Authenticated Secure Communications in Wireless Networks[1]

A. Abdel-Hafez[†], A. Miri[†] , and L. Orozco-Barbosa [†,‡]

[†]School of Information Technology and Engineering
University of Ottawa, Ottawa, Ontario, K1N 6N5, Canada
E-mail:{ahafez, samiri, lorozco}@site.uottawa.ca
‡Universidad de Castilla La Mancha, Department of Computer Engineering
Campus Universitario s/n, 02071 Albacete, SPAIN
E-mail: lorozco@info-ab.uclm.es

***Abstract*: The lack of fixed infrastructure, both physical and organizational, and the highly dynamic nature of ad hoc wireless networks present major difficulties in the definition of the security mechanisms for such networks. The management of cryptographic keys in such spineless networks is a key area of research into security services of such networks. Traditional key management solutions that require a fixed infrastructure or centralized services are neither suited nor efficient for use in ad hoc networks. In this paper, we propose a fully distributed authenticated key agreement protocol based on the use of elliptic curve cryptosystems. The proposed protocol allows authorized group members to generate their session key without the need for a trusted third party. A modification of this protocol is also proposed to improve the protocol efficiency using a group clustering technique.**

1. Introduction

Wireless networks have become an integral part of all kinds of network infrastructure. Whether they are instituted as the means of communication between cellular phones or as the avenue of contact between military agents on the battleground, wireless is an indispensable medium of correspondence.

Wireless networks are implemented in two flavours, ad hoc and structured environment. In a structured network, nodes can use a fixed base stations to relay messages back and forth. This type of network is evinced in the cellular market, and it is easy to be supported by implementing a suitable mechanism responsible for providing security and reliability of that environment. Conversely, in an ad hoc network there is no a set infrastructure, and nodes must communicate by routing each other's messages. Such networks can be used, for instance, in the battlefield and rescue missions, where the nodes have little computational power and memory.

This lack of structure in ad hoc networks makes them more vulnerable to attacks than structured networks. This is in addition to the already existing weakness in wireless networks due to the use of radio waves as the common communication medium which make them easily accessible through the use of the right kind of radio. Consequently, it is crucial to secure the sensitive data manipulated through such networks. This is achieved by encrypting the message using a common private-key or public-key encryption protocols, of which the private-key encryption is generally faster. On the other hand key management is a well-known problem in private-key encryption, especially in such spineless environments.

Creating a common key for a group of communicators with all members contributing to the key is called *Group Key Agreement*. For very rapidly changing ad hoc networks, the exchange of encryption keys may have to be addressed on demand and without assumptions about a priori negotiated secret. There are many proposed key agreement protocols designed for wireless networks based on symmetric key cryptography. The reason for preferring symmetric key is that the wireless devices are not fully qualified to perform the heavy computations required in public-key cryptography. However, a symmetric-key solution requires an on-line trusted third party (TTP) to distribute the session keys. On the other hand, there are many public-key based protocols in the literature for group key establishment (see for example GDH.2 [2] Hypercube and Octopus [4]), but they have not been designed with the special nature of ad hoc networks in mind. Moreover, these protocols require many modular exponentiations, which is computational costly and may not be suitable with the current technology.

The purpose of this paper is to find a secure and efficient key agreement protocol for a group of communicating devices in an ad hoc wireless network. We present an authenticated key agreement protocol based on elliptic curve cryptographic techniques. The required computational processes (point multiplication) in elliptic curve techniques are much faster than those needed for other public-key techniques used.

The remainder of this paper is organized as follows. Section 2 overviews the attributes of wireless communications systems. In Section 3, we present the common threats to and vulnerabilities of wireless networks compared with those of wired networks. We review some previous related work in Section 4. In Section 5, we explain our authenticated group key agreement protocol. In Section 6, we explain the way we adopt the clustering techniques to extend our protocol and improve its efficiency. Our conclusions and suggested future work are given in Section 7.

[1]This work has been supported in part by an NSERC grant and by a grant CICYT TIC2000-1151-C07-02 of the Spanish government

2. Attributes of Wireless Communication Systems

One of the major challenges in ad hoc networks security is that ad hoc networks typically lack of a fixed infrastructure both in form of physical infrastructure such as routers, servers and stable communication links and in the form of an organizational or administrative infrastructure. Another difficulty lies in the highly dynamic nature of ad hoc networks since new nodes can join and leave the network at any time. The major problem in providing security services in such infrastructure-less networks lies on how to manage the cryptographic keys that are needed.

When designing protocols for ad hoc networks, whether routing protocols or security protocols, it is important to consider the characteristics of the network and realize that there are many "flavours" of ad hoc networks. Ad hoc wireless networks generally have the following characteristics [10]:

- Dynamic network topology: The network nodes are mobile and thus the topology of the network may change frequently. Nodes may move around within the network, the network can be partitioned into multiple smaller networks or be merged with other networks.
- Limited bandwidth: The use of wireless communication typically implies a lower bandwidth than that of traditional networks. This may limit the number and size of messages sent during protocol execution.
- Energy constrained nodes: Nodes in ad hoc networks will most often rely on batteries as their power source. The use of computationally complex algorithms may not be possible. This also exposes the nodes to a new type of denial of service attack, the sleep deprivation torture attack [10] that aims at depleting the nodes energy source.
- Limited physical security: The use of wireless communication and the exposure of the network nodes increases the possibility of attacks against the network. Due to the mobility of the nodes the risk of them being physically compromised by theft, loss or other means will probably be greater than that for traditional network nodes.

In many cases the nodes of ad hoc network may also have limited CPU performance and memory, e.g. low-end devices such as PDA's, cellular phones and embedded devices. As a result certain algorithms that are computationally or memory expensive might not be applicable.

3. Analysis of Threats and Vulnerabilities in Wireless Networks

In this section, we concentrate on wireless networks using the radio path as transmission medium. The basic security needs are similar to those those for wired networks. However problems related to these needs are stressed in wireless networks due to the use of the radio path. We can list some of the security threats related to wireless networks as follows:

- The passive eavesdropping is very easy in the radio environment, when one sends a message over the radio path, everyone equipped with a suitable transceiver in the range of the transmission can eavesdrop on the message. The sender or intended receiver has no means of knowing if the transmission has been overheard or not, so this kind of eavesdropping is absolutely undetectable. In wireless environment the ease of eavesdropping justifies quite costly procedures to guarantee the confidentiality of the network traffic.
- When we have a wireless network as a part of a wired large network, it offers one interface to the attacker, requiring no physical arrangements, to intrude on our network. This attack is known in the literature as *transitive trust attack*, since if the attacker can fool the wireless part of the network to trust the mobile he controls, then it is very difficult to prevent any hostile actions taken by the adversary. This makes the need for an efficient authentication mechanisms crucial for the security of wireless networks. In all cases all parties of the transmission should be able to authenticate each other.
- Due the nature of radio transmission, wireless networks are vulnerable to denial of service attacks. If the attacker has a powerful enough transceiver, he can easily generate radio interference so that the wireless network is unable to communicate using radio paths. Protection against this kind of attack is very difficult and expensive. The only complete solution is to have the wireless network physically secure, but this is applicable only in very rare cases. It is easy however, for authorities to locate the transceiver used to generate interference, so the attacker has limited time before the transceiver is found.

4. Related Work

Ad hoc wireless network security research often focuses on secure routing protocols, which form an essential component of ad hoc wireless network security. However, all such protocols assume that key distribution has taken place or in the best cases they are partially described, (see for example, SEAD [9], Ariadne [8], ARAN [6], SPINS [15]). Only recently there have been some tentatives to define the key distribution problem in ad hoc networks as discussed below.

In [12], Khalili *et al.* do not address clearly how the nodes involved would authenticate each other in the initialization stage of their proposal. There is an opportunity for impersonation attacks right from the beginning. Moreover, there is no concrete method given for

generating the master key (for both public and private master keys). Zhou and Hass [18] introduce the idea of distributing a certificate authority (CA) through the network in a threshold fashion, at the time of configuring the network. However, the authors do not address the resource limitations of devices in ad hoc networks as public-key and threshold cryptography are (in general) computationally expensive and need to be tailored to the resources and constraints of low-power devices. In [14], the authors propose acceleration techniques for the key establishment protocols using techniques that involve the assistance of a base station called Server-Aided Secret Computation (SASC). In their scheme, the SASC is responsible of exchanging information with the base station to get that all expensive computations be carried out by the server. In such protocols, a prior arrangement of the base station is required and restricts the independence of nodes in return for assistance from the base station. A password-based authenticated key exchange protocol has also been proposed by Askon and Ginzboorg in [1].

Our proposed protocol is based on an extension of the two-party authenticated key agreement to group communication, which will address some of these deficiencies. Group key establishment protocol in wired networks has been a very active research area, due to the popularity of group communications collaborative applications. In this area, there are many existing solutions for group key establishment. Most of these solutions propose different ways to extend the well known Diffie-Hellman key exchange protocol [7] to a multiparty key agreement, see for example [4, 11, 16, 17]. None of these protocols are suitable for ad hoc wireless networks mainly due to the fact that they demand that the network topology follow a prescribed structure. Moreover, the complexity of these protocols (communication and computation cost) is always of $O(n)$, where n is the number of participating members, which poses a scalability problem, especially with large group sizes. Some of these protocols also require global broadcast capabilities, which is not always possible in ad hoc networks.

5. Group LMQSV Protocol (GLMQSV)

In this section, we explain our proposed approach, which provides an authenticated key agreement protocol suitable for ad hoc wireless netowrks, GLMQSV protocol (see Algorithm 1). GLMQSV protocol can be considered as an extension to one of the recent authenticated two-party key agreement protocol by Law, Menezes, Qu, Solinas, and Vanstone (LQMSV), presented in [13]. In this protocol, we assume that the group members are distributed logically in a ring. In Section 6, we will define an extension to this protocol which will remove this requirement, and will improve the efficiency of the protocol. The main protocol consists of two stages: in the first stage, the contribution of all group members are collected, while in the second stage, the last member in the group (we will refer to it as a group leader) sends a unicast message to each of the group members allowing them to calculate a common session key. The most interesting property of this protocol is that the authentication property is achieved within the execution of the key agreement steps i.e., there is no need to use extra primitives or separate algorithms to provide authentication.

5.1. Protocol Description

In the following description (and subsequently), S_A and s_a are the $A's$ long-term public and private keys consequently, while R_A and r_a are the public and private ephemeral keys. We will describe the protocol considering four members in the group as an example, but the protocol can easily be extended to any group size. The group members will be M_1, M_2, M_3, M_4, where M_4 is the group leader. The first step consists of 3 rounds, in each round each member calculates its public share using its long-term and ephemeral secret key. For example

$$M_1 \rightarrow M_2 : r_1P$$

$$M_2 \rightarrow M_3 : r_1r_2P, s_1s_2P, r_2P, r_1P$$

$$M_3 \rightarrow M_4 : r_1r_2r_3P, r_1r_3P, r_2r_3P,$$

$$r_1r_2P, s_1s_2s_3P, s_1s_3P, s_2s_3P, s_1s_2P$$

Note that M_1 does not need to send its long-term public share since it should be publicly available. In the last round M_4 sends the following messages to the other group members.

$$M_4 \rightarrow M_1 : r_2r_3r_4P, r_4s_2s_3s_4P$$

$$M_4 \rightarrow M_2 : r_1r_3r_4P, r_4s_1s_3s_4P$$

$$M_4 \rightarrow M_3 : r_1r_2r_4P, r_4s_1s_2s_4P$$

By the end of this round, all the group members are able to calculate the same two points ($r_1r_2r_3r_4P$, $r_4s_1s_2s_3s_4P$). The generated common secret can be any function of these two points:

$$K_n = \mathcal{F}(r_1r_2r_3r_4P, r_4s_1s_2s_3s_4P)$$

The suggested possibilities for the function $\mathcal{F}$ can be exclusive OR, Weil or Tate pairings, hashing or any function of the coordinates of the two generated points.

Algorithm 1 *Group LMQSV Protocol*

Step 1 (contributions collection):
round i; $i \in [1, n-1]$
1. M_i *selects* $r_I \in \mathbb{Z}_p^*$
2. $M_i \rightarrow M_{i+1}$*:*
$\{\frac{r_1 \ldots r_i}{r_j}P, \frac{s_1 \ldots s_i}{s_j}P|_{j\in[1,i]},\ r_1 \ldots r_iP,\ s_1 \ldots s_iP\}$
Step 2 (Key calculation): round n
3. M_n *selects* $r_n \in \mathbb{Z}_p^*$
4. $M_n \rightarrow M_i$ *:*$\{\frac{r_1 \ldots r_n}{r_i}P, \frac{r_n s_1 \ldots s_n}{s_i}P|_{i\in[1,n-1]}\}$
5. Upon receipt of the above, M_i *computes:* $\frac{r_1 \ldots r_n}{r_i}r_iP, \frac{r_n s_1 \ldots s_n}{s_i}s_iP$
6. $K_n = \mathcal{F}(r_1 \ldots r_nP, r_n s_1 \ldots s_nP)$

5.2. Discussion

In this section, we will first give a heuristical proof of the security of GLMQSV protocol. We can see from the analysis of the protocol that all the messages transmitted through the protocol can be represented as $\frac{\prod s_i}{s_i}P; i \in [1,n]$ and $\frac{\prod r_i}{r_i}P; i \in [1,n]$, where n is the number of members in the group. Assuming the intractability of elliptic curve discrete logarithm problem, an intruder will not be able to calculate the generated shared secret, even if the intruder has access to a long-term key s_i of an authorized group member M_i. The only possible attack can be done as following: suppose that s_i has been compromised, the intruder can strip this value of from the message transmitted by M_i and adds another value s_I (I here relates to Intruder). The result from this attack will be as following: all the group members except M_i will calculate a secret value $\mathcal{F}(r_1r_2...r_n, r_ns_1s_2...s_I...s_nP)$, while M_i will calculate $\mathcal{F}(r_1r_2...r_n, r_ns_1s_2...s_nP)$. Meanwhile the intruder will not be able to calculate either keys since the intruder has no access to the ephemeral secret key r_i of M_i. This attack can be easily discovered, by fulfillment of key confirmation property.

Beyond the security of the system, the complexity of the protocol has always been an important issue when designing group key management systems. From the conceptual perspective, we are interested in two major cost aspects: the cost of communications, that is the number of serial rounds[2], number and the type of messages, and the maximum message size; and the cost of computations. Here for the computational cost, we consider the cost of the modular exponentiations, since it is the most costly computational process. There is always a trade off between both costs, but this is typically based on the underlying communication facilities and on the applications. In ad hoc wireless networks, we should consider both of them. On one hand, the mobile devices are often small and portable. Therefore, they do not have much memory or computational power and they are probably not tamper-resistant. On the other hand, the connections in ad hoc networks are usually unreliable. Consequently, the number and size of messages should be reduced as much as possible, especially multi hop messages. In our protocol, the total required computations are distributed among all the group members, which reduces the required computation power for each member. Although the number of required messages may be larger than that of some other existing protocols like GDH.2 [17] and Hypercube [4], we should mention that most of the manipulated messages are one hop message (except the last message) making these more reliable than a multi-hop messages. In the next section, we will introduce a significant improvement in complexity of the existing protocols (as well as GLMQSV) by using an appropriate clustering technique. To conclude, we can claim that GLMQSV provides an authenticated key agreement protocol with comparable complexity to the previous unauthenticated protocols. The characteristics of GLMQSV are summarized in the following table:

Total number of rounds	$R = n$
Total number of messages	$M = 2(n-1)$
Exp. by the i-th member, M_i	$Ex_i = 2(i+2)$
Exp. for M_n	$Ex_n = (n-1)$
Max. message size for i-th member	$M_s = 2(i+1)$

[2]The serial rounds denotes the sequence of all operations which have to be performed sequentially. Therefore, parallel operations are counted only once in computing the cost.

6. Clustered group key agreement protocols

The previous discussion of GLMQSV protocol shows that the overhead of the key agreement protocols (both communication and computation) increases linearly with the size of the communicating group, which is not efficient especially for large group sizes. Moreover, the required distribution of group members in GLMQSV protocol may not be suitable for some applications using ad hoc networks. In this section, we present a modified version of GLMQSV protocol to address these two drawbacks. Our approach is based on multi-level clustering of the universal group into small, size-bounded clusters, which is typical of many of application of ad hoc networks (military, and rescue mission applications for instance). There are many techniques to distribute the group members into clusters (see for example [5]). In the following section, we explain our proposal for distributing the group members into cluster.

6.1. Distribution of Members

We consider that each node is equipped with a secret *group identity key* K_{IG}, a one-way hash function $\mathcal{H}$, a strong symmetric encryption algorithm and certain public parameters for the key generation protocol. The group identity key guarantees the authenticity of the node as a member of a group, not its individual identity. We also consider that each node has its local identifier (ID) and has the ability to compute its *weight*. The node weight is a numerical quantity which expresses the current status of the node. There are many factors which affect the calculation of the node weight: mobility, battery power level, distance from the other nodes, values related to the surrounding environment (terrain, temperature, battery power, etc.) [3]. Since we are concerned with the protocol efficiency and since some nodes (in our protocol) will be required to perform more computation than others, we will only consider the factors which affect the computational capability of the node.

In the first step, each node makes its active neighbours aware of its presence by broadcasting an initial IAMALIVE message, containing its ID and weight encrypted with $K_H^i = H(S_K^{i-1})$, i.e, the key is obtained by applying the one-way function H to the key generated from the previous key regeneration. Initially, $S_K^0 = K_{IG}$. Once the nodes have gathered information about their neighbours, the second step begins (cluster

construction). An initiator sends a message to its neighbours (Note that any member can be an initiator). The initiator compares the weights of the first $d-1$ replies from its neighbours[3], and assigns a certain number to each one (including itself) based on the weight of each member. For example (see Figure 1), the lightest weight node will be indexed as M_1 and the heaviest weight will be indexed as M_d. Those nodes ($M_i; i = 1, \ldots, d$) constitute the root cluster, C_0. Each member, M_i of the root cluster will declare itself as a leader. Then, it broadcasts to its 1-hop neighbours (except C_0 members) a message, IAMLEADER. A node receiving IAMLEADER message replies with IAMCHILD message to its leader. The leader confirms its leadership to the first $d-1$ children accompanied with an indexing to each node related to its weight. For example the lightest weight node will be the first node in the indexing scheme, $M_{i.1}$ and the second heaviest one will be assigned as $M_{i.2}$ and so on. These children mark themselves as a child to this leader and constitute cluster C_i. Note that the cluster size may vary with the upper limit of $d-1$. The process continues for the members of the second level clusters, namely each member, $M_{i.j}$, where $j = 1, \ldots, d-1$, broadcasts to its 1-hop neighbours (except its cluster members and its ancestor) a message, IamLeader. A node receiving IamLeader message replies with IAMCHILD message to its leader. The leader confirms its leadership to the first $d-1$ children accompanied with an indexing to each node related to its weight as previously mentioned. Note that if this member does not receive any reply within a certain period, its status will not change to a leader but continue as a child, this condition applies to all members including C_0 members. This process continues until the children do not get any replies meaning that all members have been assigned a place in a certain cluster. The previous construction can be seen as considering each member of the root cluster is a root member of a different tree. As a result, we have d subtrees for a multi-rooted tree. The height of the d subtrees can vary according to the distribution of the members in the network, even within the same subtree, it may have different height branches (in fact this is a more realistic construction). If the distribution of the members is uniformly distributed within a certain cluster size, we can say that our hierarchical structure is a well balanced structure consisting of a d-root tree with $(d-1)$-ary subtrees each with maximum height h_i.

The key generation protocol takes place once the hierarchical structure has been created. In other words, when the members do not receive any replies from their neighbours and confirm themselves as children. This case can be reached in each subtree independently, which means that subtrees can start the protocol from different points. The first phase of the protocol is the same for all subtrees, so we discuss the protocol in one subtree and the same can be applied to others. The second phase of the protocol starts when each subtree ends up with a certain secret value known by all the members of the subtree. The main idea in adopting clustering in group key agreement protocols is to let members of clusters at the same level generate the cluster session key using our protocol GLMQSV [4] as a building block. After agreeing on the cluster session key, the cluster leader engages with the upper level cluster in generating the upper level cluster session key. After agreeing on the upper level cluster session key, the lower level cluster leader transmits the upper level cluster session key to the lower level members encrypted with the lower level cluster session key. This process continues until the root cluster members calculate the global session key and broadcast it to the lower level members encrypted with the suitable key. The efficiency of this protocol comes from concurrent processing of the protocol by all the clusters in the same level.

7. Conclusion and Future Work

In wireless ad hoc networks where the reliability of the communication link is not guaranteed, we prefer most communications to be one hop instead of multi-hop communications, so the serialization of group members is preferred to limit multi-hop communications.

In this paper, we have proposed an authenticated key agreement protocol which address most of the constraints and characteristics of ad hoc wireless networks. The most important feature of our protocol is that a common session key has been generated using contribution from all the network members without the assistance of a central authority. Also the proposed protocol is efficient in both communication and computation cost.

Several lines of future work are possible. First, due to the dynamic nature of ad hoc wireless, our solution should handle adjustments to group secrets after membership changes. We are working now to consider the protocol maintenance after membership changes or member movement through the network . Second, a formal security analysis of the proposed protocol is needed. Third, a concert measure of the protocol performance to figure out which clustering topology provides the best efficiency is required.

REFERENCES

[1] N. Asokan and P. Ginzboorg. "Key Agreement in Ad-hoc Networks". *Journal of Computer Communications*, 23(17):1627–1637, Nov. 2000.

[2] G. Atenies, M. Steiner, and G. Tsudik. "New Multiparty Authentication Services and Key Agreement Protocols". *IEEE Journal on Selected Areas in Communications (JSAC)*, 18(4):628–639, Apr. 2000.

[3] Note that in this stage only the members with the heaviest weight in their neighborhood will reply

[4] Note that our clustered approach is general and can adopt any other key agreement protocol as a building block

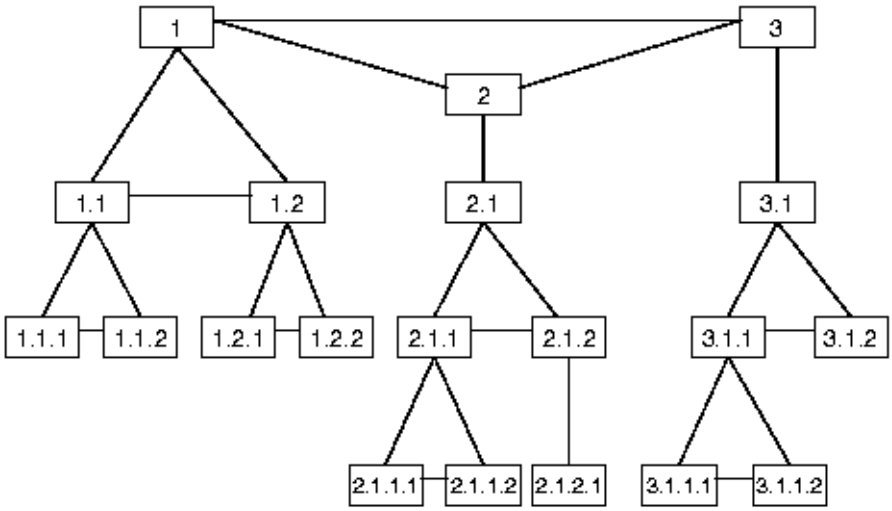

Figure 1: Distribution of group members

[3] S. Basagni, K. Herrin, D. Bruschi, and E. Rosti. "Secure Pebblenets". In *Proc. of 2001 ACM International Symp. on Mobile Ad Hoc Networking and computing, pp. 156–163*, Long Beach, CA, USA, Oct. 2001. ACM Press.

[4] K. Becker and U. Wille. "Communication Complexit of Group Key Distribution ". In *Proc. of ACM CCS'98, pp. 1–6*, 1998.

[5] M. Chatterjee, S.K. Das, and D. Turgut. " WCA: A Weighted Clustering Algorithm for Mobile Ad hoc Networks". *Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks)*, 5(2):193–204, Apr. 2002.

[6] B. Dahill, B. Levine, E. Royer, and C. Shields. "A Secure Routing Protocol for Ad Hoc Networks". Technical Report UM-CS-2001-037, University of Massachusetts, Aug. 2001.

[7] W. Diffie and M. Hellman. "New Direction in Cryptography". *IEEE Transactions on Information Theory*, 28(5):644–654, Nov. 1976.

[8] Y. Hu, A. Perrig, and D. Johnson. "Ariadne: A secure on-demand routing protocol for ad hoc networks". In *Proc. of 8th ACM International Conference on Mobile Computing and Networking MobiCom'02*, 2002.

[9] Y. Hu, A. Perrig, and D. Johnson. "SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks". In *Proc. of 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '02), pp. 3–13*, Jun. 2002.

[10] M. Ilyas. *"The Handbook of Ad Hoc Wireless Networks"*. CRC Press, Washington D.C., 2003.

[11] I. Ingemarsson, D. Tang, and C. Wong. "A Conference Key Distribution System". *IEEE Transactions on Information Theory*, 28(5):714–720, Sep. 1982.

[12] A. Khalili, J. Katz, and W. A. Arbaugh. "Towards Secuity solutions for Trully Ad Hoc Networks". In *Proc. of IEEE Workshop on Security and Assurance in Ad Hoc Networks, in Conjuction with the 2003 International Symposium on Applications and the Internet*, Jan. 2003.

[13] L. Law, A. Menezes, M. Qu, J. Solinas, and S. Vanstone. "An Efficient Protocol for Authenticated Key Agreement Protocol". *Design, Codes and Cryptography*, 28(2):119–134, Mar. 2003.

[14] S. Lee, S. Hong, H. Yoon, and Y. Cho. "Accelerating Key Establishment Protocols for Mobile Communication". In *Proc. of 4th Australasian Conference, ACISP'99, pp. 51–63*, LNCS 1587, 1999.

[15] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar. "SPINS: Security Protocols for Sensor Networks". In *Proc. of Mobile Computing and , pp. 189–199*, 2001.

[16] M. Steiner, G. Tsudik, and M. Waidner. "Cliques: A New Approach to Group Key Agreement". In *Proc. of the 18th International Conference on Distributed Computing Systems (ICDCS'98), pp. 380–387*, May 1998.

[17] M. Steiner, G. Tsudik, and M. Waidner. "Key Agreement in Dynamic Peer Groups". *IEEE Transactions on Parallel and Distributed Systems*, 11(8):769–780, Aug. 2000.

[18] L. Zhou and Z. Haas. "Securing Ad Hoc Networks". *IEEE Network Magazine*, 13(6):24–30, Nov./Dec. 1999.