EURASIP Journal on Wireless Communications and Networking 2005:4, 579–589

Secure, Redundant, and Fully Distributed Key Management Scheme for Mobile Ad Hoc Networks: An Analysis

Deepti Joshi

Department of Electrical and Computer Engineering, Wichita State University, Wichita, KS 67260, USA

Kamesh Namuduri

Department of Electrical and Computer Engineering, Wichita State University, Wichita, KS 67260, USA
Email: kamesh.namuduri@wichita.edu

Ravi Pendse

Department of Electrical and Computer Engineering, Wichita State University, Wichita, KS 67260, USA
Email: ravi.pendse@wichita.edu

Received 21 June 2004; Revised 12 May 2005; Recommended for Publication by Athina Petropulu

Security poses a major challenge in ad hoc networks today due to the lack of fixed or organizational infrastructure. This paper proposes a modification to the existing "fully distributed certificate authority" scheme for ad hoc networks. In the proposed modification, redundancy is introduced by allocating more than one share to each node in order to increase the probability of creating the certificate for a node in a highly mobile network. A probabilistic analysis is carried out to analyze the trade-offs between the ease of certificate creation and the security provided by the proposed scheme. The analysis carried out from the intruder's perspective suggests that in the worst-case scenario, the intruder is just "one node" away from a legitimate node in compromising the certificate. The analysis also outlines the parameter selection criteria for a legitimate node to maintain a margin of advantage over an intruder in creating the certificate.

Keywords and phrases: key management schemes, security, sensor networks.

1. INTRODUCTION

A network can have mainly three types of infrastructure [1]: routing infrastructure consisting of routers and stable communication links; server infrastructure consisting of on-line servers such as dynamic host configuration protocol (DHCP) server, domain name system (DNS), and certificate authority (CA) server, in order to provide services to the network; administrative infrastructure consisting of servers supporting the registration of users, issuing of certificates, and handling of other network configuration tasks.

Ad hoc networks are characterized as infrastructure-less networks. They are emerging to be "anywhere anytime networks" [2]. The main difference between traditional networks and ad hoc networks is the lack of a central administration. Central administration is responsible for providing security services such as defining the security services, policies for the network and predistribution of keys to all the participants. The nodes in an ad hoc network are assumed to be energy-constrained, mobile, and can support limited security [3]. Physical security is limited because the nodes can be turned off or stolen by intruders. Military tactical networks, personal area networks, sensor networks, and disaster area networks are good examples of practical ad hoc networks.

Ad hoc networks are one of the most researched areas in the present day world. A secure networking system must have one or all of the following characteristics [4]: confidentiality, authentication, integrity, nonrepudiation, and availability. Dynamic topology, limited bandwidth, and hard constraints on energy need to be taken into account when developing a security protocol for ad hoc networks. Network origin, transmission range, node capabilities, and network transiency are other factors that might affect the design of a security protocol.

The traditional mechanisms of providing security cannot be applied to ad hoc networks due to their high computational complexity. The security protocol proposed should have low computational complexity and yet provide a high degree of security.

One of the security protocols proposed for ad hoc networks is based on the certificate authority mechanism. In this mechanism, the certificate authority's private key is first divided into parts. These parts or key shares are then distributed among the nodes in the network (one key share per node). In order to communicate, the nodes have to recreate the key. The certificate authority key can be recreated by combining a minimum number of key shares from the total number of shares. The bottleneck arises when the number of nodes required to recreate the key are not found in the communication range (or vicinity) of the node trying to communicate.

In this paper, a modification to the existing "fully distributed certificate authority scheme" is proposed to overcome this bottleneck. In the modified scheme, a node is allocated more than one key share by incorporating redundancy into the network. If more than one key share is given to each node, then the number of nodes required to recreate the CA key are reduced. Thus, a legitimate node will increase its chances of recreating the CA key by the redundancy added to the key management scheme. This redundancy, however, poses a challenge since the chances of an intruder entering the network and compromising the CA key is increased. Hence, the key management scheme should be designed in such a way that the designer can make a choice between ease of recreating the CA key for a legitimate user and the difficulty of compromising the CA key for an illegitimate user or intruder.

An intruder is defined as a node (or its owner) with knowledge of the key management scheme and is capable of recreating the CA key after obtaining sufficient number of key shares. While the legitimate node is programmed with its own key shares, an intruder starts with no key shares at all. While a legitimate node forms a coalition of neighboring nodes to create the certificate, an intruder captures nodes one at time to do the same task. Consider the worst-case scenario in which the intruder also forms a coalition of the same number of nodes as a legitimate node. In this worst-case scenario, the intruder is just "one node" away from the legitimate node in compromising the CA key. Hence, the design criterion for the key management scheme can be stated as follows: choose the parameters of the key management such that the gap between the probabilities of creating the CA key with "y" neighboring nodes and "$y-1$" neighboring nodes is sufficiently large to minimize the compromise.

The rest of the paper is organized as follows. Section 2 discusses the background and related work in ad hoc network security. Section 3 discusses the mathematical formulations needed for the security protocol. Section 4 describes the proposed security protocol. Section 5 presents a probabilistic analysis of the proposed protocol. Section 6 discusses the results and analysis. Section 7 concludes the paper.

2. SECURITY IN AD HOC NETWORKS: BACKGROUND AND RELATED WORK

Security attacks can be classified into active and passive attacks. Passive attacks can be caused by eavesdropping or sniffing the network traffic. This is the easiest form of attack and can be done easily in many network environments. Active attacks involve obstruction or fabrication of data transmission by an intruder. In the traditional encryption techniques, whenever one party has to send data to the other, the sender encrypts the data using the common key. The receiver then decrypts the data using the same key. This mechanism is called the symmetric key encryption [5]. In case of asymmetric key encryption, every node has a public/private key pair. Public keys are known to everyone in the network. When one node has to communicate with the other node, it encrypts the data with the receiver's public key. When the receiver receives data, it decrypts it using its private key.

The Diffe-Hellman (DH) key exchange algorithm [4] was one of the first public key algorithms proposed in the literature. It provides a way of exchanging keys securely. RSA is a similar kind of algorithm that also helps in secure exchange of keys. Digital certificates employ public key infrastructure to provide authentication and integrity of the information being transferred. A certificate is a statement issued by trusted party saying that it verifies that the public key belongs to the user. In the popular network authentication techniques such as Kerberos [6], standard X.509 [7], and PKIX [8], the communicating parties authenticate each other using a certificate created by a certificate authority (CA). This kind of approach cannot be used in an ad hoc scenario because maintenance of a centralized approach is difficult and may not be feasible. Moreover, this approach is not scalable and the CA servers can be a point of single failure in the network as it can be compromised by a simple DoS attack.

Pretty good privacy (PGP) [9, 10] follows a web-of-trust model, in which we have a trusted third party like a certificate authority (CA) which authenticates the nodes by issuing certificates. All the nodes trust this CA and its issued certificates. The CA signs every certificate with its private key. The public key for a node is published by a CA in a user certificate. Any two nodes that want to communicate encrypt the information with the recipient nodes' public key. The recipient node then decrypts the information by using its own private key. A certificate authority is responsible for issuing, revoking, renewing, and providing directories of digital certificates. There are two kinds of trusted third parties. An online trusted third party (TTP) will participate not only in establishing the link but also in communication, whereas an offline link participates only in the establishment of the link. Examples of TTP are key distribution center (KDC), key translation center (KTC), and certificate authority (CA).

The disadvantage of using a TTP mechanism is that if the CA is compromised, the intruder can sign certificates using the CA's private key. To overcome this bottleneck, many solutions were proposed in the literature. The secret sharing approach proposes that the CA's private key should be divided and shared among the ad hoc nodes in the network.

Table 1: Variables description.

Symbol	Description
n	Number of nodes
k	Minimum number of shares required to recreate the CA key
q	Number of shares per node
y	Number of neighbors
$f(x)$	Sharing polynomial
$\mathrm{sk}_{\mathrm{CA}}$	Private key of the CA
S	Secret to be shared
S_i	Share of the ith node
$f_{\mathrm{update}}(x)$	Update function
g^{a_i}	Witness for a_i
d_{ij}	Shuffling factor
$\underline{S_p^j}$	Partial share before shuffling
$\overline{S_p^j}$	Partial share after shuffling
Cert	Certificate of the requesting node
cert_i	Partial certificate generated by the node
$P_{\mathrm{legitimate}}(\mathrm{CA})$	Probability of a legitimate node recreating the CA key
$P_{\mathrm{intruder}}(\mathrm{CA})$	Probability of an intruder compromising the CA key

Security function sharing has been an active area of research in the field of cryptography [11, 12, 13, 14, 15, 16, 17, 18, 19]. By distributing the services of the certificate authority (CA), the availability of the services is increased and the probability of having the single point of failure compromised is reduced. Threshold secret sharing is discussed in [20, 21]. The concept of proactive secret sharing discussed in [22] provides robustness to the existing threshold cryptography methods by renewing the shares periodically.

In the next section, the mathematical formulations needed to calculate the probability of recreating the CA key are discussed.

3. DISTRIBUTED KEY MANAGEMENT: MATHEMATICAL FORMULATIONS

In this section, the mathematical formulations needed for the security protocol and its probabilistic analysis are discussed. Table 1 describes the various variables used in this section.

3.1. Secret sharing

This method is based upon Shamir's secret sharing model proposed in [20]. In a (k, n) threshold sharing scheme, n denotes the number of nodes and k denotes the minimum number of shares needed to recreate the CA key. Suppose a secret S is to be shared between n nodes, identified by $\mathrm{id}_i = 1, 2, 3, \ldots, n$. The dealer performs the following steps.

(1) A prime number p is chosen such that $p > \max(S, n)$.
(2) A sharing polynomial $f(x) = a_0 + a_1 x + \cdots + a_{k-1}x^{k-1}$, where $a_0 = \mathrm{sk}_{\mathrm{CA}}$ (private key of the CA).
(3) The shares for each node are calculated by the equation

$$S_i = f(\mathrm{id}_i) \bmod p. \tag{1}$$

(4) The shares are then distributed to the respective nodes.

In order to reconstruct the secret key, Lagrange interpolation technique is used:

$$f(x) = \sum_{i=1}^{k} S_i * l_{\mathrm{id}_i}(x) (\mathrm{mod}\, p), \tag{2}$$

where $l_{\mathrm{id}_i}(x)$ is called the Lagrange coefficient of id_i and is defined as

$$l_{\mathrm{id}_i}(x) = \prod_{j=1,\, j\neq i}^{k} \frac{x - \mathrm{id}_j}{\mathrm{id}_i - \mathrm{id}_j}. \tag{3}$$

The shareholders have no idea about each others' shares. If a node potentially gains knowledge about k shares, it can reconstruct the secret itself.

3.2. Proactive secret sharing

Given sufficiently long time, an intruder can compromise k nodes and reconstruct the secret. It is therefore important that the shares be updated periodically [22]. This is done using proactive secret sharing. The share update can be achieved by adding an update function $f_{\mathrm{update}}(x)$ to the existing sharing polynomial function $f(x)$:

$$\begin{aligned} f(x) &= a_0 + a_1 x + \cdots + a_{k-1}x^{k-1} (\mathrm{mod}\, p), \\ f_{\mathrm{update}}(x) &= b_1 x + b_2 x^2 + \cdots + b_{k-1}x^{k-1} (\mathrm{mod}\, p), \\ f_{\mathrm{new}}(x) &= f(x) + f_{\mathrm{update}}(x) = a_0 + (a_1 + b_1)x \\ &\quad + \cdots + (a_{k-1} + b_{k-1})x^{k-1} (\mathrm{mod}\, p). \end{aligned} \tag{4}$$

The shares are recalculated and distributed to the respective nodes.

3.3. Verifiable secret sharing

If any shareholder provides an invalid share, the reconstructed secret will not be the same as the original secret. This can be avoided using verifiable secret sharing [18]. The following steps are involved in the verifiable secret sharing scheme.

(1) Before the shares are distributed the dealer publishes the witnesses for sharing polynomial $g^{a_0}, g^{a_1}, g^{a_2}, \ldots, g^{a_{k-1}}$.
(2) Each node can check its share by verifying

$$g^{S_i} = g^{a_0} * \left(g^{a_1}\right)^{\mathrm{id}_i} * \cdots * \left(g^{a_{k-1}}\right)^{\mathrm{id}_i^{k-1}}. \tag{5}$$

The underlying trust model used is the TTP model [23]. In this model, we have a trusted entity or a trusted CA. This CA arbitrates the trust by signing certificates. Many of the aforementioned protocols [9, 12, 21] use this model.

In general, a node is trusted if k nodes claim trust in that node. As mentioned before, the services of the certificate authority are distributed to specialized servers in the secret sharing paradigm. These services include registration, initialization, certification, key update, revocation, certificate and revocation notice distribution.

3.4. Partially distributed certificate authority

Zhou and Haas [21] proposed a threshold cryptography scheme in which the certificate authority services would be divided among a certain number of specialized servers and the CA key would be divided among all the nodes. Each node is capable of generating a partial certificate. In order to recreate the CA key, any node must have a minimum of k partial certificates. This mechanism assumes that we have at least some nodes with high computational power (to act like the servers).

Every node and the CA have a public and private key pair. The CA's public key is known to all the nodes and the private key is shared among the nodes according to Shamir's secret sharing scheme [20]. The bottleneck in this case is that we needed to have special servers with high energy. If these nodes were to fail, the security paradigm fails. The CA services provided in this scheme are similar to those of the fully distributed scheme which will be discussed in the latter part of this section.

3.5. Fully distributed certificate authority

Partially distributed certificate authority scheme, discussed in the previous section requires the use of specialized high-energy nodes. This assumption is not always valid in an ad hoc network and hence becomes a bottleneck. To overcome this bottleneck, Luo and Lu [2] proposed a fully distributed CA solution. It uses a (k, n) threshold scheme in order to distribute an RSA certificate-signing key to all the nodes in the network. If there are n nodes in a network, the CA private key is divided into n shares. A minimum of k shares is required to recreate the CA key. This eliminates the necessity of having specialized high-energy nodes. It also uses proactive secret sharing mechanisms to protect against the compromise of the CA's signing key. When an intruder enters the network and compromises one node, it becomes as good as a valid node. To overcome this problem, an intrusion detection system is required to be present in the network. This intrusion system identifies the misbehaving/compromised nodes and removes them from the network.

The services provided by the CA are share initialization, share update, certificate issuing, certificate renewal, and certificate revocation. The services provided by the CA are summarized in the remainder of this section.

3.5.1. Share initialization

In this solution the services of the CA are distributed to all the nodes of the network instead of special servers as in partially distributed CA. The dealer first initializes k nodes and then these k nodes initialize the rest of the network. The certificate services include certificate renewal and certificate revocation. The system maintenance includes the process of addition of new nodes and providing them with a new certificate authority shares. The following are the steps involved in the share initialization stage.

(1) The dealer generates a sharing polynomial $f(x) = a_0 + a_1 x + \cdots + a_{k-1}x^{k-1}$, where $a_0 = \mathrm{sk}_{\mathrm{CA}}$ (private key of the CA).
(2) Every node is supplied with its polynomial share (S_i) $S_i = f(\mathrm{id}_i) \bmod p$, where id_i is the unique node identifier.
(3) The dealer publishes k public witnesses for the coefficients of the sharing polynomial. It then destroys the polynomial and quits.
(4) Each node then verifies its share by checking

$$g^{S_i} = g^{a_0} * \left(g^{a_1}\right)^{\mathrm{id}_i} * \cdots * \left(g^{a_{k-1}}\right)^{\mathrm{id}_i^{k-1}}. \tag{6}$$

Whenever a new node joins a network, it needs to find a coalition of k nodes in order to create its own key share. This is because of the absence of the dealer; the new node can form a key share by combining the subshares, which it gets from the coalition nodes.

Consider a node p joining the network. A node i which is already initialized can generate its subshare using the following equation:

$$S_{p,i} = S_i * l_{\mathrm{id}_i}(\mathrm{id}_p). \tag{7}$$

The node then combines all the partial subshares to create its own share as follows:

$$S_{p,i} = \sum_{i=1}^{k} S_{p,i} = \sum_{i=1}^{k} S_i * l_{\mathrm{id}_i}(\mathrm{id}_p) = f(\mathrm{id}_p) \bmod N. \tag{8}$$

The joining node should only get to know the final share because $l_{\mathrm{id}_i}(\mathrm{id}_p)$ is a publicly known value. Any other details would allow the new node to recreate the key shares belonging to the k coalition nodes. To overcome this problem, the nodes rearrange the generated partial shares accordingly so that only the value of the shares change but not the secret shared. The following are the steps involved in the process of share initialization for a joining node p.

(1) The joining node p locates a coalition of k nodes $B = (\mathrm{id}_1, \ldots, \mathrm{id}_k)$ and broadcasts an initialization request.
(2) Every node in the coalition verifies the certificate cert_p, of the joining node p and checks that it has not been revoked.
(3) Each pair of nodes (i, j) in the coalition agree on a shuffling factor d_{ij}. One node generates the shuffling factor, encrypts it with the public key of the other node, and signs it before sending it to the other node. It also generates and signs a public witness $g^{d_{ij}}$. The witness is needed to detect and identify any misbehaving coalition nodes if they generate an invalid shuffled partial share. All the shuffling factors and their witnesses are sent to the node p.

(4) The node p then distributes the shuffling factors and the witnesses received to all the nodes in the coalition.
(5) Each node in the coalition j now generates a partial share $S_p^j = S_j * l_{\mathrm{id}_j}(\mathrm{id}_p)$ and shuffles it using the shuffling factor. The shuffled partial share is generated as follows:

$$S_p^{-j} = S_p^j + \sum_{i=1, i\neq j}^{k} [\,\mathrm{sign}\,(\,\mathrm{id}_i - \mathrm{id}_j\,)\,] \bmod N,$$
$$\mathrm{sign}(x) = \begin{cases} -1, & x \leq 0, \\ 1, & x > 0. \end{cases} \tag{9}$$

(6) Every node sends its partial share to p.
(7) Node p verifies each share and generates its share.

3.5.2. Share update

Proactive secret sharing is used and the shares are updated periodically in order to make the protocol robust. A polynomial $f_{\mathrm{update}}(x)$ is added to the existing sharing polynomial and a new sharing polynomial $f_{\mathrm{new}}(x)$ is formed. The shares are recalculated and distributed.

3.5.3. Certificate issuing

In a distributed CA system, the certificates are not issued. The certificates initially created, are only maintained. The dealer is responsible for initializing, registering, and certifying new nodes in the network.

3.5.4. Certificate renewal

Whenever a node p has to renew its certificate, it sends a request for renewal to a coalition of k nodes. Each node then checks its CRL to determine whether the old certificate has been revoked. If it has been revoked, then the nodes deny the request. Otherwise they agree to serve the request and a new partial certificate (cert_i) is generated and sent.

3.5.5. Certificate revocation

If a certificate is revoked, the public key interface provides a mechanism to inform users about the revoked certificate. Most common method used is certificate revocation list (CRL). A CRL consists of a list of revoked certificates. Every node maintains a CRL.

If a node discovers that any other neighboring node is misbehaving, it adds that node to its certificate revocation list (CRL) and floods an accusation against the node in the network. The nodes which receive this broadcast check whether the node which broadcasted this CRL is a part of its own CRL. If it is, then this broadcast is ignored, otherwise it is accepted and changes are made to the CRL.

3.6. Issues with fully distributed certificate authority

We have to obtain at least k shares in order to form the CA's signing key. If a node is unable to find $(k-1)$ other nodes, then the key is not formed and hence all the communication comes to a standstill. This is possible in a highly mobile environment.

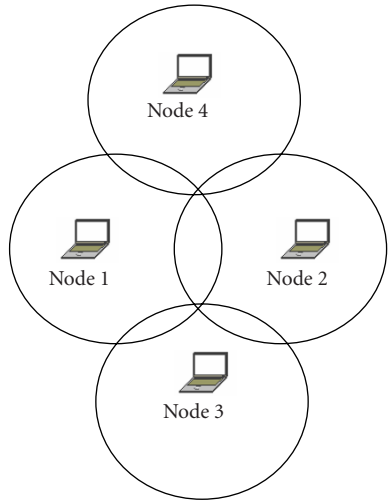

Figure 1: Initial network.

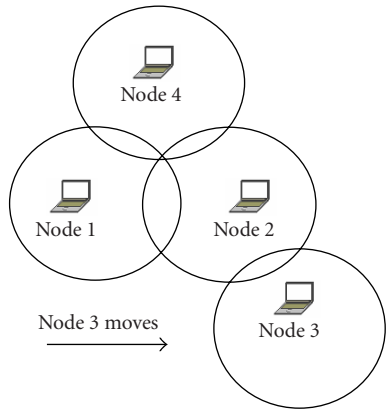

Figure 2: Node 3 moves to another position.

For example, consider a network with four nodes. In the initialization state the CA's private key is divided into 4 shares and suppose a node requires 3 shares to recreate the key. This situation is shown in Figure 1.

Suppose node 3 moves to a location where it has only one neighbor. In this case node 3 cannot recreate the CA key. This situation is shown in Figure 2.

To overcome this bottleneck, the number of shares per node can be increased. The extra shares required can be obtained by introducing redundancy into the network. This proposed solution is discussed and analyzed in detail in the next section.

4. PROPOSED MODEL

In order to overcome the aforementioned bottleneck, the number of key shares per node can be increased using redundancy in key shares. In the traditional fully distributed certificate authority scheme, the number of key shares per node is one. In the modified scheme, the number of key shares per node is increased to q.

The distinct n shares are first calculated using the sharing polynomial where the secret to be shared is the private key of the certificate authority. Using redundancy, these n shares are allocated to all the nodes such that each node gets q shares. Now, the total number of shares including the redundant shares is $(n \cdot q)$. The key distribution can be done in the following manner. First, every node is allocated one distinct share. Then the other $(q-1)$ shares per node are selected from the $(n-1)$ remaining shares such that each node gets q distinct shares.

Consider a network with n nodes. The total number of shares in this scenario, including the redundant shares, is $(n \cdot q)$. The number of distinct shares for a group of y nodes would range from a minimum of y to a maximum of n.

Consider the network discussed earlier, shown in Figure 2. Let the minimum number of shares required in this scenario be 3 ($k = 3$). Suppose that node 3 wants to recreate the CA key. Using the original fully distributed certificate authority scheme, node 3 cannot recreate the CA key because in the traditional scheme the number of key shares per node is one.

In the modified scheme the number of key shares per node is increased to q. Hence, the number of nodes required to recreate the CA key is less than k. In the above example if the number of shares per node is increased to 2 ($q = 2$), node 3 can recreate the CA key.

The increase in the number of shares per node increases the possibility of the node recreating the CA key even if the number of neighbors is less than k. Hence, in the modified scheme, the total number of nodes required to recreate the CA key can be less than $(k-1)$, since any node trying to recreate the CA key can get the k required shares from less than $(k-1)$ nodes. With the increase in the number of shares per node, the number of nodes needed to recreate the CA key is reduced.

Certificate authority services such as share initialization, certificate issuing, certificate renewal, and certificate revocation are provided in a way similar to the original fully distributed CA scheme.

The level of security in case of a single share per node is high, because the intruder has to compromise at least k nodes in order to know the key. This security level decreases when we assign more than one share to the node, as the number of nodes to be compromised decreases. However, this redundancy helps the ad hoc nodes to be more mobile and yet be able to recreate the CA key. The analysis below discusses the trade-off between the degree of security and the ease of recreating the CA key in the proposed scheme.

However, when an intruder enters the network and compromises one node, it becomes as good as a valid node. To overcome this problem, an intrusion detection system is required to be present in the network. This intrusion system identifies the misbehaving/compromised nodes and removes them from the network.

The q shares are chosen at random to increase the security provided by the protocol. If shares distributed are fixed, then the level of security decreases as the node knows the node IDs of the corresponding nodes along with the shares.

The next two sections discuss the analysis of the proposed mechanism and discuss the level of security provided by the modified scheme.

5. EASE OF CERTIFICATE RECREATION VERSUS SECURITY: A PROBABILISTIC ANALYSIS

In this section, we estimate the probability of recreating a certificate when a node is able to communicate with less than k nodes. The security of a network is quantified as the probability of a malicious node compromising the CA key. For the analysis, consider a scenario in which a node has $y(< k)$ neighbors. This coalition might result in at least y and at most n distinct key shares. In order to calculate the total number of ways $(f(y+l))$ in which the CA key can be recreated, consider the number of ways in which the key shares can be distributed among y nodes such that we have $y, y+1, y+2, \ldots, n$ distinct keys. Each node is allocated one distinct share followed by $(q-1)$ additional shares from the remaining $(n-1)$ key shares. The number of ways $(y+l)$ key shares can be gathered from y neighbors is given by

$$f(y+l) = {}^{n}C_{y+l}\,{}^{(y+l)}C_{y}\,(y!)\left({}^{(y+l-1)}C_{q-1}\right)^{y}, \tag{10}$$

where the first term represents the number of ways $(y+l)$ keys can be selected from n keys, the second term represents the number of ways y keys can be selected from $(y+l)$ keys, the third term represents the number of ways these y shares can be allocated to the y nodes, and the fourth term represents the number of ways in which the remaining shares can be allocated to the y nodes. The probability of recreating the CA key given y neighbors is given by

$$p_{\text{legitimate}}(y) = \begin{cases} \dfrac{\sum_{l=k-y}^{n-y} f(y+l)}{\sum_{l=0}^{n-y} f(y+l)} & \text{if } (y \cdot q) \geq n, \\ \dfrac{\sum_{l=k-y}^{y\cdot q-y} f(y+l)}{\sum_{l=0}^{y\cdot q-y} f(y+l)} & \text{if } (y \cdot q) < n, \end{cases} \tag{11}$$

where the numerator considers the cases in which at least k shares required to recreate the CA key can be found and the denominator considers all cases including the cases where the required k key shares cannot be found. The above equation also takes into account the maximum number of distinct key shares a legitimate node can gather from a coalition of y nodes, which is either $(y \cdot q)$ or n depending on whether $(y \cdot q)$ is greater than or equal to n or less than n.

5.1. Intruder's perspective

This section presents an intruder's perspective in order to quantify the level of security offered by the proposed key management scheme.

If an intruder wants to enter the network using an invalid certificate, his requests will not be served by the nodes. On the other hand, a node could enter the network with a valid certificate and then start compromising other nodes.

At some point, the validity of the certificate will expire. From this point onwards, the intruder will not be able communicate with other nodes. This is a naïve intrusion scenario, in which the intruder gets the certificate only once and gets to compromise the information flowing through the network until the certificate is revoked.

A more advanced intrusion can take place as follows. The intruder starts by capturing one node compromising q number of shares. Then the intruder continues to compromise other nodes one at a time until enough key shares needed to recreate the CA key are obtained. This type of intrusion can be compared to "*spying*." The *spying node* pretends to be a legitimate node and continues its covert operations until it gets caught (through intrusion detection techniques). The spying node has as much knowledge and capability as a legitimate node. However, it needs to work towards getting the required neighboring nodes and key shares to recreate the CA key.

From this perspective, it can be observed that an intruder is one node away from the legitimate node in compromising the CA key. Assume that a legitimate node requires a coalition of y number of nodes including itself, to create a valid CA key. An intruder, being as knowledgeable as the legitimate node, also requires the same number of nodes to form the CA key. However, an intruder starts with zero key shares, whereas a legitimate node starts with its own share (q) of keys given at the time of deployment. Thus, the intruder is just one node away from the legitimate node in compromising the certificate in the worst-case scenario. In this scenario, an intruding node forms a coalition of "y" nodes including itself, and the chances of recreating the CA key for an intruder can be represented as follows:

$$p_{\text{intruder}}(y) = p_{\text{legitimate}}(y-1). \tag{12}$$

The probability of the CA's private key being compromised quantifies the intruders knowledge of the CA key. In other words, $p_{\text{intruder}}(y)$ is an estimate of the intruder's ability to compromise the network after forming a coalition of y nodes including itself.

This analysis leads to an important observation: in order to protect the network, the difference between $p_{\text{legitimate}}(y)$ and $p_{\text{intruder}}(y)$ should be maximized. Since $p_{\text{intruder}}(y) = p_{\text{legitimate}}(y-1)$ in the worst-case scenario, we have the following proposition.

Proposition 1. *In order to reduce the chances of compromise, the CA key management scheme should be designed to maximize the difference between the probability of creating the CA key with y nodes and the probability of creating the CA key with $(y-1)$ nodes. In other words, a legitimate node has a margin of advantage over an intruder when the parameters of the key management scheme (k, q, n) are selected in the region where $(p_{\text{legitimate}}(y) - p_{\text{legitimate}}(y-1))$ is large.*

6. RESULTS AND ANALYSIS

In this section, the theoretical results obtained in the previous section are further analyzed. This analysis aids a network designer to choose appropriate parameters for implementing the proposed key management scheme. The analysis is carried out in two parts. The first part focuses on the ease of certificate creation for a legitimate node due to the added redundancy in the key management scheme. The second part of the analysis considers intruder's perspective in conjunction with that of a legitimate node in order to provide an insight into the selection of the parameters (k, q, n) for a secure design of the key management scheme.

6.1. Ease of certificate key recreation for a legitimate node

Figure 3 shows the probability of recreating the CA key as a function of the total number of nodes (n) in the network. Results are plotted for two different scenarios. In Figure 3a, the values of y, q, and k are fixed at 5, 3, and 10, respectively, and in Figure 3b, the values of y, q, and k are fixed at 7, 4, and 20, respectively.

As the total number of nodes in a network increases, the number of distinct shares allocated to the nodes increases. This increases the probability of gathering the required k shares from among the one-hop neighbors. Hence, the probability of the CA key being recreated increases with the increase in the total number of nodes in the network.

Figure 4 shows the probability of recreating the CA key as a function of the number of neighboring nodes for a given node in the network. For the first scenario, the values of n, q, and k are fixed at 20, 3, and 10, respectively, and for the second scenario, the values of n, q, and k are fixed at 40, 4, and 20, respectively.

As the number of neighbors for a given node increases, the possibility of finding k distinct key shares increases. Hence, the ease of recreating the certificate also increases.

Figure 5 shows the probability of recreating the CA key as a function of the number of shares per node in the network. For the first scenario, the values of n, y, and k are fixed at 20, 5, and 10, respectively, and for the second scenario, the values of n, y, and k are fixed at 40, 7, and 20, respectively.

As the number of shares per node increases, the possibility of finding k distinct shares also increases. Hence, the probability of recreating the CA key increases.

Figure 6 shows the probability of recreating the CA key as a function of the minimum number of shares required to recreate the CA key. For the first scenario the values of n, y, and q are fixed at 20, 5, and 3, respectively, and for the second scenario the values of n, y, and q are fixed at 40, 7, and 4, respectively.

As the number of minimum shares required to recreate the CA key increases, the security of the network as a whole increases but the ease of recreating the CA key for a given node decreases. The value of k depends on the desired level of security. Higher values of k result in high degree of security at the expense of reduced chances of creating the CA key.

6.2. Intruder's perspective

In this section, we investigate the security of the proposed key management scheme from an intruder's perspective. The proposed redundancy in the key management scheme increases the ease of creating the CA key for a legitimate node

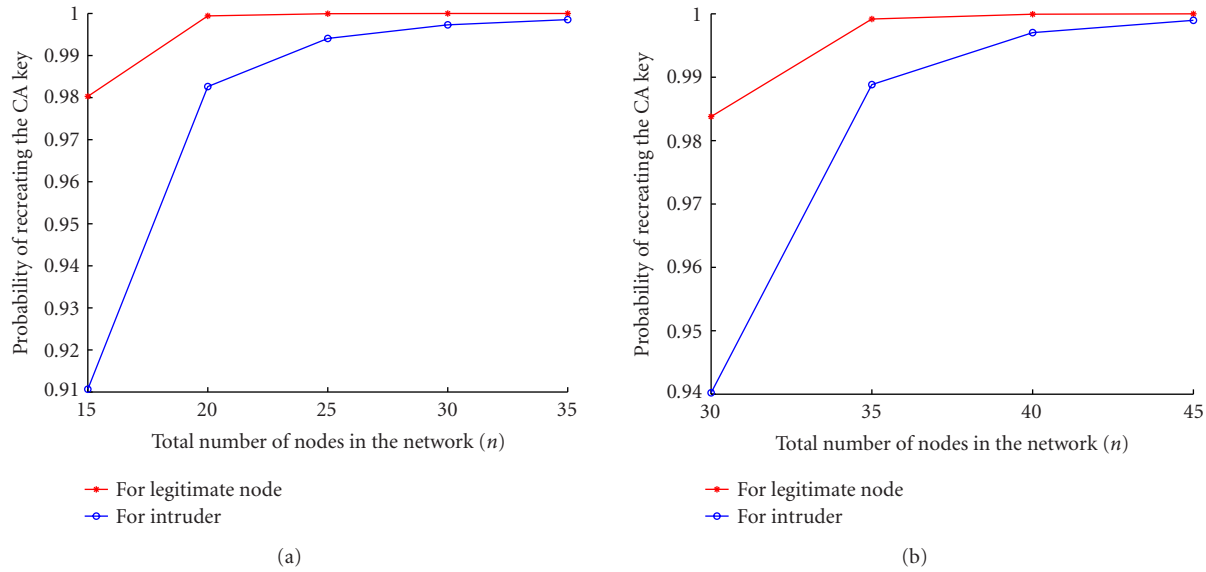

Figure 3: Number of nodes versus probability of recreating the CA key: (a) $y = 5$, $k = 10$, $q = 3$ and (b) $y = 7$, $k = 20$, $q = 4$.

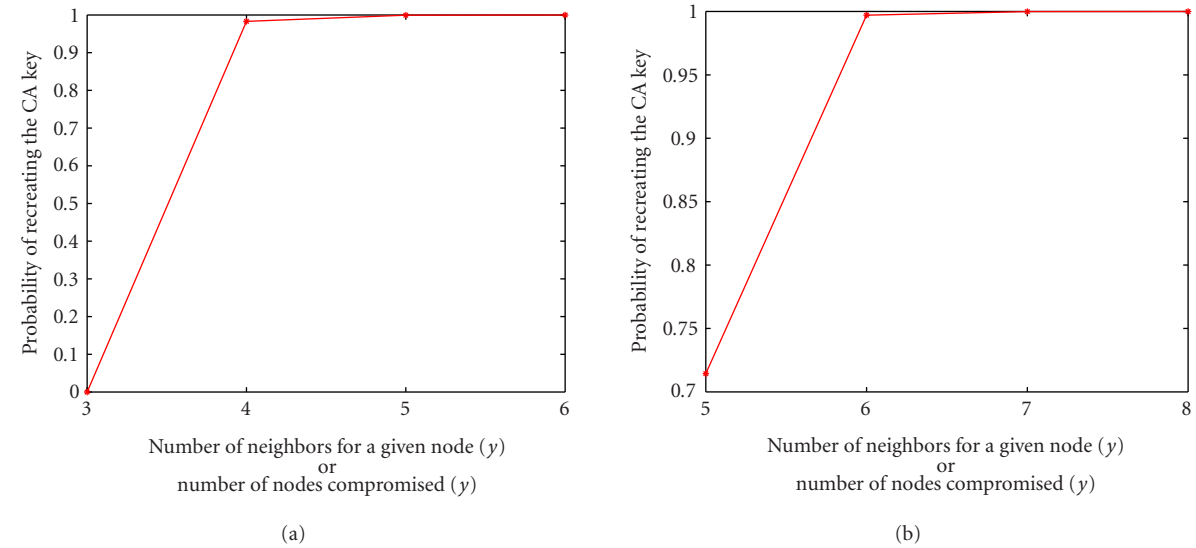

Figure 4: Number of neighbors versus probability of recreating the CA key: (a) $n = 20$, $k = 10$, $q = 3$ and (b) $n = 40$, $k = 20$, $q = 4$.

at the expense of reduced security level. The intruder's perspective is expected to provide the network designer with the trade-offs involved in designing the key management scheme.

Four different scenarios are analyzed by varying each of the parameters n, k, q, and y, while keeping the remaining three parameters fixed. In each scenario, the probability of recreating the CA key is compared with the probability of an intruder compromising the CA key. The plots clearly indicate that the appropriate values for the design parameters are in the regions in which a legitimate node has a significant margin (in terms of probability of recreating the key) over the intruder.

Figure 3 shows the probability of a legitimate node recreating the CA key and the probability of an intruder compromising the CA key as a function of the total number of nodes in the network. These plots clearly indicate that the margin of advantage for a legitimate node over the intruder diminishes as n is increased.

At first look, the graphs suggest that the margin of advantage for a legitimate node is not really significant. However, this observation should be interpreted in the worst-case situation, in which the intruder is able to behave exactly like a legitimate node and succeeds in capturing several neighboring nodes.

Figure 4 plots the probability of compromising the CA key as a function of the number of nodes captured. In Figure 4a, n, q, and k are set to 20, 3, and 10, respectively, and in Figure 4b, n, q, and k are set to 40, 4, and 20, respectively. As the number of nodes compromised increases, the fraction

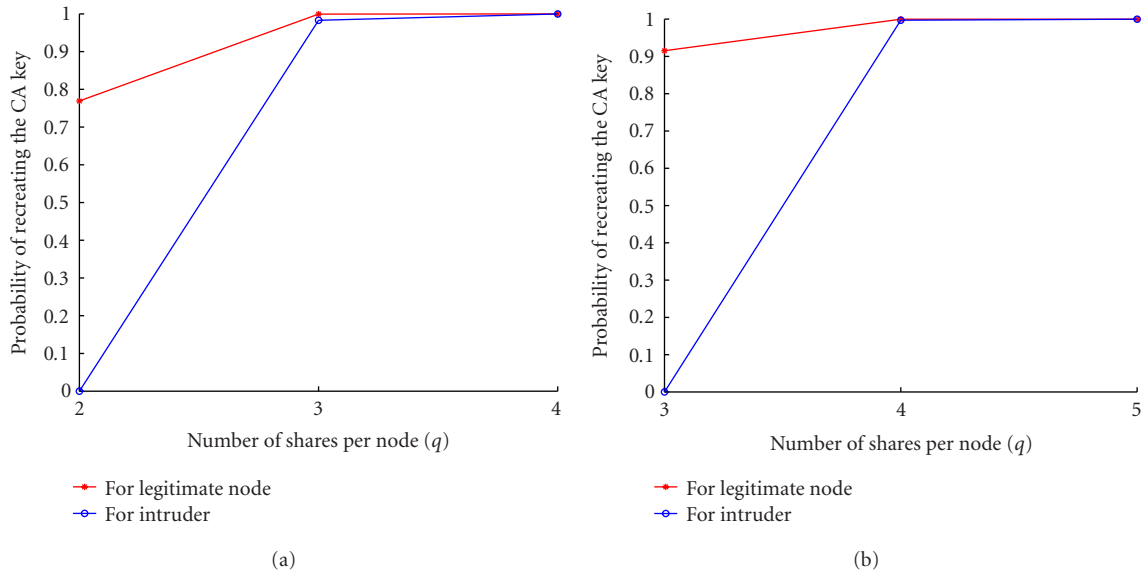

FIGURE 5: Number of key shares per node versus probability of recreating the CA key: (a) $y = 5$, $k = 10$, $n = 20$ and (b) $n = 40$, $k = 20$, $y = 7$.

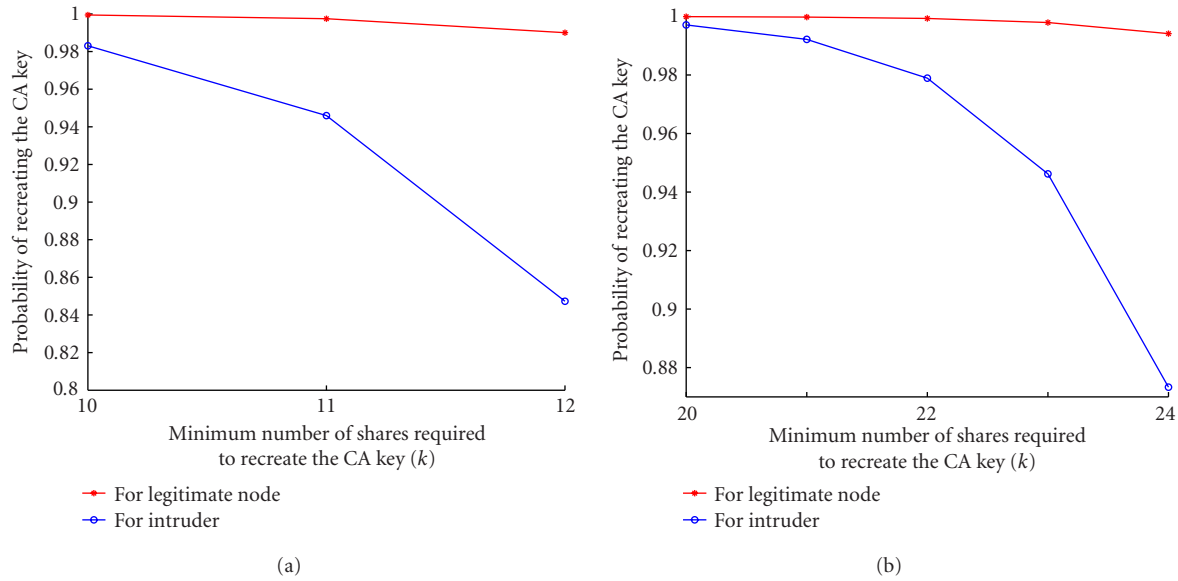

FIGURE 6: Minimum number of key shares required to recreate the CA key versus probability of recreating the CA key: (a) $n = 20$, $q = 3$, $y = 5$ and (b) $n = 40$, $q = 4$, $y = 7$.

of the distinct shares compromised increases and hence the probability of the CA key being compromised increases at a very fast pace. The plots point out that the CA key is practically compromised if 5 out of 20 nodes (with $k = 10$ and $q = 3$) or 7 out of 40 nodes (with $k = 20$, and $q = 4$) are captured by the intruder.

Figure 5 shows the probability of a legitimate node recreating the CA key and the probability of an intruder compromising the CA key as a function of the number of shares (q) per node. The plots suggest that when q is small, a legitimate node has significant margin of advantage over the intruder. As the number of shares per node increases, the number of shares compromised when y nodes are compromised increases. This leads to an increase in the probability of compromising the CA key. In Figure 5a the values of n, y, and k are fixed at 20, 5, and 10, respectively, and in Figure 5b the values of n, y, and k are fixed at 40, 7, and 20, respectively.

Figure 6 shows the probability of a legitimate node recreating the CA key and the probability of an intruder compromising the CA key as a function of the minimum number of

key shares required to recreate the CA key. The plots suggest that large values of k provide significant advantage to the legitimate node over the intruder.

In Figure 6a the values of n, y, and q are fixed at 20, 5, and 3, respectively, and in Figure 6b the values of n, y, and q are fixed at 40, 7, and 4, respectively. As the minimum number of shares required to recreate the CA key increases, the number of shares which are to be compromised increases and hence the probability of compromising the CA key decreases.

7. CONCLUSIONS

In this paper, a modification to the existing fully distributed certificate authority scheme is proposed to make it suitable for a mobile ad hoc network in which forming a coalition of large number of nodes is often difficult. The concept of redundancy in key shares is introduced to increase the probability of recreating the CA key. With redundancy, the level of security provided by the network is less than that of the original scheme. However, the nodes in the ad hoc network can be more mobile than in the original scheme. The ease of certificate recreation and the level of security provided by the modified scheme are analyzed to provide the choices and trade-offs for a network designer.

ACKNOWLEDGMENTS

This research work was carried out under the NSF DUE Grant 0313827. The authors would also like to thank Ms. Aparna Nagesh for performing the simulations required for the plots.

REFERENCES

[1] K. Fokine, "Key management in ad hoc networks," M.S. Thesis, Linköping University, Linköping, Sweden, 2002.
[2] H. Luo and S. Lu, "Ubiquitous and robust authentication services for ad hoc wireless networks," Tech. Rep. TR-200030, Department of Computer Science, University of California, Los Angeles, Los Angeles, Calif, USA, 2000.
[3] A. Khalili, J. Katz, and W. A. Arbaugh, "Toward secure key distribution in truly ad hoc networks," in *Symposium on Applications and the Internet Workshops (SAINT '03 Workshop)*, 2003.
[4] W. Stallings, *Cryptography and network security: principles and practices*, Prentice Hall, Englewood Cliffs, NJ, USA, 2003.
[5] C. P. Pfleeger and S. L. Pfleeger, *Security in Computing*, Prentice Hall, Englewood Cliffs, NJ, USA, 2003.
[6] J. Kohl and B. Neuman, "The Kerberos network authentication service (version 5)," RFC-1510, 1993.
[7] A. Aresenault and S. Turner, "Internet X.509 public key infrastructure," draft-ietf-pkixroadmap-06.txt, 2000.
[8] R. Housley, W. Ford, W. Polk, and D. Solo, "Internet X.509 public key infrastructure certificate and CRL profile," RFC 2459, 1999.
[9] S. Garfinkel, *PGP: Pretty Good Privacy*, O'Reilly and Associates, California, USA, 1995.
[10] A. Abdul-Rahman, "The PGP Trust Model," *EDI-Forum: The Journal of Electronic Commerce*, vol. 10, no. 3, pp. 27–31, 1997.
[11] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," in *Proc. 28th IEEE Annual Symposium on the Foundations of Computer Science (FOCS '87)*, pp. 427–437, Los Angeles, Calif, USA, 1987.
[12] Y. Frankel, P. Gemmell, P. Mackenzie, and M. Yung, "Proactive RSA," in *17th Annual International Cryptology Conference (CRYPTO '97)*, Santa Barbara, Calif, USA, August 1997.
[13] T. Wu, M. Malkin, and D. Boneh, "Building intrusion tolerant applications," in *Proc. 8th USENIX Security Symposium (Security '99)*, pp. 79–91, Washington, DC, USA, August 1999.
[14] Y. Frankel, P. Gemmall, P. MacKenzie, and M. Yung, "Optimal-resilience proactive public-key cryptosystems," in *38th IEEE Annual Symposium on Foundations of Computer Science (FOCS '97)*, pp. 384–393, Miami Beach, Fla, USA, October 1997.
[15] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Robust and efficient sharing of RSA functions," *Journal of Cryptology*, vol. 13, no. 2, pp. 273–300, 2000.
[16] R. Canetti, S. Halevi, and A. Herzberg, "Maintaining authenticated communication in the presence of break-ins," *Journal of Cryptology*, vol. 13, no. 1, pp. 61–105, 2000.
[17] Y. Desmedt and Y. Frankel, "Shared generation of authenticators and signatures (Extended Abstract)," in *11th Annual International Cryptology Conference (CRYPTO '91)*, pp. 457–469, Santa Barbara, Calif, USA, 1991.
[18] Y. Frankel and Y. G. Desmedt, "Parallel reliable threshold multi-signature," Tech. Rep. TR-92-04-02, Department of EECS, University of Wisconsin-Milwaukee, Milwaukee, Wis, USA, 1992.
[19] L. Gong, "Increasing availability and security of an authentication service," *IEEE J. Select. Areas Commun.*, vol. 11, no. 6, pp. 657–662, 1993.
[20] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.
[21] L. Zhou and Z. J. Haas, "Securing ad hoc networks," *IEEE Networks*, vol. 13, no. 6, pp. 24–30, 1999.
[22] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, "Proactive secret sharing or: How to cope with perpetual leakage," in *Proc. 15th Annual International Cryptology Conference (CRYPTO '95)*, vol. 963 of *Lecture Notes In Computer Science*, pp. 339–352, Santa Barbara, Calif, USA, August 1995.
[23] R. Perlman, "An overview of PKI trust models," *IEEE Network*, vol. 13, no. 6, pp. 38–43, 1999.
[24] J. Song and L. E. Miller, "Empirical analysis of the mobility factor for the random waypoint model," in *Proc. OPNETWORK*, Washington, DC, USA, August 2002.

Deepti Joshi received the Bachelor's degree in computer science and engineering in 2002, graduating with distinction from Jawaharlal Nehru Technological University, Hyderabad, India. She received her Master's degree in electrical and computer engineering from Wichita State University, Wichita, Kansas, in 2004. Her research interests include cryptography, network security, voice over IP, and ad hoc networks.

Kamesh Namuduri received his B.E. degree in electronics and communication engineering from Osmania University, India, in 1984, M. Tech. degree in computer science from University of Hyderabad in 1986, and Ph.D. degree in computer science and engineering from the University of South Florida in 1992. He has worked in C-DoT, a telecommunication firm in India

from 1984 to 1986. Currently, he is with the Electrical and Computer Engineering Department, Wichita State University, Wichita, Kansas, as an Assistant Professor. His areas of research interest include information security, image/video processing and communications, and ad hoc sensor networks. He is a Senior Member of IEEE.

Ravi Pendse is an Associate Vice President for Academic Affairs and Research, Wichita State Cisco Fellow, and Director of the Advanced Networking Research Center at Wichita State University, Wichita, Kansas. He has received his B.S. degree in electronics and communication engineering from Osmania University, India, in 1982, M.S. degree in electrical engineering from Wichita State University, Wichita, Kansas, in 1985, and Ph.D. degree in electrical engineering from Wichita State University, Wichita, Kansas, in 1994. He is a Senior Member of IEEE. His research interests include ad hoc networks, voice over IP, and aviation security.