AKE in Clustered Ad Hoc Networks

Maarit Hietalahti

December 18, 2006

Abstract

Efficient authenticated group key establishment is the pre-requirement for having group-wide encrypted communications in wireless ad hoc networks. Clustering has brought scalability to ad hoc networks in many ways, now we look at its benefits to group key agreement. Several solutions for authenticated key establishment (AKE) in clustered ad hoc networks are surveyed. A new solution for clustered AKE is presented, one that is based on AT-GDH and the broadcast group key protocol. This new protocol is found to be very efficient with a communication complexity logarithmic to the number of clusters.

1 Introduction

Constructing group wide keys in a large ad hoc network is a complicated task that may be unachievable due to the dynamic nature of ad hoc networks. Splitting the problem to pieces, clustering the network, is a usual solution suggested for routing already in [1]. Clusters are supposed to have more stable internal connections due to the greater amount of links between nodes in a same cluster. Therefore, contributory group keys are easier to establish and manage inside clusters. On the other hand, clusters are assumed to stay together longer than the nodes do in average, which makes inter-cluster key agreement more sensible. Thus, clustering may bring the necessary scalability into key establishment in very large networks.

In ad hoc networks, every pair of nodes cannot reach each other within one hop. This issue of restricted topology, what H. Shi and M. He [2] call *the neighbors communication problem* can be solved with little help from graph theory. A key agreement protocol (AT-GDH) using this method has been presented already in [3] and an extension with clustering will be presented in this paper.

Rest of this paper is organized as follows. Section 2 explains the concepts of clustering and hierarchy, along with (non-clustered) group key agreement and

some examples. Section 3 briefly describes the circumstances of an ad hoc network environment and their effect to the task. Section 3.1 lists security requirements for a group key agreement, Section 4 goes through some existing clustered group key establishment methods and Section 5 presents the new clustered group key agreement protocol. Section 6 concludes the paper and sketches some lines for future work.

2 Background

2.1 Clustering and hierarchical routing

A cluster is a collection of nodes (geometrically) close together. Clusters can be formed deliberately for a common cause or they can form as a reaction to a factor that is common to the nodes. A cluster-head is a special node in a cluster that acts as a leader for the cluster, for the purposes of routing or initializing the cluster formation, for example. Cluster-heads are not always necessary, some clustering protocols do not use them at all.

A hierarchical structure in a network is composed of nested groupings (clusterings) of nodes, forming a tree topology. Hierarchical structures are often used in routing. A route from a leaf node to another is formed via the routes between their respective groups. One of the first papers describing hierarchical routing (in a fixed network) is "*Hierarchical routing for large networks; performance evaluation and optimization*" [1], where optimal clustering structures are determined so as to minimize the size of the routing tables. The price for this is the increase in the average message path. However, bounds were found for the maximum increase of path length, so that in the limit of a very large network, enormous table size reduction may be achieved with essentially no increase in network path length. The performance of the proposed hierarchical routing system was evaluated in [4].

A *two-tier ad hoc network* means a hierarchical network consisting of only two layers. In other words, the nodes are clustered in some way, and the clusters can have cluster-heads, but there are no nested clusterings. Two-tier networks are most common hierarchical structures in the literature, as several layers of hierarchies are expected to waste too many usable paths.

In order to arrange into clusters, the nodes need to run a clustering algorithm. Many algorithms need the knowledge of the whole network topology, while others perform the computations knowing only the neighboring nodes and their possible cluster-memberships [5, 6].

Clustering algorithms differ in what types of clusters they produce. Many clustering algorithms choose special nodes, cluster-heads, that take care of the cluster formation and later of the maintenance of the cluster [5, 7, 8, 9, 10, 11].

Some clustering algorithms form cliques, i.e., clusters where every node is at a one hop distance from every other node [12]. In Figure 1, a network is clustered into cliques. Some only require that the distance to the cluster-head is one hop [9, 10, 11].

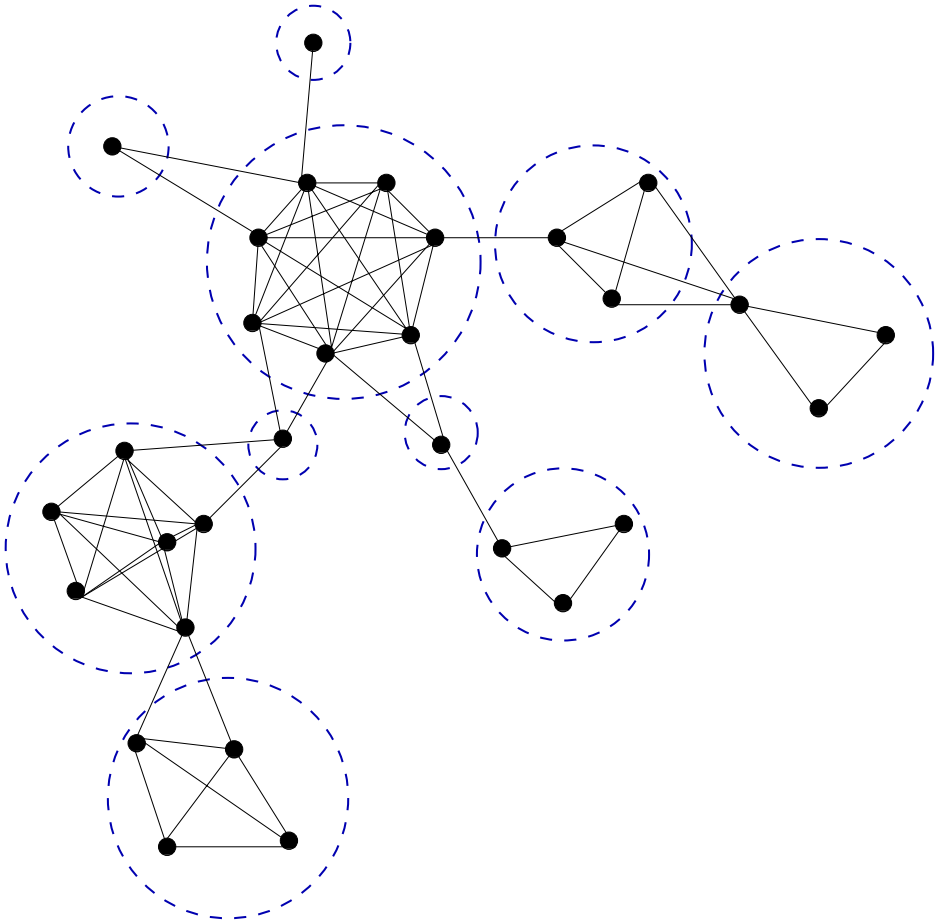

Figure 1: An example of clustering

The role of a cluster-head varies in different protocols. In [13] the cluster-heads are not used as routers, but merely for pointing the direction of the cluster. In [6, 14, 11] the cluster-heads are only used in the cluster formation, but not in routing. Gateways are border nodes that relay messages from one cluster to another. If clusters are allowed to overlap, a gateway usually belongs to more than one cluster.

The clusters are assumed to stay together longer than the nodes do in average. Clusters are supposed to have more stable internal connections due to the greater amount of links between nodes in a same cluster. When clusters form as a result of some common background, they are likely to have a lot of internal communications as well. However, when nodes are no longer equally important in maintaining connectivity, a well-connected node (cluster-head) may become a single point of failure, a target for attacks aimed at cutting down the cluster's connections to other clusters. Very deep hierarchies can reduce the amount of routes, which also leads to single weak points in the network.

More information on clusterings can be found, for example, in [15].

2.2 Group key agreement

The purpose of key establishment is to create a common key for a group of two or more participants to be used for encryption and authentication of their communications. For two participants, the *Diffie-Hellman key exchange* is often the most convenient choice. The multi-party case requires a generalization of a two-way key exchange.

There are *distributory* and *contributory* group key protocols. A contributory protocol means that all participants take part in the key generation and guarantee for their part that the resulting key is fresh. Key distribution, on the other hand, means that the key is generated by one party and distributed to the other participants. This cannot be done without the help of a previously agreed-on secret that is used in encrypting the new session key. There is also a method called key pre-distribution, whereby the key is completely determined by the previously agreed-on initial key material.

An undirected graph is a *tree*, if it does not contain any cycles and is connected, i.e., a tree is a minimally connected graph. A rooted tree contains one special node that has no parents, the root. A rooted tree is a ternary tree if all of its branch nodes have 3 or less children. A binary tree's branches have 2 or less children.

2.3 Broadcast protocol

This protocol was presented by Burmester and Desmedt [16]. It assumes that every node is at a one hop distance from another. The protocol is accomplished with only two broadcasts per node.

G is a finite cyclic group and g is a generator of G.

1. Each node m_i selects a random exponent r_i and broadcasts $z_i = g^{r_i}$ 2. Each node m_i computes and broadcasts $x_i = (z_{i+1}/z_{i-1}1)^{r_i}$ 3. Each node computes the session key $k_i = z_{i-1}^{nr_i} x_i^{n-1} x_{i+1}^{n-2} \cdots x_{i+n-2}$.

2.4 TGDH

TGDH [17] employs Diffie-Hellman key exchanges in binary key trees. The described structure of the results from the dynamic group key operations such as join, leave, merge and partition. There is no initial key agreement protocol.

The key structure in TGDH is very general, it can be used to describe the key structure of any bipartite group Diffie-Hellman key agreement where the resulting keys are used recursively as the new exponents. For example, the key structure of the protocol Hypercube [18] is the same as that of TGDH with perfect binary tree where all leaves are at the bottom level.

The key structure of TGDH The nodes are denoted $\langle l, v\rangle$, which means the v-th node at level l in a tree. Each node $\langle l, v\rangle$ is associated with the key $K_{\langle l,v\rangle}$ and the blinded key $BK_{\langle l,v\rangle} = f(K_{\langle l,v\rangle})$ where the function $f()$ is modular exponentiation in prime order groups, analogous to the Diffie-Hellman protocol. α is the exponentiation base, p and q prime integers. M_i is the i-th group member.

The keys are computed recursively as follows:

$$\begin{aligned} K_{\langle l,v\rangle} &= (BK_{\langle l+1,2v+1\rangle})^{K_{\langle l+1,2v\rangle}} \bmod p \\ &= (BK_{\langle l+1,2v\rangle})^{K_{\langle l+1,2v+1\rangle}} \bmod p \\ &= \alpha^{K_{\langle l+1,2v\rangle} K_{\langle l+1,2v+1\rangle}} \bmod p \\ &= f(K_{\langle l+1,2v\rangle} K_{\langle l+1,2v+1\rangle}) \end{aligned}$$

The resulting group secret is $K_{\langle 0,0\rangle}$, the root's key.

[19] extend TGDH (Tree-based Group Diffie-Hellman) protocol to improve the computational efficiency by utilizing pairing-based cryptography. They use bilinear pairings in a ternary key tree which applies to any two-party and three-party key agreement protocol.

2.5 AT-GDH

AT-GDH (Arbitrary Topology Generalization of Diffie-Hellman) employs a spanning tree. A spanning tree contains only the (one hop) links used in initial key agreement. This avoids the neighbors communication problem, as the Diffie-Hellman key exchanges are done only with one-hop neighbors. The operations propagate over the network along the spanning tree.

A spanning tree can be done in several ways, literature on multicast tree construction and network flooding contain applicable solutions. Below we will describe one possible protocol for constructing a spanning tree where the node initiating the protocol becomes the root. The key agreement can begin right after the spanning tree is ready.

In the initial state it is assumed that the nodes know their neighbors and that all links are two-way. The initiator sends a message to each of its neighbors. It thereby becomes the root of the spanning tree and its neighbors become its children. After receiving a message, a node acknowledges it and sends a similar message to all its neighbors, except to the parent. The nodes that acknowledge a message from a node become its children in the tree. When a node gets more than one of these messages, it acknowledges and processes only the message that it receives first, and consequent messages are ignored. This continues until every node has received a message. A leaf is a node that does not receive acknowledgments from any of its neighbors. The spanning tree has now been constructed and all nodes know their parent and their children.

All leaf nodes (nodes with no children) start by selecting a random secret exponent and blind it and send the result to their respective parents (See Figure 2) After a node has received the blinded keys from all its children, they select their exponents and form Diffie-Hellman-type keys with their children repeatedly using the resulting key as the new exponent. The nodes do not send these keys to the children yet. The secret formed with the last child serves as the node's new private key, which the node blinds and sends to its parent. When this parent has received similar messages from all its children, it can repeat the same computation. This continues until the root has received all the blinded keys of its children. The root repeats the same kind of computation as all the other parent nodes. The secret key formed thus between the root and its last child (and all other nodes) will be the shared session key material for the entire network. In the last phase of the protocol, the blinded keys needed for extracting the group key are propagated up the tree from the parents to their children starting from the root.

Key Agreement Protocol for an Arbitrary Tree:

Initialisation: Let G be a finite cyclic group of order q, and let α be a generator of G. The participants are assumed to pick their secret exponents randomly from $\mathbb{Z}_q$. It is also assumed that there exists a bijection $\varphi : G \longrightarrow \mathbb{Z}_q$. Here the participants are identified with their universal address in the tree.

Phase 1

Round 1 For all nodes $x = y.i$ with $c_x = 0$

1. x selects a random $k_x \in \mathbb{Z}_q$
2. $x \rightarrow y : \alpha^{k_x}$

Rounds 2 . . . h For all nodes x with $c_x \neq 0$

1. x selects a random $e_x \in G$
2. x waits to receive $\alpha^{k_{x.j}}$ for all $j = 1, \ldots, c_x$
3. x calculates $k_x = \varphi(K(x, c_x))$ from

$$\begin{aligned} K(x,0) &= e_x \\ K(x,j) &= \alpha^{k_{x.j}\varphi(K(x,j-1))} \text{ for } j = 1, \ldots, c_x \end{aligned}$$

4. $x \rightarrow y : \alpha^{k_x}$

Phase 2

Rounds $h + l$ $l = 1, \ldots, h$ For every node $x.i$ on level l, $x \rightarrow x.i$: $M_{x.i}$, where

$$M_{x.i} = \langle M_x, \alpha^{\varphi(K(x,i-1))}, \alpha^{k_{x.(i+1)}}, \alpha^{k_{x.(i+2)}}, \alpha^{k_{x.c_x}} \rangle$$

with M_ϵ being empty.
The resulting common key is $K(\epsilon, c_\epsilon) = k_\epsilon$

AT-GDH does not contain group key management mechanisms, or authenticate the resulting key explicitly. AT-GDH can be used in any connected network topology with bidirectional links, because a spanning tree can always be constructed in such a network.

The number of synchronous rounds AT-GDH needs to gather and distribute the blinded keys is twice the height of the tree. The height of the tree is usually

logarithmic to the number of nodes in the network, depending on the spanning tree algorithm and the topology of the underlying network links.

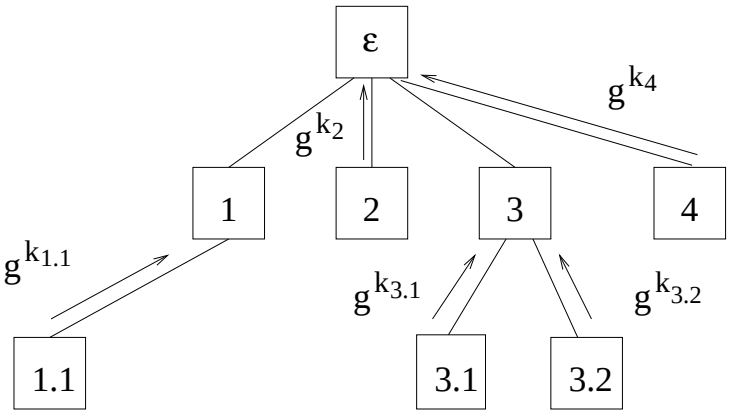

Figure 2: AT-GDH step 1.

3 The challenges of group key establishment in ad hoc network environment

The limitations of ad-hoc network environment pose some drastic demands on the group key establishment protocols. First, a global broadcast is most probably out of the question, that is, it is not probable that an arbitrary node will have direct connections to all other participant nodes. But on some occasions, a local broadcast from a node to its neighbors is feasible. Also, no fixed topology, such as a ring or a star can be assumed. Consequently, protocols requiring a specific topology either cannot be used at all or become inefficient.

In other words, every pair of nodes cannot reach each other within one hop. The issue, what H. Shi and M. He call *the neighbors communication problem* can be solved with the help of graph theory. A method has been presented already in [3] and a revision of it will be presented later in this paper.

The lack of infrastructure means that there are initially no third parties that can be trusted to calculate a random key safely and to distribute it. A lack of common history implies the lack of previously agreed shared secrets.

3.1 Requirements for group key establishment

In the context of group key exchange, implicit key authentication means that a principal can be sure that no-one outside the group can learn the key without the help of a dishonest participant. Key confirmation means that after the key has been established, the participants are assured that all legitimate participants do share the same key. As this would require many all-to-all messages, which may not even be possible in a sparse connection ad-hoc network, achieving key confirmation is not practical. Explicit Key Authentication means that both implicit key authentication and key confirmation hold, i.e., all legitimate participants know the key and no outsiders do.

An active adversary should not be able to mislead honest participants as to the final outcome. A compromise of past session keys should not allow a passive adversary to find out future session keys and should not allow impersonation by an active adversary in the future. Independence of long term and short term secrets is important when there is an additional long term secret present, for example, private keys of a public-key algorithm or passwords used in authentication.

4 Existing AKE schemes for clustered ad hoc networks

Clusters, other groups and the whole network may have needs for common session keys for efficient encrypted communication inside the group, for example, to securely broadcast a message. Group key systems are useful for this too: a common symmetric key can be used in encrypting a message meant for all nodes inside a group.

A generic model for key establishment in clustered ad hoc networks works along these lines: First, nodes form clusters with some clustering method. Then a key-tree is formed from the clusters, sometimes the tree extends inside clusters, sometimes the clusters are considered as single vertices in the tree. After this, the initial key agreement begins. Usually keys are established in subgraphs first, and then combined for a whole group wide key. The Diffie-Hellman key exchange (bipartite or tripartite) is typically used recursively as a basis for the group keying. A group key is constructed so that every node can calculate it using its own secret and the blinded secrets of others, or combinations of them. In some scenarios the messages are signed and key confirmation messages are sent for authentication purposes.

Rhee et al. [20] present an architecture for key management in hierarchical mobile ad hoc networks. They use implicitly certified public keys (ICPK) [21], an ID-based public key scheme where the public key of each participant is derived

from its identity. It provides computationally efficient implicit authentication. A key confirmation message added to the key agreement protocol makes the protocol explicitly authenticated. A two layered hierarchy is prompted by a physically two-layered network, ground nodes and unmanned aerial vehicles. The layers use different key management methods, the clusters of nodes below use a centralized system, while the aerial vehicles use TGDH. The centralized system inside clusters is not contributory.

Another hierarchical key agreement is proposed in [22]. This is a multilevel hierarchy, where a node can have several cluster keys according to the cluster and its superclusters it belongs to. However, it is not completely contributory. Keys are agreed among cluster-heads on the same level and then distributed to their respective clusters.

Hybrid key management [23] propose a clustered key establishment, where each cluster selects a cluster-head that makes a key agreement with other cluster-heads. After that, the cluster-head distributes the key to the cluster. Thus, other nodes in the cluster do not contribute to the key. Clustering is made according to the geometric locations of the nodes. The key agreement used can be any group key agreement protocol, for example GDH [24].

A cluster-tree-based group key agreement ACEKA is presented in [2]. ACEKA uses ternary trees with the Joux tripartite Diffie-Hellman key agreement [25]. There is a virtual backbone and virtual nodes in addition to the real nodes. ACEKA uses cluster-heads and "sponsors" for management. Authentication by signing every message using ID-based cryptography, with a variant of the ElGamal signature scheme.

5 Clustered AT-GDH

First, the network is divided into clusters with a clustering mechanism that creates very stable clusters. Nodes in a cluster are at a one hop distance from each other, i.e., cliques. In this kind of a cluster, the most efficient group key agreement protocol is the broadcast protocol by Burmester and Desmedt explained before. It takes only two rounds of broadcasts, after which each node can calculate the common group key from its own secret exponent and the blinded shares of others.

When every cluster has a common secret key, the clusters agree a group key by AT-GDH protocol. Cluster-head can represent its cluster and use the cluster key as its secret exponent. After the AT-GDH protocol run, cluster-heads distribute the needed key parts also in their cluster, so that other nodes can also calculate the network wide group key. The structure of the resulting cluster-tree is presented in Figure 3.

Now that cluster-heads are not necessarily at a one hop distance from each

other, the messages need to be relayed. The gateways relaying the messages are members of a cluster, and know the cluster secret already. However, it affects the communication complexity by adding extra links to the path.

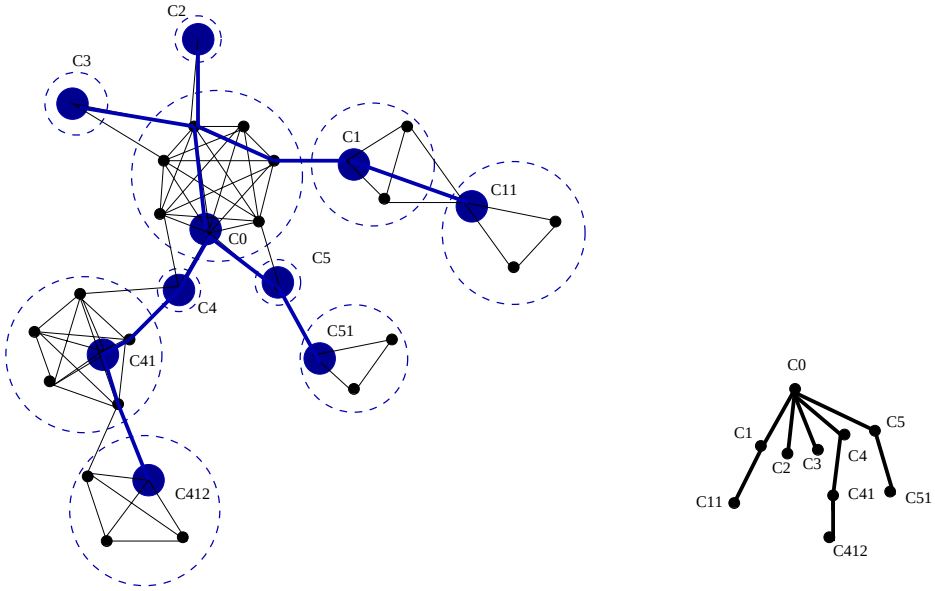

Figure 3: Clustered AT-GDH

5.1 Efficiency

This clustered group key agreement is efficient, because radio connections can easily create large cliques. Every clique forms a group key in two rounds, i.e., constant amount of rounds. The amount of AT-GDH synchronous rounds is now logarithmic to the number of clusters. In the end, cluster-heads broadcast the key parts in one round. The resulting communication complexity is logarithmic to the number of clusters.

5.2 Authentication

Previously group key agreements, like the authenticated GDH, A-GDH, relied much on the implicit key authentication. The group key can not be constructed without the secret share of one of the participants. However, Pereira and Quisquater [26] showed that it is impossible to design a scalable authenticated group key agreement protocol on the same building blocks as A-GDH. Hence other authentication

methods are needed. Authentication with ID-based crypto, such as the ICPK public keys with key confirmation messages could be used here, as it is independent of the group key establishment method used.

6 Conclusions and future work

Clustering is a versatile solution in ad hoc networks, its benefits can be seen in routing and other operations requiring efficient gathering and propagation of information among the network. It was seen here that clustering can also help in creating a symmetric group key for fast encrypted communications. Some existing solutions for clustered group key establishment were surveyed and a new protocol was proposed. The cluster-based extension of AT-GDH combined to the broadcast group key protocol turned out to be very efficient, the number of rounds was found to be logarithmic to the number of clusters.

Clustered AT-GDH could be more efficient with tripartite key exchange realized with bilinear pairings as in [19]. The form of the tree and clusters also affects the efficiency of the group key establishment. However this needs more research.

In an ad hoc network where nodes are mobile, a mere group key establishment is not always enough. The group key needs maintenance. At least the key should be updated when nodes join or leave the network, to preserve its contributory property. Neither AT-GDH or the clustered extension proposed here have group key maintenance which is outside the scope of this paper and left for future work.

References

[1] L. Kleinrock and F. Kamoun. Hierarchical routing for large networks; performance evaluation and optimization. *Computer Networks*, 1:155–174, 1977.

[2] H. Shi and M. He. Authenticated and communication efficient group key agreement for ad hoc networks. In *To appear at CANS '06*, December 2006.

[3] M. Hietalahti. Efficient key agreement for ad hoc networks. Master's thesis, Helsinki University of Technology, Department of Computer Science and Engineering, Espoo, Finland, May 2001.

[4] F. Kamoun and L. Kleinrock. Stochastic performance evaluation of hierarchical routing for large networks. *Computer Networks*, 3:337–353, November 1979.

[5] S. Basagni. Distributed clustering for ad hoc networks. In *Proceedings of the International Symposium on Parallel Architectures, Algorithms, and Networks (ISPAN)*, pages 310–315, Perth, Australia, June 1999. IEEE.

[6] S. E. Virtanen and P. Nikander. Local clustering for hierarchical ad hoc networks. In *Proceedings of WiOpt: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pages 404–405, Cambridge, UK, March 2004. IEEE.

[7] P. Basu, N. Khan, and T. D. C. Little. Mobility based metric for clustering in mobile ad hoc networks. In *Proceedings of the International Workshop on Wireless Networks and Mobile Computing (WNMC)*, pages 16–19, 2001.

[8] M. Bechler, H.-J. Hof, D. Kraft, F. Phälke, and L. Wolf. A cluster–based security architecture for ad hoc networks. In *Proceedings of the IEEE Conference of the Computer and Communications Societies INFOCOM*, Hong Kong, China, March 2004. IEEE.

[9] T.-C. Hou and T.-J. Tsai. An access–based clustering protocol for multihop wireless ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 19(49):1201–1210, July 2001.

[10] A. Ephremides, J. E. Wieselthier, and D. J. Baker. A design concept for reliable mobile radio networks with frequency hopping signaling. volume 75, pages 56–73. IEEE, IEEE Press, January 1987.

[11] M. Gerla and J. T.-C. Tsai. Multicluster, mobile multimedia radio network. *Wireless Networks*, 1:255–265, 1995.

[12] P. Krishna, N. H. Vaidya, M. Chatterjee, and D. K. Pradhan. A cluster–based approach for routing in dynamic networks. *ACM SIGCOMM Computer Communication Review*, pages 49–65, April 1997.

[13] J. Sucec and I. Marsic. Clustering overhead for hierarchical routing in mobile ad hoc networks. In *Proceedings of the IEEE Conference of the Computer and Communications Societies INFOCOM*, volume 3, New York, USA, June 2002. IEEE Press.

[14] C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 15(7):1265–1275, 1997.

[15] M. Steenstrup. Cluster–based networks. In C. E. Perkins, editor, *Ad Hoc Networks*, pages 75–138. Addison Wesley, 2001.

[16] M. Burmester and Y. Desmedt. A secure and efficient conference key distribution system. In *Advances in Cryptology –Proceedings of EUROCRYPT*, volume 950 of *LNCS*, pages 275–286, Perugia, Italy, May 1994. Springer.

[17] Y. Kim, A. Perrig, and G. Tsudik. Simple and fault–tolerant key agreement for dynamic collaborative groups. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, pages 235–244, Athens, Greece, November 2000. ACM.

[18] K. Becker and U. Wille. Communication complexity of group key distribution. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, pages 1–6, San Francisco, CA, USA, November 1998. ACM Press.

[19] S. Lee, Y. Kim, K. Kim, and D. H. Ruy. An efficient tree–based group key agreement using bilinear map. In *Applied Cryptography and Network Security ACNS*, volume 2486 of *LNCS*, pages 357–371. Springer, 2003.

[20] K. H. Rhee, Y. H. Park, and G. Tsudik. A group key management architecture for mobile ad-hoc wireless networks. *Journal of Information Science and Engineering*, 21:415–428, 2005.

[21] C. Günther. An identity–based key exchange protocol. In *Advances in Cryptology –Proceedings of EUROCRYPT*, volume 434 of *LNCS*, pages 29–37, 1989.

[22] G. Yao, K. Ren, F. Bao, R. H. Deng, and D. Feng. Making the key agreement protocol in mobile ad hoc network more efficient. In *Applied Cryptography and Network Security ACNS*, volume 2846 of *LNCS*, pages 343–356. Springer, 2003.

[23] X. Li, Y.Wan, and O. Frieder. Efficient hybrid key agreement protocol for wireless ad hoc networks. In *Proceedings of the IEEE International Conference on Computer Communications and Networks*, 2000.

[24] M. Steiner, G. Tsudik, and M. Waidner. Diffie–hellman key distribution extended to group communication. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, pages 31–37, New Delhi, India, March 1996. ACM, ACM Press.

[25] A. Joux. A one round protocol for tripartite Diffie–Hellman. In *Proceedings of Algorithmic Number Theory Symposium IV*, volume 1838 of *LNCS*, pages 385–394. Springer, 2000.

[26] O. Pereira and J.-J. Quisquater. Generic insecurity of cliques-type authenticated group key agreement protocols. In *Proceedings of the IEEE Computer Security Foundations Workshop (CSFW)*. IEEE Computer Society Press, June 2004.