Security and pseudo-anonymity with a cluster-based approach for MANET

Abderrezak Rachedi, Abderrahim Benslimane

Security and Pseudo-Anonymity with a Cluster-based approach for MANET*

Abderrezak Rachedi and Abderrahim Benslimane
LIA/CERI, University of Avignon, Agroparc
BP 1228, 84911 Avignon, France
Email: {abderrezak.rachedi, abderrahim.benslimane}@univ-avignon.fr

***Abstract*—In this paper, we propose an anonymous protocol to secure nodes which have important roles in the network. We focus in the clustering approach to secure the Mobile Ad Hoc Networks (MANETs). In each cluster, a confident node is selected to ensure the Certification Authority (CA) roles; however, the cluster security depends in the security of the CA node. Therefore, we present an Anonymous Dynamic Demilitarized Zone (ADDMZ) to protect the CA node identity and to avoid the single point of failure in the cluster. ADDMZ is formed by a set of confident nodes which have a high trust level between them and their goal is to filter the communication between the cluster member node and the CA node. Moreover, we draw one's inspiration from military defence mechanisms such as: camouflage and identity change mechanisms. We present protocol to realize these mechanisms by using the identity based cryptographic from bilinear maps. The security analysis is proposed to discuss the proposed protocols.**

I. Introduction

In recent years, researcher's attention is turning towards the security in Mobile Ad Hoc Network (MANETs) mainly due to their characteristics and their application fields. In MANETs, the privacy issue becomes more crucial and important for mobile nodes because of MANETs' characteristics particularly the open network in which the radio is shared by all nodes. The node's identity is exposed to the channel eavesdropping. Without trying to secure the identity of the nodes which have an important role in the network like security services, the vulnerability can be exploited by attackers to create the Denial of Services (DoS) attacks. The security in MANETs consists to ensure the authentication, authorization, confidentiality, data integrity and dynamic trust model evolution. Many solutions were proposed in literature to secure MANETs, however, few of them take into account the real MANETs' characteristics such as: mobility, open network, energy limitation, etc.

In our previous works, we proposed a new architecture based on a trust model and a secure clustering algorithm in order to create a dynamic key management system adapted to MANETs' characteristics [5]. The main idea consist in distributing the Certification Authority (CA) in each cluster and ensuring the security of these CA nodes by a new mechanism called DDMZ (Dynamic Demilitarized Zone). However, the identity of the CA node and the set of nodes which form the DDMZ are not protected. That means that any unknown node (not confident) can eavesdrops the communication and find the identity of these important nodes. This information can be useful for the attacker to plan attacks against the CA node in order to disturb the cluster operation. Therefore, if the CA node is compromised, that means that the security of the cluster is calling into question.

In this paper, we focus on the secure distributed architecture to ensure the security and we introduce another security's parameter, called anonymity. The goal is to ensure the sensitive security services such as: CA and Registration Authority (RA) nodes without disclosing the nodes' identity. We use the simple designed verifier signature (SDVS) [1] to generate the dynamic pair-keys and we use the pseudonym for a confident identity instead of their real identity in order to mask the real identity and to protect them against potential attacks. We improve the DDMZ concept by introducing the anonymity concept to design the Anonymous Dynamic Demilitarized Zone (ADDMZ). The idea consists in making the CA identity node hidden for unknown nodes and the nodes with a low trust level. To reach this goal, the identity change and camouflage mechanisms are presented. Furthermore, we propose a new protocol to establish the ADDMZ and the communication intra and extra ADDMZ. Moreover, the secure protocol of the communication between clusters is investigated and presented.

The rest of the paper is organized as follows. The section II is devoted to the summarization of the distributed architecture and the DDMZ concept. Furthermore, we present some existing works with the anonymity concept. Moreover, we summarize the simple designed verifier signature (SDVS). In section III, we present the proposed protocols named ICCP which is based on identity change and camouflage mechanisms. In section IV, we investigate the security analysis of the ICCP and we present its performance. Finally, the section V concludes the paper with future works.

II. Related works

In our previous work, we proposed a distributed hierarchical architecture which divided the network into clusters to secure the network [5]. In this architecture, we have defined a trust model to assign different roles such as the certification authority (CA) and the registration authority (RA) roles in each cluster. We also proposed the secure distributed clustering algorithm (SDCA) to divide the network into a certain number of clusters. Furthermore, we introduced the new concept of Dynamic Demilitarized Zone (DDMZ) to secure the CA node

* This work is supported by the ANR "Agence Nationale de la Recherche - France" within the project framework ARA/CLADIS.

in each cluster. A DDMZ is an intermediate zone between unknown nodes and the CA node in each cluster. It is formed by a set of confident nodes, where at least one has the RA role. The RA role consists in filtering the communication between the CA and other nodes in order to protect the CA node against any potential attack. However, this architecture does not ensure the anonymous communication between the confident nodes.

There are several approaches of anonymous communication: Zhang et al. [3] proposed the anonymous communication protocol called MASK. In MASK protocol, the authors assume that the system's administrator generates a large set of pseudo identities (IDs) for each node. However, each node has a certain pseudo ID set. The set's size should be large enough in order to avoid the vulnerability to find the pseudonym by attackers. The real problem is that the pseudo IDs work like real identities and that the attackers are able to identify each node. In addition, the pseudo identity maintenance and management are costly. Rahman et al., [4] proposed RIOMO protocol to improve the MASK weakness by reducing pseudo IDs' maintenance costs which nodes take only one pseudo ID from system's administrator and generate their own pseudo IDs for an anonymous communication. Another work dealing with anonymity and privacy proposed a secure dynamic distributed in [6] which is based on "the onion routing protocol" [7]. This protocol ensures the route anonymity but not a strong location privacy. Kong et al. [2] proposed an Anonymous On-Demand Routing (ANODR) based on topology and broadcast to improve the receiver's anonymity. ANODR is an on-demand protocol based on trapdoor information in the broadcast. Trapdoor information is a security concept that has been widely used in encryption and authentication schemes.

TABLE I
TABLE OF VARIABLES AND NOTATION

ID_i	Real identity of the node i
ID_{Pi}	Pseudonyme of the node's identity
$< K_i^+, K_i^- >$	Public and Private Keys of the node i
$< rK_j^+, rK_j^- >$	Real public and Private Keys of the node j
SK_i^j	Session key shared between nodes i and j
K_g^i	Key of the group i
$Ea_K(M)$	Cryptogram of M encrypted by public cryptography algorithm (RSA, ElGamal) and using a key K
$SIN_{K_i^-}(M)$	Signature of the message M generated by the node i
$Es_K(M)$	Cryptogram of M encrypted by symmetric cryptography algorithm (AES, 3DES) and using a key K

III. ICCP: IDENTITY CHANGE AND CAMOUFLAGE PROTOCOLS

A. Preliminary

The military defence mechanisms such as: camouflage and identity change mechanisms are inspired by the animals' defence mechanisms. Many animals use the camouflage mechanism to avoid the predator's attacks, we can quote as example the iguana's camouflage when it is perched in trees. The chameleon is an example of the identity change. Therefore, for our solution we adopt the identity change mechanism for confident nodes and the camouflage mechanism for the CA and RA nodes in order to secure the DDMZ. Hence, the goal is to mask the identity of all confident nodes particularly the CA and RA nodes and protect their activities against eavesdropping and traffic analysis attacks.

We consider that each confident node has both a real identity and a pseudonym, and it also has two pairs of keys: real keys (private/public) and dynamic pair-keys generated according to the SDVS scheme [1] for each a cluster's configuration or formation. The set of the notations used in the paper are listed in table I.

B. Identity change of confident nodes

In order to realize a confident chameleon node with identity masking, we used the bilinear maps and the mechanism developed in [4]. We suppose that each confident node has secret point SP_i which depends on the real identity of the confident node. However, SP_i is generated as follows: first, the system determines two groups $\mathbb{G}_1$ (additive group) and $\mathbb{G}_2$ (multiplicative group) of the same prime order q. Secondly, it determines a bilinear map $f : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ and two collision resistant cryptographic hash functions H_1 and H_2 defined as follows: $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$ and $H_2 : \{0,1\}^* \rightarrow \{0,1\}^\ell$ where ℓ-bit fixed length output. Thirdly, it generates the secret $S_c \in \mathbb{Z}_q$ of confident community nodes, but neither the confident nodes nor other nodes know the secret S_c. The confident nodes securely receive their secret point before the nodes' deployment. In addition, the system parameters $\{\mathbb{G}_1, \mathbb{G}_2, f, H_1, H_2\}$ are known to the confident nodes. Therefore, each confident node has a secret point $SP_i = S_c.H_1(ID_i)$ where ID_i is its real identity. When a confident node (ID_i) wants to change its identity for any security reason, it generates a new pseudo identity called ID_{Pi} and its pseudo secret point (SP_{Pi}) as follows:

$$\begin{cases} ID_{Pi} = r_i.H_1(ID_i) \\ SP_{Pi} = r_i.SP_i = r_i.S_c.H_1(ID_i) = S_c.ID_{Pi} \end{cases}$$

where r_i is a random number generated by node ID_i.

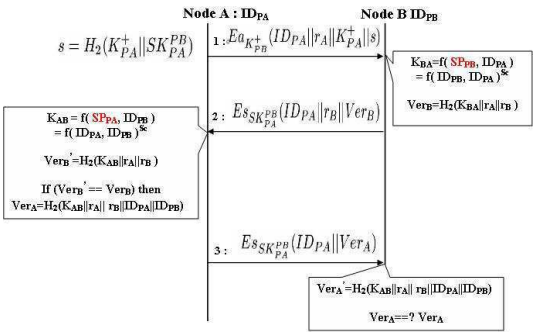

Fig. 1. Anonymous confident nodes authentication

When two nodes A and B want to check if each other is confident node, the process is shown in figure 1. Let's suppose that the new pseudo identity of nodes A and B with their associate secret point are called as: $\{ID_{PA}, SP_{PA}\}$ and

$\{ID_{PB}, SP_{PB}\}$. Node A sends its new identity with the random value:$\langle ID_{PA}, r_A, K^+_{PA}, s\rangle$ to node B, where $s = H_2(K^+_{PA}||SK^{PB}_{PA})$ and $SK^{PB}_{PA} = (K^+_{PB})^{K^-_{PA}}$. Once node B has received this information and after deciphering by its private key, it calculates the session key $SK^{PB}_{PA} = (K^+_{PA})^{K^-_{PB}}$ for the next encryption packet and it checks the integrity of PA's public key and the session key by using the parameter "s". If the checking procedure is going well, it computes a key K_{BA} according to the bilinear properties, $K_{BA} = f(SP_{PB}, ID_{PA}) = f(ID_{PB}, ID_{PA})^{S_c}$ and it generates the random r_B, and then it generates $Ver_B = H_2(K_{BA}||r_A||r_B)$. Then, a node B sends this information $\langle ID_{PA}, r_B, Ver_B\rangle$ to node A encrypted by the session key. When node A receives this information and after deciphering by the session key, it computes the key K_{AB} like node B did it: $K_{AB} = f(SP_A, ID_{PB}) = f(ID_{PA}, ID_{PB})^{S_c} = K_{BA}$, then node A computes $Ver'_B = H_2(K_{AB}||r_A||r_B)$. If $Ver_B == Ver'_B$ then it calculates $Ver_A = H_2(K_{AB}||r_A||r_B||ID_{PA}||ID_{PB})$ and it sends encrypted information $\langle ID_{PA}, Ver_A\rangle$ by the session key to node B. Once node B has received this information and deciphering operation going well, computes it the $Ver'_A = H_2(K_{AB}||r_A||r_B||ID_{PA}||ID_{PB})$ and compare it with Ver_A if it matches, then node B deduces that node A is a confident node.

According to the trust model developed in [5], each node has a trust metric (Tm) which defines the node's trust level. Only a confident node can have the highest trust level ($Tm = 1$). However, when the unknown node ID_k (which does not belong to a set of confident nodes) joins the cluster, the cluster's CA gives it a low trust level and increases it when the monitoring process gives its positive evaluation. Once ID_k has reached the conditions to become confident, we need to answer to this question: how a new confident node obtain its secret point SP? In our model, only the CA node can generate the confident status by changing the identity of the unknown node changes from ID_k to $ID_{Pk} = H_1(ID_k).ID_{Pi}$ where $CA_i = ID_{Pi}$ is the CA node's pseudo identity. Moreover, the CA node generates the secret point $SP_{ID_{Pk}}$ to ID_{Pk} as follows:

$$\begin{aligned} &SP_{ID_{Pk}} = r_i.SP_{ID_i}.H_1(ID_k) \\ &= r_i.S_c.H_1(ID_i).H_1(ID_k) = S_c.ID_{Pi}.H_1(ID_k) \\ &= S_c.ID_{Pk} \end{aligned}$$

Once a node ID_k has received its new pseudonym identity ID_{Pk} and its corresponding Secure Point $SP_{ID_{Pk}}$, it can authenticate and be authenticated by any confident node.

C. Security of CA and RA nodes in ADDMZ

In order to secure the CA and RA nodes, we adopt the camouflage mechanism. Once the CA node is elected, it changes its identity by generating the new pseudonym identity named ($CA_i = ID_{Pi}$) according to the identity change mechanism illustrated above. $CA_i = r_i.H_1(ID_i)$, where r_i is a random variable generated by ID_i. Moreover, CA_i uses the SDVS scheme [1] to generate dynamically the cluster's (i) private and public keys at each cluster's formation and configuration. Then, CA_i establishes the secret session key ($SK^{RA_j}_{CA_i}$) with each RA_j node according to the SDVS scheme. The CA_i node establishes the session key as follows:

- It computes $K^-_{CA_i} = x_i.H_1(CA_i)$ where x_i is a random number in $\mathbb{Z}_p$ and then, it computes the public key of the cluster i ($K^+_{CA_i}$) and the value "s" to ensure the integrity of K^+_{CA} and $SK^{RA_j}_{CA_i}$ as follows: $K^+_{CA_i} = g^{K^-_{CA_i}}$ and $s = H_2(K^+_{CA}||SK^{RA_j}_{CA_i})$ where $SK^{RA_j}_{CA_i} = (K^+_{RA_j})^{K^-_{CA_i}}$.
- It forms the message $M = [\#Id_p||CA_i||RA_j||K^+_{CA_i}||s]$, where $\#Id_p$ is a unique identifier for each packet in the entire network which is randomly selected. Then, it encrypts the message M by the public key of the RA_j node as follows: $C = Ea_{K^+_{RA_j}}(M)$.
- The CA_i sends the packet P (P=$\langle C\rangle$) to RA_j.

Once the node at one hop from the CA_i has received the packet P, it tries to decrypt the cryptogram "C" by using its private key $K^-_{N_j}$, if the deciphering operation is successful, then the receiver node deduces that it is the destination and it checks the integrity of the packet P, otherwise it is not the destination and it drops the packet.

The CA_i node repeats the same operation with each RA node, and then, it shares the session key ($SK^{RA_j}_{CA_i}$) with all RA nodes (RA_j). Furthermore, the CA_i uses the shared session key with the RA nodes to generate the group key (K^i_g) of the ADDMZ. Therefore, if the size of the ADDMZ is k, then K^i_g is generated as follows:

$$K^i_g = H_2(SK^{RA_1}_{CA_i}||SK^{RA_2}_{CA_i}||...||SK^{RA_k}_{CA_i})$$

The set of RA nodes which form the DDMZ of the cluster i takes the pseudonym "$DDMZ_i$". We use the same principle of broadcast used by ANODR [2] to secure CA and RA nodes' receiver anonymity. For instance, each packet transmitted to the CA_i or any RA node in the cluster (i), the destination address should be quoted as $DDMZ_i$. No node, even located at one hop from the $DDMZ_i$ is able to identify the RA node's pseudonyme and to know the real identity of the RA nodes which form the DDMZ. Therefore, in order to secure the RA nodes identity, the public and the private keys (K^-_{ddmz}, K^+_{ddmz}) of the Anonymous DDMZ need to be generated. These keys are based on the secret shared group key K^i_g between a set of RA and CA_i nodes. The private key of the ADDMZ is known only by RA and CA nodes and is calculated as follows: $K^-_{ddmz} = H_1(K^i_g)$. However, the public key of the ADDMZ is calculated as follows: $K^+_{ddmz} = g^{K^-_{ddmz}}$.

D. Intra-cluster communication

In the intra-cluster communication, we distinguish two types of communications: intra-ADDMZ communication and extra-ADDMZ communication.

1) intra-ADDMZ communication: The communication intra-ADDMZ does not exceed one hop from the CA. Only the CA and RA nodes are able to decrypt the information in a

packet broadcast in this zone. However, the CA can privately communicate with each RA node.

The communication between the CA and RA nodes is encrypted by the K_g^i in the case of the broadcast packet for ADDMZ. However, in the case of the private communication between the CA_i and RA_j, the both shared session keys ($SK_{CA_i}^{RA_j}$) and the group key K_g^i are used. Once, RA_j wants to privately send the message m to the CA_i node, it forms the packet as follows: $P = \langle Q \rangle$, where $Q = Es_{K_g^i}(\#Id_p, CA_i, RA_j, C)$ and C is the cryptogram encrypted by the session key $SK_{CA_i}^{RA_j}$ ($Q = Es_{SK_{CA_i}^{RA_j}}(m)$). Only RA and CA nodes are able to deciphering the packet by using the group key and checking the destination and source address which can be RA or CA nodes.

2) Extra-ADDMZ communication: In order to mask the identity of RA nodes which generate the pair of keys (K_{ddmz}^-, K_{ddmz}^+) based on the group key K_g^i. The public key of the ADDMZ (K_{ddmz}^+) and the public key of the CA role ($K_{CA_i}^+$) are broadcast by the CA_i node in the HELLO cluster beacon via RA nodes to all the nodes in the cluster (i). The HELLO cluster beacon is periodically generated by the CA node in order to maintain the cluster and to distribute the cluster's identity information which comes down to the ADDMZ public key and to the public key of the CA role. The hello cluster packet named P_{Hello} is formed as follows: $P_{Hello} = [\#Id_p, hop, DDMZ_i, K_{CA_i}^+, K_{ddmz}^+, S]$, where the $hop = hop_{max} - 1$ which is the cluster's size and $S = SIN_{K_{CA}^-}(\#Id_p || K_{CA_i}^+ || K_{ddmz}^+)$.

In order to realize the camouflage of the pseudo identity of the CA and of RA nodes and according to the previous section, we use the anonymous broadcast address. In the case of IEEE 802.11, a predefined multicast address can be used as source or destination MAC address [2].

When node N_i receives P_{Hello}, it checks if ($hop - 1 \geq 0$) then it continues the checking operation ; otherwise it drops the packet. Then, it checks the packet identifier ($\#Id_p$), in other words, it checks if the packet has already been received or not. If the packet has not been received it continues to check the integrity and the authentication of P_{Hello} by using the public key of the CA role ; otherwise the packet is rejected. In the case of the whole checking procedure is going well and the receiver node has its certificate from the CA_i, it forwards the packet to its neighbours after the hop parameter has been updated. Moreover, it adds its certificate in the packet and it saves the $\#Id_p$ and the reception time of the packet T_{recv}. The $\#Id_p$, $DDMZ_i$ and T_{recv} are important to route the packet to the $ADDMZ_i$ and to form the routing table based on the virtual circuit identifier (VCI) concept [2]. This concept permits to route the packet according to the virtual identity. The format of the forwarded packet by the node N_i is as follows: $\langle \#Id_p, hop-1, N_i, Cert_{CA_i}(N_i), K_{CA_i}^+, K_{ddmz}^+, S \rangle$ where the certificate format is defined as follows:

$$Cert_{CA_i}(N_i) = SIN_{K_{CA_i}^-}[N_i || status || K_{N_i}^+ || validtime]$$

The status parameter determines the security level attributed to the node N_i, if N_i is unknown, the CA_i allocates the visitor's status with a low trust level for N_i node. In addition, the "validtime" parameter determines the valid time duration of the certificate. The forwarding operation is repeated as described above until the border nodes are reached which means $hop - 1 \leq 0$.

3) Certification request: If N_i wants to join the cluster (i), it requests the CA_i by sending the certification request packet, as follows:

$$N_i \rightarrow DDMZ_i : \langle \#Id_p, Cert\_req, hop, DDMZ_i, N_i, S \rangle$$

where, $S = SIN_{K_{ddmz}^-}(N_i || K_{N_i}^+)$. All certification request should passed via the $DDMZ_i$ (RA nodes), before arrived to the CA_i. The member nodes of the cluster do not accept to forward the packet of the nodes which do not have the valid certificate except the certification request.

E. Inter-clusters communication

The inter-clusters communication is ensured by the border nodes. For security reasons, not all border nodes can ensure the link between two clusters but they need to have a high trust level to get the getaway status GW, for more details the reader can refer to the trust model in [5]. The communication between GW nodes and ADDMZ (Anonymous DDMZ) must be encrypted. When the border node N_x with a high trust level receives from clusters i and j the cluster beacon HELLO packet P_{Hello}, it will securely request the $ADDMZ$ for each cluster to obtain the GW certificate. The GW certification is generated by CA_i and CA_j nodes after a mutual checking procedure, to be that CA_i and CA_j are confident nodes. The N_x forms the packet to request the gateway certificate as follows:

$$N_x \rightarrow DDMZ_i :$$
$$\langle \#Id_p, hop, DDMZ_i, N_x, Cert_{CA_i}(N_x), U, S \rangle$$

where,

$$\begin{cases} U = Ea_{K_{ddmz_i}^+}(N_x || Cert_{CA_j}(N_x) || CA_j || K_{CA_j}^+ || K_{ddmz_j}^+) \\ S = SIN_{K_{N_x}^-}(\#Id_p || DDMZ_i || Cert_{CA_i}(N_x) || U) \end{cases}$$

Once the $ADDMZ_i$ receives the gateway certification request, it first checks the validity of $Cert_{CA_i}(N_x)$, then it checks the integrity and the validity of $Cert_{CA_j}(N_x)$, then the trust level of the node N_x. If the checking procedure is going well, the $ADDMZ_i$ forwards the packet to the CA_i. The CA_i needs to check that the real identity of the CA_j belongs to the confident community. Hence, the anonymous inter-cluster authentication is needful.

Anonymous inter-clusters authentication: Once the CA_i wants to check if the CA_j role is ensured by the confident node and create the virtual private network between both clusters i and j. The CA_i generates the packet to the CA_j with the random "$r_i = challenge$" used to generate the CA_i ($CA_i = r_i.H_1(ID_i)$) and its pseudonym CA_i. Then, it sends to RA_y the packet which is formed as follows:

$$CA_i \rightarrow RA_y : \langle Es_{k_g^i}(\#Id_p || RA_y || CA_i || Q_1) \rangle$$

where Q_1 is defined as follows:

$$\begin{cases} Q_1 = Es_{SK_{CA_i}^{RA_y}}(\#Id_p, hop, N_x, U_1) \\ U_1 = Ea_{K_{CA_j}^+}[K_{CA_i}^+ || r_i || S] \\ S = SIN_{K_{CA_i}^-}(\#Id_p || N_x || K_{CA_i}^+ || CA_i || r_i) \end{cases}$$

Once the $ADDMZ_i$ has received the packet, only RA_y takes on the certification request of node N_x forwards the packet to its neighbours.

$$DDMZ_i(R_y) \rightarrow N_x : \\ \langle \#Id_p, hop, N_x, Cert_{CA_i}(DDMZ_i), U_1, S \rangle$$

where, $S = SIN_{K_{ddmz_i}^-}(\#Id_p || N_x || Cert_{CA_i}(DDMZ_i) || U_1)$

When the N_x receives the packet from $ADDMZ_i$ and after the integrity and authentication checking procedure has been carried out by using the parameter S, it uses its certification $Cert_{CA_j}(N_x)$ to communicate with the cluster j and it sends the following packet:

$$N_x \rightarrow DDMZ_j : \\ \langle \#Id_p, hop, DDMZ_j, Cert_{CA_j}(N_x), U_1, S \rangle$$

where, $S = SIN_{K_{N_x}^-}(\#Id_p || DDMZ_j || Cert_{CA_j}(N_x) || U_1)$ and U_1 is the same block receiver as the one from the $DDMZ_i$.

Once, the $DDMZ_j$ has received the packet from N_x and after checking the hop, $Cert_{CA_j}(N_x)$ and integrity with authentication of the packet the $DDMZ_j$ forwards the packets to the CA_j as follows:

$$DDMZ_j(RA_x) \rightarrow CA_j : \langle Es_{K_g^j}(\#Id_p || CA_j || RA_x || Q_2) \rangle$$

where, $Q_2 = Es_{SK_{CA_j}^{RA_k}}(\#Id_p, hop, N_x, U_1)$ and RA_x is the member of the $DDMZ_j$.

After deciphering and checking the $\#Id_p$ and hop parameters, the CA_j node decrypts the U_1 block with its private key and checks the integrity of N_x and $K_{CA_i}^+$ parameters. If the whole verification procedure is going well the CA_j computes the key $K_{i,j}$ and the Ver_j parameters as follows:

$$\begin{cases} K_{i,j} = f(SP_j, CA_i) = f(CA_j, CA_i)^{Sc} \\ Ver_j = H_2(K_{i,j} || r_i || r_j) \end{cases}$$

where, the SP_j is the secret point of the node CA_j and the r_j is the random challenge generated by node CA_j in order to generate its pseudonym $CA_j = r_j.H_1(ID_j)$. Then, CA_j node sends the parameters r_j and Ver_j to node CA_i by the same way as described above.

Once the CA_i has received the packet, the CA_i will use the session shared key with RA_y to decipher the packet and after checking the parameters S of the packet's signature by using the public key of the CA_j in order to be sure that a packet is generated by the CA_j. Moreover, the CA_i uses its private key $K_{CA_i}^-$ to decipher the parameter U_2 (cf figure 2). If the deciphering operation is successful then it computes the $K_{i,j}$ and checks the Ver_j parameter.

$$\begin{cases} K_{i,j} = f(SP_i, CA_j) = f(CA_i, CA_j)^{Sc} \\ Ver'_j = H_2(K_{i,j} || r_i || r_j) \end{cases}$$

If Ver'_j equals to Ver_j then the CA_i deduces that CA_j is a confident node. Then, CA_i generates Ver_i so that CA_j checks if the CA_i is a confident node.

$$Ver_i = H_2(K_{i,j} || r_i || r_j || CA_i || CA_j)$$

Using the same process, node CA_i sends Ver_i to CA_j. Once the CA_j has received the parameters and after the checking procedure has been carried out, CA_j computes Ver'_i ($Ver'_i = H_2(K_{j,i} || r_i || r_j || CA_i || CA_j)$) and checks if $Ver'_i == Ver_j$ then CA_j are now sure that CA_i node is a confident node. Therefore, the gateway certification of N_x node is generated by both nodes CA_i and CA_j. Figure 2

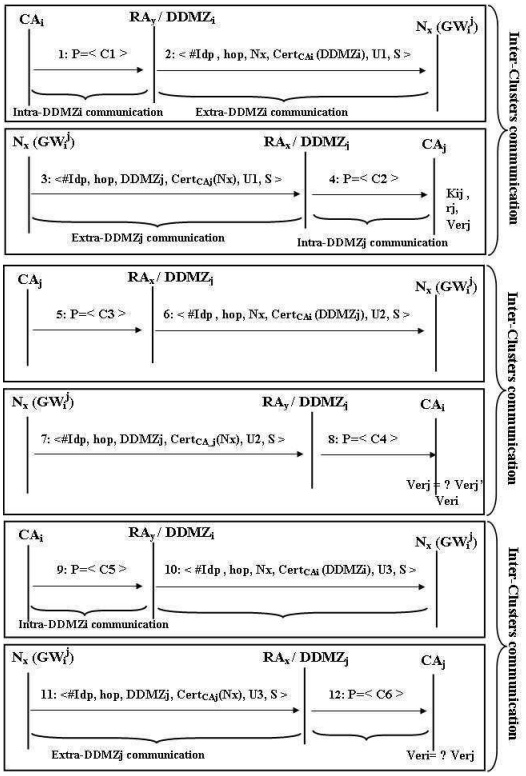

Fig. 2. Anonymous CA nodes authentication

illustrates the anonymous CA nodes authentication protocol. With this protocol any CA node can anonymously authenticate other CA nodes in adjacent clusters.

IV. Security and performance analysis

A. Security analysis

Identity privacy of confident nodes: with ICCP the confident nodes' identities are protected by using the pseudonym or pseudo identity, so that no node is able to guess the real identity of the confident nodes from the pseudonyms. The pseudo identity mechanism permits to dynamically change the pseudo identity of confident nodes such as: CA and RA nodes for each cluster's formation or new cluster's configuration. Security of security services: in order to secure the CA and the RA nodes in each cluster, we did not limit only on the protection of the confident nodes' identity but we

proposed another mechanism to protect the sensitive roles of confident nodes which belong to the DDMZ. We named this mechanism the camouflage mechanism. This mechanism ensures the privacy of the pseudo identity of nodes which belong to the DDMZ group by using the broadcast address allocated to this group.

Group key of DDMZ: the DDMZ's group key is the result of one way hash function of the session keys (SK) set shared between the CA and RA nodes. The RA nodes based on the DDMZ's group key (K_g) to constitute the DDMZ pair keys private and public ($\{K^-_{ddmz}, K^+_{ddmz}\}$). Whenever the group of RA nodes changes because one or more RA node joins or leaves the DDMZ, the group key (K_g) is updated by the CA node. The fact that, the RA nodes are confident, for security reason, updating the group key of the DDMZ ensures the privacy of the DDMZ secret. Hence the DDMZ pair keys private or public change when the DDMZ group key changes. However, even if the attacker has obtained this group key, it is unable to know the session keys or to compromise the cluster.

DoS attacks: Usually the attacker needs to eavesdrop the communication in order to detect the nodes' activities and plan its attack against them. However, with ICCP protocol the attacker cannot determine who ensures the CA or RA activities. If the attacker node wants to plan an attack against the CA node or the RA node, as first step it needs to identify them. Even if the attacker uses the traffic analysis attacks in order to identify the RA or CA nodes, it can just know if it is located in the vicinity of the DDMZ but it can in no way identify the RA nodes or the CA node, because the RA nodes use the pseudonym DDMZ to communicate as RA nodes and its pseudo identity to communicate normally. However, it is possible to attack RA nodes by selecting randomly nodes at the attacker's neighbourhood, but the risk to detect the attacker is high. Let's suppose that the attacker successes in compromising one confident node N_c, the attacker can obtain the secret point SP_c of N_c node and also its real identity ID_c. However, the attacker cannot compromise the entire trust model and unmask the confident nodes, because with this information the attacker can just check if any node belongs to the set of confident nodes or not.

B. Performance analysis

TABLE II
TABLE OF DEFINITION AND NOTATION FOR PERFORMANCE ESTIMATION

T_P	Time for the pairing function computation
T_X	Time for the modular exponentiation in G_1
T_M	Time for the modular multiplication in G_1
T_H	Time for the hashing computation
T_E	Time for asymmetric encryption operation
T_D	Time for asymmetric decryption operation
T_S	Time for symmetric encryption and decryption operation

In order to analyse the performance of ICCP protocol in term of time complexity (TC) in different phases, we define the set of notations illustrated in table II.

The identity change phase: the time complexity of the identity change for each N_x is estimated as follows: $TC(N_x) = T_H + 2.T_M$ which is acceptable for each cluster's configuration or formation.

Anonymous confident authentication phase: the time complexity of the anonymous authentication between two confident nodes N_x and N_y is estimated as follows:

$$\begin{cases} TC(N_x) = T_E + T_X + T_P + 2.(T_H + T_S) \\ TC(N_y) = T_D + T_X + T_P + 2.(T_H + T_S) \end{cases}$$

This phase is executed before the cluster's formation and in the first communication between two clusters.

The anonymity establishment in the cluster: the time complexity to establish the anonymity in the cluster is estimated for the CA and RA nodes as follows:

$$\begin{cases} TC(CA) = 2.T_M + (3+k).T_H + (2+k).T_X + k.T_S + T_E \\ TC(RA) = k.T_D + 2.(T_X + T_H) + T_S \end{cases}$$

where k is the number of RA nodes in the cluster.

V. CONCLUSION

In this paper, we investigate on the existing anonymity protocols, and we proposed the identity Change and Camouflage Protocols (ICCP) based on some military defence mechanisms. The proposed protocol named ICCP is based on the clustering approach particularly our hierarchical architecture [5]. We design a mechanism that allows any confident node to anonymously authenticate other confident nodes. Furthermore, we illustrate, how we can establish an anonymous DDMZ (Dynamic Demilitarized Zone). In addition, two kinds of the intra-cluster communication is presented: intra-ADDMZ and Extra-ADDMZ. Furthermore, the inter-cluster communication is investigated and the protocol to CA nodes authentication is presented. The ICCP is designed to resist against different attacks such as: DoS or capture attacks. In order to evaluate the ICCP, the time complexity and security analysis are presented. The ICCP can be extended to secure the routing protocol. As future work, we plan to implement and simulate the ICCP in heterogeneous nodes.

REFERENCES

[1] X. Huang, W. Susilo, Y. Mu and F. Zhang, *Short (identity-based) strong designated verifier signature schemes*, 2^{nd} International Conference Information Security Practice and Exprience, LNCS 3903, pp.214-225, 2006.

[2] J. Kong, X. Hong and M. Gerla, *An Identity-Free and On-Demand Routing Scheme against Anonymity Threats in Mobile Ad Hoc Networks*, IEEE Transactions on Mobile Computing, Vol. 6, pp. 888-902, 2007.

[3] Y. Zhang, W. Liu and W. Lou, *Anonymous Communication in Mobile Ad Hoc Networks*, IEEE INFOCOM'05, 24th Annual Joint Conference of the IEEE Computer and Communications Societies, 2005.

[4] Sk. Md. M. Rahman, A. Inomata, T. Okamoto, M. Mambo and E. Okamoto, *Anonymous Secure Communication in Wireless Mobile Ad-hoc Networks*, Cryptology ePrint Archive, Report 2006/328, *http://eprint.iacr.org/*, 2006.

[5] A. Rachedi and A. Benslimane, *A Secure Architecture for Mobile Ad Hoc Networks*, 2nd International Conference on Mobile Ad-hoc and Sensor Networks (MSN'2006), LNCS 4325, Hong Kong, pp. 424-435, 2006.

[6] K. El-Khatib, L. Korba, R. Song and G. Yee, *Secure Dynamic Routing Algorithm for Ad hoc wireless Networks*, International Conference on Parallel Processing Workshops (ICPPW'03), 2003.

[7] M.G. Reed, P.F. Syverson, D.M. Goldschlag, *Anonymous connections and onion routing*, IEEE Journal on Selected Areas in Communications, Vol.16, Issue 4, pp.:482-494, 1998.