A Framework for Key Management in Mobile Ad Hoc Networks

George C. Hadjichristofi[1], William J. Adams[2], and Nathaniel J. Davis[3]

[1] ghadjich@vt.edu

[2] wjadams@vt.edu

Bradley Department of Electrical and Computer Engineering,

Virginia Polytechnic Institute and State University, Blacksburg, Va 24060

[3] wjadams@vt.edu

Department of Electrical and Computer Engineering,

Air Force Institute of Technology, Wright, Patterson AFB, OH 45433

Abstract

Background: Key management in a mobile ad-hoc environment is complicated by frequently partitioning network topology. Recently proposed key management systems (KMSs) provide limited functionality in highly partitioned mobile ad hoc networks (MANETs). In this paper we present a framework for key management that provides redundancy and robustness for Security Association (SA) establishment between pairs of nodes in MANETs.

Methods: Our KMS uses a modified hierarchical trust Public Key Infrastructure (PKI) model in which nodes can dynamically assume management roles. The system ensures high service availability for the network members through a number of schemes. A novel behavior grading mechanism provides security criteria for the network nodes and aids the management functions of the KMS to revoke or reissue certificates for nodes. This mechanism is based on the notion of trust, and more specifically on SAs among nodes in the entire network.

Results: In this paper, we give an overview of the framework of the system and provide an overhead analysis of behavior grading. In addition, we present a performance analysis of the system based on certificate issuance and acquisition using a Monte Carlo and an NS2 simulation. Finally, we compare this KMS with current threshold cryptography schemes and we describe the system's implementation and interoperation with the FreeS/WAN IPsec product.

Conclusions: This KMS increases service availability for all nodes, increases flexibility in accommodating new nodes, minimizes pre-configuration, and can dynamically reconfigure itself based on the network environment.

Keyword: ad hoc networks, key management, security associations.

I. Introduction

A mobile ad hoc network (MANET) is a collection of independent mobile nodes. MANET links are wireless, which results in communications that are less dependable than wired links and have capacity constraints. A MANET is vulnerable to eavesdropping and the nodes in this network often have little physical protection. In order to counteract some of these threats, a MANET uses

mechanisms such as IP Security (IPsec), to secure transmitted data. However, prior to IPsec deployment, nodes need to establish Security Associations (SAs). During the establishment of an SA, two nodes authenticate one another using certificates, which are a primary form of identity verification. A Key Management System (KMS) creates, distributes, and manages these certificates. Thus, the KMS is at the heart of the network's defenses.

We developed a KMS that:

1) provides high service availability in highly partitioned networks,
2) avoids the transitivity of trust by introducing trust grading of the network nodes,
3) requires minimal pre-configuration during the network deployment phase,
4) does not make any assumptions about pre-existing trust, and
5) can accommodate new nodes joining the network.

The contribution of this paper is the KMS framework and, more specifically, the unique way the various components that comprise a KMS interoperate. Of significant importance is the introduction of a behavior grading scheme that interoperates with the KMS and aids the certificate revocation and reissuance process. We provide an analysis of the availability of the KMS and compare it with threshold cryptography schemes. We discuss integration of this KMS with FreeS/WAN IPsec [33].

The rest of the paper is organized as follows: Section II presents an overview of the previous work conducted related to key management. Section III presents the various roles that nodes undertake in the KMS and describe their functions based on those roles. Section IV describes the flexibility of the KMS with regards to new nodes joining the network. The mechanisms that enable SA establishment are presented in Section V. Next, in Section VI, we explain security aspects of our scheme, such as behavior grading. In Section VII, we give an overhead analysis of the KMS based on behavior grading. In Sections VIII and IX we present the performance analysis of our scheme and compare it to threshold cryptography schemes. Section X describes the integration of the KMS with FreeS/WAN IPsec. Finally, Section XI summarizes our contribution.

II. State of the Art

Other researchers have studied ways to develop key management schemes that enable the establishment of SAs in MANETs. Zhou and Haas [1] proposed a partially distributed key management service for MANET. Their proposal involved the use of a threshold cryptographic key. A system-wide public/private key pair, K/k, was created for the entire KMS in lieu of establishing a single CA. The system's public key, K, was known by all nodes in the network but the private key, k, was divided into n shares (s1, s2 … sn), assigning one share for each server. In addition, each server was pre-configured with the public key of the other servers. When signing a certificate, a server used its partial private key to generate a partial signature. The partial signature was then sent to an arbitrary selected server, which computed the certificate from the partial signatures and sent it to the node that requested the certificate. Distribution of trust was achieved using threshold cryptography with an (n, t+1) configuration. The service periodically computed new shares of the private key in order to tolerate compromised servers.

Authors in [2] proposed a similar threshold cryptography system that allowed new nodes joining the network to obtain a share of the KMS' private key. The advantage of this scheme as compared to [1], was that it increased availability since any t+1 nodes in the local neighborhood of the requesting node could issue or reissue certificates. In addition, the load of key management service was spread over a higher number of servers. The authors implemented both implicit and explicit revocation. Explicit revocation required all the nodes in the network to maintain a certificate revocation list. Based on other experts in the field [5], this system seemed to be vulnerable to the Sybil attack [7] because of the network-wide distribution of the private key.

Yi and Kravets [3] extended the work done in [1] for application in a MANET. They defined tunable parameters that could be used in the operation of their KMS. For revocation, they implemented a

simple certificate revocation list approach. Nodes built full revocation certificates using partial revocation certificates transmitted via a network-wide flood.

The authors in [4] continued this line of investigation and applied the KMS proposed by Zhou and Haas [1] on a cluster-oriented network. Cluster heads were assigned the role of signing certificates for other nodes. The authors introduced the idea that nodes must present a certain number of warranty certificates verifying their credentials before they could obtain a certificate from a cluster head (and become full members of the network). Those warrants were obtained from existing full members of the network.

Unlike our KMS, these threshold cryptography KMSs [1][2][3][4] assumed a certain level of pre-configuration. This level of pre-configuration provided less flexibility in terms of initializing the KMS and dynamically setting up CAs during the network lifetime. In addition, all of the schemes, except the one in [2], seemed to be unsuitable for MANETs with relatively high mobility and low connectivity, since a single node was dependent on a group of servers to obtain partial certificates. Furthermore, the combination of wireless nodes being unavailable due to the inherent nature of a MANET environment, or compromised due to malicious attacks, could render the key management service inoperable. The low availability of these schemes is shown in our analysis in Sections VIII and IX. In our scheme a single Delegated Certificate Authority (DCA) could provide services to nodes. As a safeguard, we counteracted the possibility that the DCAs could be unavailable through the use of Trusted Peers (TPs) serving as repositories for existing nodes and generic nodes serving as Temporary Certificate Authorities (TCAs) for new nodes. (For the purpose of this KMS we refer to CAs as DCAs implying the existence of a Root Certificate Authority (RCA).)

In addition, current research in threshold cryptography schemes did not fully address two important issues: information propagation and verification across multiple DCAs, and revocation authorization and response. The first issue is that new nodes joining the network have to present their information (e.g. public key) to multiple CAs to be able to obtain a sufficient number of partial certificates. Their information needs to be communicated to the CAs via out-of-band methods so that the authenticity of the information is verified by other CAs. In a highly dynamic environment, certificate reissuance can be hindered if a node does not communicate via out-of-band methods with a sufficient number of CAs. This problem can be eliminated to some extent by having the new node obtain a certificate from a trusted off-line RCA. However, in an open system where nodes leave and join the network, enforcing that all new nodes obtain an RCA certificate is unworkable. Authors in [4] dealt with this issue through the usage of warrants but assumed that the warrant issuers were trustworthy. In addition, the requirement of collecting a number of warrants with out-of-band methods was not very flexible. In our KMS, we employ non-repudiation through a series of transactions checks to securely communicate new nodes' information among CAs. Also, we do not employ threshold cryptography, and that relaxes the constraint of communicating the information to a sufficient number of CAs before reissuing a certificate.

The second issue deals with revocation authorization and response. Revocation is a critical obstacle to the operation of a KMS in a MANET. It is used to revoke the certificate of a compromised node or CA. The limitation with implementing revocation with threshold cryptography is that a certain number of CAs has to be made aware and be convinced of the malicious activity of a particular node before they can issue partial revocation notices. Current research has not addressed how this revocation process is carried out. Some solutions did not discuss revocation [1][4]. Other approaches [2][3] were still susceptible to the high mobility and low network capacity of the MANET environment that lowered the responsiveness of the revocation process. Our approach consisted of a control plane of DCAs and TCAs that utilized various revocation methods combined with security alerts that notified the nodes at various levels of malicious activity. A behavior grading mechanism provided a common basis across the DCAs that justified and triggered revocation as well as security alerts.

The authors in [5] proposed a Public Key Infrastructure (PKI) anarchy model similar to PGP. The KMS allowed all the nodes in a MANET to issue certificates to each other. The nodes kept databases

of expired and updated certificates, and cooperated with each other to build chains of trust among them. The system relied on both explicit and implicit revocation executed by all nodes to provide sufficient security in the certificate chains. In their later work [8], the authors noted that this system required a costly initialization phase in terms of both overhead and time since each node had to build its local certificate repository to be able to use the services of the system. Moreover, this scheme utilized transitivity of trust, which had the fatal problem of the untrustworthiness of certificate chains that is inherent in PKI anarchy models.

The underlying assumption when deploying the anarchy model was that all the nodes in a MANET had the same role. We argue that nodes in a MANET might not have the same role, position, or physical capability. For example, some nodes might have higher battery power compared to other nodes and it is more reasonable that those nodes served as CAs. Based on this assumption, only a subset of nodes were CAs in our KMS.

In order to evade the transitivity of trust presented in their prior work [5], the authors presented an idea that each node could build a higher number of SAs with the help of its existing friends in a highly mobile environment [8]. They demonstrated that mobility increased the number of SAs established. Unlike in [8], SA establishment in our KMS was not dependent on the existence of a group of fully-trusted friends, prior to the KMS deployment. In addition, the authors did not investigate revocation for this scheme.

III. Description of the System

One of the objectives of our KMS system was to be flexible enough to provide sufficient functionality for existing as well as new nodes, while simultaneously providing reasonable security criteria to the nodes in order to establish trust between them. Functionality of the KMS referred to the ability of the KMS to provide services to the network nodes, the level of pre-configuration required for the KMS' nodes, and flexibility of the KMS to adjust to changes in the network environment (e.g. connectivity). Security of the KMS referred to the KMS' ability to provide guarantees of the correctness of the information supplied and its ability to trace the behavior of malicious nodes and respond appropriately. Our KMS has also been presented in [10].

In our network environment, we assumed that nodes could leave and join the network at any time. Nodes could generate their own cryptographic keys and were capable of securing communication with other nodes. Our KMS used a modified hierarchical model of three levels, shown in Fig. 1. The roles that were undertaken by the nodes in the hierarchical model were: RCA, DCA, and TCA.

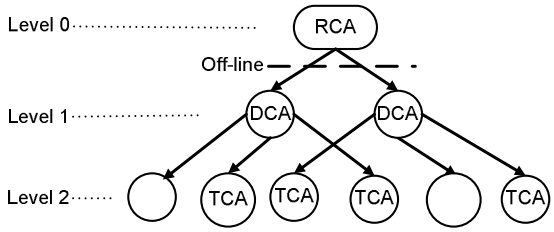


Fig. 1 Modified hierarchical PKI model.

An RCA was off-line and signed certificates for new nodes, which in turn could be used as proof for registering in the network or establishing temporary SAs. DCAs managed, stored and distributed certificates for the nodes within the network. The nodes that volunteered to be DCAs had more important positions within the network (e.g., administrators) and were not constrained by battery power or processing capabilities. This assumption was reasonable since not all the nodes in a network have the same position or function. Any node that was not a DCA could assume the role of a TCA. Nodes could be TCAs regardless of their position or reputation in the network. TCAs signed temporary certificates for physically collocated new nodes joining the network to enable them to

establish temporary SAs with other network nodes. In the next paragraphs, we describe the KMS functions of the nodes according to their particular role in the hierarchical model. We focus mainly on certificate issuance and revocation.

A. Root Certificate Authority (RCA)

An RCA node was off-line so that it would be protected from malicious attacks. The dynamic nature of a MANET did not allow it to actively participate in the network, and restricted it from contributing to the certificate management functions of the KMS, such as reissuance. Therefore, the RCA's main function was limited to providing certificates for nodes planning to register into the network. This function was significant because any node that had an RCA certificate could register into the network and serve as a DCA. Registration was carried out dynamically and a node was authorized by sending a request with its RCA-certificate to the existing control plane of DCAs. The notion of utilizing an RCA certificate to serve as a DCA differed from the work carried out in [1] and was based on work carried out in [2].

In [1], the authors assumed that CAs were pre-configured with the public keys of one another meaning that there was pre-existing trust between those CAs. In a real world scenario, this pre-existing trust can not exist unless the group of CAs' users physically meet prior to the KMS deployment and authenticate one another via out-of-band methods. Clearly, this is unworkable considering the nature of a MANET environment. In [2], the authors assumed that there is an off-line RCA that issued temporary certificates for CAs. We believed that this method provided more flexibility compared to [1] because it did not rely on some level of pre-existing trust among a group of CAs. However, it was limited by the requirement that all nodes joining the network served as CAs. Thus, all nodes had to authenticate using out-of-band methods with an off-line RCA before obtaining a share of the private key of the KMS and serving as CAs. In our KMS, only a subset of nodes had to possess RCA certificates to serve as DCAs compared to [2], which made the method in [2] more applicable to our system. Therefore, by incorporating this method in our system, RCA certificates promoted minimal pre-configuration of the DCAs at the initial steps of KMS's deployment and facilitated dynamic deployment of additional DCAs during the network lifetime.

New nodes that possessed an RCA certificate and did not intend to be DCAs could enjoy increased availability when joining the network by temporarily establishing SAs with other nodes using that certificate. In addition, possession of an RCA certificate allowed the nodes to register into the network at a higher trust level without physically encountering a DCA. If new nodes possessed an RCA certificate it was noted on their DCA certificate. In this way, higher confidence was placed on the authenticity of the certificate by existing nodes in the network (see Section IV; Table I).

An RCA certificate was obtained via out-of-band methods. The RCA certificate format was similar to the X.509 v3 format. RCAs utilized implicit revocation for their certificates. The life of the certificates was short because the off-line RCA could not inform the network nodes of a revoked certificate. The certificates expired after a period of time long enough for the nodes to enter the network and register with a DCA. Once a node registered in the network and obtained a DCA certificate, it did not need to periodically update its RCA certificates. This requirement would be unworkable given that the RCA was off-line.

B. Delegated Certificate Authority (DCA)

The KMS was comprised of a number of DCAs with the main objective of providing high service availability to the network nodes. Threshold cryptography schemes [1][3][4] decreased availability of the KMS in a partitioned network as the total number of CAs being compromised or unavailable diminished the ability of a node to obtain service. In our KMS, we increased availability by allowing a single CA to generate a certificate. The KMS enforced the verification and detection of invalid

certificates through the use of non-reputable transactions spanning over more than two nodes and through the use of the behavior grading scheme (see Section VI; Fig. 4).

The DCA certificate format was similar to the X.509 v3 with some extensions [34]. The extensions were: (1) the method of registering in the network, (2) possession of an RCA/TCA certificate, (3) the node's KMS role in the network, (4) the TCA-signed certificates validity period (see Section IIIC) and (5) the behavior metrics of the node in the network, as discussed in Section VI.

The DCAs utilized two revocation schemes in order to promote robustness and increase security in the network. As suggested in [3], nodes during revocation received a number of revocation notices from more than one DCA before taking the appropriate revocation action (e.g., dissolve an SA). The two methods of revocation were immediate and routine revocation. Immediate revocation was the process through which DCAs explicitly revoked the certificates of particular nodes. Immediate revocation was carried out at the DCA level based on security policies that were applied to the entire network. Certificates were revoked when certain levels of malicious activity were reached.

Routine revocation was introduced in order to increase certificate availability in a highly partitioned network environment. Routine revocation relaxed the time constraints that were imposed through periodic reissuance of the certificates. With routine revocation, DCAs advertised a certain certificate serial number or time of issuance before which all certificates would be invalid, as well as time of expiration. The idea of advertising a serial number or time of issuance, called First Valid Certificate was suggested by the authors in [13, pp. 384]. The objective of the First Valid Certificate was to keep certificate revocation lists short and allow certificates not to have a predetermined expiration time. In this KMS we extended this notion by adding the window of time field, which indicated the time certificates would expire. The expiration date allowed nodes a window of time to reissue their certificate and reestablish their trustworthiness in the network.

Routine revocation was implemented because we felt that it was inappropriate to implicitly (periodically) revoke certificates for our network environment. With implicit revocation, all the nodes periodically reissue their certificates. The certificates are valid for a period of time indicated on their certificates, and nodes have to reissue their certificates before the end of that period of time. The CAs set this period of validity at the time of issuance/reissuance of a certificate. The inability to dynamically adjust this period of validity during the lifetime of a certificate introduces two problems. First, the CAs are unable to dynamically balance the overhead imposed by implicit revocation with the security of the system. We argue that a cooperative environment such as a MANET, tends to be healthy most of the time and the majority of nodes abide by the network rules. Therefore, periodically revoking certificates introduces extra overhead in a resource-limited network when not necessarily needed. Setting a short validity period increases the overhead induced by implicit revocation. Increasing the period of validity decreases this overhead. However, a CA has to wait for the reissuance of the nodes' certificates to set a shorter validity period and thus use stricter security policies.

The second problem with implicit revocation is that it can decrease availability. Nodes may be unable to communicate with a DCA before their certificate expires. One way to diminish this problem is to have nodes reissue a certificate some time before the certificate expires. However, based on the dynamic nature of the network, nodes may not be able to accurately assess connectivity in the network and accurately predict the time needed to reissue a certificate. The utilization of the routine revocation introduced more flexibility by informing nodes that their certificate would expire and that they had to initiate a reissuance procedure. With the usage of the window of time field, nodes were given enough time to locate a DCA and reissue their certificate. Therefore, routine revocation in our KMS provided higher availability as opposed to implicit (periodic) revocation.

DCAs carried out routine revocation based on the system-wide security policy. Stricter security policy implied that a greater percentage of certificates in the network would be reissued within a period of time. The value of the window of time field was determined according to the DCAs' ability to communicate with other nodes/DCAs.

Unlike other schemes, the revocation schemes in our scheme were complemented with security alerts in order to increase their effectiveness in informing nodes of malicious behavior. Security alerts were carried out at both the DCA level and at the node level. The DCAs utilized security alerts to inform the nodes when certain threshold levels of malicious activity were reached. At the node level each node had its own individual security policies that were reported to the DCAs during registration or certificate reissuance. The node's security policies specified the behavior values (see Section VI) that it could tolerate for its trusted peers in the network. Once those behavior thresholds were reached, the DCAs informed that particular node. The node could then decide on the action it would take, such as breaking the SA with its peer. The security alerts introduced more checks and balances into the network since nodes were made aware of the levels of malicious activity throughout the network lifetime. The thresholds of malicious activity were based on behavior grading (see Section VI).

C. Temporary Certificate Authority (TCA)

In an open system new nodes that enter the network need to establish SAs with other peers when the DCAs are unavailable. A new node could obtain an RCA certificate with a short life and establish temporary SAs. This mechanism does aid new nodes joining the network and was incorporated in this KMS. However, it provided some level of inflexibility because it forced new nodes to register with an RCA using out-of-band methods. In order to further facilitate new nodes joining the network, we utilize TCAs.

In contrast to DCAs, any node in the network could serve as a TCA regardless of their position or reputation (see Section VI) in the network. TCAs signed temporary certificates for physically collocated new nodes to enable them to establish temporary SAs with existing network nodes. TCAs authenticated the new nodes using out-of-band methods or through other similar methods [6].

The TCA mechanism is different from the warrants scheme in [4]. The authors in that paper suggested that a node collected a series of warranty certificates from existing nodes via out-of-band authentication and sent them to threshold CAs to obtain partial certificates. We argue that this approach is not very dynamic and limits availability in highly partitioned networks because a node would have to authenticate out-of-band with a group of other nodes. In our KMS, possession of one or more certificates from collocated nodes was not required to obtain a DCA certificate. In addition, the certificate generated by a DCA after showing possession of one or more TCA certificates was not necessarily fully-trusted. When a certificate was generated, possession of TCA certificates was simply noted on the DCA certificate indicating that one or more other nodes verified the node's ID information. This functionality enabled each node to place a different trust level on its peer's certificate (see Table I).

The TCA certificate format was similar to the X.509 v3 certificate but was extended to include the DCA certificate of the TCA that signed the certificate. Since the TCA's trustworthiness was displayed on its DCA-signed certificate (via behavior grading), this extension provided the network nodes with an indication of the trustworthiness of the certificate issuer. By default, a TCA-signed certificate was given a lower trust level than a DCA-signed certificate. The level of trust of a TCA-signed certificate could be translated by other nodes as being equal to, or less than, that of the TCA that signed the certificate, as shown in row 7 of Table I.

The validity period of the TCA-issued certificates was controlled by the DCAs based on their network-wide security policies. (This period was recorded on the TCAs' certificate when issued or reissued by the DCAs.) The validity period acted as an indicator of invalid certificates issued by the TCAs, since TCAs attached their own certificates to the temporary certificates that they issued to the new nodes.

TCAs only utilized implicit (periodic) revocation. Certificates issued to new nodes were given short validity periods and allowed to expire. The short validity period motivated/forced nodes to register with DCAs and establish permanent credentials in the network.

D. Out-of-band TCA and DCA authentication

In order to assess the need as well as the effectiveness of the existence of the DCA and TCA control plane an analysis was carried out to investigate the time that it would take for a node to come to close proximity to a DCA or TCA and obtain a certificate via out-of-band methods. The mobility model used was the Random Waypoint model. Fig. 2 depicts the time required for a node to come within 5 meters of a DCA or TCA when moving at 1 m/s, and 5 m/s. We assumed, for the purpose of this investigation, that once a node came within 5 meters from its peer it could modify its direction of movement and move closer to its peer to facilitate out-of-band authentication. This assumption models real life interaction between people who see someone they know or wish to meet and will walk out of their way to meet them. Fig. 2 showed that the time taken to come to close proximity to a DCA is relatively longer compared to a TCA. For example, a new node moving at 5 m/s in a 40 node network took approximately an hour to come close to a TCA and between 2 hours (20% DCAs) to 5 hours (10% DCAs) to come close to a DCA. Thus, the existence of the TCA functionality provided increased service availability to a new node until it could register with a DCA. Even though, these results could vary with a different mobility model they still provided some level of validation of the importance of the existence of multiple DCAs and TCAs.

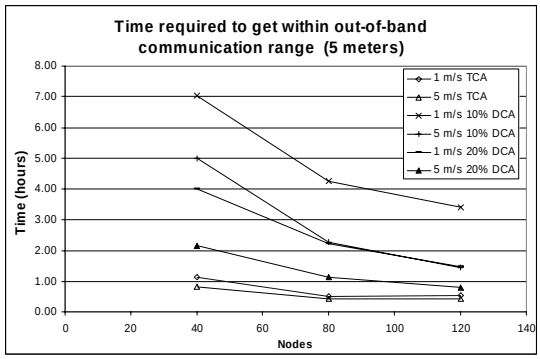


Fig. 2 Balancing availability with DCAs and TCAs.

IV. Flexibility in Joining the Network

New nodes that possessed an RCA or TCA certificate could establish temporary SAs that had short lifetimes. In this way, new nodes were encouraged to register with a DCA using out-of-band methods to increase their trustworthiness, as previously mentioned. The KMS system introduced further flexibility for new nodes joining the network by allowing them to obtain certificates via DCAs in two ways: (1) via out-of-band methods and (2) by sending their credentials through peer nodes (via online connectivity). A new node did not have to physically encounter a DCA because the authentication method was recorded on its certificate. The information served as an initial indicator of trust that could be placed on the certificates (i.e., the nodes' identity) during an SA negotiation.

Table I shows an example of how the combination of authentication method and certificate possession information was translated by a node to various trust levels. These trust levels could vary according to a node's security policy. Out-of-band authentication was considered more trustworthy as it implied face to face contact and earned nodes higher trust from their peer nodes during SA establishment. In this way, a node was motivated or forced to authenticate with a DCA via out-of-band methods in order to increase the trustworthiness that could be placed on its certificate by peer nodes. However, connecting through peers to obtain a certificate promoted more flexibility for new nodes, as described above. Based on row 1 of Table I, if node X during an SA establishment presented its DCA certificate to node Y, which was obtained by using its RCA certificate and authenticating via out-of-band methods, then this registration process provided the highest level of

trust with regards to the authenticity of the node's ID. As a result, node X was given an initial trust level of 100 by node Y. However, in row 3 of Table I, possession of an RCA certificate provided less confidence on the trustworthiness that could be placed on a certificate because of the inability of the DCAs to connect and dynamically obtain revocation information from an RCA. Even though an RCA certificate did not expire, it could have been revoked by the RCA. Therefore, the node was given a trust level of 90 by its peer. Furthermore, in row 4 of Table I, if node X obtained its DCA certificate by connecting to a DCA through peer nodes, then it was only given a trust level of 85 by node Y. Node Y trusted node X's certificate more in the scenario of row 4 compared to the scenario of row 5 because node X showed a TCA certificate to the DCAs, proving that another node in the network has verified the same credentials. Authenticating via the DCA through peer nodes without possessing any type of certificate would reasonably yield the lowest level of trust.
Thus, this scheme of trustworthiness assignment based on the authentication method and possession of TCA or RCA certificates promoted more flexibility because a new node that had no behavior grading could establish SAs with peer nodes at various trust levels.

TABLE I
TRUST LEVELS OF NODE X AFTER ESTABLISHING AN SA WITH NODE Y

X's Certificate	Authentication Method With DCA		Certificate Possession		Certificate Duration/ SA Lifetime	**Y's** Initial Trust level
	Out-of-band	Peer Connectivity	TCA	Root		
1 DCA	√		N/A	√	Long	100
2 DCA	√		N/A		Long	95
3 DCA		√	N/A	√	Long	90
4 DCA		√	√		Long	85
5 DCA		√			Long	65
6 RCA				√	Short	90
7 TCA			√		Short	≤TCA's

1 = lowest trust level;100 =highest trust level.

V. Facilitating SA Establishment

The certificates of nodes were required during an SA establishment so that nodes could authenticate each other. The certificates in any KMS can be stored on CAs or on repositories. Storing the certificates on CAs simplifies the management of the certificates and provides more control because expired or revoked certificates can be immediately deleted by the CA. However, this approach imposes a higher workload on a CA because the CA needs to store and send the certificates to the nodes. Furthermore, it may provide limited service availability in case all of the CAs are unavailable due to network partitioning. Specific nodes can be selected as repositories, in addition to the CAs, to distribute the certificates on behalf of CAs, increase availability, and decrease the workload of the CAs. However, utilizing repositories introduces communication overhead because CAs have to periodically update the revoked certificates stored on the repositories. In addition, as the various functionalities of the KMS are spread over more nodes the KMS becomes more vulnerable to security attacks. Another limitation is that the selection of a node to be a repository should be carefully assessed so that a node is not malicious and it can be trusted to provide the certificates to the nodes in place of the DCAs. In our scheme, we utilize TPs to serve as repositories. TPs were the

nodes that trusted a particular peer i.e., shared an SA with that peer. TPs only stored the certificates of the nodes that they trusted. This approach provided higher guarantees with regards to certificate distribution because those TPs were more likely to distribute certificates for their peers. By default, the TPs automatically obtained the certificates of peer nodes during SA establishment, thus simplifying the repository selection process and dynamically assigning repositories. In this way, the workload of managing the certificates was spread among all the nodes in the network.

At the beginning of an SA establishment, an existing network node queried any of the available DCAs. If all of the DCAs were unavailable due to network partitioning, a node obtained a peer's certificate by utilizing the TPs of its peer, as shown in Fig. 3. In this example, node C wanted to establish an SA with node F. Since all DCAs were unavailable, node C queried node F for its list of TPs. Once node C received that list containing the addresses of node A, node D, and node I, it could query any of them for F's certificate. The selection of which TPs to query was based on node C's perception of their trustworthiness. First, node C would query any "mutual" TPs, in this case node D, that it shared with the peer node. If node D was unavailable, the second approach would be to query the rest of the TPs in node F's list, nodes A or I, as those were unknown and presumed less trustworthy. Node C could also elect to obtain its peer's certificate from one or more TPs, if available, and select the most recent one.

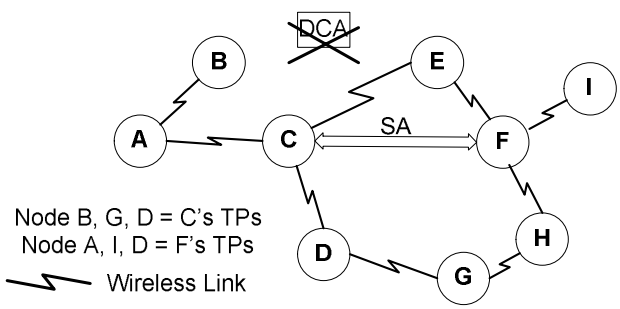


Fig. 3 Establishing SAs without the help of DCAs.

A high number of TPs per node increased the probability that a node could establish SAs with its peers, an advantage that can be proven mathematically. In a network where every node trusted at least one other node and all DCAs were unavailable, the number of additional SAs that were established could be calculated using (1).

$$SAs\ w/out\ DCAs = \frac{N(N-1)}{2} - \left\lceil \frac{N}{2} \right\rceil = \left\lfloor \frac{N(N-2)}{2} \right\rfloor \tag{1}$$

In the equation above, N is the number of nodes in the network, excluding DCAs. Equation (1) assumed that each node required the establishment of an SA with every other node in the network when no DCAs were available. (The ceiling value of N/2 accounted for the scenario that the network might have an odd number of nodes.)

Table II derived from (1) and shows the effectiveness of the proposed scheme with regards to availability. For example, in a network of 80 nodes, 3,120 additional SAs could be established without requiring the presence of a DCA. The effectiveness of this functionality does not imply that the presence of DCAs was not required at all, but rather that the system could be independent of DCA availability during SAs establishment, if needed.

It is important to note that the ruleset of the certificates used in the KMS differed from the X.509 v3 ruleset because multiple versions of a single certificate could exist within a window of time. That window of time was imposed by the DCAs through routine revocation (see Section IIIB). Each version of the certificate of a node could potentially contain different behavior grading information (see Section VI) but not different personal information (public key, ID). The DCAs held the

certificates with the most up-to-date behavior grading information. A node could establish an SA with a peer according to its individual security policies or trust thresholds, if it was satisfied with the reputation of that peer (that was recorded on the peer's certificate). If the node did not trust its peer based on that certificate and the certificate was not the most recent one, the node could obtain the latest certificate from a DCA, or if the DCAs were unavailable, obtain the certificate from other TPs of that peer. Since the possession of the node's most updated certificate by its TPs increased its chances of successful future SAs establishment with other nodes, a node was encouraged or motivated to periodically obtain and distribute an updated copy of its certificate to its existing TPs. In addition, through frequent updates the node reinforced its status as a trustworthy node. The frequency of this update was based on the node's security policy.

TABLE II
ADDITIONAL SAS ESTABLISHED WITHOUT DCAS

Nodes	SAs
10	40
20	180
40	760
80	3120
100	4900

Copies of the certificates of a node (including behavior grading) were stored on all the DCAs as well as on the TPs of that node. If a node's certificate was requested from a DCA and the DCA had recently received updated behavior grading information for that node, then the DCA reissued an updated certificate to reflect the latest behavior grading prior to distributing it in the network.

The notion of "friends" that was used in [8] is different from the notion of TPs used in our KMS. The authors in [8] presented the idea that each node could build SAs with the help of its existing (i.e., pre-configured) friends. Based on the results presented, they assumed a number of pre-existing friends for each node and equated SAs with complete trust. According to [8], two friends "trusted each other to *always* provide correct information about themselves and they had already established SAs between each other." Friends signed certificates for other nodes with which they shared SAs. In our scheme, two TPs did not *always* trust one another to provide the correct information even though they shared an SA. We assumed that absolute trust did not exist between friends and adopted a behavior grading scheme that integrated a node's behavior with its identity (see Section VI). In our KMS, TPs did not sign certificates for each other but only acted as repositories when the DCAs were unavailable. In addition, the number of a node's TPs was not fixed but fluctuated according to its reputation/behavior within the network.

VI. Balancing Flexibility and Availability

In order to balance the flexibility and increased availability of the KMS, security was provided by introducing two concepts in addition to revocation and security alerts: non- repudiation and behavior grading.

First, all transactions of the KMS were verified by at least two other nodes or DCAs in a non-reputable manner. The originating nodes signed the transactions of the KMS, providing proof of the origin of each transaction. This functionality was important because it prevented any node or DCA from modifying the data transferred and allowed the detection of malicious activity by those nodes. An example of this procedure is certificate issuance, which is shown in Fig. 4. In step 1, Node A digitally signed its personal information and sent it to DCA1. DCA1 authenticated node A out-of-band and generated a certificate for node A. In step 2, DCA1 forwarded node A's signed information and/or node A's certificate to DCA2 and DCA3. In step 3, DCA 2 and DCA3 sent an acknowledgement to node A, which could be a hash of its signed information. In this way, DCA1 could not modify the information because it was signed by node A. DCA1 could elect to

communicate with one or more DCAs, based on the security policy of the network as well as the network environment. Since the certificate issuance involved more than one DCA, it provided a balance of power and ensured information propagation. Any malicious behavior was recorded by the behavior grading scheme, the description of which follows.

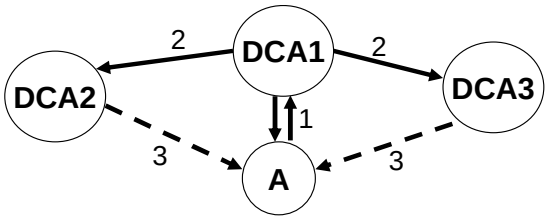


Fig. 4 Certificate issuance.

The KMS maintained sufficient levels of security by combining node authentication with an additional element, *node behavior*. A behavior grading scheme required each node to grade the behavior of other nodes. It was envisioned as a central, data processing layer, as shown in Fig. 5. At the lower layer, an intrusion detection system (IDS) [14] or monitoring scheme [15][16] could provide periodic performance observations to the network nodes. In a self-organized environment, such as a MANET, each node could utilize both the IDS information and the behavior grading information from the KMS to decide whether to trust or distrust a peer. In [11] we provided a method that a node could use to aggregate feedback from the behavior grading scheme of the KMS and from an IDS in order to produce a reputation index. The node could use that reputation index and decide whether to trust its peer based on its individual security policy. Once the node decided to trust or distrust its peer it reported its trust decisions to the behavior grading mechanism of the KMS. The behavior grading scheme would then collect this information and tie it to the certificates of the nodes. Existing nodes could in turn use these credentials to negotiate with their peers and decide whether to establish or dissolve SAs.

In addition, the KMS utilized this information to set its security policies related to reissuance, revocation, and security alerts thresholds. Thus, the KMS could dynamically assess the malicious activity in the network and initiate revocation or utilize security alerts. This layered approach meant that the behavior grading system was independent of the type of IDS since nodes could utilize any type of feedback generated by an IDS, such as routing activity.

The premises of the behavior grading scheme were based on existing concepts that were deployed in reputation management system [14][15]. However, the significance of the behavior grading scheme was that its parameters recorded the results from the aggregate input collected about nodes' activity instead of grading a particular activity (e.g., a node does not forward packets). Nodes could collect different inputs from one or more IDSs or any other observations that could be aggregated and recorded in a binary form: trusting or distrusting a node. The notion of utilizing different types of feedback information corresponds to real life situations. In real life, we choose our trusted friends by considering a number of different situations and experiences with an individual rather than one specific experience. Each individual has different criteria when deciding whether to trust someone and can give more emphasis on different experiences. Another important aspect of aggregating input in binary form was that the overhead imposed by the behavior grading scheme was not high except during the initial deployment of the network when all nodes desired to establish SAs with one another. Furthermore, in a network that is healthy most of the time, fluctuation of trust and thus reporting of trust/distrust would tend to be low. This concept is demonstrated with the pyramid shape in Fig. 5. Information flowed from IDS to the behavior grading scheme to the KMS. The higher the layer of a function on the pyramid, the less information needed to be communicated to a fewer number of DCAs.

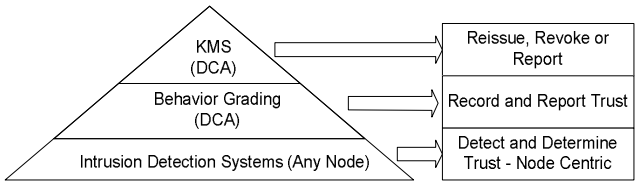
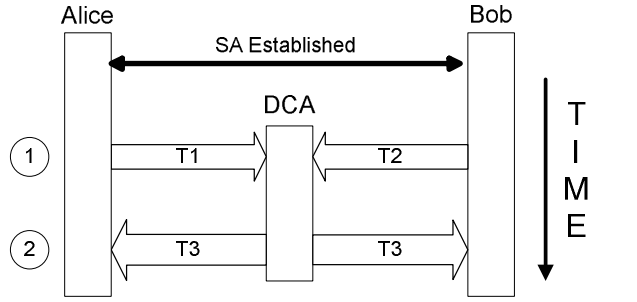


Fig. 5 Incorporating node activity into the KMS via behavior grading.

Overall, the behavior grading scheme provided the nodes and the KMS with a layer of abstraction of the trustworthiness of nodes, which was based on the overall activity of the nodes in the network. Through behavior grading, the nodes were motivated to do what was best for them while at the same time contributing to the entire network. Nodes were not as dependent on strict identity verification since they also had the ability to judge the trustworthiness of a peer node based on its behavior in a network as perceived by all nodes in the network. As a result, the need to periodically reissue certificates (via routine revocation) to enforce higher security in the KMS was not as frequent.

The behavior grading scheme recorded the nodes' level of trustworthiness using three parameters: positive reputation, negative reputation and complaint counter. *Positive reputation* indicated the number of TPs of a node. After an SA was established between two nodes, the two nodes reported the ID of their new TP to a DCA as shown in Fig. 6. The nodes reported their trust to a different DCA than the one that distributed the certificates to them prior to the SA establishment. This functionality allowed DCAs to check the integrity of the certificates distributed by other DCAs, and prevented any DCA from modifying the reputation or any other information of a particular node. Furthermore, it was required that both nodes registered their SA in order to get a positive reply that their trust had been recorded on their certificates. This mechanism enforced the notion of having more than two nodes be involved in a transaction and discouraged selfish nodes, since their peer was informed if they did not report their trust. Overall, positive reputation motivated the nodes to collaborate and improve their reputation, and allowed re-socialization of nodes that may have being wrongly victimized in the network. Messages T1 and T2 in Fig. 6, represents the report of trust from each party. The curly brackets in each message followed by the key of the corresponding party denoted that the messages were signed from each party providing non-repudiation. Message T3 was comprised of messages T1 and T2 as well as the updated certificates of each peer node. Receiving the updated certificates was optional, as discussed in Section VII.



$T1=\{Trust\ Bob,CertB\}k_A$
$T2=\{Trust\ Alice,CertA\}k_B$
$T3=\{Trust\ Bob\}pk_{DCA1}+\{Trust\ Alice\}pk_{DCA1}$ and/or CertA and/or CertB

Fig. 6 Recording of positive reputation.

A node's *negative reputation* indicated the number of peers that no longer trusted a particular node. It was motivated by the notion that nodes in a network, like people in our society, have a natural tendency to complain about other nodes/people. If a node deemed that its peer was no longer trusted it could elect to break the SA with its TP and inform the DCA. In our KMS, a complaint inferred that the complaining node had no SA with that particular peer.

Once the DCA obtained the complaint from a node, it recorded the complaint and sent a copy of the reply and/or updated certificates to both nodes that had an SA (see Fig. 7). In this way, the malicious node was in effect isolating itself by decreasing the number of nodes that trusted it in the network. In addition, the notification sent to the malicious node indicated immediate repercussions by negatively affecting that node's reputation.

However, if a malicious node complained about a "good" node, then the notification from the DCA would inform the "good" node to stop communicating with the malicious node that complained, in case they still had an SA. Thus, in effect the complaint isolated the malicious node. Even though, in this scenario the behavior record of a good node would be falsely modified, the KMS would not revoke the certificate of that node based on a single complaint. The KMS judged revocation based on the overall behavior of the node in the network with regards to the network-wide security policy. Furthermore, the ability to give a "bad" grade by reporting distrust was independent of the existing reputation of the nodes. In this way, "good" nodes turning "bad" were prevented from "attacking" the rest of the network nodes.

A malicious node roaming the network could only complain about other nodes with which it had an SA. Therefore, the number of nodes that could complain about a particular node was limited to the number of its TPs. Furthermore, a malicious node could only submit one complaint for the peers with which it had SAs, thus avoiding a stacking attack. (A stacking attack occurs when a node keeps complaining about its peer, and builds up that peer's negative reputation.) In Fig. 7, message C1 represent the complaint of a peer (Alice). Messages C2 and C3 were comprised of C1 and an optional updated certificate for each peer. The messages of the complaint process were again signed providing non-repudiation.

The *complaint counter* was recorded at the same time that the negative reputation occurred. The complaint counter showed the number of times that a node complained about its TPs and thus discouraged a malicious node from roaming around the network, establishing SAs with peer nodes, and then complaining about them. A node would not associate with a node that complained about a large number of its TPs.

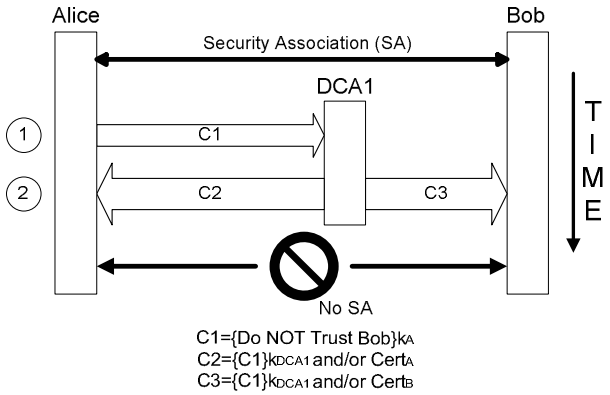


Fig. 7 Recording of negative reputation.

The criteria used for revoking a certificate reflected both the node's behavior history as well as its recent behavior. The decision to revoke was therefore based on both the total number of complaints as well as the frequency of complaints about a node. These values were modified by the DCAs to reflect the dynamic changes in a MANET based on the existing complaint frequency of the nodes in the network. Allowing for a high frequency of complaints made the system less sensitive to changes in behavior and vice versa.

DCAs set the network-wide policies for the various parameters of the KMS, such as the frequency of reporting malicious activity via security alerts, by negotiating a set of policies with each other. These

policies were based on each DCA's view of the network (and determined by factors such as network environment and malicious activity in the network).

In order to provide a *balance of power* among DCAs and increase the security of the system, each DCA could also grade its peer DCAs. The grading was based on the correctness of the DCAs' functions. Malicious activity could occur in the form of incorrectly displayed information by the DCAs and it was identified by the lack of proof that justified the information modifications. This requirement for balance of power among DCAs required a network to have at least three DCAs.

DCAs were given some immunity by the behavior grading of the KMS, as a motivation for their services in the network. This immunity protected them against a number of malicious attacks. The level of immunity was based on the network size and the type of environment over which the network was deployed (e.g. hostile vs. friendly). The immunity was again set up by a series of negotiations among DCAs.

VII. Overhead of Behavior Grading

The overhead of the KMS was categorized according to the behavior grading functionalities: positive reputation recording and negative reputation recording.

In the positive reputation overhead analysis, we considered the worst case that would require the most overhead. In this case, the KMS recorded all trusted pairs in a mesh connection, which is shown in (2). N represents the number of nodes in the network. The process involved two steps as shown in Fig. 6: nodes informed the DCA and the DCA confirmed and replied to the nodes. The summation of each step simplified to (3).

$$SAs_mesh,\ \omega = \frac{N(N-1)}{2} \tag{2}$$

$$\begin{aligned} Load_mesh &= \mathrm{Re}gistration\ \ + DCA_reply \\ &= (2\omega * |T_1| / |T_2|) + (2\omega * |T_3|) \\ &= (2\omega(|T_1| + |T_3|)) \\ &= (N(N-1)\,(|T_1| + |T_3|)) \end{aligned} \tag{3}$$

We investigated the negative reputation overhead analysis from the perspective of a single node. We assumed the worst case where a node had SAs with the rest of the network and complained about all of them. The process involved 2 steps, as shown in Fig. 7: reporting the malicious node and obtaining acknowledgement and updated certificates from a DCA. The summation of each step simplified to (4).

$$\begin{aligned} \mathrm{Re}port_DCA &= ((N-1)|C_1|) \\ DCA_response &= ((N-1)|C_1|) + (2|C_2|(N-1)) \\ Load_node &= \mathrm{Re}port_DCA + DCA_response \\ &= (N-1)(2|C_1| + |C_2|) \end{aligned} \tag{4}$$

Nodes elected whether they wanted to receive their updated certificates with every SA established or with every complaint. This decision was based on their security policies and any network resource constraints, and was controlled with parameters T3 and C2 in (3) and (4) (in Fig. 6 and Fig. 7) respectively. Thus, these parameters introduced flexibility in the system to dynamically adjust its overhead requirements based on the overhead constraints within the network. Stricter security policies imposed by the DCAs/nodes resulted in more frequent certificate updates that lead to nodes having more up-to-date behavior information of their peers. On the contrary, relaxed security policies had a lower impact on the network for both processing and network overhead but did not distribute the latest behavior grading information to the nodes.

The transactions among TPs could be carried out over existing secure tunnels, such as IPsec tunnels. An analysis of the IPsec overhead is presented in our previous work [17]. Even though the behavior grading information was automatically propagated to a subset of DCAs, information synchronization in MANET is beyond the scope of this paper. An examination of synchronization protocols for mobile devices has been offered in [36].

VIII. Monte Carlo Simulation

We investigated the performance of this KMS with regards to certificate issuance and acquisition to assess its effectiveness in providing service in a highly partitioned network environment. In addition, we compared certificate issuance in our scheme and in threshold cryptography schemes. Certificate issuance/reissuance required nodes to communicate directly with any of the available DCAs whereas certificate acquisition was dependent on the existence of DCAs and/or TPs.

This investigation was carried out with a Monte Carlo simulation analysis. This type of analysis is without a time axis. The simulation was done with locally developed C code. Seven hundred different static network topologies were generated and the connectivity was analyzed based on network parameters, such as radio range and node density. This number of network topologies ensured a 95% confidence interval of the mean. In our model, we ignored any communication limitations arising from the lower protocol layers, and assumed that there was no contention and transmission of data was error-free.

A. ***Performance metrics***

The results of interest for our system were:

Availability: the percentage of nodes out of the total nodes in a network that could contact any DCA or TP.

ASPLN: the Average Shortest Path Length between any node in a network and its nearest DCA/TP. ASPLN represented the path length to obtain a service. The reasoning was that shorter the path length, the higher the probability that the node could obtain service in a MANET environment. (The term “Shortest” in the ASPLN parameter signified that the shortest paths between all sources and destinations were derived using Dijkstra’s algorithm.)

The results of interest for certificate issuance for threshold cryptography schemes were:

PRN/D (Percentage of Reachable Nodes/DCAs): the percentage of DCAs that were reachable by a single node during certificate issuance when that node was not isolated in the network.

Iso (Node is Isolated): the percentage of times that a randomly selected node was isolated from the network due to partitioning and could not communicate with any DCAs.

B. ***Simulation parameters***

All nodes had the same radio range and all links were bi-directional. The simulation assumed that two nodes could communicate with each other when their radio range was equal or greater to the distance between them. The radio ranges selected in our analysis were between 100-300 meters, and facilitated data collection for both partitioned and connected networks (see Fig. 8). We assumed a fixed area of 1000 m x1000 m with 40, 80 and 120 nodes. The nodes were uniformly distributed in the network. The percentage of nodes selected to be DCAs were 10% and 20% of the total number of nodes as was used in previous research [30][31]. In addition, a higher percentage of DCAs would decrease the security of the network as it would provide a higher number of points of attacks and increase the communication overhead between the DCAs. Similar to DCAs, the percentage of nodes selected to be TPs were 10% and 20% of the total number of nodes. Thus, in a network with 20%

DCAs and 20% TPs, a node's peers could potentially acquire its certificate from a combined 40% of the nodes in the network but could reissue its certificate only from the 20% DCAs.
Guichal and Toh evaluated centralized and distributed service location protocols for pervasive wireless networks [30]. Parts of their analysis could be applied to the certificate issuance and certificate acquisition analysis of the KMS. They demonstrated that service availability increased with an increasing number of servers for fixed path lengths. Their results were based on an average node degree of connectivity of 3.2. However, those results did not suffice for the purpose of this research. We built on Guichal's and Toh's work by using a variety of node degrees in the network demonstrating how service availability varied with connectivity in partitioned and connected networks. We measured availability regardless of the path length and also provided an analysis of how the path length to obtain service varied with different levels of connectivity in the network.

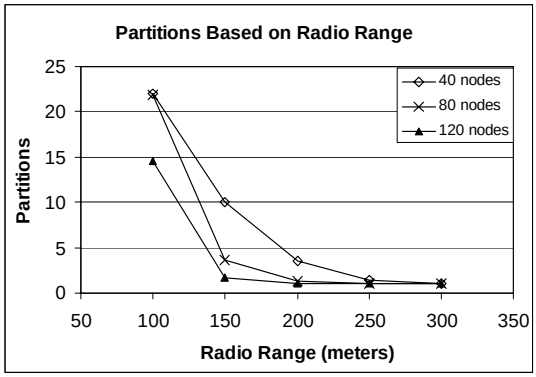


Fig. 8 Number of network partitions.

C. *Simulation Analysis*

Fig. 9 and Fig. 10 depict the availability of the KMS based on the radio range and, more specifically, its effectiveness in distributing certificates in a partitioned environment. As the number of DCAs increased from the centralized case (1 DCA) to 10% DCAs (of the total number of nodes), they distributed or issued certificates to nodes more effectively. In addition, the existence of 10% of TPs for a node (20% DCAs and TPs) could significantly increase SA establishment as a node could more easily obtain other nodes' certificates.

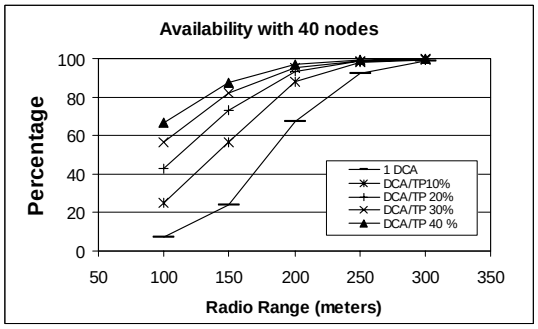


Fig. 9 Availability with 40 nodes.

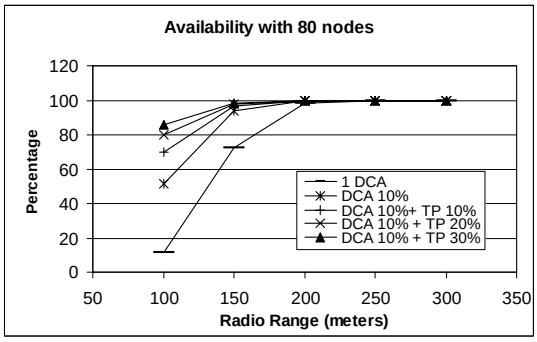


Fig. 10 Availability with 80 nodes.

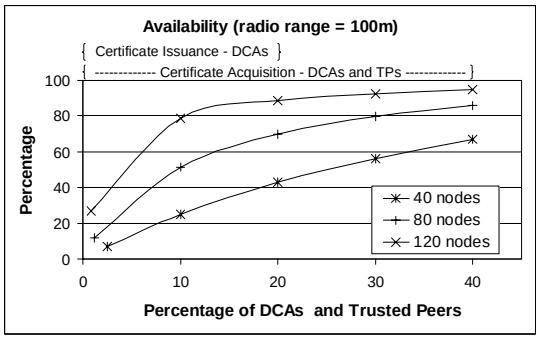


Fig. 11 Availability in a highly partitioned network.

By fixing the radio range at 100 meters, Fig. 11 illustrates availability in a highly partitioned network. The effectiveness of certificate issuance can be observed by the increase in availability as the DCAs increase from the centralized case on the left of the graph to a maximum of 20% DCAs. In the case of certificate acquisition, the whole range (centralized case-40%) could be considered, since nodes could either obtain certificates from a maximum of 20% DCAs and 20% TPs. The relative increase in availability, for the 10%- 40% range of DCAs and TPs, was higher for the 40 node network (25-67%) as compared to the relative increase of the 80 node (51-85%) and 120 node (79-95%) networks, because the 40 node network was more partitioned.

When using a higher radio range, such as 250m (see Fig. 9), the network became connected and the entire network was encompassed in a single partition. At this radio range, the availability increased to 100% and ASPLN was used to determine the effectiveness in the service availability of the KMS by illustrating the shortest path to a server. Fig. 12-14 demonstrate the variation in the path length depending on the number of DCAs/TPs, and radio range. As the number of DCAs/TPs increased, the path length decreased to a value of one (as shown in Fig. 13). Overall, the path length with the presence of DCAs/TPs varied between 1 and 3 hops, which was relatively low compared to the centralized case (see Fig. 12, Fig.13 - 1 DCA). In addition, within a 3 hop path length it was more likely that a node obtained service from a DCA or TP in a MANET environment.

It is important to notice the “hump” in the path length in Fig. 14 for the 40 node network, which was caused by the change in connectivity. This variation in path length was non-intuitive because ASPLN did not decrease as the radio range increased but fluctuated due to the existence of network partitions. The reason for this variation was that, initially, the short radio range partitioned the network into a number of small size partitions (see Fig. 8). Therefore, if a DCA was available in a small partition, the path length to that DCA was short, since it was constrained by the size of the partition. As the connectivity increased, the partition size of the existing partitions increased since nodes joined to form bigger clusters. If a DCA was available in that cluster, then the path length to that DCA was on average longer. As the radio range and the degree of connectivity increased even more, the ASPLN reached a maximum value (e.g., 190m for 40 nodes), which to some level represented the connectivity at which maximum cluster sizes existed. If nodes used an even higher radio range, clusters started to merge, enabling the nodes to find shorter paths to the DCAs/TPs. As a result, the ASPLN decreased exponentially. Thus, as the network connectivity increased from a highly partitioned to a connected network, the ASPLN increased to a maximum value and then decreased exponentially.

Another notable observation was that the ASPLN variation for the 80 node network was not similar to the 120 node network (see Fig. 14). The 120 network had a higher degree of connectivity and was at different phases of the path length variation. More specifically, the trend of the path length for the 120 node network was similar to the final stage of the 80 node network (>200m), where ASPLN decreased exponentially.

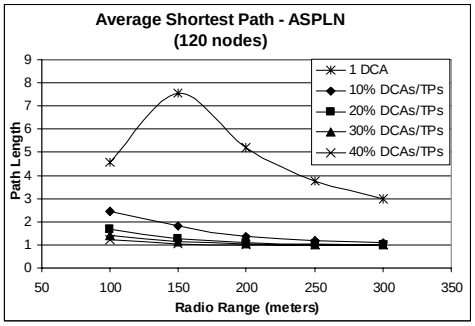


Fig. 12 Shortest Path to a certificate server (120 nodes).

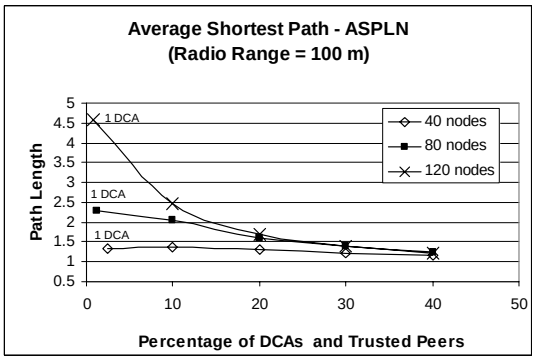


Fig. 13 Shortest path to obtain service (r=100m).

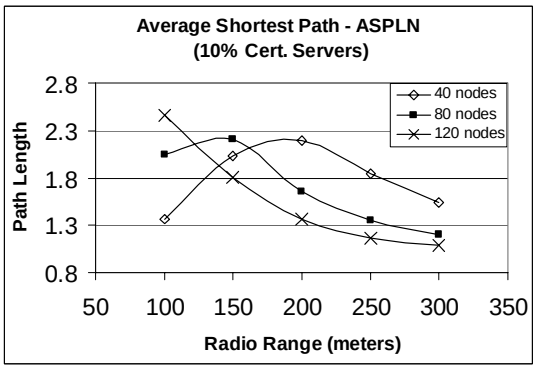


Fig. 14 Impact of radio range on ASPLN.

Threshold cryptography schemes lead to less accessibility to DCAs during certificate issuance especially in rapidly deployed partitioned environments. In these schemes, a single node had to contact a percentage of nodes acting as CAs and obtain partial certificates before it could build its certificate. Fig. 15 demonstrates the ability of a single node to reach a percentage of CAs, when that node was not isolated in the network (as shown in Fig. 16). For example, for a 40 node network with 10% DCAs and a radio range of 150 meters, a node could only reach 22% of the number of DCAs (see Figure 15) and it would be isolated 10% of the time (see Figure 16). If the node had to access 80% of the DCAs in the network to obtain partial certificates then it had to retransmit a number of times to be able to access a sufficient number of DCAs. In our KMS, access to 22% DCAs guaranteed certificate issuance from the first connection attempt to a DCA (assuming 4 DCAs).

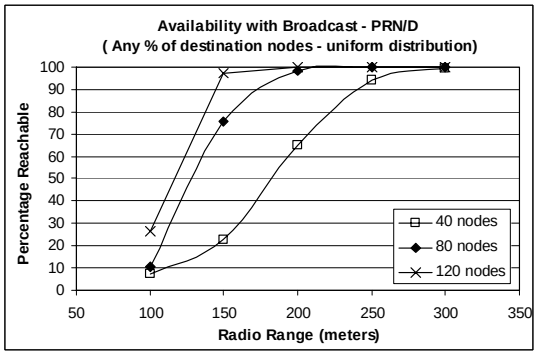


Fig. 15 DCAs reached by a single node.

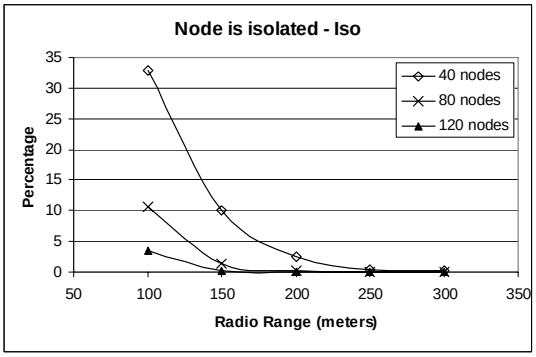


Fig. 16 Probability that a node is isolated in the network.

Summarizing, the effectiveness of the KMS is shown in Table III. In a partitioned network, certificate issuance with 10% DCAs for the 40 node network increased by 18% compared to the centralized case. Furthermore, certificate acquisition for the 40 node network with 10% DCAs and 10% TPs increased by 36%, as compared to the centralized case. Thus, it was not necessary that each node had a large number of TPs to obtain high service availability in the network. The path length to a server with 10% DCAs was less than 3 hops compared to less than 8 hops for the centralized case.

TABLE III
PERCENTAGE INCREASE IN SERVICE AVAILABILITY COMPARED TO THE CENTRALIZED CASE

Network Size	Certificate Issuance 10 % DCAs (%)	Certificate Acquisition 10% DCAs & 10% TPs (%)
40	18	36
120	51	61

Radio Range = 100 meters;

IX. Network Simulator 2 (NS2) Simulation

The purpose of this simulation was to investigate the impact of mobility on the service availability of the KMS. The simulation was carried out using NS2 [35]. Similar to the Monte Carlo Simulation, the functionalities of the KMS analyzed were certificate issuance and certificate requisition. A comparison of the KMS with certificate issuance in threshold cryptography schemes was also provided.

The NS2 simulation evaluated a wireless ad hoc environment and thus took into account communication limitations arising from the lower protocol layers such as contention. All nodes had the same radio range.

A. *Performance metrics*

Similar to the Monte Carlo analysis, a number of metrics were used to investigate certificate issuance and certificate acquisition.

The metric of interest for certificate issuance using threshold cryptography was *APRD* (Average Period to Reach DCAs). This metric indicated the total average period required to reach a group of DCAs.

The metrics of interest for certificate issuance and certificate acquisition for this KMS were:

Success Ratio: The average number of successful attempts out of the total number of attempts to communicate with a DCA/TP. The success ratio was independent of the frequency of the attempts and could provide an indication of the average time to reach a server by using (5).

$$Average\ \ Time_DCA/TP = \left(\frac{Period_between_Attempts}{Success_Ratio}\right) \tag{5}$$

APBNCS: (Average Period Between Non-Consecutive Successes). This metric was utilized to take into account network partitions. In a highly partitioned network environment there would be a burst of successful consecutive attempts when a node was located in a partition that contained one or more DCA/TPs. However, when the node moved away from that partition there was a burst of failed attempts, until the node was able to join another partition and communicate with other DCA/TPs. The APBNCS metric disregarded consecutive successful attempts and measured the period between

bursts of successful attempts. Thus, it more accurately reflected the impact of partitioning on service availability.

B. Simulation parameters

The simulation parameters and factors selected for the NS2 simulation are shown in Table IV. A more detailed description for the selection of these parameters follows.
The service availability of the KMS was dependent on the connectivity among nodes in a MANET. The BonnMotion [32] tool was utilized in order to select the parameters that would allow investigation of connectivity in highly partitioned as well as connected networks. The BonnMotion tool could generate mobility scenarios and statistically analyze them to provide information such as the number of partitions. Fig. 17 demonstrates the number of partitions based on the radio range and node density, and proves the validity of the selected parameters. It is important to note that the connectivity was approximately the same with different speeds. The scenarios were based on the Random Waypoint mobility model.

TABLE IV
PARAMETERS AND FACTORS FOR NS-2 SIMULATION ANALYSIS

Radio Range/meters	100, 150, 200, 300
Nodes	40, 80, 120(*)
Area – fixed	1000 x 1000 m^2
DCAs	10-20 %
Trusted Peers	10-20%
Communication Interval	15 seconds
Mobility Model	Random Waypoint Model
Node Speed / m/s	3, 5, 10, 15(*)
Simulation time /seconds	20000
Warm up period /seconds	1000
Routing Protocols	AODV, OLSR

* These values were not employed with OLSR to avoid redundancy

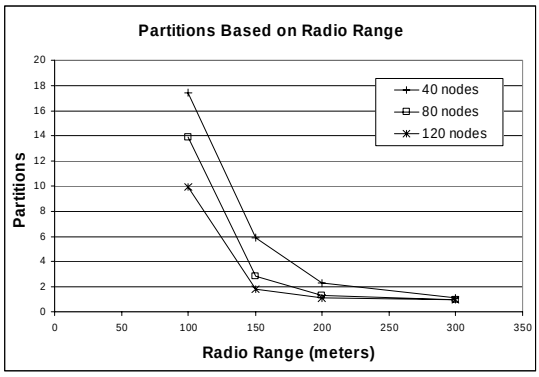


Fig. 17 Partitions based on the radio range used.

The percentage of nodes selected to be DCAs was 10% and 20% as was used in the Monte Carlo simulation analysis. Radio propagation used the two-ray ground model. The two-ray ground model predicts the received power as a deterministic function of distance. It represents the communication range as an ideal circle meaning that two nodes are connected if the distance between them is greater or equal to their radio range. Results of this research could be extended to consider other radio propagation models.
The random waypoint model utilized in this research was the only one offered in NS2 and is one of the most commonly used mobility models for simulations [18]-[22]. In a random waypoint model, every node is initially uniformly distributed within a two-dimensional space. Each node then moves

to a random uniformly distributed selected destination at a certain speed. When a node reaches its destination, it pauses for a certain time before it selects another destination and starts moving again towards that destination. The radio range of the nodes in the analysis was 100, 150, 200, and 300 meters. The speed was selected to be 3, 5, 10, 15 m/s in an attempt to map situations where nodes may move slower, such as in the case of people walking, or in the case where nodes move faster (e.g., in cars). The area was fixed to 1000 m x1000 m. The pause time was set to 5 seconds.

Unlike the distribution of the Monte Carlo analysis, the node distribution in a random waypoint model was not uniform and nodes tended to concentrate at the center of the two-dimensional space. Bettstetter analyzed the statistical properties of the random waypoint model in more depth [23][24].

The initial random distribution of mobile nodes in the random waypoint model does not represent the manner in which nodes distribute themselves when moving [25][26]. A warm up period of 1000 seconds was used to attain a steady state behavior as was suggested in [25]. In order to validate this warm-up period, simulation runs with warm up-periods of 100,000 seconds were carried out, which indicated that a longer warm up period did not impact the results obtained.

The routing protocols utilized in the simulation were the Ad Hoc On-Demand Distance Vector protocol (AODV) and the Optimized Link State Routing protocol (OLSR). The selection of these two protocols was based on their availability and proper functionality in NS2.

AODV is a reactive routing protocol for MANETs [27] whereas OLSR is a proactive routing protocol for MANETs [28]. With a reactive protocol the routing paths are built on demand whenever a node needs to send packets to a peer and it does not have a known route to that peer. On the other hand, with a proactive protocol a node maintains routing paths to its peers by periodically updating its routing table through the broadcast of control messages. Tao Lin investigated the deployment of reactive versus proactive protocols [29] and made recommendations with regards to selecting a routing protocol that is suitable for a particular MANET environment.

The communication interval was the frequency of repeating a particular scenario. In the case of certificate issuance and acquisition, the communication interval set the frequency with which a node attempted to communicate with a DCA/TP. The communication interval was set to 15 seconds for a number of reasons. First, a short interval could more accurately reflect the performance of the KMS during the dynamic changes of connectivity in the network for metrics that recorded time of execution. More specifically, the metric *APBNCS* recorded the period between non-consecutive successful attempts. Since successful attempts depended on connectivity, a shorter communication interval would more likely detect the point of break of connectivity and more accurately predict the time interval between communications when the network partitioned. In addition, in the case of *APRD*, the average period to reach a group of DCAs, the short period reflected the urgency to conduct enough DCAs to issue or reissue a certificate. Therefore, *APRD* reflected the shortest possible time to obtain a certificate.

C. ***Simulation Analysis***

Fig. 18 and Fig. 19 depict the Success Ratio of the KMS based on the number of DCAs and TPs and, more specifically, its effectiveness in distributing certificates in a partitioned environment. (The corresponding partitions based on radio range are displayed in Fig. 17.) Fig. 18 was generated while utilizing the AODV routing protocol whereas Fig. 19 was generated while utilizing the OLSR routing protocol. Even though the objective of this research was not to compare the routing protocols, Fig. 18 and Fig. 19 demonstrate that AODV provided higher Success Ratio compared to OLSR.

As previously mentioned, the effectiveness on *certificate issuance* could be observed by considering 10-20% out of the combined number of DCAs and TPs shown on the x-axis of the graph. For the case of *certificate acquisition*, the whole range of DCAs and TPs (centralized case-40%) could be considered, since nodes could either obtain certificates from DCAs or TPs. As expected, the Success Ratio for the centralized case, demonstrated by the data points on the far left of the graphs, was

lower compared to the scenarios that involved more than one DCA or TP. The existence of 10% DCAS (of the total number of nodes), improved the ability to issue/reissue a certificate. In addition, the existence of 10% of TPs for a node (20% combined DCAs and TPs) could more easily facilitate the establishment of SAs.

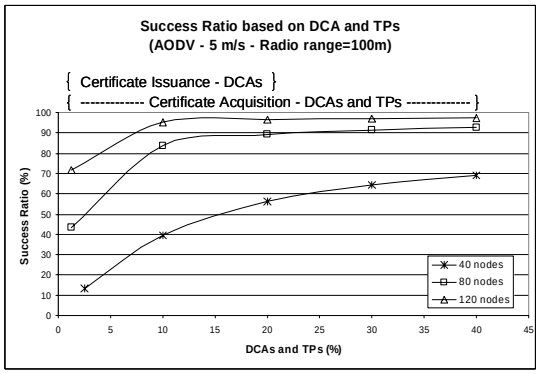


Fig. 18 Success Ratio with the AODV protocol.

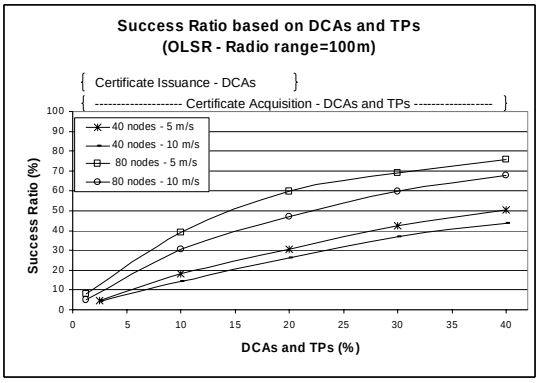


Fig. 19 Success Ratio with the OLSR protocol.

Fig. 18 shows that the relative increase in the Success Ratio as the number of DCAs and TPs increased was higher for the 40 node network as compared to the 80 node and 120 node network, because the 40 node network was more partitioned (as shown in Fig. 17). As a result, the 80 and 120 node lines tended to be horizontal or have a smaller gradient when utilizing more than 10% DCAs/TPs. The OLSR protocol was less sensitive to changes in connectivity due to its proactive nature and therefore a higher node density yielded an overall lower success ratio as compared to the AODV protocol (see Fig. 19). In addition, since OLSR did not reactively build routing tables, the Success Ratio to obtain a certificate was more dependent on the existence of DCAs/TPs as compared to AODV. Thus, utilizing a higher number of DCAs and TPs did not quickly yield an almost horizontal line as in the case of using 10% DCAs with AODV (see Fig. 18) but rather a more inclined line showing a higher relative increase in the Success Ratio with the usage of DCAs/TPs.

Fig. 20 and Fig. 21 indicate the impact of node speed on the Success Ratio with the AODV and OLSR protocols respectively. These graphs could act as a guide to derive the possible deviation of the Success Ratio with regards to the node speed. For example, with the AODV protocol the success ratio increased with the increasing node speed whereas with the OLSR protocol it decreased.

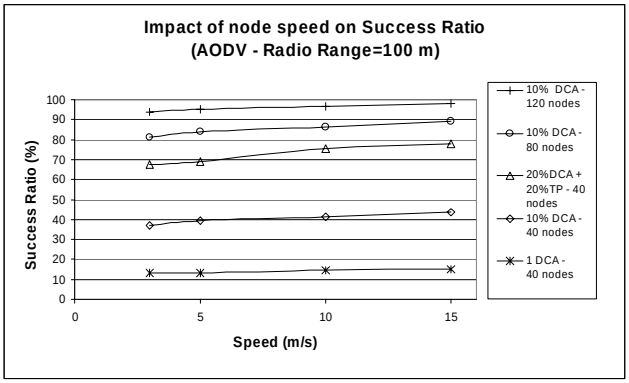


Fig. 20 Success Ratio deviation based on the speed of nodes (AODV).

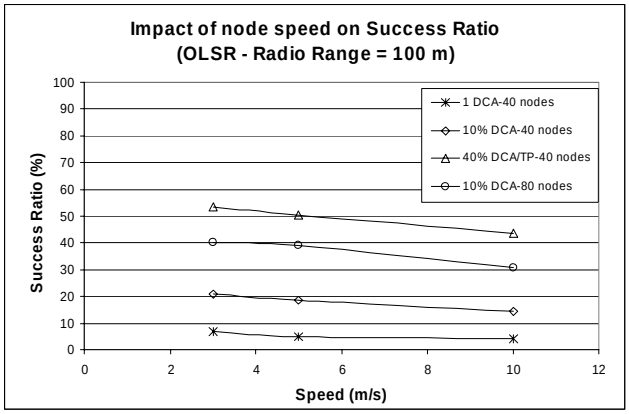


Fig. 21 Success Ratio deviation based on the speed of nodes (OLSR).

Fig. 22 and Fig. 23 indicate the variation of the Success Ratio with regards to the radio range of the network nodes. As the radio range increased, connectivity increased causing the Success Ratio to converge to 100%. The usage of DCAs and TPs pushed the Success Ratio to 100% at a shorter radio range compared to the centralized case. For example, in Fig. 22, the Success Ratio with the centralized case in a 40 node network converged to approximately 100% at 300 meters radio range instead of 200 meters when using 10% DCAs and 10% TPs. As previously mentioned the OLSR protocol was less responsive to connectivity and thus the radio range required to converge to 100% Success Ratio was longer.

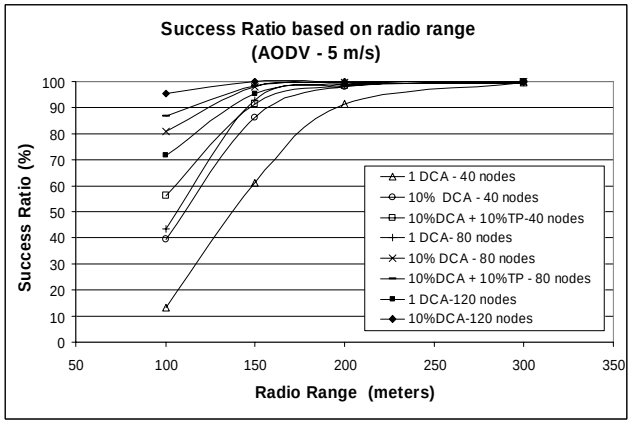


Fig. 22 Impact of radio range on the Success Ratio (AODV).

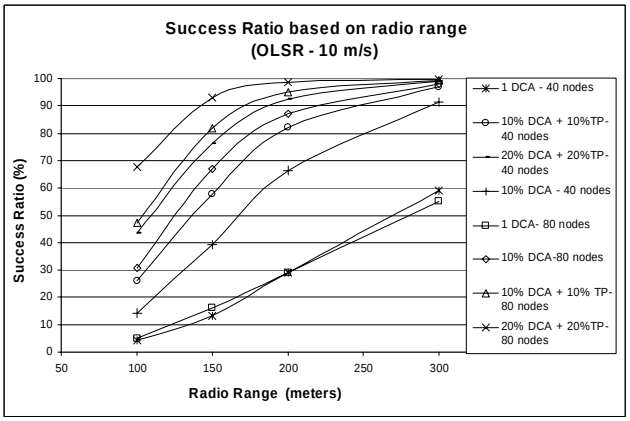


Fig. 23 Impact of radio range on the Success Ratio (OLSR).

Fig. 24, Fig 25, and Fig. 26 demonstrate the average period between non-consecutive successes and more specifically the impact of network partitioning for highly partitioned network environments (with radio range of 100m). In Fig. 24, with the centralized case, a node had to wait for approximately an average of 510 seconds (8.5 min) for the OLSR protocol and 320 seconds (5.3 min) for the AODV protocol in order to reissue its certificate or obtain a certificate of its peers to establish an SA. However, by utilizing 10% DCAs or higher the APBNCS was kept below 100 seconds (1.5 minutes) in most of the scenarios. As the number of DCAs and TPs in the network increased the APBNCS converged to 50 seconds and was less impacted by network partitioning. Fig. 25 depicts the impact of radio range on the APBNCS metric. As the radio range increased, connectivity increased and APBNCS converged to 30 seconds. With the OLSR protocol a node required a higher average period compared to the AODV protocol. Fig. 26 presents a different perspective of the APBNCS. As the speed increased the impact of partitions on the APBNCS was smaller. It is important to note that in the centralized case the average period decreased from 280 seconds to 130 seconds which was a higher rate compared to the other scenarios. However, the other scenarios already possessed a relatively lower APBNS compared to the centralized case.

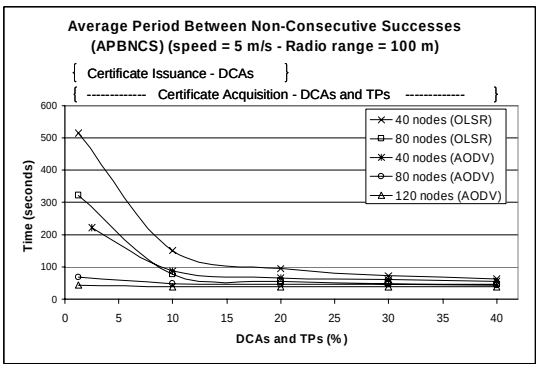


Fig. 24 Average period to obtain service.

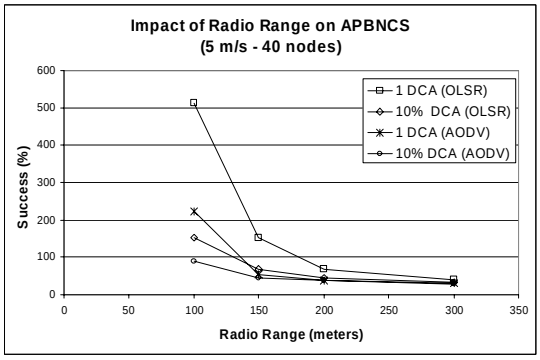


Fig. 25 Impact of radio range on APBNCS.

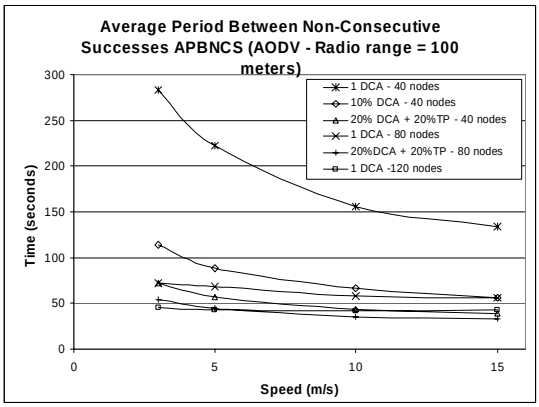


Fig. 26 Impact of node speed on APBNCS.

Threshold cryptography schemes lead to less accessibility to DCAs during certificate issuance especially in partitioned environments. Fig. 27 shows the time taken for a node to contact a percentage of DCAs. For example, the time taken to communicate with 100% of the 10% of DCAs (4 DCAs for 40 node network) and obtain partial certificates ranged between 900 and 1100 seconds for the OLSR protocol and 100 and 500 seconds for the AODV protocol. In our KMS, a single DCA could generate a certificate within a period of 100 seconds. Fig. 28 demonstrates how the average time to contact 100% of the DCAs varied based on the radio range. As expected, with the OLSR protocol a node required a longer time to contact a group of CAs.

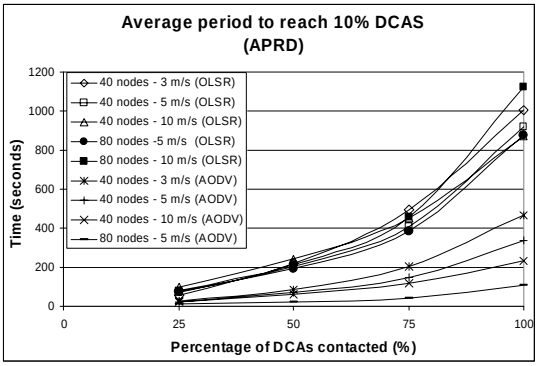


Fig. 27 Average period to issue a certificate.

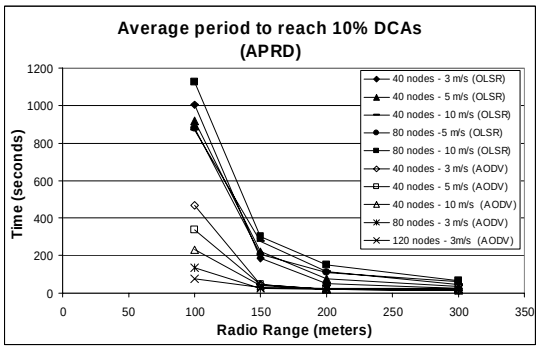


Fig. 28 Impact of radio range on certificate issuance.

Summarizing, the effectiveness of the KMS is shown in Table V. For example, in a partitioned network, certificate issuance with 10% DCAs for the 40 node network increased by 14% compared to the centralized case for both OLSR and AODV protocols. Certificate acquisition for the 40 node network with 10% DCAs and 10% TPs increased by 25% for OLSR and 43% for AODV, as compared to the centralized case.

TABLE V
PERCENTAGE INCREASE IN SERVICE AVAILABILITY COMPARED TO THE CENTRALIZED CASE

Routing Protocols	Network Size	Certificate Issuance 10 %DCAs (%)	Certificate Acquisition - 10% DCAs & 10% TPs (%)
OLSR	40	14	25
	120	35	55
AODV	40	14	43
	120	23	24

Radio range = 100 meters;

X. KMS Implementation

The KMS was implemented in the test bed shown in Fig. 31, which represents a MANET. The gateways had subnet nodes attached. Further information related to this testbed was presented in [9]. The objective of this implementation was to provide a proof of concept of the effectiveness of the KMS in distributing certificates and provide proof of its interoperability with the existing FreeS/WAN IPsec implementation.

The existing IPsec implementation offered limited service availability due to strict dependence on the DNSs. Nodes authenticated each other by obtaining the public key of their peers from the DNS (as shown in Fig. 29). If any of the DNSs were unavailable, which is a common challenge in a MANET due to network partitioning, an IPsec SA negotiation failed. In order to address certificate availability of the KMS the existing IPsec implementation was made aware of the various KMS functionalities that could provide the required authentication information needed to establish an IPsec SA. The increased service availability was provided through DCAs and TPs. The existing IPsec implementation was modified so that IPsec on each gateway communicated with a KMS client, instead of a DNS, whenever it had to establish an IPsec tunnel with a peer Gateway (see Fig. 30). The KMS client was deployed on each Gateway in the MANET and was responsible for collecting the required information on behalf of IPsec. The DCA functionality was deployed on only a few nodes.

The certificates on the DCAs complied with X.509 v 3 certificates and were extended to provide any other information, such as behavior grading. Once peers successfully authenticated one another and an SA was established, the IPsec mechanism reported the trust to the KMS client, which in turn reported the information to a DCA (positive reputation). In addition, each node reported its SA to the TopoView application. This network management service was implemented using Scotty [12] and showed the network topology as it dynamically changed. The TopoView application received this information and dynamically displayed the IPsec tunnels in the network. Fig. 31 shows IPsec SAs between gateways 10.0.01, 10.0.0.7 and 10.0.0.12 as displayed in TopoView.

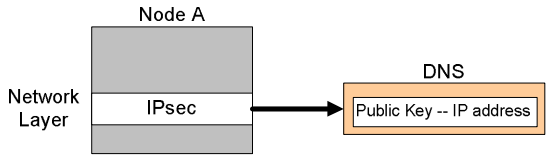


Fig. 29 Existing IPsec architecture.

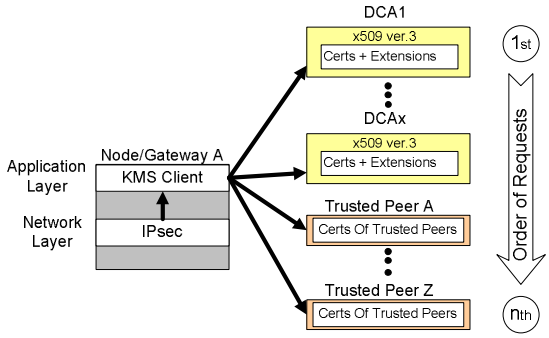


Fig. 30 Modified IPsec implementation.

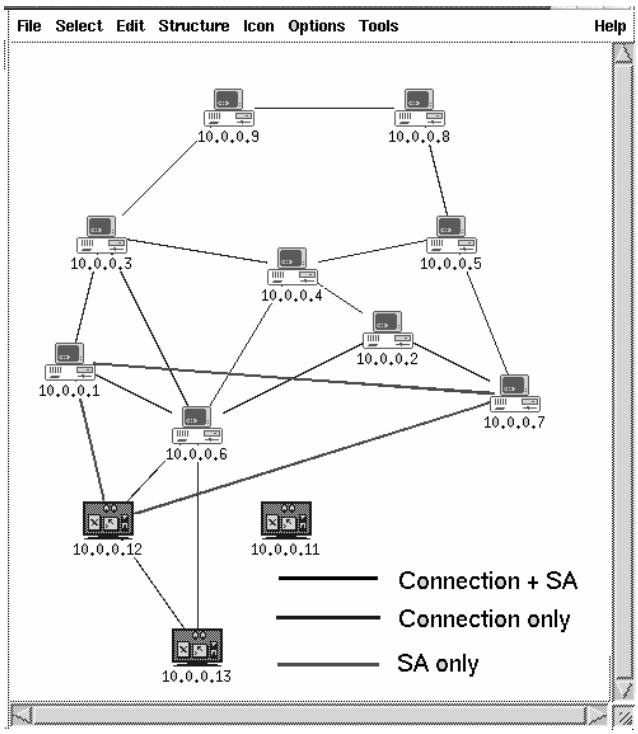


Fig. 31 MANET Topology.

XI. Conclusion

We have presented a framework for a distributed KMS that increased service availability for highly partitioned networks. Our system integrated a number of components in a unique way to overcome

the limitations of previous KMSs. The system utilized a modified hierarchical PKI model consisting of a control plane of RCAs, DCAs, and TCAs. The RCAs authenticated new nodes and issued them RCA certificates. New nodes could use the RCA certificates to register in the network and serve as DCAs, minimizing pre-configuration. In addition, new nodes could establish temporary SAs in the absence of DCAs, thus introducing more flexibility into the KMS. The DCAs issued, revoked, distributed and managed certificates based on the behavior grading of the nodes and the security policies at the network and node level. The TCAs aided new nodes to join the network by issuing temporary certificates whenever DCAs were unavailable. In addition, the TPs of each node acted as repositories, dynamically spreading the repository overhead and increasing the availability of certificates in a partitioned network.

Security in the KMS was provided via immediate and routine revocation, security alerts, behavior grading and non-repudiation. The behavior grading scheme of the KMS relaxed the need of relying on strict identity verification and allowed nodes to judge other nodes based on their trustworthiness. Trustworthiness was expressed in the form of network-wide SAs between nodes in the entire network. The KMS avoided transitivity of trust because it did not utilize chains of trust. The transactions of the system were recorded in a non-reputable manner and were verified by more than two other nodes, providing balance of power among nodes and DCAs.

Our simulations demonstrated that deploying a number of DCAs and utilizing TPs could significantly increase availability and aid SA establishment in a highly partitioned network environment. In addition, our scheme could provide higher guarantees for issuing certificates to nodes compared to threshold cryptography schemes.

The KMS was implemented and integrated with the existing IPsec implementation. The combination of DCAs and TPs provided higher functionality and facilitated the establishment of IPsec SAs between nodes.

References

[1] L. Zhou and Z. Haas, “Securing ad hoc networks,” IEEE Network, November/December 1999, vol. 13, no. 6, pp. 24–30.

[2] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, “Providing robust and ubiquitous security support for mobile ad hoc networks,” in Proceedings of the 9th International Conference on Network Protocols (ICNP), November 2001.

[3] S. Yi, R. Kravets, “MOCA: Mobile certificate authority for wireless ad hoc networks,” 2nd Annual PKI Research Workshop Program (PKI 03), Gaithersburg, Maryland, April 2003.

[4] M Bechler, H.-J. Hof, D. Kraft, F. Pahlke, and L. Wolf, “A cluster-based security architecture for ad hoc networks,” IEEE INFOCOM, 2004.

[5] S. Capkun, L. Buttyan, and J. Hubaux, "Self-organized public-key management for mobile ad hoc networks," IEEE Transactions on Mobile Computing, 2003, vol. 2, pp. 52-64.

[6] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong, “Talking to strangers: Authentication in ad-hoc wireless networks,” in Proc. Symp. on Network and Distributed System Security (NDSS), San Diego, Feb 2002.

[7] J. Douceur, “The sybil attack,” in Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS), 2002.

[8] S. Capkun, J.-P. Hubaux, and L. Buttyan, "Mobility helps security in ad hoc networks," presented at Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing, Maryland, USA, June 2003.

[9] L. DaSilva, S. Midkiff, J. Park, G. Hadjichristofi, K. Phanse, T. Lin, and N. Davis, “Network mobility and protocol interoperability in ad hoc networks,” IEEE Communications Magazine, November 2004, Vol.42, pp. 88-96.

[10] G. C. Hadjichristofi, W. J. Adams, and N. J. Davis, "A framework for key management in mobile ad hoc networks," in Proceedings of the International Conference on Information Technology Coding and Computing, Las Vegas, Nevada, April 2005, vol. 2, pp. 568-573.
[11] W. J. Adams, G. C Hadjichristofi, and N. J. Davis, “Calculating a node’s reputation in a mobile ad hoc network,” in Proceedings of the 24th IEEE Performance, Computing, and Communications Conference (IPCCC), Phoenix, Arizona, April 2005, pp. 303-307.
[12] “Scotty - Tcl extensions for network management applications,” http://wwwhome.cs.utwente.nl/~schoenw/scotty/, available 05/30/05.
[13] C. Kaufman, R. Perlman, and M. Speciner, Network Security: Private Communication in a Public World, 2nd ed. Prentice Hall PTR, Upper Saddle River, N.J. 07458, 2002.
[14] Y. Zhang and W. Lee, “Intrusion detection in wireless adhoc networks,” in Proceedings of the Sixth International Conference on Mobile Computing and Networking (MobiCom 2000), Boston, MA, August 2000.
[15] S. Buchegger and J.-Y. Le Boudec, "Nodes bearing grudges: towards routing security, fairness, and robustness in mobile ad hoc networks," in Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, 2002, pp. 403-410.
[16] P. Michiardi and R. Molva, “CORE: A collaborative reputation mechanism to enforce cooperation in mobile ad-hoc networks,” in Proceedings of the Sixth Joint Working Conference on Communications and Multimedia Security, 2002, pp 107 –121.
[17] G. C. Hadjichristofi, N. J. Davis, and S. F. Midkiff, "IPsec overhead in wireline and wireless networks for web and email applications," in Proceedings of the IEEE Performance, Computing, and Communications Conference (IPCCC), Phoenix, AZ, April 2003, pp. 543-547.
[18] R. V. Boppana and S. P. Konduru. “An adaptive distance vector routing algorithm mobile, ad hoc networks,” in Proceedings. of the 2001 IEEE INFOCOM and Joint the Computer and Communications Societies, 2001, vol. 3, pp. 1753-1762.
[19] G. Pei, M. Gerla and X. Hong, “LANMAR: Landmark routing for large scale wireless ad hoc networks with group mobility,” in Proceedings. of IEEE/ACM MobiHOC, 2000, pp. 11-18.
[20] J. J. Garcia-Luna-Aceves and M. Spohn, “Source-tree routing in wireless networks,” in Proceedings. of IEEE International Conference on Network Protocols, 1999, pp. 273-282.
[21] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark. “Scenario based performance analysis of routing protocols for mobile ad-hoc networks,” in Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking, 1999, pp. 195-206.
[22] E. M. Royer and C. E. Perkins, “Multicast operation of the ad-hoc on-demand distance vector routing protocol,” in Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking, 1999, pp. 207-218.
[23] C. Bettstetter, “Mobility modeling in wireless networks: categorization, smooth movement, and border effects,” in ACM Mobile Computing and Communications Review, 2001, vol. 5, no. 3, pp. 55-67.
[24] C. Bettstetter, G. Resta, and P. Santi, “The node distribution of the random waypoint mobility model for wireless ad hoc networks,” IEEE Transactions on mobile computing, July-September 2003, vol.2, No. 3.
[25] J. Yoon, M. Liu, and B. Noble, "Random waypoint considered harmful,"in Infocom03, April 2003.
[26] T. Camp, J. Boleng, and V. Davies, “A survey of mobility models for ad hoc network research,” in Wireless Communication & Mobile Computing (WCMC): Special issue onMobile Ad Hoc Networking: Research, Trends and Applications, 2002, vol. 2, no. 5, pp. 483-502.
[27] C. E. Perkins, E. M. Belding-Royer, and S. R. Das, “Ad hoc on-demand distance vector (AODV) routing,” Internet Engineering Task Force (IETF) draft, November 2002.Available at http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-12.txt.

[28] T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot,"Optimized link state routing protocol," Internet Engineering Task Force(IETF) draft, March, 2002. Available at http://www.ietf.org/internet-drafts/draftietf-manet-olsr-06.txt.

[29] Tao Lin, "Mobile ad-hoc network routing protocols: methodologies and applications," Dissertation, Virginia Polytechnic Institute and State University, available April 2004.

[30] G. Guichal and C.-K. Toh, "An evaluation of centralized and distributed service location protocols for pervasive wireless networks," in 12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2001, vol.2, pp. E-55-E-61.

[31] K. Phanse and L. A. DaSilva, "Protocol support for policy-based management of mobile ad hoc networks," 2004 IEEE/IFIP Network Operations and Management Symposium (NOMS 2004), Seoul, Korea, April 19-23, 2004, pp. 3-16.

[32] "BonnMotion - A mobility scenario generation and analysis tool," http://web.informatik.unibonn.de/IV/Mitarbeiter/dewaal/BonnMotion/, available 05/10/05.

[33] Linux FreeS/WAN, http://www.FreeS/WAN.org, available July 30, 2003.

[34] S. Kent, "Privacy enhancement for internet electronic mail: Part II: Certificate-based key management," RFC 1422, Network Working Group, IETF PEM, February 1993

[35] "The network simulator – ns-2," http://www.isi.edu/nsnam/ns/, available 02/21/05.

[36] S. Agarwal, D. Starobinski and A. Trachtenberg, "On the scalability of data synchronization protocols for PDAs and mobile devices," IEEE Network (Special Issue on Scalability in Communication Networks), July/August 2002,Vol. 16, No.4, pp.22-28.

George C Hadjichristofi received his M.S. degree in computer engineering at Virginia Tech in 2001 and has recently completed his Ph.D. degree in computer Engineering at Virginia Tech (June 2005). During his PhD, he conducted research in the areas of IPsec deployment and key management in MANETs that was supported by the Office of Naval Research (ONR) through the Navy Collaborative Information Technology Initiative (NAVCIITI). His research interests include network security, wireless networks, and mobile computing. He is a member of the IEEE, and a member of the Eta Kappa Nu, and the Golden Key National Honor Society.

William J. Adams earned a BS degree in Computer Engineering from Syracuse University in 1986 and a MS degree in Computer Systems Engineering from the University of Arkansas in 1994. Following his graduate studies, he was assigned to the United States Military Academy as an Assistant Professor in Computer Science. In addition to his academic achievements, he has over 18 years experience in network operations and planning in the United States Army Signal Corps, most recently as a project leader and network engineer for the Supreme Headquarters, Allied Powers Europe. He is currently a Ph.D. student at Virginia Tech en-route to rejoin the faculty of the United States Military Academy at West Point, NY. His academic interests include network and mobile security and access control.

Nathaniel J. Davis IV has recently assumed the position of Professor and Head of the Department of Electrical and Computer Engineering at the Air Force Institute of Technology, Wright-Patterson AFB, OH. Prior to his new position, Davis was a Professor of ECE at Virginia Tech for 16 years. Dr. Davis received his Ph.D. from Purdue University in 1985 and his MS and BS degrees from Virginia Tech in 1977 and 1976, respectively. All were in electrical engineering. Dr. Davis's research interests include computer communications networks, computer security, and embedded systems applications. He is a Senior member of the IEEE and a member of the Eta Kappa Nu, Tau Beta Pi, and Sigma Xi.