Design a secure composite key-management scheme in Ad-Hoc Networks using Localization

Seyed-Mohsen Ghoreishi* and Morteza Analoui*,

*Iran University of Science and Technology, School of Computer Engineering , Tehran, Iran

Summary
A mobile Ad-Hoc network is a collection of wireless mobile nodes, dynamically forming a temporary network without the use of any existing network infrastructure or centralized Administration. Providing security support for mobile Ad-Hoc networks is hard to achieve due to the vulnerability of the links, the limited physical protection of the nodes, and also this fact that wireless networks are susceptible to attacks ranging from passive eavesdropping to active interfering and also mobile users demand "anywhere, anytime" services. In this paper, we present a new Composite Key Management scheme in Ad-Hoc networks. our approach works to decrease complexity of PKI in traditional public-key certificated-based systems and securely improves key-revocation and key-renewal approaches in ID-based systems. in this approach third trusted party (TTP), in which gives offline secret shared-key to each user, securely generates users private key(like PKG in ID-based systems) and also can play the role of directory service in Certificated-Authority like traditional certificated-based public key systems. More precisely if one user wants to obtain certificated public-key of another user, he can ask from TTP and get his desirable public-key.

Key words:
PKG , TTP, key-management, Identity-based, certificated-based.

1. Introduction

Mobile Ad-Hoc networks are different from existing networks by the fact that they don't rely on any infrastructure. This means the network do not have base station, access point, remote servers, etc. so mobile nodes that are within each other's radio range, communicate directly via wireless link, while those that are far, rely on other nodes to relay messages as routers. Node mobility in Ad-Hoc networks causes they have a dynamic network topology. More precisely, since nodes in the network are mobile with unrestricted movement, the configuration of the network topology can change very rapidly. Figure 1 shows an example: initially node A and D have a direct link between them. When D moves out of A's radio range, the link is broken. However, the network is still connected, because A can reach D through C, E and F.

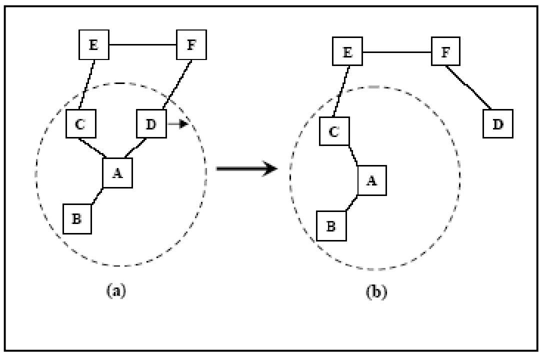


Fig. 1 Topology variation in Ad-Hoc networks

According to the mentioned topics, the most significant difference between mobile Ad-Hoc networks and traditional networks is that an Ad-Hoc network relies on wireless communication to keep the network connected. Also, this is important to say that the topology of the Ad-Hoc network is dynamically changing and the nodes of the Ad-Hoc network are often mobile. The lack of infrastructure, dynamic network topology and error-prone wireless connectivity, cause frequent link damages implying occasional connectivity. Therefore, protocols for mobile Ad-Hoc networks need to mitigate the unreliability of basic network services by taking on a fully distributed, self-organizing nature. The nature of these networks, makes them vulnerable to security attacks. So, a major challenge in the design of mobile Ad-Hoc networks is to protect their vulnerability from these attacks. Example of these attacks includes passive eavesdropping of the wireless channel, denial of service attacks by malicious nodes and attacks from compromised entities or stolen devices. Unlike wired networks where an adversary must gain physical access to the wired link or sneak through security holes at firewall and routers, wireless attacks may come from anywhere along all directions. So Ad-Hoc networks will not have a clear line of defense, and every node must be prepared for encounters with an adversary. For this reason, in some applications this requires a shift in models with respect to the traditional security solutions for

Manuscript received September 5, 2009
Manuscript revised September 20, 2009

wireless networks. Also, we still rely on traditional cryptographic primitives.

In this paper, we propose a new composite public-key management system that allows users to create new public-keys and obtain private-keys correspond to that public-keys and revoke them without help of trusted authority. so key-revocation and key-freshness approach is done based on the node's movement and location in the network. One of the differences between this scheme and ID-based schemes is that node's public-key is not the identifier (it just depends on the identifier of the node and is variable). Nevertheless, the most important difference between this proposed scheme and ID-based schemes is to privilege key-revocation and key-freshness to each node. This problem is significant in users activities (especially in this proposed scheme each node can have its own public-key and private-key based on its location which this can be one of the security requirements in military applications). Also this problem cause private-key revocation and private-key renewal perform voluntary by nodes which this is one of the significant advantages of proposed scheme.

On the other hand, like Identity-based cryptography systems, in this protocol each node has its unique identifier so that this identifier is the basis of establishment of the public key in that node. consequently, this proposed public-key system like ID-based systems, does not require a mechanism for certificate management and thus reduces the complexity of managing Public Key Infrastructure (PKI).

In proposed scheme, we present a way to produce public-keys based on their location history and identifier so that these public-keys are unique. Then, like ID-based key management systems, each node after movement and producing its public-key, applies to TTP and requests a private-key corresponds to its public-key. Like PKG[1] role in IBE[2] schemes, TTP is trusted private-key producer center that can produce private-keys correspond to public-keys. After requesting a new private-key, if TTP could authenticate that identity, should send a new private-key safely to that node via a channel (usually we assume that the channel is insecure). In this scheme we assume that each node can acquire its location and other nodes location in the network, so TTP can acquire nodes location using secure localization algorithms too. One of the assumptions in this proposed scheme is when a node enters the network, in addition to have its unique identifier (which TTP should guarantee this uniqueness), it should receive a shared-key with TTP from TTP in offline. When each node moves in the network, it can change its public-key and receive a private-key from TTP based on its new location. This causes private-key revocation and private-key renewal perform easily and this is one the most significant advantage of this proposed scheme (in addition to reducing the complexity of managing PKI). Nevertheless, one of the most important differences between this scheme and ID-based schemes is changing the public-key. In ID-based schemes, public-key is fixed (as each node public-key is its identifier). But in proposed scheme, public-key is not fixed and depends on the last public-key and new location (and also depends on node's identifier implicitly).

In conclusion, we review all the topics in this scheme. First, we decide to present key management idea in Ad-Hoc networks. After introducing localization, we present a model of proposed system and producing public-keys for each node. In fact last section is about our scheme (including three protocols). First, we present a secure private-key extraction protocol, then we present a protocol which two identities that know each other with primitive identifier can communicate with each other, and then we introduce a secure communication protocol, based on nodes location.

2-Key management in Ad-Hoc networks

Security problems are always significant discussions in mobile Ad-Hoc networks so that in many investigated areas within the mobile Ad-Hoc network field, there are still unsolved security problems which in spite of all presented solutions in this area, they are still open and need to work on. But, we can summarize their main feature so that: cryptographic techniques are often at the center of solving security problems in mobile Ad-Hoc networks and hence, they need key management [2],[4].

In this section, we discuss about some concerns in security problems of Ad-Hoc networks and in addition peer-to-peer key management position in security problems in Ad-Hoc networks. we present the main mechanism to provide security in all security problems. After introducing key management, we investigate it's functionality and efficiency. We introduce key management requirements in Ad-Hoc networks and discuss about peer-to-peer key management for them. The main idea of this section is that in addition to classification of protocol publications in this field, present "identity-based" and "certificated-based" key management schemes more precisely and completely as they are the basis of proposed protocol.

2-1-Security problems in Ad-Hoc networks

Many studies are done in mobile Ad-Hoc networks field in many aspects. This means, there are many investigations in Ad-Hoc networks security area in order to different goals. However, It is widely acknowledged that cryptographic mechanisms can provide some of the

[1] Private Key Generator
[2] Identity-Based Encryption

strongest techniques to ensure the availability, integrity and confidentiality for mobile Ad-Hoc networks security problems [1],[4]. Also, secure key management with a high-availability feature is at the center of providing network security via cryptographic mechanisms [3]. In fact, many cryptographic-based mechanisms that solve mobile Ad-Hoc networks security problems, have a direct reliance on an efficient and secure key management infrastructure. This means key management technique is an open approach area in the Ad-Hoc network security field [1].

2-2-Key management: Definition and Function

Before we explain key management in details, it is necessary to define two terminologies: keying relationship and keying material. A keying relationship is the state wherein network nodes share keying material for use in cryptographic mechanisms [1],[3]. Also, the best definite-on for keying material can be security parameters in the system. The keying material can include public/private key pairs, secret keys, initialization parameters and none-secret parameters supporting key management in various instances [1].

Now, key management can be defined as a set of techniq-uees and procedures supporting the establishment and maintenance of keying relationship between authorized parties [3].

In summary, key management integrates techniques and procedures to establish a service supporting [1],[3]:

(1) Initialization of system users within a network;
(2) Generation, distribution, and installation of keying material;
(3) Control over the use of keying material;
(4) Update, revocation and destruction of keying material;
(5) Storage, backup/recovery and archival of keying material;
(6) Bootstrapping and maintenance of trust in keying material.

In each key management service, authentication is the basis of secure communication, as without a robust authentication mechanism, the remaining security goals (confidentiality, data integrity and availability) are not achievable. One way to authenticate an entity is to investigate the secret of that entity. This means an entity should have a secret to prevent impersonation so that other entities can verify that entity using this secret. It is obvious that reach to secure key management six goals that mentioned above, depends on security of identity authentication. However, the attainment of these goals is not easy because user's authentication without using a trusted authority is very difficult [1](and in many cases impossible).

The basic function of key management schemes is to establish keying material. Keying establishment can be divided into key agreement and key transport [3].

Key agreement allows two or more users to derive shared keying material as a function of information contributed by each of the protocol participants, such that no party can predetermine the resulting value [3]. In key transport protocols, one party creates or otherwise obtains keying material, and securely transfers it to the other party or parties [3]. Both key agreement and key transport can be achieved using either symmetric or asymmetric techniques. Also, a hybrid key establishment scheme makes use of both symmetric and asymmetric techniques in an attempt to exploit the advantages of both techniques [3].

2-3- Requirements of key management schemes

From security perspective, in each key management scheme, we assume attributes for each key so that this key management can be assessed based on requirements to satisfy the attributes. In this subsection we are just going to introduce these requirements. We summarize these requirements to Confidentiality, Key authentication, Key confirmation, Key freshness, Perfect forward secrecy, Resistant to known key attacks, Forward secrecy, Backward secrecy, Key independence, Availability, Robustness, Survivability, Efficiency and Scalability[1].

2-4- Peer-to-peer key management for mobile Ad-Hoc networks

As this paper shows, we present a new composite peer-to-peer key management in Ad-Hoc networks. But it is necessary to know related works on peer-to-peer key management field and then verify and classify them. Generally, we can put peer-to-peer key management protocols for mobile Ad-Hoc networks in one of these subsets based on their approach [1]:

(1) Partially distributed certificate authority;
(2) Fully distributed certificate authority;
(3) Parallel key management;
(4) Certificate chaining-based key management;
(5) Cluster-based key management;
(6) Predevelopment-based key management;
(7) Mobility-based key management;
(8) Identity-based key management.

Most of these subsets use public-key cryptography in each distributing keys and authentication process and also to reach confidentiality, data integrity and availability [2],[3]. Since proposed scheme in this paper is a combination of Identity-based key management and certificate-based traditional schemes, we are going to introduce first three methods briefly. For study other methods you can refer to [5,6,7,8,9] .

2-4-1- Partially distributed certificate authority

This key management system include n server and is done based on public/private (K/k) key pair and certification authority [2]. With this assumption that CA is a vulnerable point in the network against attacks, this method tries to distribute authority between nodes and divides. The goal of this system is that public-key K is known to all nodes in the network and tries to divide private-key k into n shares $s_1, s_2, \ldots, s_n$ one share for each server. Also, each server i in addition to have a public/private key pair (K_i/k_i), knows all public-keys of all nodes (ordinary nodes or server nodes) in the network. This method is presented in Figure 2 [2].

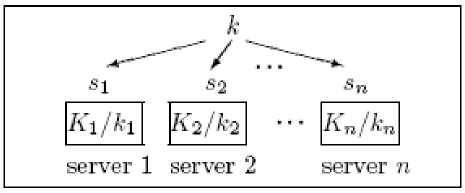


Fig. 2 Key management service *K/k* configuration

It is necessary to say that this method is done based on (n, t+1) configuration (n≥3t+1). In this configuration n is the number of servers in the network and (t+1) is number of servers that participate as a combiner (we present it by C) in digital signature generation. Also, this method is based on threshold signature that is presented in Figure 3 [2].

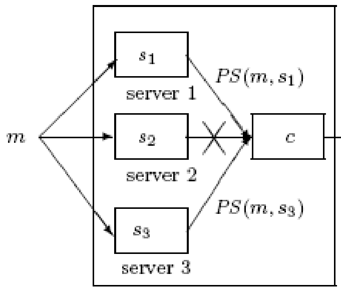


Fig. 3 Threshold signature *K/k* generation

As we see in Figure 3 Each server i after receiving a share s_i for each message, generates a partial signature $PS(m, s_i)$ and sends it to combiner. Then, the combiner is able to compute the signature. All nodes in the network investigate the accuracy of the signature using CA's public-key. As we see In Figure 3 , when servers 1 and 3 forward their signature shares to the combiner C, even though server 2 fails to generate its partial signature, C is able to generate signature (because it has private-key k).

Now, we investigate the main features of key management method which are key revocation and key renewal. In certificate-based systems, each node's public-key authority, depends on its certificate. This means the revocation and renewal of these keys are actually revocation and renewal of their certificate. In the partially distributed certificate systems, first a trusted mobile node generates a certificate revocation and then broadcasts its partially signed certificate revocation in the network. In the next step, (t+1) nodes must agree to certificate revocation. After all, nodes that received (t+1) partial certificate, will reconstruct the certificate revocation and update their certificate revocation list [1]. For certificate renewal, all the servers cooperate and are able to calculate new private-key shares using threshold cryptography methods without revealing them. After they update their list, servers discard previous shares and generate partial signature (or certificate) using new shares [2].

2-4-2- Fully distributed certificate authority

This method [10] is based on RSA cryptography system (refer to [11]) and system keys are presented as {SK; PK}. In this scheme SK is the system private/secret key and PK is the system public-key. In other words, SK is used to certificate authenticity for all nodes in the network and PK is used to verify authentic certificates. In this method, to establish a CA role in the network, SK is shared between all nodes in the network and each node v_i holds a secret share P_{v_i}. Any k of such share holders can collectively simulate the role of CA. Besides the system key pair, each node v_i holds a personal RSA key pair in the form of {SK_i , PK_i} and a certificate $cert_i$ in the form of <v_i, PK_i, T >. It is necessary to say that in this form, T is known as node i's public-key validity time. Also, a certificate is valid only if it is verified by system secret-key SK. In this method, each nodes certification is done using key SK (which is distributed in the network). Node v_i requests new certificate from any collection of k nodes, typically among its one hop neighbors. When other nodes receive node v_i's certificate request, check their records. If they verify node v_i is a legitimate node, they return a partial certificate using their shared SK. Otherwise, the request is rejected. By collecting k partial certificate, node v_i combines them together and generates a new certificate. Now, we investigate the main features of key management method which are key revocation and key renewal. In certificate-based systems, each node's public-key authority depends on its certificate. This means the revocation and renewal of these keys are actually revocation and renewal of their certificate. In this method records that node v_i

maintains consist of two parts: its direct monitoring data on neighbor nodes and certificate revocation list. If node v_i see misbehaving of its neighbor v_j, marks v_j in its CRL[1] and broadcasts a signed accusation against this convicted node in the network. Other nodes after receiving this message refer to their certificate revocation list and if the number of accusation reaches k, they do not connect to that node anymore. For certificate renewal, all nodes generate a random polynomial (so that its variables encrypt using SK key) and broadcast it in the network. Then node's new shares from SK are verified using k neighbor nodes and shares will update and return to each node.

Therefore, public-key revocation and renewal process in certificate-based systems (centralized, partially distributed or completely distributed) is a complex process and has high calculation cost. The reason is user's need to refer to the CA and receive certificated public-key of another user and the high cost of establishment and management Public Key Infrastructure (PKI [11]).

2-4-3- Review Identity-based key management systems

The most important concept of Identity-based cryptography (which first time presented in 1984 by Shamir in [12]), is to present a new cryptography algorithm so that it replaces user's public Identifier (like email or IP address) with their public-key. One of the most advantages of this kind of cryptography is that it doesn't need the public-key digital certificates and thus, it reduces system complexity and the cost for establishing and managing Public Key Infrastructure (PKI).

The operation of Identity-based cryptography systems

We start the basic concepts of Identity-based cryptography schemes with this scenario. The elements of this scenario are sender Alice, receiver Bob and a third trusted party which is called PKG (which it's most important role is generating private key for users). In this scenario Alice wants to encrypt her information and send it to Bob using Bob's Identifier (like email, IP address, digital image and…). Receiver Bob also needs to establish a secure connection to PKG and receive its public-Identifier's private-key to decrypt the cipher-text.

Identity-based systems basis

In this subsection, we are going to explain identity based cryptography systems basis using presented roles in previous scenario. Before presenting this model, first we represent security parameters of each elements of this system. Since Identity-based systems are established based on asymmetric cryptography, each element has a public/private key pair. So, in this system PKG has PK_{PKG} public key and sk_{PKG}secret key.

All nodes in the network have PKG's public-key so that they can send their message securely to PKG. but, PKG do not provide its private-key in the network and uses it to generate private keys for other nodes. Now, we investigate Identity-based systems operation in cryptography.

Investigate identity based cryptography

we can divide IBE[2] methods into four stages (the simple model is represented in Figure 4 [13]):

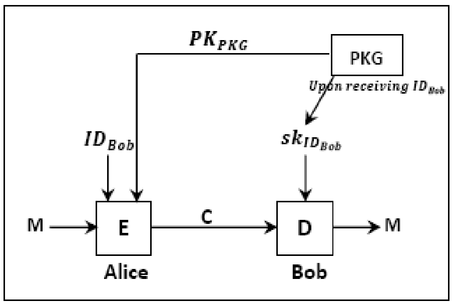


Fig. 4 Identity-Based Encryption

1_ **setup**: in this stage, PKG generates its public/private keys (which are shown as sk_{PKG} and PK_{PKG}). It is necessary to say that PKG provides its public-key to all nodes in the network (which can be a constant system parameter for a long period).

2_ **private-key extraction**: receiver Bob requests and after confirming by PKG, obtains $sk_{ID_{Bob}}$ private-key related to its unique identifier ID_{Bob} .

3_ **encryption**: using Bob's identifier ID_{Bob} and public-key PK_{PKG} related to PKG, sender Alice obtains encrypted message C from the message M and sends it to Bob.

4_ **decryption**: in this stage, Bob wants to obtain message M from the cipher text C. to accomplish this, Bob uses his private-key $sk_{ID_{Bob}}$.

Key revocation and renewal in Identity-based systems

In this section, we are going to investigate the most important basis of a key management method which is key revocation and key renewal [14]. In identity-based methods, public-key's certificate establishes based on expiration date. For example, suppose Bob's Identifier is "Bob@compony.com". So, Alice should uses

[1] Certification Revocation List

[2] Identity-Based Encryption

"Bob@company.com‖current-year" public-key to send the

encrypted message (in this public-key ‖ is concatenation

operation between two strings). After using this public-key structure, Bob have to refer to PKG once a year and receive new private-key (and thus, private-key expiration is done). Obviously, if we want to decrease private-key validity time in this system, we can use a smaller Grain size. For example, we can use current-day phrase instead of current-year so that users have to receive new private-keys every day. So, the main advantage of Identity-based methods against certificate based methods is less complexity of PKI. Also, Alice does not need to refer to CA to receive new certificate when validity time of Bob's private-key expires. But, the disadvantage of Identity-based methods is that users cannot revoke or renew their private-key every-time and all of them should receive their new private-key in a specific time. While, in some applications users have to renew their private-key for security purposes.

3-Review of localization

Localization of the wireless infrastructure-less networks have many potential applications in both military and civilian applications. These applications include [15]:

Robotic land-mine detection, Target tracking, Battle field surveillance, Wildfire detection, Environmental monitoring, and Traffic regulation.

Thus, many protocols have been devised to enable the location discovery process which in all these protocols the focal point of location discovery is based on a special node known as beacon (landmark) node. These beacon nodes know their location through manual configuration or GPS, so that each node can receive their location from them and obtain its location using other ways [15].

3-1- Localization systems process

The localization process is classified into two stages [15]. In the first stage, a node estimates its distance to other nodes in its vicinity using features of the received signals. In the second stage, node uses all the estimated distance to compute its actual location. The method used in the second stage can be classified into three main groups as follows:

- Triangulation
- Trilateration
- Multilateration

For more study refer to [15].

3-2- Localization techniques investigation

In this section, we are going to investigate two distribution localization techniques, so that each node is able to obtain its location. In these applications, each node is able to compute its location using one of these strategies:

DV-Hop method

This method [16] uses a classical Distance-Vector Algorithm to obtain distance of all nodes in the network to landmark nodes and exchange them with hop. In this method, first each landmark node computes its distance to other landmarks and then estimates an average value for a hop and broadcasts it to the network. With receiving this value, each node can compute its distance to landmark using Triangulation method in meter. Each node maintains a table $\{X_i, Y_i, h_i\}$ which (X_i, Y_i) is landmark node location and h_i is distance to that node (based on hop). And this table is update if nodes are mobile and with its neighbors. Also, with equation (1), each landmark computes average value for a hop as follows:

$$c_i = \frac{\sum \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}}{\sum h_i}, i \neq j, \text{ all landmarks } j \qquad (1)$$

The advantages of DV-Hop method are its simplicity and lack of spread of measurement error and the weakness is that it only works for isotropic networks. (for more information refer to [16]).

Sec-Hop method

In this section, we present a secure localization system based on hop by hop multilateration method [17]. Using the proposed secure hop (Sec-Hop) Algorithms, each node can securely determines the number of hops to landmark node and then using this information, estimates its distance to landmark node and then computes its location by solving multilateration equations. The Sec-Hop algorithm has four phases operations which are done as follows [17]:

Phase 1(commitment distribution phase):

In this phase, each landmark node (for example landmark node i) creates the links as follows:

$$link_1^i, link_2^i, \dots, link_n^i \qquad (2)$$

In this chain, n is the number of links which we can estimate diameter of the network using this. i is landmark node number that creates this link and $link_n^i$ is known as commitment value. At the start of localization process, each landmark will broadcast its n'th link (or will insert this value into nodes so that nodes cannot impersonate landmark).

Phase 2 (secure hop by hop propagation)

Each node maintains a table whose attributes are (*Xi, Yi, hi*). The localization algorithm begins with each landmark broadcast. Each landmark broadcasts its ID, hop count and the first element of the hash chain. More precisely, landmark begins localization process by sending the following packet:

$$\mathrm{landmark}_i \rightarrow \mathrm{Broadcast}\left(i, HC = 0, \mathrm{link}_1^i\right) \quad (3)$$

Then, each node after receiving hop count and link number and investigate the value of receiving hash chain, check authenticity of the hop. If they found to be authentic, the hop count is increase and the link becomes hash again. If not, the hop count and corresponding link are discarded.

Phase 3 (correction information propagation)

This phase is similar to that proposed in DV-Hop and is called correction phase. The scenario begins so that at the end of phase 2, each landmark gets all the localization packets from all other landmarks and knows the number of hops between itself and other landmarks. Thus, with the known landmark position, it can compute the average distance per hop. After these steps, each landmark broadcasts the average hop distance to the network.

Phase 4 (location estimation)

Once a node with unknown location has estimates its distance to a number of (more than three) landmarks, it can compute its location using multilateration. Each node uses the correction packet (sending packer in phase 3) to compute its location to landmarks it is connected to. The multilateration obtains corresponding node's coordinates using the landmark coordinates and the distance to them. It is necessary to say that the distance to each landmark is calculated as a product of the average hop distance of the nearest landmark and the number of hops to that landmark.

3-3- Investigation methods of the claimed position

In this section we are going to introduce a new method to investigate the location of the other nodes in the network. These schemes process are classified into two groups according to the location interpretation. The first group are schemes which investigates existence of a particular node in a specific Region, while the second group wants to obtain exact location of their particular node. Most of these schemes are based on claim-investigate so that with simulation of verifier and prover, they want the authenticity of location claim by prover node using verifier node. The scheme which we investigate here is in the group which wants to obtain exact location of their particular node. To study about schemes which investigate existence of a particular node in a specific region you can refer to [18].

Location verification system (LVS)

Location verification system is done based on Sec Hop algorithm (refer to previous subsection) and is designed to verify accuracy of location claims [17]. We present a scheme in this system that allows a centralized entity like a base station to be verifier (V) and verify location claims from prover node called P. it is necessary to say that this scheme is used in conjunction with the Sec Hop scheme. The location claims by a node can be in the form of:

({Verification tokens}, claimed location, node's ID)

Verifier V returns "success" if it can verify the authenticity of the location claim and if not returns "failure". The main problem is that prover P needs to prove that the hop count h it claims to have corresponding to landmark i is the actual hop count and the hop count has not been altered. Each node has a hash link for each landmark and can use it to prove the authenticity of the hop count claim. If "P" wants to prove that it has a hop count h corresponding to landmark i, it should send $link_h^i$ to the verifier. In order to verify authenticity of the location claims, there should be a mechanism so that nodes cannot modify the link. For instance, a node which has $link_h^i$ link that corresponds to hop count h, should not be able to compute $link_{h+1}^i$ or $link_{h-1}^i$. It is necessary to say that although an attacker isn't able to compute $link_{h-1}^i$ because hash function is one-way, but can compute $link_{h+1}^i$ and can pretend that node "P" is in another location.

In order to facilitate the verification process, we propose some modifications in this secure localization system with two goals [17]:

1_ even though it is more difficult to prevent the generation of a hop count, but this system propose a scheme to prevent a node from being able to report this higher hop count to the verifier V.

2_ prevent Back mail or Framing attack. These attacks in the investigations of authenticity of the location cause the legitimate location claim from a node to be rejected by the verifier.

In this system, each node maintains a reachable one-way hash function for every landmark. It is necessary to say that each chain is used to authenticate the source of link that a node broadcast. For notational convenience the elements of one-way hash function chain are presented as Alink (to be different from link in Sec-Hop algorithm). Thus, Alink chain which node j maintains corresponds to landmark i is as follows:

$$Alink^{i}_{1,j}, Alink^{i}_{2,j}, \ldots, Alink^{i}_{n,j} \quad (4)$$

It is also necessary to say that the n'th Alink values (or each node commitment) are known to all neighbor nodes and verifier V. when each node propagate hop count information to a specific landmark, it also attaches a Alink from its own hash chain (correspond to that landmark). Node j uses $Alink^{i}_{n-h,j}$ to authenticate hop-count h corresponding to landmark i.

Each neighbor node that receives a hop count and a Alink from its neighbor can authenticate the hop count using commitment information corresponding to each neighbor. The added hash chain in each node provides the following security features:

- Authentication: the one-way hash chain in each node allows a node to authenticate localization information from neighbor nodes.

-Non-Repudiation: in each node commitments corresponding to each landmark are revealed to the neighbors and verifier. This prevent repudiation of the node and also prevent framing attacks.

Based on this scheme, during location verification each prover P sends a verification token to the V which includes these values:

$$P \longrightarrow V: link^{i}_{h}, \ Alink^{i}_{n-h,N} \quad (5)$$

Where i is landmark, N is a neighbor of P and $P \neq N$.

In fact, when P sends $link^{i}_{h}$, it claims that its distance to landmark i is hop h and prove this claim by sending $Alink^{i}_{n-h,N}$ value.

If, in this scheme, just $link^{i}_{h}$ value was sent to the verifier, attacker M could claim the hop (h+Δ), because it is able to calculate forward links. But, in this scheme, verification token which contains a link of the form $Alink^{i}_{n-h,N}$ prevent this process.

More precisely, to fake a higher hop count h, attacker M is expected to sends $Alink^{i}_{n-(h+\Delta),N}$ which M≠N. but, since attacker M receives a link of the form $Alink^{i}_{n-h,N}$ to propagate hop h

It is infeasible to calculates $Alink^{i}_{n-(h+\Delta),N}$.

Thus, we can investigate authenticity of the location claim by the nodes in the network.

4- our composite key management system

Our proposed system includes a set of protocols in mobile Ad-Hoc networks which nodes have different identities in this network and they can enter the network or exit whenever they want. We assume all nodes are tamperproof in the network, which means that with compromising a node, security parameters of that node are not revealed (in other words, security policies are done in order to secure the communication channels and not to secure the nodes). From a security ability perspective, nodes can be divided into two parts in the network: TTP and normal nodes (peer entities). Since mobile Ad-Hoc networks cannot rely on any form of central administration or control, this is essential to avoid a single point of attack [2]. Thus, we try entities which provide services get into the form of partially distributed or fully distributed in this network. From a security perspective, distributing the functionality of network services to as many nodes as possible avoids a single point of attack. Generally, despite of all mentioned material, this proposed scheme is independent from TTP structure and the advantages and disadvantages of each structure. Thus, instead of using centralized TTP, we can use distributed TTP to generate private-key or obtain user's public-key.

4-1-System Elements

In this section, we introduce nodes, their security parameters and system attacker.

4-1-1-Network legal nodes

In this subsection, we introduce nodes. As mentioned earlier, nodes have different identities and can enter the network or exit whenever they want. Also, from security ability viewpoint, we can divide these nodes into two parts:

1_ normal nodes

2_ trusted nodes (which can generate private-key from identifiers that are called TTP)

Normal nodes: these nodes that have one identifier (we represent it with ID) receive private-key related to identifier ID, TTP public-key and also pre-shared key with TTP (that it can be used in symmetric cryptography), before they enter the network. Regarding to security ability, these nodes can do cryptography calculations and generate random values.

TTP: these nodes are trusted nodes that can generate private key for each identity. This means like PKG role in

ID-based schemes, TTP has a secret-key sk_T and a public-key PK_T and is able to generate private-key for each user, using sk_T and node's public-key. Also, like CA role in conventional "certificated-based" public key systems, we can use TTP to get node's certificated public-key.

4-1-2-Network attackers

In this section we introduce network attackers. We assume that attacker is an internal element of the network (means that attacker creates a forged identifier or compromises other node's identifier). An attacker is able to:

1_ eavesdrop transmission messages in the network

2_ impersonate a node and play the role of that node in communications

3_ be an active element. Means it can:

_change the content of the message

_ repeat the message

_ participate in protocols

But, an attacker is not able to:

1_ inverse the one-way hash function

2_ obtain plain-text from cipher-text without having private-key

3_ obtain any information from private-key using public-key and cipher text

4_ analysis the random values generator (which means can't guess random values generated by random generator).

4-2-Public key structure

In this section we investigate node's public-key which has significant role in this proposed scheme. First of all, we know that each node has an identifier before entering the network which can be derived from publicly available user's identity information such as email or IP address. In the beginning of entering the network, TTP should guarantee the uniqueness of the identifier and then, send private-key related to the identifier to that node. Note that node's identity is specified by this identifier. In this section, we first describe how to generate public-key based on these identifiers and then we introduce this public-key characteristics. The final purpose of this section is to generate a unique public-key in each time snapshot and use them like identifiers in ID-based key management schemes. After all, we describe the method to generate public-keys. But before it, we introduce some notations:

x_n^i: the nth location of node i (nth location which node i receives private-key from TTP by changing the public-key)

PK_n^i: the nth public-key of node i

ID^i: node i's identifier (private-key of this identifier is given to the node by TTP in offline)

Q_i : A replace value for ID^i as: Q_i=H(ID^i)

h: A collision-free-one-way hash function as: h: $\{0,1\}^* \rightarrow \{0,1\}^L$

H: A collision-free-one-way hash function as: H: $\{0,1\}^* \rightarrow \{0,1\}^{L-1}$

||: Concatenation operation between two strings

Now, we describe the way to generate these unique public-keys. To generate unique public-key for each node, we use its identifier and location history. The location history of a node is a sequence of locations which node was there before, and changed its public-key on those locations. we assume that h and H are collision-free one-way hash functions with output length equal to L and (L-1). also ID^i is node i's identifier. Now, assume that node i is in x_n^i location and wants to receive private-key (for nth time) in this location. Node i can calculate its public-key using equation (6).

$$\begin{cases} PK_n^i = h(PK_{n-1}^i \parallel x_n^i) & n > 1 \\ PK_1^i = Q_i & n = 1 \end{cases} \qquad (6)$$

Which in this equation, PK_{n-1}^i is node i's previous public-key (node i (n-1)th public-key) and || is concatenation.

Obviously, this equation (6) is included these features:

_ Nodes public-key is different in a specific time

_ Public-key of a particular node is different in different times

_ There is no way to represent that PK_n^i public key (n>=2) is related to the ID^i identifier (Unless we maintain nodes location sequence which is not beneficial)

_ Thus, we cannot link the next public-key of a node to its identifier, unless by using TTP

Now, we show that the public-key which is generated by equation (6) is unique. This theorem can prove the problem:

Theorem(1): the public-key which is generated by equation (6) is unique.

Proof:

We first assume that mth public-key of node i is equivalent to its nth public-key of node j (m>=n)

This means: $PK_n^j = PK_m^i$

Then:

$$PK_n^j = PK_m^i \Rightarrow h(PK_{m-1}^i \| x_m) = h(PK_{n-1}^j \| x_n)$$

Since, h function is one-way and collision-free, so:

$$\Rightarrow PK_{m-1}^i \| x_m = PK_{n-1}^j \| x_n \Rightarrow \begin{cases} PK_{m-1}^i = PK_{n-1}^j \\ x_m = x_n \end{cases}$$

Now, using inducting analysis we can show that:

$$PK_{m-n}^i = Q_j$$

Here, we have two conditions:

1_ m = n , so $ID^i = ID^j$

Since nodes identifier is unique, this condition does not happen.

2_ m>n , so $PK_{m-n}^i = Q_j$

Then:

$$m > n \Rightarrow m - n > 0 \Rightarrow PK_{m-n}^i = h(PK_{m-n-1}^i \| x_n)$$

$$\Rightarrow h(PK_{m-n-1}^i \| x_n) = Q_j$$

Which Q_j is node j's identifier that its length is equal to (L-1) as we assumed before. But PK_{m-n}^i is not node i's identifier and it is the output of the hash function h, Thus its length is equal to L and obviously, these two values are not equal. So, the public-key which is generated by equation (6) is unique.

■

4-3-Proposed secure key management scheme

In this section we propose our key management scheme which we claim that is secure against impersonation and presented attackers. This proposed scheme use three different protocols in order to reach its goals. The first protocol which is called protocol_1 is a communication protocol between user and TTP and is designed to private-key extraction (in this case TTP role is like PKG role in ID-based systems). It is necessary to say that each user communicate to TTP in two states: first case is for private-key extraction and the other case is when user wants to have certificated public-key from another peer entity that in this condition, TTP role is like CA role in conventional "certificated-based" public-key systems.

The second presented protocol which is called protocol_2 is a protocol that two peer entities want to setup a connection so that entity which requests a connection, wants to setup a connection with a specific identity. In other words, because entity which requests connection knows receivers identifier, so it uses a predetermined identity to establish a connection.

The third protocol which is called protocol_3 is designed to setup a connection between two peer entities. This protocol is executed whenever an entity wants to establish a connection with a node that only knows this node is in a specific location. In other words, sender node (entity which requests a connection) is not aware of receiver node's identity and it only wants to communicate with an entity which is in a specific location.

Now, it is necessary to specify when each node executes protocol_1 (communication protocol between user and TTP to extract private-key). When running protocol_2, if the situation of each connection parties change from the state which they run protocol_1 for the last time, they can either run protocol_2 or not, because in protocol_2 each communication peer entity starts its communication based on another party's identifier and not location. But when running protocol_3, if each communication party changes his location from the state which he executed protocol_1 for the last time, he must run protocol_1 again, because in protocol_3, each communication peer entity starts its communication based on another party location and not its identifier. Now, we investigate these three protocols in details.

4-3-1- protocol_1

In this section, we investigate proposed protocol related to private-key extraction which is an interface protocol between one peer entity node and TTP. But, before describing this protocol, first we introduce some notations (however, we are familiar with them from the past):

N_i: node i

$k_{i,T}$: node i's pre-shared key with TTP (TTP send this key to node in offline before it enter the network)

PK^{i}_{n} : nth public-key of node i (the length of this public key is assumed to L if n>1)

ID^{i}: node i's identifier. It is necessary to say that TTP sends private-key related to this identifier in offline to node i.

$k_{PK^{i}_{n}}$: private-key related to PK^{i}_{n} public-key.

x^{i}_{n} : the nth location of node i (nth location which node i receives private-key from TTP)

R, R', R'' : random values that are generated in the protocol (we assume these values are generated by a random generator)

PK_{T} :TTP's public-key. TTP gives this key to all nodes before they enter the network.

Now, after introducing all these notations, we investigate each node interaction with TTP (for private-key extraction). Each node (if it changes its location and need to receive a new private-key) after calculate its new public-key, should refer to the TTP to receive private-key related to that public-key. For this reason, nodes should introduce themselves to TTP to receive private-key and TTP should authenticate them. After this step, a secure session establishes between node and TTP, and TTP is able to give a new private-key to node safely. As mentioned earlier, node i receives a pre-shared key called $k_{i,T}$ before entering the network to communicate with TTP. Also, we assume that nodes are tamperproof which means if a node is compromised, its secret keys are not revealed. Since in this proposed scheme each node can change its public/private key regarding to change its location and for every new key it should communicate to TTP, so pre-shared key $k_{i,T}$ is used many times in the network. To avoid this problem, we can use new shared-key $k^{n}_{i,T}$ to secure node i communication with TTP which is derived from equation (7)

$$k^{n}_{i,T} = k_{i,T} \oplus k_{PK^{i}_{n-1}} \tag{7}$$

And thus, each communication new shared-key can be used (to receive private-key) just in public-key validity time and this can guarantee security of the pre-shared key. Figure 5 shows protocol_1. Before explaining this protocol in details, it is necessary to express these assumptions:

_ TTP can obtain node's location safely. In fact, TTP can obtain node's exact coordinate using a secure localization protocol.

_ TTP can generate random values. In other words, TTP is equipped to a random generator.

$1: N_i \rightarrow TTP: E_{k^{n}_{i,T}}(PK^{i}_{n-1} \| x_n) \| PK^{i}_{n-1}$

$2: TTP \rightarrow N_i : R$

$3: N_i \rightarrow TTP: E_{k^{n}_{i,T}}(R \| R')$

4: TTP **Authenticates** N_i **'s location**

$5: TTP \rightarrow N_i : E_{k^{n}_{i,T}}(R' \| k_{PK^{i}_{n}})$

Fig. 5 (Protocol_1)'s steps

Now, we explain proposed protocol step by step.

Protocol_1 explanation step by step:

1: in this step node i send its previous public-key (PK^{i}_{n-1}) and current location to TTP. These values encrypt with the last shared-key between node i and TTP and using one of the symmetric cryptographic systems so that attacker cannot change these values. It is necessary to say that TTP has a table that contains identifier, last public-key and pre-shared key of each node. so, after each request, TTP can extract pre-shared key $k^{n}_{i,T}$ from the table.

Then TTP can calculate $k^{n}_{i,T}$ using equation (7). After all, it decrypts receiving message using equation (8)

$$D_{k^{n}_{i,T}}\left(E_{k^{n}_{i,T}}(PK^{i}_{n-1} \| x^{i}_{n})\right) \tag{8}$$

It is clear that TTP should authenticate each node which requests a connection. node authentication has two steps:

_ Node identity authentication

_ Node location authentication (of course, this authentication can be done after one of the steps from 1 to 4)

2: TTP send random value R to node i as a challenge to authenticate node identity.

3: node i generates random value R' and sends $E_{k^{n}_{i,T}}(R \| R')$ encrypted message to TTP.

Then, TTP decrypts this cipher-text. This means that if TTP observe R in the decrypted message, it can authenticate node i successfully.

4: in this step TTP authenticate node i's location using one of secure localization algorithms. As we mentioned before, this kind of authentication can be done after one of the steps from 1 to 4.

5: after node i authentication, TTP encrypts new private-key and R' (which was sent by node i) with new shared-key $k^{n'}_{i,T}$ using one of symmetric cryptography algorithms and sends it again to node i.

Some notations about protocol_1

_ Since we assumed that TTP use a secure localization algorithm to certificate node's location, so an attacker cannot give wrong information about its location to TTP.

_ R and R' provide key freshness and avoid replay attack.

_ Security of the communication depends on $K^{n'}_{i,T}$. This itself depends on $k_{i,T}$ and $k_{PK^i_{n'-1}}$. since $k_{PK^i_{n'-1}}$ can change when node changes its location, $K^{n'}_{i,T}$ can get new values with node location changes. Although, if an attacker compromises a node, it can't obtain its secret-key, and because as we assumed nodes are tamperproof, this condition does not happen.

4-4- Communication protocols between two peer entities

In order to set up a secure connection between nodes, they should authenticate each other. Generally, nodes communication method is divided into two parts:

1_ communication via identity: in this case, two nodes that know each other's identifier want to communicate with each other (in other words, a node wants to communicate with a specific identity). To communicate using identity, each node should have a unique identity. In this proposed scheme we assume each node's identifier is also its identity. Thus, to authenticate a node, the identity of that node should be authenticated.

2_ communication via location: in this case, nodes communicate with each other via their location. Each node instead of communicate with a specific identity; communicate with an identity which is in a specific location.

Later, we present a secure protocol for both communications.

4-4-1- Communication protocol via identity: protocol_2

In protocol_2, two specific identities want to communicate with each other. These two nodes know each other from their identifier, but since node's public-key change during the time (because they are mobile); a node that wants to communicate with a specific identity should obtain its public-key. Because we assumed that TTP maintains last public-key and identifier of each node, so if a node with identifier ID^i wants to communicate with another node with identifier ID^j, with this assumption that n' and n are number of times that node i and j received a new private-key from TTP, the protocol is run as Figure 6 .

$$1: N_i \rightarrow TTP : E_{k^{n'}_{i,T}}(ID^i \| ID^j \| R) \| PK^i_{n'-1}$$

$$2: TTP \rightarrow N_i : E_{k^{n'}_{i,T}}(PK^j_n \| R)$$

$$3': N_i \rightarrow N_j : E_{ID^j}(m \| ID^i) \| PK^i_{n'}$$

$$3: N_i \rightarrow N_j : E_{PK^j_n}(ID^i \| R) \| PK^i_{n'}$$

$$4: N_j \rightarrow TTP : E_{k^n_{j,T}}(ID^j , ID^i , R')$$

$$5: TTP \rightarrow N_j : E_{k^n_{j,T}}(PK^i_{n'} , R')$$

$$6: N_j \rightarrow N_i : E_{PK^i_{n'}}(R \| R'' \| g^a) \| PK^j_{n}$$

$$7: N_i \rightarrow N_j : E_{PK^j_n}(R'' \| g^b) \| PK^i_{n'}$$

$$8: N_i \rightarrow N_j : E_{SK}(m)$$

Fig. 6 (Protocol_2)'s steps

These assumptions are considered about this protocol:

- TTP can generate random values.
- Each node can generate random values.
- Encrypted algorithms which are used in this protocol are secure.
- Each node's identifier is unique.
- Each node is known with its identifier.

Now, we explain protocol step by step.

Explain protocol_2 step by step

1: in the first step, node i requests node j's public-key from TTP. Therefore, node i encrypts its identifier, node j's identifier and a random value R, and sends it to TTP. This information should be encrypted so that an attacker cannot change it. Also, random value R is inserted in the encrypted package to avoid replay attack.

When TTP receives this cipher-text from node i, calculates new shared-key between node i and itself using equation (7) and then decrypts this cipher-text using calculated key and one of symmetric cryptography systems and extracts node j's identifier from the package.

2: in this step, TTP sends node j's last public-key to node i. This message is encrypted with random value R and calculated shared-key and sends to node i. node i decrypts the encrypted message that received from TTP. If node i observe random value R in the message, can sure that TTP sent this message and message has freshness.

3: we explain this step in two disjoint cases. If node i wants to have a connection-less communication with node j and only wants to send its information to node j, it should encrypt the package with node j's public key and send it. Since during this time it is possible that node j changes its location and obtains a new public-key, node i sends the public- key which is used to encrypt the message and cipher text to node j. Based on this protocol, each node stores P numbers of its last public-key so that if it receives a new message during its location changes, it can decrypt the message. Thus, P is a security parameter for a node, which means if P becomes great, validity time of node's last public-key also becomes great. However, It helps an attacker to have more time to obtain node j public key. On the other hand, if P becomes small, the possibility of a secure communication decreases (because it is possible that node j's public-key changes more than P times when node i receives node j's last public-key from TTP until it performs step 3 of this protocol). Thus, with regard to the environmental situation, make a correct choice of security parameter P is an important problem.

In second case, if node i wants to have connection-oriented communication with node j, these steps is done:

in this step, node i introduces itself to node j and then node i encrypts its identifier and random value R, and sends them with its public-key PK_n^j to node j. node j decrypts this cipher-text with $k_{PK_n^j}$.

4, 5: to obtain node i's public-key, like step 1 and 2, node j receives node i's last public-key from TTP.

6: in this step, node i and j want to generate a session-key to secure their communication. To generate session-key, we can use Diffie-Hellman protocol [11]. Node j encapsulates random value R which is received from node i and random value R'' which node j generates it, and g^a ("a" is a secret value that only node j knows) and encrypts this packet with PK_n^i to send it to node i. node i decrypts this cipher-text with $k_{PK_n^i}$ and if it observe the value R in the plain-text, it can authenticate node j.

7: node i encrypts random value R'' and g^b ("b" is a secret value that only node i knows) with PK_n^j and sends it to node j.

Node j decrypts this message with $k_{PK_n^j}$ and if it observes random value R'' it can authenticate node i.

8: both node i and j obtain session-key from equation (9) (based on Diffie-Hellman protocol):

$$SK_{ij} = g^{ab} \qquad (9)$$

8: node i encrypts its message with session-key SK_{ij} and sends it to node j.

Some points about protocol_2

- Values R , $R'$and R'' are used to avoid replay attack. These values also guarantee freshness of the generated keys.

- If security parameter P becomes small, it is possible that other node's public-key which they used it to encrypt their message, is revoked (because it is possible that destination node changes its location many times) and then destination node cannot decrypt received message.

- In this protocol nodes do not have to change their key when they change their location. Also, when nodes change their location, they can receive a new private-key from TTP. Since to generate session key, g^a and g^b values are encrypted, thus, man-in-the-middle attack does not happen.

- Each node authenticate the other node (by sending R and R''). Thus, authentication is mutual.

- Authentication need to communicate with TTP that it is a weakness of this protocol.

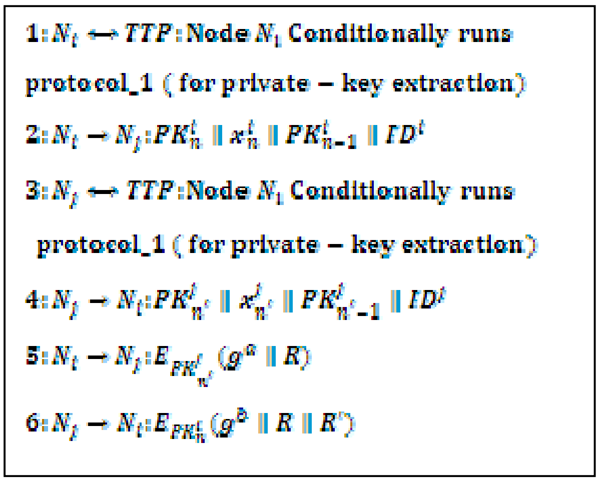
In next subsection we present a communication protocol via location.

Secure communication protocol via location: protocol_3

Now, we assume node i wants to communicate with node j which is in special location. So, it should first obtain node j's identifier (node j identifier should be authenticated for node i). nodes can use geographical routing or broadcast their messages for communication.

Now, in this section we present a protocol in which nodes communicate with each other via location and use geographic routing or broadcasting messages to send their package. It is necessary to say that in the protocol_3, we assume n' and n are the times that node j and i receive a new private-key from TTP. Our purpose to present this protocol is to secure the communication which means nodes authenticate each other using their location.

Protocol_3 process steps are illustrating in Figure 7 .



1: $N_i \longleftrightarrow TTP$: Node N_i Conditionally runs protocol_1 (for private – key extraction)

2: $N_i \rightarrow N_j : PK_n^i \parallel x_n^i \parallel PK_{n-1}^i \parallel ID^i$

3: $N_j \longleftrightarrow TTP$: Node N_i Conditionally runs protocol_1 (for private – key extraction)

4: $N_j \rightarrow N_i : PK_{n'}^j \parallel x_{n'}^j \parallel PK_{n'-1}^j \parallel ID^j$

5: $N_i \rightarrow N_j : E_{PK_{n'}^j}(g^a \parallel R)$

6: $N_j \rightarrow N_i : E_{PK_n^i}(g^b \parallel R \parallel R')$

Fig. 7 (Protocol_3)'s steps

Describing protocol_3 step by step:

1: execution of this step is conditional. If node i did not receive private-key for its current location, then communicates with TTP and executes protocol_1 to receive private-key related to its new public-key so that node i can obtain its private-key related to its new location.

2: node i wants to communicate with a node which is in the special location (for example node j is in this location). Thus, node i sends its new public-key, its previous public-key and its location to node j (in this step, node i do not know node j's identifier yet).

After receiving these three values, node j authenticates node i's location using a secure localization algorithm as we explain in subsection 3-3). Then, investigate the accuracy of its public-key using equation (6) that is:

$$PK_n^i = h(PK_{n-1}^i \parallel x_n^i)$$

3: execution of this step is conditional. If node j did not receive private-key for its current location, then communicates with TTP and executes protocol_1 to receive private-key related to its new public-key so that node j can obtain its private-key related to its new location.

4: node j sends its new public-key, its previous public-key and its location to node i. After receiving these three values, node i authenticates node j's location using a secure localization algorithm. Then, investigates the accuracy of its public-key using this equation (6) that is:

$$PK_{n'}^j = h(PK_{n'-1}^j \parallel x_{n'}^j)$$

5: in this step, node i and j want to generate a session-key to secure their communication. To generate session-key, we can use Diffie-Hellman protocol. Node i encapsulates random value R and g^a ("a" is a secret value that only node i knows) and encrypts this packet with $PK_{n'}^j$ to send it to node j.

7: node j encrypts random values R and g^b ("b" is a secret value that only node j knows) with PK_n^i and sends it to node i.

Node i decrypts this message with $k_{PK_n^i}$ and if it observe random value R it can authenticate node i.

8: both node i and j obtain session-key from equation (9) (based on Diffie-Hellman protocol):

$$SK_{ij} = g^{ab}$$

Some points about presented protocol

- Since the main goal of this protocol is establish a secure communication via location and since each node is able to obtain other node's location, an attacker cannot give a fake location to other party. On the other hand, since all public-keys of each node are unique according to theorem(1), an attacker cannot lies its public-keys (the last public-key or previous public-key) in the steps 2 and 4. If an attacker lies its public-keys, another party will discover this during investigation of equation (6) and do not authenticate attacker. Thus, an attacker is not able to have a connection with other nodes by giving them wrong information.

- If attacker wants to impersonate identity of the corresponding node at the other time by receiving information of both connection parties in steps 2 and 4, R and R' values do not allow it to do this. In addition, these random values are able to prevent replay attack and can guarantee freshness of session-key.

- It may be assumed that if an attacker after above messages, at the other time can obtain the private-key of one of the parties, insert itself to that party location and instead of information in steps 2 and 4(the last public-key or previous public-key), send fake information (compromised node's public-keys) to another party. But, it is necessary to say that in this protocol, connection parties should present their correct location, because the main goal of this protocol is to establish a secure connection between nodes which are in a specific location.

The other advantage of this protocol is that nodes do not need TTP to authenticate themselves. Beside protocol optimization, this can prevent TTP to be bottleneck and single point failure problem (as mentioned earlier, this is significant in wireless infrastructure-less networks).

5- Some recommendations for future investigations

Although new proposed schemes of this paper could solved many expected problems, but we can recommend this open problems for future investigations:

- Security of the proposed schemes depends on precision and security of the localization methods in Ad-Hoc networks. Future investigations will find a secure method to localize nodes in the network and specially use them in the protocols of each of the proposed methods.

- The nature of the Ad-Hoc networks require that trusted nodes design based on one of the distribution methods (while proposed scheme in this paper is done independent of TTP's reliable structures). Future investigations can find a secure and comprehensive method to distribute TTP and how nodes communicate to them.

- Apply to a trusted identity (to receive node's public key) is a weakness in key management methods, While communication parties in proposed scheme have to find a solution for this weakness to receive another node's public-key.

[1] Johann, V. M.,Dawoud, D., and Mcdonald, S. 2007. "A Survey on Peer-to-Peer Key Management for Mobile Ad Hoc Networks". ACM Computing Surveys, Vol. 39, No. 1
[2] Zhou, L. and Haas, Z. J. 1999. "Securing ad hoc networks". IEEE Netw. (Special Issue on Network Security) 13, 6, 24–30.
[3] Menezes, A., Vanoorschot, P., AND Vanstone, S. 1996. "Handbook in Applied Cryptography". CRC Press, Boca Raton, FL.
[4] Hubaux, J.P., Buttyan, L., and Capkun, S. 2001. "The quest for security in mobile ad hoc networks". In Proceedings of MobiHoc'01.
[5] Capkun, S., Buttyan, L., and Hubaux, J. 2003. "Self-organized public-key management for mobile ad hoc networks". IEEE Trans. Mobile Comput. 2, 1, 52–64.
[6] Ngai, E. C. H.,Lyu, M. R., and Chin, R. T. 2004. "An authentication service against dishonest users in mobile ad hoc networks". In Proceedings of the IEEE Aerospace Conference.
[7] Eschenauer, L. and Gligor, V. D. 2002. "A key-management scheme for distributed sensor networks". In Proceedings of the 9th ACM Conference on Computer and Communication Security (CCS'02).
[8] Capkun, S., Hubaux, J., and Buttyan, L. 2006. "Mobility helps peer-to-peer security". IEEE Trans. Mobile Comput. 5, 1, 43–51.
[9] Yi, S. and Kravets, R. 2004. "Composite key management for ad hoc networks". In Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'04).
[10] Luo, H., Zerfos, P., Kong, J., Lu, S., and Zhang, L. 2002. "Self-securing ad hoc wireless networks". In Proceedings of the Seventh International Symposium on Computers and Communications (ISCC'02).
[11] Buchmann, J. A. "Introduction to cryptography", Springer-Verlag , 2001. ISBN: 0-387-95034-6.
[12] Shamir, A. "Identity-based cryptosystems and signature schemes," Advances in Cryptology - CRYPTO '84, LNCS 0196, pp. 47-53, Springer-Verlag, 1985.
[13] Baek, J. , Newmarch, J. , Safavi-Naini, R. , Susilo, W. "A Survey of Identity-Based Cryptography" 2004.
[14] Boneh, D. and Franklin, M. "Identity-based encryption from the Weil pairing," Advances in Cryptology - CRYPTO '01, LNCS 2139, pp. 213-229, Springer-Verlag, 2001.
[15] Srinivasan, A. and Wu, J., "A survey on secure localization in wireless sensor networks". In: Furht, B. (Ed.), Wireless and Mobile Communications, CRC Press/Taylor and Francis Group, Boca Raton/London.2007
[16] D. Niculescu, B. Nath, "Ad Hoc Positioning System(APS)", IEEE GlobeCom 2001. pp. 2926-2931.
[17] Lee,Y.H., Phadke,V. , Lee,J.W. and Deshmukh, A. ," Secure Localization and Location Verification in Sensor Networks" , Springer-Verlag. 2005.
[18] N. Sastry, U. Shankar, D. Wagner, "Secure Verification of Location Claims", Proceedings of the 2003 ACM workshop on Wireless Security, 2003. pp. 1-10.