# Distributed Security and Reliable Routing In the Cluster-Based Ad Hoc Network

Meng-Yen Hsieh and Yueh-Min Huang
*Department of Engineering Science, National Cheng-Kung University, Tainan 710, Taiwan*
tab.hsieh@mail.hku.edu.tw huang@mail.ncku.edu.tw

***Abstract***- *In mobile ad hoc networks, security and routing schemes are suggested by many ways. Some proposals use anonymous paths to protect communications with security and privacy against traffic analysis. Most anonymous communications are implemented with a global view of the wire network. One study provides to establish anonymous routing paths with distributed route construction algorithm in ad hoc networks. However, the algorithm does not have the reliability characteristic. The ad hoc network is often formed in groups. It is believed that the cluster-based network is easy to manage and solve network related problems. In this paper, we provide reliable communications to recover the remaining anonymous routing path in a cluster-based ad hoc network after a link was broken or a participant left its routing chain. We also give algorithms about the reliable anonymous communications to verify our proposal.*

**Keywords:** Ad hoc network, Distributed security, Reliable Routing, Cluster.

## 1. Introduction

In the recent years, mobile ad hoc networking technique is improving and popular due to careful research and strong papers. The goals of a large amount of research are (1) to response rapidly and correctly to requirements of ad hoc networking over quickly changes -- [6] for example, (2) to minimize and optimize transmission, processing, and the usage of storage resource -- [9] for example, and (3) to securely establish routing paths and communicate against attack as well as possible [4][7][8]. Some research provides cluster-based control structures and algorithms to efficiently use resource and for use in a wireless ad hoc network, for instance, [1][2][3]. Several different cluster-based ad hoc networks are presented and compared. Examples of such cluster-based wireless networks include: *link-cluster architecture* (LCA) [2], *virtual subnet architecture* [3], *adaptive routing using clusters* (ARC) [1], and so on. The *adaptive routing using clusters* loosely based on the *link-clustered architecture* (LCA) creates a one-level clustered hierarchy on an ad hoc network of nodes. These clusters are produced according to node proximity. Each cluster has a leader called cluster head to control and maintain information of members.

In wireless ad hoc environments, many applications apply unicast to delivery sensitive and valuable information possible regarding the nature and location of communication entities. Most network-based anonymity communications supply the methods of hiding the information. For instance, the DC-nets [10], Crowds [11], MIX networks [12] and Onion Routing [13] provide anonymous communication to protect message exchanges against network attacks. Their proposals demand the networks' topology, before establishing anonymous paths for end-to-end communications.

One paper issued by [4] proposes a distributed path construction protocol between end nodes for anonymous communication without the global topology in ad hoc networks. In [4], with executing two phases adopting *dynamic source routing* (DSR) protocol, the source node and destination node both gain a shortest routing path between them and key related information of all participants as intermediate nodes of the path. Thus they communicate securely by multiply encrptions using key information of all participants.

Achieving security in wireless ad hoc networks is a difficult and complex task because of the nature of the wireless environment and the lack of infrastructure [7]. Some proposed papers address the cluster head as PKI or CA to provide fundamental security services including authentication, digital signatures, and key distribution and management for cluster-based ad hoc networks. We make good use of the characteristic of clusters to enhance security. In our proposed procedures was designed with one aim of providing a secure end-to-end communication to apply one-hop and one-level cluster-based ad hoc networks. However, we need less re-clustering happened. And, distributed certification authority and secret sharing scheme are applied to our cluster-based concept for secure ad hoc networks.

The remainder of the paper is organized as follows. Secure and routing problems are briefly discussed in section 2. We initial a cluster-based ad hoc network with key distribution and construct an anonymous routing path in section 3. Our proposed reliable communications between two ends are presented in section 4. In section 5, our algorithms' characteristic is described. Section 6 concludes the paper.

## 2. Secure and Routing Problems

The mobile ad hoc network could has two mainly problems, security and routing problems. With the radio transmission, wireless nodes can broadcast a message to neighborhood. A malicious node can intrude the network to attack wireless nodes' communications. Due to dynamic topology of mobile ad hoc networks, wireless nodes can difficultly generate correct routing paths for communications and manage them.

### 2.1 Secure problems

There are two main classifications about security attacks, based on the nature of the attacker.

- Passive attacks:
  The attackers don't basically influence network communications and only eavesdrop or monitor the transferring message.
- Active attacks:
  The attacks is serious than passive attacks. They not only monitor network transmissions, but also affect network communications, for instance, modifying transferring messages or obstructing transmissions.

### 2.2 Routing problems

In wireless ad hoc networks, node mobility could introduce scalability and reliability problems about routing path establishment between wireless nodes.

- Scalability problem
  The construction of a routing path is most convenient with knowledge of global network topology. However, gaining and keeping the knowledge are hard and useless in wireless ad hoc networks. Because of wireless node population growth and mobility, a network global view is always changeable. When a lot of intermediate nodes participate in a routing path in ad hoc networks, the communications between nodes could has delayed action and secure problems
- Reliability problem
  Wireless node mobility must bring the reliability problem. A routing path will be incorrect during transferring messages due to broken links among nodes or the left nodes which is out of the path.

## 3. Initialization

In this section, we initial our cluster-based ad hoc network, introduce key distribution and function, and establish a anonymous routing path between two nodes.

### 3.1 Cluster Architecture and Cryptography

We prefer the ARC protocol to construct a one-level clustered hierarchy on ad hoc networks. Each cluster involves a cluster head and other members which are one or more gateways and zero or more ordinary nodes. The ARC doesn't bring the *rippling effect* which will be happened in the LCA and other cluster-based protocols. According to the ARC, nodes send "Hello" messages twice during clustering in the mobile ad hoc network. We assume initially each node has a unique identity, *id*, and enough capabilities of computing, key construction and distribution, and certification issue. Each node is able to establish asymmetric or symmetric keys. In the paper, the cryptography are based on secure end-to-end communications of [4] and authorization and key distribution of [14].

A cluster head are assigned the role of a single distributed CA , chosen by [14]. We propose a secret key of the CA, the so-called $K_{CA}$, is distributed over all cluster heads and a serial number called $SN_{CA}$ is corresponding to the key. Just like threshold cryptography, every cluster head holds a fragment of the $K_{CA}$ and a common $SN_{CA}$. Only cluster head can be allowed to gather secret shares from other enough cluster heads to reconstruct and employ the key.

Every node joing in this network holds a unsigned self-generated key pair, which will be used for providing secure routing discovery message after the key pair is signed. A node becomes a member of a cluster after direct connecting and logging on a cluster head and owns a signed pair of public/private keys, $PK_{c,id}/SK_{c,id}$ in the cluster after being authenticated by distributed CAs. Therefore the cluster head assures other members of the reliability of its identity and shares a individual symmetric key, $K_{cluster,id}$ ($K_{c,id}$), and a individual serial number, $SN_{cluster,id}$ ($SN_{c,id}$), with it. A serial number is corresponding to a symmetric key. Figure1 shows states of a new node during logging on.

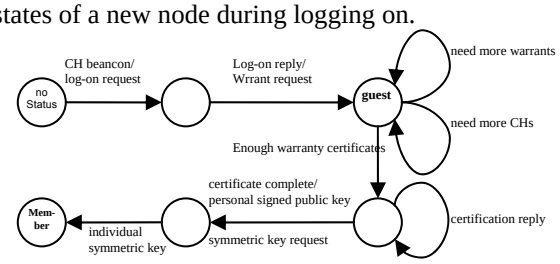


figure 1. States of a new node during log-on

A public key of a node, singed by distributed CA after authorization has completed successfully, will be issued with its certificate over all cluster heads. A node could receive two or more symmetric keys due direct connecting to two or more cluster heads. The first symmectric key issued by first-connected cluster head, the so-called master head, has highest priority for encryption in the node, avoiding confusing key usage in the period of routing discovery. Any cluster head needs to make periodic updates of symmetric keys and serial numbers of its members. And any member can request its cluster head to update its $K_{c,id}$

and $SN_{c,id}$, after finishing a session of communications The upates are executed in their privately respective secure channel.

### 3.2 Construct routing path with distributed anonymous route protocol

The major target of establishing anonymous path in ad hoc networks is to permit intermediate nodes as participants to join in the routing path construction protocol with jeopardizing the anonymity of the communicating nodes. We adopt the security dynamic distributed routing algorithm of [4], divided into two phases, the path discovery phase and the path reverse phase, to gain a shortest routing path.

The path discovery phase is executed with distributed information gathering about participants from the source node, called S, to the destination node, called D. S will produce a path discovery message consisting of four parts to discover D. The front tree parts are the following format:

$\boldsymbol{TPK_{c,S}, E_{PK_{c,D}}(Id_D, K_{c,S}), E_{K_{c,S}}(Id_S, PK_S, TPK_{c,S}, TSK_{c,S}, SN_{c,S}, PL_S, P_S, Sign_S(M_S))}$. **where** $\boldsymbol{M_S = H(TPK_{c,S}, TSK_{c,S}, ID_D, K_S, ID_S, PK_S, SN_{c,S}, PL_S, P_S)}$. The fourth part is about participants to append and encrypt their key related information ($K_{c,id}$,$SN_{c,id}$) to the message. Each participant uses its symmetric key shared with its cluster head expect cluster heads. Any cluster head participating in this part makes use of the $K_{CA}$ and $SN_{CA}$ instead of symmetric key. A path discovery message has traveled with participants, 1…*i*, from S to D would have the following format:

$\boldsymbol{TPK_{c,S}, E_{PK_{c,D}}(Id_D, K_S), E_{K_{c,S}}(Id_S, PK_S, TPK_{c,S}, TSK_{c,S}, SN_{c,S}, PL_S, P_S, Sign_S(M_S)), E_{TPK_{c,S}}(Id_1, K_{c,1}, SN_{c,session,1}, Sign_{Id1}(M_1)), E_{TPK_{c,S}}(Id_2, K_{c,2}, SN_{c,session,2}, Sign_2(M_2)),\ldots, E_{TPK_{c,S}}(Id_i, K_{c,i}, SN_{c,session,i}, Sign_i(M_i))}$.

The path reverse phase returns the information form D to S along the routing path producing by the path discovery phase. After receiving the path discovery message, D decrypts it and gains these keys and serial numbers of all participants. If a first participant as the node *i* receives the message from the D, the encrypted message would be the following format:

$\boldsymbol{E_{K_{c,i}}(E_{K_{c,i-1}}(\ldots\ldots\ldots E_{K_{c,1}}(E_{K_{c,S}}(SN_{c,session,1},K_{c,1}, SN_{c,session,2},K_{c,2},\ldots,SN_{c,session,i},K_{c\_i},PL_R,P_R),SN_{c,session,S}), SN_{c,session,1}),\ldots), SN_{c,session,i-1}), SN_{c,session,i}}$

Each participant accepts the message, skins one encryption layer with its key according to the $SN_{c,session,Id}$, and forwards the message to next participant. S(D) can encrypt and transfer data to D(S) with session keys of all participants, after finishing the two phases above. S makes the following layer encryption for the delivery data (DATA).

$\boldsymbol{E_{K_{c,1}}(E_{K_{c,2}}(\ldots\ldots\ldots E_{K_{c,i}}(E_{K_{c,D}}(DATA), SN_{c,session,D}), SN_{c,session,D}),\ldots), SN_{c,session,2}), SN_{c,session,1}}$

As well as D can produce the following layer encryption to transfer DATA to S.

$\boldsymbol{E_{K_{c,i}}(E_{K_{c,i-1}}(\ldots E_{K_{c,1}}(E_{K_{c,S}}(DATA), SN_{c,session,S}), SN_{c,session,1}),\ldots), SN_{c,session,i-1}), SN_{c,session,i}}$.

## 4. Reliable Communication

During delivering multi-layer encrypted messages from S(D) to D(S), as explaning above, a participant could fall away to fail a routing path. The section proposes extra algorithms for anonymous commincations to provide reliability, dealing properly with a disappeared participant being out of the routing path. We believe that the other participants still stay in their locations during the delivery of messages. Cluster heads in our paper is represented rescuers to be able to decrypt the messages encrypted with keys of their members. We depend on characteristic of cluster heads to support our reliability with two strategies. First, we replace the disappeared participant by its cluster head which has shared the symmetric key to be able to *unlock* encrypted message. Second, We attempt to find the next participant of this disappeared participant by way of findings of this cluster head or adjacent cluster heads, because we believe it is one of members in this cluster or adjacents clusters of the disappeared participant. We give a exmple as the following figure.

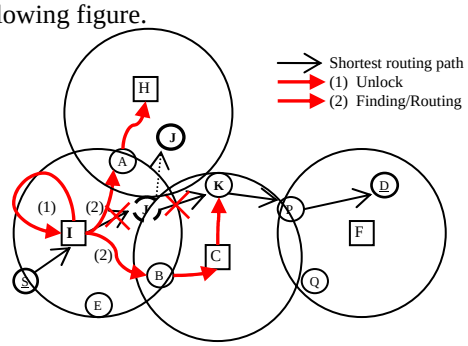


Figure 2. reliable communication

We use the word "*unlock*", representing the decryption of a cluster head for a member which is out of a routing path. A symmetric key is extracted from key storage by a serial number. If the particpant, out of a routing path, is a cluster head, other cluster heads in the network can *unlock* since all cluster heads employ common key information ($K_{CA}$,$SN_{CA}$). When the find is successful, the original routing path will be recovered and the message can be transferred forward again. For applying our strategies, we write two algorithms. The first is a function for original nodes or gateways, and the other is for cluster heads.

### 4.1. Algorithms of participants in the routing path

In our cluster-based ad hoc networks, all participants of a routing path are classed with *normal participants*, representing original nodes or gateways,

and *special participants*, representing cluster heads.

About the first algorithm, we assume that $N_i$ is represented as the current *normal* participant receiving a multi-layer encrypted message, called $E_{Ki}(M')$. $M'$ is the remainding multi-layer encrypted message on the delievery after removing the *i*th participant's encryption layer. $N_i$ gets its right symmetric key according to $SN_{cluster,session,i}$ to remove one encryption layer of the input and forwards the remainder to next participant represented as $N_j$ along a shortest routing path. Finally, the destination node will surely obtain the exposed message after decrypting the last encryption layer.

By accident, $N_i$ could have a fail forwarding due to the sudden disappearance of $N_j$ in the path. In our algorithm, the message will be labeled with a RH-tag and redirected to its cluster heads. A RH-tag standing for lock triggers off some actions of cluster heads. This algorithm could have a regular decryption output, $E_{Kj}(M')$, or a exceptionally fail forwarding result, $RH\text{-}E_{Kj}(M')$.

```
A normal participant receives a multi-layer encrypted message
Input:
  E_Ki(M'): a multi-layer encrypted message received by N_i.
  ps: K_i: the symmetric key of N_i corresponding to its SN.
Output:
  E_Kj(M'') or RH- E_Kj(M'')
Begin
  Successfully decrypt E_Ki(M') to E_Kj(M'') with K_i ;
  IF Don't get the exposed data then
    forward E_Kj(M'') to N_j;
    IF Fail forwarding with E_Kj(M'') then
        Label a RH-tag ;
        Redirect RH- E_Kj(M'') to its master head;
    End IF
  ELSE
        Obtain the exposed data ;
        (Finish transferring from the source to the destination)
  End IF
End
```

Algorithm 1: The algorithm of a normal participant

The second algorithm is more complex than the first one. The algorithm takes a cluster head (***CH***) as a reciver accepting an input, possibly representing three types, the so-called $E_K(M')$, $RH\text{-}E_K(M')$, and $BN\text{-}E_K(M')$. A ***CH*** is either a *special participant* accepting the $E_K(M')$ or just a cluster head accepting $RH\text{-}E_K(M')$ from one of its members or adjacent cluster heads or $BN\text{-}E_K(M')$ from adjacent cluster heads. Other assumptions are that $N_i$ is the current participant, $N_j$ is the next participant and disappeared in the routing path, $N_k$ is the next-next participant, and $S'$ is a set of the adjacent cluster heads of the *CH*. We writes three procedures with three kinds of input message in the second algorithm.

With the $E_{K_i}(M')$ in the first case, this procedure is similar to the first algorithm. The ***CH*** representing a current *special participant*, $N_i$, receives the message, removes one encryption layer, and forwards the remainder to $N_j$. This algorithm differs from the first one since the forwarding is unsuccessful. After the failure forwarding is happened, $N_i$ will directly *unlock* $E_{K_j}(M'')$ to $E_{K_k}(M''')$, removing the *j*th participant's encryption layer without the demand of a RH-tag because $N_j$ is a member of $N_i$. Then it labels $E_{K_k}(M''')$ with a BN-tag to broadcast and route to each of $S'$ for finding $N_k$. The result of this procedure is to produce an output, $BN\text{-}EK_k(M)$, due to failure forwarding to $N_j$.

```
A cluster head (CH) receives an input;
Input:
  E_Ki(M), RH-E_Kj(M), or BN-E_Kk(M)
  PS: K_i: the symmetric key of N_i.
  N_i is the current participant,
  N_j as the next participant is disappeared in the routing path,
  and N_k is the next-next participant.
Begin
  Case1: the E_Ki(M'), forwarded from the former participant
  Successfully decrypt E_Ki(M') to E_Kj(M'') with K_i ;
    IF Don't get the exposed data then
      forward E_Kj(M'') to N_j ;
      IF Fail forwarding with E_Kj(M'') then
        Unlock E_Kj(M'') to E_Kk(M''') ;
        Label E_Kk(M''') with a BN-tag.;
        Broadcast BN- E_Kk(M''') and Route to each of S';
      End IF
    ELSE
        Obtain the exposed data ;
        (Finish transferring from the source to the destination)
    End IF
 Case2: RH- E_Kj(M), forwarded from one of its members or one of S' ;
        Try to unlock E_Kj(M);
        IF Successful unlock then
          IF N_k is a member then
                Forward E_Kk(M') to N_k ;
          ELSE
                Label a BN-tag;
                Route BN- E_Kk(M') to each of S' ;
          End IF
        ELSE
          Do nothing and Route RH-E_Kj(M') to each of S' ;
        End IF
 Case3: BN-E_Kk(M), forwarded from one of S' ;
        IF N_k is a member or self then
          Rip BN-tag and Send E_Kk(M) to N_k;
        ELSE
          Abandon BN- E_Kk(M);
        End IF
End
```

Algorithm 2: The algorithm of a *CH*

With $RH\text{-}E_{K_j}(M)$ in the second case, the *CH*, not a participant, is one of cluster heads of $N_i$ to try to *unlock*. If $N_j$ is a member, the *CH* will successfully *unlock* and rip(remove) the RH-tag. And based on the remainder, $E_{K_k}(M')$, it check again if $N_k$ is a member in this cluster. Finally, $E_{K_k}(M')$ could be forwaded to $N_k$, or labeled with a BN-tag to route to each of $S'$. If the *CH* unsuccessfully *unlocks* $RH\text{-}E_{Kj}(M')$, doing nothing and only routing it to each of $S'$.

In the third case, the *CH* receives the $BN\text{-}E_{K_k}(M)$ from a adjacent cluster head, indicating the *k*th participant is a member in its cluster. This case is only happened after successfully *unlocking* $E_{K_j}(M)$. If $N_k$ is not a member of the CH, the receipt will be abandoned. Therefore, a cluster head representing a receiver has different procedures with three possible

different inputs in the second algorithm.

### 4.2 Various States Of A Transferring Message

In our algorithms, a transferring message between two nodes could have four kind appearances, ***M***, ***$E_K(M')$***, ***RH-$E_K(M')$***, and ***BN-$E_K(M')$***. They could be switched each other during the period of transmission. We describes these relationships between four types.

***M*** → ***$E_K(M')$*** : The source node starts to transfer data, encrypted multiply with symmetric keys of all participants of a shortest routing path, to the destination node at the beginning of a transferring message.

**One *$E_K(M')$* → Another *$E_K(M')$*** : A multi-layer encrypted message must pass all participants before it arrives to the destination. By receiving one multi-layer encrypted message, a participant forwards the remainder to next participant after skining one encryption layer on the message.

***$E_K(M')$*** → ***RH-$E_K(M')$*** : When *a normal participant* has a fail forwarding, a RH-tag may be added into the transferring message.

***RH-$E_K(M')$*** → ***RH-$E_K(M')$*** : A cluster head is not able to *unlock* a RH-tag multi-layer encrypted message instead of routing the changeless message to adjacent other cluster heads.

***RH-$E_K(M')$*** → ***BN-$E_K(M')$*** : A cluster head receiving a RH-tag message can successfully *unlock* for solving the problem about sudden disappearance of next participant. Then, without the location of next-next participant, a BN-tag takes the place of the RH-tag. The BN-tag message will be routed to to adjacent cluster heads.

***RH-$E_K(M')$*** → ***$E_K(M')$*** : A cluster head, receiving the RH-tag encrypted message, *unlocks* successfully. And it also finds the next-next participant. Then it rips the RH tag and routes the message to next-next participant. This translation could be happened by a cluster head on successful *unlock* and knowledge of next-next participants' location.

***$E_K(M')$*** → ***BN-$E_K(M')$*** : *A special participant* may unsuccessfully forward a multi-layer encrypted message to next participant. Then it uses the symmetric key which has shared with next participant to decrypt one layer of the message. Then, it will label a BN-tag on the remainding message and route to adjacent cluster heads.

***BN-$E_K(M')$*** → ***$E_K(M')$*** : The transalation is happened when a cluster head, receiving a BN-tag message, only rips the BN-tag to send the remainder to its member or self.

***$E_K(M')$*** → ***M*** : *A participant* is the destination and decrypts the last encryption to obtain exposed data. Hence, This translation is happened at the ending of a transferring message.

## 5. Characteristics of Algorithm

### 5.1 Non Source-Based Routing

In this paper, before proceeding anonymous communication, a routing path from a source to a destination must be established without a global view of the ad hoc network topology. Moreover, the proposal gathers routing information by forwarding the *path discovery* and *path reverse* messages and eliminate the load of managing routing centrally.

But this protocol has some disadvantages. Due to no control from source routing, source can not select the routing path based on certain criteria. And the source node can not limit the maximum number of participants on the routing path to avoid the delay for real-time interactive applications. This protocol is only applied to an ad hoc network consisting of hundred nodes, not thousand nodes.

### 5.2 Resilient Against Path Hijacking

If a node in ad hoc network accepts a *path discovery* message, it appends some secretly information and broadcasts to all adjacent nodes with one hop distance. An attack called "path hijacking" [4] can be happened. A malicious node, not authorized by any cluster head, might attend the establishment of the routing path to forward the message only to other proximity malicious nodes, resulting in a path with only malicious nodes.

The protocol can demonstrate to be resilient against path hijacking, through the source node successfully receives the *path reverse* message. The *path discovery* message will never reach the destination to not trigger the production of the *path reverse* message. Truly, the actual hijacking does not take place, because other *path discovery* messages might still arrive at the destination node and trigger a well *path reverse* phase. And the partial path hijacking can be caused that several malicious nodes have joined successfully to the establishment of the routing path. But it still does not threaten the anonymity of the data traffic. In this paper, we aussme all cluster heads have enough certificates not to be malicious nodes.

### 5.3 Reliable forwarding

In our approach, the source node encrypts multiply a message with symmetric keys of all participants of routing path according to message forwarding of participants in the opposite sequence. In addition the multi-layer encrypted message involves participants' ID distributed and located on each layer, so each participant receives the message to be able to decrypt its layer and get next participant ID to forward the remainding message. During transferring the message, a link state between two participants can be broken or next participant can be

out of the routing path so that the current participant can't forward the message to next participant. But the other participants, except next participant, still stay in the routing path. Our solution to this problem has two steps. First, a cluster head which next participant belongs to must be found for *unlocking* the encryption layer of next participant of the message. Second, the cluster head has to try to route the message to next-next participant. We carry out the soultion with our designing algorithms.

Because a cluster head is a leader in a cluster, it shares every member's symmetric key each other. It means that it can *unlock* any encryption of its members using their symmetric keys. Based on the characteristic of the one-level and one-hop cluster-based ad hoc network, the next participant is located in either the cluster or an adjacent cluster of the current participant in a routing path. By the way, the next-next participant is located in either the cluster or an adjacent cluster of next participant. According to our algorithms, a multi-layer encrypted message from the source could still been transferred continuously and reliably to the destination along the original path, even if one participant was out of the routing path.

Due to common security, $K_{CA}$ and $SN_{CA}$, on using by all cluster heads, other cluster head will *unlock* the message, even though a cluster head participating inside is out of the path.

## 6. Conclusion

In this paper, anonymous communications and reliability are two key points about end-to-end communications in the cluster-based ad hoc network. We also design our key distribution and the distributed CA based on [14]. Each cluster head authorizes and manages all symmetric keys of its members for the establishment of routing paths. We adopt the [4] about constructing a routing path for anonymous communications without catching the network topology and the related information about participants. Then we propose additional reliable protocols for recovering the original path to carry on the anonymous communication, after a participant can't be visible in the path. The advantage of our approach includes no-source based routing, anonymity of the source and destination, resilient against path hijacking, and reliable forwarding

Some disadvantages should be mended in the future, including the inability to avoid failed or malicious nodes during routing discovery, inefficiency about the recovery of the broken routing path due to the left of continuous two or more participants, and the demand of great computation capability and storage space of each node, cluster heads especially. Then, the paper expects few frequency of the cluster head revocation to reduce the influence of cluster control structures. So we prefer the selection of cluster heads appointed in [1], although it is insufficient for our requirements. .

## References


[1] E.M. Belding-Royer, Multi-Level Hierarchies for Scalable Ad hoc Routing, in: Wireless Networks 9, 461–478, 2003.

[2] D.J. Baker and A. Ephremides, The architectural organization of a mobile radio network via a distributed algorithm, IEEE Transactions on Communications 29(11) (1981) 1694–1701.

[3] J. Sharony. An Architecture for Mobile Radio Networks with Dynamically Changing Topology Using Virtual Subnets. ACM Mobile Networks and Applications (MONET) 1(1): 75-86, August 1996.

[4] Khalil El-Khatib, Larry Korba, Ronggong Song, and George Yee. Security Dynamic Distributed Routing Algorithm for Ad Hoc Wireless Networks, the International Conference on Parallel Processing (ICCP), October 6-9, 2003

[5] D. Johnson and D. Maltz: Dynamic source routing in ad hoc networks. In T. Imielinski and H. Korth, editors, mobile computing. Kluwer Academic,1996.

[6] [Perkins 1997] C. Perkins: Ad hoc on demand distance vector (AODV) routing. http://www.ielt.org/internet-draft/draft-ietf-manet-aodv-00.txt, 1997. IETF Internet Draft

[7] J. Lundberg, Routing Security in Ad Hoc Networks, http://citeseer.nj.nec.com/400961.html

[8] Seungjoon Lee, Bohyung Han, Minho Shin Robust Routing in Wireless Ad Hoc Networks 2002 International

[9] Computation of Minimal Uniform Transmission Power in Ad Hoc Wireless Networks

[10] D. Chaum: The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability. Journal of Cryptography, vol. 1, no. 1, pages 65-75, 1988.

[11] M. K. Reiter and A. D. Rubin: Crowds: Anonymity for Web Transactions. ACM Transactions on Information and System Security, vol. 1, no. 1, pages 66-92, Nov. 1998.

[12] D. Chaum: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. Communications of the ACM, vol. 24, no. 2, pages 84-88, Feb. 1981.

[13] M. Reed, P. Syverson, and D. Goldschlag: Proxies for anonymous routing. In 12th Annual Computer Security Applications Conference, pages 95-104. IEEE, Dec. 1995.

[14] M. Bechler, H.-J. Hof, D. Kraft, F. Pahlke, L. Wolf : A Cluster-Based Security Architecture for Ad Hoc Networks. the IEEE Communications Society conference, 2004.