Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2006, Article ID 47453, Pages 1–19
DOI 10.1155/WCN/2006/47453

Static and Dynamic 4-Way Handshake Solutions to Avoid Denial of Service Attack in Wi-Fi Protected Access and IEEE 802.11i

Floriano De Rango, Dionigi Cristian Lentini, and Salvatore Marano

Department of Electronics Informatics and Systems (D.E.I.S.), University of Calabria, Via P. Bucci, 87036 Rende, Cosenza, Italy

Received 10 October 2005; Revised 2 April 2006; Accepted 13 June 2006

This paper focuses on *WPA* and *IEEE 802.11i* protocols that represent two important solutions in the wireless environment. Scenarios where it is possible to produce a *DoS attack* and *DoS flooding* attacks are outlined. The last phase of the authentication process, represented by the *4-way handshake* procedure, is shown to be unsafe from DoS attack. This can produce the undesired effect of memory exhaustion if a flooding DoS attack is conducted. In order to avoid DoS attack without increasing the complexity of wireless mobile devices too much and without changing through some further control fields of the frame structure of wireless security protocols, a solution is found and an extension of WPA and IEEE 802.11 is proposed. A protocol extension with three "static" variants and with a resource-aware dynamic approach is considered. The three enhancements to the standard protocols are achieved through some simple changes on the client side and they are robust against DoS and DoS flooding attack. Advantages introduced by the proposal are validated by simulation campaigns and simulation parameters such as attempted attacks, successful attacks, and CPU load, while the algorithm execution time is evaluated. Simulation results show how the three static solutions avoid memory exhaustion and present a good performance in terms of CPU load and execution time in comparison with the standard WPA and IEEE 802.11i protocols. However, if the mobile device presents different resource availability in terms of CPU and memory or if resource availability significantly changes in time, a dynamic approach that is able to switch among three different modalities could be more suitable.

1. INTRODUCTION

One of the greatest challenges to any organisation is securing its data against unauthorised access, particularly those organisations that use wireless network technologies to manage information.

Wireless networks offer organisations and user benefits such as portability, flexibility, increased productivity, and lower installation costs. However, risks are inherent in any technology. Some of the wireless risks are similar to those of wired networks, some are exacerbated, and some are new.

The most significant source of risks in wireless networks is the use of radio waves to transmit data. Radio waves move through the air and can be intercepted by any attacker with the appropriate equipment.

Security measures prevent and reduce the risk of unauthorised access into wireless network resources. On the other hand, these measures can reduce productivity, by creating additional processes and work.

In this context an extension to the authentication phase of Wi-Fi protected access (WPA) [1–5] and IEEE 802.11i [6, 7] is proposed in this paper. WPA and IEEE 802.11i represent, until now, two important solutions to security issues in wireless networks and they have been considered in our analysis. Some particular risk scenarios, where the 4-way handshake of the authentication phase of these two protocols can fail, is analysed. Specifically, we consider a previously proposed solution published in [8, 9] (solution I) but simulation results are added and more implementation details are provided. Then, a second solution (solution II), suggested but not tested by He and Mitchell, is also considered, and a third novel solution that tries to release memory at the mobile device is also investigated and compared with the previous solutions. Because these solutions are prefixed at the terminal, they are called *static*. These extensions to the standard protocols need just a few operations on the client side without introducing additional fields in the WPA and IEEE 802.11i protocol frame format. This permits the standard protocols to be used and to avoid the possible issues of DoS attacks. Simulation results show the benefits introduced by these static solutions in terms of CPU and memory load. However, as experienced by simulation results, the best solution does not exist if one of these static approaches are used because mobile-devices can present different characteristics and resource availability. Thus, a dynamic resource-oriented approach is proposed (solution IV). This approach tries to take full advantage of the single solutions through a dynamic thresholds mechanism that is able to switch among the three previously analysed solutions. In order to implement this mechanism, just some modifications to the mobile device need to be performed such as a monitoring software that is able to account for the current CPU and memory load and a switching center that permits the 4-way handshake messages exchange modality to be changed in accordance with solutions I, II, and III.

The paper is organised as follows: Section 2 presents a brief overview of the work related to the security of wireless LAN; WPA and IEEE 802.11i are presented in Section 3; the 4-way handshake is introduced in Section 4; Section 5 presents some DoS and DoS flooding attack scenario during the 4-way handshake phase of the WPA/IEEE 802.11i; in Section 6 three static solutions to avoid DoS attacks are presented (two of them have been qualitatively and without simulations considered in [8, 9]) and a fourth dynamic solution is proposed; FSA modelling is reported in Section 7; Section 8 presents the simulation results; finally the conclusions are summarised in the last section.

2. RELATED WORK

In the last few years the attention of industry and the academic world has been focused on security protocols over wireless networks and specifically on WLAN. A lot of authentication protocols, in particular EAP (extension access protocol), such as transport layer security (TLS), MD5, tunnelled TLS (TTLS), light extensive access protocol (LEAP), protected EAP (PEAP), SecureID, SIM, GTC, and AKA have been proposed in the literature [10].

In order to offer data confidentiality equivalent to a wired network, the IEEE 802.11 standard [11] defined wired equivalent privacy (WEP). However, numerous researchers have shown that none of the data confidentiality, integrity, and authentication could be achieved through the intrinsic mechanism of this protocol [7, 12].

WEP is based on a cryptography system that was alleged to offer privacy, authentication, and integrity.

A secret key K of 40 bits is shared by APs and clients of the wireless network and it is queued at an initialisation vector (IV) of 24 bits. The string, obtained by this process, is encrypted by RC4 in order to obtain the encoding key (key stream) of messages in clear.

The authentication process is only *one-way* (client-side). This means that only the client needs to be authenticated by the AP and not vice versa. This behaviour does not give guarantees about the network that the client is referring to.

WEP uses the RC4 algorithm that is well known to be unsafe if the same keys are used several times in it. The initialisation vector, applied to the RC4, is 24 bits and this determines the repetition of the keys after a while permitting a *known-plaintext* attack [13]. A further characteristics of WEP is the restarting of VI (or IV) if a packet collision occurs.

Another issue of WEP protocol is the manual distribution of keys over all AP stations. In this key management scheme, all clients of the same basic service set (BSS) use the same key, thereby reducing the privacy of the other users.

The authentication phase is not very safe because it makes use of the same key as the encryption in input to RC4 to authenticate the client. Another bad use of the mechanisms in WEP protocol is the integrity check through the CRC-32 algorithm. Cyclic redundancy check (CRC) is a noncryptographic function f that has the linearity property. This means that the function f is linear if $f(a) XOR f(b) = f(a XOR b)$. Details of CRC techniques can be found in [4].

This linearity property can be used by the hackers to violate the integrity of packets.

All these security issues were analysed by the internet security, applications, authentication, and cryptography (ISAAC) of University of California and then, by three people (Fhurer et al.) in a famous paper [14]. The failure of WEP is due to the choice of the project managers in the security architecture deployment.

Although WEP fails to satisfy any security requirements, it is not practical to anticipate users to completely discard their devices with WEP already implemented. Hence, Wi-Fi alliance proposed interim solution, called protected access (WPA), to ameliorate the vulnerabilities by reusing the legacy hardware. WPA adopts a temporal key integrity protocol (TKIP) for data confidentiality and the Michael algorithm, a weak keyed message integrity code (MIC), for improved data integrity under the limitation of the computation power available in the devices. Moreover, in order to detect replayed packets, WPA implements a packet sequencing mechanism by binding a monotonically increasing sequence number to each packet. In addition, WPA provides two improved authentication mechanisms. For more details about WPA, see [1]. Despite these security enhancements of WPA, a weakness is predestined since WPA appears owing to the limitations of reusing the legacy hardware. Although TKIP key-mixing function has stronger security than WEP key-scheduling algorithm, it is not so strong as expected [15]. Moreover, Michael algorithm is designed to provide only 20 bits (or possibly slightly more) of security in order to minimise the impact on the performance, which means an adversary can construct one successful forgery every 2^{19} packets. Some countermeasures have been adopted [6]. However these countermeasures may allow DoS and DoS flooding attacks. In addition, the 802.1X authentication may be vulnerable to session hijacking and man-in-the-middle (MitM) attacks [2]. In order to provide an enhanced MAC layer security, IEEE 802.11i was proposed [6]. Under the assumption of upgrading the hardware, 802.11i defines a countermode/CBC-MAC protocol (CCMP) that provides strong confidentiality, integrity, and replay protection [16]. Moreover, an authentication process, combining the 802.1X authentication and key management procedures, is

Table 1: Comparison of WEP, WPA, and IEEE802.11i features.

	WEP	WPA	IEEE 802.11i
Encryption algorithm	RC4	RC4 (TKIP)	AES-128
Key length	40 bit	128 bit	128-192-256 bit
IV length	24 bit	48 bit	48 bit
Authentication	Only client	Mutual	Mutual
Integrity	CRC	MIC	CCM-MIC
Key type	Static	Dynamic	Dynamic
Key distribution	Manual	Dynamic	Dynamic
HW compatibility	Easy	Easy	Difficult

performed to mutually authenticate the devices and generate a fresh session key for data transmissions. However, also the IEEE 802.11i can present some weakness to possible DoS attacks such as shown by the authors in [8, 9]. This paper starting from the He and Mitchell's work focuses on the 4-way handshake procedure giving some implementation guide to simulate the DoS attacks on the WLAN networks and showing the performance degradations when DoS or DoS flooding attacks are led to the mobile devices. Moreover, a variant with memory release and a dynamic approach that tries to combine the different solutions analysed in this paper is proposed. In the following a brief overview of WPA and IEEE 802.11i and 4-way handshake will be given.

3. OVERVIEW OF WPA AND IEEE 802.11i

When the security bugs of WEP were outlined, it was clear that the big issue was to find a right solution to the security in a short time as possible.

Wi-Fi alliance and the IEEE 802.11i agree with the need to develop a novel standard, able to overcome all the security bugs. This standard was the IEEE 802.11i [6]. However, this novel protocol would have caused an incompatibility with 200 million wireless devices around the world. Thus in order to resolve WEP bugs and to offer backward compatibility and a migration toward the IEEE 802.11i, the Wi-Fi protected access (WPA) has been proposed. Even if in June 2004 the more complete IEEE 802.11i was released; WPA represents till now the most used security standard. It resolves WEP bugs and offers a light-weight transition toward more complex security protocol such as 802.11i.

WPA arose as a solution to WEP inconsistency and it is considered one of the best security dynamic protocols; the novel standard IEEE 802.11i is also called *WPA2* [16].

Differently from the WEP, *Wi-Fi protected access* (WPA) implements a set of functions called *robust security network association* (RSNA) that offers a greater security level, authentication, and integrity check functions.

For data encryption, WPA uses the default *temporary key integrity protocol* (TKIP), maintaining backward compatibility through further WEP support.

TKIP makes four distinct enhancements to WEP (Table 1). Firstly, it increases the IV size from 24 to 48 bits, meaning that key reuse is no longer a worry. Secondly, it forces the sequence number to increase monotonically to avoid replay. Thirdly, it mixes the sequence number and transmits the address with the WEP base key to derive a per-frame key. Finally, it includes a message authentication code (MIC) of the source and destination addresses, the priority, and the plaintext data, to allow forgeries to be detected.

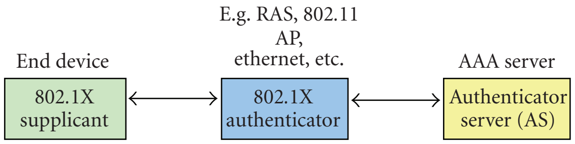


Figure 1: Supplicant, authenticator, and authentication server in IEEE802.1X/EAP.

Regarding the authentication phase, WPA has its strong point in the 802.1X/EAP architecture that offers mutual authentication between the AP and the client, the negotiation of the authentication scheme and dynamic key distribution (this avoids the manual configuration of the static key).

The 802.1X architecture provides three entities involved in the communication: the supplicant (S) that represents the client or the peer that asks for network access; the authenticator (A) that is the AP that offers the access service; the authentication server (AS) that is represented by the server that realises the authentication and the authorization phases toward the client such as *remote authentication dial in user service* (RADIUS), and DIAMETER [17, 18] (see Figure 1).

However, WPA permits the use, alternatively to the standard authentication mode with 802.1X/EAP, of the preshared key (PSK) mode. This modality defines a preconfiguration of the secret key, set manually or through other agreements on devices, that had been introduced by WPA (and maintained in 802.11i) in order to permit its use in small office home-based networks (SOHO) environment and in overall environments where it is not possible to use a structured and hierarchical structure such as RADIUS [17].

Three logically distinct subphases can be observed inside the 802.1X authentication phase:

(i) *EAP initialising*,
(ii) *EAP-TLS handshake*,
(iii) *4-way handshake*.

In this work attention will be focused on the *4-way handshake* phase.

The 4-way handshake is preceded by an EAP procedure. Even if the EAP procedure is not an effective part of 802.11i, we mention it for completeness. We refer, for example, to EAP-TLS procedure. At the end of the EAP-TLS handshake, the supplicant encrypts a *premaster key* with the public key of server (this key is communicated by the server to the client during the TLS-handshake) and sends it to server, which can decrypt the message with its private key. In this way, supplicant and server share the premaster key. Starting from this last knowledge, the *pairwise master key* (PMK) can be derived in the same time on both the client and the server, applying a specific cryptography function called *pseudorandom function* (PRF). At this point, the server will transfer its own PMK through a *RADIUS accept* packet on the AP to which the client station is associated.

If WPA with *preshared key* (PSK) is applied, the PMK will be equal to the PSK and the previous phases will be completely jumped. From this point the server, through the sending of PMK, delegates A to do overall security functions toward S.

Once S and A share PMK, they can complete the authentication phase with the *4-way handshake*. This last sub-phase is the core of the key management scheme of the WPA protocol. Starting from the common PMK, S and A are able to obtain the *temporary key* that is useful for the data encryption. It is possible to get this temporary key without explicitly exchanging it among the sides. The temporary key is one component of the *pairwise transient key* (PTK).

Specifically, the first 128 bits of the PTK is called session "key confirmation key" (KCK), which is used to prove possession of the PMK. The second 128 bits is the session "key encryption key" (KEK), which is used to distribute the current broadcast key. The temporal key is the remaining bits of the PTK.

Through the 4-way handshake, 802.1X/EAP permits different temporary keys to be obtained for each user, for each session, and also for each packet. In this way WPA resolves the security issue of WEP associated with the static keys and with the manual distribution of these keys among participants to protect communication; WPA permits configuration parameters to be negotiated, to generate a temporary key, and to change this key on the packet basis in a secure manner.

In 2004, the "IEEE Task Group i" released the final version of the IEEE 802.11i protocol. Because this protocol is based on RSNA architecture of WPA and it guarantees backward compatibility with WPA, it is also called WPA2 [6].

The main difference between WPA and IEEE 802.11i is the cryptography algorithm applied to data transmission. WPA uses TKIP (based on RC4), while 802.11i applies a novel protocol called CCMP (countermode with CBC-MAC protocol). The latter is based on AES. CCMP represents an extension of 802.11i because TKIP is still supported but it needs a change in the hardware architecture. CCMP improves the security level of wireless communication but it produces computational overhead needing hardware upgrading and new coprocessors able to manage the cryptography processing load in real time. On the contrary TKIP in WPA can be executed mostly by low-power processors that are supported in the overall existing APs and, moreover, TKIP reuses the WEP hardware to offload most of the computational expenses from the software.

Because WPA is light in comparison with WPA2, it better meets the needs of wireless devices (palmtop or other little devices) with few available resources. Another characteristic of WPA against WPA2 is the interoperability between IEEE 802.11b (but not only) and IEEE 802.11i devices. These reasons slow down the migration from WPA to WPA2.

4. THE 4-WAY HANDSHAKE

In previous sections we have focused on the interesting novel aspect of the authentication phase of WPA and 802.11i, where it is possible to generate and dynamically distribute different temporary keys on session, user, and packet basis. This assures a high robustness against hacker attacks versus WEP. After 802.1X authentication is completed, 802.11i begins to secure the link by executing the 4-way handshake.

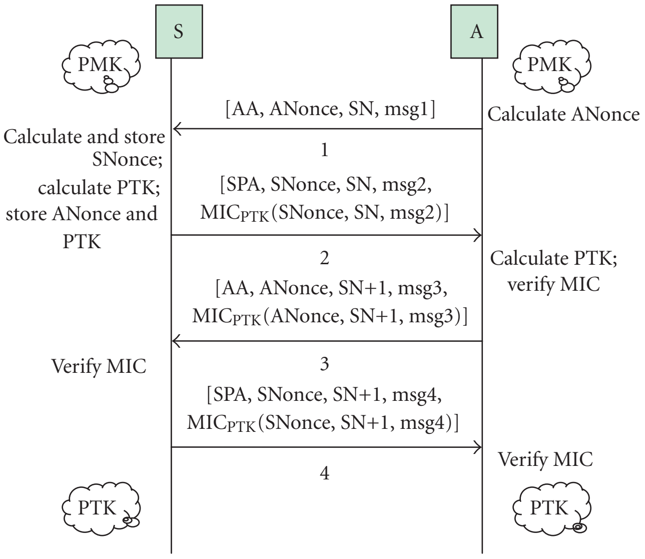

S
A
PMK
PMK
[AA, ANonce, SN, msg1]
Calculate ANonce
Calculate and store SNonce; calculate PTK; store ANonce and PTK
1
[SPA, SNonce, SN, msg2, $\mathrm{MIC}_{\mathrm{PTK}}$(SNonce, SN, msg2)]
2
Calculate PTK; verify MIC
[AA, ANonce, SN+1, msg3, $\mathrm{MIC}_{\mathrm{PTK}}$(ANonce, SN+1, msg3)]
Verify MIC
3
[SPA, SNonce, SN+1, msg4, $\mathrm{MIC}_{\mathrm{PTK}}$(SNonce, SN+1, msg4)]
Verify MIC
4
PTK
PTK

Figure 2: The 4-way handshake messages.

The 802.11i 4-way handshake procedure makes the following steps.

(a) It derives a fresh session key (TKIP).
(b) Through transmission and receiver timers management and handshake messages it synchronises its operations.
(c) It distributes a broadcast key from the AP to the station.
(d) It verifies that peer is live.
(e) It confirms that peer possesses the station.
(f) It binds the MAC addresses of the station and AP to this key.

In the 4-way handshake only 4 types of messages are considered and structured as follows (see Figure 2):

(i) Msg 1: [AA, ANonce, SN, Msg1];
(ii) Msg 2: [SPA, SNonce, SN, Msg2, MIC(SNonce, SN, Msg2)];
(iii) Msg 3: [AA, ANonce, SN + 1, Msg3, MIC(ANonce, SN + 1, Msg3)];
(iv) Msg 4: [SPA, SNonce, SN + 1, Msg4, MIC(SNonce, SN + 1, Msg4)];

where

(1) *AA represents the MAC address of AP (A) wireless card;*
(2) *SPA is the MAC address of S wireless card;*
(3) *ANonce is a random value generated by A;*
(4) *SNonce is a random value generated by S;*
(5) *SN represents the sequence number of the message;*
(6) *MsgX identifies the type of message X.*

The protocol starts with the generation of a random bits string called “nonce.” This nonce is generated only once. At the beginning A generates this nonce (ANonce) and it puts this one inside the first message (Msg1) sent to S. After receiving Msg1, S will know AA, SN, and Anonce. These values can be useful in the generation of PTK. S will produce a novel *nonce* called SNonce that will be used with PMK and ANonce to generate the PTK in the following way:

$$\text{PTK} = \text{PRF}(\text{PMK}, \text{ANonce}, \text{SNonce}, \text{AA}, \text{SPA}), \qquad (1)$$

where PRF is a pseudorandom cryptographic function.

After calculating PTK, S will store ANonce, SNonce, and PTK and it will send the Msg2 to A. In Msg2 the MIC of other fields that travel in clear on the wireless channel will be also inserted. It is important to observe that the MIC value is calculated through the PTK previously obtained value and for this reason it is univocally dependent by PTK.

At the reception of Msg2, A has to calculate the novel PTK with the same procedure adopted by S. It is possible to calculate the PTK because S, after the reception of Msg2, knows SNonce. Through the PTK value it is possible to calculate again the MIC value associated with the PTK and to compare this new value with the MIC inserted in the Msg2. If MIC_{msg2} is the same of MIC calculated by A, the sender of Msg2 is identified and A can be sure to have sent ANonce to the right node (S) and to share with him the same PMK.

At this phase of the procedure, S and A share PTK and they have to only give confirmation of the applied sharing among them. Thus A sends to S the Msg3 (with ANonce and MIC values) and S and, after verifying the integrity, concludes the handshake with the sending of Msg4.

Moreover, the 4-way handshake provides an asymmetric scheme of alert as presented below:

(i) if S and A receive a message with invalid SN or MIC values, they will discard the message; this approach avoids the “man-in-the-middle” attack [2];
(ii) if S does not receive the Msg1 within a time stamp (TS), it will disassociate, deauthenticate, and start the authentication procedure again;
(iii) if A does not receive the Msg2 (or Msg4) within TS, it will try to send the Msg1 (or Msg3) again; so after k attempts, it will deassociate S.

This asymmetric behaviour is necessary to avoid deadlock as shown in Figure 3: if Msg2 is lost before arriving at its destination and A does not send the Msg1 again, the handshake procedure will terminate without completing its task; the same issue could happen if A sends Msg1 again but S is not available to accept Msg1 because it is waiting for Msg3 (Figure 3).

WPA resolves WEP bugs but it does not represent a universal model of invulnerability. As all session-oriented protocols, WPA also is sensitive to a specific category of attack.

It is important to observe how WPA in preshared key mode offers the same key sharing issues as WEP. If the privacy issues of the key are not considered, it is important to configure PSK on the AP and on all stations that have to communicate with the AP. However, because the PSK is the same as PMK in the starting phase of the 4-way handshake, the overall authentication phase is not executed and only the final 4-way handshake is applied. This mechanism does not offer high security and thus the producers of Wi-Fi suggest using WPA in preshared key only in the home of the SOHO environment or in environments where there is a low probability of attempting attacks on security [8, 9].

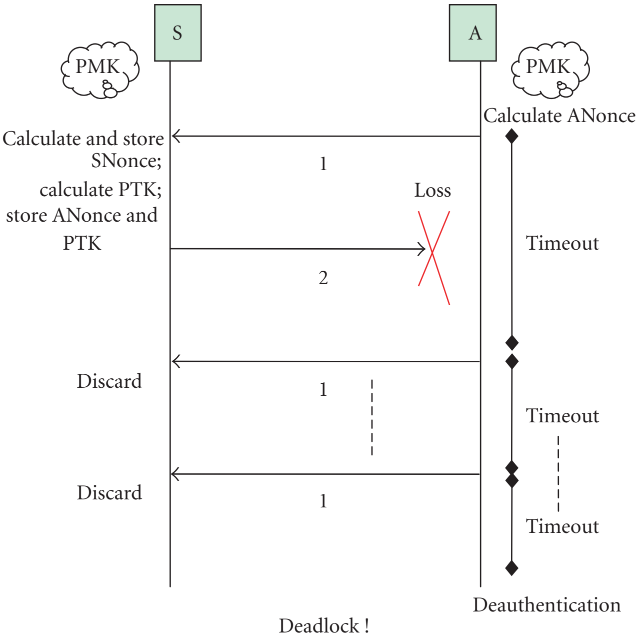


Figure 3: Deadlock situation after packet loss in the case of symmetric behaviour.

During 2002, when WPA started to appear on the market, some important publications denounced the security bugs of WPA in standard mode (e.g., authentication 802.1X/EAP) and in PSK mode. These bugs create a lot of problems were provided. The denounced bugs offer the possibility of trying attacks such as man-in-the-middle, session hijack, and denial of service (DoS). The first two types of attacks were possible only if a one-way authentication was used (e.g., EAP-TLS and PEAP). The third type consisted of spoofing of 4 types of EAP protocol messages. So in this case the possibility of hacking referred to

(i) flooding *associate requests* or *EAOPL-start* packets to AP (even if EAPOL-start messages is not a serious problem in practice);
(ii) falsifying the *EAP-failure* messages;
(iii) falsifying the *EAP-logoff* or *disassociate-request* messages;
(iv) falsifying the *deauthentication* message.

For details of the attack techniques we refer to [2, 9, 10, 19–21]. All security bugs outlined in 2002, except problems attached to disassociate or deauthenticate requests, have been avoided in the following release of WPA and in WPA2. In 2003, in order to show the vulnerability of WPA, a solution was adopted by Moskovitz [22]. However this approach can

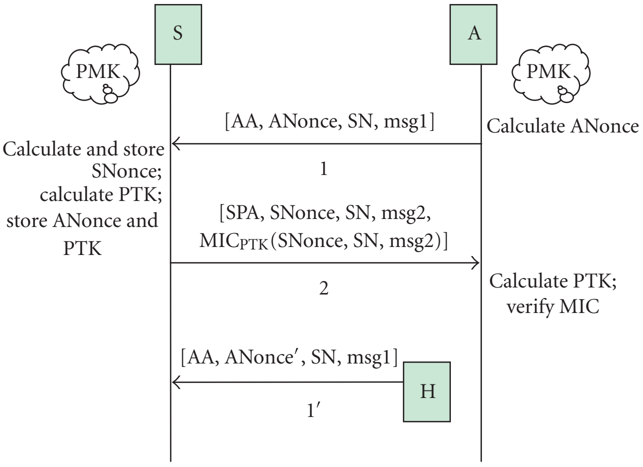

S
A
PMK
PMK
[AA, ANonce, SN, msg1]
Calculate ANonce
Calculate and store SNonce; calculate PTK; store ANonce and PTK
1
[SPA, SNonce, SN, msg2, MIC_{PTK}(SNonce, SN, msg2)]
2
Calculate PTK; verify MIC
[AA, ANonce′, SN, msg1]
H
1′

Figure 4: Hacker intrusion after Msg2 forwarding.

be considered a theoretical conjecture and with few chances to be applied in a real context. This potential attack was avoided through the adoption of AES-CCMP rather than RC4-TKIP in WPA2.

Contrary to previous works, in this paper, after a deep analysis of WPA and WPA2 protocol dynamics attention is focused on the *4-way handshake* procedure and even when it is used in standard mode (not only in PSK mode). If a hacker is able to successfully attack the key generation and distribution process, then the overall security system will be damaged.

5. DoS AND DoS FLOODING ATTACKS AGAINST IEEE 802.11i 4-WAY HANDSHAKE

In the current WLAN systems, DoS attacks are very easy to mount; furthermore, once an adversary successfully mounts a DoS attack, more advanced attacks, such as MitM [2], could be subsequently constructed. Therefore, it is necessary to deploy a security mechanism that can defend against DoS attacks. In the following, in accordance with the works [8, 9], some possible DoS attack scenario is presented.

The weak point of 4-way handshake is represented by the first message (Msg1). It is the only message that does not use the MIC field that is very important to guarantee the integrity and the authenticity of A.

Thus, Msg1 can be falsified and a hacker can easily know all its fields such as the MAC address, ANonce, SN, and message type. We recall, as explained in the previous section, that S calculates and stores PTK together with ANonce and Snonce (see formula (1)). Through PTK, S calculates the MIC to be inserted in Msg2 and sends it to A. After receiving the Msg2, A calculates its PTK and then MIC. At this point, hacker (H) can play a role that prepares Msg1 to the message similar to that sent from A to S (Figure 4). This new Msg1′ message differs from Msg1 only in the nonce because this value is randomly generated locally in the device.

The device S calculates the PTK in the knowledge of ANonce received with Msg1. Let the value generated by H be indicated with ANonce′ so that it is possible to discriminate this from the value created by A (ANonce).

If H is able to send its message (Msg1′) after S sends Msg2 and before S receives Msg3, S should accept Msg1′ and it will calculate a novel value PTK that will be indicated by PTK′. In other terms PTK′ will be a function of PMF, ANonce′, and SNonce:

$$\text{PTK}' = \text{PRF}\,(\text{PMF}, \text{ANonce}', \text{SNonce}, \text{AA}, \text{SPA}). \qquad (2)$$

The effect produced by the hacker is the storing of two new values (*ANonce*′ and *PTK*′) and the sending of a new message (*Msg2*′) from S to A. This new message will be *silently discarded* by A in accordance with the protocol specification.

In this time A will send the message Msg3 to S with its own ANonce value. After receiving the Msg3, S will notify a failure in the integrity check because $\text{MIC}_{\text{PTK}} \neq \text{MIC}_{\text{PTK}'}$. This is due to the PTK′, derived by ANonce′, which produces a different MIC at S. Thus a discarding of Msg3 is produced without giving any communication to A.

We recall how the *silent discarding* was appositely introduced by the project manager of WPA in order to avoid attacks such as MITM (man-in-the-middle attack).

After *timestamp expiration*, the authenticator A, because it does not receive the Msg4, will send Msg3 again. This novel Msg3 will again be discarded by S. After the *n*th attempt at transmission and timestamp expiration, A will deauthenticate S and the hacker will achieve his task: *to make a DoS attack* [9, 10, 19–21] (Figure 5).

It is important to observe that after each reception of Msg1, supplicant S stores ANonce and PTK values in its own station. Thus, if the hacker achieves a multiple attack, it is possible to achieve a *DoS flooding* or *DoS memory exhaustion* attack (such as shown in Figure 6); if the attack is repeated through flooding by the hacker, S is forced to store a lot of ANonce and PTK values producing memory exhaustion. This attack is possible with both WPA and 802.11i protocols.

On reflection, the standard IEEE 802.11i supports a mechanism to avoid the modification of PTK on S. This mechanism forces S to calculate a PTK and a *temporary PTK* (TPTK), after receiving the Msg1. In this way S can modify only the TPTK for each reception of Msg1 and PTK can be modified only after the reception of Msg3 and the verification of its integrity.

This approach resolves the security issue only if the multiple instances of connection are sequentially executed; in a scenario such as that previously described, the DoS attack continues to be applicable because S will calculate its own MIC through the TPTK value, while the received Msg3 will carry in a MIC calculated through the PTK value. Thus, because ANonce is different from ANonce′, then PTK ≠ TPTK and the MIC verification will fail with subsequent Msg3 discarding (Figure 7).

However, the attack is not so easy for the hacker. The IEEE 802.11i standard supports two mechanisms to reduce the action range of possible DoS attacks. The first mechanism uses PMKID in Msg1 and the second mechanism is the *link layer data encryption* (LLDE) protocol.

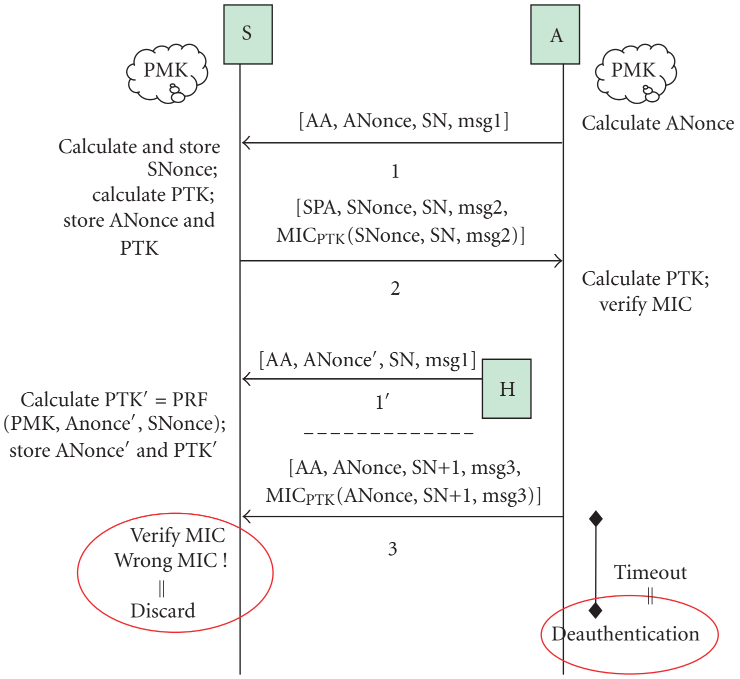

S
A
PMK
PMK
[AA, ANonce, SN, msg1]
Calculate ANonce
Calculate and store SNonce; calculate PTK; store ANonce and PTK
1
[SPA, SNonce, SN, msg2, MICPTK(SNonce, SN, msg2)]
2
Calculate PTK; verify MIC
[AA, ANonce′, SN, msg1]
H
1′
Calculate PTK′ = PRF (PMK, Anonce′, SNonce); store ANonce′ and PTK′
[AA, ANonce, SN+1, msg3, MICPTK(ANonce, SN+1, msg3)]
3
Verify MIC Wrong MIC ! || Discard
Timeout
||
Deauthentication

Figure 5: DoS attack in 4-way handshake phase.

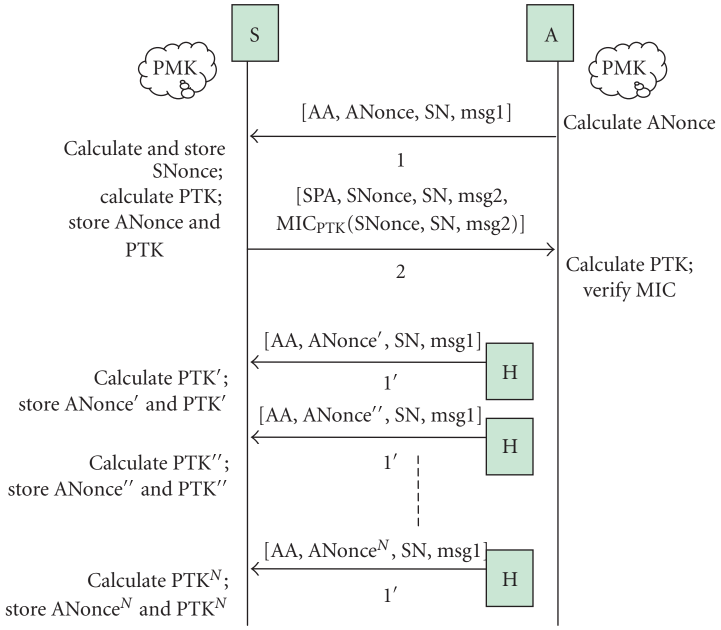

S
A
PMK
PMK
[AA, ANonce, SN, msg1]
Calculate ANonce
1
Calculate and store SNonce; calculate PTK; store ANonce and PTK
[SPA, SNonce, SN, msg2, MICPTK(SNonce, SN, msg2)]
2
Calculate PTK; verify MIC
[AA, ANonce′, SN, msg1]
H
1′
Calculate PTK′; store ANonce′ and PTK′
[AA, ANonce″, SN, msg1]
H
1′
Calculate PTK″; store ANonce″ and PTK″
[AA, ANonce^N, SN, msg1]
H
1′
Calculate PTK^N; store ANonce^N and PTK^N

Figure 6: DoS flooding attack in 4-way handshake phase.

The PMKID is a further field included in all Msg1 sent from A to S. It is calculated as follows:

$$\text{PMKID} = \text{SHA}\,(\text{PMK}, \text{T}), \qquad (3)$$

where T = [“PMK name” ||AA||SPA] and SHA is the well-known *one-way hash function* SHA-1 at 128 bits. This value cannot be reproduced by the hacker because he cannot know the PMK and he cannot think of inverting the hash function. Thus H cannot send Msg1 to S before A sends its Msg1 otherwise H could be discovered.

Without the PMKID, the DoS flooding attack could be attempted all the time. Even before sending the Msg1 from A to S, this attack could be produced by H without becoming aware of the Msg1.

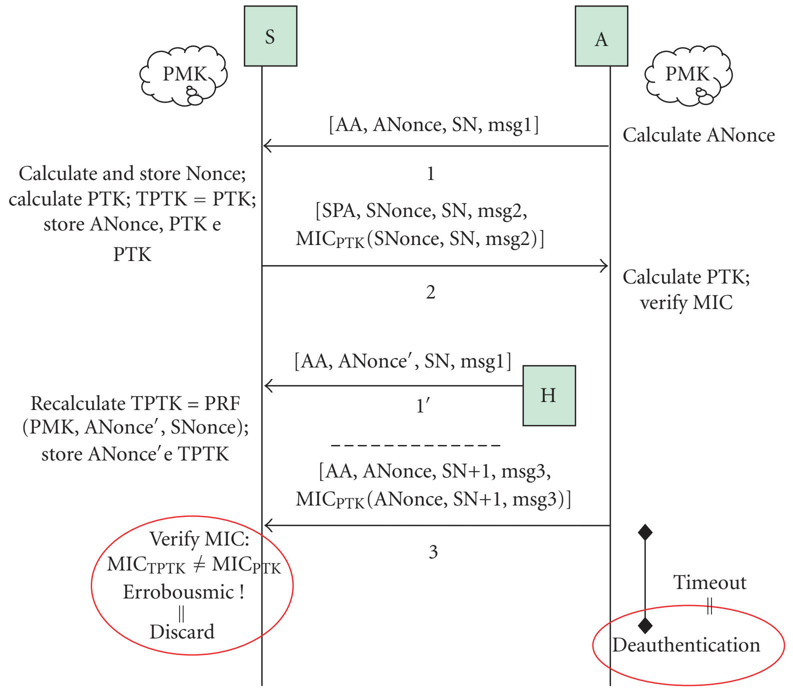

S
A
PMK
PMK
[AA, ANonce, SN, msg1]
Calculate ANonce
1
Calculate and store Nonce; calculate PTK; TPTK = PTK; store ANonce, PTK e PTK
[SPA, SNonce, SN, msg2, MICPTK(SNonce, SN, msg2)]
2
Calculate PTK; verify MIC
[AA, ANonce′, SN, msg1]
H
1′
Recalculate TPTK = PRF (PMK, ANonce′, SNonce); store ANonce′ e TPTK
[AA, ANonce, SN+1, msg3, MICPTK(ANonce, SN+1, msg3)]
3
Verify MIC: MICTPTK ≠ MICPTK Errobousmic ! || Discard
Timeout
||
Deauthentication

Figure 7: DoS attack in 4-way handshake phase with TPTK extension.

LLDE is a mechanism that permits, after having obtained the PTK from both sides, encryption of the following instances of 4-way handshake. All messages after the first 4 messages (first instance) are encrypted through the temporary keys of PTK. This approach noticeably reduces the action range of the hacker to the first instance.

Thus, the combined action of PMKID and LLDE reduces the time interval in which the hacker can operate, that is, at the only first 4-way handshake instance, and after sending the first Msg1. Even if the action range of a hacker is very small, the potential risk of DoS attachment is still present. Through correct devices such as scanners, sniffers, and packet generators, the process can be automatised and synchronised and the probability of success can be high.

6. STATIC VERSUS DYNAMIC RESOURCE-ORIENTED SOLUTIONS FOR THE 4-WAY HANDSHAKE

The main issue in the 4-way handshake procedure is the incapability to discriminate the new Msg1 request coming from the real node and the messages generated by H. If the capability of discrimination is introduced, the handshake procedure could be completed. The second issue to be overcome is the *memory exhaustion*. In fact, even if the hacker's messages are discriminated, H could still produce a *DoS flooding* attack. A solution to this second issue can be the avoidance of storing ANonce and PTK for each Msg1 offering the correct working of the 4-way handshake. At first glance, a further control field in the frame format could be added but this approach requires too much change in the original protocol.

In this work a slight change to the terminal is proposed in order to avoid the deauthentication determined by a DoS attack and memory exhaustion after the DoS flooding attack.

6.1. Static standard solution (solution I)

The proposal is easily presented in the following:

(i) on reception of the first message Msg1, S takes 3 actions, it
  (a) generates and stores SNonce;
  (b) computes PTK in the same way provided by the standard protocol;
  (c) creates and sends Msg2 (*no stores ANonce and PTK*);

(ii) on each reception of a new Msg1, S only calculates without storing the novel PTK (PTK′); through this new PTK′ it can produce the MIC value;

(iii) on reception of Msg3, in order to verify the MIC, S computes the PTK again through the SNonce value that has previously memorised and the ANonce value obtained by Msg3; in this way the identification process gives a positive response and the attack attempt is avoided (Figure 8).

The replication of Msg3 by H is not applicable because in this message the MIC field that assures the identity is present.

Also memory exhaustion is avoided because it is sufficient to store only SNonce rather than ANonce and PTK values after any reception of Msg1.

With the proposed application we obtain positive results also in the packet loss scenario. In fact, in the case of Msg2 loss, after timeout, A sends a new Msg1 to S. This message, called Msg1′, contains a new ANonce value (ANonce′) which is now legitimate and so, at the reception of Msg3, it will give positive response (Figure 9).

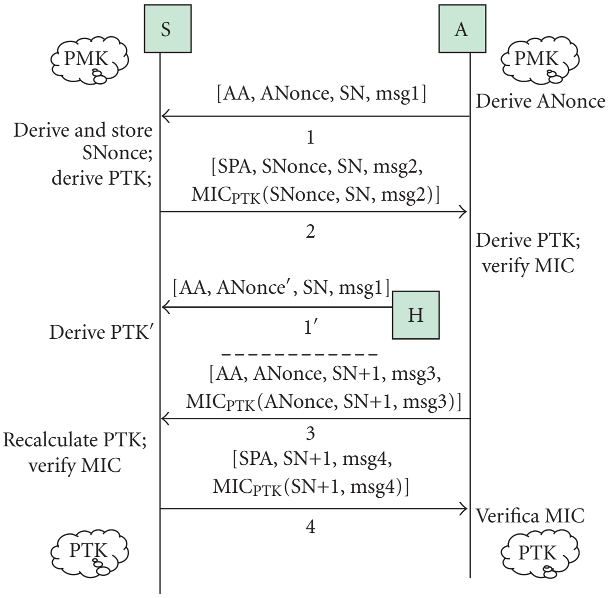

S
A
PMK
PMK
[AA, ANonce, SN, msg1]
Derive ANonce
Derive and store SNonce; derive PTK;
1
[SPA, SNonce, SN, msg2, MIC$_{PTK}$(SNonce, SN, msg2)]
2
Derive PTK; verify MIC
[AA, ANonce′, SN, msg1]
H
Derive PTK′
1′
[AA, ANonce, SN+1, msg3, MIC$_{PTK}$(ANonce, SN+1, msg3)]
3
Recalculate PTK; verify MIC
[SPA, SN+1, msg4, MIC$_{PTK}$(SN+1, msg4)]
4
Verifica MIC
PTK
PTK

Figure 8: Proposal in DoS attack scenario.

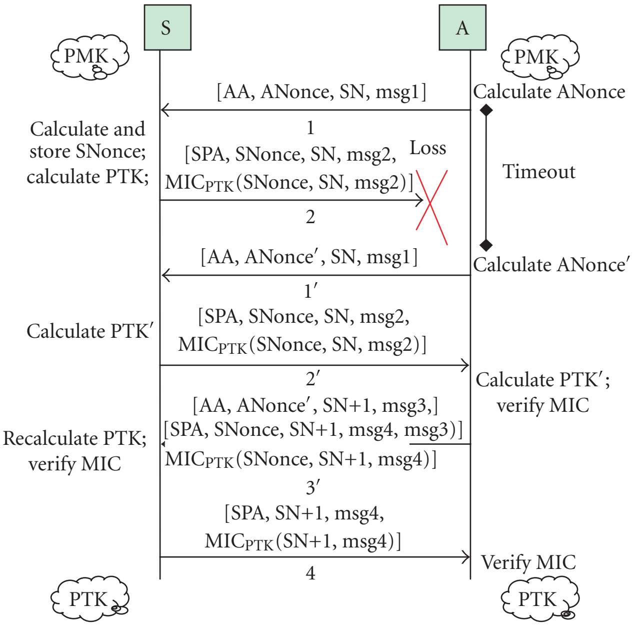

S
A
PMK
PMK
[AA, ANonce, SN, msg1]
Calculate ANonce
Calculate and store SNonce; calculate PTK;
1
[SPA, SNonce, SN, msg2, MIC$_{PTK}$(SNonce, SN, msg2)]
Loss
Timeout
2
[AA, ANonce′, SN, msg1]
Calculate ANonce′
1′
Calculate PTK′
[SPA, SNonce, SN, msg2, MIC$_{PTK}$(SNonce, SN, msg2)]
2′
Calculate PTK′; verify MIC
[AA, ANonce′, SN+1, msg3,]
Recalculate PTK; verify MIC
[SPA, SNonce, SN+1, msg4, msg3)]
MIC$_{PTK}$(SNonce, SN+1, msg4)]
3′
[SPA, SN+1, msg4, MIC$_{PTK}$(SN+1, msg4)]
4
Verify MIC
PTK
PTK

Figure 9: Proposal in packet loss scenario.

However, the proposed solution can occur in a *CPU exhaustion* issue rather than *memory exhaustion* because it forces the recalculation of PTK for each handshake instance (e.g., after the reception of Msg3). This should be avoided for low-power processors.

In this case, it is possible to apply a trade-off policy between the standard IEEE 802.11i and the presented proposal.

6.2. Static solution with trade-off variant (solution II)

In particular, the steps to be followed are presented below.

(i) On reception of the first Msg1, S has to perform the following actions:

(a) generate and store SNonce;
(b) calculate PTK;
(c) create and send Msg2;
(d) store ANonce and PTK.

(ii) On reception of each new message Msg1, S calculates PTK′ in accordance with the standard proposal.

(iii) After the reception of Msg3, S compares the ANonce value in Msg3 with the stored ANonce. If the two ANonce values (ANonce and ANonce$_{\text{Msg3}}$) are the same, S will verify the MIC of Msg3 using the stored PTK. Otherwise if the two ANonce values are different, the PTK will be recomputed and after that the MIC will be verified (Figure 10).

In this way the variant avoids storing ANonce and PTK (memory exhaustion) each time and recomputing PTK after each Msg3 arrival (CPU exhaustion).

6.3. Static solution with trade-off variant and memory release (solution III)

A further change that could be applied to the latter extension is the deallocation of memory space associated with SNonce and ANonce storage if S does not receive new Msg1s. In other terms, if S receives Msg3 after sending Msg2, it can erase SNonce and ANonce values freeing memory space. At this point S can verify the MIC values without checking the ANonce value. This new variant to the proposal should produce lower performance than the first solution (or standard proposal) but better performance than trade-off solution (or proposal with trade-off variant) in the no-hacking scenario.

In case of DoS attack the handshake is the same of the proposal without memory release and so performances should be the same. In order to test the efficacy and the improvement of the proposed solution, a simulation campaign was conducted as presented in the next section.

Figures 11 and 12 represent the last proposal handshakes in no-hacking and in DoS attack scenarios.

6.4. Memory and CPU load aware dynamic solution

Through benefits and drawback analysis of three mechanisms for IEEE 802.11i protocol, it is possible to propose a solution that tries to unify the benefits of the single mechanisms. For example, the supplicant could be equipped of an intelligent software module that monitors system parameters (or network parameters) and on their basis it decides to adopt the mechanism I, II, or III. If the supplicant wants to control the CPU and memory load levels, threshold levels could be introduced and if the device overcomes these levels the system can switch among solution I, II, or III.

Considering the unpredictability of an attacker, both situations of DoS attack and no-hacking are considered.

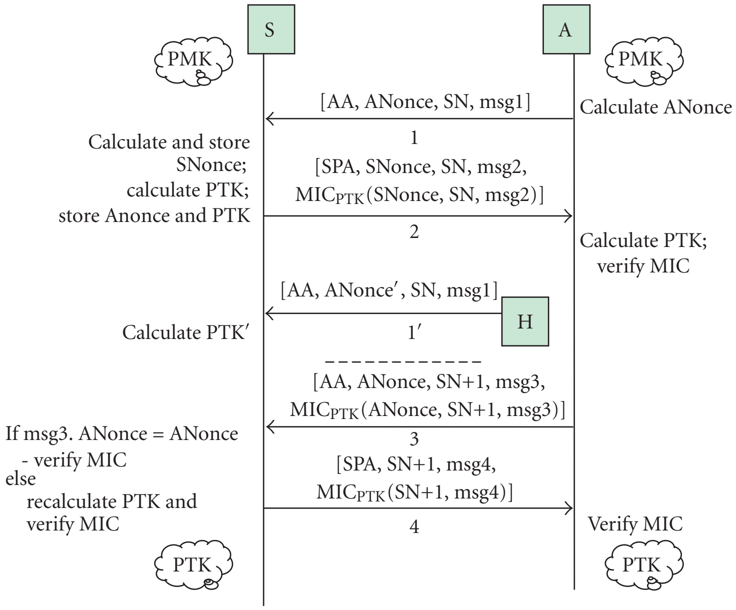

S
A
PMK
PMK
[AA, ANonce, SN, msg1]
Calculate ANonce
1
Calculate and store SNonce; calculate PTK; store Anonce and PTK
[SPA, SNonce, SN, msg2, MIC$_{PTK}$(SNonce, SN, msg2)]
2
Calculate PTK; verify MIC
[AA, ANonce′, SN, msg1]
H
Calculate PTK′
1′
[AA, ANonce, SN+1, msg3, MIC$_{PTK}$(ANonce, SN+1, msg3)]
3
If msg3. ANonce = ANonce - verify MIC else recalculate PTK and verify MIC
[SPA, SN+1, msg4, MIC$_{PTK}$(SN+1, msg4)]
4
Verify MIC
PTK
PTK

Figure 10: Proposal with trade-off variant.

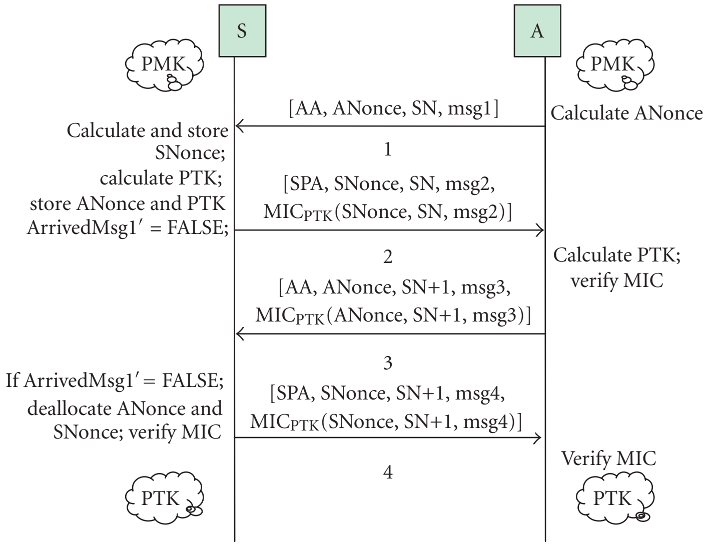

S
A
PMK
PMK
[AA, ANonce, SN, msg1]
Calculate ANonce
1
Calculate and store SNonce; calculate PTK; store ANonce and PTK ArrivedMsg1′ = FALSE;
[SPA, SNonce, SN, msg2, MIC$_{PTK}$(SNonce, SN, msg2)]
2
Calculate PTK; verify MIC
[AA, ANonce, SN+1, msg3, MIC$_{PTK}$(ANonce, SN+1, msg3)]
3
If ArrivedMsg1′ = FALSE; deallocate ANonce and SNonce; verify MIC
[SPA, SNonce, SN+1, msg4, MIC$_{PTK}$(SNonce, SN+1, msg4)]
4
Verify MIC
PTK
PTK

Figure 11: Proposal with trade-off variant and memory release variant in no-hacking scenario.

As previously explained, the solution I avoids the risk of memory exhaustion but it can produce an excessive CPU load.

Thus its usage depends of the device characteristics or device's resource availability. On the other hand, solution II reduces the CPU load requested by the protocol execution to offer a good security level. However this solution needs more memory usage. Thus if the device uses its memory for other operations or it is limited in the memory, it can be unsuitable. The third solution tries to reduce the memory storage when no-hacking scenario is present. Obviously to use the solution III, some mechanism to be aware of the no-hacking scenario should be implemented on the system. This last solution represents a trade-off of two previous solutions if no attack is led to the system. The solution I is the best approach if the device can use a good CPU resource with low memory storage. The solution II is the optimal solution if limited memory availability is present and the device needs to avoid the memory exhaustion issue.

In Table 2, a qualitative evaluation of three solutions with the possible attacks are considered.

However, we have to observe that the network parameters and resources change in the time and if a static scheme such as solution I, II, or III could not be the best solution in

Table 2: Qualitative evaluation of attacks and CPU/mem exaustion risks in three solutions (I, II, and III).

		Avoid DoS attack	Avoid DoS-F attack	Mem load (risk of mem exhaustion)	CPU load (risk of CPU exhaustion)
Solution I	No-hack Scenario	Yes	Yes	—	∗∗
	DoS attack scenario	Yes	Yes	—	∗ ∗ ∗
Solution II	No-hack scenario	Yes	Yes	∗	—
	DoS attack scenario	Yes	Yes	∗	∗
Solution III	No-hack scenario	Yes	Yes	—	—
	DoS attack scenario	Yes	Yes	∗	∗

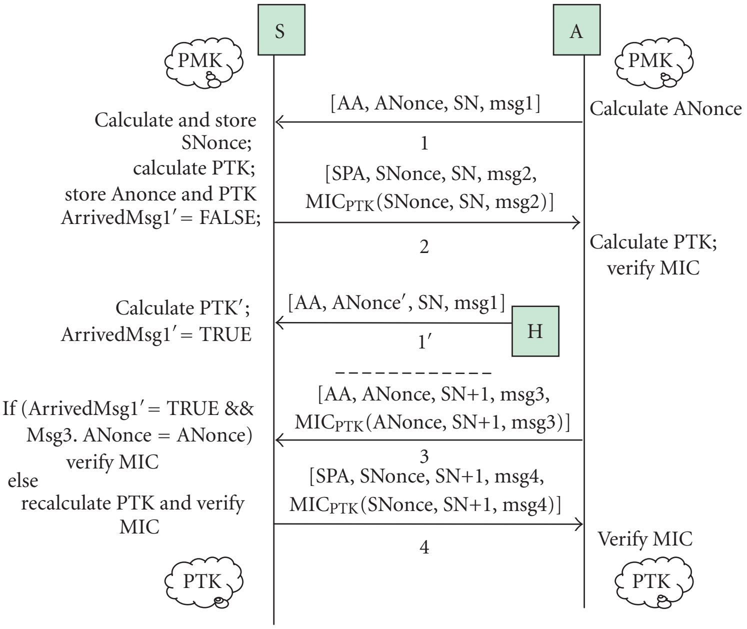

S
A
PMK
PMK
[AA, ANonce, SN, msg1]
Calculate ANonce
Calculate and store SNonce;
calculate PTK;
store Anonce and PTK
ArrivedMsg1′ = FALSE;
1
[SPA, SNonce, SN, msg2, MIC$_{PTK}$(SNonce, SN, msg2)]
2
Calculate PTK;
verify MIC
Calculate PTK′;
ArrivedMsg1′ = TRUE
[AA, ANonce′, SN, msg1]
H
1′
If (ArrivedMsg1′ = TRUE &&
Msg3. ANonce = ANonce)
verify MIC
else
recalculate PTK and verify MIC
[AA, ANonce, SN+1, msg3, MIC$_{PTK}$(ANonce, SN+1, msg3)]
3
[SPA, SNonce, SN+1, msg4, MIC$_{PTK}$(SNonce, SN+1, msg4)]
4
Verify MIC
PTK
PTK

Figure 12: Proposal with trade-off and memory release variant in DoS attack scenario.

a time window. This case could be more suitable to adopt a dynamic approach where solutions I, II, and III can be implemented in the supplicant and they are adopted if some conditions are verified. Thus it is possible to introduce a controller in the mobile device that continuously checks every X millisecond the risk threshold levels associated to the CPU or memory resources. This CPU and memory levels are compared with risk threshold levels that are computed through a system analysis and that can be given by the device manufacturer. For example, it is possible for the supplicant to start the 4-way handshake following the solution I. However if some risk threshold is overcome, it can switch in the other modalities (solution II or III). In Table 3 the switching scenario for CPU and memory loads is presented.

For example, if at time t1 a sensible increase of CPU load is observed, the device will adopt the 4-way handshake of solution II and it will keep this solution also if an overcome of the memory load threshold is verified under DoS attack. After the attack, if the warning levels of the device are under the alert values, it is unuseful to waste memory and it is possible to switch to the solution III. Then, if the CPU or memory load goes under the risk threshold values, solution II or I can be adopted. Flowchart diagram and pseudocode for the dynamic 4-way handshake policy are introduced (see Figure 13 and Algorithm 1).

Table 3: Choice of suitable solution under normal and high CPU and memory threshold values in hacking and no-hacking scenarios.

	Normal mem load	High mem load	—
Normal CPU load	I	I	No-hack scenario
	I	I	DoS attack scenario
High CPU load	II	III	No-hack scenario
	II	II	DoS attack scenario

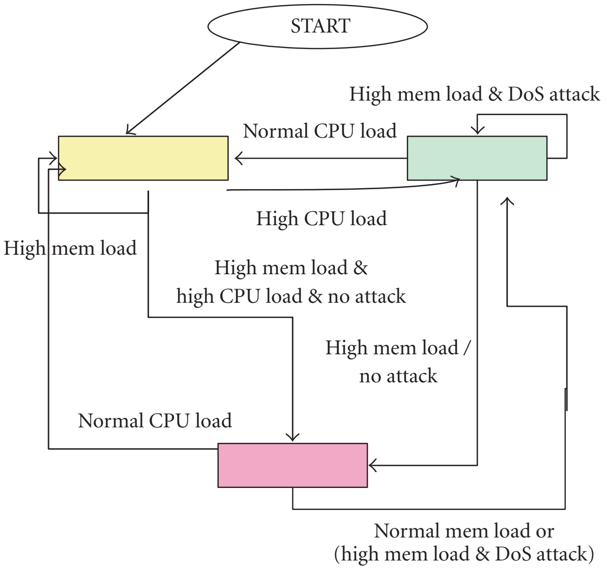


Figure 13: Flowchart diagram of the dynamic resource-oriented approach.

In the pseudocode presented below (see Algorithm 1), it is possible to understand the resource-oriented switching policy among the three solutions. Two thresholds are considered (mem and CPU thresholds). For example, in the example below they are fixed to 100. Two variables (memLoad and cpuLoad) need to be used in order to account the current resource usage. Two functions (readCPUload and readMEMload) are applied to get the current resource availability values.

7. FSA MODELLING

In this paragraph we will attempt to create a mathematical model that can summarise what has been dealt with discursively up to now. The aim is to arrive systematically at the simulation phase, where for "simulation" is intended the emulation of the real system through a preliminary study of the mathematical model representing the system.

The finite state automa (FSA) can therefore provide an optimum model with which to point out the important characteristics of the 4-way handshake protocol and from which to begin a discrete simulation, through java "actors," of the working of the said protocol.

Specifically, since the simulation phase sets the objective of testing the speculative theories formulated in the previous paragraph, the modelling will deal with both the dynamics of the WPA protocol as it is today (obviously with the necessary simplifications already seen) and those of how it would be with the proposed changes (general proposal, compromise proposal, and compromise proposal with variant).

For each of the cases there should be three-orientated graphs: one for the supplicant, one for the authenticator, and one for the hacker.

In reality, since the proposed changes only affect the client's behaviour, it is felt unnecessary to repeat the identical automa of the authenticator and the hacker.

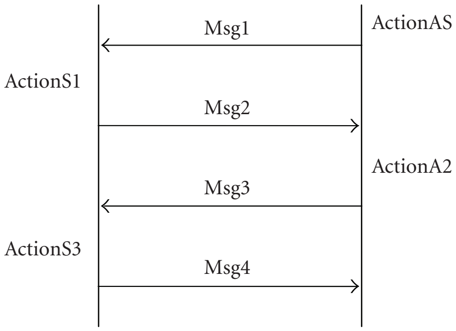


Figure 14: Actions and message exchange of the automata.

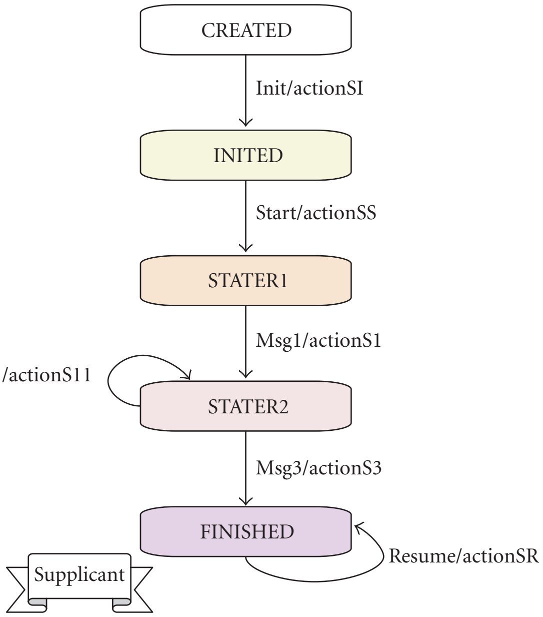


Figure 15: FSA of supplicant in WPA protocol.

To facilitate understanding of the automa refer to Figures 15, 16, 17, and 18.

Before implementing the simulation program, a finite state automata (FSA) formalism was applied to define the entities involved in the communication process and to simulate the actions of possible attackers (e.g., hackers). The FSA are used to outline the characteristics of the 4-way handshake protocols. Specifically, a supplicant, authenticator, and hacker FSA are defined. In this paper, the FSA and their actions are presented.

To formalise the actions carried out by the automa we will use the following notations (see Figure 14):

(i) *actionAS*: action effected by A before sending Msg1;
(ii) *actionS1*: action effected by S immediately after the reception of Msg1;
(iii) *actionA2*: action effected by A after the reception of Msg2;
(iv) *actionS3*: action effected by S after the reception of Msg3.

```
const MEM_thershold = 100;
const CPU_threshold = 100;
bool attack ← false;
Double memLoad ← 0;
Double cpuLoad ← 0;
begin main:
        goto prop1;
end main;
begin prop1:
        attack ← isAttack();
        cpuLoad ← readCPUload();
        memLoad ← readMEMload();
        if (cpuLoad > CPU_threshold)
                if ((memLoad > MEM_threshold) AND (attack = false))
        goto prop3;
        else
        goto prop2;
        else
                goto prop1;
end prop1;
begin prop2:
        attack ← isAttack();
        cpuLoad ← readCPUload();
        memLoad ← readMEMload();
        if ((memLoad > MEM_threshold) AND (attack = false))
                goto3;
        if (cpuLoad ≤ CPU_threshold)
                goto prop1;
        else
                goto prop2;
end prop2;
begin prop3:
        attack ← isAttack();
        cpuLoad ← readCPUload();
        memLoad ← readMEMload();
        if ((memLoad ≤ MEM_threshold) OR ((memLoad > MEM_threshold) AND (attack = true)))
                goto2;
        if (cpuLoad ≤ CPU_threshold)
                goto prop1;
        else
                goto prop3;
end prop3;
```

Algorithm 1: Pseudocode.

At this point the automa of the supplicant, authenticator, and hacker can be represented, along with their respective semantic tables.

Each automa should be read in the following way:

(i) if S is in the CREATED state (starting state) and receives an “init”-type signal, then it carries out action SI and immediately after that it passes to the INITED state;

(ii) if S is in the INITED state and receives a message of the “start”-type, then it effects the actionSS and passes to the STATER1 state;

(iii) if S is in state STATER1 and receives a “Msg1”-type message, then it carries out the actionSS and passes to the STATER2 state;

and so on. For example, for the FSA of the supplicant, see Figure 15; for the authenticator and the hacker see Figures 16 and 17. The actions are outlined in Tables 4–6.

The actions are outlined in Table 4.

In the same way, following the semantic table of the actions, also the automa relative to A represented below is self-explanatory.

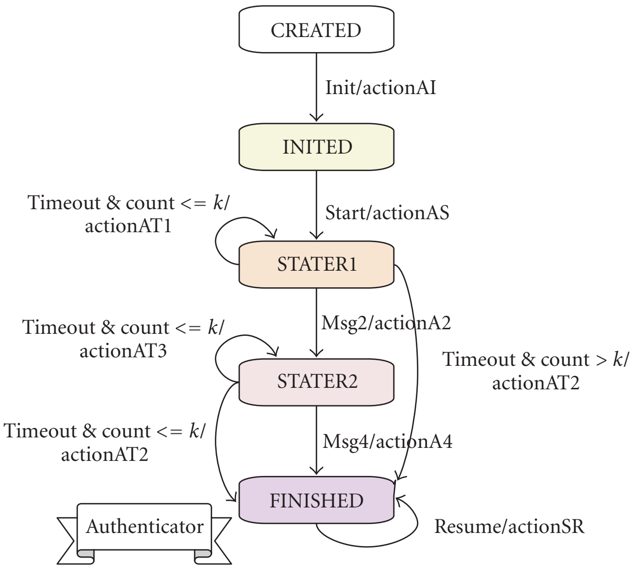


Figure 16: FSA of authenticator in WPA protocol.

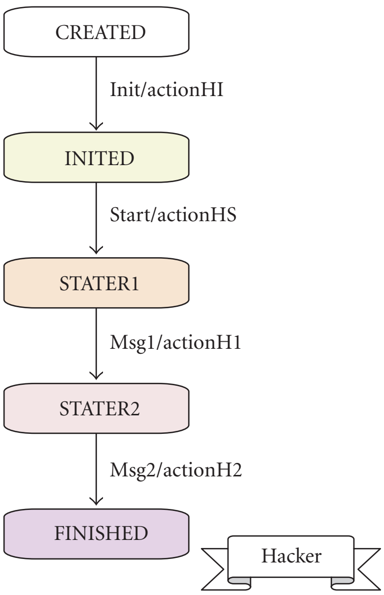
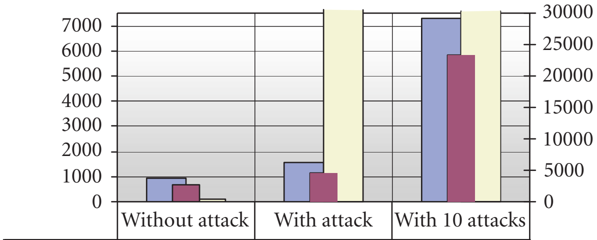


Figure 17: FSA of hacker in WPA protocol.

The automa of hacker H is the most linear since it reflects the ease with which it can mechanically sniff the communication between S and A and introduce itself at a suitable moment (after the sending of Msg2) to carry out the attack (actionH2).

Now let us see how the actions of S and A change themselves according to what is introduced by the 3 proposals presented (Tables 7–9).

The FSA are not shown because they have the same states as the standard WPA automas.



	Without attack	With attack	With 10 attacks
▨ Mem load	948.9	1582.46	7284.5
□ Time exec	110	20129	20449
■ CPU load	2674.97	4485.19	23371.42

Figure 18: WPA protocol performance in no-hacking, DoS attack, and DoS flooding attack scenarios.

Table 4: Actions of supplicant FSA.

ActionSI	Associate a specific A to S
ActionSS	NOP (no operation)
ActionS1	Generate and store SNonce Calculate PTK Send Msg2
actionS11	Calculate PTK Send Msg2
ActionS3	Calculate PTK Verify MIC, (i) if positive ACK is verified, send Msg4, (ii) otherwise, NOP
ActionSR	Send resume

Table 5: Actions of authenticator FSA.

ActionAI	Association of a specific S to A
ActionAS	Generation and storing of ANonce Msg1 forwarding Start of timer
ActionA2	SNonce; storing PTK; calculation MIC validation Msg3 forwarding Start of timer
ActionA4	MIC validation, (i) if positive result, termination; (ii) otherwise, NOP
ActionAT1	ANonce′ genearation Msg1 forwarding Start of the timer
ActionAT2	NOP
ActionAT3	A new forwarding of Msg3 The timer starts
ActionAR	Print results

Table 6: Actions of hacker FSA.

ActionHI	Associate a specific S to A
ActionHS	NOP (no operation)
ActionH1	Store AA and SN
ActionH2	Generate ANonce′ and send Msg1′

Table 7: Actions of supplicant FSA of static standard solution.

ActionSI	Associate to S a specific A
ActionSS	NOP (no operation)
ActionS1	Generate and store SNonce Calculate PTK Store ANonce and PTK Send Msg2
ActionS11	Calculate PTK Store ANonce and PTK Send Msg2
ActionS3	Verify MIC, (i) If positive verification, send Msg4; (ii) otherwise, NOP;
ActionSR	Send resume

Table 8: Actions of supplicant FSA of static solution with trade-off variant.

ActionSI	Associate a specific A to S
ActionSS	NOP (no operation)
ActionS1	Generate and store SNonce Calculate PTK Store ANonce and PTK Send Msg2
ActionS11	Calculate PTK Send Msg2
ActionS3	If ANonce = Msg3. ANonce verify MIC, (i) if positive ack, send Msg4, (ii) otherwise, NOP Else Calculate PTK, Verify MIC, (i) if positive ack, send Msg4, (ii) otherwise, NOP
ActionSR	Send resume

8. PERFORMANCE EVALUATION

After the definition of the FSA, a discrete event simulator was built in JAVA ver.1.4.2 in order to verify the benefits and the drawbacks of the proposed approach.

Starting therefore from an ASF model the code of the *simulazione*.java* file was designed, with which the supplicant, authenticator, and hacker classes were implemented by means of technique and "actors."

Table 9: Actions of supplicant FSA of static solution with memory release.

ActionSI	Associate a specific A to S
ActionSS	NOP (no operation)
ActionS1	Generate and store SNonce Calculate PTK Store ANonce and PTK Send Msg2
ActionS11	ArrivedMsg1′ = TRUE Calculate PTK Send Msg2
ActionS3	If ArrivedMsg1′ = FALSE, then Release ANonce and SNonce, Verify MIC, (i) if positive ack, send Msg4, (ii) otherwise, NOP Else if ANonce = Msg3. Anonce, then Verify MIC, (i) if positive ack, send Msg4, (ii) otherwise, NOP, Else Calculate PTK Verify MIC, (i) if positive ack send Msg4; (ii) otherwise, NOP
ActionSR	Send resume

The dynamic behaviour of the three actors was simulated in 4 different protocol architectures: standard WPA/IEEE 802.11i protocol, other three static solutions (solution I, II, and III). Then, a comparison between static solutions and dynamic resource-oriented approach (solution IV) has been carried out.

8.1. Simulation scenario and parameters

The simulation task is the performance evaluation of the 4-way handshake of WPA and IEEE 802.11i in the case of DoS attack like the scenario defined in Section 5.

The simulation parameters are

(i) *attack result:* it represents the number of completed attacks towards the mobile devices;

(ii) *memory load:*[1] it represents the cost associated to the memory consumption;

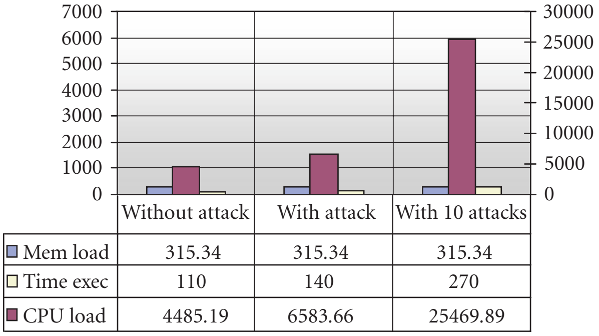
(iii) *CPU load:*[1] it represents the cost associated to the CPU usage;

(iv) *execution time:* it is the time necessary to complete the 4-way handshake procedure.

[1] Client-side. (In our simulation the server-side memory and CPU load has also been calculated, but only the supplicant performances are presented in this work.) Computational time is expressed in milliseconds.

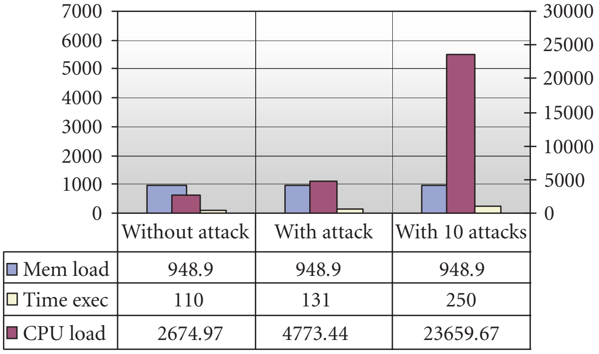
TABLE 10: Costs (mem and CPU loads) associated to the 4-way handshake operations.

Cost associated to the storage (mem load) and computation (CPU load)	Value
Nonce cost associated with memory storage	315
PTK cost associated with the memory storage	318
MIC cost associated with the CPU operations	288
PTK cost associated with the CPU operations	1810



	Without attack	With attack	With 10 attacks
🟦 Mem load	315.34	315.34	315.34
□ Time exec	110	140	270
🟥 CPU load	4485.19	6583.66	25469.89

FIGURE 19: Solution I performance in no-hacking, DoS attack, and DoS flooding attack scenarios.



	Without attack	With attack	With 10 attacks
🟦 Mem load	948.9	948.9	948.9
□ Time exec	110	131	250
🟥 CPU load	2674.97	4773.44	23659.67

FIGURE 20: Performance of trade-off solution (solution II) in no-hacking, DoS attack, and DoS flooding attack scenarios.

There are 12 possible cases that are implemented and they are described below.

(i) Standard protocol (WPA/IEEE 802.11i) without attack, WPA/IEEE802.11i with DoS attack, and WPA/IEEE 802.11i with DoS flooding attack.
(ii) Solution I without attack, with DoS attack, and with DoS flooding attack.
(iii) Solution II without attack, with DoS attack, and with DoS flooding attack.
(iv) Solution III without attack, with DoS attack, and with DoS flooding attack.

The simulation parameters adopted in our environment are presented in Table 10.

Also if the value above has been arbitrarily chosen, the approach has a general validity because the cost associated with the CPU and memory consumption need to be referred to the availability of memory and CPU. This availability can change in a mobile device depending on the other operations that device is performing. Thus it is important to focus on the possibility to reduce the resource availability for a prefixed cost associated with CPU or memory storage operations. In order to validate proposals, simulation data are represented in histograms.

Specifically, with reference to modelling by means of FSA, it is easy to understand that the critical operations which affect performances of the client-side protocol are without doubt “actionS1,” “actionS3,” and “actionS11;” in these in fact the greatest memory and CPU loads occur.

8.2. Simulation results

Simulation results depicted in Figure 18 show the performance of the WPA/IEEE 802.11i standard protocol. It is possible to observe the positive result of both DoS attacks (simple DoS and DoS flooding) and the consequent system degradation in terms of memory and CPU loads: the memory load doubles in DoS attack case and it is 8 times the initial memory load in the case of a DoS flooding attack (with 10 flooding messages); the execution time changes from 110 milliseconds (with no attack) to 20000 milliseconds in the DoS attack scenario and this is reputable to the positive result of the attack.

As can be seen in Figure 19, the solution I presents a better performance than standard protocol (WPA/IEEE 802.11i) in terms of memory load with and without attacks. The memory load value is constant in all the scenario because the operations that S effects are similar. Good improvements can be verified also for the execution time in the case of intrusion because DoS attack and DoS flooding attack are now avoided. However, the CPU load values increase as shown in Figure 19 and as outlined in the previous section.

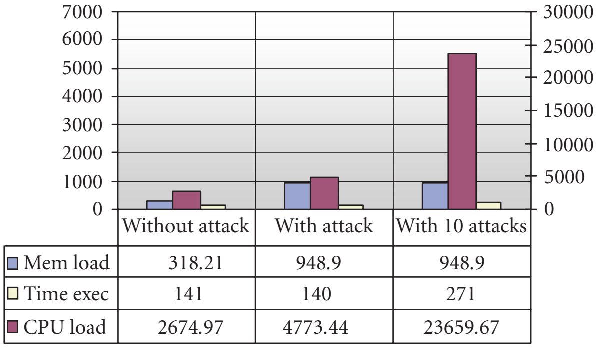
Thus, in order to avoid CPU exhaustion in the case where the hacker can attempt DoS flooding attacks with a high rate, trade-off (solution II) is considered and simulation results are presented in Figure 20.

Memory consumption is lower than the standard proposal in the hacking scenario and the CPU load is maintained low as in the standard WPA/IEEE 802.11i protocol.

In other terms, the proposal with trade-off variant offers the following advantages:

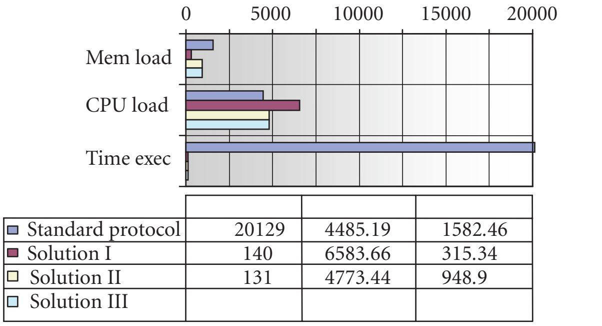
(i) DoS attack and DoS flooding attack are avoided;
(ii) CPU load equals that of the WPA/IEEE 802.11i standard 4-way handshake;
(iii) memory load independent of attack type;
(iv) execution time similar to that of standard proposal.

However, solution II does not make a distinction in the memory occupation if the device is under attack or if no attack is present. This suggested a variant to solution II should be proposed that consists in the memory release if no-attack scenario is considered. This kind of situation can be detected



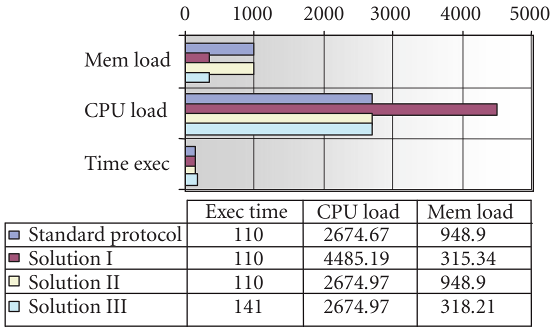
	Without attack	With attack	With 10 attacks
■ Mem load	318.21	948.9	948.9
□ Time exec	141	140	271
■ CPU load	2674.97	4773.44	23659.67

FIGURE 21: Performance of trade-off with memory release solution (solution III) in no-hacking, DoS attack, and DoS flooding attack scenarios.



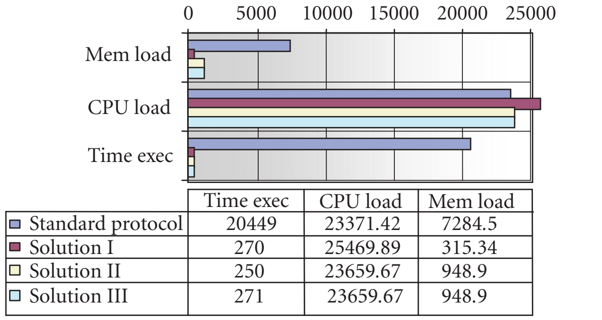
■ Standard protocol	20129	4485.19	1582.46
■ Solution I	140	6583.66	315.34
□ Solution II	131	4773.44	948.9
□ Solution III			

FIGURE 23: Simulation parameter values of standard protocol, standard proposal, trade-off proposal, and trade-off proposal with memory release variant in a DoS attack scenario.



	Exec time	CPU load	Mem load
■ Standard protocol	110	2674.67	948.9
■ Solution I	110	4485.19	315.34
□ Solution II	110	2674.97	948.9
□ Solution III	141	2674.97	318.21

FIGURE 22: Standard protocol simulation parameter values, standard proposal simulation values, trade-off proposal, and trade-off with memory release variant proposal simulation values in a no-hacking scenario.



	Time exec	CPU load	Mem load
■ Standard protocol	20449	23371.42	7284.5
■ Solution I	270	25469.89	315.34
□ Solution II	250	23659.67	948.9
□ Solution III	271	23659.67	948.9

FIGURE 24: Simulation parameter values of standard protocol, standard proposal, trade-off proposal, and trade-off with memory release variant proposal in a DoS flooding attack scenario.

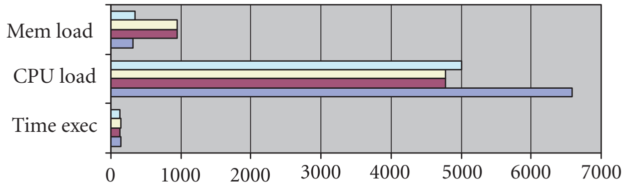
by the mobile device after a monitoring of the 4-way handshake procedure. If the previous handshake procedure had been executed without observing malicious messages for a certain amount of time, the risk threshold of the device can be reduced and a no-attack scenario can be considered. The threshold setting and the no-attack scenario evaluation are outside the scope of this paper. Anyway, through this consideration, further advantages offered by proposed solution III can be verified in a lower memory load in the no-hacking scenario such as the one depicted in Figure 21. The other parameters reflect the same performance as the trade-off proposal without memory release.

Thus, the previous graphics above, permit quantification of the CPU load, memory load, and 4-way handshake execution time when the three solutions (I, II, and III) are adopted.

Now, in Figures 22, 23, and 24 data have been reelaborated in order to give a more complete picture of parameter values related to the standard WPA/IEEE 802.11i protocol, the standard proposal (solution I), the trade-off proposal (solution II), and the trade-off with memory release variant proposal in no-hacking scenarios (solution III).

In Figure 22 where no-hacking scenario is considered, if the most important resource to be considered is the CPU load, it is possible to observe the best performance of WPA/IEEE 802.11 protocol without the He and Mitchell extension (solution I). On the other hand, if the memory load is considered the most important, solution I and solution III perform better. If we have a wide resource availability, and the execution time can be important, WPA/IEEE 802.11i without enhancements, solutions I, and II are the best. Obviously the WPA/IEEE 802.11i protocol without He and Mitchell extension can not be considered because the system would be weak to a DoS attack. Thus, a DoS attack and DoS flooding attacks have been simulated and their effects have been depicted in Figure 23. Also in this case, depending on the importance given to resources consumption, in some case solution I can be the best or some times solution II or solution III.

In this way, simulation campaigns, suggest that we should propose a dynamic approach such as that explained in Section 6 that tries to combine these distinct solutions. Just to give an idea of the improvements introduced by the dynamic approach, a graph, where DoS attack is considered, with the static and dynamic resource-oriented approaches is shown in Figure 25. In this case, solution IV offers the best performance in terms of mem load and execution time maintaining a low CPU load (comparable with solutions II and III).



	Time exec	CPU load	Mem load
□ Solution IV	134	5012	340.23
□ Solution III	140	4773.44	948.9
■ Solution II	131	4773.44	948.9
■ Solution I	140	6583.66	315.34

FIGURE 25: Simulation parameter values of static solutions (I, II, and III) and dynamic solution (IV) in a DoS flooding attack scenario.

9. CONCLUSIONS

Two well-known security protocols WPA and IEEE 802.11i are evaluated. WPA and IEEE 802.11i are robust to a lot of attacks and they overcome the security bugs of WEP protocol. However, in some specific scenarios, these protocols are not able to avoid a DoS attack. After defining the potential risk situation, which can occur in the last phase of the authentication process (the 4-way handshake), three static solutions that avoid the attack and realise different CPU and memory consumptions are investigated and evaluated through simulations. Simulation results validate the proposal to enhance the WPA protocol and avoid a DoS attack and DoS flooding attack. However, efficiency in terms of CPU load and memory exhaustion depends on the specific adopted solution. Specifically, we suggest to adopt a dynamic resource-oriented approach that tries to get the best behaviour of the three distinct solutions. This mechanism is based on CPU and memory load thresholds and it can meet the different resource availabilities of mobile devices during the data transmission.

REFERENCES

[1] J. Edney and W. A. Arbaugh, *Real 802.11 Security: WiFi-Protected Access and 802.11i*, Addison Wesley, New York, NY, USA, 2003.

[2] A. Mishra and W. A. Arbaugh, “An initial security analysis of the IEEE 802.1X standard,” Tech. Rep. CS-TR-4328, University of Maryland, College Park, Md, USA, February 2002.

[3] A. Mishra, N. L. Petroni Jr., W. A. Arbaugh, and T. Fraser, “Security issues in IEEE 802.11 wireless local area networks: a survey,” *Wireless Communications and Mobile Computing*, vol. 4, no. 8, pp. 821–833, 2004.

[4] W. Stallings, *Cryptography and Network Security*, Prentice Hall, Englewood Cliffs, NJ, USA, 3rd edition, 2003.

[5] W. A. Arbaugh, N. Shankar, J. Wang, and K. Zhang, “Your 802.11 network has no clothes,” in *Proceedings of the 1st IEEE International Conference on Wireless LANs and Home Networks*, Suntec City, Singapore, December 2001.

[6] IEEE Standard for Information technology—Telecommunications and Information exchange between systems—Local and metropolitan area networks - Specific requirements, Part 11, Amendment 10: Medium Access Control (MAC) Security Enhancements, IEEE Std 802.11i-2005.

[7] N. Borisov, I. Goldberg, and D. Wagner, “Intercepting mobile communications: the insecurity of 802.11,” in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MOBICOM ’01)*, pp. 180–188, Rome, Italy, July 2001.

[8] C. He and J. C. Mitchell, “Analysis of the 802.111 4-way handshake,” in *Proceedings of the ACM Workshop on Wireless Security (WiSe ’04)*, pp. 43–50, Philadelphia, Pa, USA, October 2004.

[9] 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)*, San Diego, Calif, USA, February 2005.

[10] J. Bellardo and S. Savage, “802.11 Denial of service attacks: real vulnerabilities and practical solutions,” in *Proceedings of the 12th USENIX Security Symposium*, Washington, DC, USA, August 2003.

[11] IEEE Standard 802.11-1999, Information technology—Telecommunications and Information exchange between systems—Local and metropolitan exchange between systems—Local and metropolitan area networks—Specific requirements—Part11: Wireless LAN Medium Access Control and Physical Layer Specifications,1999.

[12] J. S. Park and D. Dicoi, “WLAN security: current and future,” *IEEE Internet Computing*, vol. 7, no. 5, pp. 60–65, 2003.

[13] W. A. Arbaugh, “An inductive Chosen Plaintext Attack, against WEP/WEP2,” Presentations to IEEE 802.11 TGi, May 2001.

[14] S. Fhurer, I. Mantin, and A. Shamir, “Weaknesses in the key scheduling algorithm of RC4,” in *Proceedings of the 8th Annual Workshop on Selected Areas in Cryptography (SAC ’01)*, Toronto, Canada, August 2001.

[15] V. Moen, H. Raddum, and K. J. Hole, “Weaknesses in the temporal key hash of WPA,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, no. 2, pp. 76–83, 2004.

[16] B. Adoba, “WEP2 Security Analysis,” IEEE doc.:802.11-00/253, May 2001, http://www.cs.umd.edu/waa/attack/frame.htm.

[17] C. Rigney, S. Willens, A. Rubens, and W. Sympson, “Remote Authentication Dial In User Service (RADIUS),” RFC 2865, June 2000.

[18] P. R. Calhoun, S. Farrell, and W. Bulley, “Diameter CMS Security Application,” March 2002, http://www.diameter.org/drafts/latest/draft-ietf-aaa-diameter-cms-sec-04.txt.

[19] C. L. Schuba, I. V. Krsul, M. G. Kuhn, E. H. Spafford, A. Sundaram, and D. Zamboni, “Analysis of a denial of service attack on TCP,” in *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 208–223, Oakland, Calif, USA, May 1997.

[20] CERT, “DoS Attack,” http://www.cert.org/tech_tips/denial_of_service.html.

[21] D. B. Faria and D. R. Cheriton, “DoS and authentication in wireless public access networks,” in *Proceedings of the ACM Workshop on Wireless Security (WiSe ’02)*, pp. 47–56, Atlanta, Ga, USA, September 2002.

[22] R. Moskovitz, “Weakness in Passphrase Choice in WPA Interface,” November 2003, http://wifinetnews.com/archives/002452.html.

Floriano De Rango was born in Cosenza (CS), Italy, in 1976. He received the degree in computer science engineering in October 2000, and a Ph.D. degree in electronics and communications engineering in January 2005, both from the University of Calabria, Italy. From January 2000 to October 2000 he worked in the Telecom Research Lab CSELT in Turin as Visiting Scholar Student. From March 2004 to November 2004 he was a Visiting Researcher at the University of California at Los Angeles (UCLA). Since November 2004 he joined the DEIS, University of Calabria, as a Research Fellow. He served as a Reviewer of VTC'03, ICC'04, WCNC'05, Globecom'05, WTS'05, Wireless-COM'05, IEEE Communication Letters, and JSAC. His interests include satellite networks, IP QoS architectures, adaptive wireless networks, and ad hoc networks.

Dionigi Cristian Lentini was born in Mottola (TA), Italy, in 1980. In July 2005 he graduated in computer science engineering (course in electronic and telecommunications) from the University of Calabria, Italy, where he began the research activity. He was qualified for engineering profession in January 2006 and he joined the Engineers Society of Province of Taranto, Italy. Since October 2005 he works in Capgemini SpA., in Rome, as an Analist Consultant Programmer for Hi-Tech/Space and Defence Services unit. Web and RIA applications programming, relational DB design, network security, and cryptographic algorithms and protocols represent some of his major interests in the technology field. Webpage: http://www.ingegnere.135.it.

Salvatore Marano graduated in the Department of Electronics Engineering at the University of Rome in 1973. In 1974 he joined the Fondazione Ugo Bordoni. Between 1976 and 1977 he worked at ITT Laboratory in Leeds, United Kngdom. Since 1979 he has been an Associate Professor at the University of Calabria, Italy. His research interests include performance evaluation in mobile communication systems.