Towards a Security Framework for an Established Autonomous Network

D. Ztoupis, K. Zarifis, I. Stavrakakis
Department of Informatics and Telecommunications
University of Athens
Athens, Greece
d.ztoypis@di.uoa.gr, k.zarifis@di.uoa.gr,
ioannis@di.uoa.gr

C. Xenakis
Department of Technology Education and Digital Systems
University of Piraeus
Piraeus, Greece
xenakis@unipi.gr

***Abstract*—This paper focuses on the security of the Athens Wireless Metropolitan Network (AWMN), which is an established autonomous network. More specifically, it presents and analyzes the possible security attacks that threaten AWMN, its users and the provided services. Attempting to counteract these attacks and the related risks, we study and evaluate the application of known security measures in AWMN that aim at providing authentication services. Authentication is the most vital property of secure communications. Finally, we propose and highlight an authentication model that satisfies the requirements of an autonomous network, like AWMN. The proposed model combines the web of trust and free trust policy approach of the Pretty Good Privacy (PGP) protocol and the non-infrastructure solution of the self-organized public-key management scheme.**

***Keywords*— autonomous networking, security, trust, authentication**

I. INTRODUCTION

Athens Wireless Metropolitan Network (AWMN) is a wireless community that started forming in 2002 [1]. Currently, there are 1900 active nodes in the Attica area, while 2500 more have shown interest in connecting to the network and are awaiting its expansion. As a result, AWMN is today one of the largest wireless network communities on Earth. The network is not a product or a service but rather a place of education, research, entertainment and experimentation, providing a wide variety of services such as mail, FTP, web hosting and game servers, VOIP, P2P file sharing, etc [1]. There is no subscription or any other type of fee and participation is open to anyone.

The network architecture of AWMN is presented in Fig. 1. AWMN is composed of the backbone (BB) network and the access network. The BB network consists of BB nodes, which are responsible for the routing of the transferred data. The routing protocol currently used in AWMN is the Border Gateway Protocol (BGP) [2]. BB links that connect the BB nodes are implemented using the 802.11a standard. Each BB node usually has more than one or two BB links, and thus local loops or star topologies are created within the BB network, resulting in a final complex topology. Based on the number of the established connections, the BB nodes are divided in three categories: (i) nodes with more than two active BB links (Cx category), (ii) nodes with two active BB links (Bx category), and (iii) nodes with one BB link (Ax category). Apart from the BB network, the BB nodes also set up the wireless access network by establishing Access Points (AP) for the wireless clients. In the access network, the connections between the APs and clients are implemented using the 802.11b standard. Based on the analyzed network topology, there is a peer to peer relationship among BB nodes in the BB network, and a hierarchical relationship between the BB nodes that act as APs and the wireless clients in the access network. Currently, there are approximately 850 active BB nodes and 1050 client nodes.

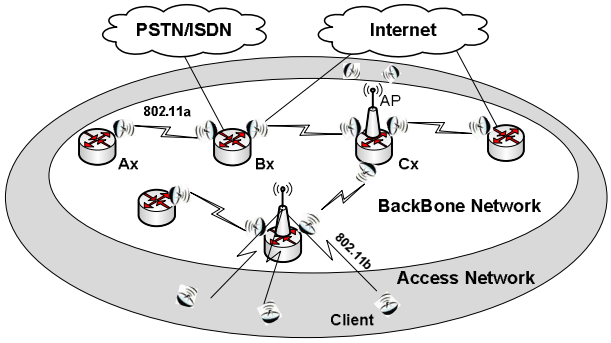

Figure 1. AWMN Topology

AWMN is a rapidly expanding network that hosts services and functionalities similar to those of the Internet. Furthermore, there are gateways linking it to the latter. Consequently, AWMN has to deal with the security risks and threats that threaten the public Internet. In addition, being a wireless network, AWMN is susceptible to any kind of malicious attack that the wireless technology can undergo, since the wireless links can be accessed by anyone who is in range. Lastly but maybe most importantly, AWMN is an experimental network, which means that so far secure activities rely solely on the members' earnestness. Therefore, anyone can become a BB node, and in other words, a router. On the other hand, in classic wired networks routing is the responsibility of providers, which follow specific standards guaranteeing that packets will be forwarded, without being eavesdropped, modified, etc. This, however, is not the case in AWMN. Due to the community based, educational and open source character of the network, the nodes do not feel the need to address security concerns aggressively. This attitude, although understandable at this stage of development, has the side effect of increasing the network's overall vulnerability even more. Until today no actual security measure has been taken, and data transfer is far

from secure, making the network vulnerable to a large set of attacks.

This paper focuses on the security of AWMN, which is an established autonomous network. More specifically, it presents and analyzes the possible security attacks that threaten the network, its users and the provided services. Attempting to counteract these attacks and the related risks, we study and evaluate the application of specific security measures in AWMN that aim at providing authentication services. Authentication is the most vital property of secure communications. There is little sense in trying to create a secure channel with a peer entity, without a guarantee that it is exactly the entity it claims to be. It is a prerequisite for confidentiality, integrity and consequently non-repudiation. Finally, we propose and highlight an authentication model that satisfies the requirements of an autonomous network, like AWMN. The proposed model combines the web of trust and free trust policy approach of the Pretty Good Privacy (PGP) protocol [3] and the non-infrastructure solution of the self-organized public-key management scheme [4].

The rest of this paper is organized as follows: Section II briefly presents and analyzes the possible attacks that threaten AWMN. Section III studies and evaluates the application of known security measures in AWMN that aim at providing authentication services. Section IV proposes and highlights an end-to-end authentication model that combines the web of trust and free trust policy of the PGP approach with the non-infrastructure solution of the self-organized public-key management scheme. Finally, section V contains the conclusions.

II. POSSIBLE ATTACKS AGAINST AWMN

Attacks in AWMN can be carried out from foreign sources (i.e., external attack) as well as nodes belonging to the network (i.e., internal attack) and can be further divided into passive and active attacks, targeting any network layer. Based on the three basic factors that define risk (i.e., Criticality, Vulnerability, Threat [5]), the potential attacks against AWMN are presented and analyzed [6].

Passive Eavesdropping: AWMN is vulnerable to passive eavesdropping attacks, since the transmitted packets are not encrypted. Therefore, any external node that is located close to a link can perform this type of attack by using a sniffer or a wireless card operating in promiscuous mode. In addition, an internal BB attacker can sniff the by-passing packets effortlessly. Apart from compromising the confidentiality of users' data, the potential attackers can get valuable network information such as valid MAC/IP addresses, network topology, etc., that will help them perform other type of attacks.

Authentication - Deauthentication attack: This attack is mainly carried out by external nodes that target mainly the APs and wireless clients. The majority of the AWMN BB nodes use MAC filtering for authentication, keeping track of the MAC addresses of their known clients. When a client wants to connect to an AP, it sends an authentication frame to it that includes the client's MAC address. After receiving the authentication frame, the AP checks if the included MAC address exists on its list. If so, the client is authenticated. The deauthentication procedure works accordingly. Thus, an attacker can sniff the MAC address of a client-target and send a spoofed DEAUTH frame to the AP that the client is connected to. Then, the attacker can send a spoofed authentication frame in order to authenticate itself to the AP. The attacker can also deauthenticate all the authenticated clients of the AP, by impersonating the AP and regularly broadcasting spoofed DEAUTH frames with an omni-directional antenna, forcing the clients to re-authenticate.

Impersonation Attack: The impersonation attack can be easily implemented in AWMN since there is no strong authentication between nodes. A node confirms the identity of another using only the IP address, which can be easily sniffed from the unencrypted IP packets. In addition, BB nodes know the other nodes' IP addresses, as they need them for packet routing. An attacker, impersonating a BGP-router, may forward false routing information in order to produce extra traffic, force packets to follow longer routes (i.e., adding extra delays), throw packets in loops, or overload other routers (i.e., performing DoS). Moreover, the attacker can terminate a communication between two BGP peers by sending a false Notification message or an Open message after the BGP connection establishment.

Man-in-the-middle Attack: An attacker intersects a connection between two nodes (i.e., A & B), and impersonates A when sending to B and vice versa. Every packet sent between A and B passes stealthily through the attacking node, which can modify or discard it. This type of attack is difficult for external attackers, since they have to physically intersect the line between the nodes. This means that the establishment of probably bulky equipment (i.e., laptop, antennas) at inaccessible locations (i.e., rooftops) is required. On the other hand, every BB node is a possible man-in-the-middle attacker. A malicious BB node can easily carry out the attack to the packets going through it, by simply dropping them (i.e., black hole attack). It can also route packets to wrong destinations causing delays. Moreover, due to the lack of security measures for integrity, a BB node can modify data or routing packets. While the modifications of data packets will only affect the communication between the two nodes, modifying routing packets can even result in a total network breakdown.

A BB attacker can also modify the BGP Update messages that it receives, before forwarding them. The attributes that can be modified include:

- The AS_Path field, which denotes the set of nodes that must be traversed to be reached the advertised destination.
- The list of IP prefixes in the Network Layer Reachability Information (NLRI) field that describes routes.
- The list of IP prefixes in the Withdrawn routes field.

If the attacker deletes AS numbers from the AS_Path field, it forces other routers to select paths that go through it. Then, the attacker can launch a black hole attack. Moreover, the modifications to the AS numbers can cause the formation of

loops. On the other hand, the last two types of modifications can cause network malfunction and hindering of data transfer because: (i) packets may not follow optimum paths, (ii) available routes can be considered unavailable and vice versa, and (iii) routers may overload and thus be forced to drop packets. Finally, a BB attacker can generate and distribute fake BGP Update messages (i.e., fabrication attack). This attack differs from modification in the fact that the attacker creates new messages with false data rather than modifying passing Update messages.

III. Evaluation Of Existing Security Solutions

This section studies and evaluates the application of known security mechanisms in AWMN that aim at providing authentication services. The studied authentication mechanisms are divided in two categories: i) *link-layer authentication* and ii) *end-to-end authentication*. Link layer authentication involves authentication between two neighboring nodes, whereas end-to-end authentication describes the authentication procedure between two non-adjacent nodes. In the latter case the mechanism has to involve many intermediate nodes which complicates the process.

A. Link-layer authentication

In link layer authentication, a pre-shared secret key (PSK) has to be agreed between two neighboring nodes before the communication channel is established. This key is thereafter used for encrypting and digesting exchanged messages. This procedure, however, requires either a secure channel through which the pre-shared key will be initially exchanged, or better yet a physical meeting between the persons in charge of the nodes. Although this is one of PSK's drawbacks, it raises no difficulty in AWMN since two prospective neighbors usually have to meet beforehand anyway, in order to arrange matters such as antennae aligning. Thus, the shared key can be prearranged and exchanged with absolute security. Each node in the network has to hold a number of keys, one for each neighbour, which are usually less than five. The advantages of this mechanism include:

- Direct authentication since there is no need for a trusted third party (TTP) to contribute to the authentication procedure.
- Use of symmetric key cryptography, which is less computationally expensive than public key and can provide faster encryption and digesting.

The Wi-Fi Protected Access (WPA) protocol uses the PSK authentication. WPA/PSK provides full security at link-layer preventing all the types of external attacks (i.e., authentication /deauthentication, man-in-the-middle, impersonation, eavesdropping, etc) that exploit security weaknesses at layer 2.

B. End-to-end Authentication

1) Classic PKI: End-to-end authentication in classic wired networks is accomplished through the deployment of a Public Key Infrastructure (PKI). PKI's function is based on the existence of an absolutely trusted entity, known as Certificate Authority (CA). Every node trusts the CA and blindly accepts any certificate signed by it as valid. On the other hand, any unsigned certificate is valueless.

This single CA solution is easy to be implemented in AWMN, however it raises some problems. First of all, the CA presents a single point of failure and thus a successful DoS attack against it may cause a dysfunction to the whole network. Moreover, the CA-node would become a congestion point since it would have to exchange authentication messages with every network node. Finally, there is a problem of granting a single node the authority to act as a CA. This means that the whole trust system would rely on a single entity, and it is probably difficult to find one in an autonomous network that would be globally accepted as trustworthy. Therefore, it would make sense to opt for a more distributed solution.

2) Multiple CAs: To overcome the inconveniencies of classic PKI, several authentication models that are based on the existence of multiple CAs have been proposed [7]. In these models authentication services are provided by multiple replicated authentication servers. These servers use secure channels to exchange the certificates that they create or withdraw among them. Such an authentication scheme solves only some of the problems presented in the previous paragraph. Every node is served by the closest authentication server, preventing the occurrence of congestion points at the CAs. Furthermore, a successful DoS attack against one of the authentication servers will not affect the whole network, and it is highly improbable that all the CAs will be successfully attacked simultaneously. Despite the use of several central authorities, this scheme is still partially centralized and a question arises concerning which nodes will take the responsibility of acting as CAs. In [7], it is suggested that the most trustworthy nodes are assigned this task, without mentioning an election method. Even if these commonly trusted nodes were to be found, there would be an imbalance throughout the nodes, as some would have to spend more resources than others.

3) PGP: Contrary to the models discussed previously, the email encryption protocol (i.e., Pretty Good Privacy - PGP) [3] follows a distributed authentication method, in which all nodes contribute equally. In PGP, every node can act as a CA. When a node A believes that a public key belongs to another node B, it issues a certificate verifying this fact. In this case, A is called as an introducer of B. Node A can check the authenticity of the public key of B based on the certificates associated to it. If A trusts the issuers of these certificates as introducers, then it is convinced about the key's validity. All the certificates issued within the network form a certificate net, called web of trust. Every node collects keys and certificates in its repository, called keyring, forming a web of trust subgraph, which is illustrated in Fig. 2. An edge A->B means that A has issued a certificate for B. The metric on the edge describes how much A trusts B as an introducer: C=complete trust, M=marginal trust, UT= no trust. The authentication decisions of A are based on this subgraph.

A PGP node is free to apply its own trust policy by assigning a certain level of trust to every key held in its keyring (i.e., no trust, marginal trust and complete trust). This trust level describes how much it trusts certificates signed by another node or how much it trusts the other node as an introducer. Based on these trust values, a node decides if it will accept a key as valid or not. A key is considered valid if it has at least C certificates from completely trusted introducers, or it has at least M certificates from marginally trusted introducers. The specific values of C and M are determined by the node, by adjusting the related parameters (i.e., COMPLETES_NEEDED and MARGINALS_NEEDED). For instance, in Fig. 2, if A assigns the value 1 to the COMPLETES_NEEDED parameter, then every key signed by C and B is automatically valid for A (i.e., D & F). On the other hand, if B sets MARGINALS_NEEDED equal to 3, then it cannot authenticate E, since E's key has been signed only by two marginally trusted introducers, D & F. Trust Depth (or CERT_DEPTH) is another important parameter of PGP, which defines the maximum length of a trustworthy certificate-path, indicating how many levels deep you can nest trusted introducers. In the example of Fig. 2, A has to set the value of Trust Depth at least equal to 2 in order to authenticate E. When Trust Depth equals zero, then A has no introducers and authenticates the nodes directly, without taking into account the others' opinions.

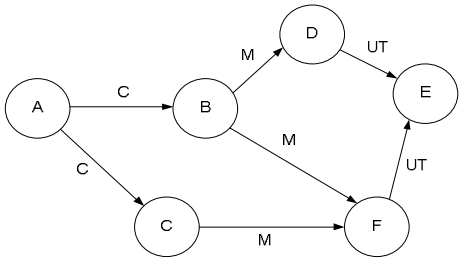

Figure 2. Node A's keyring

In contrast to the PKI schemes, where a node has to accept every key signed by the CA, in PGP every host by itself decides on the authenticity of a public key. Such a framework that follows the web of trust and free trust policy approach seems to suit well in a wireless community network, like AWMN, because every node participates equally in the authentication process. Moreover, it can be expanded to a generic reputation model, without adding new trust metrics. In such a case, a node will assign a trust value to another based on its general behavior and not only on its behavior through the certificate signing. Therefore, a node may consider a completely trusted introducer as trustworthy for the provisions of other services e.g., file sharing, than an untrusted node and vice versa. This will motivate nodes to develop cooperative behaviors leading to the increase of the number of introducers and facilitating the authentication procedure.

However, PGP was initially designed for the Internet that is quite different from AWMN. In PGP implementations over the Internet, public keys and certificates are usually stored and distributed by keyservers (or certificate directories). When a host wants to find a certificate of a user, it has to refer to a keyserver to obtain all the data needed to evaluate the validity of the user key. One of the most important features of the keyserver is the ability to locate certificate trust paths, which connect two keys, functioning as a pathfinder. However, the subgraph formed by a node using its keyring cannot guarantee this function. This is because the subgraph is formed reactively: the node stores a certificate in its keyring only when it wants to authenticate another node. In the example of Fig 3, node A is trying to authenticate E, and therefore it obtains from a keyserver a signed certificates for E. A notices that D and F have signed the key of E, but since A has never communicated with F in the past, neither F's key nor the certificates that introduce F (the dotted lines B→F and C→F) are included in the A's repository. Although the trust paths {A → B → F → E} and {A → C → F → E} exist, A is not aware of them. Finally, the implementation of keyservers in AWMN presents the same drawbacks with the multiple CAs approach, described previously.

4) Threshold Cryptography: Zhou and Haas have proposed the use of threshold cryptography for the distribution of trust in ad hoc/autonomous networks [8]. In this model the private key of the CA is split into n shares, which are distributed to n special nodes (i.e., servers). A node can obtain a valid certificate only if the latter has been signed by at least t special nodes. The fact that no particular node holds the whole secret key of the CA makes this model robust against external attacks. On the other hand this solution, as well as the one described in [9], is a hierarchical solution and thus their application on AWMN raises the problems discussed previously.

Luo et al. [10] have proposed a model where every node holds a share of the CA's private key. It has the advantage of being fully distributed since every node participates in the authentication service, but any t malicious nodes that are colluding are able to produce fake certificates. Increasing the threshold t minimizes the probability of t malicious nodes to collude. However, such a solution adds extra delay in the authentication process, as it requires more signings, which are computationally expensive, in order to publish a valid certificate. Moreover, this model does not allow each node to apply its own trust policy. Specifically, every node has to accept a certificate as valid, even if it knows none of the t nodes that have signed it. On the contrary, it is preferred for a node to trust a single but proven trustworthy peer rather than many unacquainted peers. Finally, in autonomous networks it is considered more preferable for each node to decide on which of the t signatures it will accept as valid, and hence be able to decide whether it will accept or reject a certificate.

5) Self-organized Public-key Management scheme: Capkun et al. [4] have proposed a self-organised public-key management scheme for mobile ad-hoc networks, where every node can issue certificates. In order to avoid the hierarchical storage and distribution of certificates by several certificate directories, they suggest that every node holds its own

repository, which contains all the certificates that have been issued by the node itself or other nodes. Therefore, every node is a keyserver eliminating the need for a central repository. This is achieved by simple flooding, where neighboring nodes periodically exchange all the certificates that they have signed or received from other neighbors. Thus, after a certain time period, named as convergence time, a newly signed certificate will have been distributed to every node. However, such an approach requires high storing capacity on every node and consumes bandwidth resources.

IV. PROPOSED AUTHENTICATION MODEL

In this section, we propose an end-to-end authentication model that combines the web of trust and free trust policy of the PGP approach with the non-infrastructure solution of the self-organized public-key management scheme presented previously. Based on the evaluation of the existing end-to-end authentication solutions for their application in AWMN, the proposed model should satisfy the following design goals:

- Incorporates the PGP policy mechanism.
- Avoids the pathfinder problem (example in Fig. 3) by enabling the detection of existing trusted paths in the deployed keyrings.
- Minimizes the required storage capacity (certificate repository).
- Eliminates the bandwidth consumption (certificate exchange).

In order to achieve these goals, the proposed authentication scheme enables the exchange of certificates only between trusted peers. Thus, the nodes that will participate in the certificate exchange for a specific node are defined by its PGP trust policy. Each time the node issues a certificate, it informs its completely trusted and possible marginally trusted nodes by sending to them the new signed certificate.

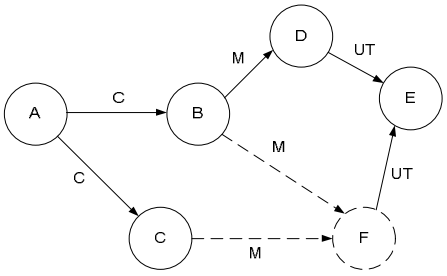

Figure 3. An example of node A's subgraph

The integration of trust metrics in the proposed model gives every node the ability to apply its own authentication policy and makes the authentication mechanism more robust. In Capkun's scheme, every node stores every other node's certificates in its repository, regardless of the possibility of never using most of them. This occurs because there is not a predefined trust policy on each node. In addition, every node trusts all the others, until it is given a reason not to do so. If a node receives a certificate that conflicts with a previous one (i.e., two different entities have the same public key), then the node investigates the validity of both certificates and discards the invalid. However, until that moment the first certificate, which might be invalid, is part of the node's trust graph. On the other hand, in PGP the node has to be assured first about the validity of a certificate, before it adds it to its keyring. Similarly, in the proposed model a keyring will only store trusted certificates, making the scheme more robust.

As mentioned above, each time a node signs a certificate, it will inform its trusted peers. As a result the nodes' keyrings are constructed proactively, meaning that any node is aware of all the available trust paths at any time (i.e., a new trust path is stored in the keyring at the time of its creation). When a node needs to authenticate another node, it has to consult its keyring. If there exists no trust chain in the keyring connecting the node to the target, there is no trust path to it and the node cannot authenticate the target. On the contrary, in classic PGP scenarios the node has to consult a pathfinder every time it wants to authenticate another node, checking for a trust chain. In the proposed model this is done proactively and thus after the certificate exchange procedure every node becomes its own pathfinder. Let us assume the example of Fig. 3, where nodes B and C issue certificates for F (at time t1 and t2 respectively) indicating the validity of F's key, and that they marginally trust F as an introducer. Since both B and C trust A, they have to inform A of their trust towards F by sending their certificates to it. Then, A can add the edges {B→F} and {C→F} to its subgraph forming a trust path that leads from A to F. At any time t, t>t1, t2 the node A will be able to authenticate E, since E's key has been signed by F, and A's keyring contains two trust paths leading to F (A→B→F and A→C→F).

Regarding the requirements in storage capacity, we have seen that storing every certificate issued by every node (i.e., Capkun model) leads to the usage of unnecessary memory space in each node. In the proposed model every node stores only the certificates issued by its trusted peers, reducing the storage capacity requirements. To reduce even more the storage requirements, without affecting the system's performance, a node can recycle the useless certificates. Let us assume again the example of Fig. 3, where node A has communicated enough times with node F to safely assign it a trust value. In this case, there is no need for A to store the certificates that B and C have published for F. If we assume that A marginally trusts F, then A's keyring will look like the one shown in Fig. 4. The dotted lines indicate the certificates that are valueless to node A.

As far as bandwidth is concerned, exchanging certificates only between trusted peers does not necessarily guarantee reduced consumption compared to the Capkun's flooding model. To achieve this many parameters have to be taken into account such as the number of hops to the trusted peers, the frequency of exchanges, the number of common trusted peers, etc. We assume that the trust level between two nodes is bidirectional. Thus, if a node X trusts marginally another node Y, then Y also trusts X marginally. This assumption will help with the reduction of bandwidth consumed by certificate flooding. Such bidirectional trust relationships (which are the most probable scenario according to the real life relationships)

will lead to the formation of trusted nodes groups, similar to the real life groups of friends. Nodes belonging to the same group will most probably have common introducers, and thus common keyrings. The percentage of common introducers depends on the values that each node has assigned to the parameters COMPLETES_NEEDED, MARGINALS_NEEDED and Trust Depth. The smaller the deviation in these parameters, the more common introducers the grouped nodes will have. This is not far from reality: PGP users on the Internet usually set these parameters as COMPLETES_NEEDED=1, MARGINALS_NEEDED=2 and Trust Depth=2. In this case, it is not imperative that a node exchanges certificates with each one of its trusted peers. Exchanging with some of them – preferably the closest ones – is enough for a node to form a trust graph that provides a capable pathfinder. In this case, a mechanism is needed to decide which will be the preferred trusted peers. Our future work is to examine all these parameters towards a bandwidth-efficient model.

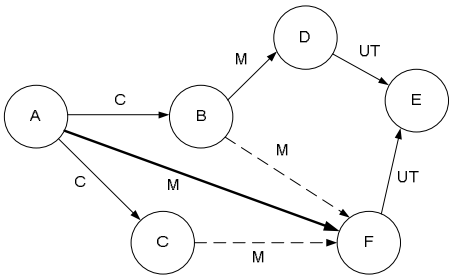

Figure 4. The keyring of node A

Evaluating the proposed authentication model, we can deduce that: (i) it is a fully distributed solution since every node participates equally in the provided authentication services, (ii) it does not require high storage (and possible bandwidth) capabilities, and (iii) it follows the web of trust/free trust policy that suits well to the community character of AWMN. However, the proposed model encounters some problems when applied to a newly formed network. This is because in a new topology it will take some time to build up trusted relationships among nodes. Thus, during the first phase of negotiations, the local repositories of all the involved nodes will hold very few key certificates. Because of this, it is very likely that there will be no certificate chain connecting two nodes who try to authenticate and communicate to each other for the first time. On the contrary, when applied to a network that has been operational for more than five years like AWMN, no time for the establishment of relationships is required, since trust has been already built up. A question arises regarding a newly introduced node. In this case, if the new node personally knows some the existing nodes of the network, then the procedure described earlier can be followed. Otherwise, the new node can rely on the keyrings of its neighbors until its own trust policy is formed. Of course this fact includes the risk of initially trusting a malicious neighbour.

V. CONCLUSION

AWMN is today one of the largest wireless network communities on Earth. Because of the community based, educational and open source character of the network; AWMN is vulnerable to a large set of attacks, such as passive eavesdropping, authentication-deauthentication, impersonation, man-in-the-middle, etc. To address these security concerns, this paper has studied the application of specific security mechanisms in AWMN that aim at providing authentication services. Moreover, it has evaluated the effectiveness of these mechanisms pointing out their advantages as well as the potential drawbacks. To overcome the weaknesses of the studied mechanisms, we have proposed and highlighted an end-to-end authentication model that combines the web of trust and free trust policy of the PGP approach with the non-infrastructure solution of the self-organized public-key management scheme. The proposed model satisfies the following design goals that suits well to the community character of AWMN: (i) it is a fully distributed solution since every node participates equally in the provided authentication services, (ii) it does not require high storage (and possible bandwidth) capabilities, and (iii) it follows the web of trust/free trust policy.

ACKNOWLEDGMENT

This work has been supported in part by the project CASCADAS (IST- 027807) funded by the FET Program of the European Commission and the project NoE CONTENT (IST-384239).

REFERENCES

[1] AWMN Wireless Node Database: http://wind.awmn.net/

[2] [RFC1771] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, March 1995.

[3] A. Abdul-Rahman, "The PGP trust model," EDI-Forum: the Journal of Electronic Commerce, 10(3):27–31, 1997.

[4] S. Capkun, L. Buttyan and J.-P. Hubaux, "Self-organized Public-Key Management for Mobile Ad Hoc Networks," IEEE Transactions on Mobile Computing, vol. 2, no 1, pp. 52-64. January 2003.

[5] T. Bass and R. Robichaux, "Defense-in-depth revisited: Qualitative Risk Analysis Methodology for complex network-centric operations," IEEE MILCOM 2001.

[6] K. Zarifis, D. Ztoupis and C. Xenakis, "Security Issues in an Established Autonomous Wireless Network," PhD poster presented in the 6th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2007), Corfu, Greece, June 2007

[7] A. A. Pirzada and C. McDonald, "Kerberos Assisted Authentication in Mobile Ad hoc Networks," Proc. 27th Australasian Computer Science Conference (ACSC'04), vol. 26, pp. 41-46, 2004.

[8] L. Zhou and Z. J. Haas, "Securing ad hoc networks," IEEE Network Magazine, vol. 13, no. 6, pp. 24–30, November/December 1999.

[9] S. Yi and R. Kravets, "Moca: Mobile certificate authority for wireless ad hoc networks," 2nd Annual PKI Research Workshop, Gaithersburg, Maryland, April 2003.

[10] H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang, "Self-Securing Ad Hoc Wireless Networks," in proceedings of the 2002 IEEE Symposium on Computers and Communications, July 2002..