

Authentication Based on Multilayer Clustering in Ad Hoc Networks

Keun-Ho Lee

*Department of Computer Science & Engineering, Korea University, 1, 5-Ga, Anam-dong, Sungbuk-ku, Seoul 136-701, Korea
Email: root1004@korea.ac.kr*

Sang-Bum Han

*Department of Computer Science & Engineering, Korea University, 1, 5-Ga, Anam-dong, Sungbuk-ku, Seoul 136-701, Korea
Email: topflite@korea.ac.kr*

Heyi-Sook Suh

*Department of Computer Science & Engineering, Korea University, 77-6 Sejong-ro, Jongro-gu, Seoul 110-760, Korea
Email: suh@moe.go.kr*

SangKeun Lee

*Department of Computer Science & Engineering, Korea University, 1, 5-Ga, Anam-dong, Sungbuk-ku, Seoul 136-701, Korea
Email: yalphy@korea.ac.kr*

Chong-Sun Hwang

*Department of Computer Science & Engineering, Korea University, 1, 5-Ga, Anam-dong, Sungbuk-ku, Seoul 136-701, Korea
Email: hwang@disys.korea.ac.kr*

Received 30 June 2004; Revised 2 August 2005

In this paper, we describe a secure cluster-routing protocol based on a multilayer scheme in ad hoc networks. This work provides scalable, threshold authentication scheme in ad hoc networks. We present detailed security threats against ad hoc routing protocols, specifically examining cluster-based routing. Our proposed protocol, called “authentication based on multilayer clustering for ad hoc networks” (AMCAN), designs an end-to-end authentication protocol that relies on mutual trust between nodes in other clusters. The AMCAN strategy takes advantage of a multilayer architecture that is designed for an authentication protocol in a cluster head (CH) using a new concept of control cluster head (CCH) scheme. We propose an authentication protocol that uses certificates containing an asymmetric key and a multilayer architecture so that the CCH is achieved using the threshold scheme, thereby reducing the computational overhead and successfully defeating all identified attacks. We also use a more extensive area, such as a CCH, using an identification protocol to build a highly secure, highly available authentication service, which forms the core of our security framework.

Keywords and phrases: authentication, clustering, cluster head, ad hoc network, multilayer.

1. INTRODUCTION

Mobile ad hoc networks consist of devices that are autonomously self-organized into networks. In ad hoc networks, the devices themselves are the network, and this allows seamless communication, at low cost, with a self-organizing capability, which makes mobile ad hoc networks completely different from any other networking solution.

Mobile ad hoc networking is one of the most innovative and challenging areas of wireless networking. Ad hoc networks are a key step in the evolution of wireless networks. An ad hoc network is a collection of wireless mobile hosts forming a temporary network without the aid of any established infrastructure or centralized administration. Securing an ad hoc routing protocol presents challenges because each user brings their own mobile unit to the network, without the centralized policy or control of a traditional network. Many ad hoc routing protocols have been proposed, and clustering-based protocols include “cluster-based routing protocol” (CBRP) [1], “adaptive routing using clustered hierarchies” (ARCH) [2],

This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

the “distributed clustering algorithm” (DCA) [3], and “distributed mobility-adaptive clustering” (DMAC) [3]. Mobile ad hoc networks’ security issues have become a central concern and are increasingly important. Ad hoc networks cannot be used in practice if they are not secure, because ad hoc networks are subject to various attacks. Wireless communication links can be intercepted without noticeable effort, and communication protocols in all layers are vulnerable to specific attacks [4]. Studies of secure cluster routing based on multiple layers in ad hoc networks have been carried out using “authenticated routing for ad hoc networks” (ARAN) [5] and in [4, 6].

In this paper, we demonstrate possible ways to exploit ad hoc routing protocols, define various security environments, and offer a secure solution with “authentication based on multilayer clustering for ad hoc networks” (AMCAN). We detail the ways to exploit protocols that are under consideration by [1, 2, 3, 4, 5, 6].

Our proposed protocol detects and protects against malicious actions by multilayer parties in one particular ad hoc environment. We propose an authentication protocol that uses certificates containing a Diffie-Hellman key agreement and a multilayer architecture so that CCH is achieved using the threshold scheme, so that the number of essential encryptions reduces the computational overhead and successfully defeats all identified attacks.

Our evaluations show that AMCAN has minimal performance costs in terms of processing and networking overhead for the increased security that it offers. While this basic idea has been proposed before in [2, 3, 5], we are the first to apply it to a clustered network. Our algorithm addresses issues of authentication and multilayer security architecture and helps to adapt the complexity to the scalability of mobile end systems. Moreover, an extensive evaluation involves the reduction of CH traffic using CCH.

In this paper, we first overview cluster routing protocols in ad hoc networks, and briefly overview security goals, common techniques for authentication, and threshold cryptosystems, as well as related work for securing ad hoc networks in Section 2. Section 3 describes our security concept in detail as a CCH construction algorithm and presents authentication based on multilayer clustering for ad hoc networks (AMCAN). An important contribution of our work is the evaluation of the CCH construction and security architecture in Section 4. Those measurements are based on different authentication models, which are presented in this section, and we also show the results of security and network performance analyses of AMCAN. Finally, Section 5 concludes the paper and considers further research.

2. RELATED WORK

There are numerous proposals for clustering and multilayer routing schemes. This section presents two aspects of AMCAN, including those that are most closely related to the cluster organization and security requirements in ad hoc networks.

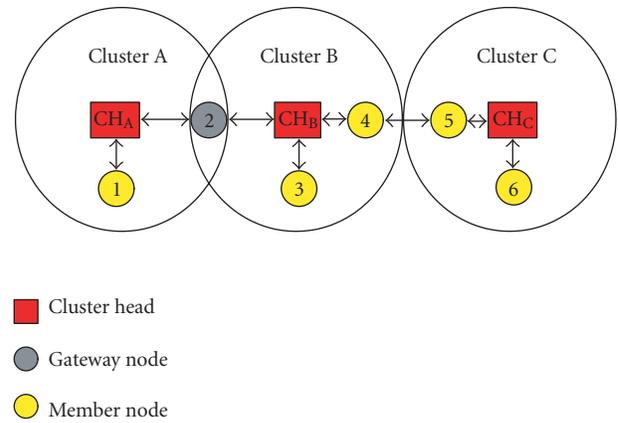


FIGURE 1: Clustering-based architecture.

2.1. Clustering in ad hoc networks

A comprehensive overview of different clustering strategies is presented in [8]. In this section, we present several of the cluster-based control structures and associated control algorithms that have been proposed for use in large dynamic networks. A cluster-based control structure promotes more efficient use of resources in controlling large dynamic networks. With cluster-based control, the physical network is transformed into a virtual network of interconnected node clusters. Each cluster has one or more controllers acting on its behalf to make control decisions for cluster members and, in some cases, to construct and distribute representations of cluster state for use outside the cluster [2, 8].

CBRP [1] is a routing protocol designed for use in mobile ad hoc networks. The protocol divides the nodes of the ad hoc network into a number of overlapping or disjoint two-hop-diameter clusters using a distributed method. The cluster-based architecture was devised to minimize the flooding of route discovery packets. This kind of architecture is most suitable for large networks with several nodes. The entire network is divided into a number of overlapping or disjoint two-hop-diameter clusters, as shown in Figure 1. A cluster head (CH) is elected for each cluster to maintain cluster membership information. A cluster is identified by its CH ID. Intercluster routes are discovered dynamically using the cluster membership information kept by each CH. By clustering nodes into groups, the protocol efficiently minimizes the flooding traffic during route discovery and speeds up this process. A node regards itself as being in a cluster if it has a bidirectional link to the head of the cluster. In the current implementation of CBRP, the node with the lowest node ID is elected as the CH.

All of the nodes broadcast a HELLO message periodically. The HELLO message also contains tables carrying information about the neighboring nodes and adjacent clusters. These HELLO messages are useful for maintaining up-to-date two-hop topology. An in-depth study of cluster-based networks has been published [1].

ARCH builds on the foundations of adaptive routing using clusters (ARC) [2] to create a multilevel hierarchy that is

able to adjust its depth dynamically in response to the changing conditions of the network. ARCH conforms to the maximum hierarchical depths proven to be the theoretical optimum. As such, the protocol lends itself well to hierarchical addressing structures. When used with hierarchical addressing, it should be extremely beneficial for reducing routing table size.

2.2. Security protocol in ad hoc networks

The security requirement, which typically strives for ad hoc networks security goals like authentication, availability, confidentiality, integrity, and the nonrepudiation of communicating entities, is of particular importance as it forms the basis for achieving the other security goals. Encryption of ad hoc networks security is worthless if the communication partners have not verified their identities beforehand. Authentication of entities and messages is realized in different ways using either symmetric or asymmetric cryptographic algorithms. Authentication enables a node to ensure the identity of the peer node that it is in communication with. Without this, an attacker could impersonate a node, thereby gaining unauthorized access to a resource and sensitive information and interfering with the operation of other nodes.

While a symmetric algorithm depends on the existence of a preshared key, authentication using asymmetric cryptography requires a secure mapping of public key infrastructures (PKI). PKIs use digitally signed certificates to verify a key owner's identity. Each user has to prove their identity to a certification authority (CA) and in turn receives a digitally signed certificate proving the ownership of the public key. Distributing the signing key and the functionality of a CA over a number of different nodes by means of secret sharing and threshold cryptography is a possible solution to this problem, as we will study here [4].

Threshold cryptosystem

A threshold cryptosystem is a distributed implementation of a cryptosystem, in which the secret key is a secret that is shared among a group of nodes. These nodes can then decrypt or sign messages by following a distributed protocol. The goal of a threshold scheme is to protect the secret key in a fault-tolerant way. Namely, the key remains secret, and correct decryptions or signatures are always computed, even if the adversary corrupts less than a fixed threshold of the node. Desmedt and Frankel introduced threshold cryptosystems [13]. In particular, they presented a threshold cryptosystem based on the Diffie-Hellman problem. The secret sharing scheme [14] is important for threshold cryptosystems. The idea of secret sharing is to start with a secret, and divide it into pieces called shares, which are distributed amongst users such that the pooled shares of specific subsets of users allow reconstruction of the original secret. We now describe the Shamir $(t \cdot n)$ -threshold secret sharing scheme. Suppose p and q are large primes such that q divides $p - 1$, and g is an element of order 1 in Z . It is assumed that p , q , and g are known publicly. Unless otherwise stated, all arithmetic

TABLE 1: Variables and notation used in ARAN.

K_{A+} : public key of node A.
K_{A-} : private key of node A.
$\{d\}K_{A+}$: encryption of data d with key K_{A+} .
cert_A : certificate belonging to node A.
t : timestamp.
e : certificate expiration time.
N_A : nonce issued by node A.
IP_A : IP address of node A.
RDP: route discovery packet identifier.
REP: REPLY packet identifier.
SPC: shortest path confirmation packet identifier.
RSP: recorded shortest path packet identifier.
ERR: ERRor packet identifier.

will be computed modulo p . The scheme is described in the following protocol. Distribution of trust in our key management service is accomplished using threshold cryptography [16, 17]. An $(n, t + 1)$ -threshold cryptography scheme allows n parties to share the ability to perform a cryptographic operation so that any $t + 1$ parties can perform this operation jointly, whereas it is infeasible for at most t parties to do so, even by collusion.

ARAN protocol

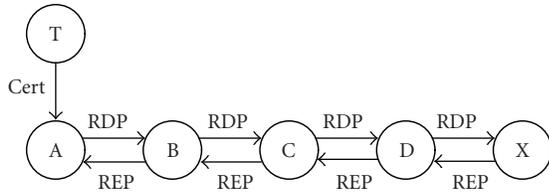
The ARAN protocol can detect and protect against malicious actions by third parties and in the ad hoc environment. ARAN is composed of two distinct stages. The first stage is simple and requires little extra work from peers beyond traditional ad hoc protocols. Nodes that perform the optional second stage increase the security of their route, but incur an additional cost for their ad hoc peers who may not comply. ARAN makes use of cryptographic certificates for the purposes of authentication and nonrepudiation. It consists of a preliminary certification process, a mandatory end-to-end authentication stage, and an optional second stage that provides secure shortest paths. The optional stage is considerably more expensive than providing end-to-end authentication. There are twelve steps necessary to implement ARAN [5].

In [5], vulnerabilities and attacks specific to AODV and DSR protocols are discussed and the two protocols are compared with the ARAN protocol. The ARAN protocol uses a preliminary cryptographic certification process, followed by an end-to-end route authentication process, which ensures secure route establishment. The protocol does not specify any specific key distribution algorithm. On joining the network, each node receives a certificate from the trusted server.

In this partition, we briefly review ARAN protocol. We first describe the notations used throughout this paper in Table 1.

There are totally twelve steps to implement ARAN:

- (1) $T \rightarrow A$: $\text{cert}_A = [\text{IP}_A, K_{A+}, t, e]K_{T-}$,
- (2) $A \rightarrow \text{broadcast}$: $[\text{RDP}, \text{IP}_X, \text{cert}_A, N_A, t]K_{A-}$,



T: Trust server
 Nodes: A, B, C, D, X
 RDP: Route discovery packet identifier
 REP: Reply packet identifier

FIGURE 2: ARAN simple ad hoc network model.

- (3) B → broadcast: $[[RDP, IP_X, cert_A, N_A, t]K_{A-}]K_{B-}, cert_B,$
- (4) C → broadcast: $[[RDP, IP_X, cert_A, N_A, t]K_{A-}]K_{C-}, cert_C,$
- (5) X → D: $[REP, IP_A, cert_X, N_A, t]K_{X-},$
- (6) D → C: $[[REP, IP_A, cert_X, N_A, t]K_{X-}]K_{D-}, cert_C,$
- (7) C → B: $[[REP, IP_A, cert_X, N_A, t]K_{X-}]K_{C-}, cert_C,$
- (8) A → broadcast: $SPC, IP_X, cert_X,$
 $\{[IP_X, cert_A, N_A, t]K_{A-}\}K_{X+},$
- (9) B → broadcast: $IP_X, cert_X, SPC, IP_X, cert_X,$
 $\{\{[IP_X, cert_A, N_A, t]K_{A-}\}K_{X+}\}K_{B-}, cert_B\}K_{X+},$
- (10) X → D: $[RSP, IP_A, cert_X, N_A, route]K_{X-},$
- (11) B → C: $[ERR, IP_A, IP_X, cert_C, N_B, t]K_{B-},$
- (12) T → broadcast: $[revoke, cert_T]K_{T-}.$

Figure 2 shows totally how to process ARAN situation. The idea to use a distributed certification authority based on a shared certification key and threshold cryptography for securing ad hoc networks was presented by [15]. Our approach is based on modification idea of ARAN protocol used by [5, 15], but introduces several new concepts, like a cluster-based network architecture, a process for admitting new participants, and end-to-end access control within the multilayer in the ad hoc networks. The ARAN protocol cannot be a configuration for a large area. If ARAN is large area, ARAN has a lot of overhead.

In this paper, we show how our proposed AMCAN reduces the computational overhead and successfully defeats all identified attacks in a large area.

3. AUTHENTICATION BASED ON MULTILAYER CLUSTERING FOR AD HOC NETWORKS

3.1. Scenario for an experiment in AMCAN

Our proposed scheme is based on the following assumptions. First, mobile nodes in an ad hoc network usually communicate with one another via an error-prone, bandwidth-constrained, insecure wireless channel. The physical layer of the network is vulnerable to denial-of-service (DoS) attacks. As there is no way to protect from DoS attacks, we do not consider physical attacks. Second, the CH knows which nodes are in its own cluster. Therefore, the CH manages the

IDs of cluster members (i.e., when the CH receives a communication request, it can identify members of its own cluster). Third, we consider CH a trusted member. The CH is similar to the server in [15]. Actually, one can trust the section area CH, even if a member node is abnormal. Therefore, we used the CCH (control cluster head) key in a network. Finally, the CCH selected always trusts CH.

The AMCAN protocol requires the use of a trusted certificate server T (CCH) in a cluster. A CH is a certificate server T for authenticated nodes in a cluster. A CCH authenticates the CH for the CCH private key. A CCH is a root-layer certificate trust server. CH certification uses communication between the nodes in a cluster. All the nodes of a network know the public key for the system. Suppose that we have a pair of public and private keys. The CCH and CH use the certificates to keep the Diffie-Hellman key [17] agreement. Our proposed scheme should minimize the communication load in order to extend the overall lifetime of the system. The CH knows who is in its own cluster. We use the key when exchanging certificates to enable secure communication. Figures 3 and 4 illustrate how the service is configured. Moreover, we propose applying the use of ID-based [18, 19] cryptography to abate the overhead effect on exchanging the public key. ID-based public key exchange is weighted more than the RSA algorithm. An ID-based public key is suitable in a mobile ad hoc network.

3.2. Configuration of a multilayer cluster

In this section, we describe an efficient authentication algorithm for the set up and maintenance of cluster organization in the presence of node mobility that modify, thus satisfying the DMAC and the ARCH for the ad hoc clustering routing protocol. We make two main modifications to the original DMAC and ARCH algorithms as shown in Figures 3 and 4. We use the concept of low-maintenance clustering and mobility-aware clustering schemes. Low-maintenance clustering schemes aim at providing stable cluster architecture for upper-layer protocols with little cluster maintenance cost. By limiting reclustering situations or minimizing explicit control messages for clustering, the cluster structure can be maintained well without excessive consumption of network resources for cluster maintenance. Mobility-aware clustering takes the mobility behavior of mobile nodes into consideration. This is because the mobile node's movement is the main cause of changes to the network topology. By grouping mobile nodes with similar speed into the same cluster, the intra-cluster links can be greatly tightened and the cluster structure can be correspondingly stabilized in the face of moving mobile nodes. The cluster topology is initialized and maintained through the periodic transmission of HELLO messages by each node. This makes this algorithm suitable for both clustering set up and maintenance authentication from the CH, which was not available in authentication solutions.

3.2.1. CH selection algorithm

The selection of the CH uses the DMAC algorithm in [3]. The DMAC in our clustering algorithm includes only two

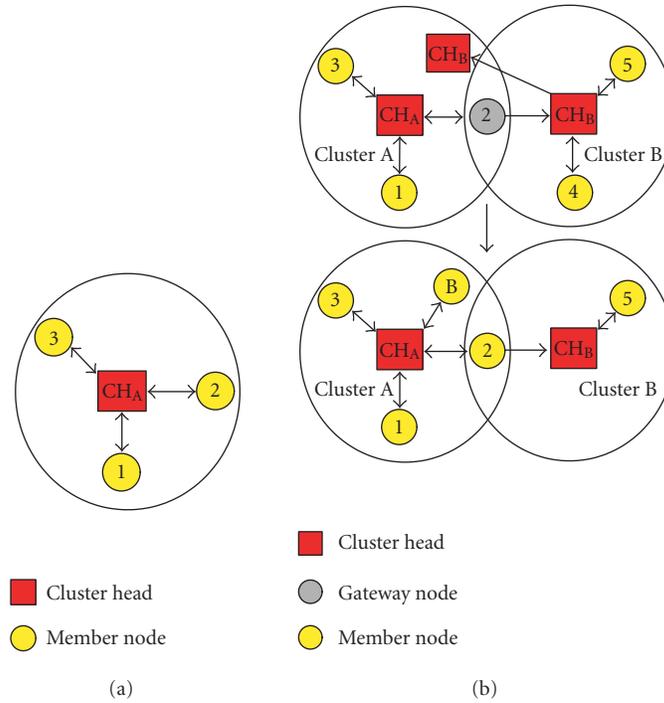


FIGURE 3: The CH selected when joining a CH between cluster A and cluster B (parameter priority lowest ID: $A > B$ in cluster, $1 > 2 > 3 > 4 > 5$ in nodes) (a) Normal cluster. (b) Cluster A moves CH_B into cluster B.

conditions to change the CH. Figure 3 shows the DMAC state in the two conditions. One is when two CHs come within the range of each other, another is when a node becomes disconnected from any other cluster. This is an improvement over existing algorithms, which select the CH every time the cluster membership changes. The DMAC algorithm assumes that a message sent by a node is received correctly within a finite time by all its neighbors. The DMAC also assumes that each node knows its own ID, weight, and role of all its neighbors. In addition, each node knows its power of nodes.

Here, we use the same two types of messages used in the DCA (namely, $Ch(v)$ and $Join(v, u)$) [3]. In the following we use $Cluster(v)$ and $ClusterHead$ to indicate the set of nodes in the cluster whose $ClusterHead$ is v and the $ClusterHead$ of a node's cluster, respectively. v 's Boolean variable $Ch(v)$ is set to true if v has sent a Ch message. Its variables $ClusterHead$, $Ch(\cdot)$, and $Cluster(\cdot)$ are initialized to nil, false, and ϕ , respectively. The following is the description of the two M -procedures as executed at each node v . In DCA algorithm, on receiving a Ch message from a neighbor u , node v checks if it has received from all its neighbors z , such that $w_z > w_u$, a $Join(z, x)$ message. In this case, v will not receive a Ch message from these z , and u is the node with the biggest weight in v 's neighborhood that has sent a Ch message.

At the clustering set up, or when a node v is added to the network, it executes the CH selection procedure (see Algorithm 1) in order to determine its own role. If its neighbors include at least one CH with a greater weight, then v will join it. Otherwise it will be a CH [3].

```

Initialize
begin
  if  $\{z \in (v) : w_z > w_v, \wedge Ch(z)\} \neq \phi$ 
  then begin
     $x := \max_{w_z > w_v} \{z : Ch(z)\}$ ;
    send  $Join(v, x)$ ;
    ClusterHead: =  $x$ 
  end
  else begin
    send  $Ch(v)$ 
     $Ch(v) := true$ ;
    ClusterHead: =  $v$ ;
    Cluster( $v$ ): =  $\{v\}$ 
  end
end;
Repeat—On receiving ClusterHead( $u$ )
begin
  if  $(w_u > w_{ClusterHead})$  then begin
    send  $Join(v, u)$ ;
    ClusterHead: =  $u$ ;
    if  $Ch(v)$  then  $Ch(v) := false$ 
  end
end;
    
```

ALGORITHM 1: CH selection procedure.

At the clustering set up, or when a node v is added to the network, it executes the procedure Initialize in order to determine its own role. If among its neighbors there is at least a cluster head with bigger weight, then v will join it. Notice

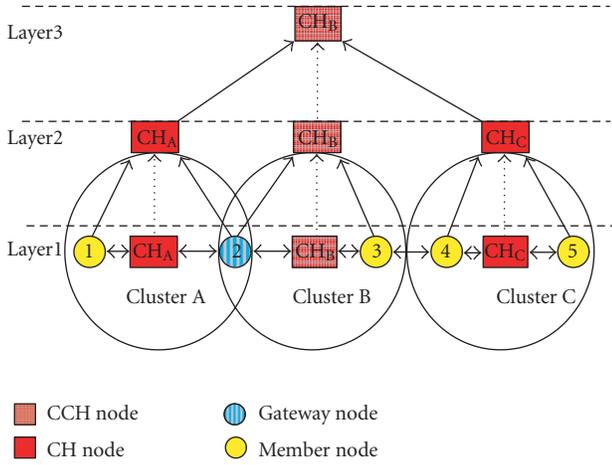


FIGURE 4: CCH selection process with multiple layers (parameter priority lowest ID : B > A > C in cluster, 1 > 2 > 3 > 4 > 5 in nodes).

that a neighbor with a bigger weight that has not decided its role yet will eventually send a message. If this message is a Ch message, then v will affiliate with the new cluster head. When a neighbor u becomes a cluster head, on receiving the corresponding Ch message, node v checks if it has to affiliate with u , and it checks whether w_n is bigger than the weight of v 's cluster head or not. In this case, independently of its current role, v joins u 's cluster [3].

3.2.2. CCH selection algorithm

In this section, our proposed scheme describes the CCH for managing a CH. The CCH selection scheme uses the ARCH algorithm. The CCH has information on all the CHs and takes charge of certificates between CHs. AMCAN uses the ARAN protocol based on the CCH selection algorithm. Figure 4 shows the authenticated architecture for multiple layers using the ARCH algorithm. Source node 1 in cluster A communicates with destination node 5 in cluster C. Before designing the details of our algorithm, we noted that the CH selected the self-stabilizing leader.

On receiving the message $\text{Join}(u, z)$, the behavior of node v depends on whether it is a cluster head or not. In the affirmative, v has to check if either u is joining its cluster ($z = v$: in this case, u is added to $\text{Cluster}(v)$) or if u belonged to its cluster and is now joining another cluster ($z \neq v$: in this case, u is removed from $\text{Cluster}(v)$). If v is not a cluster head, it has to check if u was its cluster head. Only if this is the case, v has to decide its role: it will join the biggest cluster head x in its neighborhood such that $w_x > w_v$ if such a node exists. Otherwise, it will be a CCH (ControlClusterHead). The CCH is v . The CCH roles need slow mobility, lowest of ID, and enough of energy in CHs. u parameter contents included mobility, ID, and energy (see Algorithm 2).

3.3. Design of AMCAN

3.3.1. Protocol scheme

In this section, we describe the detailed operation of AMCAN. AMCAN consists of a preliminary certification process

```

begin
  if Ch( $v$ )
  then if  $z = v$ 
    then  $\text{Cluster}(v) := \text{Cluster}(v) \cup \{u\}$ 
    else if  $u \in \text{Cluster}(v)$ 
      then  $\text{Cluster}(v) := \text{Cluster}(v) \setminus \{u\}$ 
    else if  $\text{ControlClusterHead} = u$  then
      if  $\{z \in (v) : w_z > w_v \wedge \text{Ch}(z)\} \neq \emptyset$ 
      then begin
         $x := \max_{w_z > w_v} \{z : \text{Ch}(z)\}$ ;
        send  $\text{Join}(v, x)$ ;
         $\text{ControlClusterHead} := x$ 
      end
    else begin
      send Ch( $v$ )
       $\text{Ch}(v) := \text{true}$ ;
       $\text{ControlClusterHead} := v$ ;
       $\text{Cluster}(v) := \{v\}$ 
    end
  end
end

```

ALGORITHM 2: CCH selection procedure.

and three mandatory stages: CCH authentication for CHs, a node joins a cluster for the first time, and authentication for end-to-end of session key exchange. So far, we have surveyed several existing solutions for CCH key establishment based on the Diffie-Hellman key exchange. These involve sharing the CCH key communication securely with all members. However, as all members share the same secret key, they cannot communicate with another member using the end-to-end method. Moreover, if anyone has their key stolen, all the members must reestablish the CCH key. The core of the matter is sharing the same secret key with all members. It is impossible for all members to share one secret key because all nodes cannot trust each other in an ad hoc network.

For this reason, we classify all members into two types of trust level: trusted members and untrusted members. Only the CH for trusted members in a cluster can establish a CCH key. Untrusted members authenticate and communicate with other untrusted members using a session key, which is generated by certificate exchange through an authenticated path. AMCAN achieves end-to-end security services and executes partial authentication in all clusters.

3.3.2. System model

There are three different scenarios in which authentication needs to be performed. These are when the CCH authenticates the CH, when a node joins a network for the first time, and when a node from a cluster wishes to communicate for end-to-end key exchange. All the CHs have their own pair of public/private keys and a CCH partition for the stable security of the network. Nodes communicate using a common cluster key within the same cluster. Suppose that all nodes know the public key for the system, and that they have their own public/private key pair. Outside reply attack on a message can be prevented by sending an encrypted timestamp with the message.

TABLE 2: Variables and notation used in AMACN.

CCH: trust server of control CH
CH_A : cluster head in cluster A
ID_X : identity of X
$K_{S,CH}$: secret key shared with S and CH
$Time_1$: current time
S: member node in CH_A
X: member node in CH_B

As large area networks are adaptive to a hierarchical architecture, cluster-based networks are used. A large communication area is divided into several section areas (clusters). Each section CH can participate in only one section and manages the communication units (cluster member nodes) within its section area. Each unit (S) wants to communicate with another unit (X) in another section area, S needs to know whether X wants to communicate with unit S. If so, they will also want to communicate mutually using a secure end-to-end method. Therefore, they exchange a certificate with each other through an authenticated path, so that they can authenticate each other and establish a session key for secure end-to-end communication. Figures 4, 6, and 7 illustrate how the service is configured.

3.3.3. Notation

We use the notation listed in Table 2 to describe the proposed scheme.

3.3.4. CCH authenticated for CH using threshold cryptosystem

In our case, the n CHs of the key management service share the ability to sign certificates. For the service to tolerate t compromised CHs, we use an $(n, t + 1)$ -threshold cryptography scheme and divide the private key, k , of the service into n shares (CH_A, CH_B, CH_C), assigning one share to each CH. We call (CH_A, CH_B, CH_C) sharing of K . Figure 5 illustrates how the service is configured.

Given a service consisting of three CHs, let K/k be the public/private key pair of the service. Using a $(3,2)$ -threshold cryptography scheme, each CH_i gets a share s_i of the private key k .

For a message m , CH_i can generate partial signatures $PS(m, s_i)$ using its share s_i . The correct CH_A and CH_C both generate partial signatures and forward the signatures to a combiner, c . Although CH_B fails to submit a partial signature, c can generate the signature $(m)_k$ of m signed by CH using the private k .

AMCAN consists of a preliminary certification process, a mandatory end-to-end authentication step, and an optional second step that provides threshold cryptosystem. Option step of the AMACN reduced more overhead than end-to-end authentication of ARAN.

CCH requires the use of a trusted certificate server T [5]. All CHs receive a certificate from CCH in Figure 6. A CH

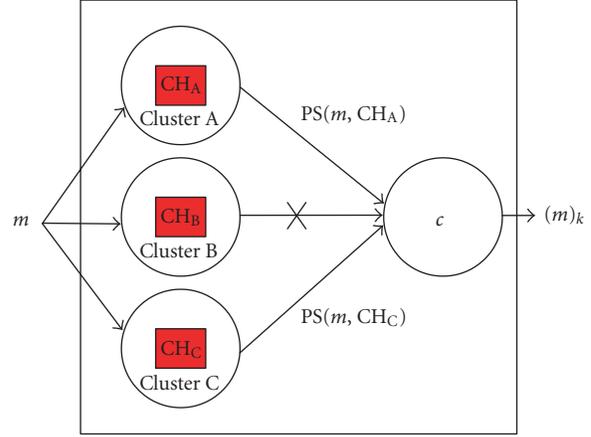


FIGURE 5: Threshold signature.

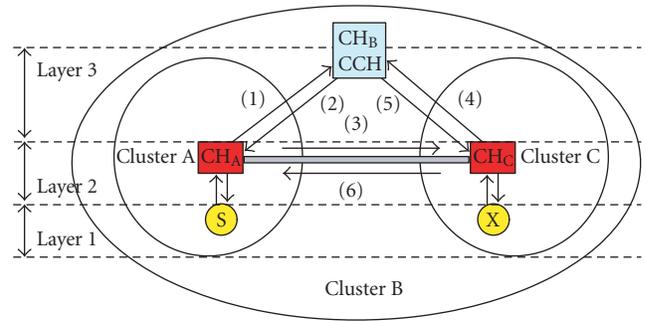


FIGURE 6: Authentication process for multiple layers within a large cluster network.

certificate has the following form:

$$CCH \rightarrow CH_A : cert_{CH_A} = [ID_{CH_A} || K_{CCH+} || e || Time_1]. \quad (1)$$

The certificate contains the ID address of the CH, the public key of the CCH, timestamp $Time_1$ for when the certificate was created, and time e at which the certificate expires. These variables are concatenated and signed by the CCH. Every CH must maintain fresh certificates with the trusted server and must know the CCH public key. CH_A sends a request message with a timestamp to CCH for a public key request to communicate with CH_B . If sending an encrypted message, CCH uses a private key that CH_A decrypts using the CCH public key.

3.3.5. A node joins a cluster for the first time

The ID address of the ID_{CH} , node S's certificate ($cert_S$), a nonce N_{CH} , and the current time t are all signed with A's private key. Each time S performs route discovery, it increases the nonce monotonically. Nodes then store the nonce they last saw with its timestamp. In Figure 8, the node S appeared as nodes 2 and 3:

$$CH \rightarrow S : cert_S = [[ID_{CH} || K_{CH+} || e || Time_1], N_{CH}]. \quad (2)$$

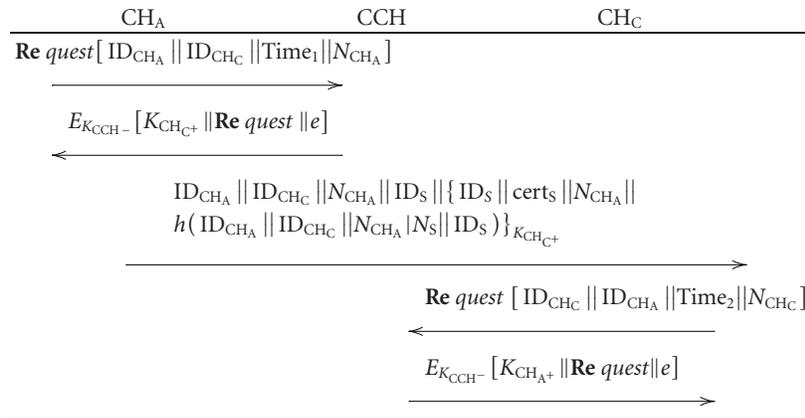


FIGURE 7: CHs authenticated from CCH.

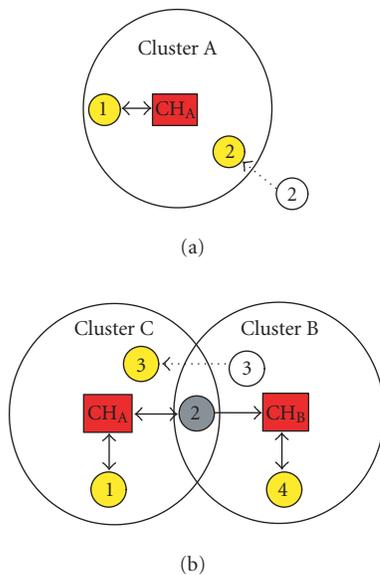


FIGURE 8: A node joins a cluster for the first time. (a) New node 2 joins cluster A for the first time. (b) Node 3 joins cluster A from cluster B.

The CH generates a random number N_{CH} and sends it to entry node A with its own cluster key. Source node A establishes a path message as a multicast to its own CH. Entry node A stores a cluster key for communication within the cluster. The public key for the encryption of random number N_{CH} is sent to CH.

3.3.6. Authentication for end-to-end of key exchange

So far, we have considered security services for communication from one cluster member to a cluster head. In an ad hoc network environment, securing the end-to-end path from one mobile user to another is the primary concern. The end-to-end security service minimizes the interference from intermediate nodes, especially malicious nodes. In this subsection, we present secure end-to-end authentication and

a key exchange protocol between one cluster member and another. The end-to-end key exchange progress is described in Figure 9. The end-to-end key exchange uses the Diffie-Hellman key as the public key.

Figure 6 shows the authentication process for multiple layers in large ad hoc networks. The CCH authenticates CHs. There are 7 steps required to implement AMCAN. Figure 9 shows the end-to-end authentication between CHs communicating after authentication using the CCH.

First, using a previously shared secret key K_{S,CH_A} , S sends a message to CH_A requesting communication with X. Since ID_S is encrypted using K_{S,CH_A} , other nodes except S and CH_A do not know the node with which S wishes to communicate. As $cert_S$ and N_S are also encrypted, they can be transferred securely.

Upon receiving the request, CH_A checks that S is a member. If so, this equals the progress leaving out steps (2) and (6) (i.e., $CH_A = CH_C$). Otherwise, CH_A asks the other cluster heads where X is using the CH_C public key, which was previously established in step (3) between cluster heads. Let X be a member of CH_B .

In step (3), X is informed of the request from S to communicate with him. CH_C sends S's certificate along with N_{CH_C} . Upon deriving the public key for S from the certificate, X calculates the session key $K_{X,S} = (PK_S)^{k_X} \bmod p$, which will be shared between S and X. X uses $K_{S,X}$ in step (4) to let CH_C know that it accepts S's request for communication. CH_C and CH_A pass to S the part of the message in step (4) that contains X's confirmation using $K_{S,X}$. CH_C and CH_A also forward X's certificate to S. Upon receiving a message including X's certificate, S can calculate the session key $K_{S,X} = (PK_X)^{k_S} \bmod p$ using PK_X derived from $cert_X$.

Finally, S and X share the same secret key, and S communicates with X by sending back X's nonce encrypted using the shared key $K_{S,X}$. We propose a reliable algorithm that runs strong authentication for each packet. This time, CCH performs authentication for all CHs, and CH authenticates the certification authority (CA) for all nodes in a cluster. The CH key is used to exchange the session key secretly. Therefore, all the messages described above can be forwarded for

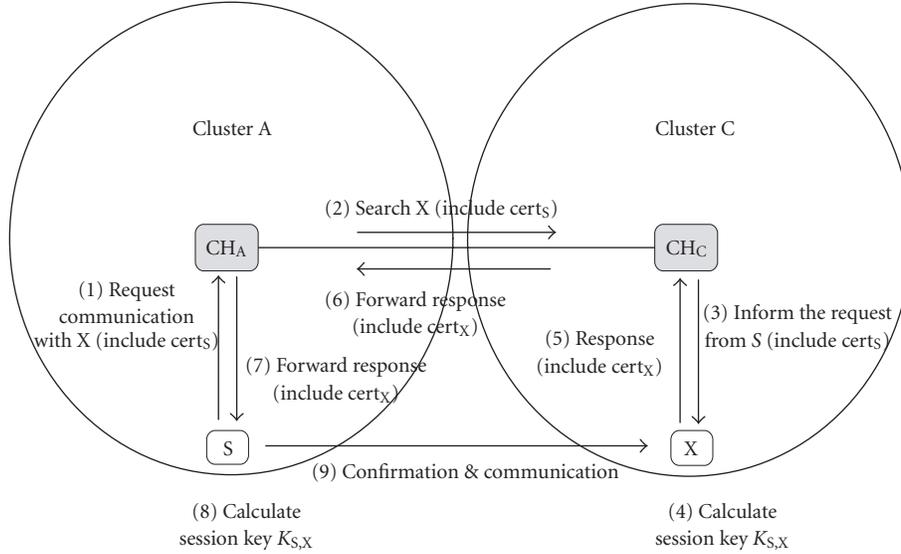


FIGURE 9: End-to-end authentication between clusters after the CHs are authenticated from the CCH.

reference by appending them to routing packets when a route is discovered.

4. EVALUATION AND PERFORMANCE ANALYSIS

4.1. Experiment of energy and mobility becoming a CCH

We used tools within Matlab to simulate the algorithm described in Section 3.2 for networks with varying node density (λ) and different values of the parameters p and k . Each node in the network chooses to become a CH with probability p and advertises itself as a CH to the nodes within its radio range. This advertisement is forwarded to all the nodes that are no more than k hops away from the CH. Any node that receives such advertisements and is not itself a CH joins the cluster of the closest CH. Any node that is neither a CH nor has joined any cluster itself becomes a CH. Because we have limited the advertisement forwarding to k hops, if a node does not receive a CH advertisement within time duration t (where t units is the time required for data from the CH to reach any node k hops away) it can infer that it is not within k hops of any volunteer CH and hence become a forced CH. Moreover, this limit on the number of hops allows the CH to schedule periodic transmissions to the processing center. To generate the network for each simulation experiment, the location of each node is found by generation of two random numbers uniformly distributed in $[0, 2a]$, where $2a$ is the length of a side of the square area in which the nodes are distributed. In all of these experiments, the communication range of each node was assumed to be 1 unit. To verify that the optimal values of the parameters p and k of our algorithm computed according to [20] formulae (11) and (13) do minimize the energy spent in the system, we simulated our clustering algorithm on node networks with 50, 100, and 200 nodes distributed uniformly in a square area of 10 square units. We have, without loss of generality, assumed that the

cost of transmitting 1 unit of data is 1 unit of energy. The processing center is assumed to be located at the center of the square area. For the first set of simulation experiments, we considered a range of values for the probability p of becoming a CH in the algorithm proposed in Section 3.2. For each of these probability values, we computed the maximum number of hops k allowed in a cluster using (13) and used these values for the maximum number of hops allowed in a cluster in the simulations. We simulated in a cluster in the simulations. We simulated the clustering algorithm 100 times for each density and each of the probability values and used the average energy consumption over the 100 experiments to plot the graph in Figures 10 and 11.

4.2. Compare ARAN and AMCAN

In this section, we compare the efficiency properties of the existing CCH key establishment protocol and our proposed scheme. We also compare end-to-end security and move distance within a cluster. Table 3 presents the total message and the total number of move distance operations necessary for each protocol. The efficiency numbers for existing solutions are given in tables for each protocol. None of the existing solutions achieve end-to-end security. In AMCAN, variable c is the number of CHs. We assume that CCH establishment among CHs uses ARCH, CBRP, and DMAC. As AMCAN also establishes authentication based on a trust layer, it also achieves end-to-end security.

We evaluated the performance of our protocol and identified the advantages and limitations of the proposed approach. In this paper, our proposed AMCAN protocol is used in an ad hoc network environment. The certificate mechanism uses the certification method from the ARAN identification protocol within a cluster. The CH establishes a member node that is worthy of trust by the members of a CH. Falsehood certification in the certification process can be achieved. AMCAN is a little more stable for certification of

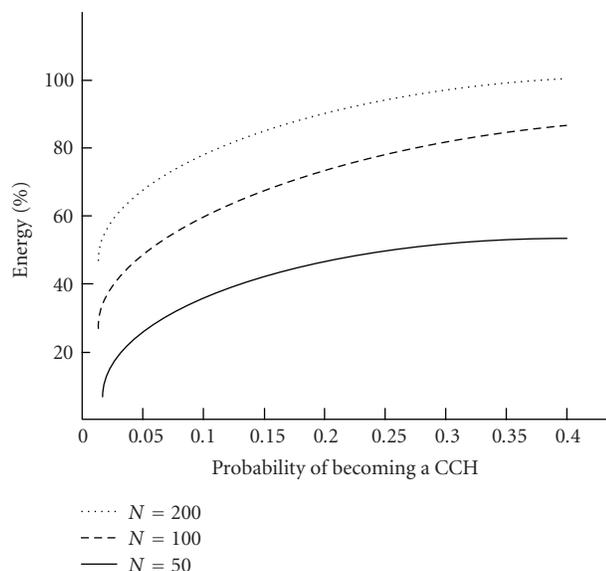


FIGURE 10: Total energy in a network of n nodes distributed in an area of 10 square units for different values of probability of becoming a CCH in the algorithm in Section 3.2.

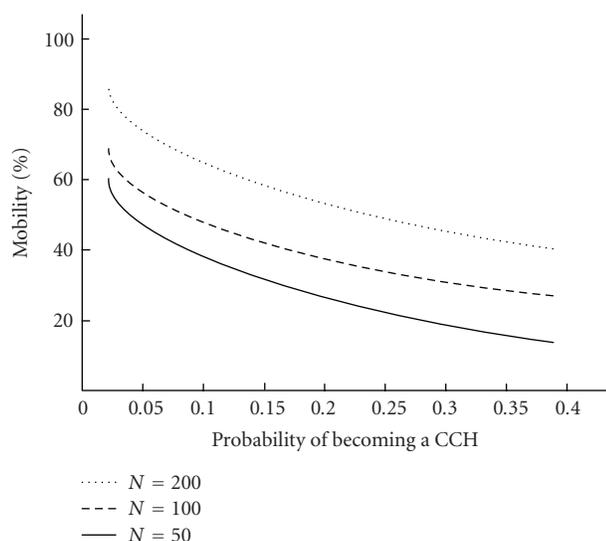


FIGURE 11: Mobility in a network of n nodes distributed in an area of 10 square units for different values of probability of becoming a CCH in the algorithm in Section 3.2.

CH using CCH and has fewer processing operations. The ARAN protocol distinguishes the nodes of a local distance area as a cluster. Table 4 presents AMCAN superior for large networks as it was designed for use in such networks. The AMCAN protocol has strong security as it uses the CCH to obtain a higher level of security than that of ARAN.

The advantages and limitations of the proposed approach have been identified. The certificate mechanism uses the certification method of the ARAN identification protocol within a cluster. AMCAN minimizes the process of chang-

ing certificates by using clustering-routing protocols. An analysis of its stability verified its authentication, efficiency, safety, and scalability. Authentication and nonrepudiation use a cryptographic certificate. Each node receives a certificate from the CH.

We evaluated three performance metrics.

- (i) Unauthorized participation: AMCAN participation accepts only packets that have been signed with a certified key issued by a trusted authority. There are many mechanisms for authenticating users to a trusted certificate authority. The trusted authority is also a single point of failure attack.
- (ii) Spoofed route signaling: since only the source node can sign using its own private key, nodes cannot spoof other nodes in route instantiation. Similarly, reply packets include the destination node's certificate and signature, ensuring that only the destination can respond to route discovery.
- (iii) Reply attacks: reply attacks are prevented by including a nonce and a timestamp with the routing message.

AMCAN minimizes changes in the certificate process of cluster networks. The analysis of scalability verified the authentication, efficiency, safety, and scalability of the method.

Protocol analysis

We need to show that the above protocol is an AMCAN.

Lemma 1. *The protocol described in Section 3 is designed for AMCAN.*

Proof. The protocol can be performed as follows: receiver CH_C authenticates $ID_S \parallel ID_{CH_A} \parallel cert_S \parallel N_S$ for intercluster. Sender CH_A sends CCH including $ID_{CH_A} \parallel ID_{CH_C} \parallel Time_1 \parallel N_{CH_A}$. AMCAN further improves the stability by the use of a nonce. AMCAN can reduce system energy use by dividing the parts to be handled in each CH. The CCH offers safe authentication of each node through management of the CHs. \square

Computation costs

The computation costs are calculated as

$$K_{S,X} = (PK_X)^{k_S} \bmod p, \quad (3)$$

and our protocol uses an encryption/decryption protocol that requires a total of 1 operation of $K_{S,X} = (PK_X)^{k_S} \bmod p$, which can be computed efficiently using the standard AMCAN. The CCH is achieved using the threshold scheme, thereby reducing the computation overhead because the ARAN protocol step has 12 steps but the AMCAN protocol step has 7 steps.

5. CONCLUSION

In this paper, we examined possible methods for use against ad hoc routing protocols, defined various security

TABLE 3: Performance evaluation on each protocol.

Item	Protocol	
	ARAN	AMCAN
Encryption algorithm	RSA digital signature	Diffie-Hellman
Total number of session keys	n	2
Total number of message	n (n : node number)	n/c (c : cluster number)
End-to-end security of area	X (small area)	O (small and large area)
Move distance	More 2 hops	2 hops

X: no (impossible), O: yes (possible).

TABLE 4: Characteristics on each protocol.

Item	Protocol	
	ARAN	AMCAN
Authentication	O	O
Efficiency	O	O
Safety	O	O
Scalability	X	O

X: no (impossible), O: yes (possible).

environments, and offered a secure solution with authentication based on multilayer clustering for ad hoc networks (AMCAN). We showed ways to exploit two protocols that are under consideration for clustering-based routing protocols and the ARAN identification protocol. Clustering-based protocols are efficient in terms of network performance. Our proposed protocol, called AMCAN, detects and protects against malicious actions across multiple layers and by peers in one particular ad hoc environment. AMCAN introduces authentication, efficiency, safety, and scalability to an ad hoc environment as part of a minimal security policy. Our evaluation showed that AMCAN has minimal performance costs in terms of processing and networking overhead for the increased security that it offers. In this paper, we examined the certification process for clustering routing protocols in ad hoc networks, and designed a certification protocol for AMCAN. The basic idea of AMCAN is to propose a CCH that has top-layer authority. We propose an authentication protocol that uses certificates containing an asymmetric key and a multilayer architecture so that the CCH is achieved using the threshold scheme, thereby reducing the computational overhead and successfully defeating all identified attacks. We also use a more extensive area, such as a CCH, using an identification protocol to build a highly secure, highly available authentication service, which forms the core of our security framework.

ACKNOWLEDGMENT

This work was done under a University Fundamental Research Program supported by Ministry of Information & Communication in Republic of Korea.

REFERENCES

- [1] M. Jiang, J. Li, and Y. C. Tay, "Cluster based routing protocol (CBRP) functional specification," IETF internet draft, MANET working group, August 1999, <http://www.comp.nus.edu.sg/~tayyc/cbrp/draft-ietf-manet-cbrp-spec-01.txt>.
- [2] E. M. Belding-Royer, "Multi-level hierarchies for scalable Ad Hoc routing," *Wireless Networks*, vol. 9, no. 5, pp. 461–478, 2003.
- [3] S. Basagni, "Distributed clustering for Ad Hoc networks," in *Proc. International Symposium on Parallel Architectures, Algorithms, and Networks (ISPA '99)*, pp. 310–315, Fremantle, Australia, June 1999.
- [4] M. Bechler, H.-J. Hof, D. Kraft, F. Rahlke, and L. Wolf, "A cluster-based security architecture for Ad Hoc networks," in *Proc. 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, vol. 4, pp. 2393–2403, Hong Kong, March 2004.
- [5] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer, "A secure routing protocol for Ad Hoc networks," in *Proc. 10th IEEE International Conference on Network Protocols (ICNP '02)*, pp. 78–87, Paris, France, November 2002.
- [6] A. C.-F. Chan, "Distributed symmetric key management for mobile Ad Hoc networks," in *Proc. 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, vol. 4, pp. 2414–2424, Hong Kong, March 2004.
- [7] L. Zhou and Z. J. Haas, "Securing Ad Hoc networks," *IEEE Network*, vol. 13, no. 6, pp. 24–30, 1999.
- [8] M. Steenstrup, "Cluster-based networks," in *Ad Hoc Networking*, C. E. Perkins, Ed., chapter 4, pp. 75–138, Addison-Wesley, Reading, Mass, USA, 2000.
- [9] L. Venkatraman and D. P. Agrawal, "A novel authentication scheme for Ad Hoc networks," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC '00)*, vol. 3, pp. 1268–1273, Chicago, Ill, USA, September 2000.
- [10] L. Kleinrock and F. Kamoun, "Hierarchical routing for large networks: performance evaluation and optimization," *Computer Networks*, vol. 1, no. 3, pp. 155–174, 1977.
- [11] J. Sucec and I. Marsic, "Clustering overhead for hierarchical routing in mobile Ad Hoc networks," in *Proc. 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, vol. 3, pp. 1698–1706, New York, NY, USA, June 2002.
- [12] L. Buttyán and J.-P. Hubaux, "Report on a working session on security in wireless Ad Hoc networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 7, no. 1, pp. 74–94, 2003.
- [13] Y. G. Desmedt, "Threshold cryptography," *European Transactions on Telecommunications*, vol. 5, no. 4, pp. 449–457, 1994.

- [14] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.
- [15] L. Zhou and Z. J. Hass, "Securing Ad Hoc network," *IEEE Network*, vol. 13, no. 6, pp. 24–30, 1999.
- [16] Y. G. Desmedt and Y. Frankel, "Threshold cryptosystems," in *Proc. 9th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '89)*, G. Brassard, Ed., vol. 435 of *Lecture Notes in Computer Science*, pp. 307–315, Springer, Santa Barbara, Calif, USA, August 1990.
- [17] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-Hellman key distribution extended to group communication," in *Proc. 3rd ACM Conference on Computer and Communications Security (CCS '96)*, pp. 31–37, New Delhi, India, March 1996.
- [18] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. International Cryptology Conference on Advances in Cryptology (CRYPTO '84)*, vol. 196 of *Lecture Notes in Computer Science*, pp. 47–53, Springer, Santa Barbara, Calif, USA, August 1985.
- [19] D. Boneh and M. K. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. 21st Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '01)*, vol. 2139 of *Lecture Notes in Computer Science*, pp. 213–229, Springer, Santa Barbara, Calif, USA, August 2001.
- [20] S. Bandyopadhyay and E. J. Coyle, "Minimizing communication costs in hierarchically-clustered networks of wireless sensors," *Computer Networks*, vol. 44, no. 1, pp. 1–16, 2004.

Keun-Ho Lee received the B.S. degree in computer science from SoonChun-Hyang University, Korea, in 1998, and the M.S. degree in electronic commerce from SoonChunHyang University, Korea, in 2001. He is currently a Ph.D. candidate in computer science and engineering at Korea University, Korea. He is also a researcher in the Research Institute of Computer Information and Communication at Korea University. His research interests include ad hoc, sensor, ubiquitous, and mobile communication security.



Sang-Bum Han received the B.S. degree in computer science from Seoul National University of Technology, Korea, in 1997, and the M.S. degree in computer engineering from Korea University, Korea, in 2001. He is currently a Ph.D. candidate in computer science and engineering at Korea University, Korea. He is also a Manager of the Network Operation Center in Korea Telecom. His main fields are in wireless network, ad hoc mobility management, and network security.

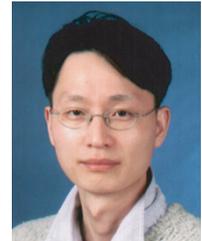


Heyi-Sook Suh received the B.S. degree in computer science from SookMyung Women's University, Korea, in 1988, the M.S. degree in computer education from Korea University, and the Ph.D. degree in computer science from Korea University. She is currently an Assistant Junior Official in the Ministry of Education & Human Resources, Seoul, Korea. She is also a researcher at the Center for Modeling & Simulation in Korea Institute for Defense Analyses (KIDA). Her



main fields are in wireless network, mobility management, network security, HLA/RTI, and so forth.

SangKeun Lee received his B.S., M.S., and Ph.D. degrees in computer science and engineering, Korea University, Seoul, South Korea, in 1994, 1996, and 1999, respectively. Since 2003, he has been an Assistant Professor in computer science and engineering, Korea University, Seoul, South Korea. His research interests include data management in mobile/pervasive computing systems, location-based information systems, XML databases, and mobile ad hoc networks.



Chong-Sun Hwang received the M.S. degree in mathematics from Korea University, Korea, in 1970, and the Ph.D. degree in statistics and computer science from the University of Georgia in 1978. From 1978 to 1980, he was an Associate Professor at South Carolina Lander State University. He is currently a Full Professor in the Department of Computer Science and Engineering at Korea University, Seoul, Korea. Since 1995, he has been a Dean in the Graduate School of Computer Science and Technology at Korea University. His research interests include distributed systems, distributed algorithms, and mobile computing systems.

