Online ISSN 1848-3380, Print ISSN 0005-1144
ATKAFF 57(4), 1020–1034(2016)

Mohammad M. Shurman, Mamoun F. Al-Mistarihi, Khalid A. Darabkh

Dynamic Distribution of Security Keys and IP Addresses Coalition Protocol for Mobile Ad Hoc Networks

DOI 10.7305/automatika.2017.12.1233
UDK 654.16-049.5:004.725.7.056.057.4*T-DAAP

Original scientific paper

In mobile adhoc networks (MANETs) a tree-based dynamic address auto-configuration protocol (T-DAAP) is one of the best protocols designed for address assignment as far as the network throughput and packet delays are concerned. Moreover, MANET security is an important factor for many applications given that any node can listen to the channel and overhear the packets being transmitted. In this paper, we merge the address assignment with the security key delivery into one protocol, such that a node in the MANET is configured with IP address and security key simultaneously. To the best of our knowledge, no single protocol provides concurrent assignment of IP addresses and security keys for MANET nodes. The proposed method, which is based on T-DAAP, shows significant enhancements in the required control packets needed for assigning network nodes IP addresses and security keys, MAC layer packets, total end-to-end delay, and channel throughput over those obtained when using separate protocols. Additionally, it provides not only efficient security keys to the nodes from the first moment they join the network, but also secure delivery of the address and security key to all participating nodes. It is noteworthy to mention that providing a complete security model for MANET to detect and countermeasure network security threats and attacks is beyond the scope of our proposed protocol.

Key words: Mobile Adhoc networks, IP address, security key, packet delay, channel throughput, control packets

Dinamička distribucija sigurnosnih ključeva i koalicijski protokol IP adresa za mobilne ad hoc mreže. Kod mobilnih ad hoc mreža (MANET) dinamički protokol za autokonfiguraciju adresa baziran na stablu (T-DAAP) je jedan od najboljih protokola dizajniranih za dodjelu adresa iz perspektive propusnosti mreže i i kašnjenja paketa. Štoviše, sigurnost MANET-a je važan faktor za mnoge aplikacije s obzirom da bilo koji čvor može osluškivati kanal i slučajno čuti pakete koji se šalju. U ovom radu, dodjela adresa i dostava sigurnosnih ključeva spojeni su u jedan protokol tako da je čvor u MANET-u konfiguriran simultano s IP adresom i sigurnosnim ključem. Prema saznanjima autora, niti jedan postojeći protokol ne pruža istovremeno dodjeljivanje IP adrese i sigurnosnog ključa za MANET čvorove. Predložena metoda, koja se bazira na T-DAAP-u, pokazuje značajna poboljšanja u odnosu na metode koje koriste odvojene porotokole, kod traženih kontrolnih paketa koji su potrebni za dodjeljivanje IP adresa i sigurnosnih ključeva čvorovima mreže, MAC paketa, ukupnog end-to-end kašnjenja i propusnosti kanala. Dodatno pruža ne samo efikasne sigurnosne ključeve čvorovima od trenutka kad se priključe mreži, nego i sigurno dostavljanje adrese i sigurnosnog ključa svim čvorovima koji sudjeluju u mreži. Važno je spomenuti da je pružanje cjelokupnog sigurnosnog modela za MANET koji detektira dodatno i protumjere prijetnjama i napadima na sigurnost mreže izvan dosega predloženog protokola.

Ključne riječi: mobilne ad hoc mreže, IP adresa, sigurnosni ključ, kašnjenje paketa, propusnost kanala, kontrolni paketi

1 INTRODUCTION

Opposed to the infrastructure wireless networks where each user communicates directly with an access point or base station [1-10], mobile ad hoc network, or MANETs are a kind of wireless ad hoc networks that become increasingly involved in many daily activities ranging from simple applications such as home networks to critical applications as in military networks [11-18]. In all adhoc applications, nodes at the lead of a network joining must be assigned an IP address or identification in order to communicate with other nodes in the network without confusion [19-25]. In addition to identification, a shared security key must be established before communicating with other nodes. It is widely reported that the symmetric cryptography schemes are convenient for ad hoc networks while the public key cryptography schemes are not feasible on many grounds [26-29]. Node address assignment is an active research topic since it can be performed using several techniques [30-34]. Additionally, the security key assignment in MANETs is also of interest, but unfortunately, there are no widely-accepted standards among researchers heretofore.

Node addresses in MANET can be assigned using centralized, decentralized, or neighbor based approaches. Centralized approaches use a central server for address assignment in a newly joining node. The disadvantage of these methods is the possibility of single point of failure in the central server which can disrupt the entire network [35, 36]. Decentralized approaches allow the node to pick an IP address and then check for uniqueness [37,38]. The main disadvantage of these methods is that the verification required for address uniqueness/conflicts with other nodes are accomplished by network-wide flooding. In neighbor-based approaches, the network does not suffer from wide ?ooding. In other words, achieving a unique address requires a flooding to neighbors only. These approaches have been found to be the best for MANET in terms of delay, flooding, and throughput [30, 35, 39]. Hence, the proposed protocol employs a neighbor based protocol for address assignment part.

When a MANET requires security, it will be required for each node to have a security key (as mentioned above, the shared key cryptography is more convenient for MANETs). Key assignment can be performed using either central or distributed approaches [40]. Nevertheless, due to the nature of MANET, central approaches suffer from many obstacles [31], out of which the saved keys might be revealed, the central server might be stolen, hacked, or intercepted which in turn lead to having a very serious security breach. Additionally, it may go offline for a number of reasons (i.e., power failure or DoS attack) and stops distributing the keys to the nodes. Another concern may be the range of accessibility because of network partitioning [26].

Most research in MANET security and address management assumes two protocols (i.e., one protocol for address assignment and another for security key assignment). Few papers have been published concerning the autoconfiguration and key distribution in adhoc networks [41, 42]. The idea of these approaches is to correlate IP address with a security key which incurs many problems include NATing of local address, fixed mapping of IP, and security key generation. In this paper, we merge the use of address assignment and key distribution into one protocol. The proposed protocol assigns IP address to the nodes in a way complies with the neighbor-based approaches and delivers a security key to these nodes in a distributed manner resulting in preserving the network power and reducing the ?ooding as well. The novelty of proposed protocol emerges by enforcing secured distributions from the beginning of node joining the MANET. Furthermore, the performance improvements of applying the proposed protocol are extremely impressive, as we will see in section 4. Although, the proposed protocol does not address the security of the MANET, it provides the nodes with security symmetric keys to use in their security protocol.

The contributions of our work can be summarized as follow:

1. We combined security key delivery and address assignment for MANET networks into one protocol.
2. We distributed the function of our protocol among the nodes to eliminate the need of a single central server.
3. The proposed protocol provides a secure delivery of both address and key assignments to the nodes in the network.

The paper organization is as follows. In the following section, a comprehensive overview on the address assignment and key distribution protocols in MANETs is provided. Section 3 details the proposed protocol for key and address assignments in MANETs. Section 4 summarizes the simulation environment and results in addition to the significance of the proposed protocol. Conclusion and future research ideas are presented in Section 5.

2 RELATED WORK

A MANET is an autonomous system of mobile nodes connected by wireless links whereas nodes are free to move and organize themselves into a dynamic network. Each node operates as an end-system and a router to relay packets to other nodes. A MANET does not require any fixed infrastructure such as base stations or dedicated servers. Therefore, it is an attractive networking option for connecting mobile devices quickly and spontaneously as adopted in military battlefields, disaster recovery, and rescue operations.

For node address assignment in MANET, many neighbor-based protocols have been proposed [43-48]. It is worth mentioning that all of these protocols were designed for address assignments only without incorporating any secure protocol. Al-Mistarihi et al. [43] proposed a tree based topology oriented auto-configuration mechanism (T-DAAP). In which, each node can have any of three roles, which are root node, leader node or normal node. Root node is the main node in the protocol, which maintains the records of the leader nodes and their address information in its database, and performs tasks of network partitioning and merging. Leader nodes possess disjoint sets of IP addresses for assignment of IP addresses to incoming nodes.

Normal nodes do not have any special function except used for routing in case of non-availability of leader in a particular area. Distributed-based dynamic address assignment protocol (D-DAAP) [44] is based on the even distribution of address pool between the new node and its allocator. MANET configurations [45, 46] allow any new node to pick a random IP address and accordingly check with the rest of network nodes if this IP address is unique or not.

The prophet address allocation scheme [47] uses a stateful address generation function *f(n)* for each node to generate a series of random numbers for new coming nodes. Function *f(n)* is carefully designed to minimize the possibility of duplication. The initial state of *f(n)* is called the seed where different seeds may lead to different sequences. Consequently, the state of *f(n)* is updated. In prime dynamic host configuration protocol (DHCP) [48], a configured node assigns an IP address to the new node in the network. It uses a prime numbering address allocation (PNAA) algorithm to generate a unique IP. However, this approach may not evenly use the available IP address pool. The maximum allowable address may be quickly reached and the network partitioning can be detected by the absence of DHCP *recycle* message generated by the root node.

Most centralized address assignment protocols are mainly based on what has been proposed in [49], that is, a single node (leader) in the network is responsible for assigning IP addresses to other nodes joining the network. In these protocols, the address uniqueness is guaranteed but the main problems include maintaining a single leader in the network and having high communication overhead with the central server. In decentralized schemes as discussed in [50], each node configures itself with an IP address and then examines for duplicate address in the network. Usually, these protocols suffer from network-wide flooding and high communication overhead. Perkins et al. [51] proposes a stateless approach where a new node selects an IP address randomly and floods the selected address to all nodes in the network. Authors in [52] proposed a group-based address autoconfiguration scheme in a way that the network is divided into groups of two hops whereas a unique identity is assigned to each group. Each group has a leader that maintains allocation tables for the assigned addresses and other group leaders' tables. Network partitioning and merging are not well-defined in this protocol. Furthermore, it does not consider the cases of graceless departure of nodes and group leaders.

Networks security is an important factor for constructing any critical network specifically adhoc networks. In traditional wired networks, the central servers are available to provide security services for users inside the network system. For example, a central server generates the keys and distributes them securely to the nodes. In MANETs, the absence of infrastructure and central server creates new challenges in their security paradigm [26].

Distributed cryptography (or security) is a cryptography scheme suited for mobile networks [53-58] since it provides robustness and defense against a single point of failure in the central server schemes [29]. In threshold cryptography, which is considered the most reliable distributed cryptography scheme, the central server function is distributed among a group of servers in an *(n,t)* threshold manner. That is any t server (t is the threshold value and $t < n$) can collaborate to generate the key for the arriving node. When one of these t servers is compromised or fails, the rest of the group (n) can take the place of this server. This distribution provides not only a co-operative security model, but also robustness against a single point of failure.

2.1 Symmetric Key Generation System (SKGS)

Blom proposed a symmetric key generation system (SKGS) [57] based on secret sharing systems for wired networks. In SKGS, nodes are supplied with a relatively small amount of secret data (key chains) that is used to derive all the nodes' keys.

In SKGS, a central server (trusted authority) generates:

1. Global matrix G of size $n \times k$, where n is the number of nodes and k is the key chain length. This G matrix will be broadcasted to all nodes in the network.

2. Symmetric secret matrix D of size $n \times n$ known by the central server only.

Then, the key matrix for the network is calculated as $K = [D\times\ G]^T \times G$ by the central server. Since D is symmetric, K will be also symmetric. Thus, for rows i and j in K, we have $K_{i,j}= K_{j,i}$. If row i is the key chain for node i, and row j is the key chain for node j, then the element $K_{i,j}$ (or $K_{j,i}$) will be the symmetric key between node i and node j.

Because G is identified by all network participants and D is identified only by the central server, this central server delivers the i^{th} *row of* $[D\times G]^T$ to node i, where i is a network member. The external nodes (malicious nodes) fail to identify these matrices. Upon reception, node i will reconstruct G by collaboration with other threshold number of neighbors and calculates its key chain using $k_i= i^{th}$ *row of* $[D\times\ G]^T\ \times G$. This multi-step division of key generation intensifies the secrecy to their key chains and makes it more difficult for the intruders to retrieve any information about other nodes. The problem of this approach is that it relies on a central server to do the whole work, which makes it vulnerable to a single point of failure. Another problem arises from key request and delivery is that some nodes may be unable to reach the central server due to broken links or untrustworthy routing information [29, 31].

2.2 Secured Shared-key Discovery (SSD)

In [55], a distributed key pre-distribution scheme (DKPS) is proposed. The scheme is a distributed symmetric key management scheme that does not rely on a trusted third party (TTP). The DKPS is composed of three parts: distributed key selection (DKS), secure shared-key discovery (SSD), and key exclusion property testing (KEPT).

In the SSD phase, nodes determine which keys that they have chosen are shared with the other nodes without using a TTP. This process must also be done without revealing the node's own key ring. The authors suggest using a privacy-homomorphism [59] to encrypt and decrypt keys. The properties of privacy-homomorphism encryption are the following for two plaintexts *x* and *y*:

1. Additive: given $E_K(x)$ and $E_K(y)$, then $E_K(x + y) = E_K(x) + E_K(y)$.
2. Scalar multiplicative: given $E_K(x)$ and scalar *t*, then $E_K(t*x)= E_K(x) * t$
3. Non-trivial zero encryption: $E_K(0)$ has many representations in the cipher text domain.

Note that for the additive and multiplicative properties, the encrypted version of two plain texts does not require the knowledge of the actual plain texts themselves if the individual encryptions are known. The authors note that the non-trivial zero encryption is a property that many protocols miss.

If node *B* wants to check which keys it shares with node *A*, then *A* will start by forming a polynomial using its own keys. The coefficients of this polynomial are encrypted and sent to *B*. Node *B* can theoretically then create encryptions of the coefficients multiplied by some random number due to homomorphism properties. These encryptions are sent back to *A*, who decrypts and should now be able to tell through decryption to zero which ones *B* sent are the same as its own keys (but *A* does not know what the keys actually are). *A* bitmap is formed and sent back to *B*. In this paper, we will use SSD to find the common keys between any two nodes key chains.

3 PROPOSED PROTOCOL DESCRIPTION

This section explains the proposed protocol functions of providing a secure distribution of IP addresses and security keys for MANET nodes. It deserves mentioning that preliminary results of this proposal have been presented in [60]. However, the proposed protocol overlaps T-DAAP address assignment and threshold based security key assignment into one protocol. After network setup, a newly joining node during the joining process receives securely an address and a security key to associate itself with the network. The network is divided into zones where each zone has one leader, taking into consideration that the whole network has only one root.

3.1 Node Types

According to their functions, network nodes can be classified into three categories: root, leader, or normal node. The following subsections explain the role and functionalities of each category.

3.1.1 The Root

Only one node in the network should be in this state. The root controls many protocol functions. Its role is as follows:

1. Maintains all zones' leaders IP addresses and their corresponding number of free IP addresses, leaders' indices needed for candidate root selection, and shared security key of each one in its database (the network initialization is discussed later in section 3.2.1).
2. Periodically sends unicast *rootAlive* messages to every leader to announce its presence provided that the *rootAlive* message contains the root IP address, leaders set, and number of free IP addresses for each leader. When the leader receives the *rootAlive* message, it will update its database with the new information of the other leaders. Moreover, the leader replies by *leaderAlive* message that contains its IP address and number of free IP addresses.
3. When the root receives the *leaderAlive* message, it will update the number of free IP addresses of that leader. If the root does not receive the message within a pre-defined time interval, it will try to find this leader by unicasting alert message. If this leader cannot be found, the root will assume that the node is no longer a leader and deletes it from its database.
4. Responsible for security key distribution to all leaders.
5. Once a security breach is detected, it is responsible for key invalidation and new keys generation.
6. Responsible for generating the security matrices *G* and *D*. We used Reed-Solomon maximum distance separable codes (MDS) [58] for matrix *G* generation that is any threshold number of rows can reconstruct the matrix G.
7. Has common security key with every leader under its supervision.

3.1.2 The Leader Nodes

Leader nodes exist in the MANET tree structure under the supervision of the root. The number of leaders in MANETs varies and depends on the area of the network. Leaders have knowledge of available free IP addresses where each leader has a disjoint set of free IP addresses in its free IP address pool. In addition, each leader retains information of the other leaders, which includes their IP addresses and corresponding number of free IP addresses. This does not accrue a heightened load on the network since it can be accomplished easily by exchanging the triggered updates advertisement packets between the leaders. Basically, these leaders are responsible for IP address and key assignments to the new nodes joining the network. Leader acts as a root for its zone and generates G_{zi} and D_{zi} matrices. G_{zi} is distributed among the zone members in a threshold manner and D_{zi} is known only by the leader only. Leaders also preserve a table of zone node indexes along with shared security key between the leader and every node. The leader periodically broadcasts its presence implicitly by including the node type in the hello messages. After initialization, each normal node in the MANET has to be in the neighbourhood of at least one leader.

3.1.3 The Normal Nodes

The normal nodes can relay IP addresses and security keys assignments packets to the new nodes if the leaders are outside their transmission range. Additionally, they respond to the leader through periodic messages and allow the leader to find the used address space and departed nodes. Furthermore, normal nodes have the generator matrix *G* distributed among them in a *(n, t)* threshold manner, where *n* is the number of nodes inside the zone and *t* is the threshold value.

3.2 Address and Key Assignments

This section explains how the proposed protocol assigns IP addresses and security keys to the nodes in the MANET. For convenience and practical issues, zones have different matrices (G_{zi} and D_{zi}, where *i* is the zone leader index at the root). This reduces the threshold values, which in turn mitigate the number of computations in generating and regenerating the generator matrix G_{zi}, since this matrix is related to the number of nodes inside the zone only.

3.2.1 Network Initialization

At the initialization stage of the network, there will be a lower number of nodes than the threshold value, the following steps occur during initialization:

1. The network follows T-DAAP initialization phase through selecting a root node that will assign IP addresses only to the nodes at this stage until there is enough number of nodes (*no. of nodes* $\geq$ *threshold*).
2. Once the number of nodes reaches the threshold value, the root generates G_R and D_R matrices. Where any threshold (*t*) rows of G_R can be used to reconstruct G_R.
3. The root distributes rows of G_R to each node (where these nodes will be the leaders) whereas D_R will be known by the root only assuming that all these nodes are legitimate and not malicious.
4. The root calculates A_R $(A_R = [D_R \times G_R]^T\ )$ and delivers to each leader *i* its corresponding row *i*.
5. Each leader *i* collaborates with other leaders (any threshold number of leaders) to reconstruct G_R and calculates its key chain using k_i=[(i^{th} *row of* A_R)$\times$ G_R].
6. The root backs-up its information (leaders' IP addresses and security keys control information) on the leader that has the lowest index to be the candidate root. We will see the importance of this step in the root departure subsection shortly.
7. Now, leaders can mobilize to cover all the network area uniformly. Communication between any two leaders can take place securely even if they are out of range to each other since mutual shared keys are established between them (these nodes are the leaders).

The same steps are carried out by any leader located within its zone. The leader has to generate another G_{zi} and D_{zi} and distribute them to the normal nodes once there is enough number of nodes *(no. of nodes* $\geq$ *threshold)* inside its zone. The leader then backs-up its control information on one of its zone nodes that has the highest IP address value (the role of this back-up will be discussed in leader departure, section 3.2.4). It must be mentioned that the threshold value of the leader does not have to be the same as that of the normal nodes, that is to say, the number of leaders do not have to be the same as the number of zones' normal nodes. In addition, we assume that the nodes during network initialization are not of malicious behaviour.

3.2.2 Node Joining the Network

After the network initialization, any node that joins the network, which is called a requestor, will receive replies for its *addressKeyRequest* broadcast message. The reply message contains the assigned IP address for this node and a row from Azi corresponding to the node index at the leader of that zone. After the *addressKeyReqTimer* timer expires, the new node checks the responses and sets the *allocatorChosenTimer* timer. According to the received responses, one of the following possibilities may occur:

1. At least one leader that has free IP addresses is found in the responses. The node chooses the leader that has the largest free IP addresses to be the allocator and sends *allocatorChosen* message to that node. Upon receiving this message, the chosen leader calculates A_{zi} = $(D_{zi} \times G_{zi})^T$ matrix, removes the highest available IP address from its free IP address pool, as well as assigns it to the new node along with the netId, leader IP address, and i^{th} row of A$_{zi}$ via *addressKeyAssign* message.

2. None of the leaders in the responses has addresses. Node chooses one of the leaders randomly and sends *allocatorChosen* message to the selected one. When the leader receives this message, it will send a *waitPeriod* message to the requestor and extend the *allocatorChosenTimer* timer. Next to that, the allocator starts the IP address search process by broadcasting hello message within its zone in order to find a free IP address to be assigned to the new node. Once a free IP address is found, the chosen leader will calculates A_{zi} = $(D_{zi} \times G_{zi})^T$ matrix, removes the highest available IP address from its free IP address pool, and assigns it to the new node along with the netId, leader IP address, and i^{th} row of A_{zi} via *addressKeyAssign* message.

3. If none of the replies is originated from a leader, it will pick a random neighbour node from the replies and select it as its allocator. This allocator will try to search for an IP address to this new node by extending the *allocatorChosenTimer* timer. It will then start searching for a leader that has free IP addresses by broadcasting the *findLeaderAllocator* message. Any leader receives the *findLeaderAllocator* message replies with the *findLeaderAllocatorReply* message that contains its IP address and number of free IP addresses. The relay selects the leader that has the highest number of free IP addresses to be the remote allocator by sending the *leaderAllocatorChosen* message. When the leader receives this message, it will remove the highest available IP address from its free IP address pool and send it to the relay via *leaderAddressAssign* message. In this case, the normal nodes act as a relay between the requestor and remote allocator that eventually sends the IP address to the new node via *forwardIpAssign* message. When the *allocatorChosenTimer* timer expires, the node will check its status. If it finds that it is not configured, it will broadcast *addressKeyRequest* message and start again. If it is configured, it will ask any *t* neighbour nodes to collaborate to generate *Gzi* matrix. After receiving a threshold number of rows for G_{zi} from any *t* neighbours, the requestor calculates K_i = *(*i^{th} *row of* A_{zi}*)*$\times G_{zi}$ as shown by the flow chart discussed in Fig. 1.

When the new node happens to be of malicious nature, the proposed protocol will assign it an address and a key chain. Notice that, even this node is malicious and received an address and key chain, it will know which parts of its key chain is common with other nodes only, and will not be able to reveal any full key chain of another node. Next, the IDS system will be in charge to detect this node and isolate it from the network and may rekey the network reactively as in section 3.2.6.

3.2.3 Root Departure

If the root leaves the network for any reason, the leaders will not receive any *rootAlive* message. The back-up root, once it notices the absence of this *rootAlive* message, tries to find the root by broadcasting *areYouAliveRoot* message. If the root does not reply to this message, this back-up root will announce itself as the root and broadcast a *rootAlive* message periodically. Because this root contains the whole network back-up information from the previous root, the network will be operational immediately without disruption. For security purposes, the new root re-keys the leaders by generating new G_R and D_R and then distributes the rows of the new matrices as discussed in section 3.2.1. Thereafter, the new root backs-up this network information on the leader that has the lowest index to be the candidate root. During this process, the back-up leader continuously sends *hello* message to its zone to postpone leader departure action in its designated zone

3.2.4 Leader Departure

Due to node movement, the leader may leave its zone and join another zone. Additionally, it may run out of battery power, get stolen or attacked by opponents, or change its type to be the root. When any of these happens, the back-up leader must take the place of the leader. Because this leader has the zone back-up information from the previous leader, zone operations will proceed without disruption. The new leader then backs-up the zone information on one of the normal nodes available within its zone. The network reaction steps to the leader departure which is summarized in sequence as follows:

1. Once the back-up leader detects the leader migration with the absence of *hello* message, it mobilizes to become the new leader.

2. The new leader registers itself at the root by *registerLeader* message.

3. Root repeatedly looks for the previous leader by sending a *leaderAlive* message.

4. If the previous leader does not reply to the *leaderAlive* message, the root frees its IP address from its address table.

5. The new leader collects G_R by collaborating with neighbouring leaders and generates its key chain. The root will inform leaders not to respond to G_R's request except from this new leader node.

6. The root then builds new G_R and D_R matrices and securely distributes the G_R rows and $A_R = [D_R \times G_R]^T$ rows to the leaders using previous keys. Each leader calculates its new key chain by collaboration.

7. The new leader backs-up its information on a node located within its zone according to the highest IP address value criteria.

8. If the migrating leader is the root-back-up leader, then the root will detect this migration with the absence of *leaderAlive* message. The root in turn re-keys the leader-level nodes and backs-up the network data on another leader node.

3.2.5 Normal Node Departure

Normal nodes can leave the zone and enter another zone or sometimes may leave upon abnormal event like power drain. In normal operation, the nodes attempt to leave the zone must inform the zone leader for the sake of deleting its address and accordingly assigning it to a new node (if there is any). If the nodes left the zone abruptly, the zone leader detects that by periodic update replies. On the other hand, if the migrating node is the zone-back-up leader, the leader re-keys the zone nodes and backs-up the zone data on another normal node.

3.2.6 Security Key Proactive and Reactive Update

Due to the shared medium and tapping in MANET, some keys may be revealed or stolen by an intruder. For that reason, we propose reactive and proactive key update in which the zone leader must change the zone keys after a specified period or event by creating a new D_z matrix (G_z matrix will not be changed). When a leader detects a key compromise or any suspicious activity in the network, it will initiate the reactive key update phase by broadcasting an *invalidateKey* packet to the nodes located within its zone. When a node inside the zone receives this packet, it will invalidate its address key table and wait for the leader to deliver the key vector (i^{th} row of A_z) to it. Consequently, it recalculates its news key chain using $K_i = [i^{th}$ row of $A_z \times G_z]$.

3.3 Zone Communications

3.3.1 Intra Zone Communication

When two nodes i, j found inside the same zone (i.e., have the same leader) want to communicate with each other, the communication can take place immediately since they already have a common key inside their key chains (i.e., $K_{ij} = K_{ji}$).

3.3.2 Inter Zone Communication

When nodes x, y located in different zones need to exchange packets, the exchange process cannot pass off immediately since there are no known common keys in their key chains. They first must find common keys between them. This can be done with the help of secured shared key discovery (SSD) [55] and the relay function of leaders. After SSD is carried out, the two nodes will know the common keys between their key chains. This can be done through the following steps:

1. Node x informs its leader that it wants to communicate with node y but it does not have a shared key with it.

2. Leader of node x locates the leader of node y, generates a temporary G matrix *($G_{interzone\_temp}$)*, and delivers it to the leader of nodes y and x securely.

3. Leader of node y delivers $G_{interzone_temp}$to node y securely.

4. Node *x* picks a random vector in *GF(q)* and calculates B_x=V_x× $G_{interzone\_temp}$,

5. Node y also picks a random vector in *GF(q)* and calculates B_y=V_y× $G_{interzone\_temp}$

6. Nodes x and y apply SSD with the help of the leaders as relays to find the common keys between $B_x$andB_y. Authors in [54] found that the probability of finding common keys between x and y is approximately 90% when both nodes use the same $G_{interzone_temp}$assuming that *GF* parameters were chosen carefully.

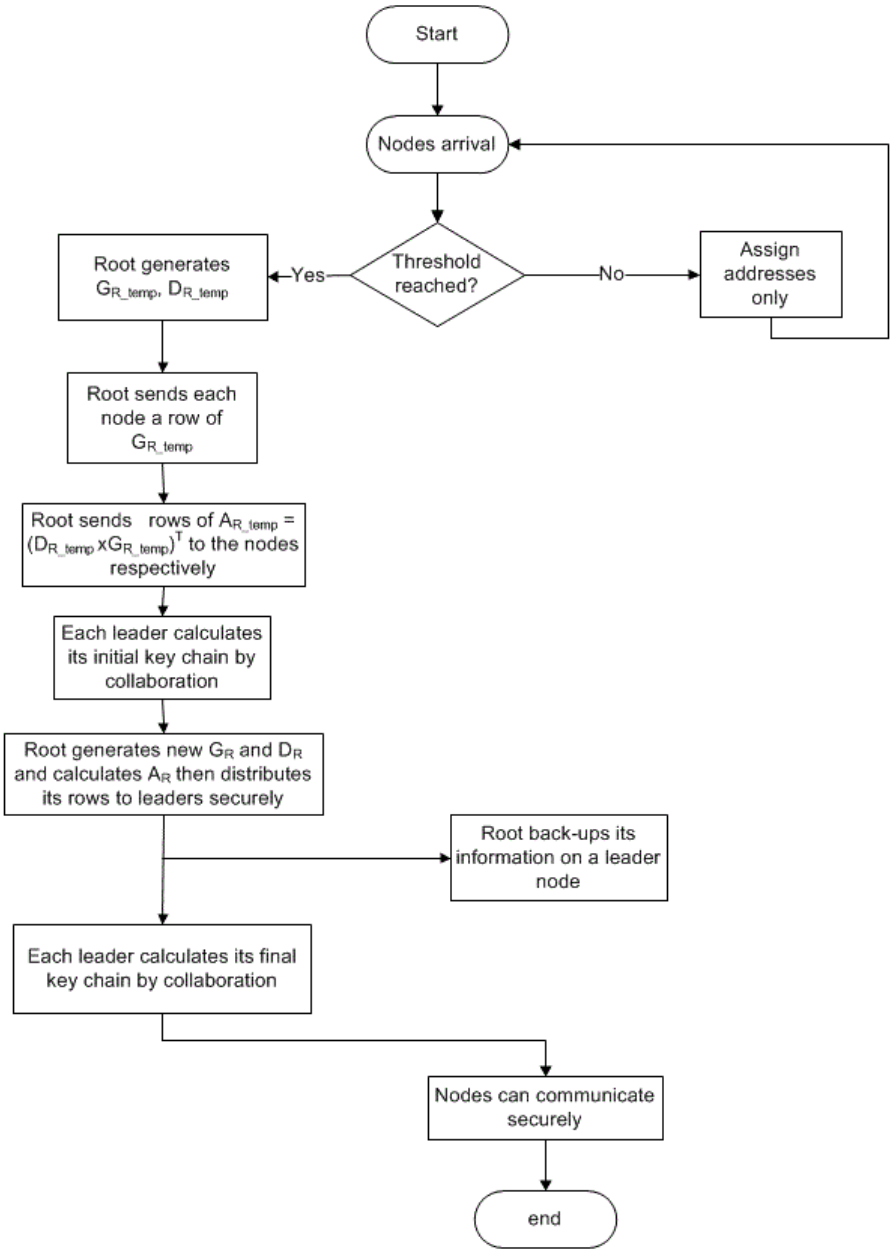
Start
Nodes arrival
Threshold reached?
Yes
No
Assign addresses only
Root generates G_{R_temp}, D_{R_temp}
Root sends each node a row of G_{R_temp}
Root sends rows of $A_{R\_temp} = (D_{R\_temp} \times G_{R\_temp})^T$ to the nodes respectively
Each leader calculates its initial key chain by collaboration
Root generates new G_R and D_R and calculates A_R then distributes its rows to leaders securely
Root back-ups its information on a leader node
Each leader calculates its final key chain by collaboration
Nodes can communicate securely
end

Fig. 1. Network initialization

7. If no common keys are found, nodes *x* and *y* generate another random vector and apply SSD again until a common key is found. Communication can occur as soon as a common key is found between nodes *x* and *y*.

With the help of leaders, inter zone communication takes place always securely even if one of the zones (or both zones) has less number of nodes than the zone threshold. The same steps as discussed previously insure a secure key between the inter zone participants.

3.4 Network Scalability

Since nodes join the network randomly, some leaders may have more nodes than the upper limit of threshold security approach. Therefore, the network must scale well for joining members. In these cases, the zone that exceeds the threshold limit will be forced to split into two zones as follows:

1. The original zone leader informs the back-up leader to be a new leader.

2. The original leader broadcasts *hello* message and then picks half of the nodes based on their signal strength to be under its administration.

3. The original zone leader sends the new leader a list of the remaining nodes, which will be under its administration along with half of its address space.

4. Both leaders now assign addresses and rekey their nodes. After that, they back-up their information on a back-up node as discussed in sections 3.2.3 and 3.2.4.

5. The original leader will act as a root (Zroot$_i$) for the new collection and can accept more nodes to join and create more zones.

6. The original leader will be a leader for upstream root (the main root) and a root (Zroot$_i$) for the downstream nodes.

4 SIMULATION AND RESULTS

To evaluate the performance of the proposed protocol, several simulation experiments were performed. Network Simulator ns-2 (version 2.33) was used for this task. Performance was evaluated by measuring the number of unicast messages per address assignment, number of control messages per node during the simulation time, and average address assignment latency.

The unicast messages measured during the simulation are the unicast messages that are used to assign an IP address and other configuration parameters to the unconfigured node. These messages include *addressReqReply, allocatorChosen, addressAssign, findLeaderAllocatorReply, leaderAllocatorChosen, forwardIpAssign,* and *waitPeriod.*

The control messages are the messages that are used to maintain the tree structure of the protocol such as the root, leader selection, and advertising. These messages are *rootAlive, leaderAlive, newRootAlive,* and *registerLeader.*

4.1 Simulation Scenarios

The proposed protocol was tested under various conditions and distribution models. In these tests, the following parameters were used:

1. Random waypoint mobility model.

2. Network area is 1000 m×1000 m.

3. Nodes move with a maximum speed of 5 meters/second.

4. The routing protocol used was the ad hoc on demand distance vector (AODV).

5. Transmission range of the node is 100 m.

6. Data link layer was IEEE 802.11 for all nodes.

7. The number of nodes in the network is 1000 node including 20 leaders.

8. The network has 20 zones initially in which no more than 50 nodes available in each zone. The number of zones and zone members can scale relative to section 3.4.

The proposed protocol was tested and compared with the other well-known protocols (i.e., T-DAAP, D-DAAP for address assignment and threshold security protocol for key assignment) in terms of not only the number of control packets needed for address and key assignments, but also the address and key assignments delay. To show the significant contributions of our proposed protocol, evaluations concerning the channel throughput and required MAC layer packets are extensively discussed. The results of the simulation demonstrate how the performance of the proposed protocol is affected by the node population and network density.

For T-DAAP and D-DAAP with threshold cryptography simulation, we assign addresses first then keys, since to distribute keys, the node must be identified and has an address to send the key to this address, while in the proposed protocol, a security key and address will be assigned at the same time.

4.2 Simulation Results

4.2.1 Address and Key Assignments Control Packets

Figure 2 depicts the average number of control packets per node for address and key assignments. We can see that the D-DAAP and threshold security require the highest number of assignment packets for all network sizes. Compared to D-DAAP and threshold security, T-DAAP and threshold security require a lower number of assignment packets. Compared to aforementioned proposed protocols, our proposed protocol clearly shows a reduction in the number of control packets needed per node for address and security key assignments. As a matter of fact, the increase in both approaches (T-DAAP or D-DAAP along with threshold security) is justified due to the increase in the threshold value to recover *G*. However, as the network grows in size and when the number of nodes reaches the threshold value, our proposed protocol outperforms the other protocols as far as the number of required control packets is concerned. In fact, our proposed protocol shows an enhancement of about 30% when the network nodes are 1000 compared to T-DAAP and threshold security scheme. On the other hand, when having 200 nodes (network threshold value), the proposed protocol requires almost the same number of control packets as required in the separate protocols (T-DAAP and threshold security scheme).

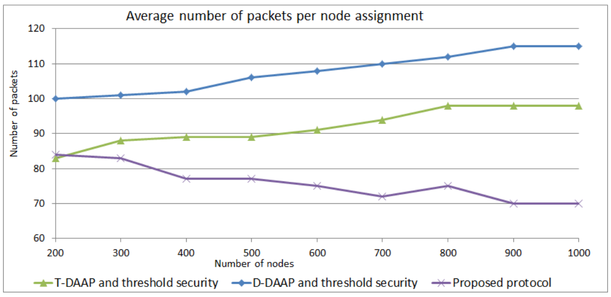

Fig. 2. Average number of control packets/node assignments

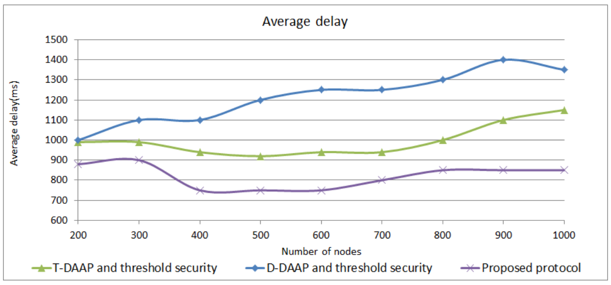

Fig. 3. Average number of control packets/node assignments

4.2.2 Address and Key Assignments Delay

Figure 3 shows the average delay required for address and security key assignments to the node using reference protocols and our proposed protocol. As far as the delay of D-DAAP and threshold security is concerned, we can see that it tends to increase when the number of network nodes is dramatically increased. On the other hand, the delay of T-DAAP and threshold security increases with a slower pace than D-DAAP and threshold security as long as the number of nodes increases due to a larger broadcast domain. It is clearly noticed that the proposed protocol shows a great reduction in the assignment delay. When the network nodes get larger and larger, this delay can be up to 26% less than that when employing T-DAAP and threshold security scheme.

4.2.3 MAC Layer Packets

The results of our proposed protocol, shown in Fig. 4, demonstrate that the MAC layer packets are significantly less compared to that when using separate protocols. As the number of nodes in the network increases, it will incur more load on the MAC layer for channel allocation, more collisions and retransmissions. It is quite interesting to notice that the proposed protocol saves about 40% of the packets to perform the same job done by T-DAAP and threshold security scheme while the amount of saving is increased to 50% of the packets once D-DAAP and threshold security scheme are employed. Positively, this significant load reduction has a great impact on the throughput of the network as will be seen shortly in the next subsection. The reason for this enhancement is due to less control packets needed to assign and maintain address and security keys by our merged protocol compared to other separate protocols.

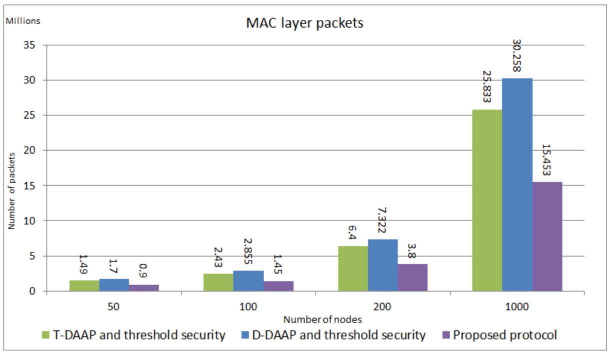

Fig. 4. MAC layer packets

4.2.4 Channel Throughput

Logically visible that mitigating the channel congestion certainly leads to having an increase in the throughput of the network since the probability of finding the channel status to be “idle” is high. Fig. 5 emphasizes this through considering different number of nodes in the network. For example, when the number of nodes is 1000, we can notice that the proposed protocol achieves a throughput of about 123 while in T-DAAP and threshold security, they achieve together a throughput of about 85. Moreover, the obtained throughput of D-DAAP and threshold security is about 75. Therefore, approximately 45% enhancement is obtained, compared with the best currently proposed, which consequently indicates that the probability of having network congestion when using the proposed protocol is much lower than that when using two independent protocols (T-DAAP and threshold security or D-DAAP and threshold security). On the other hand, the effect of increasing the number of nodes in the network has a slight effect on the throughput. For example, when the number of nodes dips down from 1000 to 50, the throughput variation is small which undoubtedly means that the proposed protocol is ultimately scalable.

A noteworthy effect of node’s movement speed on the throughput is observed, that is when the nodes movement speed is low, then the throughput of the proposed protocol is almost similar to T-DAAP and threshold security together, while in high speed the throughput for the proposed

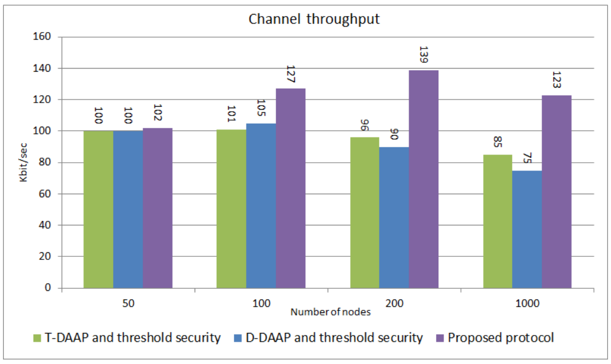

Fig. 5. Channel Throughput

protocol is higher, the logical reason for this is due the fact that nodes movement from zone to zone is higher when the nodes movement is high, and thus means more time needed to assign a key and address for the node, add to that a higher channel congestion due to higher demand for address and key assignment for leaving nodes from zone to zone.

4.2.5 Protocol Reliability and Security

The proposed protocols is not meant to provide security to the network, it just assign addresses and keys to be used then for security. The proposed protocol ability to assign keys and addresses to any new node was investigated. When the new node happens to be of malicious nature, the proposed protocol will assign it an address and a key chain. Notice that, even this node is malicious and received an address and key chain, it will know which parts of its key chain is common with other nodes only, and will not be able to reveal any full key chain of another node. Next, the IDS system will be in charge to detect this node, isolate it from the network, and may re-key the network reactively. In addition, networks active nodes can neglect the request from nodes identified as malicious by IDS about matrix G. Consequently, the approach will distribute keys and addresses reliably and securely (no other node can calculate other node's key chain) to new nodes.

The security protocol is slightly better than that of the two separate protocols as it depends on the IDS ability to detect and react the malicious nodes. With higher throughput and lower delay for the proposed protocol, the IDS will detect malicious nodes faster compared with the two separate protocols. Besides, attacks can target key assignment only or address assignment only. Thus, fewer attacks can affect the proposed protocol compared with separate protocols.

5 CONCLUSION AND FUTURE WORK

In this paper, we succeeded eventually to develop a yet efficient protocol that overlaps address and key assignments for MANET nodes. Simulation results are very promising and show impressive improvements in the packet delay, channel throughput, and number of control packets needed to perform the address and key assignments, as well as MAC layer packets over that obtained in the other known separate protocols (T-DAAP and threshold security or D-DAAP and threshold security). As a summary and compared with the best available in the literature, we obtain the improvements of about 26%, 45%, 30%, and 40% for the packet delay, channel throughput, control packets, and MAC layer packets, respectively. However, there are a number of potential future research directions for this work. It will be extremely interesting to modify the proposed protocol to accommodate two different networks along with two different roots. On the other hand, studying the effect of threshold value variation on the behavior of proposed protocol will be a valuable addition.

REFERENCES

[1] K. A. Darabkh, I. Jafar, G. Al Sukkar, G. Abandah, and R. Al-Zubi, “An Improved Queuing Model for Packet Retransmission Policy and Variable Latency Decoders,” *IET Communications*, vol. 6, no. 18, pp. 3315–3328, December 2012.

[2] K. A. Darabkh and R. S. Aygun, "Performance Evaluation of Sequential Decoding System for UDP-Based Systems for Wireless Multimedia Networks", *Proceedings of 2006 International Conference on Wireless Networks (ICWN'06)*, Las Vegas, Nevada, pp. 365-371, June 2006.

[3] K. A. Darabkh and R. Aygun, “Improving UDP Performance Using Intermediate QoD-aware Hop System for Wired/Wireless Multimedia Communication Systems,” *International Journal of Network Management*, vol. 21, no. 5, pp. 432–454, September 2011.

[4] K. A. Darabkh and R. S. Aygun, "Quality of Service and Performance Evaluation of Congestion Control for Multimedia Networking", *Proceedings of 2006 International Conference on Internet Computing (ICOMP'06)*, Las Vegas, Nevada, pp. 217-223, June 2006.

[5] K. A. Darabkh and R. S. Aygun, "Quality of Service Evaluation of Error Control for TCP/IP-Based Systems in Packet Switching ATM Networks", *Proceedings of 2006 International Conference on Internet Computing (ICOMP'06)*, Las Vegas, Nevada, pp. 243-248, June 2006.

[6] K. A. Darabkh, B. Abu-Jaradeh, and I. Jafar, “Incorporating Automatic Repeat Request and Thresholds with Variable Complexity Decoding Algorithms over Wireless Networks: Queuing Analysis,” *IET Communications*, vol. 5, no. 10, pp. 1377–1393, July 2011.

[7] K. A. Darabkh, "Evaluation of Channel Adaptive Access Point System with Fano Decoding," *International Journal of Computer Mathematics*, vol. 88, no. 5, pp. 916–937, March 2011.

[8] K. A. Darabkh, "Queuing Analysis and Simulation of Wireless Access and End Point Systems using Fano Decoding," *Journal of Communications*, vol. 5, no. 7, pp. 551-561, July 2010.

[9] K. A. Darabkh and B. Abu-Jaradeh, "Bounded Fano Decoders over Intermediate Hops Excluding Packet Retransmission," *Proceedings of IEEE 24th International Conference on Advanced Information Networking and Applications (AINA 2010),* , Perth, Australia, pp. 299-303, April 2010.

[10] K. A. Darabkh and B. Abu-Jaradeh, "Buffering Study over Intermediate Hops including Packet Retransmission," *Proceedings of IEEE International Conference on Multimedia Computing and Information Technology (MCIT-2010)*, Sharjah, U.A.E, pp. 45-48, March 2010.

[11] Khalid A. Darabkh, "Fast and Upper Bounded Fano Decoding Algorithm: Queuing Analysis," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 1, pp. 1-12, January 2017.

[12] K. A. Darabkh and R. S. Aygün, "TCP Traffic Control Evaluation and Reduction over Wireless Networks Using Parallel Sequential Decoding Mechanism," *EURASIP Journal on Wireless Communications and Networking*, vol. 2007, Article ID 52492, pp.1-16, November 2007.

[13] Mamoun F. Al-Mistarihi, Rami Mohaisen, Ashraf Sharaqa, Mohammad M. Shurman,and Khalid A. Darabkh, "Performance Evaluation of Multiuser Diversity in Multiuser Two-Hop Cooperative Multi-Relay Wireless Networks using MRC over Rayleigh Fading Channels," *International Journal of Communication Systems*, vol. 28, no. 1, pp. 71-90, January 2015.

[14] M. Shurman, M. Al-Mistarihi, A. Mohammad, K. Darabkh, and A. Ababnah,"Hierarchical Clustering Using Genetic Algorithm in Wireless Sensor Networks," *Proceedings of 36th IEEE International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO 2013),* , Opatija, Croatia, pp. 479-48, May 2013.

[15] K. A. Darabkh, S. Ismail, M. Al-Shurman, I. Jafar, E. Alkhader, and M. Al-Mistarihi, "Performance evaluation of selective and adaptive heads clustering algorithms over wireless sensor networks," *International Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 2068–2080, November 2012.

[16] Mohammed Hawa, Khalid A. Darabkh, Loay D. Khalaf, and Jamal S. Rahhal, "Dynamic Resource Allocation Using Load Estimation in Distributed Cognitive Radio Systems," *AEÜ - International Journal of Electronics and Communications*, vol. 69, no. 12, , pp. 1833–1846, December 2015.

[17] Khalid A. Darabkh, Abeer M. Awad, and Ala' F. Khalifeh, "New Video Discarding Policies for Improving UDP Performance over Wired/Wireless Networks," *International Journal of Network Management*, vol. 25, no. 3, pp. 181–202, May/June 2015.

[18] Khalid A. Darabkh, Huda IbeidAffiliated withDepartment of Computer Engineering, The University of Jordan, Iyad F. Jafar, Raed T. Al-Zubi, "A generic buffer occupancy expression for stop-and-wait hybrid automatic repeat request protocol over unstable channels," *Telecommunication Systems*, DOI 10.1007/s11235-015-0115-5.

[19] Khalid A. Darabkh and Ola Alsukour, "Novel Protocols for Improving the Performance of ODMRP and EODMRP over Mobile Ad hoc Networks," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 348967, pp.1-18, October 2015.

[20] Raed Al-Zubi, Marwan Krunz, Ghazi Al-Sukkar, Mohammed Hawa, and Khalid A. Darabkh, "Packet Recycling and Delayed ACK for Improving the Performance of TCP over MANETs*," Wireless Personal Communications*, vol. 75, no. 1, pp. 943-963, March 2014.

[21] Mohammad Shurman, Mohammad Alfawares, Mamoun F. Al-Mistarihi, and Khalid A. Darabkh, "A Collaborative Reputation Approach to Avoid Misbehaving Nodes in MANETs," *Proceedings of the 2014 IEEE International Multi-Conference on Systems, Signals & Devices, Conference on Communication & Signal Processing,* Castelldefels-Barcelona, Spain, pp. 1-4, February 2014.

[22] Mohammad Shurman, Noor Awad, Mamoun F. Al-Mistarihi, and Khalid A. Darabkh,"LEACH Enhancements for Wireless Sensor Networks Based on Energy Model," *Proceedings of the 2014 IEEE International Multi-Conference on Systems, Signals & Devices, Conference on Communication & Signal Processing,* Castelldefels-Barcelona, Spain, pp. 1-4, February 2014.

[23] S. S. Ismail, A. I. Al Khader, and K. A. Darabkh," Static Clustering for Target Tracking in Wireless Sensor Networks," *Global Journal on Technology (Selected Paper of COMENG-2014)*, vol. 8, pp. 167-173, 2015.

[24] Khalid A. Darabkh, Wijdan Y. Albtoush, and Iyad F. Jafar, "Improved Clustering Algorithms for Target Tracking in Wireless Sensor Networks," *Journal of Supercomputing*, DOI: 10.1007/s11227-016-1898-1, October 2016.

[25] Khalid Darabkh, Noor Al-Maaitah, Iyad Jafar, and Ala Khalifeh, "Energy Efficient Clustering Algorithm for Wireless Sensor Networks," *Proceedings of 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET 2017)*, Chennai, India, March 2017.

[26] B. Wu, J. Wu , and M. Cardei, "A Survey of Key Management in Mobile Ad Hoc Networks," *Handbook of Research on Wireless Security*, Y. Zhang, J. Zheng, and M. Ma (eds.), Idea Group Inc., ISBN: 978-1-59904-899-4, 2008.

[27] Y. Zhang, W. Liu, and W. Lou, "Anonymous Communications in Mobile AdHoc Networks", *Proceedings of the 24th IEEE International Conference on Computer Communications (INFOCOM 2005)*, Miami, Florida, USA, pp.1940-1951, March 2005.

[28] S. C.-H. Huang, F. Yao, M. Li, and W. Wu, "Lower bounds and new constructions on secure group communication schemes," *Theoretical Computer Science*, vol. 407, no. 1, pp. 511-523, November 2008.

[29] N. Vimala and R. Balasubramaniam, "Distributed Key Management Scheme for Mobile Ad-Hoc Network-A Survey," *Global Journal of Computer Science and Technology,* vol. 10, no. 2, pp. 7-11, April 2010.

[30] J. Sheu, S. Tu, and L. Chan, "A Distributed IP Address Assignment Scheme for Ad Hoc Networks," *Proceedings of the 11th International Conference on Parallel and Distributed Systems 2005 Proceedings (ICPADS'05)*, Fukuoka, Japan, pp. 439- 445, July 2005.

[31] S. Kim, J. Lee, and I. Yeom, "Modeling and Performance Analysis of Address Allocation Schemes for Mobile Ad Hoc Networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 1, pp. 490 - 501, January 2008.

[32] X. Chu, Y. Sun, K. Xu, Z. Sakander, and J. Liu, "Quadratic Residue Based Address Allocation for Mobile Ad Hoc Networks," *Proceedings of the IEEE International Conference on Communications (ICC'08),* Beijing, China, pp. 2343 - 2347, May 2008.

[33] S. Yi and R. Kravets,"Moca: Mobile Certificate Authority for Wireless Ad Hoc Networks," *Proceedings of the 2nd Annual PKI Research Workshop (PKI'03),* Gaithersburg, Maryland, USA, April 2003.

[34] Y. Ozturk and V. Nagarnaik, "A scalable distributed dynamic address allocation protocol for ad-hoc networks," *Journal of Wireless Networks,* vol. 17, no. 2, pp. 357-370, February 2011.

[35] X. Li, Y. Deng, V. Narasimhan, A. Nayak, and I. Stojmenovic , "Localized address autoconfiguration in wireless ad hoc networks," *Proceedings of 2010 International Conference on Wireless Communications and Signal Processing (WCSP'10),* Suzhou, China, pp.1-6, October 2010.

[36] N.I.C. Wangi, R.V. Prasad, M. Jacobsson, and I. Niemegeers, "Address autoconfiguration in wireless ad hoc networks: protocols and techniques," *Proceedings of IEEE Wireless Communication*, vol.15, no.1, pp.70-80, February 2008.

[37] S. Park, E. J. Lee, J. H. Ryu, S.-S. Joo, and H. S. Kim, "Distributed Borrowing Addressing Scheme for ZigBee/IEEE 802.15.4 Wireless Sensor Networks," *ETRI Journal*, vol.31, no.5, pp525-533, October 2009.

[38] E. Ancillotti, R. Bruno, M. Conti, A. Pinizzotto, "Dynamic address autoconfiguration in hybrid ad hoc networks," *Pervasive and Mobile Computing*, vol. 5, no. 4, pp.300-317, August 2009.

[39] N. Fernandes, M. Moreira, O. Carlos, and M. Duarte,"An Efficient Filter-based Addressing Protocol for Autoconfiguration of Mobile Ad Hoc Networks," *Proceedings of the 28th IEEE International Conference on Computer Communications (INFOCOM 2009),* Rio de Janeiro, Brazil, pp.2464-2472, April 2009.

[40] L. Zhou, Z. Haas, "Securing adhoc networks, " *IEEE Network Magazine*, vol.13, no.6, pp. 24-30, November 1999.

[41] H. Zhou, M. Mutak, and L. Ni, "Secure Autoconfiguration and Public-key Distribution for Mobile Ad-hoc Networks," *Proceedings of 6th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2009)*, Macau SAR, China, pp. 256-263, October 2009.

[42] M. Taghiloo, J. Taghiloo, M. Dehghan, "A Survey of Secure Address Autocon?guration in MANET", *Proceedings of the 10th IEEE Singapore International Conference on Communication Systems (ICCS2006)*, Singapore, pp. 1-5, March 2006.

[43] M. F. Al-Mistarihi M. Shurman, and A. Qdaimat, "Tree based dynamic address autoconfiguration in mobile ad hoc networks," *Computer Networks,* vol. 55, no. 8, pp. 1894-1908, February 2011.

[44] M. R. Thoppian and R. Prakash, "A Distributed Protocol for Dynamic Address Assignment in Mobile Ad-Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 5, no.1, pp. 4-16, January 2006

[45] K. Weniger, M. Zitterbart, "Address Autoconfiguration in Mobile Ad-Hoc Networks: Current Approaches and Future Directions," *IEEE Network Magazine (Special issue on Ad-Hoc Networking)*, vol. 18, no. 4, pp. 6-11, July 2004.

[46] S. Nesargi and R. Prakash, "MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network," *Proceedings of the 21st Annual Joint Conference of IEEE Conference on Computer Communication (INFOCOM'02)*, New York, NY, USA, pp. 206-216, June 2002.

[47] H. Zhou, L. Ni, and M. Mutka, "Prophet Address Allocation for Large Scale MANETs," *Proceedings of the 22nd Annual Joint Conference of IEEE Conference on Computer Communication (INFOCOM'03)*, San Francisco, CA, pp. 1304-1311, April 2003.

[48] Y. Hsu and C. Tseng, "Prime DHCP: A Prime Numbering Address Allocation Mechanism for MANETs," *IEEE Communications Letters*, vol. 9, no. 8, pp. 712-714, August 2005.

[49] R. Droms, "Dynamic Host Configuration Protocol," *Network Working Group* – RFC 2131, March 1997.

[50] H. Kim, S. Kim, M. Yu, J. Song, P. Mah, "DAP: Dynamic Address Assignment Protocol in Mobile Ad-hoc Networks, " *Proceedings of the 11th Annual IEEE International Symposium on Consumer Electronics (ISCE 2007),* Dallas, Texas, USA, pp. 1-6, June 2007.

[51] C. Perkins, J. Malinen, R. Wakikawa, E. Royer, and Y. Sun, "IP Address Autoconfiguration for Ad Hoc Networks," *IETF Internet draft*, draft-ietf-manet-autoconf-01.txt, November 2001.

[52] S. Mukhtar, "Group based Address Autoconfiguration Scheme for Mobile Ad-Hoc Networks," *Proceeding of Student Conference on Engineering Sciences and Technology (SCONEST 2005)*, Karachi, Pakistan, pp. 1- 4, August 2005.

[53] K.-H. Lee, S.-B. Han, H.-S. Suh, C.-S. Hwang, and S. Lee, "Authentication Protocol Using Threshold Certification in Hierarchical-cluster-based Ad Hoc Networks," *Journal of Information Science and Engineering*, vol. 23, no. 2, pp. 539-567, March 2007.

[54] M. Al-Shurman, S-M. Yoo, B. Kim, "Distributive Key Management for Mobile Ad Hoc Networks," *Proceedings of 2008 International Conference on Multimedia and Ubiquitous Engineering (MUE 2008)*, Busan, Korea, pp. 533- 536, April 2008.

[55] A. C-F. Chan, "Distributed Symmetric Key Management for Mobile Ad Hoc Networks," *Proceedings of the 23rd IEEE International Conference on Computer Communications (INFOCOM 2004)*, Hong Kong, China, pp. 2414 - 2424, March 2004.

[56] T. Matsumoto, H. Imai, "On the Key Predistribution System: A Practical Solution to the Key Distribution Problem," *Proceedings of the Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology (CRYPTO '87)*, Santa Barbara, California, USA, Lecture Notes in Computer Science 293 Springer 1988, pp.185-193, 1988.

[57] R. Blom, "An Optimal class of symmetric key generation systems," *Proceedings of the EUROCRYPT84 workshop on Advances in cryptology: theory and application of cryptographic techniques*, Paris, France, pp. 335-338, April 1984.

[58] I. S. Reed, G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics (SIAM),* vol.8, no.2, pp. 300-304, June 1960.

[59] R. Rivest, L. Adleman, and M. Dertouzos. "On Data Banks and Privacy Homomorphisms", *in Foundations of Secure Computation, Academic Press,* pp. 169-179, 1978.

[60] M. Shurman, M. Al-Mistarihi, and K. Darabkh,"Merging Dynamic Address Autoconfiguration and Security Key Protocols in Mobile Ad Hoc Networks," *Proceedings of 36th IEEE International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO 2013),* Opatija, Croatia, pp. 441-445, May 2013.

Mohammad M. Shurman received the B.Sc. degree in Electrical and Computer Engineering from Jordan University of Science and Technology, Irbid, Jordan, M.Sc. and Ph.D. degrees Computer Engineering-Wireless Networks from University of Alabama–Huntsville (UAH) in 2000, 2003, and 2006, respectively. Presently he is an Associate Professor with the Network Engineering and Security Department, Jordan University of Science and Technology, Irbid, Jordan. His research interests include wireless Ad hoc networks, security and key management of wireless networks, wireless sensor networks, network coding, wireless communication and mobile networks, software defined networks, cognitive radio, WiMAX, 4G and 5G technology, edge computing, cloud computing and underwater acoustic WSN.

Mamoun F. Al-Mistarihi received the B.Sc. and M.Sc. degrees in Electrical Engineering from Jordan University of Science and Technology, Irbid, Jordan, M.E.E. and Ph.D. degrees in Electrical Engineering from University of Minnesota, Minneapolis, MN, USA, in 1992, 1996, 2005, and 2005, respectively. From 1994 to 2000, he was with the Royal Scientific Society, Amman, Jordan. Presently he is an Associate Professor with the Electrical Engineering Department, Jordan University of Science and Technology, Irbid, Jordan. His research interests include digital signal processing, image processing, digital signal processing for communications, wireless communications and mobile networks, performance evaluation of wireless communication systems over fading channels, security of wireless systems, WiMAX, Ad Hoc networks, and wireless sensor networks..

Khalid A. Darabkh received the PhD degree in Computer Engineering from the University of Alabama in Huntsville, USA, in 2007 with honors. He has joined the Computer Engineering Department at the University of Jordan as an Assistant Professor since 2007 and has been a Tenured Full Professor since 2016. He is engaged in research mainly on wireless sensor networks, queuing systems and networks, multimedia transmission, and steganography and watermarking. He authored and co-authored of at least ninety research articles and served as a reviewer in many scientific journals and international conferences. He serves on the Editorial Board of Telecommunication Systems, published by Springer, and Computer Applications in Engineering Education, published by John Wiley & Sons. Additionally, he serves as a TPC member of many reputable IEEE conferences such as GLOBECOM, LCN, VTC-Fall, PIMRC, ISWCS, and IAEAC. Dr. Darabkh is the recipeient of 2016 Ali Mango Reward for Distinguished Reseracher in Jordan. Moreover, he is a member of many professional and honorary societies, including Eta Kappa Nu, Tau Beta Pi, Phi Kappa Phi, and Sigma XI. He was selected for inclusion in the Who's Who Among Students in American Universities and Colleges and Marquis Who's Who in the World. As administrative experience at the University of Jordan, he served as Assistant Dean for Computer Affairs in the College of Engineering from Sept 2008 to Sept 2010. Additionally, he served as Acting Head of the Computer Engineering Department from June 2010 to Sept 2012.

AUTHORS' ADDRESSES
Assoc. Prof. Mohammad M. Shurman, Ph.D.
Network Engineering and Security Department,
Faculty of Computer and Information Technology,
Jordan University of Science and Technology,
P.O. Box 3030, Irbid 22110, Jordan Phone: + (962)
2-7201000, Fax: + (962) 2-7201077,
Email: alshurman@just.edu.jo
Assoc. Prof. Mamoun F. Al-Mistarihi, Ph.D.
Electrical Engineering Department,
Faculty of Engineering,
Jordan University of Science and Technology
P.O. Box 3030, Irbid 22110, Jordan
Phone: +(962) 2-7201000, Fax: +(962) 2-7095018
Email: mistarihi@just.edu.jo
Prof. Khalid Darabkh, Ph.D.
Computer Engineering Department,
Faculty of Engineering and Technology,
The University of Jordan,
Amman 11942, Jordan
Phone: + (962) 6-5355000, Fax: + (962) 6-5300813,
Email: k.darabkeh@ju.edu.jo.

Received: 2015-03-01
Accepted: 2017-04-18