WATCHMAN: An Overlay Distributed AAA Architecture for Mobile Ad hoc Networks

Amir R. Khakpour, Maryline Laurent-Maknavicius, and Hakima Chaouchi
GET/INT, Institut National des Télécommunications, 91011 Evry, France

***Abstract*— Access control concerns in MANETs are very serious and considered as a crucial challenge for operators who prospects to employ unrivaled capabilities of such networks for different applications. We propose a novel hierarchical distributed AAA architecture for proactive link state routing protocols notably OLSR [1]. This proposal contains a lightweight and secure design of an overlay authentication and authorization paradigm for mobile nodes as well as a reliable accounting system to enable operators to charge nodes based on their connection duration time. We also suggest a hierarchical distributed AAA (Authentication, Authorization, and Accounting) server architecture with resource and location aware election mechanism. Moreover, this proposal mitigates the OLSR security issues [2] noticeably and eventually defines a node priority-based quality of service.**

***Index Terms*—AAA Architecture, MANET, OLSR, Security.**

I. Introduction

MANETs are multihop-based wireless networks in which nodes are roaming freely in an arbitrary way [3]. This open and dynamic topology raises some security issues in which the network itself or overlay networks need a trusted infrastructure to function properly. For instance, the nodes associate to the network and participate in routing and forwarding messages with low and sometimes no security considerations. If we want to use ad hoc technology for public services deployment, it is important to identify the participating nodes. In this case, nodes are required to prove their identity to the network which is nothing but a group of mobile nodes connecting together. Thus, trust management in such networks which has no specific authority reference point by which the node's identity verification could be done causes some difficulties. To face these difficulties, decentralized trust management [4] and PolicyMaker [4] suggested by Blaze et al. try to formulate the security credentials and policies under a common language. It also performs the authorization by a simple query to the compliance checker using a key instead of a user, or person who asks for authorizations, Keynote [5], SPKI [6], and others [7] approximately follow the same concept.

On the other hand, Zhou and Haas [8] argued to use threshold cryptography for key management in the ad hoc environment. To deal with low physical security and availability constraints, they suggest distributing the trust to a specific number of nodes. Therefore, the chance that *t+1* nodes be compromised is significantly less than the chance of one node failure or compromising.

However, the authentication, authorization, and accounting (AAA) method we propose is quite different compared to existing methods. In this architecture, nodes are known by their public/private keys. Besides, the operator assigns an Authorization Trust Level (ATL) to each node by which a node can access to different network resources. We also introduce an accounting system according to how long node's connection lasts in order to perform billing, network trend analysis, and capacity planning [17].

Looking into the AAA architecture in mobile environment, since the mobile ad hoc networks applications are not commercial in many domains, the AAA architecture does not receive a great deal of attention in MANET community. The current and proposed security paradigms are confined to performing authentication [7,8,18] with some limited authorization features along with a few accounting methods. However, these methods and techniques are not comprehensive enough for the network operators to define different accounting classes and charge the users based on the consumed network resources and requested services. Thus, emerging AAA architectures for MANETs to bind these basic functionalities and provide some features to enable network operators to define different policies supplying different levels of resources/service and charging the users accordingly is needful.

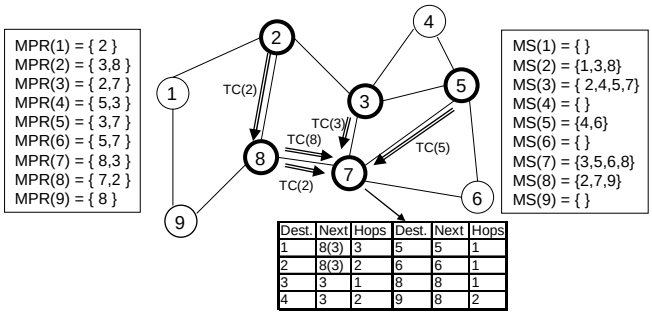
On the technical point of view, implementing a centralized AAA architecture using static repositories as user directories, AAA policy database, and accounting logs on distributed and mobile systems is challenging. Considering all above facts, the main contribution in this paper is to provide a light overlay AAA architecture with simple data structures and replicated repositories with dynamic architecture. We also tried to distribute the different AAA component on the nodes through different election procedures to avoid network bottleneck and single point of failure.

The remainder of this paper is organized as follows. In section II, we survey protocols related to our work in brief. In section III, we discuss the methodology of this architecture. We present some specific security and QoS features and prospects on WATCHMAN in section IV and V. Section VI summarizes our conclusions and addresses the considered future work.

II. Background

A. Optimized Link State Routing (OLSR)

The OLSR is a proactive and table-driven routing protocol for mobile ad hoc networks. The idea behind this protocol is



Dest.	Next	Hops	Dest.	Next	Hops
1	8(3)	3	5	5	1
2	8(3)	2	6	6	1
3	3	1	8	8	1
4	3	2	9	8	2

Fig. 1. OLSR sample calculation for MPR, MS set, and routing table

to reduce the number of flooded routing messages in the network significantly. In OLSR, each node needs to report its set of neighbors with their corresponding link state to its adjacent nodes periodically through *HELLO* messages. Using these HELLO messages, each node *i* selects a set of 1-hop neighbors, called *multipoint relays* MPR(i), which connect i to the 2-hop neighbor(s). It also recognize if it is the member of MPR(*j*) or not (where *j* is the one of the *i*'s 1-hop neighbors). MPR nodes process the receiving HELLO messages but will not forward them; instead, they create Topology Control (TC) messages containing the MPR Selector (MS) set, where MS(*i*) is the set of nodes which have chosen *i* as their MPR. The TC messages are generated periodically and be forwarded only by MPRs to all of the nodes in the network. Eventually, all nodes calculate their routing table based on the receiving TC messages. Fig. 2 demonstrates a simple example of OLSR signaling in which node 7 receives the TC messages from the MPRs to calculate its routing table.

OLSR is suited to dense network with random and sporadic network traffic contrary to classic link state routing protocols use flooding for message propagation. Thanks to periodic TC generation, the routing tables are quickly updated when a link status changes. Therefore, if a node joins to or detaches from the network, all the nodes detect it and recalculate their routing table.

B. Security Issues in OLSR

The main security issues in OLSR are referring to node misbehavior. A malicious node can harm the network in the following ways: [2]

--Jamming: A node can generate and send a huge volume of junk messages to its neighbors. These messages can be either control messages like TC messages that can influence the entire network because they are broadcasted to all nodes, or HELLO messages to impact the neighbors' performance and to occupy the node available bandwidth for data transmission.

--Incorrect message generation: Adversaries can easily send fake and erroneous HELLO messages and TC messages to their neighbors. It can affect the MPR selection process, the routing calculation, and finally damage the whole network functionality.

--Incorrect traffic relaying: Since nearly all nodes are performing as routers to relay messages (control messages like TC messages and data messages), any misbehaving node can make routing disruptions or data flow interruption. Replay attack (also called wormhole attack) in which a node replays the recorded control messages of a region in another region is also categorized in this group of attacks.

To counter these security issues, some solutions are provided. Adjih et al. in [10] employ cryptographic functions to create signatures for the control message in addition to different time stamps for auditing the control messages. Hong et al. [11] also exploited cryptographic functions and proposed SOLSR to resolve some of the mentioned attacks.

III. METHOD DESCRIPTION

A. Basic Definitions and Assumptions

A typical AAA architecture consists of a "*client"* who demands for permission to connect to the network, a *NAS (Network Access Server)* which authenticates and authorizes client for connection like an "*authenticator"*, and an *AAA server* which verifies user identification according to user and policy repository [12]. Likewise, in WATCHMAN, the client or "*supplicant*" is a single mobile node roaming around and seeking for beacon signals from a network to attach to and become a new member. Besides, since it is not feasible to define a single access gateway for a mobile ad hoc network, each non-selfish node (we assume having a technique to identify selfish nodes, for instance reputation based techniques) being already accepted as an ad hoc member could be "*authenticator*" of the supplicant. And finally, a node or group of nodes that are able to be *AAA server* is elected to be either a *Master AAA Server (MAS)* or a *Slave AAA Server (SAS)* (Fig. 2). In one judicious scenario, the operator could configure his own trusted nodes to be AAA nodes.

We also define three data structures:

--User Database (UDb): UDb is a database held by AAA servers including the users' information and used by AAA different modules. (Fig 3.a)

--Trust Cache Table (TCT): TCT is a table held by each connected node which includes the network authenticated nodes in WATCHMAN architecture. Accordingly, a record will be added to the table after a successful authentication, and will be eliminated if a node is not available for *TCT_timeout*. (Fig 3.b)

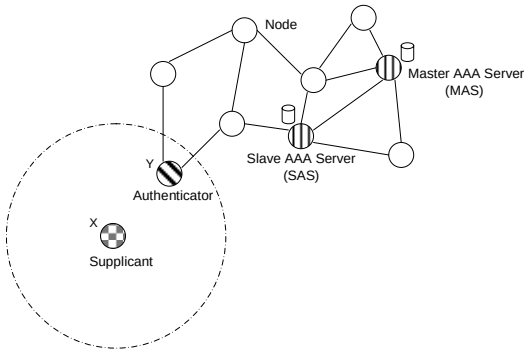


Fig. 2. AAA framework architecture in pure ad hoc networks with *N=1*

User ID	Public Key	Authorization Trust Level (ATL)	Custom Parameters
Alice	+pB/dU….	C4	Enable

a) User database sample

Network Address (IP)	ATL	Starting Time	Last Presence Time
192.168.32.49	C4	12:37:23	13:23:02

b) Trust Cache Table (TCT) sample

User ID	Network Address (IP)	Authentication Time	Last Presence Time
Alice	192.168.32.49	12:37:22	13:28:37

c) Accounting log sample

Fig. 3.Different tables samples

--Accounting Information: It is a set of log files that keeps track of user's connectivity which are created and updated by each AAA server individually. (Fig 3.c)

The basic assumption within this paper is those nodes which are network members are authenticated, known and trustable, although they have different trust levels by which the access to different available services and resources on the network is controlled. In the scenario where the ad hoc technology is used to extend the coverage of the network operator, we also assume that the network is initialized by a node which is delegated from the network operator as MAS to distinguish further nodes being eligible to connect to the network or not.

B. AAA Server Election and Assignment

1) The overview and definitions: As part of AAA paradigm, the network needs an AAA server which has repositories for authentication, authorization, and accounting logs. But since the network is mobile and each node is prone to fail or lose its connectivity to the network, due to redundancy purposes, we need at least two AAA servers in the network. To define the responsibility of each server, we need to have one Master AAA Server (MAS) and $N(\geq 1)$ Slave AAA Servers (SASs). Besides its AAA functionality, the master server must elect slave servers among the nodes. This election starts when the number of available SASs is less than N. N is a parameter assigned by the operator based on the expected ad hoc network size to support load balancing and to lessen the network traffic generated for authentication handling. This election is performed based on the willingness (w) of each node to be AAA server. This value which is calculated by nodes individually is a function of the following parameters:

$$w = F(R,T,A,L) = A \cdot \frac{(Rc \cdot R + Tc \cdot T + Lc \cdot L)}{3} \qquad (1)$$

--AAA server node factor (*A*): A node administrator can choose to be an AAA server candidate or not. *A= {0, 1}.*

--Resources (*R*): The nodes which have more available resources such as power and hardware resources (e.g. RAM, CPU, etc.) are preferred to be AAA server.

--Authorization Trust Level (*T*): Those nodes which are more trustable by the operator have more chance to be AAA server.

--Location (*L*): To have a fully and homogeneously scattered AAA service, nodes which have bigger *L* value would preferably choose to be AAA server. The *L* value could be precisely calculated if nodes use global location systems such as GPS, otherwise, the hop count between the master and the candidate could be an appropriate value, especially in the case when *N=1*.

The coefficients *Rc*, *Tc*, and *Lc* are defined by the operator to weight each parameter in order to fulfill the network administration policies.

** Note: 1) All parameters and coefficients are within [0,1].*

2) A node with w=0 is not eligible to be an AAA server.

We also define two election modes:

--Emergency Mode: When the number of connected nodes with $w > 0$ is less than *N*. It means that the MAS can not do election, therefore, as soon as a node with $w > 0$ connects to the network, it is chosen as SAS.

--Normal Mode: In which MAS accomplishes the election process based on *w* value.

According to these definitions, the network starts working with one node which is already MAS and in emergency mode. So each incoming node which authenticates to MAS and has $w > 0$ is assigned as SAS and inherits the required information from MAS. The information MAS delivers to SAS is:

--List of AAA servers with their corresponding *w* values.

--User Database (UDb): contains all the registered users' ID, their corresponding public-key, their authorization trust level, and operator custom parameters for each user which would be helpful for special cases in advanced authentication, e.g. user disabled, out-of-credit, and etc. Each parameter will be explained in the following sections.

--Operative Parameters: Some parameters defined by the operator such as election willingness coefficients (*Rc*, *Tc*, and *Lc*), number of SASs *(N)*, and *w_threshold*. Since *w* is a function of time-variant variables, the SASs calculate their own *w* periodically and send their updated *w* value to other SASs, whenever the difference between the new *w* and the last one exceeds the threshold value ($w\_threshold \in [0, 1]$).

--Accounting logs.

2) Election process: In normal mode, when a SAS fails or loses its connectivity to the network, and eventually removed from TCT, it is dismissed to be an AAA server. The MAS follows the SASs attendance by its TCT, if one of them is missed in TCT, it discerns that the number of available SASs is less than N. It therefore starts election for a new SAS. As the first step, it broadcasts an Election Request (ER) to all nodes, to ask them for their w value. ER message contains the w value coefficients along to the MAS Selection Time (MST) which is the time a node is assigned as a MAS. After the nodes properly respond to the ER message, the candidate who has the largest w is chosen as the winner of the election and inherits the information from the MAS to be a SAS. Finally, MAS informs other SASs about the new SAS and its corresponding w value.

In case of MAS failure, SASs will realize its absence by the TCT and automatically the SAS with the largest *w* value is

self-assigned as a MAS and it does the election for the new SAS.

Different types of nodes have different responses to broadcasted ER messages. Nodes which do not respond have *w=0* and are not taken into account in AAA architecture as AAA server. Other regular nodes send their own *w* value as well as SASs along to an indicator to signify of the node is SAS or not. However, MAS replies with a message pointing out that the node is a MAS and the time it is MST (This special case is discussed in section III.B.5).

3) SAS dismissal: MAS is able to dismiss a SAS and start election. This is probable when the received w values has considerable difference with the SASs w values since the MAS prefers nodes with high w values to be assigned as SAS. The dismissal process is started by a releasing request from MAS, accordingly SAS replies by its accounting logs and be dismissed. In next step, MAS will update other SASs about the dismissal as well as the updates for new SAS assignment. MAS is also able to dismiss itself and make SASs to select a new MAS.

4) Operator connection and MAS discovery: If the operator needs to connect to the ad hoc network either to change any information such as adding users to database or to retrieve the accounting logs for billing system, it is supposed to connect to the MAS directly. Because the MAS is the only AAA server who has the authority to update the UDb or other administrative factors, and broadcast the update messages to SASs.

As a matter of fact, the operator known with the highest ATL value (0xFF) and lowest MST value (equal to zero) is authorized and eligible to dismiss a MAS and appoint a desired node to a MAS. Furthermore, using *MAS Discovery* which is simply done by a flooded ER message accompanied by analysis of the ER responses as well as the network TCT allows the operator to monitor different AAA architecture components, and in case it counters any defects in any framework elements, it would be able to remove and repair them administratively.

5) Multi-MAS Phenomenon: The Multi-MAS Phenomenon might happen while a sparse ad hoc network is split into smaller networks because of mobility. If the gap between networks having at least one AAA server, and it happens for K times in sequence. The number of AAA servers after the merging, will be

$$(N+1) \le Num_{MAS+SAS} = (1+N) \cdot \prod_{i=1}^{K} M_i \le (N+1)^{K+1} \quad (2)$$

Correspondingly, the number of AAA servers will dramatically increase after frequent network separations. Moreover, the merged network endures more than one MAS, which causes problems in later elections.

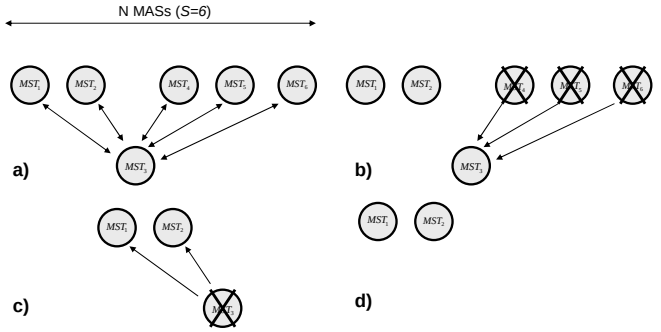


Fig. 4. One step of EMEA by comparing the MST values.

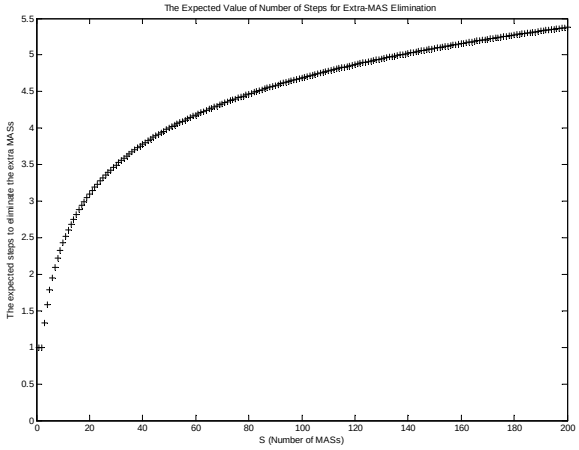


Fig. 5. The mathematical expectation of number of steps to remove the extra MASs in EMEA.

Fig. 4 shows the Extra-MAS Elimination Algorithm (EMEA). In this figure, we suppose S as the number of MASs, MST_i (the time that node *i* is assigned as a MAS). And the MASs are assorted in the way that $MST_1 < MST_2 < MST_3 < ... < MST_S$.

Based on WATCHMAN, the Multi-MAS Phenomenon is exposed when the MAS whose SAS(s) is missing broadcasts the ER message for the election. The other MASs which receive the ER messages compare the MST field with their own MST followed by dispatching their MST value to the MAS which sends the ER message (Fig 4.a). The elimination algorithm is based on removing the MASs with greater MSTs. Consequently, the MAS with greater MST will collect the accounting logs from its SASs and hand over them together with its own accounting logs to another one (the one with lower MST) (Fig 4.b and 4.c) and subsequently release itself and its own SASs (Fig 4.d). Ultimately, the MAS(s) sends the new accounting logs to its SASs for redundancy purposes. Using this technique which is one step of EMEA, not all but some of the MASs are dismissed.

The expected value (mathematical expectation) of the number of steps needs to be taken to remove all unnecessary MASs and resolve this phenomenon is:

$$P_1 = 1, \quad P_2 = 1, \quad P_3 = 1 + \frac{1}{2}P_1 + \frac{1}{2}P_2$$

$$P_i = 1 + \frac{1}{i-1}\sum_{j=1}^{i-1} P_j \qquad (i \ge 3)$$

$$E(Number\ \ of\ \ steps\ \ for\ S\ MASs) = \frac{1}{S} \cdot \sum_{i=1}^{S} P_i$$

Where P_i is the expected number of steps if node *i* is chosen.

The Fig. 5 shows the effectiveness of the algorithm in removing the extra MASs. For instance, if in large-scale ad hoc network the number of MASs exceeds 200 nodes, the expected number of steps to remove extra MASs is approximately 5 which is low enough to remove all extra MASs quickly. Because once the ad hoc network comprises 200 MASs, it will have *200*N* SASs, and missing 5 SAS among *200*N* which triggers the EMEA step is quite probable in short time.

C. Authentication and Authorization

1) Node to Network Authentication: Suppose the disconnected node (X) is looking for signal from the ad hoc network's node to connect, it is either a former network member disconnected because of mobility or a new node who wants to join the network. After listening the network beacons from a node (Y) and performing the data link association process, nodes X and Y send a HELLO message for neighbor discovery. Node Y hears that HELLO message; it checks its Trust Cache Table (TCT) to know whether the node already authenticated or not. If the X address is not included in Y's TCT, Y won't accept the HELLO message, and it asks for authentication unless it is a selfish node. Because the authenticator needs to be an MPR after authentication process, a selfish node may not accept a new user HELLO message so it drops the message.

Node X also receives HELLO message from Y, but since X's TCT is empty, it is considered as a single node willing to authenticate as a supplicant, so it accepts the incoming HELLO messages, and start authentication. The authentication time diagram is shown in Fig. 6.

For authentication, Y changes its status to authenticator and asks for X (supplicant) ID. The supplicant's identification in this security architecture is node's public key (KU_S). The supplicant sends KU_S to authenticator. The authenticator verifies if it, itself, is AAA server or not, if it is, it looks up the public key in its own UDb, otherwise it forwards KU_S to one of the AAA servers found in a list of AAA servers it belongs. Each node knows about the AAA servers addresses. They can obtain these addresses either from their authenticator when they log into the network or by listening broadcasted election messages from MAS. If they forward KU_S and receive no

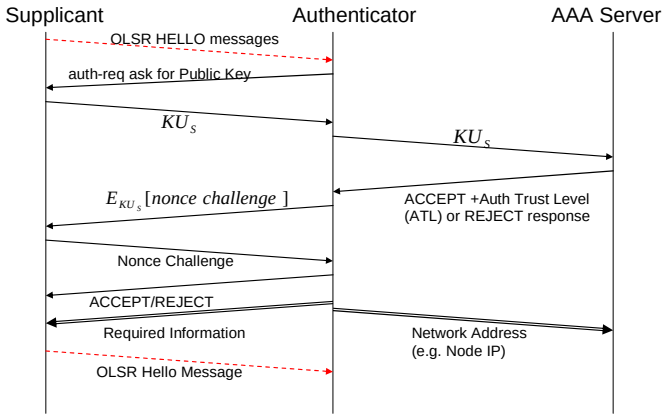


Fig. 6. Authentication time diagram

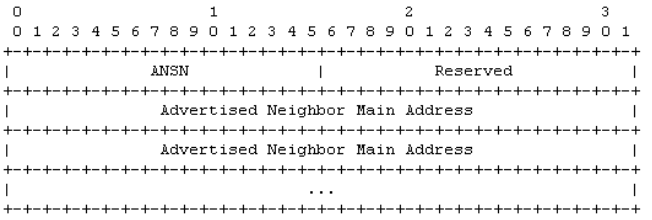


Fig. 7. OLSR TC packet format

with its private key (KR_S) and sends it back to the authenticator. If the response matches the original nonce, the authenticator would be convinced that the supplicant is the real owner of the sent public key so it grants network access to the node, and accepts supplicant's next HELLO message. It also provides the supplicant with other information such as:

--The required network address (e.g. IP address) or all information the supplicant needs to connect to the network such as the security information for example secret symmetric key, etc. Since there are no fixed presumed configurations for MANETs, this information depends upon the network considerations.

--The list of AAA server addresses

--Node's ATL

--Its own TCT

And finally, for accounting purposes, the authenticator sends the supplicant assigned network address to AAA server. With this address, the AAA server can keep track of user connections and disconnections. Besides, AAA server have to keep other AAA servers posted for new network information to node ID mapping to enable them to accomplish their accounting tasks properly. The brief description of mentioned signaling is as follows.

Authentication Algorithm Timeline

1:A→S: auth-req //Authentication request
2:S→A: KU_S //Supplicant Public-Key
3:A→AAAS: KU_S
4:AAAS→A: Resp1 //*Resp1:*ACCEPT/ATL or REJECT
5:A→S: $E_{KU_x}[No]$ //*No* is a nonce generated by A
6:S→A: No
7:A→S:Resp2 // *Resp2:*ACCEPT or REJECT
8:A→S: Inf1 A→AAAS: inf2

Where: *A* is the Authenticator.
S is the Supplicant.
AAAS is the AAA Server.
Inf1: The required information for the supplicant after authentication.
Inf2: Node ID/Network address mapping for accounting purposes.

After a successful authentication, the authenticator is chosen as an OLSR Multipoint Relay (MPR), if it is not already the case. The authenticator is the first and may be the only node that connects the supplicant to the rest of the network members. So the new member (X) will be added to

multiple relay selector set of the authenticator, and subsequently it is advertised to the network by first incrementing the Advertised Neighbor Sequence Number (ANSN) in TC message header, and then adding the supplicant's address to its new TC message in Multipoint Relay Selector Address field [1] (Fig. 7). The authenticator also has to advertise the corresponding ATL of the node to the network, because as soon as this TC message broadcasts to the network, all nodes start updating their TCT for the new node. In this architecture, we propose to use the "*Reserved*" field in TC message to advertise the new node ATL value. Consequently, based on OLSR, it is expected that network nodes which detect that "ANSN" field is increased check the multipoint relay selector addresses for any changes of multipoint relay selector set, and if they detect a new node which is not already in their TCT, they add it with a corresponding ATL value using "Reserved" field.

In order to know more about TCT (Fig. 3.b), each record in TCT contains the node address, its corresponding ATL, and last time it was detected in TC messages. If this time does not update after a timeout (*TCT_timeout*), the record is deleted from TCT. As such in case of mobility, a node coming back to the network before the timeout expires, does not need to re-authenticate to the network; otherwise re-authentication is obligatory. Fig. 8 shows TCT table alteration after a successful authentication.

2) Network to Network Authentication: Network to Network Authentication: If a gap divides the ad hoc network into two parts and it lasts for more than *TCT_timeout*, the network merging needs re-authentication. If both sides have the AAA server, the border nodes have to re-authenticate mutually. Then, they accept their HELLO messages and the TCT tables will be updated by subsequent TC messages. Since these TC messages have no ATL values, the new nodes are added with "pending" ATL values. When a node refers to its TCT to use the ATL value of a specific node which is pending, it floods the network with its request. Nodes exploit the query/response aggregation in order to avoid replaying messages. Regular nodes and AAA servers check their TCT and user database, respectively, whether they can reply or not. The response is routed back to the node that sent the query in exactly the same way it was forwarded, so all in between nodes can update their TCT with new ATL value. If no response is received after a timeout, the record corresponding to the node is deleted from TCT.

If a dislocated network has no AAA server, its members need to re-authenticate to another network individually.

D. Accounting

According to this architecture the only factor we can monitor for the charging system from the operator's point of view is the node's connectivity time duration, although applications and services on the network may have their own accounting and billing system.

Each AAA server has an accounting log (Fig. 3.c) which was originally received from the MAS during election process. The AAA servers keep on logging all the TCT record insertion and deletion. This accounting log indicates how long each node was connected and the operator can retrieve this log by connecting to MAS for its own charging system. According to this proposal, even if one AAA server fails, the accounting information is held by another redundant node, although there might be trivial differences between their logs, which are due to TC messages propagation delay.

For billing system, the operator connects sporadically to the network. After a successful authentication, it starts MAS Discovery followed by fetching the accounting log from the MAS. Meanwhile, if the network is experiencing the Multi-MAS Phenomenon, the operator connection will resolve the phenomenon in one step.

Since the veracity of the accounting log is very important, for integrity check, the operator node may ask the MAS to retrieve the accounting logs from other SASs with a digital signature control and forward them besides its own accounting logs. This integrity also might be compromised if the operator fetching time intervals are large. Because it would be possible that all accounting logs are initiated from a forged one, suppose in case when all AAA servers except the malicious one are missing, the malicious one will become the MAS and delivers the wrong accounting logs to its new elected SASs. Avoiding this integrity jeopardy, assigning a stable node as MAS as well as frequent operator connection is highly recommended

Finally, the staled accounting logs will be removed from MAS(s) and SASs. This is very important since the multiple copies of accounting logs which are growing swiftly is considerably resource consuming. Therefore, frequent operator connection is crucially helpful for resource preservations and security considerations due to integrity of accounting logs.

For charging system, the operator needs to merge accounting logs and eliminate the time overlaps, in addition to subtract the *TCT_timeout* for each connection as well as TC message average propagation delay. It would be possible that the accounting log aggregation is accomplished in the MAS to avoid the resource consumption, in case the operator is not able to connect to the network frequently. However, an effective algorithm should be developed to prevent the MAS to burden massive computations. Besides, log aggregation in MAS leads to removing multiple copies of accounting log that

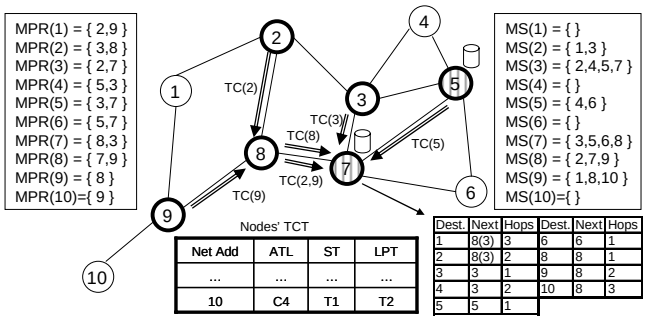


Fig. 8. After authentication of Node 10, Node 9 changes to MPR and Node 10 is added to nodes' TCT and routing tables.

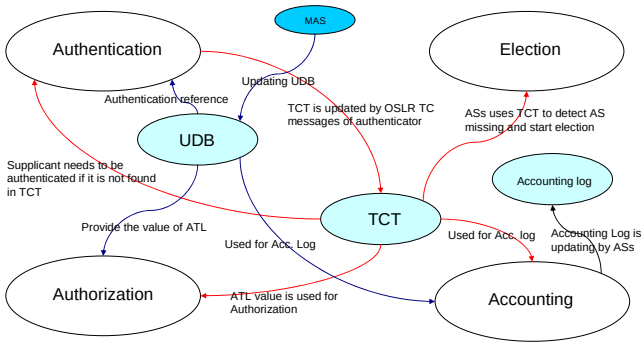


Fig. 9. Schematic view of main components of WATCHMAN, basic data structures and their interaction in a nutshell

raises some security concerns over the results integrity as well as providing a single point of trust that may be compromised by adversaries.

Fig. 9 shows briefly the basic components and data structures interacting together with different WATCHMAN framework elements.

IV. WATCHMAN AND SECURITY

A. *Trust Management*

Based on our primitive assumption addressed in section III.A, the nodes which are authenticated are known by the network and they are trustable. Thus, we follow a chain-based trust management in this architecture. This chain is initiated by the operator node which performs as MAS, and continues by authenticated nodes which inherit authorization from their authenticator as a delegate to authorize new nodes. Therefore, if one node in this chain is compromised, the chain will break and the next authenticated nodes may not be considered as trustworthy nodes. To discover these untrustworthy nodes, a quite large *auth_timeout* value is defined by which MAS asks for re-authentication for the TCT records whose authorization expired due to this timeout. In this case, the MAS sends a re-authentication request to the node; this re-authentication request is processed by the nodes MPR to ask for node re-authentication. To prevent DoS attacks, if the re-authentication request originator is in the authenticator's AAA server list, the node accepts the request and performs re-authentication; otherwise, it ensures the integrity of the request by asking its AAA server whether the re-authentication request was coming from MAS or not. If the re-authentication is not successful, the TCT record will be deleted. Additionally, taking advantage of this technique, the operator and the MAS are able to disconnect a user manually from network by sending a re-authentication request and disabling the user in UDb. In practice, the operator or the MAS utilize this technique to expel the detected malicious node from the ad hoc network.

B. *Asymmetric Cryptography using PKI*

In WATCHMAN, we use asymmetric cryptography in which the key pair is generated by the operator and offered the node before node connection. The operator also makes a copy of the public key in UDb and assigns this as user identification. Thus, WATCHMAN suggests a declassified user database to encounter user's identity spoofing vulnerability. Otherwise, it would be potentially possible for AAA servers which has the database to spoof a user identity whose ATL is the biggest one for profiting of a larger panel of services.

Since this framework is designed in a way that no confidential information is stored in or handed over among AAA systems, operational transaction either between operator and MAS or between AAA servers needs only digital signatures [13] for integrity check and rejecting spoofed messages.

In this paradigm, as long as the keys are permanent we do not need a complicated key management system; nonetheless, SASs plays the Certificate Authority (CA) role for nodes seeking for each other's public-key. This helps to secure the data transaction in hostile network environment and provide some minimum security considerations in WATCHMAN signaling for different jobs and procedures.

Furthermore, WATCHMAN may easily host the Mobile Certificate Authority (MOCA) [16] and act as a substructure for MOCA key management framework. Because MOCA is working based on network heterogeneity and in fact WATCHMAN ranked the nodes to AAA servers and regular nodes. Thus, elected AAA servers may be used as *N+1* MOCAs and empower the nodes to use *k* of them for threshold cryptography using temporary keys and threshold digital signatures for data transactions through the network.

C. *Addressing Routing Security*

As it is mentioned in section II.B, most of the attacks considered for OLSR routing are because of nodes misbehavior. In this architecture, due to the fact that all nodes need authentication before associating to the network and collaborating in the network activities, the routing security issues are significantly mitigates. Moreover, the ATL first byte assigned for security purposes might be utilized for MPR selection in order to lessen the concerns on incorrect traffic relaying and incorrect TC message generation.

D. *The Security Vulnerabilities*

Although we tried to design this AAA architecture as secure as possible and the operator knows and trusts the authenticated users, there are several vulnerabilities based on nodes misbehavior in election and authentication process. For instance:

--Adversaries can send a fake *w* value for election to be AAA servers[1].

--Node misbehaving in authentication as an authenticator or AAA server may cause to reject legitimate nodes and accept illegitimate ones; they can also send an incorrect ATL value to the network for new incoming node.

--Accounting log integrity can be jeopardized by malicious

[1] To mitigate this vulnerability, the operator might consider a quite large value for *Tc* and MAS calculate *Tc.T* term in *w* function by itself.

AAA servers[2].

--The ATL value can be manipulated by malicious nodes when they are forwarding the TC message to the network, although AAA servers are able to warn the network if they detect erroneous ATL values.

Most of these vulnerabilities could be degraded by some administrative restrictive countermeasures yet they are costly in resource consumption and transmission overhead perspectives. Besides, using some security policies and assorting nodes in hierarchies for different responsibilities based on their security considerations is a critical trade off in mobile ad hoc networks in which nodes are prone to failure and may roam with a random pattern.

V. WATCHMAN AND QUALITY OF SERVICE

A. Operation Analysis

The design of each part of this architecture is targeting a minimum signaling overhead as well as calculation cost. Accordingly, different tasks are quite fairly shared among authenticator and distributed AAA servers. The calculation cost and overhead signaling is trivial compared to OLSR signaling and routing computations.

The authentication based on the TCT table and re-authentication algorithm are well-defined according to node mobility consideration where node attaches and detaches to the network frequently. A suitable value for *TCT_timeout* avoids nodes to undergo unnecessary re-authentications. Moreover, the required signaling between authenticator and AAA server for authentication is limited to one query/response indication which may be done with digital signatures to evade any security risks in multi-hop communications through the network,

B. Priority-Based QoS

In the view of the fact that MANETs are practically heterogeneous which means that different nodes have different responsibilities and different types of traffic, we define a priority-based quality of service utilizing node's second byte of ATL value. In multi-hop routing process, nodes prioritize the traffic based on the traffic creator ATL. This would be very important for hierarchy based applications such as military and police operations.

VI. CONCLUSIONS AND FUTURE WORKS

This paper presented, WATCHMAN, a hierarchical distributed AAA architecture based on ad hoc link state routing protocols especially OLSR. This proposal targets the usage of the ad hoc technology by service operators to extend their coverage and deploy public services. This proposal contains resource-aware and secure algorithms for different AAA servers' election, a lightweight authentication and authorization method for mobile nodes, and an accounting system which empowers operator for charging and billing system. The applications of such architecture would be military missions, peer to peer gaming, home ad hoc networks, police and fire-fighting operations, etc. WATCHMAN mitigates the OLSR routing security issues and defines a priority based QoS for different nodes.

The Future works would focus on applying some intelligent and light-weight techniques to improve the AAA architecture security in order to remove some basic assumptions and expand the domain of applications for this AAA architecture.

WATCHMAN different modules are implemented and tested successfully, however the core implementation of this architecture as a plugin to OLSRD [14] package is still in progress.

[2] A permanent designated MAS can extenuate the lack of accounting logs integrity assurance.

REFERENCES

[1] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol. RFC 3626, Oct. 2003.

[2] C. Adjih, T. Clausen, A. Laouiti, P. Muhlethaler and D. Raffo, "Securing the OLSR routing protocol with or without compromised nodes in the network", Tech. Report INRIA RR-5494, Feb. 2005.

[3] J.P. Macker and M.S. Corson, "Mobile ad hoc networking and the IETF," *ACM Mobile Computing and Communications Review* 2(3):7-9, July 1998.

[4] M. Blaze, J. Feigenbaum and J. Lacy, "Decentralized Trust Management," *In Proc. of the 17th Symposium on Security and Privacy*, IEEE Computer Society Press, Los Alamitos, 1996, pp. 164 -173.

[5] M. Blaze, J. Feigenbaum, J. Ioannidis and A.D. Keromytis. The KeyNote trust management system. RFC 2704, Sept. 1999.

[6] D. Clarke, J.-E. Elien, C. Ellison, M. Fredette, A. Morcos, and R. L. Rivest, "Certificate chain discovery in SPKI/SDSI," *Journal of Computer Security*, 9(4):285–322, 2001.

[7] G. Theodorakopoulos and J. S. Baras, "Trust Evaluation in Ad Hoc Networks," *In Proc. of the 2004 ACM workshop on Wireless security*, Philadelphia, PA, USA, 2005. pp. 1-10

[8] L. Zhou and Z. J. Haas. "Securing Ad Hoc Networks," *IEEE Networks*, 13(6):24–30, 1999.

[9] C.E. Perkins and P. Bhagwat, "Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers," *In Proc. of ACM SIGCOMM'94*, London, UK, Aug-Sept 1994. pp. 234-244.

[10] C. Adjih, T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, and D. Raffo, "Securing the OLSR protocol," *In Proc. of the 2nd IFIP Annual Mediterranean Ad Hoc Networking Workshop*, Mahdia, Tunisia, June 2003.

[11] F. Hong, L. Hong and C. Fu, "Secure OLSR," *In Proc. of the 19th IEEE International Conference on Advanced Information Networking and Applications*, Tamkang University, Taiwan, March 2005.

[12] C. Metz, "AAA Protocols: Authentication, Authorization, and Accounting for the Internet," *IEEE Internet Computing*, Dec. 1999.

[13] A.J. Menezes, P.C. van Oorschot, S.A. Vanstone. *Handbook of Applied Cryptography*, CRC Press, 1997, pp. 423–435.

[14] OLSRD Package Official Website, Available : http://www.olsr.org/

[15] D. Johnson and D. Maltz, "Dynamic Source Routing in Ad hoc Wireless Networks," In *Computer Communications Review - Proceedings of SIGCOMM'96*, Aug. 1996.

[16] S. Yi and R. Kravets, "MOCA: Mobile certificate authority for wireless ad hoc networks," In *2nd Annual PKI Research Workshop (PKI'03)*, Gaithersburg, MD, Apr. 2003.

[17] A. Khakpour, M. Maknavicious, and H. Chaouchi, "Through Access Control on Mobile Ad hoc Networks: WATCHMAN Project," Tech. Report, GET/INT, LOR Dept., Apr, 2007.

[18] F. Sato, H. Takahira, and T. Mizuno, " Message Authentication Scheme for Mobile Ad hoc Networks," In *Proceedings of the 11th international Conference on Parallel and Distributed Systems (ICPADS'05)*, Fukuoka, Japan, July 2005.