International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:2, No:6, 2008

Pushing the Limits of Address Based Authentication: How to Avoid MAC Address Spoofing in Wireless LANs

Kemal Bicakci, and Yusuf Uzunay

***Abstract*—**It is well-known that in wireless local area networks, authenticating nodes by their MAC addresses is not secure since it is very easy for an attacker to learn one of the authorized addresses and change his MAC address accordingly. In this paper, in order to prevent MAC address spoofing attacks, we propose to use dynamically changing MAC addresses and make each address usable for only one session. The scheme we propose does not require any change in 802.11 protocols and incurs only a small performance overhead. One of the nice features of our new scheme is that no third party can link different communication sessions of the same user by monitoring MAC addresses therefore our scheme is preferable also with respect to user privacy.

***Keywords*—**Authentication, MAC address spoofing, security, wireless networks.

I. Introduction

AUTHENTICATION is an essential security service for modern computer and communication systems. One of the widely-used authentication techniques is address-based authentication which assumes that the identity of source could be inferred based on the network address from which packets arrive [1]. This network address could be either layer 3 address (IP address) or layer 2 address (MAC address). Though it is traditionally believed to be weak, we still see many examples in modern applications either as a counter-act to risks coming from less-sophisticated attackers or as an additional layer of defense to form multifactor authentication for improved security.

To explain it in simple words, the weakness comes from the easiness of network address impersonation. For instance, the availability of tools for IP spoofing attacks makes the UNIX "r" commands a very poor choice. However in a more careful treatment, we see that the strength of security provided by address-based authentication is based on the environment where it is implemented. For instance, in a switched LAN topology where each node has a point-to-point link to a central switch that is configured with the MAC address of the node in each link, there are two inherent properties that make network address impersonation more difficult. First, learning authorized MAC addresses through monitoring the network traffic is not easy. Second, even when the attacker learns the MAC, he needs to have a physical access to the port that MAC address is registered to. Otherwise the port security mechanism of the switch refuses to forward the packets.

On the other hand, because of the broadcast nature of communication in wireless local area networks (WLANs), it becomes easy for the attacker to masquerading as an authorized user through MAC address spoofing: As a first step, the attacker, using packet-capturing software, sniffs the network traffic and learns one of the authorized MAC addresses. Note that MAC addresses appear in clear even when encryption of data is enabled. Second step is as easy as altering the MAC address of his wireless network interface card with the authorized one. Although this basic vulnerability is well-known, MAC address authentication is widely used to decide on permitting or denying access to the wireless network.

In this paper, our objective is to improve the security provided with the MAC address authentication. More precisely stated, we propose a novel technique for WLANs to avoid impersonation attacks through MAC address spoofing.

The rest of the paper is organized as follows. Section II gives background information on WLANs. Section III explains the operation of MAC address authentication in its original form. In section IV, we propose our new scheme. Section V discusses various issues about the new scheme including its security analysis. Section VI summarizes the related work and section VII concludes the paper and gives directions for future work.

II. Background on WLANs

In this section, we provide background information on wireless network topology, wireless network security requirements, the authentication process and authentication mechanisms in WLANs.

A. Wireless Network Topology

In wireless networks, there are two modes of operation: adhoc mode and infrastructure mode. In adhoc mode, the wireless nodes communicate directly with each other without any additional network elements. Despite the low cost and plug-and-play convenience of adhoc networks, they are not widely deployed yet. Our focus of discussion in this paper is on the more common case of infrastructure mode in which

Kemal Bicakci is Assistant Professor in TOBB University of Economics and Technology, Electrical and Electronics Engineering Department, Ankara, Turkey (e-mail: bicakci@etu.edu.tr).

Yusuf Uzunay is a Ph.D student in Informatics Institute, Middle East Technical University, Ankara, Turkey (e-mail: yuzunay@ii.metu.edu.tr).

nodes communicate through a central station (access point). The access point (AP) also serves as the gateway to access other networks such as the Internet.

In a wider deployment, there may be more than one AP physically distributed. In this case, to ease the network management and to centralize the authentication decisions a central authentication server (e.g. RADIUS server) might be in place.

AP-s transmit beacon management frames at fixed intervals. Upon receipt of a frame, a node can start communication usually by sending an authentication request frame. In a properly designed network, when an access to the wired local network is granted from the wireless network, additional protection such as a firewall is enabled between the AP-s and the local network.

B. Wireless Network Security Requirements

Providing security has utmost importance in the design and implementation of wireless local area networks. As far as security is concerned, one of the most important requirements is authentication of the nodes when they contact with the access point and request an access. Access control is achieved easily after authentication by referring to an access control list.

In wireless LANs, other than authentication and access control, confidentiality and integrity of the data traveling over the wireless link also needs to be assured. This assurance is usually provided by cryptographic means. It is also important that availability of the wireless network to its authorized users should be preserved by preventing denial of service (DoS) attacks. The security as well as the efficiency of the authentication mechanism in place has also a big role against DoS attacks. The last but not the least, for security auditing which involves recording and analyzing security-relevant activities, source authentication of audit records is necessary. We will focus on authentication in this paper.

C. Authentication Process in WLANs

Regardless of the authentication method in use, the client always starts the communication with the AP by sending an authentication frame (Fig. 1). The type of authentication in use might be even “open authentication”, which simply means a null authentication algorithm. If this is the case, the access point responds back immediately with the authentication response message of type “success”. Depending on the authentication mechanism in use, additional exchange of messages should be performed before the access point sends the “authentication success” message. If an authentication server is used, in the meantime exchange of messages is also seen between the AP and the server.

Since there may be more than one AP in the range of client, it may receive an authentication response from more than one AP at the same time. In order to choose one of them, the client sends an association frame to the chosen one which responds back with the association response frame. Only after establishing authentication and association, the actual exchange of data between two parties can start. This communication continues until either the client or the AP sends a deauthentication frame. Optionally, a disassociation frame to the other party can be sent requesting to return back to authenticated-unassociated state. Both deauthentication and deassociation request messages are answered with response messages by the other party.

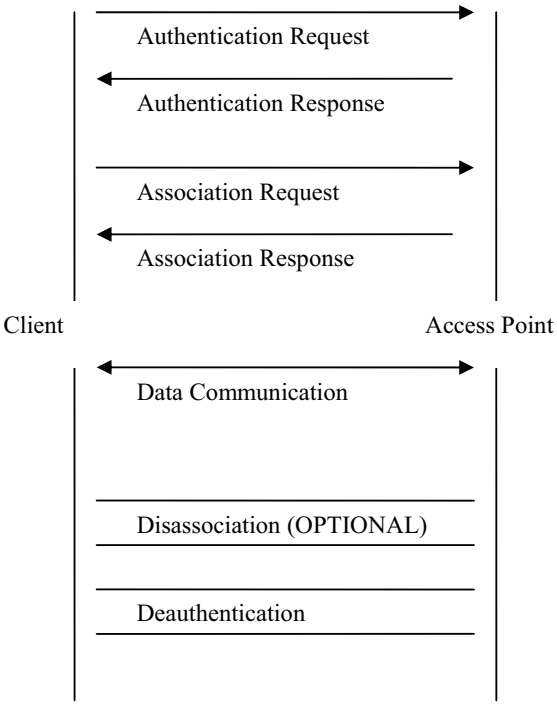

Fig. 1 A typical frame exchange between the client and access point

In other words, the communication between the client and access point can be in one of the three states: initial state, authenticated-unassociated state, and authenticated-associated state. This is depicted in Fig. 2. Note that the access point forwards frames coming from the client only in authenticated-associated state.

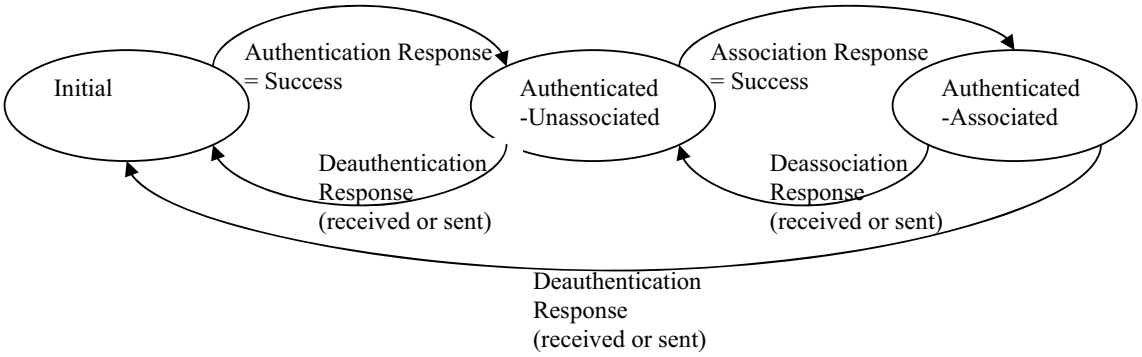

Fig. 2 Connection states of the wireless communication in 802.11 standard

D. Authentication Mechanisms in WLANs

In this section, we do not go over the details of the various security standards such as WEP, WPA or 802.1X or details of the attacks because of the protocol weaknesses. Instead, leaving the discussion of address-based authentication to the next section, we will list various authentication solutions currently available for WLANs and possible attacks to these as follows:

Mutual versus client-side only authentication: If the AP is not authenticated, an attacker can set up a fake AP near his victim and can steal his authentication credentials such as his password (man in the middle attacks).

Device authentication versus user authentication: If the authentication is based on secret information stored on the device, then this scheme is of limited use when the device is used by multiple users. This is avoided by authenticating the user instead.

Authentication protocol versus authenticated-key exchange protocol: If the protocol in place only authenticates the nodes initially without establishing a shared secret that is used to encrypt the traffic thereafter, "connection hijacking" attacks become possible.

Cleartext versus encrypted transfer of authentication credentials: If authentication credentials such as passwords are sent over the network in clear without encryption, then the mechanism is vulnerable to interception of secret credentials.

Password versus high-entropy bit strings: If the authentication is based on a low-entropy (weak) password, then this can be broken with brute force or dictionary attacks even when the communication is encrypted.

Susceptible to offline dictionary attacks or not: More advanced authentication protocols (strong password protocols [1]) make it impossible for the attacker to perform an offline dictionary attack (i.e. without contacting the access point) after intercepting the authentication frames exchanged.

Secret-key versus public-key based protocol: When the authentication between the client (authenticating node) and the AP is based on a secret key, then the secret key can be stolen by reading the secret file stored on AP. Client certificates do not have this kind of vulnerability. (Note that a similar argument can be made for the authentication between the AP and the authentication server)

Using a smartcard versus storing the private key inside the node: If the private key is inserted into a smartcard, this is more secure since it is a more difficult task to steal the private key from the smartcard than the laptop especially when the smartcard has tamper-resistance properties.

Layer 2 authentication versus Higher layer authentication: The authentication mechanisms available in wireless standards are implemented at layer 2 of the protocol stack. But one can simply not use this and instead rely on a VPN tunnel (layer 3) or SSL protocol (layer 4) to set up an authenticated link between two parties.

Authentication at the access point versus using another server: If a third-party authentication server such as a RADIUS server is used, then this server also requires security protection.

III. MAC Address Authentication

Despite the many possibilities for authentication in WLANs as we saw in the previous section, unfortunately, majority of these networks are still totally unprotected. This is mostly a human error rather than a technology issue i.e. most manufacturer turns the security features off by default because it makes the networks easier to set up. The users by default do not take the trouble to read the manuals and activate the security settings hence most wireless equipments are completely insecure from the moment it comes out of the box.

MAC address authentication is no different than others in this sense since most of the time it is also not enabled by default and needs to be configured manually. However, once it is set-up, MAC authentication has the following advantages:

User-friendliness: It is more user-friendly because when used alone after initial registration of the MAC address, the authentication is totally transparent to the user. For instance, s/he does not need to type in a password, carry a smartcard, etc.

High-availability: Although, it is not specified in the 802.11 standard, most vendors support MAC authentication.

Therefore, as we said earlier, even when other authentication mechanisms are in place, MAC authentication can be used to augment them.

Efficiency: MAC address authentication has efficiency advantage over other alternatives. Since no cryptographic algorithm is involved, it has little performance overhead compared to default settings.

We will explain the structure of a MAC address, the working principle of MAC address authentication and its security vulnerabilities in the next three subsections, respectively.

A. Structure of a MAC Address

IEEE 802.11 wireless networks use the MAC-48 identifier format. In this format, MAC address space is 48-bits and has a flat structure that means no "host" and "network" portions are allocated. The addresses are usually shown in a hexadecimal format. An example could be "00-02-44-65-3A-DF".

B. Working Principle

The working principle of MAC authentication changes with respect to whether it is used alone (Mode 1: open authentication) or as a way to augment other authentication methods (Mode 2: two-factor authentication). Below, we assume that AP forwards the requests from the clients to an authentication server which checks the requests centrally.

1) Using Only MAC Authentication

In this mode, MAC authentication works in two phases: (a) registration (b) operation

a) Registration:

Let A be the list of MAC addresses that are already registered and authorized. This list is stored on the authentication server. When, a new user needs to be registered and authorized to have access to the WLAN, user's identity is verified and MAC address of his device is added to the list A.

This registration can be carried out either offline or online. For instance in a university environment, it might be feasible for a student to fill out a paper based registration form where s/he writes down the MAC address of his/her laptop and return it to the computer center of the university. Then the university personnel can verify the student's identity by checking student ID and update the list accordingly. This same procedure can also be realized online with a MAC address registration web site. Identity verification is also necessary here and might be implemented in various ways for instance if the student already has a university mail login name and password, he can enter this information together with the MAC address of his machine. To avoid someone capturing the MAC address or the password, the channel between the student's machine and the server should be secured typically by using the SSL protocol.

Let set *A* holds the authorized addresses; MAC address *1* to MAC address *n-1* as A = $\{MAC_1 , MAC_2 , MAC_3 , \dots , MAC_{n-1}\}$. Let us denote the new MAC address as MAC_n. More formally stated, the list *A* is updated as $A = A + \{MAC_n\}$

b) Operation:

Since MAC authentication is not specified as part of the 802.11 standard, we have a variety of options to implement it. Here, we prefer to explain the one implemented in Cisco and various other companies products [2].

Upon receipt of a beacon from an AP, the wireless node sends its authentication frame. Upon receipt of this frame, the AP simply sends a response frame of type success because open authentication is used. Then, the node sends its association request frame. At this point, the AP asks to the authentication server whether the source MAC address on the frame is listed in its authorized address list. If it is, upon receipt of an accept response from the server, it responds to the node with an association success frame and switch to the authenticated-associated state. Otherwise the node is rejected by sending a failure type of association response.

Let *A* be the authorized MAC address list. The simple procedure to give the access permission decision is depicted in Fig. 3 (only the relevant part is given). A practical guide for deployment of MAC address authentication on FreeRadius server [3] is given in [4].

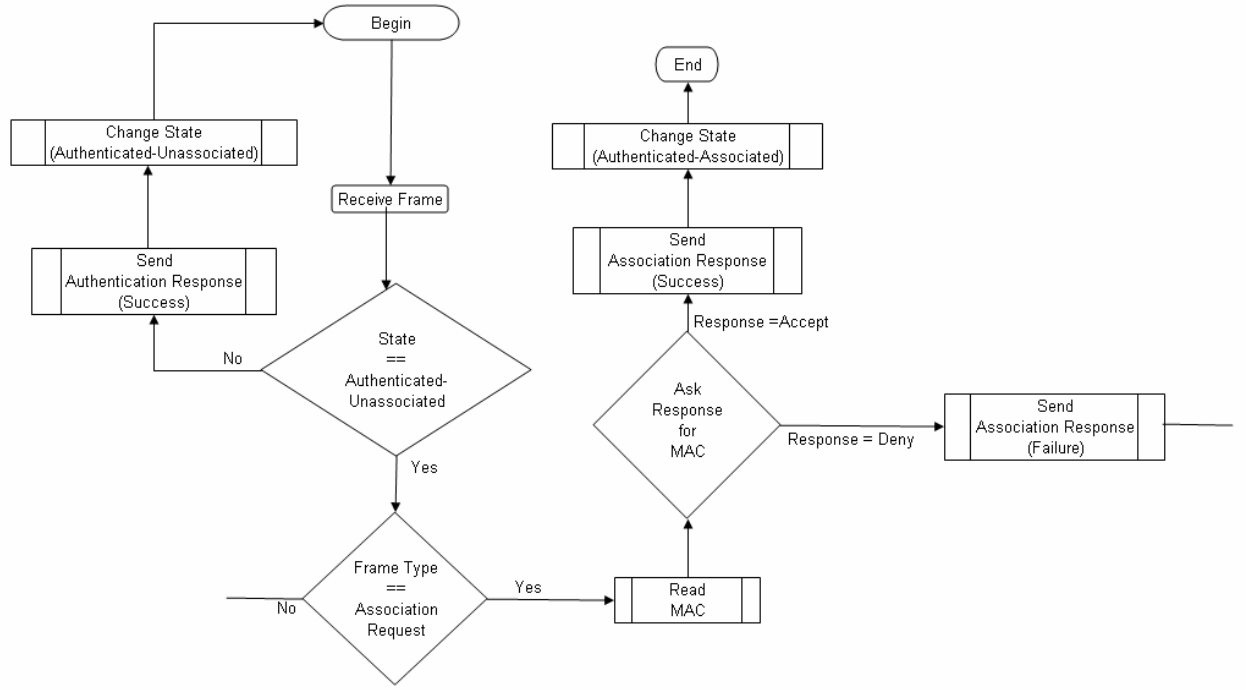

Fig. 3 Operation of MAC authentication in Mode 1

2) MAC Authentication Augmenting Other Methods

In this case, registration phase is as same as Mode 1 but the procedure to give the access permission decision is expanded. For instance in case when there is a challenge-response authentication based on a shared key (as in WEP), response to an authentication request would be of type challenge (not type success). The client returns back an encrypted challenge and only if this encrypted challenge is decrypted successfully by the server, the state is changed to authenticated-unassociated and a success message is sent. The rest of the operation is similar to the one seen in Fig. 3.

C. Security Vulnerabilities

In the introduction, we have already mentioned how easy for the attacker to listen to the wireless traffic and learn one of the authorized MAC addresses. The exact impact of this security vulnerability, however, depends on the mode of operation in use.

In mode 1 where the authentication is solely based on MAC addresses, the risks associated with MAC authentication are more severe. This is because

(1) attacks can be less sophisticated

(2) we have limited countermeasures available against these attacks.

Below, we will elaborate more on this difference.

To bypass the MAC authentication, the attacker first starts capturing the network traffic on his wireless network card with tools like tcpdump [5]. Then, from the captured traffic, he collects source MAC addresses. In order to avoid possible protection mechanisms described in the following paragraphs, optionally he waits until one of the nodes leaves the network (e.g. a deauthentication frame has been seen). Then he changes the MAC address of his wireless card with the new MAC address and sends a frame having this address to the access point. (He also uses the recorded IP address and default gateway values [6]).

Now, let us see what is different when two-factor authentication is in place. Similarly, the attacker needs to capture the traffic and learns some authorized MAC addresses first. However in this case it is not possible for him to wait till one of the authorized users quits. Because when a user quits, his communication returns back to the initial state where no frame is forwarded by the access point. To change the state back again to authenticated-associated, the attacker needs to authenticate himself successfully. Since the attacker does not hold the credentials for this authentication, the MAC address he has captured is useless after the authorized user quits the session.

Therefore, the attacker should act while the authorized user continues communication with the access point. In [6], it was demonstrated how the attacker can launch a DoS attack against the authorized user and cause his machine to crash. Before the crashed machine boots up, for a short period of time, the attacker has the capability to bypass the authentication by changing his MAC address with the MAC of the crashed machine and impersonates himself to the access point. The procedure explained here is actually one instance of connection hijacking attacks mentioned in item 3 of subsection II.D. To differentiate this attack from the simpler one based solely on passive listening, we limit the usage of the term “MAC address spoofing attack” for the later.

As far as mode 2 is concerned, there are a variety of options available to deal with this attack. First of all, as explained in section 2.3, if the authentication enables authenticated key

exchange to exchange a key used to encrypt the rest of the communication, then this attack is totally avoided since the shared secret is not known by the attacker.

The second way to avoid the attack is based on sequence number field of 802.11 frame headers. The sequence number field is 12-bits long and incremented by one for each non-fragmented frame. Although the attacker has the ability to change the MAC address, same thing is not true for the sequence number. Without the ability to access the firmware source code of the wireless card, the attacker can not alter the sequence number to an arbitrary value [6]. Hence, when the attacker hijacks the authorized connection, the frame he sends would not have a sequence number incremented by one. For instance the last frame the authorized user sends might have a sequence number of 433 whereas the frame the attacker sends very likely might have a sequence number something other than 434. This anomaly is the hint for the AP to recognize that there is something wrong. Analyzing the sequence number pattern, the access point can identify and mark the activity from the concerned MAC address as the spoofed MAC activity.

Can we do the same kind of sequence number analysis for mode 1? The answer is unfortunately no. Remember that the attacker instantiates a new connection to the AP in this case. The only thing the AP can do here is to store the sequence number of the previous session and match it with the sequence number of the first frame in the new connection. However the probability that the wireless card has sent frames to another AP in the meantime is not negligible therefore calling the gap in the sequence numbers a spoofed MAC activity carries the significant risk of being a false positive. (One might think that to avoid false positives, the sequence number analysis mentioned above can also be done centrally on the authentication server but this is not a practical solution since all AP-s now should continuously inform the server about the sequence numbers of the frames they receive.)

IV. Our Proposal

As we discussed in the previous section, it is technically very straightforward to launch a successful attack when MAC address authentication is not augmented with a second method. Hence our principal aim is to improve the security of this more user-friendly but less secure alternative. In other words, we will try to push the limits of address based authentication when used alone.

A. Our Solution in a Nutshell

Let us first think on the main reason of insecurity. The reason is the third parties' easy access to the secret information that is the MAC address itself. In our application scenario, it is obvious that using encryption to protect the MAC address is not an option. If you cannot keep secure what you use as the secret authentication credential, then there is only one alternative left. Make the secret only one-time usable, so if somebody captures it, s/he can not use it for a second time. This well-known idea has already been implemented in one-time password schemes [7].

After setting this perspective, we infer that the solution should have the following security properties: Each MAC address should be used for only one session. In other words, when a node establishes a new connection, it should not use one of the MAC addresses previously used. The authentication server should be able to securely verify the MAC addresses it receives as being authorized or not. In other words, it should reject MAC addresses generated by attackers.

The attacker should not be able to generate new authorized MAC addresses from the exhausted MAC addresses (at least in a feasible amount of time). We have a very crucial additional requirement not related to security. The solution proposed should not impose a change in the 802.11 standard. Any change in the IEEE standard (frame format, field sizes, etc.) will not be acceptable from a user point of view.

B. The Design

Now, it is the time to consider the design details. Due to the requirement of conforming to 802.11 standards (the size of the MAC addresses field can not exceed 48 bits), we have seen that hash chain [8] based solutions are not secure enough (can offer only 48-bits of security). For the sake of brevity, we skip the details why this is so. Hash-chain based solutions have the security advantage over secret-key alternatives in their capability to make the server free of any secret and hold only public information to verify the secrets. Considering that the attacks based on listening to the traffic are far more common in wireless scenarios, in our solution we assume that the servers in the system can be kept secure so that the secret information cannot be stolen from them.

The only crypto primitive used in our design will be the one-way function H() (e.g., MD5 or SHA-1). When the length of hash output is longer than the desired length, the output is truncated. Latest collision attacks to these functions are not relevant in our case, since collision resistance is not required for secure operation. The only security concern is the one-wayness property of these hash functions.

In our new design, MAC authentication again works in two steps: registration and operation.

1) Registration:

This is the step where first user identity is verified as usual either in an online or offline fashion. However the difference is that instead of a 48-bit MAC address, two parties agree on a random number (seed value) that has at least 128 bits for the reasons that will be explained in the next section. Let list S holds seed values already shared between the authentication server and users; seed 1 to seed n-1 as S = {S1 , S2 , S3 , … , Sn-1}. Let us denote the new seed value as Sn. More formally stated, the list S is updated as S = S + {Sn}.

2) Operation:

Since MAC addresses are not static in our solution, the user machine has to execute a short procedure given in Fig. 4 before initializing a new connection (or after receiving a response of type failure, see section VII). This can be performed automatically with a short script running on client's machine without user's intervention. Remember that with almost all wireless cards it is very straightforward to alter the MAC address to an arbitrary value. Here, the new MAC address value is the first 48-bits of the seed value. The new seed value is computed by hashing the old seed value (i.e. New Seed=H(Old Seed)). If the first 48 bits of the new seed

can not be used as a MAC address for some reason (e.g. if it is all 1's which is reserved for broadcasting), then the hashing operation is performed again.

On the AP side, nothing needs to be changed in our design and the procedure given in Fig. 3 is as same. However changes are required on the authentication server. This is illustrated in Fig. 5. Traditionally the authorized MAC addresses are written into a configuration file on the authentication server. For instance in FreeRadius [3], "users" file is used to hold the information. One entry in this file might be

aabbcc001122 Auth-Type := Local, Password == "aabbcc001122"

In the original MAC authentication, after lookup operation if it is seen that the MAC address received from the AP matches with one of the entries, the server sends back a response of type success. In our new scheme, the information that is stored for each client is modified. For instance one entry would look like

aabbcc001122aabbcc001122aabbcc00 Auth-Type := Local, Password == "aabbcc001122"

Note that the first field (username field) is now 128-bit long instead of 48-bit and the password field holds the first 48-bits of this field. When the MAC address is received, it is again matched with the entries in the file. If a correct match with the "Password" field in any entry is found, again a response of type success is sent back. In addition, in the new scheme the first field in the matched entry is overwritten with the hash value of the old value. Then, the password field is changed with the first 48-bits of the new computed value. So after authenticating the MAC address "aabbcc001122" the entry above is modified as follows:

b731d2b56befa4409f77cccbc0326261 Auth-Type := Local, Password == "b731d2b56bef"

Note that the update operation is only performed if the sent response is of type success.

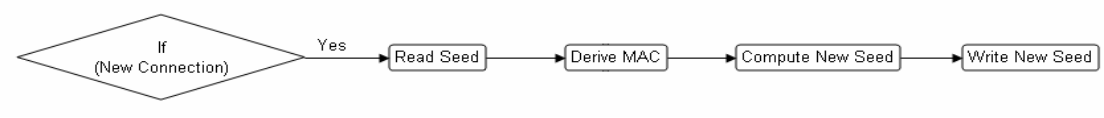

Fig. 4 Procedure for the client machine

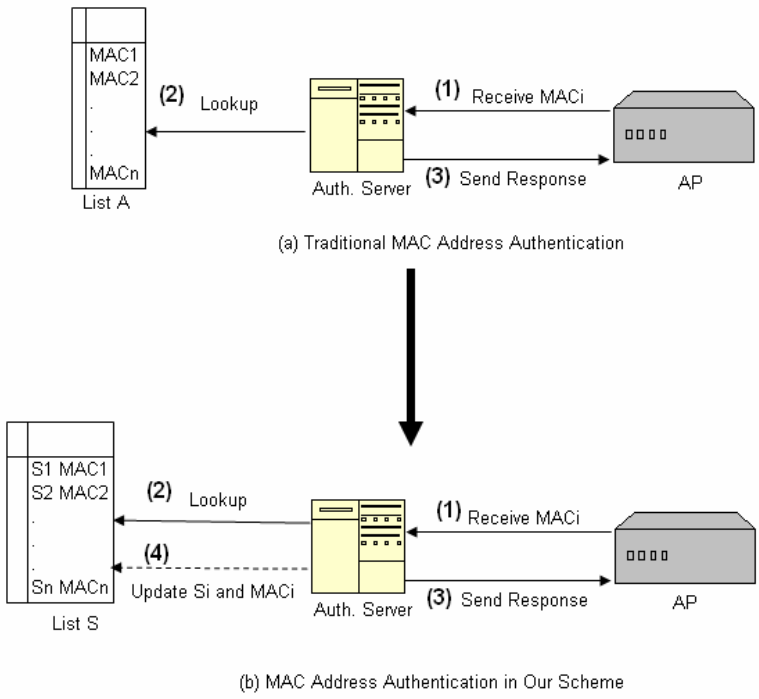

Fig. 5 Comparison of MAC address authentication procedures

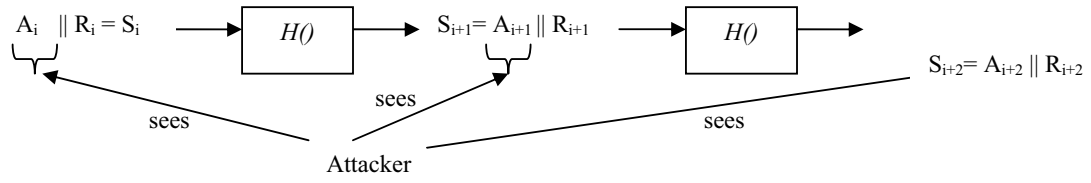

Fig. 6 Illustration of cryptographic computations and the attacker's position

V. Discussion

In this section, we make the security analysis of the new scheme and discuss performance issues. We also explore possibilities to use the proposed scheme in other applications. Finally, we argue that the new scheme has another important advantage with respect to "user privacy".

A. Security Analysis

Broadly speaking, we can talk about two kinds of security: unconditional security and computational security. In practice, the later is usually sufficient that means the security can not be broken in a feasible amount of time using modest resources. That is why the security level can be expressed as the number of cryptographic computation (e.g. hash computation) on average required to break the scheme. For instance the security level 280 means that the attacker needs to perform 280 computations on average for breaking. This level is believed to offer reasonable security for civilian applications therefore our aim in this section is to show you that the scheme proposed can achieve this level of security.

In this analysis, we assume that the attacker can intercept the network traffic at any time but can not read the secret files stored in the server. We also assume that the hash function used to generate MAC addresses is secure and has the one-way property. That means given the hash output or part of the hash output, there is no shortcut to recover the hash input (no way other than brute force attack).

As illustrated in Fig. 6, the attacker can see the first 48 bits of every hash input and hash output (|| means concatenation). If the attacker is able to learn either one of the Si values, then he can generate valid MAC addresses thereafter and succeed in impersonation. However since there is not a shortcut method, the only thing the attacker can do is brute force i.e. trying each combination for Ri one by one.

There is one important point, here. Given Ai and Ai+1 values, there is "not" only a single pair Ri and Ri+1 satisfying the equality Ai+1 || Ri+1 = Hash (Ai || Ri) and the attacker cannot break the security by finding one of these pairs because the next address Ai+2 to be accepted by the access point can only be generated by the correct Si+1= Ai+1 || Ri+1. If any pair was acceptable, the attacker would break the scheme only by 248 computations on average. Among all Ri and Ri+1 values satisfying the equality Ai+1 || Ri+1 = Hash (Ai || Ri), there is only a single pair which is usable to generate the next correct MAC address and the number of computation required to find out that pair is proportional to the length of bit string Ri. In other words, to have a security level of 280, Ri should have a length of 80 bits which corresponds to the total length of 128 bits for Si values.

We note that the security level of 80 bits is achieved only for offline attacks. If online attacks can be realizable, an attacker simply tries random elements in the 48 bit address space. Depending on the number of users in the system, using this brute force online attack, an attacker might impersonate himself easier than an offline attack. This attack has also DoS implications and can be defended by non- cryptographic means.

Another related problem is the possibility of two nodes sharing the same MAC address. However, in the appendix we show that this probability is very small and therefore can be ignored.

B. Performance Analysis

Recall that one of the advantages of MAC address authentication is its high performance. It might even be the only alternative for some small wireless devices such as barcode readers which cannot perform complex cryptographic computations.

The scheme we have proposed brings only a small overhead as compared to original MAC authentication since cryptographic hash computation can be carried out very efficiently for instance using off-the-shelf laptop computers, it takes only a few microseconds. Implementing hash algorithms on small devices is also much easier than public key crypto primitives. In the scheme we have proposed, the clients can generate MAC addresses totally offline. It is even possible to prepare a list of MAC addresses externally and load it to the device periodically.

On the server side, other than hash computation the second performance penalty is due to update operation. Authorized MAC addresses should be updated dynamically in our scheme.

C. Applications

Our novel technique to safeguard against MAC spoofing attacks is a generic one and can be tailored to improve the security of address based authentication in other applications. One particular application might be IEEE 802.15.4 wireless personal area networks which use 64-bit EUI-64 identifiers. In this case, it is possible to construct a hash chain with the output length of 64 bits to make the communicating partner free of any secret while having a reasonable security level of 264. Note that our scheme is more appropriate for star topology where there is a PAN coordinator whose role is similar to the access point in WLANs.

If the WPAN has a point to point topology or there is more than one access point in a WLAN which do not ask to the authentication server for the authorization, then implementing our scheme is more difficult and it requires real-time or near real-time synchronization of the authorized address list since the list is updated dynamically in our case. Our technique is more appropriate for small networks with one access point (or PAN coordinator) or bigger ones having a central authentication server.

D. User Privacy

One of the most exiting security properties of the proposed scheme is its capability to enhance user privacy. As we have already mentioned, standard 802.11 networks use MAC addresses as static node identifiers and even when the communication is encrypted, addresses remain in the clear therefore statistical traffic analysis and identification of users is possible.

On the other hand, if our proposed scheme is used, no third party can link different instances of MAC addresses generated by the same node. Identifying the source of packets traveling over the network by their MAC addresses is infeasible and as a result user privacy is improved.

The observant reader might have already observed that the same argument for unlinkable MAC addresses would not be valid if MAC addresses were generated with the hash chaining idea.

VI. Related Work

Security vulnerabilities associated with MAC address authentication is well-known. Applications for sniffing are freely available from the Internet (e.g., tcpdump [5], ethereal [9]). Attackers can then simply change their own MAC address to be that of an authorized node.

Previous work has concentrated mostly on MAC spoofing detection, which mainly uses the sequence number tracking technique [6], [10], [11]. But as we have explained earlier, this solution has limited applicability and might result in too many false positives.

Up to our best knowledge, our study is the first that tries to improve the security of address-based authentication in wireless applications by "preventing" MAC address spoofing.

In our previous work, we showed that address authentication using static MAC addresses might provide reasonable level of security in wired scenarios when advanced switches with the port security mechanism is used [12].

VII. Conclusion and Future Work

In this paper, we put forward a novel idea to strengthen the security provided with MAC address authentication in wireless local area networks. In the proposed scheme, each MAC address is usable for only one session and the MAC address to be used in the next session can not be computed from previous ones by the third parties. No protocol change is required in our new scheme which poses only little computational overhead.

Our work in progress is an ongoing research and development work and we need to solve a few practical problems before fully implement the new scheme. First of these is the lost frame problem. Suppose a node has updated its MAC address before sending its frame and the frame is lost and does not reach to the AP. Then, the update procedure should not be executed again and the old MAC address should be reused because the AP has not updated its database yet. A similar problem can be seen when in the meantime the node is connected to an AP which is not managed by the authentication server. To overcome the limitations posed by these frame synchronization problems, an alternative solution is to change MAC addresses using the clock as an input not the previous MAC address. Of course this is viable only after time synchronization between the authentication server and all nodes is achieved.

While the scheme proposed can prevent MAC spoofing attacks, for connection hijacking attacks where the attacker steals one of the already established connections, detection mechanisms based on sequence number analysis are useful. Therefore these two techniques complement each other for a more secure MAC address authentication.

802.11 wireless networks as deployed today have other security vulnerabilities one of which is the susceptibility of denial-of-service attacks due to lack of authentication in the deauthentication and deassociation frames. In [13], practicality of these attacks and possible countermeasures are described. In fact, the vulnerability is self-evident. If deauthentication frames are unauthenticated, by spoofing the victim's MAC address or the access point's MAC address anybody can send a deauthentication frame to the other party to exit the authenticated-associated state. Moreover, this attack can be repeated to block the victim's network access permanently. As a future work, it is promising to investigate on extensions of the new scheme in order to safeguard against this vulnerability.

Appendix

Probability of MAC Address Collision

Suppose there are two nodes in the system, then the probability of collision is simply 2-48 (to simplify, we do not take into account addresses reserved for particular use e.g. broadcasting etc.). For three nodes, the collision probability is 2-48+(2-48+2-48). In general, for a network with n nodes, the probability can be given as

$$P(C) = 2^{-48}(1+2+3+\cdots+n)$$

$$P(C) = 2^{-48}[n*(n+1)]/2 \approx n^2 * 2^{-49}$$

For instance, if n = 1000, then $P(C) \approx 2^{-29}$ which is a negligible quantity.

References

[1] C. Kaufman, R. Perlman and M. Speciner, Network Security Private Communication in a Public World, *Prentice Hall, Second Edition*, 2002.
[2] Wireless LAN Security Paper, available http://www.cisco.com/warp/public/cc/pd/witc/ao1200ap/prodlit/wswpf_wp.pdf, 2002.
[3] FreeRadius, http://www.freeradius.org/

[4] G. Me, Deployment of MAC Address Authentication based on Freeradius,
available http://www.wi-fitechnology.com/Papers+req-showcontent-id-1.html
[5] Tcpdump, http://www.tcpdump.org/
[6] J. Wright, Detecting Wireless LAN MAC Address Spoofing, white paper, available at http://www.logisense.com/docs/wlan-mac-spoof.pdf
[7] N. M. Haller, The S/KEY one-time password system. In *Proceedings of the ISOC Symposium on Network and Distributed System Security*, 1994.
[8] L. Lamport, "Password Authentication with Insecure Communication", *Communications of the ACM*, November 1981.
[9] Ethereal: A Network Protocol Analyzer, available at http://www.ethereal.com/
[10] H. Xia and J. Brustoloni. Detecting and Blocking Unauthorized Access in Wi-Fi Networks, in *Proceedings of the Networking'2004 Conference*, IFIP, Athens, Greece, *Lecture Notes in Computer Science*, 3042:795-806, Springer-Verlag, May 2004.
[11] F. Guo and T. Chiueh, ``Sequence Number-Based MAC Address Spoof Detection," in *Proceedings of 8th International Symposium on Recent Advances in Intrusion Detection (RAID 2005),* September 2005.
[12] Y. Uzunay, K. Bicakci: UNIDES: An Efficient Real-Time System to Detect and Block Unauthorized Internet Access. *Proceedings of 11th International Conference on Parallel and Distributed Systems (ICPADS 2005),* IEEE Computer Society, 2005.
[13] J. Bellardo and S. Savage, 802.11 Denial of Service Attacks: Real Vulnerabilities and Practical Solutions, *Proceedings of USENIX Security*, 2003.