Group Key Management in MANETs

Mohamed-Salah Bouassida, Isabelle Chrisment, and Olivier Festor
(Corresponding author: Mohamed-Salah Bouassida)

Campus Scientifique B.P. 239, 54506, Vandœuvre-Lès-Nancy Cedex - France

(Received Feb. 25, 2006; revised and accepted May 11, 2006)

Abstract

Specific applications like military or public emergency ones require secure group communication in ad hoc environments. The most suitable solution to provide the expected level of security to these services is the provision of a key management protocol.
This paper shows the specific challenges towards key management protocols for securing multicast communications in ad hoc networks, and provides a taxonomy of these protocols in MANETs. A new approach, called BALADE, is also presented. It is based on a sequential multi-sources model, and takes into account both localization and mobility of nodes, while optimizing energy and bandwidth consumptions.

Keywords: Ad hoc networks, group key management, multicast security, taxonomy

1 Introduction

A MANET (*Mobile Ad Hoc Network*) is a communication network characterized by the absence of any fixed infrastructure. It is formed spontaneously with the participating nodes without any preplanning. In addition to the inherently dynamic physical channel caused by shadowing and scattering, the system must adapt itself to the dynamics and the mobility of the nodes.

In parallel to the deployment of ad hoc networks, the last decade saw the constant development of multicast services within the Internet. Multicast transmission is an efficient communication mechanism for group oriented applications, such as video conference, interactive multiparty games and software distribution. One main advantage of multicast communication is to save network resources.

The combination of an ad hoc environment with multicast services, induces new challenges towards the security infrastructure to enable acceptance and wide deployment of multicast communication. Indeed, several sensitive applications based on multicast communications have to be secured within ad hoc environments. We can cite military applications such as group communications in a battle field, but also public security operations, involving fire brigades and policemen.

To prevent attacks and eavesdropping, services among which authentication, data integrity and data confidentiality need to be provided. The most suitable solution to provide these services is the establishment of a key management protocol. This protocol is responsible for the generation and the distribution of the traffic encryption key (TEK) to all group members. This key is used by the source to encrypt multicast data and by the receivers to decrypt it. Thus, only authenticated members are able to receive the multicast flow sent by the group source.

Often, multicast key distribution takes into account a challenging element known as the “1 affects n” phenomenon. After a Join or a Leave procedure, the TEK is renewed and redistributed, affecting all group members in order to maintain both forward and backward secrecies. To address this problem and to reduce its impact on the protocol performance, several approaches propose a multicast group clustering. Clustering consists in dividing the multicast group into several sub-groups. Each sub-group is managed by a local controller (LC), responsible for local key management within its cluster. Thus, after Join or Leave procedures, only members within the concerned cluster are affected by the rekeying process, and the local dynamics of a cluster does not affect the other clusters of the group. Moreover, few solutions for multicast group clustering did consider the energy issue to achieve an efficient key distribution process, whereas energy constitutes a main issue in ad hoc environments [13, 23].

In this paper, we extend and present a taxonomy of group key management protocols, dedicated to operate in ad hoc networks, then we present BALADE, a group key management protocol for ad hoc environments, used to secure multicast communications, according to the sequential multi-sources model. BALADE uses an optimized multicast cluster tree algorithm to ensure efficient key delivery, taking into account the localization and the mobility of nodes.

The remainder of this paper is structured as follows. Section 2 emphasizes the challenges of securing multicast communications within ad hoc environments. A common taxonomy of group key management protocols in wired networks is described in Section 3. Section 4 presents our taxonomy of multicast group key management approaches in MANETs. Section 5 describes the BALADE functional architecture, by detailing the security operations achieved by BALADE, the clustering algorithm and the mobility management. Finally, Section 6 concludes the paper.

2 Challenges and Constraints of Securing Multicast Communications in Ad Hoc Networks

The nature of ad hoc networks, the security level to be ensured and the characteristics of the applications to be secured complexity the deployment of the ad hoc networks associated with the availability of the multicast services. The principal constraints and challenges induced by the ad hoc environment are:

- Wireless Links: the wireless links make the network easily exposed to passive malicious attacks like sniffing, or active attacks like message replay or message alteration;

- No Infrastructure: The absence of infrastructure is one of the main characteristics of ad hoc networks. It eliminates any possibility to establish a centralized entity which concentrates the access to the network through a single point managing the various essential services. Due to this absence of infrastructure, the traditional authentication and keys distribution models are hardly applicable;

- Scalability: The size and the dynamics specific to multicast groups can be very important in ad hoc networks. We cannot control the number of members nor the adhesion frequency to the group. Thus, a group key management protocol should be adapted to this dynamics;

- Mobility: When a node moves within the network, it does not necessarily leave the multicast group and consequently does not have to be forced to re-authenticate itself to its group. Mobility implies also that the network topology and the connectivity between hosts change quickly and unpredictably. Thus, the control and the management of a mobile ad hoc environment will have to be distributed among the participating nodes of the network;

- Limited Power: ad hoc networks are composed of low power devices. These devices have limited energy, bandwidth and CPU, as well as low memory capacities. Consequently, achieving secure group communications in ad hoc networks should take into account additional factors including the energy consumption efficiency, the optimal selection of group controllers and the bandwidth saving.

Several key management protocols for securing multicast communications have been elaborated over the last decade. We present in the next section the principal approaches related to the group key management protocols in wired environments.

3 Multicast Key Management Approaches in Wired Networks

The most commonly accepted taxonomy of group key management protocols divides them into three approaches: centralized, decentralized and distributed.

- Centralized approach:
  The centralized architectures ([24, 15], ...) use only one server. This server is responsible for the generation, the distribution and the renewal of the group key. This approach is clearly not scalable since it suffers from the "1 affects n" phenomenon. Moreover, the unique key server forms a bottleneck in terms of security and resources.

- Decentralized approach:
  This approach divides the multicast group into a p fixed number of sub-groups. Each of them shares a local session key managed by a local controller, thus attenuating the "1 affects n" phenomenon. When a member joins or leaves the group, only the concerned sub-group will renew its local key. The decentralized protocols are subdivided into two categories. In the first one, each cluster or sub-group of the multicast group shares and manages a local traffic encryption key (TEK), requiring several decryption and re-encryption operations of the multicast flow, when it passes from a sub-group to another. This family of protocols is not flexible and not adapted to the dynamics of the multicast group. IOLUS [16] belongs to this set of protocols. To overcome this problem, some protocols, like AKMP [1] and SAKM [4], dynamically divide the multicast group into p sub-groups, p varying according to the group dynamics; thus reducing the overhead of multicast data encryption and decryption, while attenuating the "1 affects n" phenomenon.

  The second category of protocols uses only one TEK for all clusters in the multicast group. They hierarchically divides the multicast group into sub-groups, each of them being managed by a sub-group manager. The managers, which are not members of the multicast group, do not need to decrypt the multicast flow sent by the source. These protocols use a double encryption of the traffic encryption key, that requires more key encryption keys (KEKs). The multicast group clustering is static, and consequently not adapted to the dynamics and mobility of ad hoc networks. DEP [8] belongs to this category.

- Distributed approach:
  In this approach, also called key agreement approach

([7, 10], . . .), all group members cooperate and generate the traffic encryption key, to establish secure communications between them. The key agreement approach eliminates the bottleneck in the network, compared to the centralized approach, but is less scalable because the traffic encryption key is composed of the contributions of all group members, and needs more computation processing.

4 Multicast Key Management Taxonomy in Ad Hoc Networks

The taxonomy proposed for the group key management protocols in wired networks is not adequate for ad hoc networks, because of the specific challenges of such environment, addressed in Section 2. A new taxonomy of group key management protocols in MANETs is presented in [12]. It subdivides these protocols into two categories according to their use or not of the GPS (Global Positioning System). Our taxonomy of group key management protocols in ad hoc networks (cf Figure 1) extends and refines the classical taxonomy as in wired networks, while integrating the specifics of the ad hoc networks (the mobility support, the energy efficiency and the multi-hop awareness).

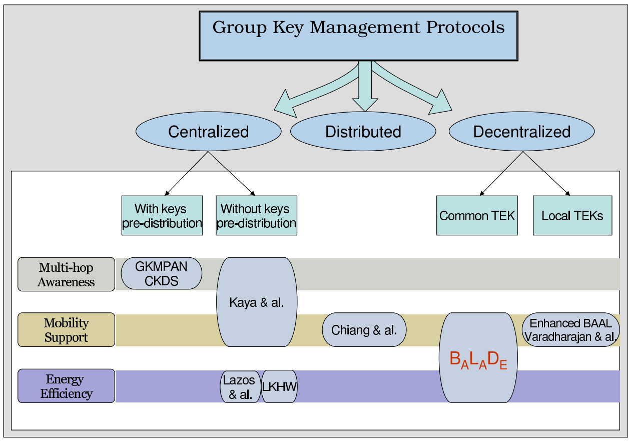


Figure 1: Taxonomy of group key management protocols in ad hoc networks

4.1 Centralized Approach

In this approach, group key management is carried out by only one entity in the network. We split this approach in two categories, with keys pre-distribution and without key pre-distribution phase.

4.1.1 With Keys Pre-distribution

These protocols configure the hosts or entities which will participate to the multicast group, off-line. This configuration is achieved by pre-deploying a set of keys on each node, so that it will be able to decrypt the multicast flow sent by the source, or to obtain the traffic encryption key when the re-keying process is triggered. The keys pre-distribution is used in MANETs because of the lack of infrastructure within ad hoc network which implies unavailability of a central entity to ensure key distribution process on-line. Two protocols belong to this family, GKMPAN [26] and CKDS [17].

GKMPAN [26] is based on a key lists pre-distribution phase to the multicast group members, and on multiple rekeying phases. The main phases of this protocol are the following:

- Key pre-distribution: each group node u obtain, off-line, before the deployment of the ad hoc network, a subset I_u of m keys out of the pool of l keys. These keys are used as key encryption keys (KEKs). The key-predistribution algorithm allows any node who knows another node's identifier j to determine the identifiers of I_j.

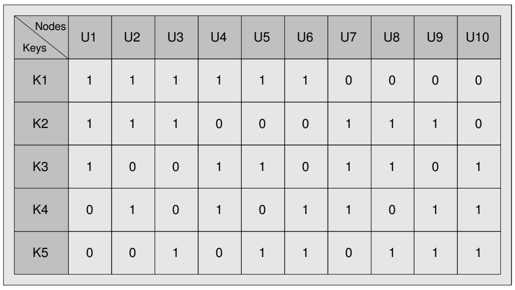
- Authenticated node revocation: when the key server decides to revoke a node, it broadcasts a revocation notification to the network, containing the identifier of the revoked node, and the non compromised key that is possessed by the maximum number of remaining nodes in the network.

- Secure group key distribution: the key server generates and distributes a new group key. The key distribution process is achieved hop by hop, by encrypting the new group key with the predeployed KEKs. When a node is compromised and is revoked by the key server, its predeployed KEKs are also compromised. To face this problem when sending the new group key, the key server determines the identifier of the non compromised KEK, shared with the maximum members of the multicast group. Then, it broadcasts a message authenticated by TESLA [9, 19], containing the new group key encrypted with this chosen non compromised KEK. Group nodes who did not hold the KEK used to the encryption of the traffic encryption key, will receive this group key, forwarded by their neighbors, encrypted with other non-compromised KEKs. So, the key server has only to deliver the new group key to its immediate neighbors, which forward it securely to their neighbors, in a hop by hop way. Thus, GKMPAN exploits the multi-hop property of the ad hoc networks, group members are both hosts and routers.

- Key update: when the group nodes decrypt and authenticate the traffic encryption key, they update their subsets of predeployed KEKs, based on this group key, and erase all the old KEKs. The compromised keys k_i are also updated by the remaining members holding these keys, using a non compromised key k_m as follows: $k_i' = f_{k_m}(k_i)$. f being a pseudo-random function.

CKDS [17] (Combinatorial Key Distribution Scheme) is an application level protocol for securing multicast communications in ad hoc networks. A centralized group controller (GC) is assumed available for distributing keys and initiating re-keying procedure.

The key distribution structure in CKDS is based on a combinatorial Exclusion Basis System (EBS) [18] in cope with CAN [21] (Content Addressable Networks), as follows. Each node in CKDS knows k keys (known keys) and does not know m keys (unknown keys). Figure 2 illustrates an example of EBS matrix, k=3 and m=2. This example in taken from [17].



Nodes Keys	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10
K1	1	1	1	1	1	1	0	0	0	0
K2	1	1	1	0	0	0	1	1	1	0
K3	1	0	0	1	1	0	1	1	0	1
K4	0	1	0	1	0	1	1	0	1	1
K5	0	0	1	0	1	1	0	1	1	1

Figure 2: EBS matrix in CKDS [17]

CAN is used to achieve a partition of all the nodes into an m-dimensional space. Thus, each node has a quadrant in the space, according to the unknown keys in the EBS scheme. If a node is compromised, the re-keying algorithm will start from the diagonal node in the partitioned space, i.e. the node which knows all the unknown keys of the compromised node. In Figure 2, if node U1 is compromised, U6, U9 and U10 can achieve the re-keying process because they know the unknown keys of U1, which are K4 and K5. Thus, the new group keys can be spread via direct flooding along the m dimensions whose keys are not known by the compromised node, isolating this node in the re-keying procedure. Thus, like GKMPAN, CKDS is multi-hop aware.
The authors show that the CKDS protocol outperforms centralized approaches such as logical key hierarchy protocols [24], in term of keys storage requirement, scalability, storage requirement and number of decryption operations per node.

4.1.2 Without Keys Pre-distribution

This category of protocols does not need an off-line pre-distribution of keys. Two protocols belong to this category, the one defined by Kaya et al. [11] and the one defined by Lazos et al. [13].

Kaya et al. [11] propose a group key management protocol, which is efficient against some constraints imposed by an ad hoc environment: mobility, non-reliable links and multi-hop communications overhead. A certification service is provided in this protocol, to ensure access control and revocation of malicious members.

Only nodes with a valid certificate should be able to access the multicast flow. It is assumed that a node wanting to join the multicast group needs a security certificate obtained off-line and signed by a trusted third party (TTP). Excluded nodes, with revoked certificates, should not be able to access to the multicast data, any more. To do this, the source of the multicast group multicasts periodically a signed certificate revocation list. The group members store this list, and can authenticate and check the access control of each new member, wanting joining the group.

This protocol aims to reduce the communication cost and complexity for the mobile multihop nodes. Indeed, nodes join the multicast group and attach themselves to the multicast tree through the closest neighbor, already belonging to the group, by using the GPS information. The join requests are broadcast in a limited range (TTL: Time To Live field), to reach any group member, and the response to the join request is sent in anycast. In addition to the communication cost profit, this fact allows the construction of a multicast tree with short paths, facilitating and optimizing the key distribution. These advantages make the protocol of Kaya et al. aware of mobility and multi-hop challenges in ad hoc environment.

The data integrity is maintained, to counter malicious attacks such as message replays. It is carried out, via the TESLA approach, that requires synchronization between source and receivers, which is expensive in a mobile network.

The proposal of Lazos et al. [13] is a centralized approach, taking into account energy efficiency within the ad hoc networks. Indeed, it improves the keys distribution scheme of LKH [24] and adapts it to static ad hoc networks by optimizing the energy consumption, via the use of the geographical localization of the group members. The basic idea of this approach is that members which are geographically close to each other, can potentially be reached by a broadcast message, or can use the same path to receive the multicast data. The ad hoc network is represented by a two-dimensional space, and the K-means [14] clustering algorithm is used to form groups with strong correlation, and to deduce the multicast tree dedicated to distribute the traffic encryption key. The K-means algorithm consists in assigning the group members to a fixed number of clusters, randomly. Then, the algorithm changes the membership of the clusters by maximizing, at each iteration, the correlation between the members of each cluster. The K-means algorithm stops when the assignment of the members to the clusters does not change, thus corresponding to the best geographical correlation of the clusters. The key distribution process, based on the K-means algorithm, is composed of the following steps:

1) Assign all the group members in one cluster;

2) Divide each cluster into 2 sub-clusters, via the K-means algorithm;

3) Use a refinement procedure to balance the number of

members per cluster;

4) Iterate Steps 2 and 3, until clusters of one or two members are created.

5) Merge clusters with only one member by pair, if possible.

6) Map the cluster hierarchy into the logical hierarchy of the LKH key distribution. Figure 3 illustrates an execution of this algorithm.

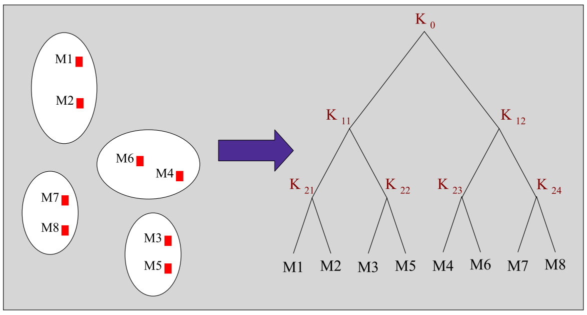


Figure 3: Key distribution tree based on the K-means algorithm [13]

LKHW [20] is a directed diffusion-based secure multicast scheme for wireless sensors networks (WSN). This protocol is a combination of LKH and direct diffusion protocols. The key distribution tree is based on LKH and the re-keying scheme is based on a direct diffusion, optimizing the consumption of energy. The security operations achieved by LKHW are the data confidentiality, data integrity and data authentication. Forward and backward secrecies are also ensured by the re-keying process in LKHW.

The actors of LKHW are the sensors and source of the group, called also sink, which is responsible for the collection of data from the sensors. The sensors are network nodes which can provide data required by the sink. These nodes have low capacities in terms of communication and computation. The main phases of LKHW are the group initialization and the re-keying process after each join or leave event in the group:

- Group initialization: the establishment of a secure group starts with the construction of the logical key hierarchy by the sink. The sink diffuses an exploratory message to all the network nodes, called “interest about interests to join”, to find nodes which can be able to provide information that it requests. Interested nodes will answer with “interests to join” messages, declaring tasks they are able to achieve. The sink collects these “interests to join” messages, and sends for each node its assigned index and the key set according to its logical localization within the LKH tree. From this point, secure communications can be initiated within the group.

- The re-keying process is triggered after each join or leave event. When a node wants to join the group, an “interest to join” message is broadcast to the sink, which answers with the set of keys according to its logical position in the LKH tree. All the group nodes should also refresh their keys sets, to ensure backward secrecies.

  When a node leaves the group, keys in the tree from the root to its position, should be renewed, ensuring forward secrecy.

  The direct broadcast scheme used in LKHW is optimized by using caches for data interest matching, suppressing duplicate messages and preventing loops.

4.2 Distributed Approach

Group key management in the distributed approach is achieved by all the multicast group members, which cooperate to ensure a secure multicast communications between them.

Chiang et al. propose a group key management protocol [6], for MANETs, based on GPS measures (latitude, longitude and altitude) and on the group key exchange protocol GDH (*Group Diffie Hellman*) [10].

During protocol initialization, each node in the ad hoc network, floods its GPS information and its public key to all the others nodes, although the authors assume that the protocol does not rely on any certification authority. Using the GPS information received from others nodes, each group member can build the network topology. When a source wants to multicast the data flow to the group members, it computes the minimum multicast tree, based on the Prüfer algorithm [6].

In addition to the Prüfer sequence, the source of the group multicasts to all the group members the group key, computed as a combination of their public keys. When receiving the Prüfer sequence, each node will decode the multicast tree, and will know whether to forward or discard packets sent by the source.

The construction of the GDH key distribution graph using the Prüfer decoding algorithm is achieved as follows:

- The key graph is composed of two types of nodes: leaf nodes (u-nodes) representing the multicast users nodes, and the k-nodes representing keys. The root of the key graph, called k_p-node, indicates the Prüfer key related to the Prüfer decoding information.

- The key distribution graph specifies a secure group (U, K, P). U is the set of multicast users, K is the set of keys, and P is the Prüfer-key (group key). Figure 4 provides an example of key graph. This example is given in [6].

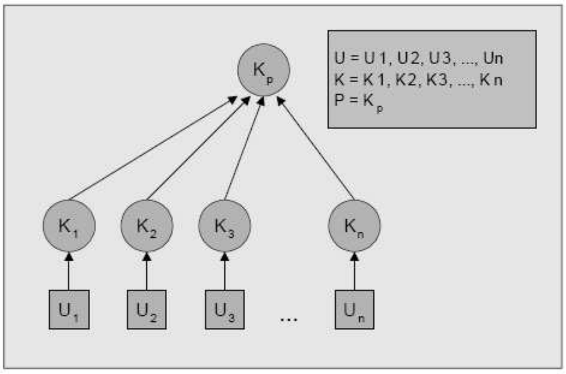


Figure 4: A key graph in Chiang et al. [6]

4.3 Decentralized Approach

The decentralized approach divides the multicast group into sub-groups or clusters, each sub-group is managed by a local controller responsible for the security management of the members of its sub-group. Two sets of protocols compose this approach. The first set uses a local traffic encryption key (TEK) within each cluster, distributed to its local members. When receiving a multicast flow, local controllers must decrypt it with the appropriate key, re-encrypt it with the local TEK of their cluster and forward it to their local members.

The second set uses only one traffic encryption key (TEK) for all group members. The source of the group uses the TEK to encrypt multicast data, and the group members to decrypt it. The challenge of such protocols is to send the traffic encryption key to all members of each clusters, securely and in time.

4.3.1 Local TEKs

The group key management protocol proposed in [22] is designed to operate within a NTDR (Near Term Digital Radio) architecture. This architecture is composed of a set of clusters, each one containing a clusterhead, and the clusterheads form the routing backbone. The mobility of nodes is taken into account in this protocol when clusters are established.

The confidentiality of the multicast communications is achieved via two types of keys:

- The cluster key group (GCK) is used to encrypt all intra-cluster traffic;

- The key encryption key (KEK) is shared by a clusterhead and a node of its cluster. This key is the combination of a secret generated by the clusterhead, and the IP address of the node. $KEK = f(s, @IP)$

The clusterhead encrypts the GCK by the KEK and distributes it to all its cluster members. Thus, all the cluster nodes can encrypt and decrypt the traffic within the cluster. Inter-cluster Communications are restricted to the clusterheads.

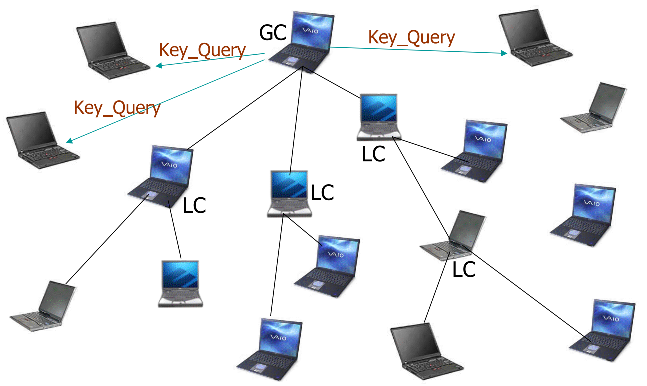


Figure 5: The TEK generation and distribution in enhanced BAAL [2]

The Enhanced BAAL protocol [2] is based on the combination of the BAAL protocol which is a group key management protocol in wired networks, and the dynamic support offered by the AKMP protocol [1] (Adaptive Key Management Protocol). Authentication and key generation are achieved via threshold cryptography [25]. Each entity of the group holds a public and a private key, generated by the server nodes of the threshold cryptography.

The principal actors of this architecture are the global controller (GC), the local controllers (LCs) and the members of the multicast group. The global controller is the source of the multicast group, and is responsible for the generation, distribution and periodic renewal of the traffic encryption key. The global controller sends a request to a defined number of threshold cryptography servers, which answer by sending their contributions. Then, the GC combines these contributions to constitute the traffic encryption key, and distributes it to all its group members. Figure 5 illustrates this key generation process.

A local controller is a member of the multicast tree, forming a sub-group with its local members. A local controller manages a local traffic encryption key, and is responsible for forwarding the multicast flow sent by the source to all its local members. A member of the multicast tree can switch to the local controller state, according to its evaluation function, which contains two metrics: the local frequency of Join and Leave events, and the number of its local members. This evaluation function is an extension of the one presented in [1], and considers the mobility of nodes when computing the evaluation metrics. This approach tends to attenuate the “1 affects n” phenomenon by integrating the dynamic support of AKMP, while limiting the overhead due to the encryption and decryption operations of the multicast data, when passing from a sub-group to another. However, these intermediate operations of data encryption and decryption remain very constraining in an ad hoc environment, with limited resources of storage and computing power.

4.3.2 Common TEK

The most important advantage of this approach is that there is no intermediate encryption and decryption operations of the multicast flow by local controllers of the clusters. This advantage is very important in ad hoc networks where resources are limited. The clustering process is used in this approach to attenuate the "1 affects n" phenomenon while renewing the key encryption keys for each sub-group. The BALADE group key management protocol uses this approach (cf Section 5).

4.4 Discussion

In this section, we compare the above discussed protocols, and analyze their performance and security properties.

The discussed protocols are compared according to the desired properties for multicast communications security in ad hoc networks (authentication, data confidentiality and integrity, nodes revocation, . . .), their computational cost related to the intermediate encryption and decryption operations, their storage cost, their efficiency against bottlenecks, their scalability and vulnerabilities. The table in Figure 6 summarizes these comparisons.

The Kaya et al., Chiang et al. and Lazos et al. propositions requires GPS information, to take into account localization of group members. GPS information is used in Kaya et al. and Lazos et al. to efficiently construct paths between network nodes. However, in Chiang et al., the GPS information is flooded within the network allowing each group member to build the network topology. This flooding is very expensive in ad hoc networks.

In addition to the clustering algorithm required in Enhanced BAAL and Varadharajan et al., the Enhanced BAAL protocol uses the threshold cryptography which requires an initial configuration in the network, to share the private secret of the certification authority between the threshold cryptography servers.

All the proposed protocols which require that each group node holds its public key, need a trusted authority which can provide the proof of identity for each node. Whereas, GKMPAN and CKDS involve only a pre-deployment phase of key encryption keys.

The validation of the keys lists in Kaya et al. and GKMPAN requires the TESLA authentication, and therefore time synchronization between group nodes and buffering reception, hard to deploy in ad hoc environments.

The security services provided by the group key management protocols include data confidentiality, which is realized by encrypting the multicast flow by the source and decrypting it by the receivers. However, the authentication and access control of the group members are only given by Kaya et al. [11] and Enhanced BAAL [2], because a certification management service is available in these two protocols.

In Kaya et al., the certification authority offers off-line security certificates to the group members, allowing them to be authenticated and to join the multicast group online.

The certification management in Enhanced BAAL is achieved via threshold cryptography, which is more adapted to ad hoc networks, and especially to the absence of any fixed infrastructure.

Nodes revocation is ensured via the key predeployment process in GKMPAN [26] and CKDS [17]. Keys of a compromised node in these two protocols will be compromised and never used when achieving the rekeying process. However, the Join procedure is difficult to deploy because a new member joining the group has to have pre-deployed keys.

The intermediate encryption operations metric is very important in ad hoc environments, due to the usually poor computational capacities of its devices. Thus, the most suitable solution in an ad hoc environment should not have to use intermediate encryption and decryption operations. The multicast data is therefore decrypted only by the members, as is carried out in CKDS, Kaya et al., Lazos et al., LKHW and Chiang et al.. The disadvantage of these protocols is that they are centralized around an entity which is responsible for the generation and the distribution of the traffic encryption key and for the diffusion of the multicast encrypted flow.

GKMPAN achieves intermediate encryption and decryption operations of the traffic encryption key and not of the whole data flow. Therefore, all the group members share the same traffic encryption key, which is distributed securely via the pre-deployed keys.

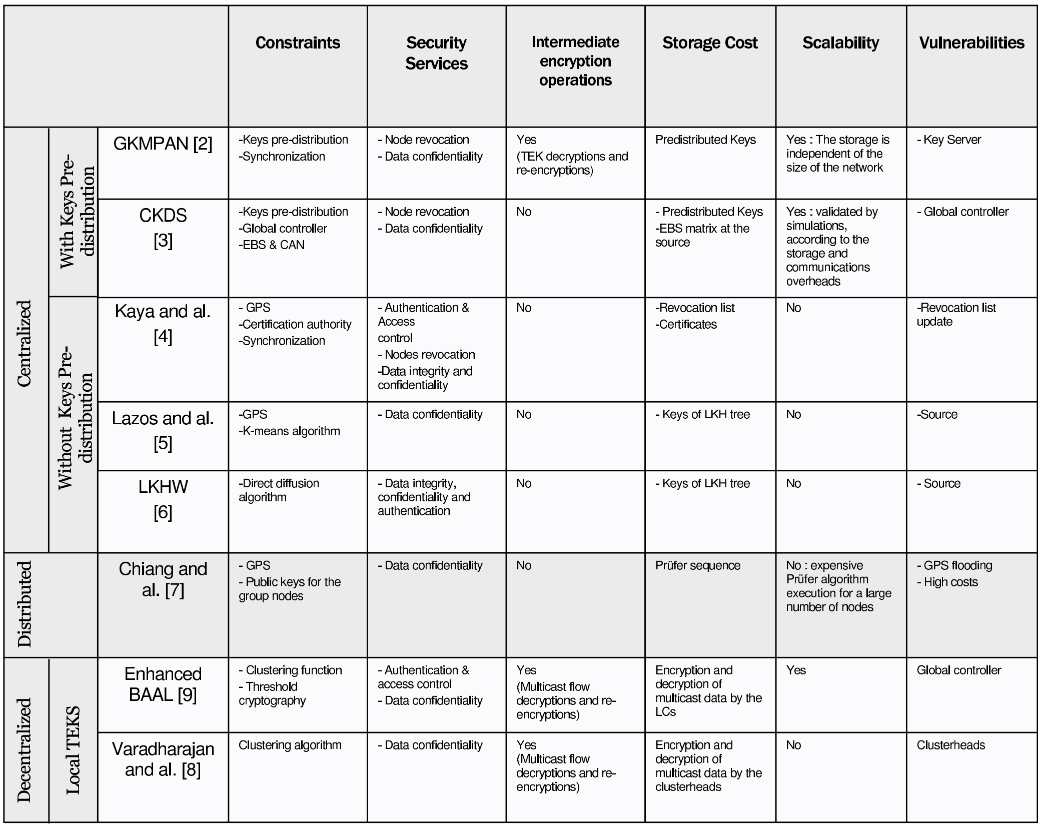
Despite of the advantages of the dynamic clustering approach presented in Section 4.3, protocols proposed in [2] and [22] are not suitable for ad hoc networks, because they use a local traffic encryption key for each cluster, thus requiring intermediate decryption and re-encryption operations of multicast data, by the local controllers or the clusterheads. Consequently, these entities become failure points of vulnerability and bottlenecks.

The storage cost is also a main challenge in ad hoc networks, with limited storage capacities. Protocols belonging to the decentralized approach, Enhanced BAAL and Varadharajan et al. require an expensive storage cost due to the intermediate operations of decryption and reencryption of the multicast flow.

The Prüfer algorithm used in Chiang et al. also requires large storage and computation capacities, especially for a large number of nodes.

The storage in lazos et al. and LKHW include the keys of the LKH tree, whereas GKMPAN and CKDS store the off-line pre-deployed keys for each node. For GKMPAN, increasing the number of predeployed keys m or decreasing the number of keys in the pool l will increase the number of direct logical paths between nodes. However, it is desirable from the storage point of view to decrease m. Moreover, a smaller m and a larger l enhance the security level.

Being a certificate based approach, Kaya et al. requires for each group member to store its certificate and also the



			Constraints	Security Services	Intermediate encryption operations	Storage Cost	Scalability	Vulnerabilities
Centralized	With Keys Pre-distribution	GKMPAN [2]	-Keys pre-distribution -Synchronization	- Node revocation - Data confidentiality	Yes (TEK decryptions and re-encryptions)	Predistributed Keys	Yes : The storage is independent of the size of the network	- Key Server
		CKDS [3]	-Keys pre-distribution -Global controller -EBS & CAN	- Node revocation - Data confidentiality	No	- Predistributed Keys -EBS matrix at the source	Yes : validated by simulations, according to the storage and communications overheads	- Global controller
	Without Keys Pre-distribution	Kaya and al. [4]	- GPS -Certification authority -Synchronization	- Authentication & Access control - Nodes revocation -Data integrity and confidentiality	No	-Revocation list -Certificates	No	-Revocation list update
		Lazos and al. [5]	-GPS -K-means algorithm	- Data confidentiality	No	- Keys of LKH tree	No	-Source
		LKHW [6]	-Direct diffusion algorithm	- Data integrity, confidentiality and authentication	No	- Keys of LKH tree	No	- Source
Distributed		Chiang and al. [7]	- GPS - Public keys for the group nodes	- Data confidentiality	No	Prüfer sequence	No : expensive Prüfer algorithm execution for a large number of nodes	- GPS flooding - High costs
Decentralized	Local TEKS	Enhanced BAAL [9]	- Clustering function - Threshold cryptography	- Authentication & access control - Data confidentiality	Yes (Multicast flow decryptions and re-encryptions)	Encryption and decryption of multicast data by the LCs	Yes	Global controller
		Varadharajan and al. [8]	Clustering algorithm	- Data confidentiality	Yes (Multicast flow decryptions and re-encryptions)	Encryption and decryption of multicast data by the clusterheads	No	Clusterheads

Figure 6: Evaluation of the group key management protocols in MANETs

revocation list, which should be updated by the source. To prevent this list to have a very large size, a technique for withdrawal of entries is used in this list of revocation, but allows revoked nodes to re-join the multicast group after a period of time.

Scalability is a key challenge in group key management protocols in MANETs. However, the majority of the proposed protocols fail to address this criterion.

Without key pre-distribution, group key management protocols do not scale with large group sizes because of their centralized architecture. Also, the protocol proposed by Chiang et al. presents a scalability issue because of flooding GPS information, and execution the Prüfer algorithm for a large number of multicast group members. GKMPAN is more scalable with a storage requirement independent of the size of the multicast group. In the same way, the scalability of CKDS was validated by simulation, according to the storage and the communication overheads.

Centralized Protocols ([11, 13, 17, 20, 26]), rely on an entity to manage keys or certificates, and thus suffer from a failure vulnerability in term of security. Moreover, a centralized server presents a bottleneck in the secure architecture and can be targeted by several malicious attacks, such as denial of service.

The protocol defined by Chiang et al. is robust and efficient, but its greater weakness is that it requires an expensive cost in communication when flooding GPS information, and in computation when computing the Prüfer sequences.

The Protocol in [22] suffers from a high computation overhead due to the clustering process and the election of a leader in each cluster. Moreover, the key management and the inter-cluster communications are restricted to only the clusterheads, which can be bottlenecks and consequently be compromised.

To face and attenuate the weaknesses of the discussed protocols, a group key management protocol in ad hoc networks should be adapted to the mobility and the dynamics in such type of network. The limited resources of energy, storage and computations capacities should also be taken into account in the design of this protocol. These requirements are the motivations for our group key management protocol in MANETs, called BALADE.

BALADE is an enhancement and adaptation of the DEP protocol, described in Section 3, to the ad hoc network context. Our protocol is based on the dynamic clustering approach, using one traffic encryption key and several key encryption keys (KEKs), one for each cluster. BALADE completely eliminates the overhead induced by encryption and decryption operations on the multicast flow, while attenuating the "1 affects n" phenomenon.

5 BALADE: Functional Architecture

5.1 Protocol Overview

BALADE is a scalable group key management protocol, for ad hoc networks, dedicated to secure multicast communications, that follows a sequential multi-sources model: at any time t, there is only one source which sends the multicast flow to the group members, and once it finishes, another source takes over. We wanted to focus on this model of group communications because it is very widespread in the Internet world, and corresponds to the characteristics of several applications in MANETs, such as audio/video conferences, MP3 broadcasting, podcasting, . . .

The basic idea of BALADE is to divide the multicast group dynamically into clusters. Each cluster is managed by a local controller which shares with its local members a local cluster key. The multicast flow is encrypted by the source with the traffic encryption key TEK and sent in multicast to all the group members. The source of the group and the local controllers form a multicast group GLC (Group of Local Controllers) and share beforehand a session key called KEK_{CCL}. Each new local controller has to join this group and receive the session key KEK_{CCL} from the source of the group, encrypted with its public key.

For the TEK distribution, the multicast source sends it to the group of the local controllers, encrypted with KEK_{CCL}. The local controllers forward the TEK to their local members, encrypted with their respective local cluster key. One main advantage of BALADE is that only the TEK is encrypted and decrypted by the local controllers and not all the multicast flow. The decryption of the multicast data is only achieved by the final receivers.

To ensure the integrity and the confidentiality of the multicast flow, a TEK re-keying process must be triggered by the source, at each data semantic unit, depending on the application. For example, a source multicasting a MP3 flow will renew its TEK after every song, and a source streaming a video flow will renew its TEK after every film or chapter. Therefore, the TEK is not changed for each membership event (Join or Leave) but for each data unit, specific to the application. Moreover, a leave in an ad hoc environment is often different from an exclusion; members can leave the group because of signal interferences or low resources but still remain in the multicast group.

This solution is realistic and pragmatic because it considers the practical requirements of the application. In addition to the group key management, BALADE proposes a management for the dynamics and the nodes mobility, adapted to the characteristics of the ad hoc networks.

In the following, we present the different security and management services achieved by BALADE.

5.2 Group Members Management in BALADE

The group members management is achieved dynamically by BALADE. The main actors of this architecture, illustrated in Figure 7, are:

- The Global Controller (GC) is the source of the multicast group. Within a sequential multi-source architecture, and at any time, there is only one GC in the multicast group.
  This entity is responsible for the generation of the traffic encryption key (TEK) and for the encryption and the distribution of the secure multicast data to the group members. The GC ensures also the renewal of the TEK, at each unit of the multicast data, according to the semantic of the application flow.

- The Local Controller (LC) is a group member, forming with its local members a cluster. A local controller must generate and distribute a local key to its local members, called KEK_{CSG}. Each LC holds a list of its local members (LPL) and multicasts the traffic encryption key, sent by the source of the group, to the members of this list, encrypted with the KEK_{CSG}; the secure multicast flow is then sent separately. The KEK_{CSG} is renewed by the local controller, after each event of join or leave in the cluster.

  The local controllers, managing the clusters of the multicast group, know the traffic encryption key, and consequently are able to decrypt the multicast data. For this reason, a local controller must imperatively be a member of the multicast group. In addition, the local controllers are ad hoc nodes, and it is not reasonable that a non member ad hoc node ensures the role of a local controller for a group.

  A local controller must obtain the permission to form and manage its cluster, from its parent controller, which checks the authenticity of the concerned local controller and its membership to the multicast group. In case of authentication success, the parent controller authorizes the local controller to join the group of local controllers and sends the ACL (Access Control List) and RL (Revocation List) lists to it.

- The Group Member (GM) is a member of the list of nodes, authorized to join the multicast group. A member of the multicast group can switch to the local controller state, via an algorithm of election of local controllers, described in Section 5.4.

5.3 Keys Management and Distribution

The source of the group starts by encrypting data with the traffic encryption key and sending it to all group members, according to the multicast tree established by the multicast routing protocol. Initially, all the group members belong to the cluster managed by the source. Thus,

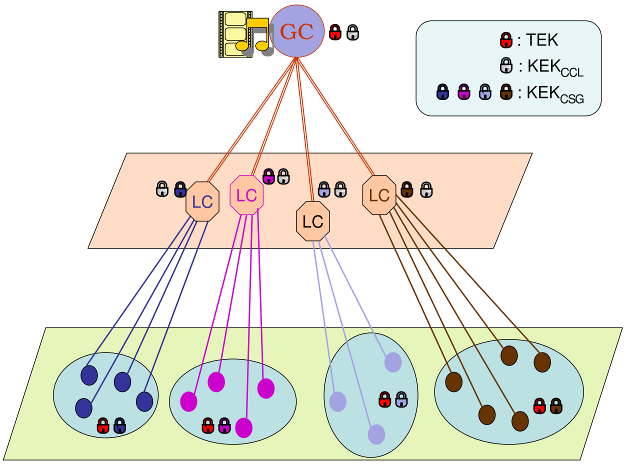


Figure 7: Members management in BALADE

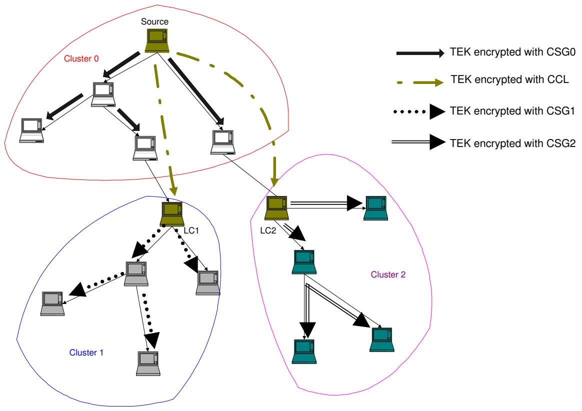


Figure 8: TEK distribution process

they receive the session key in unicast, called KEK_{CSG-0} (key of the sub-group 0), encrypted with their respective public keys. Then, dynamically, new clusters will be created. Each cluster i is managed by the local controller LC_i and shares a local cluster key KEK_{CSG-i}, managed by the LC_i.

For re-keying, the source of the group multicasts the TEK to the members of its cluster, encrypted with KEK_{CSG-0} and to the group formed by the local controllers, encrypted with their key KEK_{CCL}. The local controllers, decrypt the message, extract the TEK, re-encrypt it with their respective local cluster keys and send it to their local members. Thus, our solution attenuates the encryption and decryption processes for the local controllers, which have just to decrypt and re-encrypt the traffic encryption key. We note that each source must join the group formed by the local controllers, to be able to send the TEK encrypted with the KEK_{CCL} to the others LCs. When the source of the group switches from a node to another, the TEK distribution tree remains potentially little impacted. The KEK_{CCL} is managed by the current source, and is renewed after each join or leave of a local controller in the GLC. An illustration of the TEK distribution process is presented in Figure 8.

The renewal of the traffic encryption key TEK is triggered after each semantic unit of the multicast data, and follows the same process described above.

5.4 Dynamic Group Clustering: OMCT Algorithm

BALADE uses a dynamic clustering scheme to divide the multicast group into clusters and to elect the local controllers, according to their localization compared to the others group members. This scheme wants to optimize energy consumption and latency for key delivery. Being mobility aware, this algorithm needs the geographical location information of all the group members in the construction of the key distribution tree. Thus, we assume that within an ad hoc network, a *Global Positioning System* (GPS) is available.

During multicast group initialization, every group member is attached to the group source, called global controller (GC). This entity is responsible for the TEK generation and its distribution to all group receivers. In addition, the GC checks periodically whether the group is highly correlated, and consequently whether the key distribution process is optimal. The evaluation of the cluster cohesion is determined with a cluster cohesion parameter that we have defined as the centralization index of the cluster around the LC node:

$$\begin{aligned} Cohesion &= \frac{M}{C}, \quad \text{where:} \\ M &= \textit{members in the LC range}, \\ C &= \textit{Cluster members number}. \end{aligned}$$

This parameter measures the proximity of the cluster members compared to their controller. The bigger the number of members reachable by the controller in one hop, the closer the factor of cohesion reaches 1. With this cohesion parameter, we can verify whether a cluster is strongly correlated or not.

We define *Min_Cohesion* as the minimum threshold that a cohesion factor of a cluster should not exceed. Otherwise, the controller must take the decision to split the cluster, and the election of new local controllers LCs according to their localization, must be initiated.

The split process optimizes the number of new LCs while reaching every member of the cluster. This process is done by the OMCT algorithm [3] (Optimized Multicast Cluster Tree), whose principles are as follows:

- Clusters are highly correlated, ensuring that all the local members are reachable directly by their local controllers (one hop between local members and their local controllers);
- Two thresholds are fixed in OMCT, the maximum

and the minimum of the local members in each cluster, ensuring load balancing between clusters.

Once the first clusters are created within the multicast group, the new LCs become responsible for the local key management and distribution to their local members, and also for the maintenance of the strongly correlated clusters property.

Recursively, and at every moment of a multicast group session, the group is composed of strongly correlated clusters, ensuring that their respective cohesion is always higher than the defined minimal threshold $Min\_Cohesion$. Figure 9 contains an example of execution of the OMCT algorithm. Initially, the cohesion parameter of the cluster managed by node 1 was $\frac{4}{10}$, and after executing the OMCT algorithm, two new clusters are created, managed by the nodes 4 and 8, with cohesion parameter equal to 1.

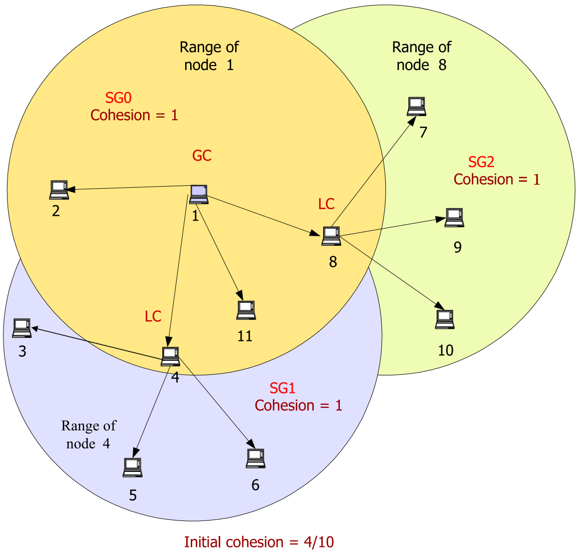


Figure 9: OMCT clustering algorithm

5.5 Mobility Management

Periodically, each LC computes the cohesion parameter of its cluster. According to the value of this parameter, it decides to execute or not the OMCT algorithm to clusterize its sub-group.

When a member moves within the network, it can be unreachable by its LC. All LCs send periodically "LC_Queries" messages containing their identities and their coordinates.

Thus, a member moving in the network, and receiving LC_Queries from LCs, chooses to join the nearest LC, according to its localization. Figure 10 illustrates this mobility scenario. Then, for its re-authentication and access control to the multicast group, this member uses a re-authentication ticket, which is a password encrypted with the traffic encryption key, and known by all the group members.

When a LC moves within the network, leaves the group or disappears due to any resources problems, it must previously, if possible, send a notification message to all its local members asking them for moving to others clusters having nearest LCs. Otherwise, if it is not possible to send a notification message, local members will, after a period of time, realize that they have lost their connectivity to the group, and then will attach to others clusters.

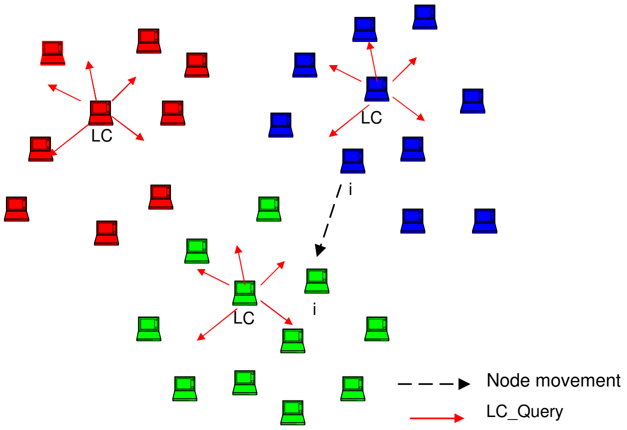


Figure 10: Nodes mobility

6 Conclusions

In this paper, we have shown that the group key management protocols dedicated to operate in wired networks, are not suited to ad hoc networks, because of the characteristics and the challenges of such environments.

Then, we presented a taxonomy of group key management protocols for securing multicast communications in ad hoc networks, considering the characteristics and the criteria of such environment, which are nodes mobility support, energy efficiency and multi-hop awareness. We discussed these protocols and compared them according to security and performance metrics, which are the security services (data confidentiality and integrity, nodes authentication and revocation, . . .), the storage cost, the vulnerabilities or the weaknesses and the scalability.

We presented afterwards BALADE, a group key management protocol dedicated to operate in ad hoc environments, to secure multicast communications, according to the sequential multi-sources model. BALADE is based on the dynamic clustering approach, using one traffic encryption key, and several key encryption keys, thus completely eliminating the overhead induced by the intermediate encryption and decryption operations on the multicast flow, while attenuating the "1 affects n" phenomenon.

BALADE uses the OMCT (Optimized Multicast Cluster Tree) algorithm to ensure an efficient and fast group key delivery, taking into account the localization and the mobility of nodes, and optimizing energy and bandwidth.

To validate the applicability of the BALADE protocol, we are implementing a cooperative jukebox application, playing MP3 streaming, within ad hoc networks. The application users are grouped in different sub-groups, and share a common and distributed playlist, from which any user can choose to add the songs that he/she wants to listen, as in a classical jukebox.

We are also validating the BALADE protocol, formally, by using the HLPSL [5] language (High Level Protocol Specification Language), and the AVSIPA[1] validating tool, in order to detect the possible security attacks and vulnerabilities, and consequently to correct them.

To improve the performance of BALADE, we plan to carry out the reliability of the keys distribution process, in ad hoc environment, where the packets loss rate is not negligible.

[1] http://www.avispa-project.org

References

[1] H. Bettahar, A. Bouabdallah, and Y. Challal, "An adaptive key management protocol for secure multicast," in *11th International Conference on Computer Communications and Networks ICCCN*, Florida USA, Oct. 2002.

[2] M. S. Bouassida, I. Chrisment, and O. Festor, "An enhanced hybrid key management protocol for secure multicast in Ad Hoc networks," in *Networking 2004, Third International IFIP TC6 Networking Conference*, LNCS 3042, pp. 725-742, Springer, May 2004.

[3] M. S. Bouassida, I. Chrisment, and O. Festor, "Efficient clustering for multicast key distribution in MANETs," in *Networking 2005, International IFIP TC6 Networking Conference*, LNCS 3462, pp. 138-153, Springer, May 2005.

[4] Y. Challal, H. Bettahar, and A. Bouabdallah, "SAKM: A scalable and adaptive key management approach for multicast communications," *ACM SIGCOMM Computer Communications Review*, vol. 34, no. 2, pp. 260-271, Apr. 2004.

[5] Y. Chevalier, L. Compagna, J. Cuellar, P. H. Drielsma, J. Mantovani, S. Mödersheim, and L. Vigneron, "A high level protocol specification language for industrial security-sensitive protocols," in *Workshop on Specification and Automated Processing of Security Requirements (SAPS)*, 2004.

[6] T. Chiang and Y. Huang, "Group keys and the multicast security in Ad Hoc networks," in *Proceedings of the 2003 International Conference on Parallel Processing Workshops (ICPP 2003 Workshops)*, pp. 385, 2003.

[7] W. Diffie and M.E. Hellman,"New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644-654, 1976.

[8] L. R. Dondeti, S. Mukherjee, and A. Samal, "Scalable secure one-to-many group communication using dual encryption," *Computer Communications*, vol. 23, no. 17, pp. 1681-1701, 2000.

[9] T. Hardjono and L. Dondeti, *Multicast and Group Security*, Computer Security Series, Artech House, 2003.

[10] I. Ingemarson, D. Tang, and C. Wong, "A conference key distribution system," *IEEE Transactions on Information Theory*, pp. 714-720, Sep. 1982.

[11] T. Kaya, G. Lin, G. Noubir, and A. Yilmaz, "Secure multicast groups on Ad Hoc networks," in *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 94-102, 2003.

[12] P. Korshunov, *Multicast Security In Ad Hoc Networks*, Technical Report HT030016M. (http://www.comp.nus.edu.sg/ cs4274/termpapers/0405-i/paper-13.pdf)

[13] L. Lazos and R. Poovendram, "Energy-aware secure multicast communication in Ad Hoc networks using geographical location information," in *IEEE International Conference on Acoustics Speech and Signal Processing*, pp. 201-204, 2003.

[14] J. B. MacQueen,' 'Some methods for classification and analysis of multivariate observations," in *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281-297, 1967.

[15] D. Mcgrew and A. Sherman, *Key Establishement In Large Dynamic Groups Using One-Way Functions Trees*, Technical report, no. 0755, May 1998.

[16] S. Mittra,"Iolus: A framework for scalable secure multicasting," in *SIGCOMM*, pp. 277-288, 1997.

[17] M. Moharrun, R. Mukkalamala, and M. Eltoweissy, "Ckds: An efficient combinatorial key distribution scheme for wireless Ad Hoc networks," in *IEEE International Conference on Performance, Computing and Communications (IPCCC'04)*, pp. 631-636, Apr. 2004.

[18] L. Morales, I.H. Sudborough, M. Eltoweissy, and M.H. Heydari, "Combinatorial optimization of multicast key management," in *International Conference on System Sciences IEEE*, pp. 332, Jan. 2003.

[19] A. Perrig, R. Canetti, D. Tygar, and D. Song,' 'The TESLA broadcast authentication protocol," *RSA Laboratories Cryptobytes*, vol. 5, no. 2, pp. 2-13, 2002.

[20] R. D. Pietro, L. Mancini, Y. Law, D. Etalle, and P. Havinga, "Lkhw: A directed diffusion based secure multicast scheme for wireless sensor networks," in *International Conference on Parallel Processing Workshops (ICPPW'03)*, pp. 397-406, Oct. 2003.

[21] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker, "A scalable content-addressable network," in *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 161-172, 2001.

[22] V. Varadharajan, M. Hitchens, and R. Shankaran, "Securing Ntdr Ad-Hoc Networks," in *IASTED International Conference on Parallel and Distributed Computing and Systems 2001*, pp. 593-598, Aug. 2001.

[23] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," in *INFOCOM 2000*, pp. 585-594, 2000.

[24] C. Wong, M. Gouda, and S. Lam, "secure group communications using key graphs," in *ACM SIGCOMM*, pp. 68-79, 1998.

[25] L. Zhou and J. Haas, "Securing Ad Hoc networks, "*IEEE Network*, vol. 13, no. 6, pp. 24-30, 1999.

[26] S. Zhu, S. Setia, S. Xu, and S. Jajodia, *GKMPAN: An Efficient Group Rekeying Scheme for Secure Multicast in Ad Hoc Networks Technical report*, Feb. 2004.

Mohamed Salah Bouassida is a researcher at HEUDIASYC laboratory in France. He has a Ph.D. degree (2006) from Henry Poincaré University, Nancy France, within the MADYNES research team in the LORIA laboratory. He obtained his master degree (2003) from the Henry Poincaré University, Nancy France and his engineer diploma (2002) from the National School for Computer Studies (ENSI), Tunis Tunisia.

His main research interests are the localization within wireless networks, the security services of group communications in the context of ad hoc networks, the establishment of group key management protocols within MANETs, taking into account the energy and bandwidth limitations while optimizing the keys delivery process. (Mohamed-Salah.Bouassida@hds.utc.fr)

Isabelle Chrisment is an assistant professor in Computer Science at the ESIAL engineer school, Henri Poincaré University, Nancy, France. She has a Ph.D. degree (1996) from Nice-Sophia Antipolis University, France and an Habilitation degree (2005) from Henri Poincaré University, Nancy, France. She achieves her research work in LORIA laboratory within the INRIA project called MADYNES whose activities are related to the management of dynamic networks and services. Her main research interest is the security service within the context of group communication and more especially the defintion of new key distribution protocols suitable for group communication. She is presently working on how to adapt group key management protocols to the ad hoc environment. (Isabelle.Chrisment@inria.fr)

Olivier Festor is a research director at INRIA. He has a Ph.D. degree (1994) and an Habilitation degree (2001) from Henri Poincaré University, Nancy, France. His research interests are in the design of algorithms and models for automated and scalable management for highly dynamic environments. Member of the IRTF NMRG, he has published more than 70 papers in network and service management and serves in the technical program and organization committees as well as in the editorial boards of several international conferences and journals. Olivier Festor was the TPC Co- chair of IM'2005. He is currently leading the EMANICS European Network of Excellence dedicated to Management Solutions for the Internet and Complex Services. (Olivier.Festor@inria.fr)