node *n* can announce its intention to form a PLTD by requesting nodes in the set *P* to join its PLTD (PLTD-n). *P* is defined as:

P = {∀ node p ∈ P ⇒ p is in the range of *n*, and ∃ $T_{n,p}$, s.t. $T_{n,p} \geq$ "good"}.

Based on its individual trust on *n*, $T_{p,n}$, each node in *P* may either accept or decline to join PLTD-n, or it could invite *n* to join its own PLTD if it has already initiated its own domain formation procedure. If at any time, the trust value of a node in PLTD-n, say *m* ($T_{n,m}$), falls below "bare" (see Fig. 1), then its domain membership is revoked by *n*, and this is announced to other members of PLTD-n.

Now, if *n* wants to include a node *z* (non-neighbor) in PLTD-n, and *z* is a one hop neighbor of, say node *m* which is already a member of PLTD-n, then *n* can request *m* to invite *z* to join PLTD-n. Based on its own trust on *z* ($T_{m,z,}$), *m* might accept or decline to forward this invitation. If *m* forwards this invitation, then *z* can make its decision based on *m*'s evaluation of trust on *n* ($T_{m,n}$), and its own trust on *m* ($T_{z,m}$). Thus, the simplest evaluation of $T_{z,n}$ could be:

$$T_{z,n} = T_{m,n} * T_{z,m} \qquad (12)$$

This scheme assumes *m*'s willingness to provide *z* with $T_{m,n}$. If *z* is included in PLTD-n, then since n is the request initiator, $T_{n,z}$ is initially set to "good" (Fig. 1) by default. $T_{n,z}$ is continuously evaluated afterwards. Simplest evaluation of $T_{n,z}$ could be:

$$T_{n,z} = T_{n,m} * T_{m,z} \qquad (13)$$

Again this scheme assumes n's knowledge of $T_{m,z}$ provided by m. If $T_{n,z}$ falls below "bad" (Fig. 1) at any time, then *n* revokes the membership of *z* in PLTD-n, and announces this decision to other member nodes. Node *m* can unilaterally decide to end its domain membership in PLTD-n at anytime based on its trust on *n* ($T_{m,n}$) falling below a certain threshold.

This scheme is significant in both maintaining an admissible level of trust within a PLTD (because domain members share a group key), and in limiting the domain size. It is important to have an upper bound on the membership size of a PLTD for control, overhead and management purposes. Domain size can also be limited by having an absolute upper bound, say *k*, on the number of member nodes.

This scheme is also extensible for establishing trust with nodes in other parts of the network, by utilizing trusted one-hop neighbors which move away to other parts of the network due to node mobility. If node *m* moves away to a different part of the network, then *n* can utilize this to establish trust with nodes in the immediate vicinity of *m*'s new location, provided it is still able to communicate with *m*. Such a trust establishment procedure would be strictly controlled by the minimum thresholds on pair-wise trust values as mentioned above in this section, and in Sec. 3.3.

5. Discussion and Conclusion

This paper presented schemes to formalize the notion of pair-wise trust between two nodes in an ad-hoc network. It also presented schemes to evaluate pair-wise trust as a combination of self trust and group trust. We suggested using this pair-wise trust to use as a basis for establishing pair-wise keys in a network. We also described extending the pair-wise trust to form trust-based domains in the network. This would be helpful in establishing group keys in the network and would also serve as a means of securely grouping nodes into domains in MANETS and would induce distributed control in such networks.

We are currently evaluating the validity of the schemes proposed in this paper through simulations. We are also performing simulations to compare the various schemes described in Sec. 2. Our current research focuses on formalizing PLTDs to include collective decision making within domains. It also includes routing information between domains, especially through regions of unknown trust. We are working on integrating node trust models with link and path trust models. Our goal is to design a comprehensive trust based model for ad-hoc networks that can assure an admissible level of security through the use of trust.

Acknowledgments

This research is supported in part by National Science Foundation Grant No. DUE-0417095. The authors like to thank Ramkumar Chinchani and Suranjan Pramanik for the useful discussions during the course of this work.

References

[1] L. Eschenauer, V.Gligor and J. Baras, "On Trust Establishment in Mobile Ad-Hoc Networks", *Proceedings of 10th International Workshop of Security Protocols, Springer Lecture Notes in Computer Science (LNCS)*, Apr. 2002.

[2] S. Zhu, S. Xu, S. Setia and S. Jajodia, "Establishing Pair-wise Keys for Secure Communication in Ad Hoc Networks", *11th IEEE International Conference on Network Protocols (ICNP'03)*, Atlanta, GA, November 2003.

[3] S. Zhu, S. Setia, S. Xu, S. Jajodia, "GKMPAN: An Efficient Group Rekeying Scheme for Secure Multicast in Ad-Hoc Networks", *Proceedings of the 1st International Conference on Mobile and Ubiquitous Systems (Mobiquitous'04)*, Boston, Massachusetts, August 22-25, 2004.

[4] P. Krishna, M. Chatterjee, N. Vaidya, D. Pradhan, "A Cluster-based Approach for Routing in Ad-Hoc Networks", *Proc 2nd Symposium on Mobile and Location-Independent Computing,* Apr. 1995.

[5] M. Bechler, H.-J. Hof, D. Kraft, F. Pählke, L. Wolf, "A Cluster-Based Security Architecture for Ad Hoc Networks", *IEEE Infocom 2004.*

[6] T. Beth, M. Borcherding and B. Klein, "Valuation of trust in open networks", *Proceedings of ESORICS 1994*, November 1994.

[7] A. Rahman and S. Hailes, "A Distributed Trust Model", *New Security Paradigms Workshop 1997, ACM*, 1997.

[8] D. Balfanz, D. Smetters, P. Stewart and H. Wong, "Talking to Strangers: Authentication in Ad-hoc Wireless Networks", *NDSS*, San Diego, 2002.

[9] J. Kong, H. Luo, K. Xu, D. L. Gu, M. Gerla, and S. Lu, "Adaptive security for multi-layer ad-hoc networks", *In Special Issue of Wireless Communications and Mobile Computing. Wiley Interscience Press*, Aug. 2002.

[10] T. Hughes, J. Denny, P. Muckelbauer, J. Etzl, "Dynamic Trust Applied to Ad Hoc Network Resources", *Autonomous Agents & Multi-Agent Systems Conference*, Melbourne, Australia, 2003.

[11] J. Kong, P. Zerfos, H. Luo, S. Lu and L. Zhang, "Providing Robust and Ubiquitous Security Support for Mobile Ad-hoc Networks", *ICNP*, Riverside, CA, 2001.

[12] M. Virendra and S. Upadhyaya, "Securing Information through Trust Management in Wireless Networks", *Workshop on Secure Knowledge Management (SKM 2004)*, Buffalo, NY, 2004.

[13] L. Zhou and Z.J. Haas, "Securing ad hoc networks," *IEEE Network Magazine, vol. 13, no. 6, pp. 24–30*, November 1999.

[14] C. Davis, "A localized trust management scheme for ad hoc networks", *Proceedings of 3rd International Conference on Networking (ICN'04)*, Mar. 2004.

[15] S. Marti, T.J. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks", *Mobicom 2000*, August 2000, pp. 255 265.

[16] Y. Huang and W. Lee, "A Cooperative Intrusion Detection System for Ad Hoc Networks", *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '03)*, Fairfax VA, October 2003.