이동 Ad Hoc망을 위한 다중 계층 클러스터링 기반의 인증 프로토콜

(An Authentication Protocol-based Multi-Layer Clustering for Mobile Ad Hoc Networks)

이 근 호 † 한 상 범 † 서 혜 숙 †† 이 상 근 ††† 황 종 선 †††
(Keun-Ho Lee) (Sang-Bum Han) (Heyi-Sook Suh) (SangKeun Lee) (Chong-Sun Hwang)

요 약 본 논문에서는 ad hoc 망에서의 다중 계층 기법 기반의 안전한 클러스터 라우팅 프로토콜을 표현하였다. 클러스터기반 라우팅인 ARCH와 DMAC의 수정을 통해 CH와 CCH의 선택 알고리즘을 이용하여, ad hoc 라우팅 프로토콜에서의 보안 위협 요소에 대한 효율적인 프로토콜인 AMCAN(Ad hoc 망을 위한 다중 계층 클러스터링 기반의 인증)을 제안하였다. 본 프로토콜은 ad hoc 망에서 임계치(threshold) 인증 기법을 통한 Shadow Key를 이용하여 확장성을 제공하였다. 제안된 프로토콜은 다른 클러스터 노드들간의 상호 신뢰 관계를 갖는 종단간 인증 프로토콜로 구성하였다. 제안된 기법은 규모가 넓은 ad hoc 망에서 임계치 인증 키 구성의 장점을 갖는다. 본 논문의 실험결과를 통해 다중 계층 라우팅 기법에서 임계치 키 구성을 이용한 임시 세션키의 안전한 분산을 통해 노드 ID 위조를 방지하고, 상호 종단간 인증과 Reply Attack 을 검출하여 안전한 채널을 구축하는 것을 확인하였다.

키워드 : 보안, 인증, 애드혹 망, 클러스터링, 다중계층, 클러스터헤드

Abstract In this paper, we describe a secure cluster-routing protocol based on a multi-layer scheme in ad hoc networks. We propose efficient protocols, *Authentication based on Multi-layer Clustering for Ad hoc Networks* (AMCAN), for detailed security threats against ad hoc routing protocols using the selection of the cluster head (CH) and control cluster head (CCH) using a modification of cluster-based routing ARCH and DMAC. This protocol provides scalability of Shadow Key using threshold authentication scheme in ad hoc networks. The proposed protocol comprises an end-to-end authentication protocol that relies on mutual trust between nodes in other clusters. This scheme takes advantage of Shadow Key using threshold authentication key configuration in large ad hoc networks. In experiments, we show security threats against multilayer routing scheme, thereby successfully including, establishment of secure channels, the detection of reply attacks, mutual end-to-end authentication, prevention of node identity fabrication, and the secure distribution of provisional session keys using threshold key configuration.

Key words : security, authentication, ad hoc network, clustering, multi-layer, cluster head

1. Introduction

Mobile ad hoc networks security issues have became a central concern and are increasingly important. Ad hoc networks can not 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 [1]. Studies of secure cluster routing based on multiple layers in ad hoc networks have been carried out using[1,2].

· 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성·지원 사업의 연구결과로 수행되었음
† 학생회원 : 고려대학교 컴퓨터학과
root1004@korea.ac.kr
topflite@korea.ac.kr
†† 정 회 원 : 교육인적자원부 교육정보화기획과 사무관
suh@moe.go.kr
††† 종신회원 : 고려대학교 컴퓨터학과 교수
yalphy@korea.ac.kr
hwang@disys.korea.ac.kr
논문접수 : 2005년 4월 8일
심사완료 : 2006년 4월 13일

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 Multi-layer Clustering for Ad hoc Networks*'(AMCAN). We present detail the ways to exploit protocols that are under consideration by [1,3,5,6].

Our proposed scheme 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(Control Cluster Head) is achieved using the threshold scheme, so that the number of essential encryptions successfully defeats all identified attacks.

While this basic idea has been proposed before [2,4,5], we are the first to apply it to a clustered network. We describe our security concept in detail as a CCH construction algorithm. Our scheme 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(Cluster Head) traffic using for threshold authentication key configuration scheme of the CCH.

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 Multi-layer 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 multi-layer 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.

2.1 Clustering in Ad Hoc Networks

A comprehensive overview of different clustering strategies is presented in [2,3]. 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].

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.

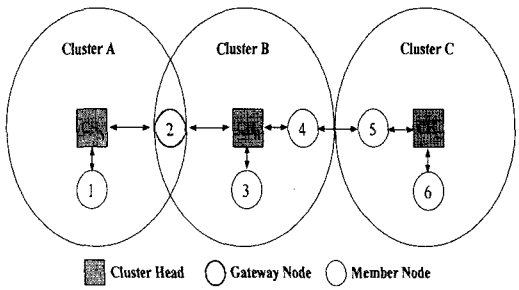


Fig. 1 Clustering-based architecture

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.

The DCA is suitable for clustering "quasi-static" ad hoc networks [3]. The DMAC algorithm adapts to the changes in network topology owing to the mobility of nodes, and is suitable for any mobile environment [3].

2.2 Security in Ad Hoc Networks

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 pre-shared 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 her 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].

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. In particular, they presented a threshold cryptosystem based on the Diffie-Hellman problem. The secret sharing scheme 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,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 will be computed in modulo p. The scheme is described in the following protocol. Distribution of trust in our key management service is accomplished using threshold cryptography [7,8]. 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.

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 non-repudiation. 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].

The idea to use a distributed certification authority based on a shared certification key and threshold cryptography for securing ad hoc networks. Our approach is based on the same general idea used by [5], 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 multi-layer cluse-ring for ad hoc networks

3.1 Scenario for an Experiment in AMCAN

Assumptions

Our proposed scheme is based on the following assumptions. All nodes have same environment. 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. 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. During the initial configuration there may exist only a CH with the repository. In this repository each nodes ID and their weight is stored along with a priority, mobility, lifetime, and energy. The AMCAN CCH only authorizes CHs until the expiry of their lifetimes. The certification automatically expires when this period expires. According to a pre-agreed set of rules, the session between two CHs are also terminated when the certification expires. All CHs requiring extend authorization must apply for a new certification before expiration of the last certification.

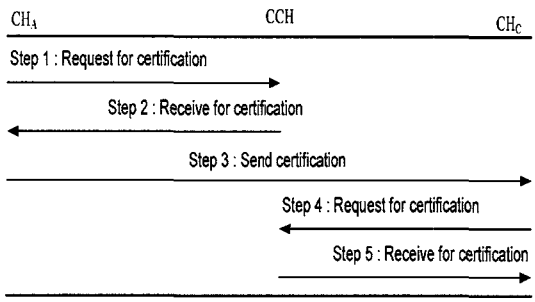
Operation of AMCAN

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 [8] 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. Figure 2 illustrates how the service is configured. Moreover, we propose applying the use of ID-based 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. The general operation of AMCAN is shown in Figure 2.

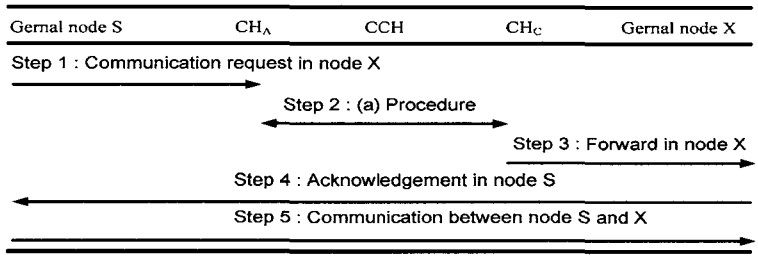
Whenever a node S desires to undertake secure communication with another node X, it sends a request to one of the AMCAN server CCH. The CCH creates a certification, which contains the session key for the requested communication and sends it back to the node S. The node S in turn sends this certification to the target node X with which communication is desired. The recipient node X acknowledge the certification and a secure session is established authenticated CHs between the two nodes using the session key provided by the CCH. The CCH also exchange data through an encrypted channel, however, they don't require the certification as they already posses the session key.

3.2 Architecture of a Multi-Layer Cluster

We describe efficient authentication algorithm for the set up and maintenance of cluster organization in the presence of node mobility that satisfies the two DMAC and the ARCH for the ad hoc clustering routing protocol. The selection of the CH and CCH uses the modification of the DMAC and ARCH algorithm in [2,3]. The modification DMAC in our clustering algorithm includes only two conditions: clustering set up to change the CH and clustering maintenance. The choice of the CHs is based on the unique ID associated to each node: the node with the lowest ID is selected as CH, then the cluster is formed by that node and all its



(a) First procedure for end-to-end authentication: CCH authenticated between CHs



(b) Second procedure for end-to-end authentication between node S and node X

Fig. 2 Operation of AMCAN

neighbors. Every node v in the network we assigned a unique identifier (ID). Each node has a unique identifier *NodeID* and knows the *Cluster ID's* of its one-hop neighbors. *NodeIDs* are the physical hardwired addresses (i.e., MAC addresses). In AMCAN, the *HID*(Hierarchical ID) of a node is defined as the sequence of the MAC addresses of the nodes on the path from the top hierarchy to the node itself. Finally, we consider weighted networks, i.e., a *weight* w_v is assigned to each node v of the networks. The selection of the CHs and *control CH* (CCH) is comparing based on the *weight* associated to each node: power quantity maximum of a node, minimum mobility in cluster, and the lowest ID in cluster.

CH Selection Algorithm

We use only two types of messages: *Ch(v)*, used by a node v to make its neighbors aware that it is going to be a CH, and *Join(v,u)*, with which a node v communicates to its neighbors that it will be part of the cluster whose CH is node u.

Here, we use the same two types of message used in the DCA(namely, *Ch(v)* and *Join(v, u)*) in [3]. In the following we use *Cluster(v)* and *ClusterHead* to indicated the set of nodes in the cluster whose CH is v and the CH 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(-)*, and *Cluster(-)* are initialized to nil, false and Ø, respectively. The following is the description of the two procedures as executed at each node v. 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 procedure CH selection 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 $\{z \in (v) : w_z > w_v \wedge Ch(z)\} \neq \emptyset$
  then begin
    $x := \max_{x_x > w_v} \{z : Ch(z)\}$;
    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

If among its neighbors there is at least a CH with bigger weight, then v will join it. Otherwise it will be a CH. Notice 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 CH. When a neighbor u becomes a CH, on receiving the corresponding CH message, node v checks if it has to affiliate with u, it checks whether w_n is bigger than the weight of v's CH or not. In this case, independently of its current role, v joins u's cluster[3].

CCH Selection Algorithm

On receiving the message *Join(u,z)*, the behavior of node v depends on whether it is a CH 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 *Cluster(v)*) or if u belonged to its cluster and is now joining an other cluster ($z \neq v$: in this case, u is removed from *Cluster(v)*). If v is not a CH, it has to check if u was its CH. Only if this is the case, v has to decide its role: It will join the biggest CH x in its neighborhood such that $w_x > w_v$ if such a node exists. Otherwise, it will be a CCH(Control ClusterHead). The CCH is v. The CCH weight need slowly mobility, the lowest of ID and enough of Energy in CHs. u parameter contents included mobility, ID and energy.

```
begin
    if Ch(v)
    then if z = v
              then Cluster (v) := Cluster (v) ∪ {u}
         else if u ∈ Cluster (v)
         then Cluster (v) := Cluster (v) \ {u}
    else if ControlClusterHead = u then
            if {z ∈ (v) : w_z > w_v ∧ Ch(z)} ≠ Ø
               then begin
                    x := max_{wz>wv} {z :Ch(z)};
                    send Join (v,x);
                    ControlClusterHead :=x
            end
     else begin
            send Ch(v)
            Ch(v) := true;
            ControlClusterHead := v;
            Cluster (v) := {v}
             end
    end;
```

3.3 Design of AMCAN

We use the notation listed to describe the proposed scheme this letter as the following:

Table 1 Variables and notation used in AMCAN

- CH_A : Cluster Head in cluster A
- ID_X : Identity of X
- $K_{S\,CH}$: Session key between S and CH or Secret key shared with S and CH
- $Time$: Current time
- S : Member node in CH_A
- X : Member node in CH_B
- K_{A+} : Public key of node A
- K_{A-} : Private key of node A
- $cert_A$: Certificate belonging to node A
- e : Certificate expiration time
- $Nonce_A$: Nonce issued by node A

Given a service consisting of three *CH*s, 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.

Assume that CH_B has been compromised. For a message m, CH_i can generate partial signatures $PS(m,CH_i)$ using its share Ch_i of the private key. 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.

The proposed key management is based on an improved cluster infrastructure. However, this scheme can supply a stronger, more secure environ-

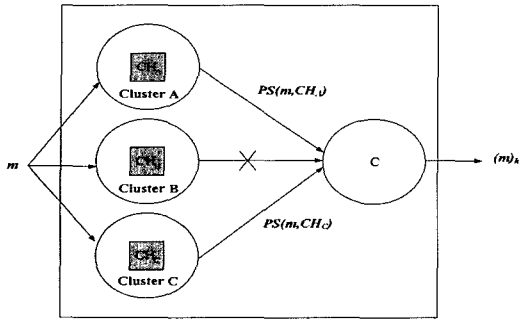


Fig. 3 Threshold authentication key configuration signature

ment based on hierarchical cluster modification using [6]. Under the proposed scheme, the encryption key is divided into n parties and distributed among n CHs. The key can only be reconstructed by acquiring the CCH selection algorithm using weight. In the proposed scheme, the n CHs of the key management services share the ability to revive the SK (shadow key) using the CCH. Our scheme shadow key is applied in the end-to-end or intercluster cases. The concept of the shadow key configuration is given in [10]. Our proposed DSRR (double static round-robin) algorithm presents the shadow mechanism. This proposed architecture design involves a hexagonal cell architecture.

In this paper, we apply shadow scheme key management using the DSRR concept. To ensure that the service can tolerate $k-1$ compromised CHs, an (n,k) threshold cryptography scheme is used and the shadow key (SK) of the service is divided into n clusters, and one share is assigned to each CH. Private key k is the SK.

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 SK of the service into n shares (shadows A - G), assigning one share to each CH. We call (shadows A - G) an $(n,t+1)$ sharing of the SK. Figure 6 illustrates how the shadow service is configured. The key management service consists of n CHs. As a whole, the service has a public/private key pair K/k. All the nodes in the network know the public key K, whereas the private key k

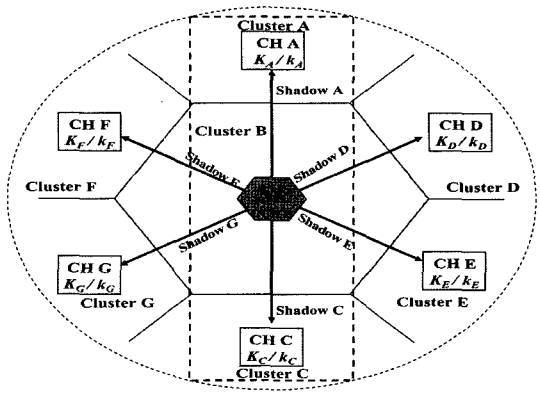


Fig. 4 The configuration of a key management shadow key service

is divided into n shares (shadows A - G), with one share for each CH. Each CH_i also has a public/private key pair K_i/k_i and knows the public keys of all nodes. Our proposed scheme is based on CH mobility using shadow keys.

The CCH takes the role of the parent CA and the CH takes that of the child CA. The CCH is the same as the top CA in a PKI. The CH and CCH are generated using the selection algorithm described in Section 3.2. The key management hierarchical architecture is shown in Figures 4. The CCH has a key pair (K,SK) using public key cryptography. SK is the shadow key that uses the CCH private key, and is used to generate the public key certificates of CH. K is the public key, which verifies the authenticity of the public key certificates of the CHs. K is distributed to every CH in the cluster network, and SK is protected by the secret sharing scheme.

AMCAN consists of a preliminary certification process, a mandatory end-to-end or intercluster authentication step, and an optional second step that provides a shadow key using a threshold cryptosystem.

- ***Authentication and Registration CH Using CCH***

AMCAN uses cryptographic certificates to bring authentication, message integrity, and non-repudiation to the cluster configuration. In this paper, this requires the use of a trusted certificate server, T, whose public key is known to all valid CHs.

The CCH requires the use of a trusted certificate server T [5]. All CHs receive a certificate from

CCH, as shown in Figure 8. A CH certificate has the following form:

$$CCH \rightarrow CH_A : cert_{CH_A} = [ID_{CH_A} \| K_{CH_A+} \| e \| Time_1]K_{CCH-}$$

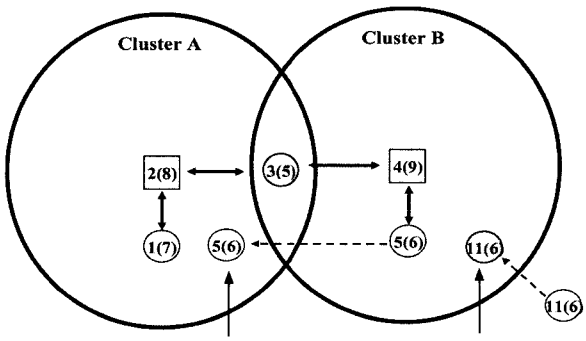
The certificate contains the ID address of the CH, the public key of the CCH, time stamp $Time_1$ for when the certificate was created, and time e at which the certificate expires. These variables are concatenated and signed by the CCH. Every CH must maintain fresh certificates with the trusted server and must know the CCH public key. CH_A sends a request message with a time stamp 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.

• ***Changing Cluster Node***

As the composition of the CH network changes dynamically when CHs are added, deleted, and merged in the network, the secret shares also must be renewed regularly because the number of shares needs to adapt to the number of CHs. Apart from that, it is necessary to ensure that the key shares are renewed after a certain period of time to make it difficult for a moving attacker to compromise several SK CHs over time. In our approach, we always combine the addition, deletion, and merger of CHs with key share renewal and only schedule additional renewals if the CH network remains unchanged for some time. The public key of the CH network must be known to all nodes in the *ad hoc* network. It is propagated via the CH beacons, which are broadcasted periodically in every cluster. Besides the public network key, a CH beacon also contains the CHs own public key, a list of nodes in the current cluster, including their status, and a list of gateways connecting to adjacent clusters. This beacon message contains information regarding the neighbors, including the clusters they belong to, adjacent clusters, and certificates. On receiving a message, a node updates its local related tables with the message information, and can detect the joining or leaving nodes. This method provides a useful means of maintaining cluster membership synchronization.

Figure 5 shows the join situation. When a node joins the cluster area for the first time, the CH detects that a new node has joined based on the messages in Figure 5(b). The system begins the CH selection algorithm for the remaining nodes that have not yet been chosen as the CH or assigned to a cluster area, when a node leaves the old cluster when a new node takes its cluster area (Fig. 5(a)). As the old CH receives newer messages from its one- and two-hop neighbors, the messages do not contain the member entities of the leaving node for a predefined time interval. Therefore, the old CH purges the member entity of this node.

When a new node joins the network and is detected by a CH, it receives the cluster key and the table containing the cluster ids, lifetime, mobility, weight, and CH public keys. When a node leaves a cluster and joins another cluster with the movement of nodes. In the new cluster, the new CH treats it as any new node joining its cluster. A mutual authentication is performed between the moved node and its new CH using the system key



(a) Node 5 joins Cluster A from Cluster B (b) New node 11 joins Cluster A for the first time

Fig. 5 A node joins a cluster for the first time

pair (*K/k*). The CH then gives the node the cluster key for the new cluster. The old cluster purges the entry for this node when it does not receive message within a predefined time interval.

The merging of two clusters into a single cluster is one of the most difficult and expensive operations. As the two cluster SKs cannot be mixed, one of them must be dropped and the other distributed over the entire network. All the certificates that had been signed with the dropped key eventually have to be reissued, although it is possible to keep the dropped key for a period of time to facilitate this process. It may become necessary to adapt a (*K*,SK) threshold for the changed number of nodes and CHs in the networks. If merging two bigger networks is difficult, any decision about the remaining network depends on parameters like the number of CHs and the number of nodes that would like to apply for a new configuration cluster using the CH selection algorithm and obtain new certificates.

• ***End-to-End Authentication Using CCH***

We have considered security services for communication from one cluster member to a CH. 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 for use between one cluster member and another. The end-to-end key exchange progress is described in Figures 6 and 7. The end-to-end key exchange uses the Diffie - Hellman

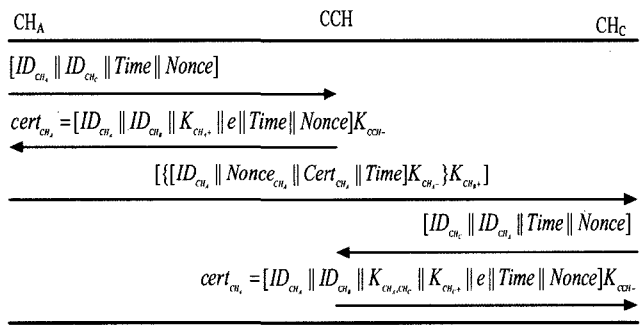


Fig. 6 CHs authentication from a CCH

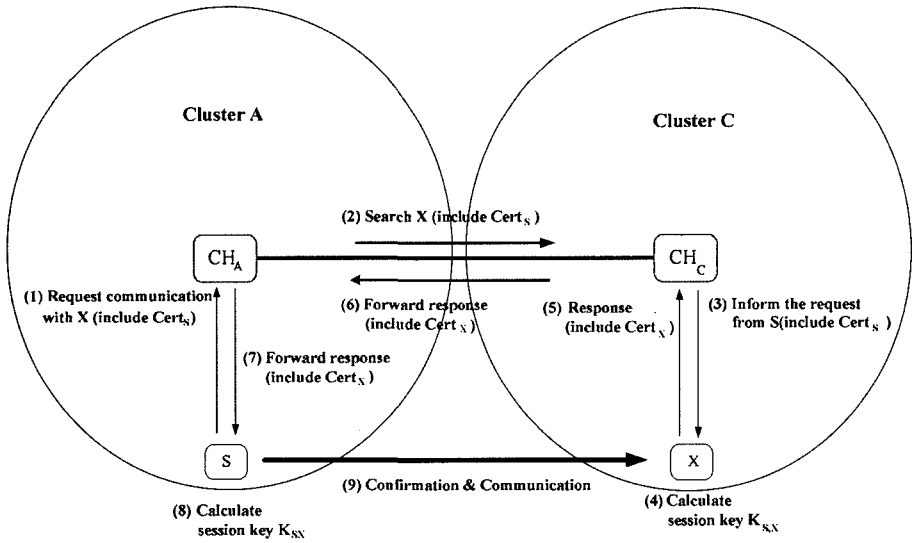


Fig. 7 End-to-end authentication between clusters after the CHs are authenticated from the CCH

key as the public key.

The CCH is a trusted certificate server. The CCH creates a certification that contains the session key for the requested communication and sends it back to the CHs. CH_A requests a certificate from its CCH for further communication with CH_C. The CCH first checks to see if both CH_A and CH_C have a valid lifetime associated with their user IDs. If so, then the CCH responds by providing CH_A a certificate to access CH_C.

$$CH_A \rightarrow CCH : [ID_{CH_A} \| ID_{Ch_C} \| Time \| Nonce]$$

All CHs receive a certificate from their CCH. A CH certificate has the following form:

$$CCH \rightarrow CH_A : cert_{CH_A} =$$

$$[ID_{CH_A} \| ID_{CH_B} \| K_{CH_A+} \| e \| Time \| Nonce] K_{CCH-}$$

Every CH must maintain fresh certificates with the trusted server and must know its CCH public key. CH_A sends a request message with a time stamp to its CCH for a public key request to communicate with CH_B. When it sends an encrypted message, the CCH uses a private key that CH_A decrypts using the CCH public key. A CH authentication protocol from its CCH is shown in Figure 8.

So far, we have considered the security services that would be used for communication from one cluster member to its CH. In an *ad hoc* network environment, the primary concern is securing the end-to-end path from one mobile user to another. The end-to-end security service used must minimize interference from intermediate nodes, especially malicious nodes. We will now present our secure end-to-end authentication with SK using a threshold authentication key configuration.

After authenticating the CH using the CCH, there are nine steps of our end-to-end key exchange procedure and authentication process for hierarchical clusters in large *ad hoc* networks, as show in Figure 9. A Diffie - Hellman key is used as the public key.

First, using a previously shared secret key $K_{S,CHA}$, S sends a message to CH_A requesting communication with X. Since ID_S is encrypted using $K_{S,CHA}$, only nodes S and CH_A know the node with which S wishes to communicate. As $Certs_S$ and $Nonces$ are also encrypted, they can be transferred securely.

On receiving the request, CH_A checks to see if S is a member. If this is the case, steps 2 and 6 shown in Figure 7 are not required. Otherwise, in step 2, CH_A asks the other cluster heads where X is using their public keys. Let us assume that X is located in cluster C. By using the CH_C public key that was previously established for communication between CHs, the search reveals that X is located in cluster C.

In step 3, X is informed of the request from S to communicate with it. CH_C sends S's certificate along with $Nonce_{CHC}$. On 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 in step 5 that it accepts S's request for communication. In step 6, 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. In step 7, S receives a message from CH_A that includes X's certificate. In step 8, S calculates the session key $K_{S,X} = (PK_X)^{k_S} \bmod p$ using PK_X derived from $Cert_X$. Finally, in step 9, S communicates with X by sending back X's nonce encrypted using their shared key $K_{S,X}$.

Our proposed algorithm is reliable because it uses strong authentication for each packet. The *CCH* performs authentication for all *CH*s. A *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 of the reference messages that are described above can be forwarded by appending them to routing packets once a route has been determined.

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

al 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 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 [9] formula (11) and (13) do minimized 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 Figure 8, 9.

4.2 Comparison of Authentication Scheme

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 2 presents the total message and 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 and DMAC. As AMCAN also establishes authentication based on a trust layer, it also achieves end-to-end security.

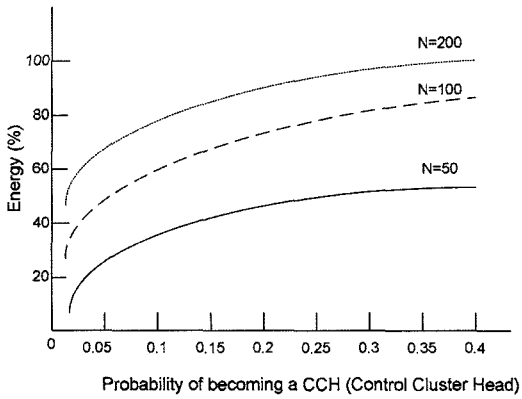


Fig. 8 Total energy in a network of n nodes distributed in an area of 10 square units for different values of probability of become a CCH in algorithm in Section 3.2

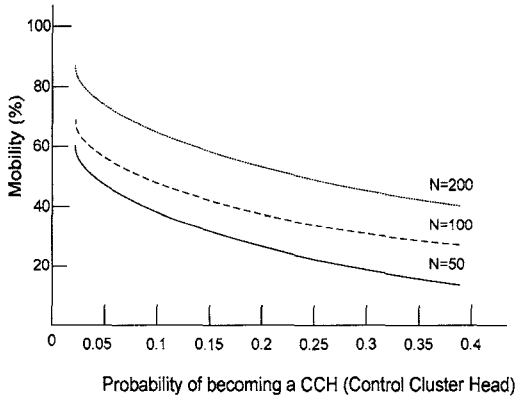


Fig. 9 Mobility in a network of n nodes distributed in an area of 10 square units for different values of probability of become a CCH in algorithm in Section 3.2

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 CH using CCH and has fewer processing operations. The ARAN protocol distinguishes the nodes of a local distance area as a cluster. AMCAN is 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 ARAN.

In Table 3 show characteristics ARAN and AMCAN. 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 changing certificates by using clustering-routing protocols. An analysis of its stability verified its authentication, efficiency, safety and scalability. Authentication and non-repudiation use a cryptographic certificate. Each node receives a certificate from the CH.

4.3 Security Analysis of AMCAN

In this section, we compare the efficiency properties of the existing clustering protocol and our proposed scheme. We discuss how AMCAN defies certain attacks possible in and ad hoc networks. 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. 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 CH using CCH and has fewer processing operations. AMCAN is 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 other clustering routing protocol.

An analysis of its stability verified its authentication, efficiency, safety and scalability. Authentication and non-repudiation use a cryptographic certificate. Each node receives a certificate from the CH.

We evaluated four performance metrics:

- **Modification attacks:** AMCAN can use the session keys for encryption the traffic flow of data and control packets. Thus, including the Diffie-Hellman key exchange the session key $K_{X,S} = (PK_S)^{k_X} \bmod p$ of message contents in every transmitted packet, guarantees the integrity of the contents along with confidentiality.
- **Fabrication attack:** The authenticity of the received control and data packets can be verified using the session keys using CCH. As the session keys are unique, fabricated packets can

Table 3 Characteristics on each protocol

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

X : Poor, △ : Normal, O : Good

Table 2 Performance evaluation on each protocol

Item	Protocol	
	ARAN	AMCAN
Encryption algorithm	RSA	Diffie-Hellman
Total number of session keys	0	2 (CH, CCH)
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	1 hop	More 2 hop

X : No(impossible), O : Yes(possible)

easily be verified and hence discarded.

- **Spoofed route attack and unauthorized participation:** AMCAN participation accepts only packets that have been signed with a certified key issued by a trusted authority using CCH. There are many mechanisms for authenticating users to a trusted certificate authority. Since only the source node can sign using its own private key, nodes cannot spoof other nodes in route instantiation. The encryption of all end-to-end traffic indirectly ensure the verification of packets, as the session keys are only held by the previously authenticated end points. As a consequence, the legitimacy of all packets is automatically verified during the decryption phase, ensuring that any packets that were spoofed are discard. Similarly, reply packets include the destination node's certificate and signature, ensuring that only the destination can respond to route discovery.
- **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.

5. Conclusion

We examined possible methods for use against ad hoc routing protocols, defined various security environments, and offered a secure solution with '*Authentication based on Multi-layer 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. 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 multi-layer architecture so that the CCH is achieved using the threshold scheme, thereby 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.

References

[1] M. Jiang, J. Li, and Y.C. Tay, "Cluster based routing protocol (CBRP)," functional specification, IETF Internet Draft, MANET working group, *draft-ietf-manet-cbrp-spec-01.txt*, Aug. 1999.

[2] Elizabeth M. BEDING-ROYER, "Multi-level hierarchies for scalable ad hoc routing," *Wireless Networks archive Volume9, Issue 5*, pages 461-478, Sept. 2003.

[3] S. Basagni, "Distributed clustering for ad hoc networks," in: *Proceedings of the 1999 International Symposium on Parallel Architectures, Algorithms, and Networks*, pages 310-315, Jun. 1999.

[4] M. Bechler, H.-j. Hof, D. Kraft, F. Rahlke, L. Wolf, "A Cluster-Based security architecture for ad hoc networks," in: *Proceedings of IEEE Conference on Computer Communications (INFOCOM)* Hong Kong, March. 2004.

[5] K. Sanzgiri, B. Dahill, B.N. Levine, C. Shields, E.M. Belding-Royer, "A secure routing protocol for ad hoc networks," in: *Proceedings IEEE Network Protocols*, pages 78-87, 2002.

[6] Lidong Zhou; Haas, Z.J.; · "Securing ad hoc networks," *IEEE Network, Volume: 13 Issue: 6*, Nov.-Dec. pages 24-30, 1999.

[7] Y. Desmedt and Y. Frankel, "Threshold cryptosystems," *Advances in Cryptology-Crypto '89 (Lecture Notes in Computer Science 435), G. Brassard, Ed, Springer-Verlag*, page 307-15, Aug. 1990.

[8] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-Hellman Key Distribution Extended to Group Communication," *3rd ACM Conference on Computer and Computer and Communications Security*, 1998.

[9] Seema Bandyopadhyay, Edward J. Coyle, "Minimizing communication costs in hierarchically-clustered networks of wireless sensors," Elsevier, Computer Networks 44 (2004) 1-16.

[10] H.-S. Suh, S.-B. Han, K.-H. Lee, C.-S. Hwang,

"DSRR organization and its algorithm for efficient mobility management in the SIP," *IEICE TRANS. COMMUN.*, vol. E87-B, no. 7, pp. 1866-1873, July 2004.

이 근 호

1998년 2월 순천향대학교 컴퓨터학과 졸업(이학사). 2001년 2월 순천향대학교 전자상거래학과 졸업(석사). 2003년 8월 고려대학교 컴퓨터학과 박사수료. 관심분야는 ad-hoc, 센서, 유비쿼터스 및 네트워크 보안

한 상 범

1997년 2월 서울산업대학교 컴퓨터과학과 졸업(이학사). 2001년 8월 고려대학교 컴퓨터학과 졸업(석사). 2003년 8월 고려대학교 컴퓨터학과 박사 수료. 현재 KT 서울강남네트워크서비스센터 PMC 팀장. 관심분야는 무선망, ad-hoc, 이동성 관리 및 네트워크 보안

서 혜 숙

1988년 2월 숙명여자대학교 전산학과 졸업(이학사). 2001년 8월 고려대학교 전산교육학과 졸업(석사). 2004년 8월 고려대학교 컴퓨터학과 졸업(박사). 현재 한국국방연구원 지식경영팀에서 서울대학교 정보화본부 정보화기획팀 사무관. 관심분야는 컴퓨터 네트워크, 이동 컴퓨팅, 이동성 관리, 네트워크 보안 및 차세대 연동체계(HLA/RTI)

이 상 근

1994년 2월 고려대학교 전산과학과(현, 컴퓨터학과) 학사 졸업. 1996년 2월 고려대학교 대학원 전산과학과(현, 컴퓨터학과) 석사 졸업. 1999년 8월 고려대학교 대학원 전산과학과(현, 컴퓨터학과) 박사 졸업. 2000년 4월~2001년 3월 동경대학교 생산기술연구소 특별연구원. 2001년 4월~2003년 2월 LG전자정보통신 단말연구소 선임연구원. 2003년 3월~현재 고려대학교 컴퓨터학과 부교수

황 종 선

1978년 Univ. of Georgia, Statistics and Computer Science 박사. 1978년 South Carolina Lander 주립대학교 조교수. 1981년 한국표준연구소 전자계산실 실장. 1995년 한국정보과학회 회장. 1982년 ~ 현재 고려대학교 컴퓨터학과 교수 1996년 ~ 현재 고려대학교 컴퓨터과학기술대학원 원장. 관심 분야는 알고리즘, 분산시스템, 데이타베이스, 이동컴퓨팅 등