An Efficient Rekeying Function Protocol with Multicast Key Distribution for Group Key Management in MANETs

N. Vimala* B. Jayaram[$] Dr. R. Balasubramanian[#]

*Asst.Professor , Department of Computer Science, CMS College of Science and Commerce, Coimbatore, India.

[$]Asst. Professor, Department of Computer Science & Engineering, PA College of Engineering and Technology, Pollachi, Coimbatore, India

[#]Dean Academic Affairs, PPG Institute of Technology, Coimbatore, India

ABSTRACT
Key management in the ad hoc network is a challenging issue concerning the security of the group communication. Group key management protocols can be approximately categorized into three: centralized, decentralized, and distributed. The much apt solution to provide the services like authentication, data integrity and data confidentiality is the establishment of a key management protocol. This paper proposes an approach for the design and analysis of region-based key management protocols with a new multicast key distribution scheme for scalable and reconfigurable group key management in Mobile Ad Hoc Networks (MANETs). The region-based group key management protocol partitions a group into region-based subgroups based on decentralized key management principles by using the Novel Re-Keying Function Protocol (NRFP). An issue is aroused by most of the centralized key management protocols on data security on group communication. The proposed system significantly reduces the computation complexity. Instead using conventional encryption algorithms, the proposed scheme employs an MDS code which is a class of error control codes, to distribute multicast key dynamically. This scheme considerably reduces the computation load of each group member compared to existing schemes employing traditional encryption algorithms. Such a scheme is advantageous for many wireless applications where portable devices or sensors need to reduce their computation as much as possible because of battery power limitations. Simply combined with any key-tree-based schemes, this scheme affords much lower computation complexity while maintaining low and balanced communication complexity and storage complexity for secure dynamic multicast key distribution.

Keywords-- Cluster Head, Group Key, Key Management Protocol, Mobile Ad Hoc Networks (MANETs), Region-based and Rekeying, multicast, MD5 codes, erasure decoding

1. INTRODUCTION
An ad hoc network is an assortment of independent nodes that communicate with each other, most regularly using a multi-hop wireless network. Nodes do not inevitably know each other and come together to form an ad hoc group for some particular reason. Key distribution systems typically involve a trusted third party (TTP) that acts as an intermediary between nodes of the network. A node in an ad hoc network has straight connection with a set of nodes, called neighboring nodes, which are in its communication range. The number of nodes in the network is not essentially preset. New nodes may join the network while existing ones may be compromised or become un-functional [1]. Key management in the ad hoc network is a challenging issue concerning the security of the group communication. Group key management protocols can be approximately classified into three categories; centralized, decentralized, and distributed [2].

MANET is one where there is no predetermined infrastructure such as base stations or mobile switching centers. Mobile nodes that are within each other's radio range communicate directly by means of a wireless network, whereas those far apart rely on other nodes to act as routers to relay its messages [3]. The most suitable solution to provide the services among which authentication, data integrity and data confidentiality is the establishment of a key management protocol. This protocol is liable for the generation and the distribution of the traffic encryption key (TEK) to all the members of a group. This key is used by the source to encrypt multicast data and by the receivers to decrypt it. Therefore only legitimate members are able to receive the multicast flow sent by the group source [4]. The elemental security services provided by every key management system are key synchronism, secrecy, freshness, independence, authentication, confirmation and forward and backward secrecy [7].

Clustering is the concept of dividing the multicast group into a number of sub-groups. Each sub-group is managed by a local controller (LC), accountable for local key management within its cluster. Furthermore, not many solutions for multicast group clustering did think about the energy problem to realize an efficient key distribution process, whereas energy constitutes a foremost concern in ad hoc environments [5] [6]. The group key is generated by the cluster head and communicated to other members through a secure channel that uses public key cryptography [14]. Clusters may be used for achieving different targets [8]. Some of them are clustering for transmission management, clustering for backbone formation, and clustering for routing efficiency. Group key management must be opposing to an extensive range of attacks by both outsiders and rouge members. In addition, group key management must be scalable, i.e., their protocols should be efficient in resource usage and able to decrease the effects of a membership change.

Previously a novel method for the design and analysis of region-based key management protocols for scalable and reconfigurable group key management in MANETs is given. It describes about the novel re-keying function protocol (NRFP) network security. A re-keying process management system for Mobile Ad-Hoc networks is designed to support in-network processing. The design of the protocol is motivated by decentralization key management for MANET covering key deployment, key refreshment, and key establishment. NRFP supports the establishment of novel administrative functions for

nodes that derive/re-derive a session key for each communication session. The protocol proposes direct connection, in-direct connection and hybrid connection. NRFP also includes an efficient protocol for local broadcast authentication based on the use of one-way key chains. A salient feature of the authentication protocol is that it supports source authentication without precluding in-network processing. Security and performance analysis shows that it is very efficient in computation, communication and storage and, that NRFP is also effective in defending against many sophisticated attacks. This region-based group key management protocols deal with outsider attacks in MANETs to preserve the security properties. A performance model to evaluate the network traffic cost generated for group key management in the proposed region-based protocol for MANETs is developed.

To enhance this scheme a new technique a new dynamic group key distribution scheme is proposed that drastically reduces computation complexity and yet maintains at least the same security degree of using symmetric encryption algorithms without increasing communication or storage complexity. In this scheme, information associated to session keys is encoded using error control codes rather than encryptions. In general, encoding and decoding of an error control code have much (at least one order, although this is hard to strictly quantify analytically) lower computation complexity than existing encryption and decryption algorithms, which has been verified by experiment. Thus, the computation complexity of key distribution can be considerably reduced. To achieve privacy the similar idea of using error control codes was implemented in [18], [19], and [20]. The major difference between these previous schemes and the proposed approach is that this scheme allows dynamic group membership changes with very low storage complexity, whereas the other schemes only work for a predefined static group.

The security strength of this scheme will be experimented and evaluated, as well as its communication, storage, and computation complexity. This scheme has low storage complexity, aside from its low computation complexity, i.e., O(1) for an individual group member and O(n) for the GC, where n is the number of group members. Concrete design parameters are used to apply this scheme to key trees, Based on the basic scheme using error control codes. Experiments are conducted to show great reduction of the proposed scheme in computation complexity than using other commonly used traditional encryption algorithms on 3-ary balanced key trees.

The remainder of this paper is structured as follows. Section 2 of this paper discusses some of the earlier proposed cluster based group key management techniques. Section 3 describes our proposed method of new region based group key management system with a new multicast key distribution scheme protocol for MANETs. Section 4 explains the performance evaluation of the proposed approach and section 5 concludes the paper with fewer discussions.

2. RELATED WORK

Key management is an indispensable part of any secure communication. Most cryptosystems rely on some underlying secure, robust, and efficient key management system. This section of the paper discusses some of the earlier proposed key management schemes for secure group communication in wireless ad hoc networks.

Maghmoumi et al. in [9] proposed a cluster based scalable key management protocol for Ad hoc networks. Their proposed protocol is based on a new clustering technique. The network is partitioned into communities or clusters based on affinity relationships between nodes. In order to ensure trusted communications between nodes they proposed two types of keys generated by each cluster head. The protocol is adaptive according to the limitation of the mobile nodes battery power and to the dynamic network topology changes. Their proposed approach of clustering based scalable key management protocol provided secured communications between the nodes of the Ad hoc networks.

A key management scheme for secure group communication in MANETs was described by Wang et al. in [10]. They described a hierarchical key management scheme (HKMS) for secure group communications in MANETs. For the sake of security, they encrypted a packet twice. They also discussed group maintenance in their paper in order to deal with changes in the topology of a MANET. Finally, they carried out a performance analysis to compare their proposed scheme with other conventional methods that are used for key management in MANETs. The results showed that their proposed method performed well in providing secure group communication in MANETs.

George et al. in [11] projected a framework for key management that provides redundancy and robustness for Security Association (SA) establishment between pairs of nodes in MANETs. They used a modified hierarchical trust Public Key Infrastructure (PKI) model in which nodes can dynamically assume management roles. Furthermore they employed non-repudiation through a series of transactions checks to securely communicate new nodes information among Certificate Authorities (CAs). They assumed that nodes could leave and join the network at any time. Nodes could generate their own cryptographic keys and were capable of securing communication with other nodes. In order to balance the flexibility and increased availability of the Key Management Scheme (KMS), security was provided by introducing two concepts in addition to revocation and security alerts: non-repudiation and behavior grading. The KMS, by combining node authentication with an additional element, node behavior, it maintained sufficient levels of security. A behavior grading scheme required each node to grade the behavior of other nodes.

A new group key management protocol for wireless ad hoc networks was put forth by Rony et al. in [12]. They put forth an efficient group key distribution (most commonly known as group key agreement) protocol which is based on multi-party Diffie-Hellman group key exchange and which is also password-authenticated. The fundamental idea of the protocol is to securely construct and distribute a secret session key, 'K,' among a group of nodes/users who want to communicate among themselves in a secure manner. The proposed protocol starts by constructing a spanning tree on-the-fly involving all the valid nodes in the scenario. It is understood, like all other protocols that each node is distinctively addressed and knows all its neighbors. The password 'P' is also shared among each valid member present in the scenario. This 'P' helps in the authentication process and puts off man-in-the-middle attack. Unlike many other protocols, the proposed approach does not need broadcast/multicast capability.

Bechler et al. in [13] described cluster-based security architecture for Ad hoc networks. They proposed and estimated a security concept based on a distributed certification facility. A network is separated into clusters with one special head node for each cluster. These cluster head nodes carry out administrative functions and shares a network key among other members of the cluster. Moreover the same key is used for certification. In each cluster, exactly one distinguished node–the cluster head (CH)–is responsible for establishing and organizing the cluster. Clustering is also used in some routing protocols for ad hoc networks.

Decentralization is obtained using threshold cryptography and a network secret that is distributed over a number of nodes. The architecture addresses problems of authorization and access control, and a multi-level security model helps to adjust the complexity to the capabilities of mobile end systems. Based upon their authentication infrastructure, they provided a multi level security model ensuring authentication, integrity, and confidentiality.

A scalable key management and clustering scheme was proposed by Jason et al. in [15]. They projected a scalable key management and clustering scheme for secure group communications in ad hoc networks. The scalability problem is solved by partitioning the communicating devices into subgroups, with a leader in each subgroup, and further organizing the subgroups into hierarchies. Each level of the hierarchy is called a tier or layer. Key generation, distribution, and actual data transmissions follow the hierarchy. The Distributed Efficient Clustering Approach (DECA) provides robust clustering to form subgroups, and analytical and simulation results demonstrate that DECA is energy-efficient and resilient against node mobility. Comparing with most other schemes, their approach is extremely scalable and efficient, provides more security guarantees, and is selective, adaptive and robust.

Apart from the above mentioned numerous researches have been conducted in the field of cluster-based group key management for mobile ad hoc networks (MANETs).

3. A NEW REGION BASED GROUP KEY MANAGEMENT FOR MANETS

The proposed region-based group key management protocol divides a group into region-based subgroups based on decentralized key management principles. This partitioning of region into subgroups improves scalability and efficiency of the key management scheme in providing a secure group communication. Figure 1 shows the partitioning of region into subgroups on the basis of decentralized key management principles [16]. It is assumed that each member of the group is equipped with Global Positioning System (GPS) and therefore each one knows its location as it moves across the regions. For secure group communications, all members of a group share a secret group key, KG. In addition to ensure security in communication between the members of each subgroup all the members of the subgroups in the region 'i' hold a secret key KRi. This shared secret key is generated and managed by a distributed group key management protocol that enhances robustness. This region-based group key management protocol will function at the optimal regional size recognized to reduce the cost of key management in terms of network traffic.

The average number of nodes in the system is N=λpA, where λp denotes the node density of the randomly distributed nodes and A indicates the operational area with radius 'r'. The random distribution of nodes is according to a homogeneous spatial Poisson process. The nodes can join or leave a group at any point of time.

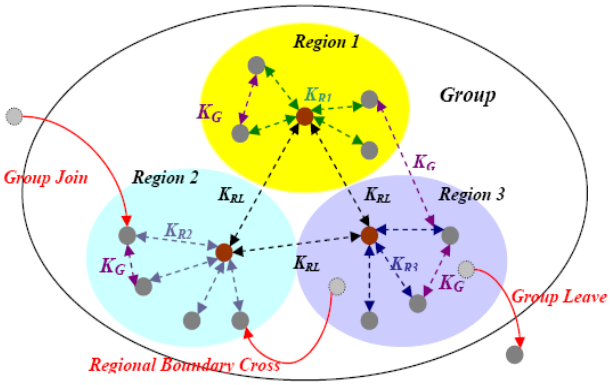


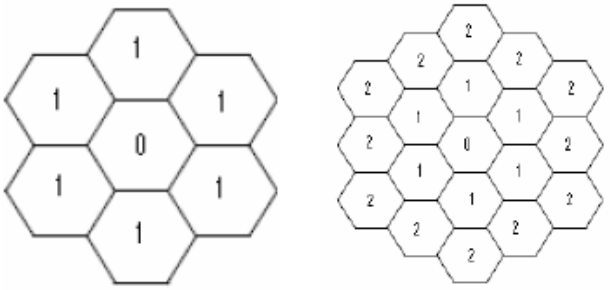
Figure 1 Region-based Group Key Management

A node may leave a group at any time with rate μ and may rejoin any group with rate λ. Therefore, the probability that a node is in any group is λ/(λ+μ) and the probability that it is not in any group is μ / (λ+μ). Let A_J and A_L be the aggregate join and leave rates of all nodes, respectively. Then, A_J and A_L, can be calculated as follows,

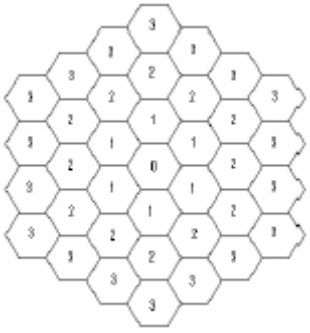
$$A_J = \lambda \times N \times \frac{\lambda}{\lambda + \mu}$$

$$A_L = \mu \times N \times \frac{\lambda}{\lambda + \mu}$$

Nodes in a group must satisfy the forward/backward secrecy, confidentiality, integrity and authentication requirements for secure group communications in the presence of malicious outside attackers.



n=1,Number of Regions=7 **n=2, Number of Regions=19**



n=3, Number of Regions=37

Figure 2. Representation of Regions in operational area

The important requirement for secure group communication is reliable transmission. This can be obtained by using acknowledgement (ACK) packets and packet retransmission upon timeout. Hexagon is used to model a region [17]. Let R(n) denotes the number of regions (i.e. $3n^2 + 3n + 1$) in the operational area. For n=3, the number of regions in the

operational area is 37, for n=2 and n=1, the number of regions in the operational area are 19 and 7 respectively. Figure 2 shows the representation of the regions in the operational area for n=1, 2, and 3.

3.1 Protocol Description

This describes the working of our proposed region-based group key management for MANETs.

1. Bootstrapping

In this initial bootstrapping process, a node within a region can take the responsibility of a regional "leader" to carry out Group Diffie Hellman (GDH). If there are multiple initiators, then the node with the smallest id will prevail as the leader and will implement GDH to completion to generate a regional key. Once a leader is generated in each region, all leaders in the group will execute GDH to agree on a secret leader key, K_{RL}, for secure communications among leaders. The group key K_G can be generated using the following, K_G = MAC (K_{RL}, c), where MAC is a cryptographically secure hash function, K_{RL} is the leader key used as the secret key to MAC, and c is a fresh counter which will be incremented whenever a group membership event occurs. The generated group key K_G is then disseminated among the group members by the group leader. This group key provides secure group communication across regions.

2. Key Management

The next important task is managing the generated key. These shared secret keys at the subgroup (regional), leader, group levels may be rekeyed to preserve secrecy in response to events that occur in the system. Therefore, whenever there occur a change in the leader of the group, the leader key, K_{RL} is rekeyed. The regional key (K_R) is rekeyed whenever there is a regional membership change, including a local member group join/leave, a node failure, a local regional boundary crossing, and a group merge or partition event.

3. View Management

In addition to maintaining secrecy, the proposed region-based key management protocol also allows membership consistency to be maintained through membership views. Three membership views can be maintained by various parties: (a) Regional View (RV) contains regional membership information including regional (or subgroup) members' ids and their location information, (b) Leader View (LV) contains leaders' ids and their location information, and (c) Group View (GV) contains group membership information that includes members' ids and their location information.

3.2 Maximum Distance Separable Codes

Maximum Distance Separable (MDS) codes are a class of error control codes that meet the Singleton bound [19, chapter 11]. Letting GF(q) be a finite field with q elements [19, chapter 4], an (n,k) (block) error control code is then a mapping from $GF(q)^k$ to $GF(q)^n$: $E(m)=c_r$, where $m= m_1,m_2,\ldots, m_k$ is the original message block, $c=c_1,c_2,\ldots, c_n$ is its code word block, and E(.) is an encoding function, with $k \leq n$. If a decoding function D(.) exists such that $D(c_{i1}, c_{i2}, c_{ik}, i_{1,} i_{2,\ldots,} i_k)= m$ for $1 \leq ij \leq n$ and $1 \leq j \leq k$, then this code is called an (n,k) MDS code. For an (n,k) MDS code, the k original message symbols can be recovered from any k symbols of its code word block. The process of recovering the k message symbols is called erasure decoding. All the symbols are defined over GF(q), and usually, $q=2^m$. The well-known Reed-Solomon (RS) codes [22] are a class of widely used MD Scodes. Notably, the RS codes and other MDS codes can be used to construct secret-sharing and threshold schemes [18], [19].

3.3 Novel Re-keying Function Protocol (NRFP)

The nodes should have the following keys: MK, which is shared by all the nodes in the network; LK, which is shared with the BS; and SK, which is shared with another node. Each of these keys is considered in turn with the reasons for including it in the prototype.

Master key (MK): This is a globally shared key that is used by the base station for encrypting messages that are broadcast to the whole group. Each node is imprinted with master key and LAFs when it is manufactured.

Local key (LK): Every node has a unique key that is injected with initial local key (LK), is shared with the base station. This key is the basic parameter for the re-keying function of the proposal and is used for secure communication between the node and the base station.

Session key (SK): Every node shares an SK with each of its immediate neighbours. In NRFP, SKs are used for securing communications that require privacy or source authentication.

LAFs: The local administrative functions include 'master function', 're-keying function', and 'derivation function' and can be imprinted with node to achieve a high-level security of node - to- node communication. The LAFs are responsible for key generation of the cluster session keys depending on which initial master key and local control key were imprinted at the time of manufacturing, whereas the HMAC is adopt of LAFs work. Master function, the derivation function is used to generate new key values based on requesting message coming from BS or CH. The re-keying process is necessary for two reasons:

a)It is simple for k to compute f(k), but computationally infeasible for f(k) to compute k.

b) $k_0, k_1, k_2,\ldots k_n$, is computationally infeasible to compute f(k), as long as it is computationally infeasible to compute k.

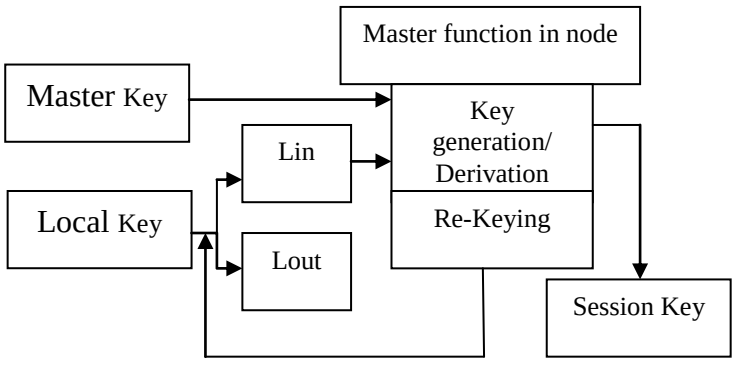


Figure 3: novel Re-keying function protocol

Prior to node deployments each node is injected with initial LK, which is the basic parameter for the re-keying function of our proposal. The re-keying function is responsible for assigning a new value to LK. The cluster head periodically refreshes to respond to the changes of the LK key, which notifies all of its members in a secret way of the new change. Functions and keys implement through the fundamental principles of the key management as following:

1) Key deployment: every node is imprinted with unique ID, MK, LK and master functions that can generate and regenerate the unique sharing key with other nodes driving from MK and LK. Those keys that were imprinted never exchange during a communication session between nodes; the only key exchanged to establish communication between two nodes is the SK.

2) Key Establishment: all nodes on the network use the same mechanism to communicate securely with each other. After deployment and the completion of the cluster head performance, the cluster head generates the SK and sends the control message to its members to encourage them to generate the SK. Intra-cluster node-to-node communications are supported for this round: when two nodes want to communicate with each other they use the same SK to establish secure communication and initiate the exchange of data. SK should be the same because all nodes are using the same derivation function.

3) Node addition: when a new node joins the network, it first must join to any cluster on the network. If it receives a cluster head beacon, the key refreshment runs inter-cluster and generates its own SK. If a node does not receive any CH beacons, it becomes its own cluster and acts as a CH of this cluster, then runs the LAFs to generate its own keys.

4) Node eviction: node eviction means that any node in the cluster leaves its region for any reason (Power consumption, node emigration, node capture, etc.). In this case, we propose two cases of node eviction:

Case 1: Member node eviction occurs when the cluster head does not receive the hello message from a certain node, CH sends a hello message to that node and waits for a reply. If it does not receive a reply within a certain time, the cluster head sends a message to all of its members to inform them to delete the node with a certain ID from the list of neighbors.

Case 2: In CH eviction when a cluster head leaves the cluster, two processes must be completed.

First, the cluster head sends messages to all of its members to inform them that it is going to leave. From each cluster member, the node members then elect the cluster head which has a highest number of a list of neighbors or the node that has the highest power. Second, if the cluster head left surreptitiously, the entire cluster member will not receive the CH beacon for a period, and then the cluster members rebuild the cluster according to cluster base process and elect a new cluster head.

3.4 Authentication

For a message authentication code (MAC) function a MAC algorithm can be generated using multiple different techniques, as long as the sender and receiver have shared secret keys. A MAC algorithm can create out of a common symmetric cipher such as DES2 or AES3. A sender wanting to send a secure message can send M encrypted, e(M), with a symmetric cipher and then resend M||K (M concatenated with K) encrypted, e(M||K). The receiver first decrypts M, d(e(M)), to generate M'. M'||K, e(M'||K) are then encrypted and compared with the e(M||K) originally sent. If the two match, then this confirms that the data was not corrupted. HMAC [13] is merely a specific type of MAC function. It works by using an underlying hash function over a message and a key. Any hashing function could be used with HMAC, although more secure hashing functions are preferable. Moreover, HMAC is computationally very fast and compact. HMAC accomplishes both of these properties because of its reliance on a given hash function which is fast and returns compact outputs.

Security features:

NRFP presents a new methodology in keying information for MANET to insure the secure communication between nodes on the networks topology. NRFP is designed to improve the cluster formation security of key management and the proposal is designed to support secure communications in networks; therefore, it provides the basic security services such as confidentiality and authentication. In addition, NRFP is to meet several security and performance requirements that are considerably more challenging to Mobile Ad-Hoc networks.

Key management protocol NRFP satisfies the following properties:

Property 1: Only the authorized user can communicate in the network. Unauthorized (outside attackers) cannot participate in the communication without proper assigned key materials.

Property 2: The session key distribution process is secure. The distribution of session key is based on the personal key share distribution scheme. A revoked network cannot recover the session key because of the key to self-generation and thus does not need to deploy Log. Because of the broadcast, an outside attacker cannot masquerade as a base station disseminating a session key and start a revocation attack either.

3.5 Group Communication Protocol

For typical group communication, we accept to use the publish/subscribe service. It is assumed that all members are interested in all published data by all members. Thus, all published data in each member are disseminated to all members whenever each node publishes its data. By taking two-level hierarchical key management structure, the published data in each node is broadcast to its members in the region, and then the leader receiving the published data distributes it to other leaders. After then, each leader broadcasts the published data to its members respectively. When all published data are disseminated to all members in this way, a group key is used to encrypt/decrypt the published data.

3.6 MDS Code-Based Rekeying on a Key Tree

The GC initialization and each member's initial join can be carried out exactly the same on a key tree as in the basic scheme for rekeying (NRFP). Thus, it is focused on the adaption of the basic scheme for rekeying on a key tree. As in other similar key-tree-based rekeying schemes, MDS codes are used to rekey from bottom (leaves) up.

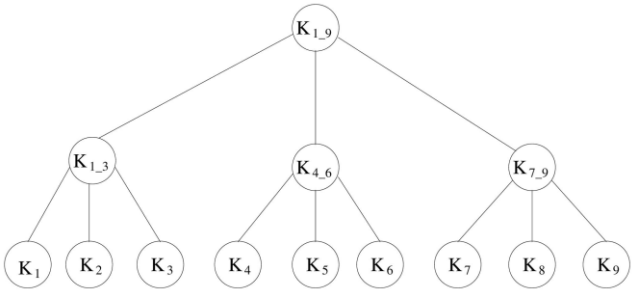


Figure 4: A key tree for a nine-member group

In Fig. 4, when the member 9 leaves, new subgroup key $K_{7\text{-}8}$ is rekeyed before the server changes the new group session key to $K_{1\text{-}8}$. When MDS codes are worn for the rekeying process, each node (leaf or intermediate) key becomes a pair of (j_i,s_i), as discussed in the previous section. The GC stores all the key pairs

on the key tree. Whenever encryptions are required for rekeying a subgroup key, a new MDS code word is constructed from all the key pairs (j_i,s_i)'S) of the corresponding immediate child nodes and then multicast by the GC. Note that in the rekeying process, each level of the key tree may use the same or different MDS codes. However, for the ease of implementation, the same MDS code can be used for all levels, since the security of the basic scheme does not depend on the MDS code. In Fig. 4, when member 9 leaves the group, the GC first uses the key pairs K_7=(j_7,s_7) and K_8=(j_8,s_8), together with a fresh random r, to construct a code word of an (L,2) MDS code and then follows the rekeying procedure of the basic scheme, as described in the previous section. After proper decoding, members 7 and 8 share a new subgroup key K_{7-8} or, more accurately, its component s_{7-8}, which is only known to them, aside from the GC. Note here that k_{7-8}=(j_{7-8},s_{7-8}) but j_{7-8} can be predetermined publicly once the key tree structure is decided. This finishes the rekeying of K_{7-8}. Next, the GC constructs another code word of an (L,3) MDS code from the subgroup keys K_{1-3}, K_{4-6}, and K_{7-8}, and the decoding output from this code word produces a new group session key K_{1-8}, which is shared by all the remaining group members. Note that when the key tree is a d-ary balanced tree, only an (L,d-1) MDS code is needed to rekey the immediate subgroup key shared by the leaf node corresponding to the just-left old member. Then, another (L,d) MDS code is needed for rekeying all the other subgroup keys and the new group session key. Since the rekeying scheme based on MDS codes doesn't change the communication and storage complexity of the underlining key-tree-based rekeying scheme, the communication complexity still remains to be O(dlogdn)

3.7 MDS Code Implementation for the Rekeying Scheme (RS)

As explained previously, d needs to be 3 to minimize the communication complexity during rekeying. Consequently, only two types of MDS codes are needed, which are (L,2) and (L,3) codes. In fact, the rekeying scheme needs only two specific MDS codes, i.e., an (L,2) code and an (L,3) code. Although any general (n,k) MDS code can be used for the rekeying purpose by setting k=2 or k =3, there are a number of optimization techniques that can be applied for special implementations of the (L,2)and (L,3) codes. As it can be seen, these techniques turn out to make the codes used for rekeying extremely fast, even though they do not readily extend to the implementations of general MDS codes.

4. EXPERIMENTAL RESULTS

The performance analysis helps identify the optimal regional size that will minimize the network traffic generated while satisfying security properties in terms of secrecy, availability and survivability. The cost metric used for measuring the proposed group key management protocol is the total network traffic per time unit incurred in response to group key management events including regional mobility induced, group join/leave, periodic beaconing, and group merge/partition events. To evaluate the performance of this proposed approach it is discussed on group join/leave cost, group communication cost and a multicast key distribution scheme.

4.1 Group Join/Leave Cost

This is the cost per time unit for handling group join or leave events. This cost also includes the cost caused by connection/disconnection events by group members.

$$C_{Join/Leave,i} = \lfloor A_J \times C_{Join,i} \rfloor + \lfloor A_L \times C_{Leave,i} \rfloor$$

Here A_J and A_L are the aggregate group join and leave rates of all members, respectively. A group join event requires the update of the regional view and the rekeying of the regional key in the region from which the join event is originated, the cost of which is C_{intra}, as well as the update of the group view and the rekeying of a group key, the cost of which is C_{group}, i.

$$C_{Join,i} = \lfloor C_{intra} \rfloor + \lfloor C_{group,i} \rfloor$$

The cost for group leave event includes two cases, namely, when a non-leader member leaves and when a leader leaves the group. Thus, the cost for a group leave event is given as follows,

$$C_{Leave,i} = C_{Leave,i}^{non-leader} + C_{Leave,i}^{leader}$$

4.2 Cost for Group Communication

It includes the cost of group communications between members. It is assumed that the publish/subscribe service is used to realize efficient group communications. For simplicity, it is assumed that all members are interested in all published data by all members, and the data are published in each node with the rate of λ_{pub}. Thus, the aggregate rate that data are published in each node is obtained as:

$$A_{pub} = N \times \left[\frac{\lambda}{\lambda + \mu}\right] \times \lambda_{pub}$$

Whenever each node publishes its data, the published data should be disseminated to all members. Taking advantage of our hierarchical key management structure, the published data can be distributed to all leaders first, and then each leader can broadcast them to its members in the region.

$$C_{GC,i} = A_{pub} \times \left(\left(N_{region,i} \times M_{pub} \times H_{region} \right) + \left(M_{pub} \times H_{region,i} \right) \right)$$

4.3 Evaluation of proposed Cryptographic Schemes

To experiment the proposed scheme, a multicast key distribution scheme is implemented to disseminate 128-bit session keys among a 3-ary balanced key tree. The proposed scheme is compared with previous cryptographic schemes. As the communication and storage complexity are the similar among all the schemes, it suffices to simply compare the computation complexity.

The comparison is performed considering the following scenario, where each three-member group has one member that departs. These departures are not controlled to happen at the same time, but in practice, they might tend to be close, for example, at the end of one movie broadcast, etc. This makes a batch process possible, which signifies that all remaining members could be rekeyed at once. Before giving out the experimental results, it is worth pointing out that any one-way hash function used in the proposed scheme can be simplified from general-sense hash function implementations. For instance, the MD5 algorithm [21, chapter18] is used as an exemplary hash function in this evaluation, which produces a 128-bit hash output from any arbitrary length input. A general MD5 input has three components: 1) input data, 2) padding bits, and 3) a final 64-bit field for length. In this case, as the input is always 128 bits, the final length field can be presented to represent 128. Moreover, all the rest bits can be set to 0 and removed from the MD5 logic. This tends to make the MD5 algorithm more efficient.

Obviously, the same method can be directly applied to other hash algorithms, for example, SHA-1 and SHA-256 [21, chapter 18], should the MD5 algorithm be considered insufficient or using longer session keys becomes necessary.

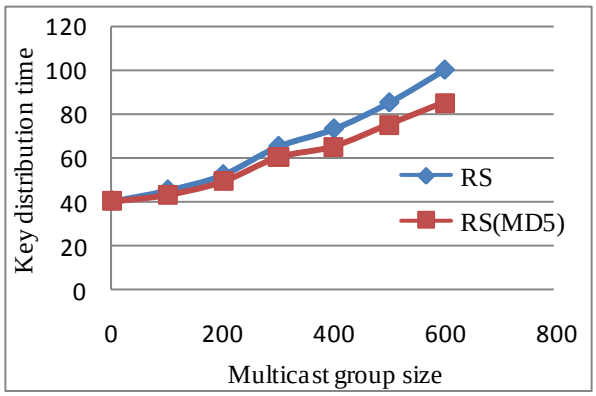


Figure 5: Computation time for key distribution (the RS (MD5) shows the proposed scheme, whereas the RS curve excludes the hash function) - GC's computation time for key dissemination.

Figure 5 displays the computation time of the key dissemination and recovery using different schemes under various multicast group sizes. The experiments are carried out on a Pentium 4 2.53-GHz machine with a 512-Mbyte memory. It can be clearly noticed that using one-way hash functions adds none-trivial computation complexity. Nevertheless, the proposed scheme still performs better than the conventional cryptographic schemes by a significant margin. The computation time of the key distribution is also compared with the conventional stream ciphers, as shown in Table 1, for a selected multicast group size. It is noticed that the computation times of both the GC and the member using the RC4 cipher are significantly larger than using other schemes. Even though RC4 is a fast stream cipher, its key scheduling process has dominant effect in this particular scenario, where only 128-bit data is encrypted/ decrypted using any given key. Results under other multicast group sizes are more similar, which are thus not duplicated here.

TABLE 1COMPUTATION TIME COMPARING TO THE RC4 APPROACH

Time (us)	RS	RS(MD5)	RC4
GC	19	28	227
Member	2	5	61

Finally, it is worth noting that this basic scheme simply reduces computation complexity by replacing cryptographic encryption and decryption operations with more efficient encoding and decoding operations. It is orthogonal to all other schemes that use different rekeying protocols and procedures. This basic scheme can be combined always with any rekeying schemes that use cryptographic encryption and decryption operations. For example, this basic scheme can be directly adapted to incorporate the so-called one-way function tree scheme, where a different rekeying protocol on a key tree is used other than the traditional scheme, to further reduce the computation complexity.

5. CONCLUSION

Key management in the ad hoc network is a challenging problem concerning the security of the group communication. This paper proposed an approach for the design and analysis of region-based key management protocols with a new multicast key distribution scheme for scalable and reconfigurable group key management in MANETs. The region-based group key management protocol classifies a group into region-based subgroups based on decentralized key management principles by using the Novel Re-Keying Function Protocol (NRFP). NRFP supports the establishment of novel administrative functions for nodes that derive/re-derive a session key for each communication session. A most important feature of the authentication protocol is that it supports source authentication without precluding in-network processing. NRFP is also efficient in defending against many sophisticated attacks. The computation complexity of key distribution is highly reduced by employing only erasure decoding of MDS codes instead of more expensive encryption and decryption computations. By combining with key trees or other rekeying protocols that need encryption and decryption operations, this scheme provides much lower computation complexity while maintaining low and balanced communication complexity and storage complexity for dynamic group key distribution. This scheme is thus convenient for many applications in various broadcast capable networks such as Internet and wireless networks.

6. REFERENCES

[1] A. Renuka, and K. C. Shet, "Cluster Based Group Key Management in Mobile Ad hoc Networks," IJCSNS International Journal of Computer Science and Network Security, vol. 9, no. 4, pp. 42-49, 2009.

[2] S. Rafaeli, and D. Hutchison, "A survey of key management for secure group communication," ACM Computing Surveys, vol. 35, no. 3, pp. 309–329, 2003.

[3] Hao Yang, Haiyun Luo, Fan Ye, Songwu Lu, and Lixia Zhang, "Security in mobile Ad-Hoc networks-Challenges and Solutions," IEEE Transactions on Wireless Communications, vol. 11, no. 1, pp. 38-47, 2004.

[4] Mohamed-Salah Bouassida, Isabelle Chrisment, and Olivier Festor, "Group Key Management in MANETs," International Journal of Network Security, vol. 6, no. 1, pp. 67-79, 2008.

[5] 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.

[6] 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.

[7] Menezes, P. V. Oorschot, and S. A. Vanstone, "handbook of Applied Cryptography", CRC Press, New York, 1997.

[8] C. E. Perkins, "Ad hoc networking", Addison-Wesley Pub Co, 1st edition December 29, 2000.

[9] Chadi Maghmoumi, Hafid Abouaissa, Jaafar Gaber, and Pascal Lorenz, "A Clustering-Based Scalable Key Management Protocol for Ad Hoc Networks," Second

International Conference on Communication Theory, Reliability, and Quality of Service, pp.42-45, 2009.

[10] Nen-Chung Wang, and Shian-Zhang Fang, "A hierarchical key management scheme for secure group communications in mobile ad hoc networks," Journal of Systems and Software, vol. 80, no. 10, pp. 1667-1677, 2007.

[11] George C. Hadjichristofi, William J. Adams, and Nathaniel J. Davis, "A Framework for Key Management in Mobile Ad Hoc Networks," International Journal of Information Technology, vol. 11, no. 2, pp. 31-61, 2006.

[12] Rony H. Rahman, and Lutfar Rahman, "A New Group Key Management Protocol for Wireless Ad-Hoc Networks," International Journal of Computer and Information Science and Engineering, vol. 2, no. 2, pp. 74-79, 2008.

[13] M. Bechler, H. -J. Hof, D. Kraft, F. Pählke, and L. Wolf, "A Cluster-Based Security Architecture for Ad Hoc Networks," Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM, vol. 4, pp. 2393-2403, 2004.

[14] Yi Jim Chen, Yi Ling Wang, Xian Ping Wu, and Phu Dung Le, "The Design of Cluster-based Group Key Management System in Wireless Networks," pp. 1-4, 2006.

[15] Jason H. Li, Renato Levy, Miao Yu, and Bobby Bhattacharjee, "A scalable key management and clustering scheme for ad hoc networks," Proceedings of the 1st international conference on Scalable information systems, 2006.

[16] Jin-Hee Cho, "Design and Analysis of QoS-Aware Key Management and Intrusion Detection Protocols for Secure Mobile Group Communications in Wireless Networks," Thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University.

[17] J. W. Wilson, and I. R. Chen, "Performance Characteristics of Location-based Group Membership and Data Consistency Algorithms in Mobile Ad hoc Networks," International Journal of Wireless and Mobile Computing, vol. 1, no. 8, 2005.

[18] A. Shamir, "How to Share a Secret," Comm. ACM, vol. 24, no. 11, pp. 612-613, Nov. 1979.

[19] R.J. McEliece and D.V. Sarwate, "On Sharing Secrets and Reed-Solomon Codes," Comm. ACM, vol. 26, no. 9, pp. 583-584, Sept.1981.

[20] R. Blom, "An Optimal Class of Symmetric Key Generation Systems," Advances in Cryptology—Proc. Workshop Theory and Application of Cryptographic Techniques (EUROCRYPT '84), pp. 335-338, 1984.

[21] B. Schneier, Applied Cryptography, second ed. John Wiley & Sons, 1996.

[22] I.S. Reed and G. Solomon, "Polynomial Codes over Certain Finite Fields," J. SIAM, vol. 8, no. 10, pp. 300-304, 1960.

AUTHOR BIOGRAPHY

N. Vimala received her B.Sc., (CS) from Avinashilingam Deemed University, Coimbatore, TamilNadu, in 1993. She obtained her M.Sc., (CS) and M.Phil degree from Bharathiar University, Coimbatore, TamilNadu, in the year 1995 and 2001 respectively. She is currently the Senior Lecturer, Department of Computer Science, CMS College of Science and Commerce, Coimbatore, TamilNadu. She has the long experience of teaching Post graduate and Graduate Students. She has produced 43 M.Phil Scholars in various universities. Her area of interest includes Network Security, Database Management Systems, Object Oriented Programming and Artificial Intelligence. She is currently pursuing her Research in the area of Network Security under Mother Teresa University, Kodaikanal, TamilNadu. She is a member various professional bodies.

B. Jayaram obtained his M.E in Computer Science and Engineering in the year 2006 from Anna University, Chennai. He is currently working as Assistant Professor, Department of Computer Science & Engineering, PA College of Engineering and Technology, Pollachi. He has previously served as lecturer prior to this he had served as an active member of the development team in ERP products at Ramco Systems, Chennai. His area of interest includes data structure, computer networks, data mining, and biometrics.

Dr. R. Balasubramanian was born in 1947 in India. He obtained his B.Sc., and M.Sc., degree in Mathematics from Government Arts College, Coimbatore, TamilNadu, in 1967 and PSG Arts College, Coimbatore, TamilNadu, in 1969 respectively. He received his Ph.D., from PSG College of Technology, Coimbatore, TamilNadu, in the year 1990. He has published more than 15 research papers in national and international journals. He has been serving engineering educational service for the past four decades. He was formerly in PSG College of Technology, Coimbatore as Assistant Professor in the Department of Mathematics and Computer Applications. He served as Associate Dean of the Department of Computer Applications of Sri Krishna College of Engineering and Technology, Coimbatore. Currently taken charge as Dean Academic Affairs at PPG Institute of Technology, Coimbatore, before which he was a Dean Basic Sciences at Velammal Engineering College, Chennai. He has supervised one PhD thesis in Mathematics and supervising four doctoral works in Computer Applications. His mission is to impart quality, concept oriented education and mould younger generation.

He is member of the board of studies of many autonomous institutions and universities. He was the principal investigator of UGC sponsored research project. He is a referee of an international journal on mathematical modeling. He has authored a series of books on Engineering Mathematics and Computer Science. He is a life member of many professional bodies like ISTE, ISTAM and CSI.