On the Way to IEEE 802.11 DoS Resilience

Ivan Martinovic, Frank A. Zdarsky, Jens B. Schmitt

disco lab | Distributed Computer Systems Lab
University of Kaiserslautern, 67655 Kaiserslautern, Germany
{martinovic,zdarsky,jschmitt}@informatik.uni-kl.de

Abstract. In this work we provide an overview of the present IEEE 802.11 security and analyse the roots of unsolved vulnerabilities based on the manipulation of the Medium Access Control (MAC) address and on the fact that management and control frames are not authenticated. These vulnerabilities are the basis for most prominent Denial-of-Service attacks on the IEEE 802.11 link layer. We discuss possible solutions and describe our concept of protection called the *Early MAC Address Binding,* which is a light-weight cryptographic protocol based on the Diffie-Hellman key exchange. Our protection mechanism does not require any changes in the current IEEE 802.11 state machine, nor any additional message exchange. As a result, both stations and access points are protected from the moment when both parties allocate resources for further communication.

Keywords:
WLANs, Security, Denial-of-Service.

1 Motivation

The blind trust in a sender's Medium Access Control (MAC) address and the lack of any authentication mechanism for IEEE 802.11 management and control frames leave no possibility for wireless nodes to verify whether the received frame was sent by its legal sender or injected into the communication by an adversary. In the early days, the IEEE 802.11 link layer parameters defined in network drivers were not easily manipulated. With the high popularity of WLANs the situation has changed and a wide spectrum of different tools simplifying the manipulation of the IEEE 802.11 link layer parameters exists today. As a result, an adversary is able to mount different attacks based on impersonation of stations or access points. The flexibility of these attacks enables the adversary to choose between stations which are already actively communicating with an access point and to selectively attack a single station, as well as to mount different flooding attacks on access points by choosing new MAC addresses.

In 2005, the IEEE 802.11 Task Group *w* (TGw) was established with the aim of creating a standard for authentication of management and control frames with an expected draft due in 2008. In this work we present a mechanism for protection of all IEEE 802.11 frames exchanged in unicast communication between a station and an access point. Our goal is to provide a light-weight security protocol which does not require any changes to the IEEE 802.11 state machine, nor any additional message exchange. In addition, the protection should take place before the access point allocates resources for establishing a connection.

The remainder of this paper is organized as follows: Section 2 provides an overview of the current IEEE 802.11 security based on the recently ratified IEEE 802.11i standard. In the same section we describe different frame classes and analyse the IEEE 802.11 state machine in order to better understand vulnerabilities based on unauthenticated control and management frames. Section 3 briefly describes known DoS attacks as a result of unauthenticated frames. The possible countermeasures against the aforementioned attacks and our protection mechanism called the Early MAC Address Binding are described in Section 4. Section 5 gives an overview of related work concerning attacks based on unauthenticated management and control frames as well as existing protection mechanisms. We conclude this work in Section 6.

2 Current State of IEEE 802.11 Security

In July 2004, the IEEE 802.11 Task Group *i* ratified a new security standard 802.11i with the goal to replace the WEP protocol with all its insecurites. As a result, the new security standard defines the Robust Secure Network (RSN) with a variety of different security services which include: enhanced authentication mechanisms, key management algorithms, cryptographic key establishment, and enhanced data encapsulation mechanisms based on the Advanced Encryption Standard (AES) [3,18]. The devices which do not support IEEE 802.11i RSNs are considered to be a part of Pre-RSN security framework.

There are also two authentication algorithms inherited from the IEEE 802.11 standard [2]: Open System authentication and Shared Key authentication. Open System is equivalent to no authentication, therefore every Open System authentication request will be granted. Shared Key authentication was based on WEP to implement challenge/response message exchange between wireless nodes. With the introduction of IEEE 802.11i RSNs, Shared Key authentication is not allowed anymore, leaving the Open System authentication to be the only mandatory IEEE 802.11 authentication algorithm required by all IEEE 802.11 devices. The reason for this is that the real authentication is now provided by the IEEE 802.1X (EAPOL) [4] protocol as a part of the IEEE 802.11i security standard. In Figure 1 the simplified sequence diagram of the IEEE 802.11 active network discovery, Open System authentication, and association procedure together with IEEE 802.11i standard are depicted. In the context of IEEE 802.1X authentication the station takes on the role of a *supplicant*, the access point that of an *authenticator* which shares a secure channel with an *authentication server* (AS).

The mutual authentication between the supplicant and the authenticator server is carried out by the IEEE 802.1X protocol using the authentication server after which both the supplicant and the authenticator hold a secret key called the Pairwise Master Key (PMK). Upon a successful completion of the 802.1X protocol, the authenticator initiates a 4-Way Handshake which confirms the existence of a supplicant, as well as the existence of a current PMK and derives a

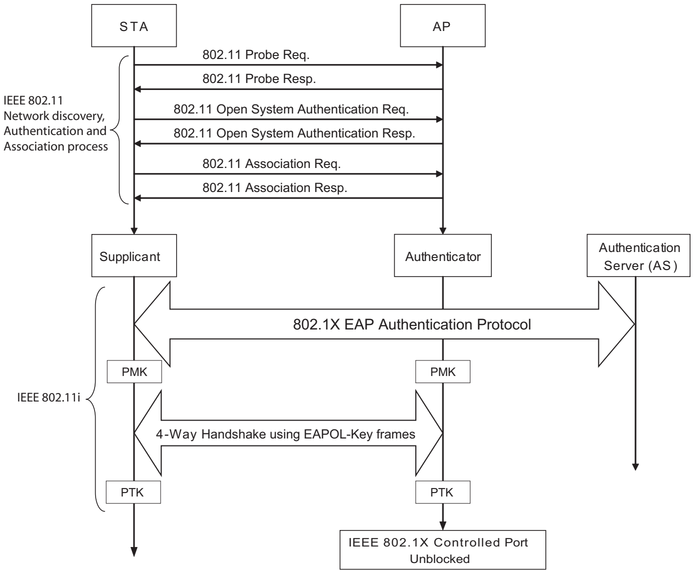

Fig. 1. 802.11i sequence diagramm

fresh Pairweise Transient Key (PTK) used for data authentication, confidentiality and replay protection. After the 4-Way Handshake, the authenticator opens its 802.1X Controlled Port and the unicast data traffic is then protected by means of confidentiality and authentication of the link layer. To simplify the infrastructure required for the implementation of IEEE 802.11i, the standard offers a Pre-Shared Key (PSK) mode of operation which substitutes the IEEE 802.1X EAP authentication protocol with a pre-shared key serving as the PMK and proceeds directly with the 4-Way Handshake.

To analyse the authentication latency of the IEEE 802.11i, we have setup a testbed using an Proxim's Orinoco AP-4000 access point running IEEE 802.11i RSN as authenticator, a notebook running Ubuntu 5.10, kernel ver. 2.6.12 with D-Link DWL-G650 network card using madwifi-ng network driver as a supplicant and for the authentication server we used the FreeRADIUS server v1.1.0. Table 1 shows the results of 10 repetitions of IEEE 802.11i divided in mutual authentication provided by the IEEE 802.1X EAP-TLS and derivation of the PTK with the 4-Way Handshake compared to IEEE 802.11 in PSK mode. As can be seen, the mutual authentication takes most of the time and together with the 4-Way Handshake it takes about 199 ms to complete. On the other side, the utilisation of the PSK dramatically decreases the latency to about 19 ms.

These results show that even if the authentication of all frames could be provided within IEEE 802.11i (e.g. by using PMK), the non negligible latency (183 ms) would still enable the adversary to intercept the message exchange and to attack the IEEE 802.11i protocol itself. As a result, the protection mechanism should be implemented before IEEE 802.11i.

	IEEE 802.11i: 802.1X TLS	IEEE 802.11i: 4-Way Handshake	IEEE 802.11i: Pre-Shared Key mode (PSK)
Mean (ms)	182.6	16.1	18.8
Std. Dev. (ms)	10.3	5.5	4.1

Table 1. Duration of IEEE 802.11i phases

Although the new security standard provides security services that successfully replace the well-known WEP, the authentication of other types of frames used within IEEE 802.11 was not addressed in this standard. To better understand the nature of DoS attacks, we briefly describe the frames and the state machine of IEEE 802.11[12].

Among data frames which are used to transport higher layer protocol data, the IEEE 802.11 standard defines *control* and *management* frames. All frames are divided into three classes (see Table 2), as follows:

- Class 1 control frames provide operations for channel acquisition, positive acknowledgement of received frames and carrier-sensing. Management frames are used for supervisory functions like network discovery and joining or leaving networks. To extend its battery life a station can go into the "sleeping" low-power mode. While the station is in low-power mode, the traffic data is buffered at the access point, which announces it by sending a traffic indication message.
- Class 2 management frames are responsible for starting and ending associations and for supporting mobile stations in moving within the same extended service set.
- Class 3 control frames are used to query the access point for buffered data after it awakes from the power-safe mode. Class 3 management frames signal the end of an authenticated relationship.

Which frame is allowed to be sent or received depends on the class and the state of the connection. As it can be seen in Figure 2, the IEEE 802.11 state machine allows class 1 frames to be sent and received in every state. This is due to their function to provide basic services for network discovery and connection management as opposed to data frames which are allowed to be transmitted only in state 3 in which the station is authenticated and associated.

	Control Frames (CF)	Management Frames	Data Frames
Class 1	Request to Send (RTS), Clear to Send(CTS), Acknowledgement(ACK), CF-End, CF-End+CF-Ack	Beacon, Probe Req. / Resp., Authentication Req. / Resp., Deauthentication, Announcement Traffic Indication Message (TIM)	None (infrastructure BSS)
Class 2	None	Association Req. / Resp., Reassociacion Req. / Resp., Disassociation	None
Class 3	Power-Save Poll (PS-Poll)	Deauthentication	All frames

Table 2. Control and management frames as defined in IEEE 802.11

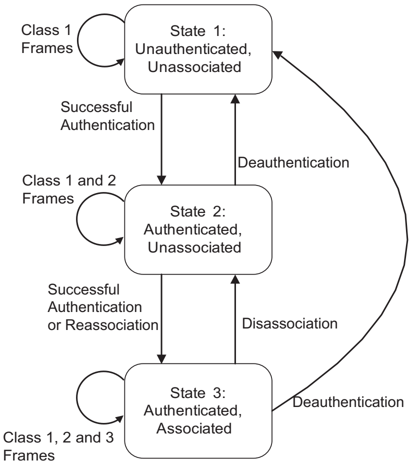

Fig. 2. 802.11 state machine

This fact together with the present situation of the IEEE 802.11 security where no authentication mechanism for control and management frames is provided, serves as a major source of vulnerabilites exploited by DoS attacks.

The adversary has a wide spectrum of different, yet very simple and efficient attacks based on sending different control and management frames with the Medium Access Control (MAC) address of a victim, e.g. the most well-known attack is to send a forged deauthentication frame to an access point with the MAC address of the victim. As a result, the victim's current state will return to state 1 where no data frames are allowed to be transmitted. With the introduction of IEEE 802.11i the situation has not changed for the better. The IEEE 802.11i authentication and key exchange are executed within state 3, providing the adversary with further possibilities for mounting attacks even on the IEEE 802.11i message exchange.

3 Attacks Based on Unauthenticated Frames

In the following subsections we briefly summarize known 802.11 attacks based on unauthenticated frames. For more detailed information and empirical data, see [7,14].

3.1 Deauthentication and Disassociation Attacks

As shown in Figure 2, wireless nodes accept all class 1 frames regardless of their connection state. The trust in the sender's MAC address and the lack of any authentication mechanisms enables an adversary to impersonate a victim's station by using only her MAC address and to send a spoofed deauthentication frame on her behalf. The receiver of the frame then proceeds with deauthentication after which the victim's connection state returns to the unauthenticated and unassociated state. As a result, any further communication will be rejected and the only way for a victim to continue is to repeat the authentication and association procedure of IEEE 802.11 and the authentication and key exchange procedure of IEEE 802.11i all over again. This creates an interruption of communication and leaves a victim with a high delay to re-establish its previous state.

This type of an attack is easily executed and has shown to be very destructive as it provides an adversary with a choice of selectively deauthenticating any authenticated and associated station. In addition, if repeated frequently, this attack creates further vulnerabilities like a total denial of service [7]. The low complexity for mounting this attack and its successful applicability on the IEEE 802.11i security standard makes this attack highly feasible. Today, there exist tools which help to execute it automatedly [10,1].

3.2 Resource Depletion Attacks

A resource depletion attack can be described as a DoS attack based on exhaustion of a system resource like memory or computational power. To mount a memory depletion attack, an adversary tries to exhaust system memory where the state information is being saved leaving no more memory resources to be used by legal clients. In state 1 there is no state information which requires to be explicitly saved at the access point so this attack is only possible when the station is in authenticated or authenticated and associated state. All that is required to exploit this vulnerability is to flood the access point with many authentication requests, all containing different MAC addresses. Each authentication request will be automatically granted due to Open System authenticaton, filling the access point's authentication table. The same attack can be executed by flooding the access point with different association requests and filling up the association table of the access point (the default number of allowed assocations on access points is typically set to 63).

3.3 Power Saving Attacks

To extend their battery life, IEEE 802.11 stations can go into the “sleeping” low-power mode. During that time, frames for these stations are buffered at their access points. Periodically, the station awakes and sends a Power-Save Poll (PS-Poll) frame to retrieve any frame buffered while it was in a power-saving mode.

In [7] a DoS attack is described based on sending an unauthenticated PS-Poll frame. More precisely, to execute this type of attack, the adversary creates and sends a PS-Poll frame with the spoofed MAC address of the sleeping station. The access points replies by sending all buffered frames and removing them from the buffer resulting with a frame loss of the still sleeping station. As already shown in Table 2, the PS-Poll frame is a control frame from class 3. Although easily executed, this type of attack is more limited and works only for those stations which are already in state 3 and are using the power-save mode.

3.4 Attacks on IEEE 802.1X (EAPOL) and IEEE 802.11i

After the IEEE 802.11 Open System authentication and association, the station starts with the 802.1X (EAPOL) authentication process followed by 802.11i 4-Way Handshake (see Fig. 1). This process can be initiated by the station sending an EAPOL-Start message to the access point or by the access point sending an EAPOL-Request Identity message to the station. The result of the authentication is signalled by EAPOL-Success or EAPOL-Failure message.

To execute an attack on the IEEE 802.1X authentication, the adversary can choose to create any of the aforementioned messages by impersonating associated stations and injecting it into legal communication. By sending many EAPOL-Start messages with different MAC addresses the adversary could start many parallel EAPOL authentication sessions mounting a denial-of-service attack on the access point.

Furthermore, a new attack on the 4-Way Handshake protocol of the IEEE 802.11i was identified in [14]. This attack is based on forging the first message of the 4-Way Handshake between the station and the access point where random parameters are exchanged. As a result, the 4-Way Handshake ends up with inconsistent values of the Pairwise Transient Key (PTK) between both parties and all prior authentication must be cancelled.

4 Protection of IEEE 802.11 Against DoS Attacks

According to IEEE 802.11i’s present situation, a mutually authenticated shared secret PMK is established after the IEEE 802.1X authentication protocol. This implies that the mechanisms for authentication of management and control frames could be implemented after 802.1X using an already established and authenticated PMK. Although this solution could be easily implemented, it would not mitigate a lot of the existing attacks, as all the prior communication and the IEEE 802.11i would still be vulnerable as described in subsection 3.4.

To mitigate this vulnerability, our main focus is to protect the communication between the station and the access point as soon as possible. Ideally, the protection should take place before the access point allocates resources for establishing connection state. The only state that does not require any allocation of resources is state 1 (unauthenticated, unassociated) in which the beacon frames and probe requests, probe responses are being transmitted.

We have for that matter developed a protection mechanism using the Diffie-Hellman (DH) key exchange which can be implemented within state 1 and finalized when the station sends an authentication request to the access point and enters state 2. Figure 3 shows an overview of the protection scheme we call the *Early MAC Address Binding*.

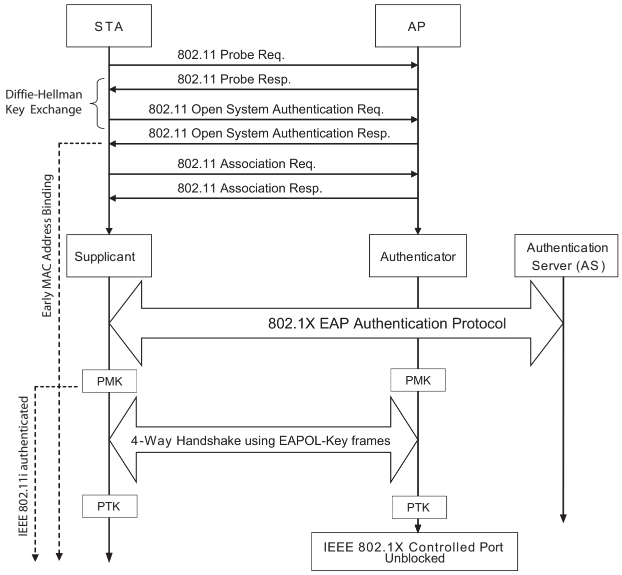
STA
AP
802.11 Probe Req.
802.11 Probe Resp.
Diffie-Hellman Key Exchange
802.11 Open System Authentication Req.
802.11 Open System Authentication Resp.
802.11 Association Req.
802.11 Association Resp.
Early MAC Address Binding
Supplicant
Authenticator
Authentication Server (AS)
802.1X EAP Authentication Protocol
PMK
PMK
IEEE 802.11i authenticated
4-Way Handshake using EAPOL-Key frames
PTK
PTK
IEEE 802.1X Controlled Port Unblocked

Fig. 3. Early MAC Address Binding

The key exchange is embedded within probe response and authentication request. Only the station which has received the probe response and sent an authentication requests will share a secret with the access point. The result of it is that each and every further frame will have a unique message authentication code based on the Diffie-Hellman shared secret. This solution can perfectly co-exist with the IEEE 802.11i security standard. Before and during the IEEE 802.1X execution, the Early MAC Address Binding protects the message exchange and after the IEEE 802.11i has been successfully finalized and PMK derived, the mutual authentication can be used to authenticate the Diffie-Hellman shared secret.

4.1 Early MAC Address Binding

We use the probe response frame to initiate key exchange which is based on the Diffie-Hellman key exchange protocol in an ephemeral-ephemeral mode and is widely used in the Internet, e.g. in the Internet Key Exchange (IKE) as a part of the IPSec key exchange protocol. The algorithms for choosing "good" parameters and their verification can be found in [16]. In the following, we introduce the notation used to describe the key exchange protocol:

- E_k/D_k: symmetric encryption / decryption with a secret key k
- a: the station's private key
- b: the access point's private key
- p, g: p is a large prime and g is a primitive root modulo p
- $PubKey_{STA}$: station's static public key (ephemeral)
- $PubKey_{AP}$: access point's public key (ephemeral)
- $SharedSecret$: Diffie-Hellman shared secret
- h: cryptographic hash function, e.g. SHA-1
- $cookie$: information created from the access point's secret key and station's MAC address

At the beginning of an operation, the access point initializes the public parameters of the Diffie-Hellman key exchange (p and g) and generates its public key $PubKey_{AP} = g^b mod p$ where b is a randomly chosen secret key. The selection of public parameters and the computation of the access point's public key is required to be done only once and may be computed off-line.

After receiving a probe request, the access point answers with a probe response which is extended by $DHParam_{AP}$ Information Element (IE) containing p, g and the access point's public key. Furthermore, together with $DHParam_{AP}$ the access point sends a *cookie*. Its function is to mitigate simple flooding of authentication requests which would consume access point resources. The *cookie* is created by hashing the MAC address from the received probe request with the secret key of the access point. As a result, the cookie is personalized to a particular MAC address of the probe request's sender. This avoids a simple collection of valid cookies made possible by the broadcast characteristic of the wireless medium.

Although the cookie mechanism makes flooding attacks more complex, it does not fully eliminate them. More details on cookie creation and its utilisation against flooding attacks will be discussed in subsection 4.2. Algorithm 1 shows the previously described steps.

After receiving a probe response, the station knows the access point's public key and the Diffie-Hellman parameters. It randomly chooses a secret key a and computes its public key $PubKey_{STA}$ and $SharedSecret$ as shown in Algorithm 2.

Algorithm 1 Generation of Probe Response

```
IF $ReceivedFrame = PROBE\,REQ$ THEN
    $DHParam_{AP} :=< p,\, g,\, PubKey_{AP} >$
    $cookie := h(b \parallel ReceivedFrame.MACAddress)$
    $PROBE\,RESP := PROBE\,RESP \parallel DHParam_{AP} \parallel cookie$
    $send(PROBE\,RESP)$
```

Algorithm 2 Generation of Authentication Request

```
IF $ReceivedFrame = PROBE\,RESP$ THEN
    $PubKey_{STA} := g^{a} mod\, p$
    $SharedSecret_{AP,STA} := PubKey_{AP}^{a} mod\, p$
    $AUTH\,REQ := AUTH\,REQ \parallel PubKey_{STA} \parallel cookie$
    $send(AUTH\,REQ \parallel HMAC_{K,1}(AuthenticationRequest))$
```

The $SharedSecret$ is defined as a mutually shared secret established between the station and the access point after an authentication request is sent to the access point. It serves as secret key to construct keyed message authentication code ($HMAC_K(Frame_i)$) for every frame i starting from the authentication request as defined in [15]. Algorithm 3 shows a computation of HMAC for every frame. Although a generic MAC frame contains a 12-bit sequence number field to discard duplicate frames, the sequence numbers are not used within control frames so we provide an explicit replay protection mechanism and build HMAC values based on their previous value ($HMAC_{K,i-1}$).

Even though the identities of parties involved have not yet been authenticated, after the authenticaton request has been sent, both parties can verify that any frame exchanged between them was neither manipulated nor injected from any third party. We call this the Early MAC Address Binding mechanism.

To be able to insert or temper with frames, an adversary must derive a correct HMAC without knowing the $SharedSecret$ which is equivalent to finding a pre-image of the chosen cryptographic function h or to computing a discrete logarithm problem, both being considered unfeasible provided that the underlying parameters were chosen cautiously. Another possibility would be to steal a valid HMAC as it is transmitted in clear. However, as the computation of the HMAC contains a hash value computed over the whole frame it will not be valid for those frames with other MAC addresses or any other modified values. Lost or

Algorithm 3 Calculation of HMAC

```
$HMAC_{K,1} := HMAC_K(AuthenticationRequest),\, K = SharedSecret_{AP,STA}$
$HMAC_{K,i} := HMAC_K(Frame_i || HMAC_{K,i-1}(Frame_{i-1}))$
```

Algorithm 4 Generation of Authentication Response

```
IF $ReceivedFrame = AUTH\,REQ$ THEN
    IF ($cookie = h(b \parallel ReceivedFrame.MACAddress)$) THEN
        $SharedSecret_{AP,STA} := PubKey_{STA}^{b} mod\, p$
        IF $check(ReceivedFrame.HMAC_{K,1})$ THEN
            $AcceptFrame(ReceivedFrame)$
            $send(AUTH\,RESP \,||\, HMAC_{K,2}(AuthenticationResponse))$
```

invalid frames are retransmitted due to a reliable transport of the IEEE 802.11 link layer which uses acknowledgements for every received frame.

Algorithm 4 shows the generation of an authentication response. Before the access point proceeds with the generation of $SharedSecret$ which is considered to be a computation-intensive operation, the received authentication request must contain a valid cookie. Verifying the cookie can be executed efficiently by using the access point secret key and the MAC address from the received frame ($ReceivedFrame.MACAddress$). If the cookie is valid, the access point computes the $SharedSecret$ and verifies the HMAC. If both checks are successful, the access point will reserve resources and proceed with state 2 by sending an authentication response frame to the station protected with the HMAC.

Computational Requirements of Early MAC Address Binding

The Early MAC Address Binding utilizes a Diffie-Hellman key exchange which is a cryptographic protocol based on a discrete logarithm problem. The performance of the protocol depends on the computational power of the station and access point to compute the $PubKey_{AP} = g^{a} mod\, p$ and the $PubKey_{STA} = g^{b} mod\, p$, respectively. Computational requirements of the Early MAC Address Binding protocol are analogous to the TLS/SSL handshake protocol which also uses the Diffie-Hellman for a key exchange within some of its cipher suites (e.g. TLS_DH_RSA_WITH_AES_128_CBC_SHA). Therefore, any device supporting the TLS/SSL protocol will be able to execute the Early MAC Address Binding. For an initial estimation, we can state that stations, starting from high-end mobile phones, e.g. Sony-Ericsson P910 to PDAs like HP iPAQ Pocket PC from Windows Mobile 2003 version (e.g. HP iPAQ 19xx), and any access point supporting IEEE 802.11i security standard using EAP-TLS authentication protocol will be able to take advantage of an Early MAC Address Binding. For a more detailed qualification and performance evaluation of cryptographic algorithms on hand-held devices we refer the reader to [17,20,6,19].

4.2 Flooding Protection

The process of network discovery is finalized after a station sends an authentication request. If the Open System authentication is used, which is always the case with IEEE 802.11i devices, when receiving the authentication request the access point proceeds automatically with state 2 and allocates resources in order to save the state information for each station, thus creating a potential for a flooding attack. To avoid this attack we have extended the probe response by an additional IE containing a cookie which will be requested by the access point in every authentication request frame before going into state 2. However, this mechanism only avoids attacks based on "blind" flooding with authentication requests. The adversary is still able to mount the same attack by sending many probe requests, collecting valid cookies and then flooding the access point with authentication requests containing cookies.

To mitigate this attack we extend the cookie mechanism by encrypting the cookie with the MAC address of a probe request sender using e.g. the fast RC4 stream cipher which is implemented in all IEEE 802.11 devices (the well-known vulnerabilites based on RC4 cipher do not impact this mechanism because it is not used for confidentiality purposes). In addition, the cookie contains a structure (key, value pair) which allows a station to check if decryption was successful. Furthermore, we change the probe response receiver's address to broadcast (see Algorithm 5).

Algorithm 5 Flooding protection

```
$Ecookie := E_k(''cookie : '' + cookie)$, with $k := PROBE\,REQ.MACAddress$
  $PROBE\,RESP.SetReceiverMACAddress(broadcast)$
  $send(PROBE\,REQ \,||\, Ecookie)$
```

As a result this mechanism avoids an arbitrary sending of authentication requests and persuades the adversary to behave in compliance with the protocol, i.e. to send a probe request, listen for every frame with the broadcast address, and check if the received frame contains the cookie by decrypting it with the MAC address which was used to send the probe request. The simplest way for the adversary to collect enough valid cookies is to send n probe requests, remember the chosen MAC addresses and for every probe response find the matching probe request where the MAC address is a valid decryption key leaving her with $(n^2 - n)/4$ decryptions on average.

Although this mechanism cannot fully avoid authentication flooding attacks, it nevertheless increases its complexity. The efficiency of this protection can be increased by choosing more computationally intensive ciphers for decryption of a cookie or by sending a cryptographic puzzle instead of a cookie. A further improvement would be to periodically change the access point's secret key b so that the time of valid cookies would be limited.

On the other hand, this mechanism brings certain disadvantages for legal stations as they also need to find a cookie by decrypting every probe response. But this disadvantage becomes relevant only in situation where a legal station starts an association process at the same time when an adversary is executing a flooding attack. It is still an open issue if the association of a legal station can be protected during the flooding attack. We leave the quantification of this tradeoff to future work.

5 Related Work

Most of the research concerning 802.11 security has been focused mainly on solving confidentiality problems caused by the Wired Equivalent Privacy (WEP)[5,8,11]. While the ratification of the new IEEE 802.11i standard provides mutual authentication and confidentiality of user data, the vulnerabilities from unauthenticated management and control frames still remain unchanged. In [7] an experimental analysis of different attacks based on MAC address manipulation is given together with other vulnerabilies based on manipulation of media access parameters. The presented experimental results show that using only commodity hardware an adversary can mount an effective denial-of-service attack on selected and all clients. As a countermeasure against deauthentication, the authors propose a simple short-term solution based on queuing of deauthentication requests. After the access point receives deauthentication request, it waits for some time (e.g. 10 seconds) to make sure that no further data was sent from the station, otherwise the deauthentication request is discarded. Although this solution is simple and easily implemented, it only partially solves deauthentication-based attacks. Moreover, due to the long delay of the deauthentication process this solution creates further problems in roaming scenarios, as well as the potential of denial of service attacks based on resource depletion. Another solution proposed in [9] is based on a cryptographic mechanism using RSA for key exchange and AES encryption for confidentiality. Their architecture is effective against all attacks based on manipulated MAC addresses, but also requires significant changes to all IEEE 802.11 devices, and it does not provide backward compatibility. In [14] the authors provide a detailed overview and analysis of the IEEE 802.11i security standard with the well-known but also additional vulnerabilites based on MAC address manipulation and lack of authentication. In [13] the same authors provide a formal analysis of the IEEE 802.11i protocol. As a result, they identify two additional DoS attacks called RSN IE Poisoning and 4-Way Handshake Blocking (which has been briefly described in subsection 3.4). They discuss countermeasures to eliminate attacks which require modifications to the IEEE 802.1X authentication protocol used within IEEE 802.11i.

The major difference between the aforementioned research work and the mechanisms proposed in this work is the focus on providing a light-weight cryptographic solution for an early protection of all management and control frames used in the IEEE 802.11 network which would supplement the existing standard without introducing any new messages or IEEE 802.11 state changes.

6 Conclusion and Future Work

In this work we have presented the Early MAC Address Binding protocol for the protection of unauthenticated management and control frames. The suggested protocol is a light-weight cryptographic protocol that can be embedded within already existing IEEE 802.11 frames. The protection of frames is established at the moment of entering the IEEE 802.11's authenticated but not associated state, which is the first state that requires resource reservation at an access point. As a result, all further message exchanges can neither be manipulated nor any other messages injected into the communication, although both nodes are still not mutually authenticated. The authentication can be provided by the IEEE 802.11i security standard, therefore the Early MAC Address Binding protocol complements the recently ratified standard. The computational requirements of the protocol are analogous to a common TLS/SSL handshake. Furthermore, we have introduced initial ideas against flooding attacks in wireless environments where due to the broadcast medium the traditional flooding protections do not hold.

7 Acknowledgements

We would like to thank Bernhard Kanz for fruitful discussions on cryptographic protocols and Adam Bachorek for his help on IEEE 802.11i testbed measurements.

References

1. Mac MakeUp. http://www.gorlani.com/publicprj/macmakeup/, 2003.
2. IEEE 802.11. Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Standard, July 1999.
3. IEEE 802.11i/D10.0. Security Enhancements, Amendment 6 to IEEE Standard for Information Technology. IEEE Standard, April 2004.
4. IEEE 802.1X. IEEE Standard for Local and Metropolitan Area Networks - Port-Based Network Access Control. IEEE Standard, June 2001.
5. W.A. Arbaugh, S. Shankar, J. Wang, and K. Zhang. Your 802.11 network has no clothes. In *Proceedings of the First IEEE International Conference on Wireless LANs and Home Networks*, pages 15–28, December 2001.
6. P. Argyroudis, R. Verma, H. Tewari, and D. O'Mahony. Performance analysis of cryptographic protocols on handheld devices. In *Proceedings of the 3rd IEEE International Symposium on Network Computing and Applications*, pages 169–174, August 2004.
7. J. Bellardo and S. Savage. 802.11 Denial-of-Service attacks: real vulnerabilities and practical solutions. In *Proceedings of the USENIX Security Symposium*, pages 15–28, August 2003.

8. N. Borisov, I. Goldberg, and D. Wagner. Intercepting mobile communications: the insecurity of 802.11. In *MobiCom '01: Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pages 180–189, July 2001.
9. D. Faria and D. Cheriton. DoS and authentication in wireless public access networks. In *Proceedings of the 2004 ACM Workshop on Wireless Security*, pages 47–56, September 2002.
10. R. Floeter. Wireless Lan Security Framework: void11. http://www.wlsec.net/void11, 2002.
11. S. Fluhrer, I. Mantin, and A. Shamir. Weaknesses in the key scheduling algorithm of RC4. In *SAC '01: Revised Papers from the 8th Annual International Workshop on Selected Areas in Cryptography*, pages 1–24, August 2001.
12. M. Gast. *802.11 Wireless Networks: The Definitive Guide*. O'Reilly, 2005.
13. C. He and J. C. Mitchell. Analysis of the 802.11i 4-way handshake. In *Proceedings of the 2004 ACM Workshop on Wireless Security*, pages 43–50, October 2004.
14. C. He and J. C. Mitchell. Security analysis and improvements for IEEE 802.11i. In *Proceedings of the 12th Annual Network and Distributed System Security Symposium (NDSS'05)*, pages 90–110, February 2005.
15. IETF. HMAC: Keyed-Hashing for Message Authentication. RFC2104, 1997.
16. IETF. Diffie-Hellman Key Agreement Method. RFC2631, 1999.
17. G. Kambourakis, A. Rouskas, and S. Gritzalis. Performance evaluation of public key based authentication in future mobile communication systems. *EURASIP Journal on Wireless Communications and Networking*, pages 184–197, February 2004.
18. C. Lambrinoudakis and S. Gritzalis. Security in IEEE 802.11 WLANs. In M. Ilyas and S. Ahson, editors, *Handbook of Wireless Local Area Networks: Applications, Technology, Security, and Standards*. May 2005.
19. N. Potlapally, S. Ravi, A. Raghunathan, and N. K. Jha. Analyzing the energy consumption of security protocols. In *ISLPED '03: Proceedings of the 2003 international symposium on Low power electronics and design*, pages 30–35, August 2003.
20. S. Tillich and J. Großschädl. A survey of public-key cryptography on J2ME-enabled mobile devices. In *Proceedings of 19th International Symposium on Computer and Infomation Sciences - ISCIS 2004*, pages 935–944, October 2004.