Authenticated Group Key Agreement Protocols for Ad hoc Wireless Networks

Ahmed Abdel-Hafez[1], Ali Miri[1], and Luis Orozco-Barbosa [2]
(Corresponding author: Ali Miri)

School of Information Technology and Engineering, University of Ottawa[1]
Ottawa, Ontario, K1N 6N5, Canada
Instituto de Investigación en Informática de Albacete[2]
Universidad de Castilla La Mancha, Campus Universitario s/n, 02071 Albacete, SPAIN

(Received Sept. 10, 2005; revised and accepted Jan. 3, 2006)

Abstract

The lack of fixed infrastructure, both physical and organizational, and the highly dynamic nature of ad hoc networks, presents a major challenge in providing secure, authenticated communication for these networks. Traditional key management solutions reported in the literature lack both the flexibility and robustness required to cope with the dynamic nature of ad hoc networks. In this paper, we propose two different n-party authenticated key agreement protocols enabling authorized nodes to generate their own session keys. The first protocol introduces a solution based on clustering techniques suitable for networks with partial structure and composed of a large number of nodes. The second protocol assumes no structure and provides authentication with a minimal increase in communication and computational overhead required.

Keywords: Authenticated Group Key Agreements, Identity-Based Pair-Wise Keys, and Clustering

1 Introduction

Ad hoc wireless networks have become an integral part of all kinds of networks, and have found applications in military operations, rescue missions and many other collaborative applications in mobile environments by providing instant network infrastructure. They are typically communication networks that do not have a pre-existing infrastructure and consist of mobile terminals that connect by relaying messages from one point to another via peer devices. The lack of infrastructure implies an absence of a central network management entity, fixed routers and name servers. 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 weaknesses of 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.

The exchange of cryptographic keys in these types of networks may have to be addressed on demand and without assumptions about a priori negotiated secrets. There are many proposed key agreement protocols designed for wireless networks based on private-key cryptography. The reason for preferring private-key cryptosystems is that the wireless devices are not fully qualified to perform the heavy computations required in public-key cryptosystems. However, private-key based solutions require an on-line trusted third party (TTP) to distribute the session keys which may not be a realistic situation in an ad hoc network setting. On the other hand, there are many group key establishment protocols in the literature based on public-key cryptography for wired static networks [2, 6, 17, 18]. These protocols have not been designed with the special nature of ad hoc networks in mind. Indeed these protocols require a fixed predefined structure, which is often impractical in the typical sparse connections of ad hoc wireless networks. Moreover, most of these protocols are only secure against passive attacks and do not provide authentication services in their key agreement protocols. This approach is especially problematic in ad hoc wireless networks where the communicating members need to authenticate each other to prevent from most active attacks.

The purpose of this paper is to define an authenticated and efficient key agreement protocol for a group of communicating nodes in an ad hoc wireless network. The remainder of this paper is organized as follows. Section 2 presents the limitations of ad hoc networks and their effects on choosing a suitable security scheme. In Section 3, we present a brief overview of the existing group key agreement protocols and discuss their suitability and limitations. The first proposed protocol A-DTGKA, given in Section 4, uses identity-based pairwise keys for entity authentication, and is suitable for networks where a partial structure exists or can be formed. Section 5 introduces the second protocol, A-BD, which is based on a

multiplicative group and does not require a fixed topology for the underlying structure. Our conclusions and future work plans are given in Section 6.

2 Wireless Communication Systems

One of the major challenges in ad hoc network security is that ad hoc networks typically lack a fixed infrastructure both in the form of a physical infrastructure, such as, routers, servers and stable communication links as well as in the form of an organizational or administrative infrastructure. Another difficulty lies on the high dynamic nature of ad hoc networks where the 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 needed cryptographic keys.

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 but the network can also 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 those of traditional networks. This may limit the number and size of messages sent during protocol execution.

- Energy constrained nodes: Nodes in ad hoc networks 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 and sleep deprivation torture attack that aims at depleting the nodes energy source.

- Limited physical security: The use of wireless communication and the exposure of the network nodes increase 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 networks may also have limited CPU performance and memory, e.g., low-end devices such as PDAs, cellular phones and embedded devices. As a result, computationally or memory expensive algorithms might not be of any use in such networks.

3 Related Work

In this section, we review some previous approaches taken towards the definition of security mechanisms for ad hoc wireless network as well as traditional key agreement protocols for structured networks. We will examine their suitability of use in ad hoc networks.

3.1 Ad hoc Wireless Network Security

A large part of the research in ad hoc wireless network security has been aimed at developing secure routing protocols. 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 [7], SPINS [15]). Only recently there have been some attempts to define the key distribution problem in ad hoc networks as discussed below.

In [13], Khalili et al. have suggested an overvall approach to key distribution that combines Identity-Based Encryption (IBE) and threshold. The proposed idea does not address clearly how the nodes involved would authenticate each other in the initialization stage of their proposal, which can create 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). A threshold distributed Certificate Authority (CA) solution was proposed by Zhou and Hass [20]. However, suitability of using expensive threshold public key algorithms required in resource constraint nodes in a typical ad hoc network was not addressed. 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, SASC is responsible for exchanging information with the base station to ensure that all expensive computations are 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].

3.2 Key Agreement Protocols for Wired Networks

In this section, we review some of the existing solutions for group key establishment and evaluate their suitability for ad hoc networks. Several solutions for extending the well known Diffie-Hellman key exchange to a multi-party key agreement have been proposed. One of the earliest proposal due to Ingemarson [11] assumes that the network nodes can be arranged into a ring. The nodes distribute the required pieces of information to compute the key. It is clear that the the minimum number of protocol rounds to come out with the key is $n-1$, which is higher than optimal. This protocol is not secure either, since a passive

eavesdropper can deduce the key if it is able to listen to all of the communication links simultaneously. Burmester and Desmedt (BD) [6] protocol executes in only three rounds:

1) Each node, M_i, generates its random exponent N_i and broadcasts $Z_i = \alpha^{N_i}$.

2) Each M_i computes and broadcasts:

$$X_i = (\frac{Z_{i+1}}{Z_{i-1}})^{N_i}$$

3) Each M_i can now compute the key:

$$S_n = Z_{i-1}^{nN_i}.X_i^{n-1}.X_{i+1}^{n-2}\ldots X_{i-2} \ mod \ p$$

The key defined by BD is different from all protocols discussed thus far, namely:

$$S_n = \alpha^{N_1N_2+N_2N_3+\ldots+N_nN_1}.$$

The protocol has been proven secure against passive attacks given that the Computational Diffie-Hellman (CDH) problem is intractable. While the BD protocol is efficient and provably secure against passive attacks [12], it is not well suited to dynamic groups and needs to be authenticated to overcome active attacks.

Another family of protocols was proposed by Steiner et al. in [19]. It requires one broadcast message at the end of each protocol run. The network topology is a linear chain where the last node has multicast broadcast capabilities. In the last step of the protocol, the key pieces needed by each participant are broadcasted to all parties.This is an important protocol class since its security has been reduced to the security of the two-party Diffie-Hellman.

Becker and Wille [4] have proposed the Hypercube protocol and an extension to it, named Octopus, aiming to minimize the number of rounds required to generate the session key. In one of its simplest forms of the Hypercube protocol, four nodes are arranged to create a shared session key by requiring just four Diffie-Hellman key exchanges. The four-way key exchange can be generalized to a 2^d ($d = 2$ in the case of 4 participants). The Octopus protocol is a modified version of Hypercube aiming to minimize the number of messages. Here the topology is such that four participants constitute a center and the remaining participants form "tentacles" being attached to one of the central nodes.

None of these protocols are suitable for all types of ad hoc networks. This is mainly because they demand that the network topology to follow a prescribed structure. The protocol to be used should be chosen so that it is always possible to arrange the nodes according to the required topology. Moreover, the complexity of most 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. Additionally, most of these protocols do not provide authentication services with their key agreement protocols, which make them vulnerable to many active attacks.

In the next section, we will propose an n-party identity-based scheme that in conjunction with an efficient clustering technique can be used to provide an authenticated secure protocol for networks with a partial structure in place or the ability to be set up. In Section 5, an alternative protocol is proposed which does not assume an underlying structure, and requires only a small number of rounds to provide authentication.

4 A-DTGKA Protocol

In this section, a framework for an efficient and scalable key agreement protocol, *Authenticated Dynamic Topology Group Key Agreement protocol (A-DTGKA)* is presented. It is based on multi-level clustering of the universal group into small, size-bounded clusters. We also show how this approach can be adapted to dynamic ad hoc wireless networks in which there is no static structure or topology.

The proposed A-DTGKA protocol comprises of two phases:

1) Organization of the network nodes into clusters in a multi-level structure.

2) Generation of the group session key.

In the following subsection, we assume that each node is equipped with a secret *group identity key*, K_{IG}, and a one-way hash function $\mathcal{H}$. The protocol also considers 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 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 the main concern of this paper is the protocol efficiency and since some nodes, in A-DTGKA protocol, will be required to perform more computation than others, the factors which affect the computational capability of the node will only be considered.

4.1 Cluster Construction

There are some constraints that should be taken into account when designing a clustering scheme. First, we have to make sure that all clusters have minimum and maximum size constraints. A maximum size constraint limits the cluster size which effectively improves the protocol performance. Ideally, we want all clusters to be of the same size so that no clusters are overburdened, or underburdened by the processing and storage requirements of the proposed key agreement protocol, although this may not a realistic assumption for ad hoc wireless networks. Our protocol is flexible enough by only requiring the cluster size to be limited to a given but arbitrary size. Furthermore, the height of each branch of the structure can vary.

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 $K^i_{\mathcal{H}} = \mathcal{H}(S^{i-1}_K)$, i.e. the key obtained from applying the one-way hash function $\mathcal{H}$ to the key generated from the previous key regeneration. Initially, $S^0_K = K_{IG}$.

Once the nodes have gathered information about their neighbours, the second step begins (cluster construction): the heaviest node within a certain region broadcasts to all its one-hop away neighbours another message which will be treated only by the heaviest node in each region. By receiving this message, the heaviest d nodes declare themselves as the whole group leaders and assigns a certain number to each one based on its weight within the leaders. For example (see Figure 1), the lightest weight node will be indexed as M_1 and the heaviest weight node 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 lightest 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, IAMLEADER message. 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 will continue until all children stop getting replies. This will mean 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 as a root member of a different tree. As a result, we get 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. 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 with d-root tree with $(d-1)$-ary subtrees each with maximum height h_i.

The main idea in adopting clustering in group key agreement protocols is to let the clusters in the same level to generate the cluster session key using our protocol, (or any other key agreement protocol) as a building blocks. After agreeing on the cluster session key, the cluster leader engages with the other upper level cluster in generating the upper level cluster session key. This process continue until the root cluster members calculate the global session key. This key is then broadcasted, one level at a time, to lower level cluster by encrypting the key using the lower level cluster session keys. The efficiency of this protocol comes from a concurrent processing of the protocol by all the clusters in the same level.

4.2 Session Group Key Generation

In this section, an authenticated group key agreement protocol will be presented which uses identity-based pairwise keys for entity authentication. In this scheme a master key, $\mathcal{S}$, can be stored distributively in order to minimize key escrow [5] with the trusted authorities (TAs). The expected group of nodes that would join the network will get their private keys from TAs, such that the private key of a user A can be computed as: $\mathcal{S}Q_A = \mathcal{S}_1 Q_A + \mathcal{S}_2 Q_A + \cdots + \mathcal{S}_k Q_A$, where the parameter $Q_A = \mathcal{H}(ID_A)$ and k is the number of the TAs in this scheme. Once every user has received the respective private key, every node can compute the pairwise shared secret key using the properties of Weil Pairing, as proposed in [16]. For example user A and B can compute:

$$\begin{aligned} e(\mathcal{S}Q_A, Q_B) &= e(Q_A, \mathcal{S}Q_B) = e(Q_A, Q_B)^{\mathcal{S}} \\ K_{AB} &= e(Q_A, Q_B)^{\mathcal{S}} \end{aligned}$$

The identities of the users are assumed to be available publically. The shared secret key that is computed is a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are groups of some prime order q, where q is chosen to be very large number. To get a shared key we can apply a strong hash function to the resulting shared secret, i.e. $\overline{K}_{AB} = \mathcal{H}(k_{AB})$. Using this scheme, each pair of parties are able to have a shared secret key *without* any information being passed and without the risk of any attack (like man-in-the-middle) during information sharing as in the case of conventional two-party key agreement protocols. Finally, each member can authenticate itself to the others by merely encrypting, the transmitted messages using the pairwise shared key.

Below, we will describe the authentication scheme used for one cluster only, which can be applied to the entire group by a natural extension (Algorithm 1). In the following, $\mathbf{E}_K(m)$ means that the message m is encrypted with a key K. As an example, we consider have a cluster with four members, namely, A, B, C, D in which that D is the cluster leader. A picks a random value r_A and, using a generator of the underlying group α, calculates α^{r_A} but instead of sending this value to the next member, it sends $\{\mathbf{E}_{K_{AB}}(\alpha^{r_A} \ \|ID_A)\|ID_A \ \|\mathbf{E}_{K_{AC}}(ID_A) \ \|\mathbf{E}_{K_{AD}}(ID_A)\}$, where ID_A is the identity of the member A, and $\|$ refers to a concatenation of two messages. Note that the member ID has been appended in plaintext to inform the recipient that this message is coming from A. Upon

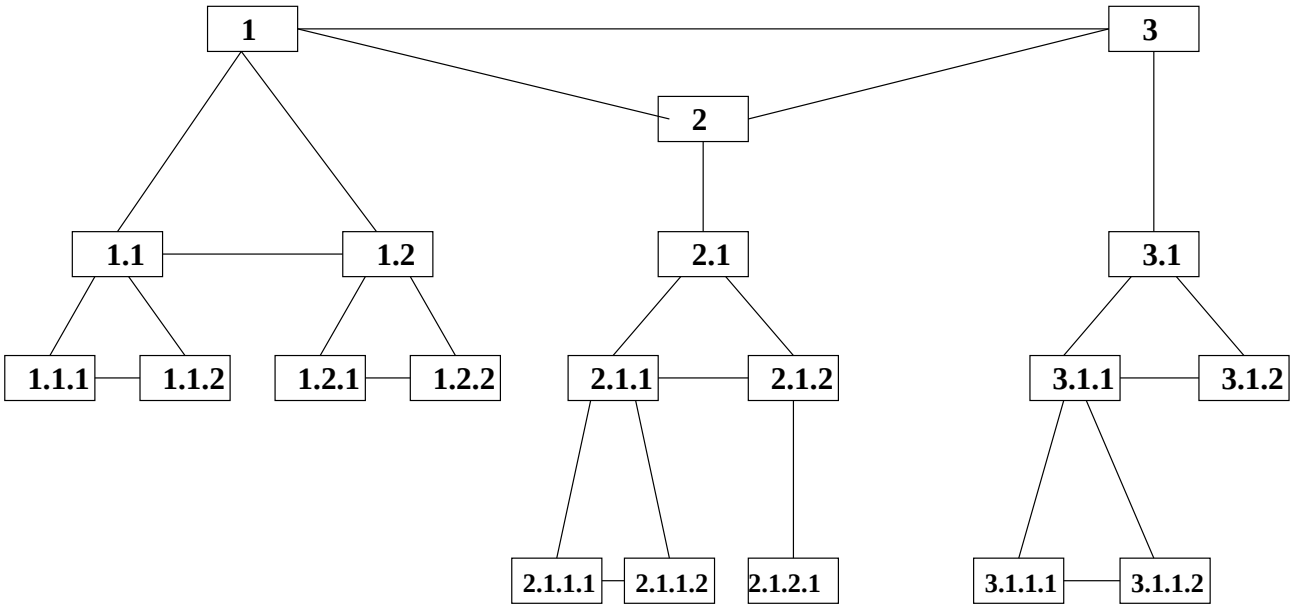

Figure 1: Distribution of group members

receiving this message, B checks the transmitter ID, uses the shared key with this member to decrypt the encrypted message. Finally B compares the encrypted ID with the plaintext one. If the check passes, A is authenticated by B, and since only B can decrypt the message, B is also authenticated by A. After the successful check, B proceeds with the second round of the protocol. B calculates $\alpha^{r_B}, \alpha^{r_A r_B}$ and forwards the rest of the message it has received with its contributions to C, namely $\{\mathbf{E}_{K_{BC}}(\alpha^{r_A} \| \alpha^{r_B} \| \alpha^{r_A r_B} \| ID_B) \| ID_B \| K_{AC}(ID_A) \| K_{BD}(ID_B) \| K_{AD}(ID_A)\}$. Note that the last two parts of the message are used to authenticate A and B to the cluster leader D. Upon receiving this message, C checks the identity of the transmitter, decrypts the first part of the message using the shared key with the transmitter where it is able to authenticate both A and B. C then calculates its contribution and sends to D:

$\{\mathbf{E}_{K_{CD}}(\alpha^{r_A r_B} \| \alpha^{r_A r_C} \| \alpha^{r_B r_C} \| \alpha^{r_A r_B r_C}) \| ID_C \| \mathbf{E}_{K_{BD}}(ID_B) \| \mathbf{E}_{K_{AD}}(ID_A)\}$. As a last round, the cluster leader (D in our example) sends the following messages to its children:

$$D \rightarrow A : \mathbf{E}_{K_{AD}}(\alpha^{r_B r_C r_D} \| ID_D) \| ID_D$$

$$D \rightarrow B : \mathbf{E}_{K_{BD}}(\alpha^{r_A r_C r_D} \| ID_D) \| ID_D$$

$$D \rightarrow C : \mathbf{E}_{K_{CD}}(\alpha^{r_B r_A r_D} \| ID_D) \| ID_D$$

Upon receiving these messages, each member (A, B, C) checks the identity of the transmitter, decrypts the message and compares the transmitter identity in the ciphertext with the one in the plaintext. If the check passes, each member (A, B, C) can authenticate D. This allows for a pairwise authentication of the cluster leader D by its children. Also all the children can calculate the cluster session key, $S_c = \alpha^{r_A r_B r_C r_D}$. For the upper level clusters, the protocol will be repeated between all the cluster members. The children decrypt the message and check the identity of the transmitter, re-encrypt the message using its cluster session key and resend it to its children. For example, (see Figure 1) in the last round (for this cluster) $M_{2.1}$ sends $\mathbf{E}_{K_{2.1,2.1.1}}(\alpha^{r_{2.1} r_{2.1.2}} \| ID_{2.1}) \| ID_{2.1}$ to $M_{2.1.1}$ and its siblings $(M_{2.1.1.1}, M_{2.1.1.2})$. In this case $M_{2.1.1}$decrypts the message and re-encrypts it using its cluster session key, $K_{2.1.1}$, where, $K_{2.1.1} = \alpha^{r_{2.1.1} r_{2.1.1.1} r_{2.1.1.2}}$.

Algorithm 1. *Authenticated DTGKA (A-DTGKA)*

INPUT : $\alpha,\ p, \mathbf{E}_k,\ \mathbf{D}_k, \mathcal{H},\ s_l|_{l\in[1,k]}, ID_i|_{i\in[1,n]},\ \hat{e}(Q,Q)$
OUTPUT: $S_n = \{\alpha^{r_1 r_2 \ldots r_n}\}$

1. *Initial setting: Each member,* M_i*, does the following:*
   - *1.1* $Q_i = \mathcal{H}(ID_i)$
   - *1.2* $sQ_i = s_1 Q_i + s_2 Q_i + ... + s_k Q_i$
   - *1.3* $K_{ij} = \hat{e}(sQ_i, Q_j)|_{j\in[1,n], j\neq i}$
2. *Round* i*,* $i \in [1, n-1]$
   - *2.1* M_i *selects* $r_i \in \mathbb{Z}_p^*$
   - *2.2* $M_i \xrightarrow{\mathrm{M} \| \mathbf{E}_{k_{i(i+1)}}(\mathrm{M} \| ID_i) \| ID_i \| \mathbf{E}_{k_{ij}}(ID_i)|_{j\in[1,n], j>i+1}\}} M_{i+1}$
   - *Where* $\mathrm{M} = \{\alpha^{\frac{r_1 \ldots r_i}{r_x}}|_{x\in[1,i]}, \alpha^{r_1,\ldots,r_i}\}$
   - *2.3* M_{i+1} *Checks* ID_i *and calculates* $\mathbf{D}_{i(i+1)}(\mathbf{E}_{k_{i(i+1)}}(\mathrm{M} \| ID_i))$
   - *2.4* M_{i+1} *Compares* ID_i *with the encrypted* ID_i
   - *2.5 If the check passes go to 2.1*
   - *2.6 Else stop and send Error message.*
3. *Round* n
   - *3.1* $M_n \xrightarrow{\mathbf{E}_{k_{in}}(\{\alpha^{\frac{r_1 \ldots r_n}{r_i}}|_{i\in[1,n]}\} \| ID_n) \| ID_n} M_i$

3.1 M_i Checks ID_n and calculates
$\mathbf{D}_{in}(\mathbf{E}_{k_{in}}(\{\alpha^{\frac{r_1 \ldots r_n}{r_i}}|_{i\in[1,n]}\} \parallel ID_n)$
3.2 M_i Compares ID_n with the encrypted ID_n
3.4 If the check passes calculate $S_n(M_i) = \{\alpha^{r_1 r_2 \ldots r_n}\}$
3.5 Else stop and send Error message.

This process continues up to the root cluster. Achieving authentication through private-key encryption makes A-DTGKA more efficient than the authenticated protocols which use a signature scheme, which is known to be very costly and more robust than hash-chains and time-synchronization authentication schemes.

The protocol described above is an authenticated key agreement protocol which is based on hierarchical authentication. That is, each member authenticates its upper level members through its leader and authenticates its lower level members through its children. Although A-DTGKA does not provide a mutual authentication between all communicating parties, we believe that hierarchical authentication can be suitable in practical applications, like the hierarchical structure of trust which exists in military applications. Meanwhile, A-DTGKA can be modified to provide mutual authentication for all the communicating parties, but this will make the protocol more complicated and it can reduce protocol efficiency.

4.3 Protocol Analysis

Beyond the security of the system, the complexity of the protocol (communication and computation) has always been an important issue when designing group key management systems. In ad hoc wireless networks, both communication costs and computation costs are important factors that should be taken into account when designing a secure protocol. On the one hand, the mobile devices are often small and portable, and therefore, 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 A-DTGKA 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 the flat settings (GDH.2, Hypercube, Octopus, etc.), most of the manipulated messages are one hop messages, since the protocol confines the cluster members to the neighbours with one-hop distance. One-hop messages are much more reliable than multi-hop messages. Also it should be noted that the message size is much lower than that of the flat settings which reduces the required bandwidth. Finally, the efficiency of A-DTGKA protocol is based on the clustering scheme, which in most cases should provide better performance. As discussed in the previous chapter, the best efficiency can be achieved when the hierarchical structure becomes fully balanced, i.e., all the clusters have approximately equal size and the number of levels is not too large.

It is easy to see that DTGKA protocol is secure against passive attacks, given the intractability of DHP or generally DLP, which is believed to be hard problem. Most of the authenticated versions of the previous key agreement protocols used a signature for authentication, which is very costly (signature and verification), whereas the mutual authentication provided by our protocol which uses indentity-based pairwise keys partially addresses some these issues and constraints.

5 Authenticated BD Protocol (A-BD)

In some applications such as mobile networks, the distribution of members in a fixed logical structure is not possible. In this section, a different setting based on a multiplicative group will be presented. The proposed protocol, A-BD (Algorithm 2), does not require a fixed topology from the underlying structure. However the group members should have the capability to broadcast messages to the group members, and receive a simultaneous broadcasts as well. These required characteristics may not be available in many environments. The proposed authenticated key agreement protocol runs only in two rounds regardless of the number of participants in the communicating group. The protocol is based on the idea used in the BD protocol discussed in Subsection 3.2. The A-BD protocol provides authentication with the same primitives as the key agreement protocol as discussed before with a minimum increase in communication and computation overhead.

The protocol assumes that each member of the group has a long-term key pair (s_i, α^{s_i}) where $i = 1, \ldots, n$, where α is a generator of the underlying group, and α^{s_i} is the long-term public key of the i-th member. The protocol runs as follows:

1) Each member, M_i, picks a random number, r_i, calculates its ephemeral public key, α^{r_i}, and broadcasts this value to the other members.

2) Upon receiving the ephemeral public key of the other members, each member, M_i, calculates $X_i = (\alpha^{s_{i-1}})^{r_i} = \alpha^{r_i s_{i-1}}$ and $Y_i = (\alpha^{r_{i+1}})^{s_i} = \alpha^{r_{i+1} s_i}$. From these two values, each member calculates and broadcasts $Z_i = \dfrac{Y_i}{X_i}$ to the other group members.

3) Upon receiving these messages, each M_i can check the correctness of the received messages by making sure that $X_i = Y_i \Pi Z_j|_{j\in[1,n], j\neq i}$. If the check fails, M_i terminates the protocol runs and sends an error message to the other group members. Otherwise, M_i calculates his view of the common secret as follows:

$$K_i = Y_i^n Z_{i+1}^{n-1} Z_{i+2}^{n-2} \ldots Z_{i-1}$$

To demonstrate the protocol, an example of a group consisting of four members is considered. In the first step, each member, M_i, calculates X_i, Y_i and Z_i as shown below:

1) $X_1 = \alpha^{r_1 s_4}, Y_1 = \alpha^{r_2 s_1}, Z_1 = \dfrac{Y_1}{X_1} = \dfrac{\alpha^{r_2 s_1}}{\alpha^{r_1 s_4}}$

2) $X_2 = \alpha^{r_2 s_1}, Y_2 = \alpha^{r_3 s_2}, Z_2 = \dfrac{Y_2}{X_2} = \dfrac{\alpha^{r_3 s_2}}{\alpha^{r_2 s_1}}$

3) $X_3 = \alpha^{r_3 s_2}, Y_3 = \alpha^{r_4 s_3}, Z_3 = \dfrac{Y_3}{X_3} = \dfrac{\alpha^{r_4 s_3}}{\alpha^{r_3 s_2}}$

4) $X_4 = \alpha^{r_4 s_3}, Y_1 = \alpha^{r_1 s_4}, Z_1 = \dfrac{Y_4}{X_4} = \dfrac{\alpha^{r_1 s_4}}{\alpha^{r_4 s_3}}$

Upon receiving $Z_j; j \neq i$, each member M_i can calculate its view of the key $S_n(M_i)$:

$$S_n(M_1) = Y_1^4 Z_2^3 Z_3^2 Z_4 = \alpha^{4(r_2 s_1)} \frac{\alpha^{3(r_3 s_2)}}{\alpha^{3(r_2 s_1)}} \frac{\alpha^{2(r_4 s_3)}}{\alpha^{2(r_3 s_2)}} \frac{\alpha^{r_1 s_4}}{\alpha^{r_4 s_3}}$$

$$S_n(M_2) = Y_2^4 Z_3^3 Z_4^2 Z_1 = \alpha^{4(r_3 s_2)} \frac{\alpha^{3(r_4 s_3)}}{\alpha^{3(r_3 s_2)}} \frac{\alpha^{2(r_1 s_4)}}{\alpha^{2(r_4 3 s_3)}} \frac{\alpha^{r_2 s_1}}{\alpha^{r_1 s_4}}$$

$$S_n(M_3) = Y_3^4 Z_4^3 Z_1^2 Z_2 = \alpha^{4(r_4 s_3)} \frac{\alpha^{3(r_1 s_4)}}{\alpha^{3(r_4 s_3)}} \frac{\alpha^{2(r_2 s_1)}}{\alpha^{2(r_1 s_4)}} \frac{\alpha^{r_3 s_2}}{\alpha^{r_2 s_1}}$$

$$S_n(M_4) = Y_4^4 Z_1^3 Z_2^2 Z_3 = \alpha^{4(r_1 s_4)} \frac{\alpha^{3(r_2 s_1)}}{\alpha^{3(r_1 s_4)}} \frac{\alpha^{2(r_3 s_2)}}{\alpha^{2(r_2 s_1)}} \frac{\alpha^{r_4 s_3}}{\alpha^{r_3 s_2}}$$

We can easily check that:

$$S_n(M_1) = S_n(M_2) = S_n(M_3) = S_n(M_4) = \alpha^{r_1 s_4 + r_2 s_1 + r_3 s_2 + r_4 s_3}.$$

To provide an additional measure of security,the members implementing the protocol should make use of a suitable function $f(S_n)$ of the generated session key S_n, such as those offered by the hash function family.

Algorithm 2. *Authenticated BD Protocol (A-BD)*

Round 1*: Contributions Collection:*
1. M_i *selects* $r_i \in \mathbb{Z}_p^*$
2. $M_i \xrightarrow{\{\alpha^{r_i}|_{i,j\in[1,n],i\neq j}\}} M_j$
3. M_i *calculates* $X_i = \alpha^{r_i s_{i-1}}, Y_i = \alpha^{r_{i+1} s_i}$, *and* $Z_i = \frac{Y_i}{X_i}$
4. $M_i \xrightarrow{Z_i|_{i,j\in[1,n],i\neq j}} M_j$

Round 2*: Key Calculation:*
5. *If* $X_i = Y_i \Pi Z_j|_{j\in[1,n],j\neq i}$
6. *Then* $K_i = Y_i^n Z_{i+1}^{n-1} Z_{i+2}^{n-2} \ldots Z_{i-1}$
7. *Else Send Failure message*

5.1 Protocol Analysis

In this section, an analysis of the proposed protocol (A-BD) in terms of its complexity and also an heuristic security assessment of the protocol is presented. The protocol requires just two rounds to complete regardless of the group size. Regarding the total number of messages, each member in the group issues two broadcasting messages. Regarding the computation cost, every member in the group performs three modular exponentiations and one inversion regardless of the number of members in the group. The message size in the protocol is fixed and can be normalized to one since it is irrelevant of the group size. It should be mentioned that A-BD is more suitable for peer groups since the rule and the responsibility of all members are the same. In other words, there is no special rule for a given member and there is no need for arranging the group members into a fixed topology, since each member broadcast its message to the other group members wherever their place in the network. However, each member should know all its one-hop away neighbours. Additionally, we have to mention that the A-BD protocol based on the ability of each member to broadcast a message to the rest of the group members and to receive messages from all other group members. These capabilities may not be a available in some networks.

In the following, we consider a heuristic proof of our proposed protocol against passive and active attacks. Two kinds of messages are transmitted during the execution of the protocol: α^{r_i} and $Z_i = \frac{\alpha^{r_{i+1} s_i}}{\alpha^{r_i s_{i-1}}}$. It is clear that it is difficult for any intruder to extract any information from these two types of messages. Even if the intruder has access to a long-term key of any member, he will not be able to get any benefit from knowing this long-term key, since the long term key of any member is transmitted in combination with the ephemeral key of the member in such a way that it is so difficult to decouple the two keys without being able to solve the DLP. As a result, we can also claim that the A-BD protocol provides implicit key authentication, since no one outside the authorized members can gain access to the shared generated secret or any partial information about it. Also, we can claim that the A-BD provides Perfect Forward Security (PFS), since knowing a long-term key of any member does not reveal any partial information about the generated secret. It is clear that the generated secret is independent of a previously generated secret in a previous session, so the key independence (forward and backward) is satisfied in the A-BD protocol.

6 Conclusion and Future Work

In this paper, we have presented two different efficient protocols, A-DTGKA and A-BD to provide authenticated secure communications in ad hoc networks. The first protocol A-DTGKA is suitable for networks where a partial structure exists or can be formed. This in addition with the proposed clustering scheme can provide for an efficient

scheme. The second protocol, A-BD, is suitable for networks where the distribution of members in a fixed logical structure is not possible, and when all the members have similar capabilities. A-DTGKA is based on using Weil Pairing, where the pairwise shared key is generated without exchanging any messages. Moreover the mutual authentication comes through private-key encryption which is more efficient than using digital signatures. Another point worth mentioning is that in this proposal provides an easy solution to the mobility issue as if a member needs to move to join another cluster within the group, he only needs to obtain the local parameters of this cluster and the cluster session key as well from the cluster leader encrypted by the global group session key. In this case, the member indexing should be changed according to its new cluster.

We stress that we do not claim that our solution completely handles the key management problem in ad hoc wireless networks. For example, our solution does not handle adjustments to group secrets after any membership change. Also the construction of clusters may be varied based on many criteria. Several lines of future work are possible. First, protocol maintenance after any membership changes or member movement through the network should be considered. Second, formal security analysis of the proposed protocols is a necessary missing step. Third, a concrete measure of the protocol performance to figure out which clustering topology provides the best efficiency is required.

Acknowledgements

This work was supported by the Natural Science and Engineering Research Council (NSERC) of Canada, and the Council of Science and Technology of Castilla-La Mancha under, grant number PBC-03-001.

References

[1] N. Asokan and P. Ginzboorg, "Key agreement in ad-hoc networks," *Computer Communications*, vol. 23, no. 17, pp. 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*, vol. 18, no. 4, pp. 628-639, Apr. 2000.

[3] S. Basagni, K. Herrin, D. Bruschi, and E. Rosti, "Secure pebblenets," in *The Proceedings of 2001 ACM International Sympusium 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 complexity of group key distribution," in *The Proceedings of the 5th ACM conference on Computer and Communications security*, pp. 1-6. ACM Press, 1998.

[5] D. Boneh and M. Franklin, "The identity-based encryption from the Weil pairing," in *Advances in Cryptology*, LNCS 2139, pp. 229-231. Springer-Verlag, 2001.

[6] M. Burmester and Y. Desmedt, "A secure and efficient conference key distribution systems," in *Advances in Cryptology - EUROCRYPT '94*, LNCS 950, pp. 275-286. Springer-Verlag, 1995.

[7] 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.

[8] Y. Hu, A. Perrig, and D. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," in *The Proceedings of 8th ACM International Conference on Mobile Computing and Networking*, pp. 12-23, 2002.

[9] Y. Hu, A. Perrig, and D. Johnson, "SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks," in *The Proceedings of 4th IEEE Workshop on Mobile Computing Systems and Applications*, 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*, vol. 28, no. 5, pp. 714-720, Sep. 1982.

[12] J. Katz and Moti Yung, "Scalable protocols for authenticated group key exchange," in *The Proceedings of The 23rd Annual International Cryptology Conference*, LNCS 2656, pp. 630-648, Springer-Verlag, Aug. 2003.

[13] A. Khalili, J. Katz, and W. A. Arbaugh, "Towards secure key distribution for truly ad hoc networks," in *The Proceedings of IEEE Workshop on Security and Assurance in Ad Hoc Networks*, pp. 342-346, Jan. 2003.

[14] S. Lee, S. Hong, H. Yoon, and Y. Cho, "Accelerating key establishment protocols for mobile communication," in *The Proceedings of 4th Australasian Conference on Information Security and Privacy*, LNCS 1587, pp. 51-63, Springer-Verlag, 1999.

[15] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: Security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, Sep. 2002.

[16] R. Sakai, K. Ohgishi, and M. Kasahara, "Cryptosystems based on pairing," in *The Proceedings of the 2000 Symposium on Cryptography and Information Security*, pp. 26-28, Okinawa, Japan, January 2000.

[17] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-Hellman key distribution extended to group communication," in *The Proceedings of the 3rd ACM conference on Computer and communications security*, pp. 31-37. ACM Press, 1996.

[18] M. Steiner, G. Tsudik, and M. Waidner,"Key agreement in dynamic peer groups," *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, no. 8, pp. 769-780, Aug. 2000.

[19] M. Steiner, M. Waidner, and G. Tsudik, "Cliques: A new approach to group key agreement," in *Proceedings of the The 18th International Conference on Distributed Computing Systems*, pp. 380. IEEE Computer Society, 1998.

[20] L. Zhou and Z. Haas, "Securing ad hoc networks," *IEEE Network Magazine*, vol. 13, no. 6, pp. 24-30, Nov./Dec. 1999.

Ahmed Abdel-Hafez received his Ph.D. in Electrical and Computer Engineering from the University of Ottawa, Ottawa, Canada in 2003. From 2000-2003, he held a teaching assistant position at the School of Information Technology and Engineering, University of Ottawa. His current research interests are wireless communications, computer security and networking.

Ali Miri received his BSc and MSc in Mathematics from the University of Toronto in 1991 and 1993 respectively, and his PhD in Electrical and Computer Engineering from the University of Waterloo in 1997. Having worked as an NSERC Postdoctoral Fellow at the University of Waterloo and the University of Toronto, he joined the School of Information Technology and Engineering (SITE) at the University of Ottawa in July of 2001, where he is currently working as an Associate Professor, and the director of Computational Laboratory in Coding and Cryptography (CLiCC). His research interests include security and privacy technologies and their applications in e-business and e-commerce, such as network security and the role of Public Key Cryptography.

Luis Orozco-Barbosa received the B.Sc. degree in electrical and computer engineering from Universidad Autonoma Metropolitana, Mexico, in 1979, the M.Sc. from ENSIMAG, France, in 1984 and the Ph.D. from University Pierre et Marie Curie, France, in 1987, both in computer science. In 1987, he joined the Multimedia Communications Research Lab, University of Ottawa, first as a Postdoctoral Fellow and then as a Research Engineer. In 1991, he joined the Department of Electrical Engineering, University of Ottawa. In 2002, he joined the Department of Computer Engineerig at the University of Castilla La Mancha, Spain. He is now the Director of the recently created Albacete Research Institute of Informatics. He has published over 200 papers in international Journals and Conferences on computer networks. His current research interests include wireless networks, video communications, simulation and performance evaluation. He serves in the technical program commitees of IFIP Networking and Personal and Wireless Communications conferences. He is a member of the IEEE.