Analysis of Communication Overhead for a Clustering-Based Security Protocol in Ad Hoc Networks

C. Maghmoumi
University of Haute-Alsace
F-68000 Colmar, France
chadi.maghmoumi@uha.fr

H. Abouaissa
University of Haute-Alsace
F-68000 Colmar, France
abdelhafid.abouaissa@uha.fr

J. Gaber
Belfort University
F-90000 Belfort, France
jaafar.gaber@utbm.fr

P. Lorenz
University of Haute-Alsace
F-68000 Colmar, France
pascal.lorenz@uha.fr

***Abstract*— In this paper, we study and evaluate the overhead for a security algorithm based on clustering in MANET networks. The analysis of the communication overhead in Ad Hoc networks is an important issue because it affects the energy consumption and the limited battery life time of the mobile nodes. The algorithm partitions the network into clusters based on affinity relationships between nodes and two types of keys which are generated by a clusterhead. The first one is shared by a clusterhead and its local members and the second one is shared by the clusterhead and its parent cluster. The performance evaluation and communication overhead analysis of the proposed protocol are presented using simulation.**

Keywords: ad hoc networks, clustering, security, key management.

I. INTRODUCTION

Moving from wired networks to wireless mobile networks leads to use pervasive networks with new network-computing challenges. Ubiquitous Computing (UC) is a concept that deals with providing available services in a network by giving users the ability to access services anytime and irrespective to their location. Pervasive Computing (PC) is often considered as the same as ubiquitous computing but the main objective in PC is to offer spontaneous services created on the fly by mobiles nodes that interact with ad hoc connections [1].

Mobile ad hoc networks (MANET) are autonomous systems created by mobile nodes without any infrastructure. Currently, MANET has gained popularity in mobile pervasive applications, such as electronic business, emergency teams, etc. These applications support group communications, auto-adaptive discovery and composition services. Most of research on the pervasive communications in MANET mainly focused on admission control and resource management (like bandwidth, energy consumption, interferences, etc.) to perform the communication in these mobile networks. However, in these types of applications, secure group communications is very critical and is a major concern.

Clustering in MANET is a challenging issue because of the dynamic network topology changes. Clustering algorithm partitions a network into different clusters, creating a network hierarchy in the network. In general, clustering algorithms can be divided into cluster formation stage and cluster maintenance stage. A particular node is elected in a cluster to manage the cluster information is known as the clusterhead, and the other nodes are its members.

In MANET, to ensure a confidential communication between two or several mobile nodes, traffic can be encrypted and only receivers can decrypt data [2, 3]. Furthermore, MANET may be highly versatile, involving short-lived communications between nodes that may never have met before, thus complicating the initial trust establishment and trust maintenance. Thus, new solutions should be introduced to support efficient and secure group communication in mobiles pervasive networks with respect to the dynamic network topology induced by the node mobility and unreliable communication. Also, this type of network does not have any trust node for key management, like a central reference, to ensure the message encryption/decryption. This cannot actually satisfy MANET dynamic environments. To solve this problem, one of the approaches is to share a secret key called "group key" [4]. When a member joins a group, the group key is rekeyed to ensure that the new member cannot decrypt previous messages, a security requirement known as backward secrecy [5]. When a member leaves the group, the group key is re-keyed to ensure that future communications cannot be decrypted by the leaving member, a security requirement known as forward secrecy.

In MANET, it is not easy to control mobile members of a cluster and the frequency of their adhesions. The security algorithm must support the mobility problem and the clusters' scalability. Therefore, to solve these problems and ensure trusted communications in a MANET environment, the major solution is to introduce an efficient key management algorithm, adequate to manage and distribute keys to cluster members in order to encrypt/decrypt multicast

data. An efficient security algorithm should provide a rapid re-keying process and be adaptive to frequent topology changes.

The analysis of communication overhead in Ad Hoc network is related to different parameters, e.g. network size, node mobility, node transmission range and network density. An efficient clustering and key management algorithm must support all these network parameters in order to minimize the messages overhead.

The rest of the paper is organized as follows. Section II overviews the related work. In section III, we present an overview on security in ad hoc networks. Section IV introduces the proposed key management protocol. Section V specifies proofs for the proposed protocol. In Section VI, we implemented and evaluated the proposed protocol and section VII draws conclusions and future works.

II. RELATED WORK

Recently, many clustering algorithms have been proposed for mobile ad hoc networks in order to improve the efficiency of routing protocols and save energy or to implement efficient flooding and broadcasting mechanisms. Haddad and Kheddouci presented in [6] a classification of topology-based approaches to define an efficient organization over the network to optimize communication protocols for routing, service discovery, resource sharing and management.

Many group security algorithms or protocols have been proposed for MANET in the literature. They can be divided into two categories: centralized and distributed protocols [11]. In centralized protocol, only a single node controls all the other nodes. Therefore, the re-encryption process is managed only by this node. This protocol can optimize network resources. However, since there is only one key manager in the group, it is probable that this node breaks down [12, 13]. In [14], the authors proposed two key agreement protocols based on the threshold cryptography using the Lagrange interpolation theorem. This approach seems theoretically efficient; however, it focuses only on a special case of scenario. In [7], a hierarchical protocol based on multicast source key is proposed. The source node provides keys to its local members and to groups' leaders. A new node that will be joined to a group should negotiate with group's leader, then, the latter informs the source node to get a new key. The only role of leaders is to manage received keys from the source node. Although, the protocol secures the network, its complexity is high due to multiple key's generation to maintain group communication security. Luo et al. [17], [18] chose a different method to distribute the certification process. They use a specially crafted key sharing algorithm distributing the key amongst all nodes instead of a subset only. Upon this, Luo et al. build an access control system based on signed tickets issued by neighbors of the node seeking access.

In [15], the authors proposed an analysis of the overhead involved in clustering for one-hop clustered ad hoc networks. This analysis captures the effects of different network parameters, e.g. node mobility, node transmission range, and network density on the amount of overhead that clustering algorithms may incur in different network environments. But the authors in this analysis focused mainly on the cluster maintenance stage and only one-hop clustering algorithms are considered. In [16], the authors introduced a cluster-based architecture for a distributed public key infrastructure. This architecture is adapted to the highly dynamic topology and varying link qualities in ad hoc networks but the overhead is very high.

The proposed protocol in this paper differs from previous studies in three ways:

1) We don't require any centralized key control component to manage and distribute keys. Encryption keys are generated by clusterhead and re-encrypted by participating sub-clusterheads.

2) A dynamic clustering algorithm that is adaptive to frequent topology changes is used.

3) Since the key distribution process is totally decentralized and the keys are shared by different communication groups, the proposed protocol can be used to build a generic security service for multicast communication.

III. SECURITY IN AD HOC NETWORKS

The security of a multicast group requires that only group members can access the data transmitted by the source, even if these data are diffused in the network. To ensure this confidentiality, a symmetric key is used by the source to encrypt data, and by the members to decrypt data. This key is called TEK (Traffic Encryption Key). The life of a session of a secure group is represented by a set of time intervals, each interval is defined by a change in the status of the group (join or leave a member) as shown in figure 1. To preserve data confidentiality of the group, it is necessary to renew the encryption key after each event (join or leave the group). A member who leaves the group should no longer be able to decrypt data.

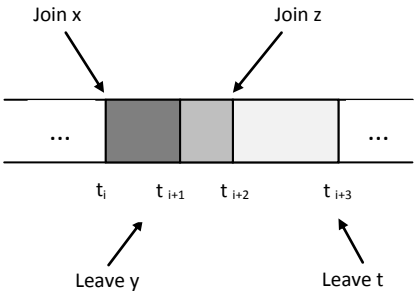

Figure 1. Life's changes of a secured group.

A. Classification of group key management approaches

Several architectures for group key management in networks have been proposed and developed; we can classify them into three approaches according to the number of TEKs used as shown in figure 2.

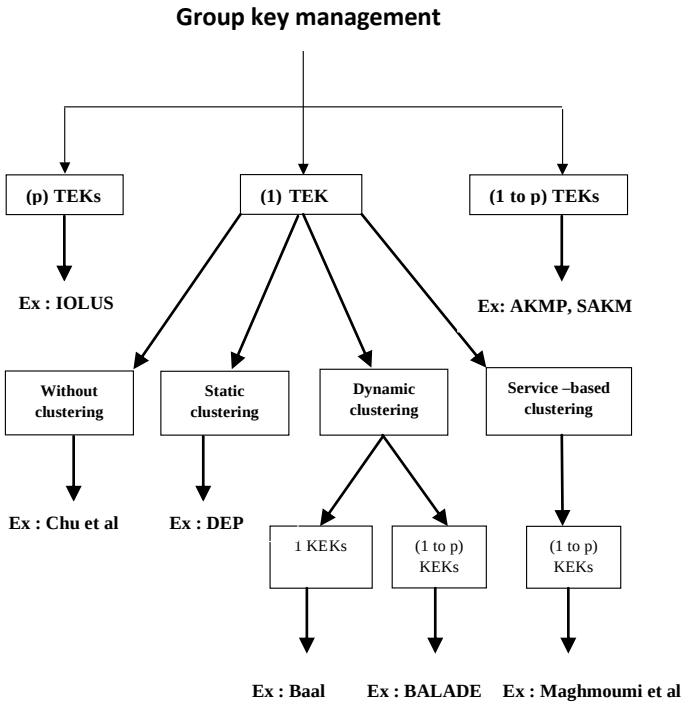

Figure 2. Classification of group key management protocols.

In the approach p TEK (Traffic Encryption Key), each subgroup shares a local TEK generated by a local controller, p is the number of sub-groups of the multicast group. This hierarchical approach is to address the problem 1 affects n. At each arrival/departure to/from the group, only the subgroup affected by this change will change its local TEK. In the approach (1 to p TEKS) the protocol begin with a centralized approach (1 affects n) and dynamically switches towards a subdivision of the group into subgroups (p TEK). The approach (1 TEK) is to share a one encryption key TEK used by the source and members in order to encrypt/decrypt multicast data, this approach can be used in a centralized architecture (no clustering) or hierarchical (static or dynamic clustering). Table 1 shows the classification of different protocols from the literature.

Protocol	Number of KEK	Clustering
IOLUS	p	static
AKMP	1 to p	dynamic
SAKM	1 to p	dynamic
Chu et al	1	no clustering
DEP	p	static
Baal	1	dynamic
BALADE	1 to p	dynamic
Maghmoumi et al	1 to p	dynamic

Table. 1. Comparison of group Key Management protocols

IV. CLUSTERING BASED SECURITY PROTOCOL

A. Clustering approach

As the topology of a MANET changes, clustering messages are generated by nodes to update a node of changes to its cluster members or clusterhead. The execution of clustering algorithms can usually be divided into cluster formation stage and cluster maintenance stage [20, 21]. Different clustering protocols may use different schemes and generally there are three types of clustering messages:

a) **Join message**, for nodes to know the neighbor's identities. The HELLO message is often used.

b) **Acknowledgement message** to accept new node in the cluster.

c) **Leave message** to remove a node from a cluster.

In what follows, for the clustering overhead analysis, we denote the network size by N, a cluster size by NC_i (the number of members in the cluster C_i), the network density by ρ, and the transmission range is r. The average cluster size NC_i is given by $NC_i = N/n$ where n is the number of clusters in the network.

Two properties for clustered networks should be ensured and any violation will trigger clustering messages at relevant nodes [15]:

P1. No cluster-heads directly connected to each other.

P2. Each node should belong to only one cluster.

The main idea underlying this protocol is to divide the ad hoc network into clusters according to affinity relationships between involved nodes [8] and uses the Key Management Protocol proposed initially in [9]. Once the clusterhead is selected, it handles two KEKs (Key Encryption Key), one shared by clusterhead and its local members, and the second is shared by the clusterhead and the parent cluster.

Affinity relationships between the nodes can be determined according to the services they provide. A service can be described by four main parts [8]:

1) the attributes.
2) the capsule.
3) constraints and requirements.
4) set of relevant semantic keywords.

Attributes contain the characteristic of the service, such as operations that can be invoked and their input and output parameters. The capsule includes information about the service localization, the invocation protocol and the port number. The constraints and requirements give information about the resources needed to execute the service. The set of semantic keywords are used by for matching relevant keywords to each nearby service. [1].

Ad hoc networks are characterized by the node's mobility of nodes; several nodes can move with different speeds. Our goal is to form stable clusters; in this case, we set a given threshold to separate the clusters formed by high-speed or slow nodes. In basic ad hoc networks, nodes can exchange [RTS, CTS, DATA, and ACK] messages, via a complete virtual graph, in order to guarantee group self-stability by the homogeneous mobility of nodes and thus ensure a reliable communication between wireless mobile nodes.

In this protocol, the source node generates a TEK encrypted in a Key Encryption Key KEK_i that should be sent to its local members. Once, each clusterhead receives the encrypted data, it decrypts it and re-encrypts it with its own KEK_j, then forwards it to its descendents. The join or leave events within each cluster results in the KEK_j re-keying by the clusterhead. Therefore, the proposed protocol belongs to the dynamic clustering algorithm with one TEK and *1* to *p* KEK, where *p* is the number of clusters that constitute the ad hoc network. That makes it possible to optimize the cost of data encrypting and decrypting processes and to reduce the 1 affects n phenomena [10].

The clustering-based key management protocol consists of two tasks:

1) Cluster Formation

- The cluster formation starts when a node N_i boots up and sends a cluster join message JOINreq to its neighbors. This message contains the description of a service D (S_i) and a number ID_i that identifies this service.

- When a node N_j receives the message JOINreq (D (S_i), ID_i), it examines the compatibility of the service S_i with a service S_j using MATCH (D (S_i), D (S_j)) and sends a response message Rep to N_i that contains the rate of available energy f(E_j) expressed by (1) [19]:

$$f(E) = \frac{E_{max} - E_{cons}}{E_{max}} \qquad (1)$$

$$E_{cons} = E_{cons} + E_{req} + \varepsilon \quad ; \varepsilon \geq 0$$

Where E_{max} is the maximum energy of the node, Econs is is the energy consumed, E_{req} is the energy required to transmit a packet and ε is the energy that can be lost in the environment due to factors not anticipated [22].

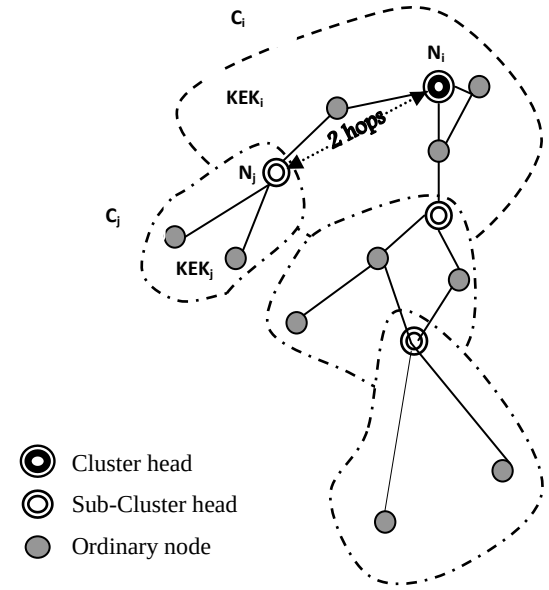

Figure 3. Ad hoc network partitioned into in 4 clusters

- When the node N_i receives the response message Rep (ID, f (E_j)), it verifies the RTT (Round Trip Time) and the f(E_j) and sends back an acknowledgement message ACK containing a flag that indicates the validation of theses parameters. Then, it adds the node N_j to the list of the cluster Ci's members that consider N_i as clusterhead (the first member of the list). Once the clusterhead is chosen, it generates a KEK that should be sent to its local members.

- If a member N_j is at two hops from the clusterhead and if there are at least two nodes belonging to the cluster via N_j, then N_j becomes a clusterhead for a sub-cluster that contains these nodes. It generates thus a KEK_j for its own cluster as shown in figure 3.

The formal description of the protocol process is described in figure 4.

When a node N_j receives JOINreq (D (S_i), ID_s) **then**
m_{ij} = MATCH(D (S_i),D (S_j))
if $m_{ij} \geq \sigma$ **then**
send RepN (ID_i, ID_s,f(E))
fi

When a node N_i receives RepN (ID_i, ID_s,f(E)) **then**
if RTT $\leq \beta$ **then**
if $f(E) \leq \alpha$ **then**
send ACK (ID_j, ID_s, non)
else
send ACK (ID_j, ID_s, ok)
$CL_i = \{CL_i \cup N_j\}$
Send ($\{CL_i\}$, KEK_i)
fi
fi

When a node N_j receives ($\{Cl_i\}$, KEK_i) **then** $CL_j = CL_i$

When a node N_j is at 2 hops from the clusterhead **then**
if $|H| \geq 2$ **then**
$CL_j = \{N_j \cup \{H\}\}$
send ($\{CL_j\}$, KEK_j)
fi
// $N_i \in \{H\}$ => N_i is at 3 hops at least from the clusterhead // and N_i received the ACK via N_j

When a node N_i receives LeavN_j (ID_s) **then**
$CL_i = \{CL_i \setminus N_j\}$
send($\{Cl_i\}$, KEK_i)

When a node N_j receives LeavH_i (ID_s) **then**
If $N_j \in \{CL_i\}$ then
send JOINreq (D (S_j), ID_s)
else
$CL_j = \{CL_j \setminus N_i\}$
send ($\{CL_j\}$, KEK_j)

Figure 4. Clustering-based security protocol

Consequently, each clusterhead handles two KEKs:
1. KEK_i: shared between clusterhead and its local members.
2. KEK_j: shared between the clusterhead and its parent cluster.

2) Cluster Maintenance

- When a member leaves the cluster C_i, it sends a leave message to the clusterhead which removes it from its list of nodes CL_i, regenerates a new KEK and transmits it to its local members except the departing member. In this step the number of messages sent is $NC_i - 1$.Where NC_i is the number of nodes in the cluster C_i.
- If a clusterhead N_j leaves the cluster, it sends to its members a leave message in addition to its parents cluster members. When the clusterhead of the parent cluster N_i receives this message, it removes it from the list, regenerates a new KEK_i and transmits it to its local members except the departing member. Each member of the leaving clusterhead send join message JOINreq to its neighbors. In this step the number of messages sent is $NC_i + NC_i - 2$.

The total number of messages is thus:

$$M_{leav} = 2NC_i + NC_j - 3 \quad (2)$$

- When a new node N_j joins a cluster, it sends messages to its neighbors. The clusterhead that accepts this node regenerates a new KEK for its new list of nodes. The number of messages sent in this step is:

$$M_{join} = NC_i + \mu \quad (3)$$

Where μ is the expected number of network neighbors of a randomly node that depends on ρ, r and N.

The figure 5 shows the process of joining the cluster and the figure 6 shows the process of leaving the cluster.

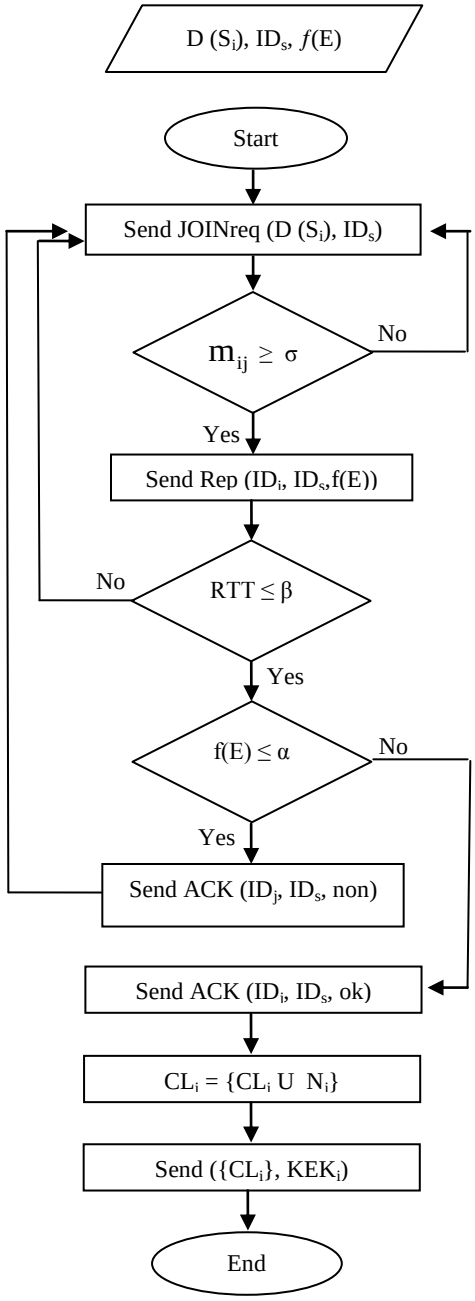

Figure 5. Process of joining the cluster

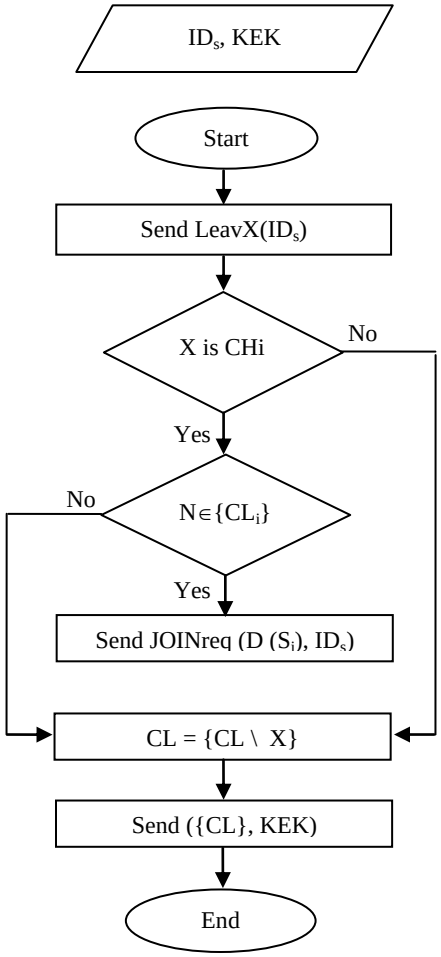

Figure 6. Process of leaving the cluster

B. Correctness proof

In this part, we explain the expected value of the number of network neighbors of a randomly chosen node μ and we analyze the capabilities of our protocol that ensures communication trust when a node joins or leaves the cluster.

Lemma 1:

The expected number of network neighbors of a randomly selected node is:

$$\mu = O((N-1)\frac{r^2}{S}) \qquad (4)$$

Proof:

The expected number of network neighbors of a randomly selected node is given in [15]:

$$\mu = (N-1)\frac{r^2\rho}{N}\left(\frac{r^2}{2}\frac{\rho}{N} - \frac{8r}{3}\sqrt{\frac{\rho}{N}} + \pi\right)$$

within an area S with density ρ, we obtain equation (4)

Lemma 2:

The new node that joins the cluster cannot decrypt past encrypted data.

Proof:

Assume that a new node N_i sends a cluster join message to clusterhead. N_i cannot decrypt messages because any node cannot decrypt data as long as it does not receive the acknowledgment from its clusterhead. In fact, when the clusterhead receives a new join message, it updates the list of members and regenerates a new key KEK_i, then sends it to its local members. The proof could drive to that every node must have a KEK key to decrypt and encrypt data traffic which proves that the proposed protocol guarantees the backward secrecy.

Lemma 3:

The node which leaves the cluster cannot decrypt the future data.

Proof:

Leaving of ordinary node from a cluster is uncomplicated. The node sends a leave message to the clusterhead that leaves this node from the list of members and regenerates a new KEK broadcasted to all local members in the new list. However, when a clusterhead wants to leave the network, it must inform the upper clusterhead to re-encryption its key and secure the data transmission of the upper cluster. Also, nodes belonging to the same clusters should re-construct a new key. Therefore, the forward secrecy is guaranteed.

Theorem:

In Ad Hoc networks, the security in multicast communications is guaranteed.

Proof:

From lemma 2, it is proved that backward secrecy is guaranteed. In lemma 3, we have proved that the forward secrecy is guaranteed. Therefore, the security in multicast communications is guaranteed.

V. SIMULATION RESULTS

From the equations (2) and (3), we can calculate the total number of KEK messages sent during the clustering for one cluster (n=1):

$$M_{total} = M_{join} + M_{leav}$$

$$\Rightarrow \quad M_{total} = 3\mathrm{NC}_i + \mathrm{NC}_j + \mu - 3$$

⇨ The total number of KEK messages sent for n clusters is:

$$M_{total} = \sum_{1}^{n-1} 3\mathrm{NC}_i + \mathrm{NC}_j + \mu - 3 \qquad (5)$$

The aim of simulations we have performed is to study the impact of transmission range of nodes r and density ρ on cluster formation and KEK messages overhead. We are also interested in studying the impact of the number of clusters with respect to the number of nodes on the number of KEK messages sent in ad hoc network.

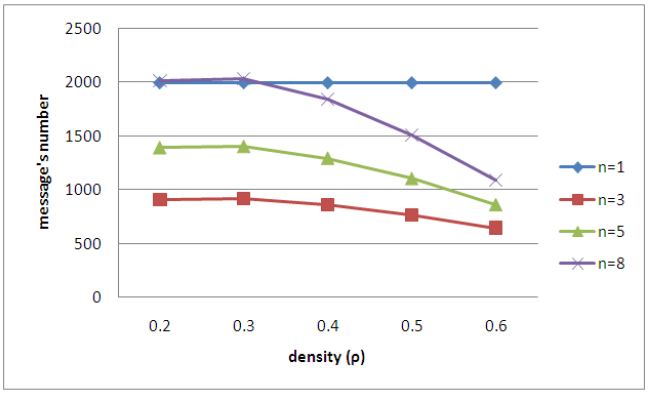

Figure 7. Number of KEK messages sent with and without clustering for N = 200 nodes and r = 30 m

The first simulation is performed (figure 7) with 95 nodes for a transmission range r = 22 meters and the network density ρ is varied from 0.2 to 0.6 (the number of nodes per unit area). In the second simulation (figure 8), we increased the number of nodes to N = 200 for a transmission range r = 30 meters with the same variation of density (ρ = 0.2, 0.3, 0.4, 0.5, 0.6). The two simulations are evaluated for a different number of clusters (n = 3, 5.8) and compared with the case where there is one cluster (n = 1) or we can say that there is no clustering in the network. The two figures (7, 8) show that when the number of clusters increases, the number of KEK messages will be decreased because each node joins or leaves the network affects a single cluster. The case (1 affect N) has been avoided because each node in this case affects only K nodes (where k is the number of nodes in a cluster).

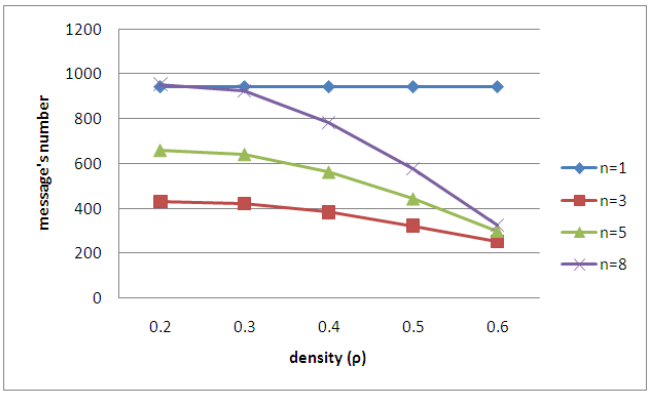

Figure 8. Number of KEK messages sent with and without clustering for N = 95 nodes and r = 22 m

The simulation results show also the benefits of using clusters for the management and maintenance of keys. Increasing the density allows us to have clusters with very high cardinality, which reduces the number of KEK messages in ad hoc network and ensures efficient key management.

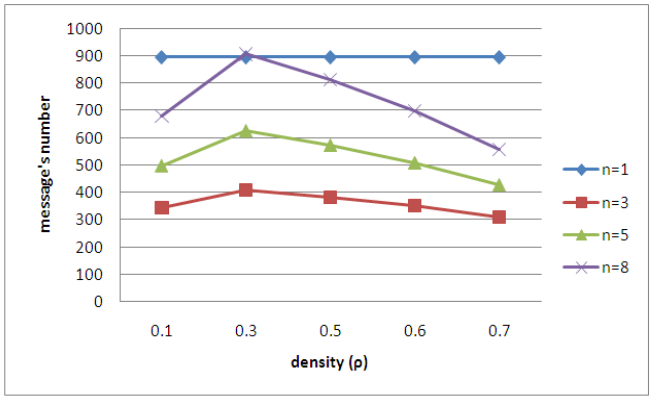

Figure 9. Number of KEK messages sent with and without clustering for N = 95 nodes and r = 18 m

In this part of simulation, the aim is to study the impact of transmission range of nodes r and density ρ on cluster formation and KEK messages overhead. The number of nodes is fixed at 95 nodes with a transmission range varied (r = 18, 20.22). We observed for each simulation the change in the number of KEK messages sent during cluster formation for several values of density ρ.

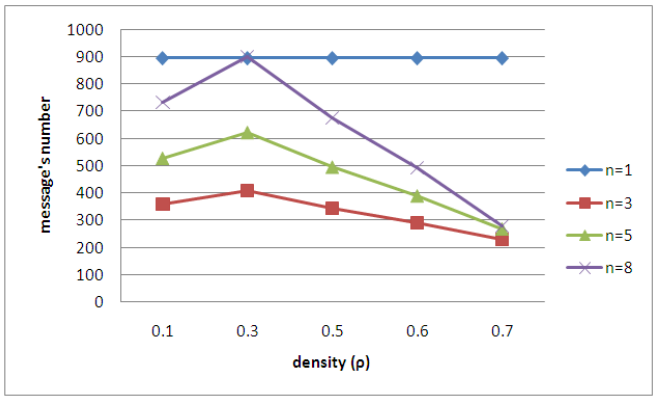

Figure 10. Number of KEK messages sent with and without clustering for N = 95 nodes and r = 20 m

Figures 9, 10, 11 show that when the transmission range of node r increases, the number of KEK messages decreases. Similarly, getting clusters with a huge amount of nodes and wide coverage will increase the probability of staying these node within the cluster, and this will lead up to decrease the number of KEK messages and increase the stability of the cluster.

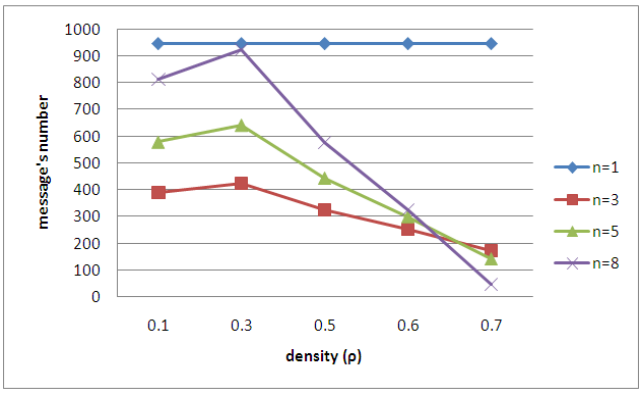

Figure 11. Number of KEK messages sent with and without clustering for N = 95 nodes and r = 22 m

In this part of simulation, we fixed transmission range of nodes r = 20 m and we increase the number of nodes with the same change of the number of clusters (n = 1, 3, 5, 8). The aim is to check the impact of the number of nodes with the number of clusters on KEK messages overhead.

In the figures 12, 13, the simulations show that when we used 95 nodes and 100 nodes with a different number of clusters (n = 3, 5, 8), the number of KEK messages decreased, but when we used 250 nodes with the same number of clusters, the number of KEK messages increased over a number of KEK messages sent in the network without clusters (n = 1) as shown in figure 14. So the number of clusters must be compatible with the number of nodes in the ad hoc network.

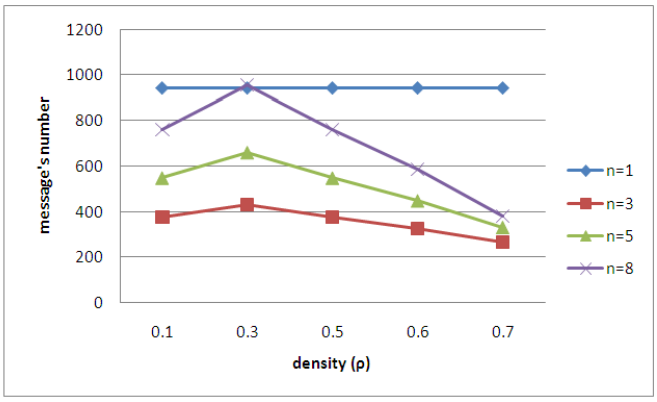

Figure 12. Number of KEK messages sent with and without clustering for N = 95 nodes and r = 20 m

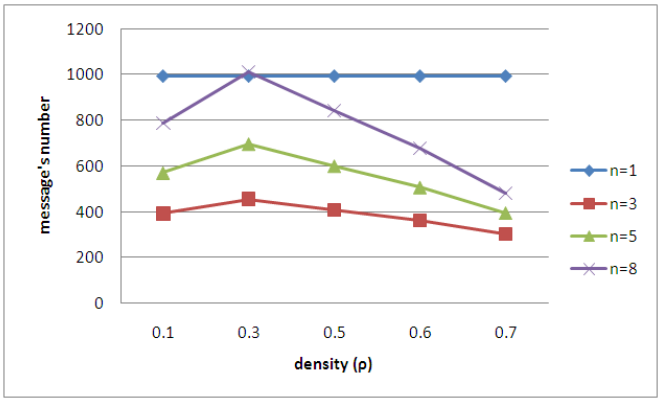

Figure 13. Number of KEK messages sent with and without clustering for N = 100 nodes and r = 20 m

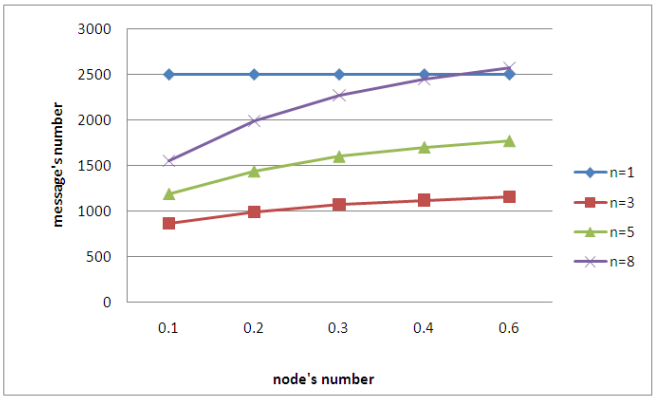

Figure 14. Number of KEK messages sent with and without clustering for N = 250 nodes and r = 20 m

VI. CONCLUSION

In this paper, we have presented an analysis of communication overhead for a security protocol based on dynamic clustering in ad hoc networks. The main idea of this protocol is based on the affinity relationships between the nodes for cluster formation, when a node is chosen as clusterhead, it generates two *KEKs*. The first key is shared by clusterhead and its local members and the second key is shared by the clusterhead and its parent cluster. The proposed protocol is scalable for large and dynamic multicast groups. For evaluate the performance of the proposed protocol, we have calculated the number of KEK messages sent during protocol steps and we have performed several simulations for analysis of communication overhead, we have studied the impact of transmission range of nodes r, density ρ and the number of clusters with the number of nodes on KEK messages overhead in ad hoc network. The work presented provides a good basis for further analysis on the performance of clustering protocols for MANET networks.

In future work, the communication overhead analysis will be investigated and compared with different clustering based protocols.

VII. REFERENCES

[1] J. Gaber, "Spontaneous Emergence Model for Pervasive Environments," Proc. IEEE GLOBECOM Workshop. Washington, November 2007.

[2] F. Stajano and R. Anderson, "The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks," Proc. 7th International Workshop on Security Protocols, 1999.

[3] F. Stajano, "The Resurrecting Duckling: What Next?," Proc. 8th International Workshop on Security Protocols, B. Crispo, M. Roe, and B. Criso, Eds., Lecture Notes in Computer Science, Vol. 2133, Berlin: Springer-Verlag, April 2000.

[4] C. K. Wong, M. Gouda, and S. S. Lam, "Secure group communications using key graphs," Proc. IEEE/ACM Transactions on Networking, vol. 8, 2000, pp.16–30.

[5] Y. Kim, A. Perrig, and G. Tsudik, "Tree-based group key agreement," ACM Transactions on Information and System Security, vol. 7, 2004, pp.60–96.

[6] M. Haddad and H. Kheddouci, "A survey on graph based service discovery approaches for ad hoc networks," Proc. IEEE International Conference on Pervasive Services. Istanbul, 2007.

[7] M. S. Bouassida, I. Chrisment, and O. Festor, "Validation de BALADE. INRIA research rapport N 5896, April 2006.

[8] M. Bakhouya and J. Gaber, "An affinity-driven clustering approach for service discovery and composition for pervasive computing," Proc. IEEE International Conference on Pervasive Services, Lyon, France, 2006.

[9] N. Kettaf, A. Abouaissa, P. Lorenz, and H. Guyennet. "A self organizing algorithm for ad hoc networks, " In Proceedings of the 10th IFIP Int. Conf. on Personal Wireless Communication, Colmar, France, August 2005.

[10] Y. M. Tseng, C. C. Yang, and D. R. Liao, "A Secure Group Communication Protocol for Ad Hoc Wireless Networks," Advances in Wireless Ad Hoc and Sensor Networks and Mobile Computing, Book Series "Network Theory and Applications," Springer 2006.

[11] K. C. Chan and S. H. Chan, "Key Management Approaches to Offer Data Confidentiality for Secure Multicast," Proc. IEEE Journal on Network, pp. 30-39, October 2003.

[12] C. C. Chang and C. Y. Chung, "An Efficient Session Key Generation Protocol," Proc. IEEE International Conference on Communication Technology, Beijing, China, pp. 203-207, April 2003.

[13] I. R. Chen, J. H. Cho, and D. C. Wang, "Performance Characteristics of Region-Based Group Key Management in Mobile Ad Hoc Networks," Proc. IEEE International Conference on Sensor Networks, Vol. 1, pp.411-419, June 2006.

[14] J. Pieprzyk and C. H. Li, "Multiparty Key Agreement Protocols," IEE Journal on Computers and Digital Techniques, pp.229-236, July 2000.

[15] X. Mingqiang, E. R. Inn-Inn and K. G. S. Winston, "Analysis of Clustering and Routing Overhead for Clustered Mobile Ad Hoc Networks", (ICDCS'06) 26th IEEE International Conference on Distributed Computing Systems, pp.46, 2006.

[16] M.Bechler, H. J. Hof, D. Kraft, F. Pahlke and L. Wolf, "A Cluster-Based Security Architecture for Ad Hoc Networks," Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, Hongkong, March 2004.

[17] H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang, "Self-securing ad hoc wireless networks," in Proc. 7th IEEE Symp. on Comp. and Communications (ISCC), Taormina, 2002.

[18] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "Providing robust and ubiquitous security support for mobile ad-hoc networks," in Proc. 9th International Conference on Network Protocols (ICNP). Riverside, California: IEEE, Nov. 2001, pp. 251–260.

[19] C. Maghmoumi, T. A. Andriatrimoson, J. Gaber, and P. Lorenz, "A Service Based Clustering Approach for Pervasive Computing in Ad Hoc Networks," in Proc. IEEE GLOBECOM 2008, December 2008.

[20] G. Venkataraman, S. Emmanuel and T. Srikanthan, "Size Restricted Cluster formation and Cluster Maintenance Technique for Mobile Ad-hoc Networks", International Journal of Network Management, Wiley InterScience, 2007, Vol.17, pp. 171-194.

[21] N. S. Yadav, B.P. Deosarkar and R.P.Yadav, "A Low Control Overhead Cluster Maintenance Scheme for Mobile Ad hoc NETworks (MANETs)," ACEEE International Journal on Network Security, Volume 1. Number 1. May 2009.

[22] C. Maghmoumi, H. Abouaissa, J. Gaber, and P. Lorenz, "A Clustering-Based Scalable Key Management Protocol for Ad Hoc Networks," in Proc. Second International Conference on Communication Theory, Reliability, and Quality of Service, France, July 2009.