EURASIP Journal on Information Security
a SpringerOpen Journal

RESEARCH Open Access

# Protected control packets to prevent denial of services attacks in IEEE 802.11 wireless networks

Mina Malekzadeh*, Abdul Azim Abdul Ghani and Shamala Subramaniam

**Abstract**

Denial-of-service (DoS) attack exploits inherent limitation of resources in wireless networks in attempt to overwhelm and exhaust their finite capacity. In wireless networks, clear-text form of control packets (CP) exhibits a security flaw that can be exploited by attackers to render the networks incapable of providing normal services. While these attacks are quite damaging in terms of consuming available processing and bandwidth resources, they are easy to conduct against the wireless networks. In this study, we propose two distinct models to prevent wireless DoS and replay attacks based on trust in CP for IEEE 802.11 wireless networks. The first model is based on original HMAC-SHA1 algorithm and the second one is based on a proposed modified HMAC-SHA1 (M-hmac) algorithm. Both models are implemented and the results are obtained and evaluated based on a number of metrics. The results show that the two models successfully prevent both wireless DoS and replay attacks. In addition, the newly proposed M-hmac algorithm provides better network performance in term of the metrics.

**Keywords:** DoS attacks, hash functions, modified HMAC-SHA1, replay attack, control packets

## 1. Introduction

There are various types of security protocols in wireless networks that protect data packets during transmission such as WEP, WPA, and 802.11i. Despite providing different levels of security, these security protocols are not able to protect control packets (CP). Consequently, the CP including RTS, CTS, ACK, CF-End, and CF-End-ACK are transmitted in clear-text form. Wireless CPs contain a 2-byte duration field with 32767 μs maximum value used to set the network allocation vector (NAV). The field shows the time that the channel will be kept busy by the originator node. During this time, other nodes are not allowed to transmit and must wait until NAV reaches zero. Because CP are not protected, it is possible for the attackers to generate false CP with large duration values in order to trigger denial-of-service (DoS) attacks. The intention is to overload the target wireless networks and crash the systems by quickly consuming the networks' capacities.

The attackers may continuously transmit false CP to the target wireless network within short intervals. The target network has to accept the false CP because the 802.11 standard does not provide any mechanism to distinguish the true CPs from the false [1-3]. Reception of large number of false CP within a short time will quickly consume the network resources until the access point (AP) becomes overloaded and crashed. When this happens, the AP has to disconnect all the users who are at the time connected to the network as it comes to a complete halt and shutdown. While these attacks are easy to implement, they have highly negative impact so that completely shut down the wireless networks and significantly degrade the quality of services.

In this study, two distinct models are proposed to distinguish the trust-based CP belonging to the authorized users from the false CP belonging to the attackers. The models enable the recipients to investigate the CP and analyze information contained in their security elements in order to accept trusted CP and discard false CP. The proposed models support all types of CP and are able to prevent both DoS attacks and replay attacks with sufficient level of security.

The remainder of this article is organized as follows. Related studies are discussed in Section 2. Section 3 presents the structures of the two proposed models. Section 4 details out the experimental setup to implement the

* Correspondence: minarz@gmail.com
Faculty of Computer Science and Information Technology, University Putra Malaysia, 43400 UPM Serdang, Selango, Malaysia

models. Section 5 presents the experimental results and discussion. Finally, Section 6 concludes the article.

## 2. Related studies

In order to address the wireless DoS attacks, different schemes have been proposed. The schemes can be classified into three general methods, namely, cryptographic, detection, and NAV validation methods. Table 1 summarizes the strategies adopted by these schemes and highlights the corresponding weaknesses.

Despite all the benefits getting from the proposed schemes in Table 1, there are still a number of notable weak points. Each scheme only addresses a specific issue from the entire security problem, therefore, requires complementary solutions. The schemes are still vulnerable to replay attacks. In addition, ignoring protection of contention-free CP by these schemes keeps the DoS attacks a threat against wireless networks. These drawbacks lead to a need to provide a security model so that while it is able to prevent the DoS and replay attacks by protecting all types of CP, it does not impose significant security cost to the wireless networks compared to the 802.11 standard model which are the main focus of this study.

## 3. Overall structure of the proposed models

In order to prevent wireless DoS and replay attacks, we propose two distinct models that provide secure control packets (SCPs). The first model employs original HMAC-SHA1 as the underlying authentication algorithm, which we refer to as O-hmac. Because it is based on O-hmac, we named the first proposed model as SCP-O. For the second proposed model, there are two steps involved. First, we propose some modifications over the HMAC-SHA1 and we refer to the modified version as M-hmac.

The main purpose of the M-hmac is to reduce the security cost and communication overheads of O-hmac while at the same time optimizing its performance. M-hmac is then used as the underlying authentication algorithm of the second proposed model, which is called SCP-M.

Implementation of the SCP-O involves three main parts which are proposed key derivation algorithm (KDA), creating two new security elements, and proposed replay-preventing mechanism. In contrast, implementation of the SCP-M in addition to these three parts requires one additional part to develop the proposed M-hmac. These four parts are presented in Figure 1 while the overall structure and implementation process of each part are explained in the following sections.

### 3.1. Proposed KDA

The implementation of both SCP-M and SCP-O models requires a key, while protection of the key that is shared

**Table 1 Wireless DoS attacks prevention schemes**

| Authors | Method | Strategy | Weaknesses |
|---|---|---|---|
| Rachedi and Benslimane (2008) [1] | Cryptographic | Add transmitter address to CTS and ACK packets; authentication using hmac-MD5 or hmac-sha1 of 80-160b | The model is incapable of preventing replay attacks and contention-free DoS attacks |
| Khan and Hasan (2008) [2] | Cryptographic | Change CRC-32 to CRC-16 and use PRF-16 from hmac-sha1 | The model has three drawbacks: modifying CRC which is used for all other types of packets, short security field, and inability to prevent replay attacks |
| Bicakci and Tavli (2009) [3] | Cryptographic | Symmetric algorithm to encrypt/hash function to authenticate the CP | The model does not specify the symmetric encryption or authentication algorithms |
| Qureshi et al. (2008) [19] | Cryptographic | Encryption using PRF-160 bits | The model only protects polling CP while still DoS attacks are possible using other types of CP |
| Chen et al. (2007) [20] | Detection | DoS attacks detection using CUSUM | The model only detects RTS and CTS DoS attacks while incapable of detecting DoS using other CP. Besides it is unable to prevent the DoS attacks |
| Sugantha and Shanmugavel (2006, 2005) [21,22] | Detection | Keep track of statistical distribution pattern of CP when a uniform pattern only belongs to attacker | The model provides only detection while unable to prevent the DoS attacks |
| Zhang et al. (2008) [23] | Detection | ENAV to give enough time to sender of data fame to receive ACK packet | The model only protects ACK packet while DoS attack is possible by other types of CP |
| Negi and Rajeswaran (2005) [24] | Validation | Introduce a new packet called CTSR to revoke the NAV if no data is sensed after RTS or CTS transmission | Unprotected new CTSR frame causes a new DoS attacks itself. Besides, the model is not able to prevent DoS attacks using other types of CP |
| Chen et al. (2003) [25] | Validation | Two timers as RTS-DATA and CTS-ACK to check reception of data and ACK packets, respectively | The model is incapable of preventing DoS attacks using contention-free CP |
| Bellardo and Savage (2003) [26] | Validation | Place limit on duration value of CP: ACK duration must be zero, discarding RTS if data frames is not sensed, ignoring isolated CTS packets | The model does not specify prevention of contention-free DoS attacks. Besides ignoring CTS packets while they may belong to hidden nodes can significantly degrade wireless network performance |

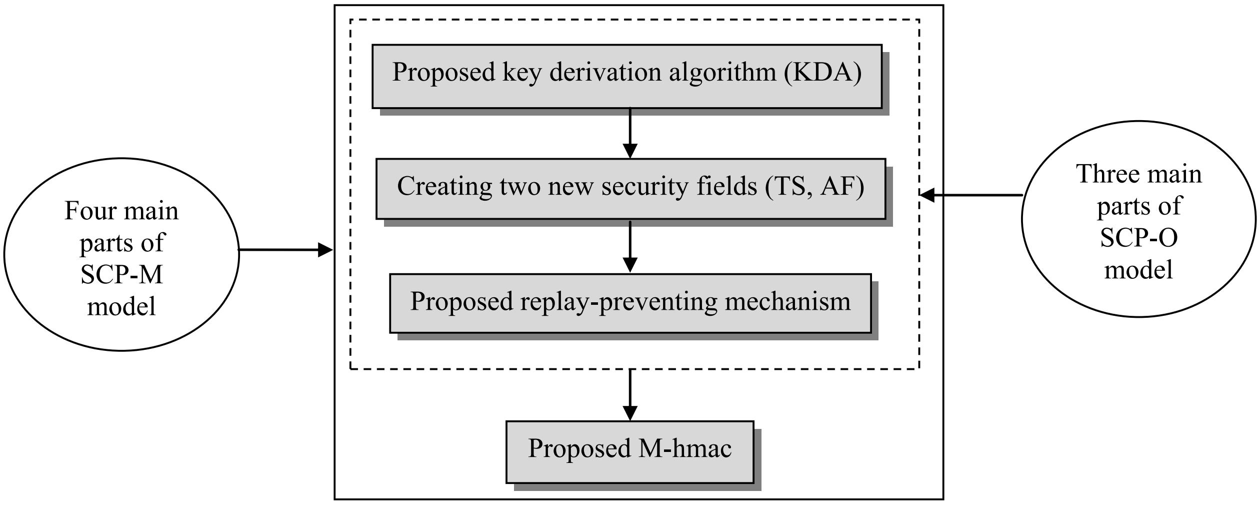

Four main parts of SCP-M model
Proposed key derivation algorithm (KDA)
Creating two new security fields (TS, AF)
Proposed replay-preventing mechanism
Proposed M-hmac
Three main parts of SCP-O model

**Figure 1 Corresponding parts in the SCP-O and SCP-M models**.

between the communication parties is an important consideration. We assume the users in wireless network to be equipped with a shared key $K$ of length $k$ bits. Nonetheless, applying the main shared key directly through the communications is insecure [4]. Based on this argument, we propose a new KDA used during the SCP transmissions. The proposed KDA hashes the main shared key with two rationales behind the design.

First, the length of the hashed password is longer than length of the original key, thus the hashed key is harder to compromise. If the length of original key is $k$ bits, then there are $2^k$ possible values that the attacker has to expose. However, for a hashed key with $n$ bits where $n > k$, the number of attempts for the attacker to break the key is $2^n$, where $2^n > 2^k$. This means that the attacker will be dealing with a more difficult task to reveal the value of the hashed key.

Second, hash functions are one-way algorithms. Therefore, finding the original key from the hash results demands high effort which can make it practically infeasible [4].

On the other hand, although hashing a key can be useful in strengthening security [5] as compared to using a shorter plaintext key, it is still possible for the attacker to compromise the key using the rainbow table attacks [6]. To avoid this issue, we incorporate two additional cryptographic salts to the original shared key before the key is being hashed. A cryptographic salt is additional information that is added to the main key before hashing takes place [7] which makes the key disclosure attacks (e.g., rainbow table attack) slower and more difficult [8].

In this study, we apply two cryptographic salts. The first cryptographic salt is referred to as *cryptsalt1* using the network name (SSID) as the value. The second cryptographic salt is referred to as *cryptsalt2* using the MAC address of AP (BSSID) as the value. The *cryptsalt1* is appended to the original shared key to generate a *cryptographic-salted-key* as follow.

$$\begin{aligned}&Cryptographic - salted - key = Original\ shared\ key\\&\parallel\ cryptsalt1\\&where \parallel\ is\ concatenation\end{aligned}$$

Once the salted cryptographic key is generated, the following processes are performed to generate the final key (FK) for both SCP-O and SCP-M models to be used during subsequent SCPs transmissions instead of the original key.

### *3.1.1. Generating FK for the SCP-O model*

To generate the FK for the SCP-O model which is called FK-SCP-O, we consider *cryptsalt2* as the input to the O-hmac algorithm with the *cryptographic-salted-key* as the key to generate FK-SCP-O key with 160 bits length as follow.

$$\begin{aligned}&FK - SCP - O = O - hmac\ [(original\ shared\ key\\&\parallel\ cryptsalt1),\ cryptsalt2]\end{aligned}$$

### *3.1.2. Generating FK for the SCP-M model*

The process of the key generation for the SCP-M model is different from key generation for the SCP-O model. To generate the FK for the SCP-M model which is called FK-SCP-M, we employ the first round of the M-hmac algorithm. Nevertheless, we exclude the second round of M-hmac in the key derivation because of the fact that a longer key poses harder constraints to decode the key value by the attackers. The length of the output in the first round of M-hmac is longer than its second round, which leads us to use only the first round of M-hmac.

We enter *cryptsalt2* as the input to the first round of M-hmac algorithm with the *cryptographic-salted-key* as the key to generate FK-SCP-M key with a length of 160 bits as follow.

$$FK - SCP - M = First\ round\ of\ M - hmac\ [(original\ shared\ key \parallel cryptsalt1),\ cryptsalt2]$$

The overall process of the proposed KDAs is described in Figure 2.

### 3.2. New security fields: TS and AF

To ensure security, imposing extra overheads to the networks is unavoidable [9]. However, because of the fact that wireless networks have limited resources [10], it is required to keep the overheads as small as possible. In this study, we define two new security fields, TS and AF, to construct the SCP as follows.

- ***TS field***: This field carries creation time of the CP and is used to verify their freshness. The TS field is 4 bytes and is appended at the end of all CP before the FCS field.

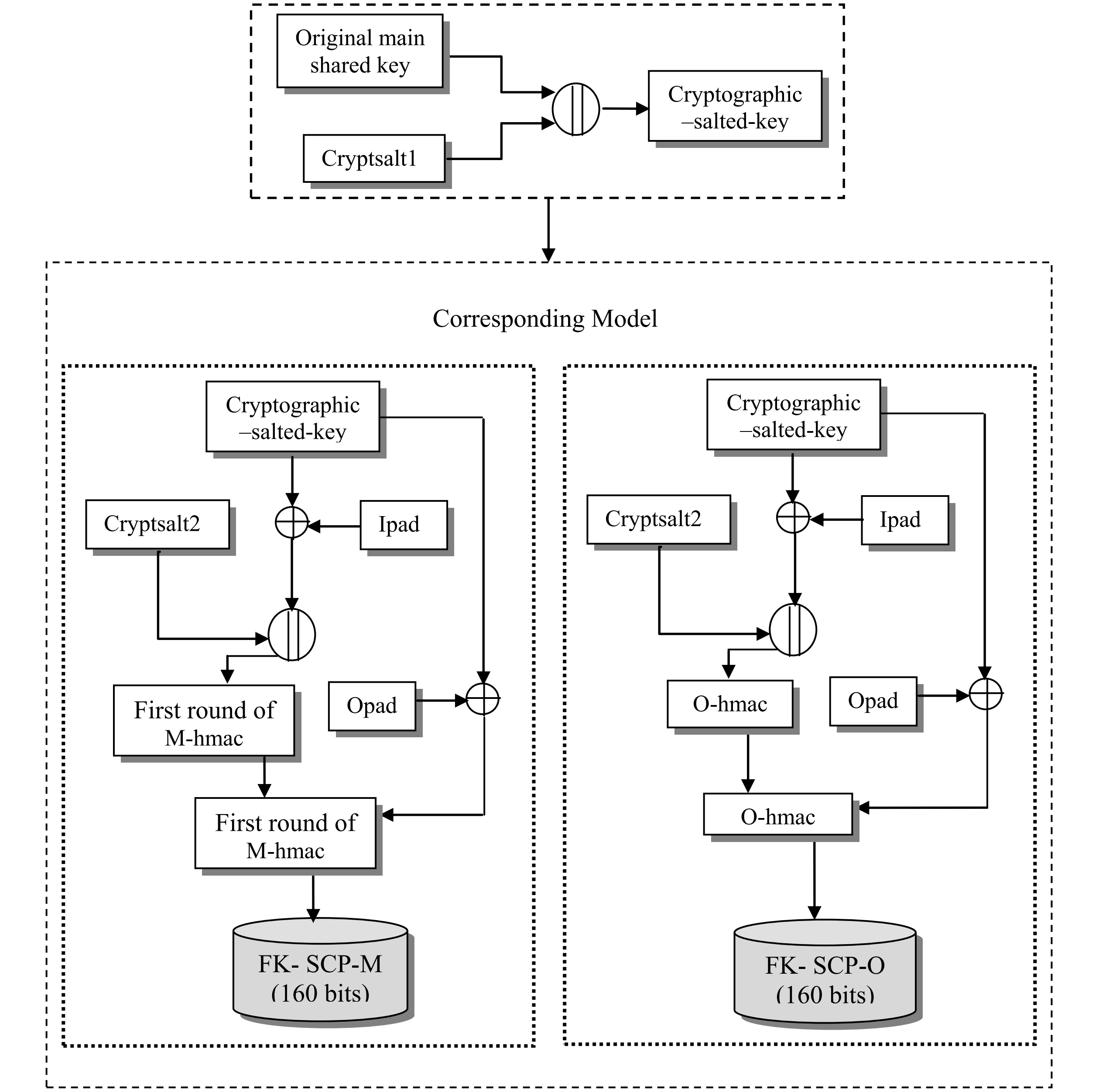


**Figure 2** Proposed KDA for the SCP-O and SCP-M models.

- ***AF field***: In order to verify integrity of the received CP and identify the validity of their originator, a new field called the authenticator field (AF) is created. This field is attached following the TS field in all types of CP to carry output results of the M-hmac and O-hmac. The length of the AF is 20 bytes in the SCP-O model as compared to 12 bytes in the SCP-M model.

### 3.3. Proposed replay-preventing mechanism

We design a replay attack protection mechanism based on threshold timeout window in order to verify freshness of the received CP. The replay attack protection mechanism is accomplished by tagging each outgoing CP with an identifier, which is the creation time of that particular CP. We formalize and determine five distinct threshold timeout windows, which are called $\mathrm{TO_{RTS}}$, $\mathrm{TO_{CTS}}$, $\mathrm{TO_{ACK}}$, $\mathrm{TO_{CF\text{-}End}}$, and $\mathrm{TO_{CF\text{-}End\text{-}ACK}}$ related to the five SCP. They determine maximum acceptable age of the corresponding SCP as presented in Figure 3.

In order to determine these five threshold timeout windows, a number of IEEE 802.11 standard notations [11] are used as presented in Table 2.

Now we define $\mathrm{T_{SCP}}$, as the required time to transmit the entire SCP including their physical header as follow.

$$T_{SCP} = \frac{L_{SCP}}{B_r} + \frac{PHY_h}{PHY_r} \tag{1}$$

In Equation 1, $L_{\mathrm{SCP}}$ is the length of the SCP and $T_{\mathrm{SCP}}$ is considered for all types of CP as $T_{\mathrm{RTS}}$, $T_{\mathrm{CTS}}$, $T_{\mathrm{ACK}}$, $T_{\mathrm{CF\text{-}End}}$, and $T_{\mathrm{CF\text{-}End\text{-}ACK}}$, which represent the required time for transmission of the secure RTS, CTS, ACK, CF-End, and CF-End-ACK packets, respectively. The followings are the timeout formulization and the corresponding values in both SCP-M and SCP-O models.

#### 3.3.1. Timeout calculation in the SCP-M model

$$TO_{ACK} = T_{ACK} + P_t + S_t + SIFS = 343\ us$$

$$TO_{CTS} = T_{CTS} + P_t + S_t + SIFS = 343\ us$$

$$TO_{RTS} = T_{RTS} + P_t + S_t + SIFS = 367\ us$$

$$TO_{CF-End} = T_{CF-End} + P_t + S_t = 357\ us$$

$$TO_{CF-End-ACK} = T_{CF-End-ACK} + P_t + S_t = 357\ us$$

#### 3.3.2. Timeout calculation in the SCP-O model

$$TO_{ACK} = T_{ACK} + P_t + S_t + SIFS = 375\ us$$

$$TO_{CTS} = T_{CTS} + P_t + S_t + SIFS = 375\ us$$

$$TO_{RTS} = T_{RTS} + P_t + S_t + SIFS = 399\ us$$

$$TO_{CF-End} = T_{CF-End} + P_t + S_t = 389\ us$$

$$TO_{CF-End-ACK} = T_{CF-End-ACK} + P_t + S_t = 389\ us$$

Upon receiving a CP, the recipient must first verify its freshness using the corresponding timeout value and the following equation.

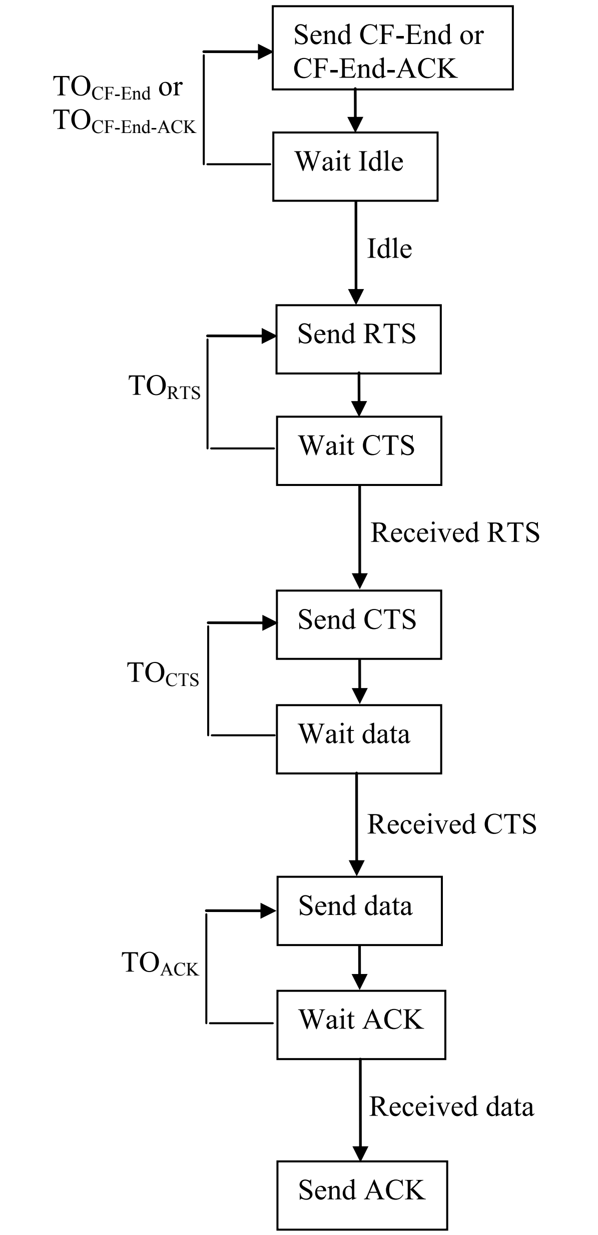


**Figure 3 Conceptual description of the five threshold timeouts and their effects.**

$$\begin{aligned} &0 \le CCT - received\ TS\ filed \le TO_{SCP} \\ &\therefore\ where\ TO_{SCP}\ is\ corresponding\ thereshold\ time\ window \end{aligned} \tag{2}$$

In Equation 2, CCT is the current clock time announced by the secure timing synchronization function in the beacon frames [12]. If the above condition is met, the received SCP is deemed fresh and is old if otherwise. Old CP will be discarded immediately by the recipient to prevent replay attacks.

**Table 2 System parameters**

| System parameter | Parameter value |
|---|---|
| Short inter-frame space, SIFS | 10 µs |
| Slot time, $S_t$ | 20 µs |
| Basic bitrate, $B_r$ | 2 Mbps |
| Physical bitrate, $PHY_r$ | 1 Mbps |
| Physical header, $PHY_h$ | 192 bits |
| Propagation delay time, $P_t$ | 1 µs |

### 3.4. Proposed M-hmac

The design of M-hmac is motivated by resource constraints in wireless networks and to optimize the network performance using the SCP-M model as compared to the SCP-O model. National Institute of Standards and Technology (NIST) considers an optional transformation function called finalization function (*g*) which can be used to derive the desired output from the original output of hash functions. This can provide more options to achieve the best desired hash results out of hash functions. Therefore, we provide the required modifications over the finalization function which can offer the following advantages.

- NIST has agreed on the presence of this function [13]. Therefore, we utilize this advantage to optimize the performance of the SCP-M model without tempering the basic structure of the HMAC-SHA1 algorithm. This can avoid unknown security flaws in the SCP-M model.
- NIST does not define any specific algorithm for this function. This feature can eliminate any limitation to provide the desired transformations in effort to enhance the efficiency of the SCP-M model.
- Since the compression function of the M-hmac is similar to HMAC-SHA1, its security analysis can be accomplished as the HMAC-SHA1 which is already well known.

The proposed M-hmac applies the same compression function as the HMAC-SHA1 algorithm with two new additional rounds to the algorithm in the finalization function. The first round of M-hmac includes a new function that involves the stream cipher encryption algorithm while the second round of M-hmac constructs the value of the AF field.

#### *3.4.1. First round of M-hmac*

In this round, there are three fields that are used as inputs to the compression function along with the FK-SCP-M as the key, namely, the TS, duration, and receiver address. The output result is 160-bit final chaining variable (FCV) that moves into the new stream cipher function (SCF). The SCF first divides the 160-bit FCV into two halves as left and right halves with the size of 80-bit each. Then, the SCF breaks both the left and the right halves separately into chunks of the same length of 4-bit each.

The SCF considers the right half chunks as keystream and the left half chunks as input message. The encryption process in the SCF is accomplished by combining each left half chunk with the corresponding right half chunk using XOR operation to generate a new 160-bit stream. The result is called transformed chaining variable (TCV) which acts as the input to the second round of M-hmac.

#### *3.4.2. Second round of M-hmac*

This round takes the 160-bit TCV and divides it into 5 words of 32-bit each, which are $\mathrm{TCV}_0$, $\mathrm{TCV}_1$, $\mathrm{TCV}_2$, $\mathrm{TCV}_3$, and $\mathrm{TCV}_4$. Next, XOR operation is applied over all the words in order to generate a 96-bit output in hex format (*H*). We call this 96-bit output as value of AF field (VAF), which is placed in the AF field of the SCP. The process of extracting the 96-bit VAF out of the 160-bit TCV is accomplished as follows.

$$\begin{aligned}
&NTCV_2 = TCV_0 \oplus TCV_4\\
&NTCV_1 = TCV_1 \oplus TCV_3\\
&NTCV_0 = TCV_0 \oplus TCV_2\\
&VAF = NTCV_0 \parallel NTCV_1 \parallel NTCV_2\\
&where \ \parallel \ is\ concatination
\end{aligned}$$

The overall process of the both rounds in M-hmac is presented in Figure 4.

### 3.5. Defense process by the SCP-O and SCP-M models

Basically, defense process by the SCP-M and SCP-O models comprises two main phases: the generation phase and the verification phase as follows.

#### *3.5.1. Generation phase*

This phase is carried out by the sender station to generate values of the TS and AF security fields. The sender station determines the creation time of the outgoing CP to be placed into the TS field. Then, the sender must generate value of the AF field. Therefore, the TS, duration, and receiver address are considered as inputs to the M-hmac/O-hmac along with the FK-SCP-M/FK-SCP-O keys. After the calculation of the sender VAF (S-VAF), this value is tagged into the AF field. Once this is done, the CP will be transmitted to the intended receiver.

#### *3.5.2. Verification phase*

This phase is carried out by the receiver station to verify the freshness and validity of the received CP. Upon receiving the CP, the receiver station will immediately discard the packets that do not have any TS or AF security elements based on the ground of wrong format. Otherwise, the receiver must first check the freshness of the received CP. Doing freshness check as the first line

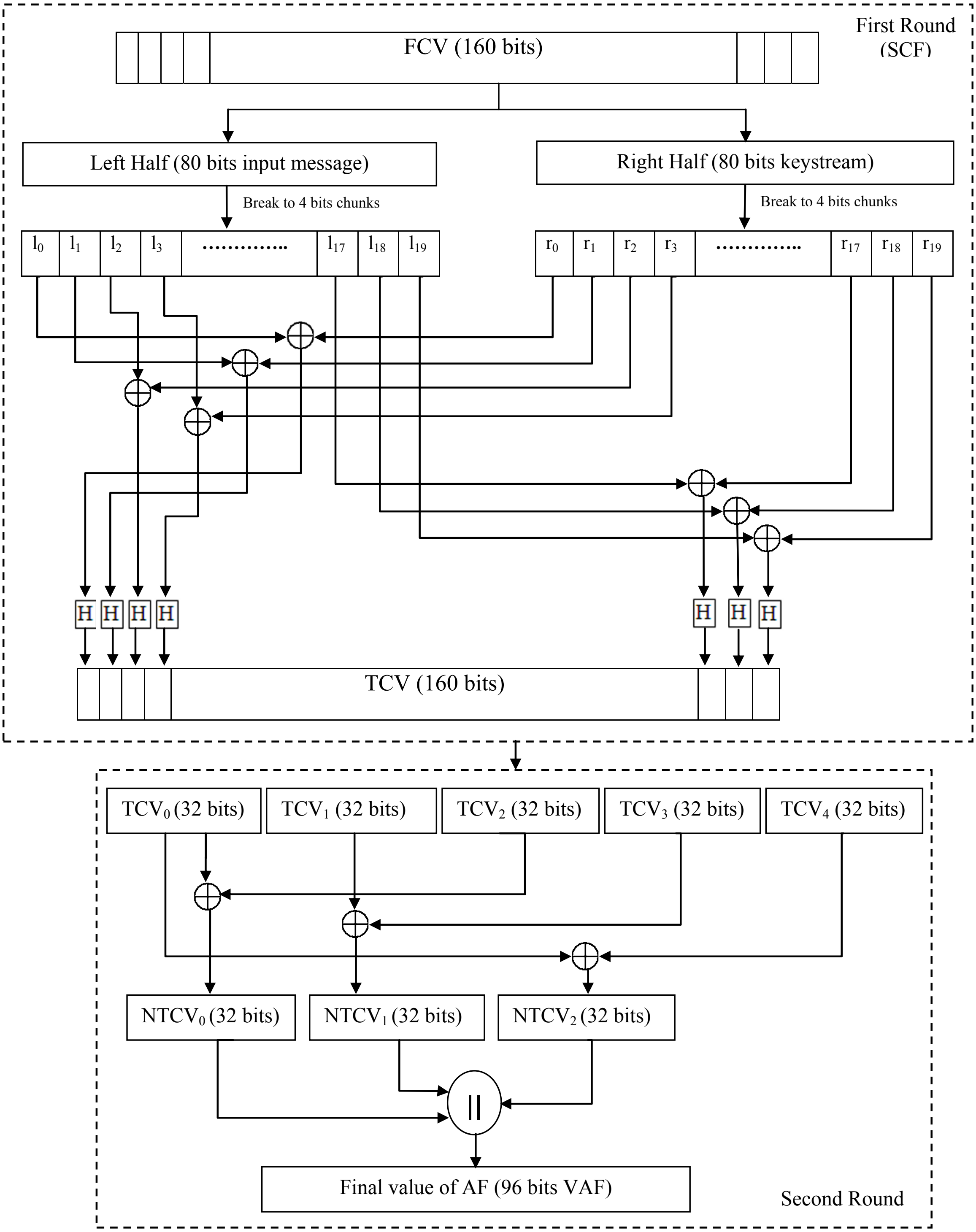


**Figure 4** Structure of the proposed M-hmac.

of defense can significantly enhance the speed and efficiency of the proposed models. By this way, the old CP are discarded immediately by the receiver without even wasting time to involve the authentication algorithm.

For the freshness check, the receiver station adheres to Equation 2 and subtracts the current clock time from the value in the TS field of the received CP. This is to check whether the result is less than or equal to the corresponding timeout value. If the condition is met, the receiver considers the CP as fresh.

At this point, if the CP is CF-End or CF-End-ACK, the receiver must verify their duration field. If the duration field of these CP is not zero, the packet is considered as an invalid packet because of its wrong format and will be discarded. Otherwise, the recipient moves to the next step to validate the authenticity of the sender and integrity of the CP.

The receiver goes through the same process as the sender to recalculate VAF, which is called Receiver VAF (R-VAF). The receiver compares this calculated R-VAF with the received S-VAF. If there is a match between these two values, the packet is accepted and its corresponding function is implemented. Otherwise the packet is discarded by the receiver. The general implementation process of the SCP-M model is presented in Figure 5.

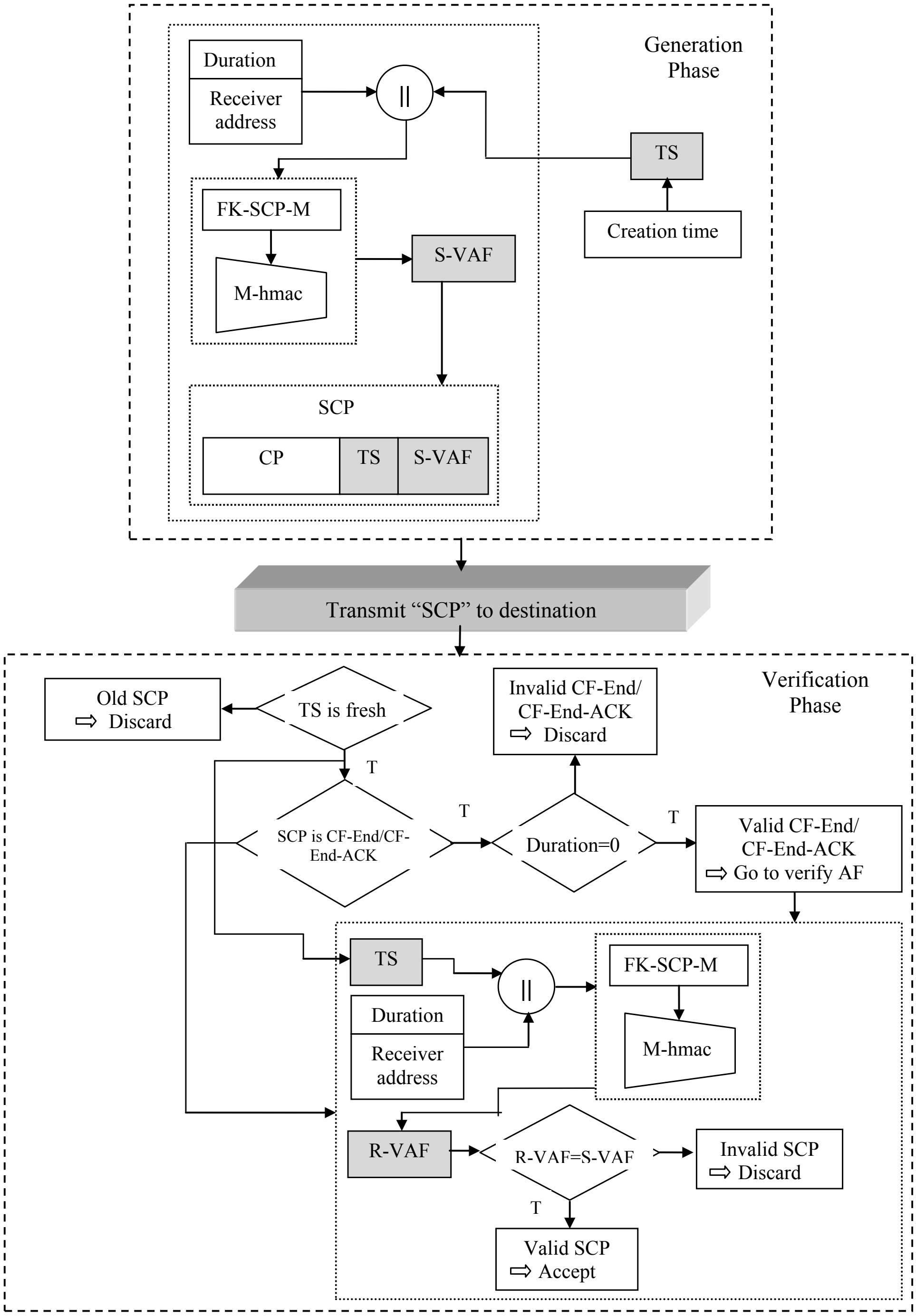


**Figure 5 Defense process by the SCP-M/SCP-O models**.

### 3.6. Security analysis

In order to start DoS attacks against the wireless networks protected by the SCP-M and SCP-O models, the attackers first have to generate valid forgery CP. Passing the authentication and integrity check requires generating valid S-VAF for the forgery CP without knowing value of the FKs. To achieve this, the attackers need to carry out some hash-based attacks against the SCP-M and SCP-O models. Therefore, in order to analyze security of the SCP-M and SCP-O models we describe the complexity of the common hash-based attacks against these models while the following notations are used.

- *Attack complexity*: The complexity of an algorithm is defined as amount of required efforts or number of operations required to break the algorithm. Therefore, unit measurement of the attack complexity is number of attempts [14]. In this regard, the security level of an algorithm is defined as the amount of work required to break the algorithm [15]. The unit measurement of the security level is bits. Thus, if $2^N$ attempt is the attack complexity, the security level of the algorithm will be $N$ bits.
- Length of the FK is denoted by $k$.
- Length of S-VAF is denoted by $m$.
- *Given information*: When an attacker attempts to perform his malicious intents, he will use all available public known information to make the attacks process faster and easier. Therefore, the attacker knows the description of the algorithms used in the system and format of the messages as well. The attacker also can observe sequence of messages with their corresponding S-VAF. However, what the attacker does not know is value of the FK which is one of the main inputs to generate S-VAF. The given information known by the attackers represented as follows.

$$\begin{cases} (x_1, z_1), (x_2, z_2), \ldots, (x_q, z_q) \\ \textit{under unknown } \mathrm{FK} - SCP - M/FK - SCP - O \\ x_i = Message\ ,\ 1 \le i \le q \\ \mathrm{y}_i = \mathrm{FCV}\ ,\ Length\ of\ \mathrm{y}_i = n \\ \mathrm{z}_i = \mathrm{g}(\mathrm{y}_i) = \mathrm{VAF}\ ,\ Length\ of\ \mathrm{z}_i = m \\ Length\ of\ \mathrm{FK} = k \end{cases}$$

The description of the common hash-based attacks against the SCP-M and SCP-O models along with their complexity is presented as follows.

#### 3.6.1. MAC guessing attack

MAC guessing attack [16] is when the attackers try to find a valid S-VAF for their forgery CP by guessing it. Since the attackers do not know the value of the key (i.e., FK-SCP-M and FK-SCP-O), they have to guess all possible values for the output result. The attacker randomly generates a tag for one fixed message of his choice and transmits it to the receiver. If by any chance, the tag is valid the receiver accepts the message and the attack is successful. However, if the tag is not verified as a valid tag by the recipient, the attacker has to repeat the process and generate another tag for his fixed message until finding a valid tag. According to [16], complexity of this attack is directly proportional to the length of tag and the applied key. The attack complexity is described as follows.

**Problem:** MAC guessing attack using the given information

**Goal:** Attacker tries to find a valid $z$ for his own $x$ such that $x \notin \{x_1, x_2,\ldots,x_q\}$

**Attack complexity:** $MIN(2^{m-1}, 2^{k-1})$ attempts

Hence, complexity of the best attack is about $2^{159}$ and $2^{95}$ for the SCP-O and SCP-M models, respectively.

#### 3.6.2. Key recovery attack

In order to have full access over the wireless channel, the attacker attempts to figure out the value of the key applied through the communications [17]. By exposing value of the key, the attacker can make any valid S-VAF. In order to implement this attack, the attacker determines a fixed input message along with a randomly selected value for the secret key. He sends these values to the same MAC algorithm applied in the target system and generates the tag. Then, the attacker appends the tag to his fixed message to transmit to the receiver in the target system. If the recipient accepts the message, it means that the attacker has the correct secret key. Otherwise, he has to try another secret key with his fixed message and this process is repeated by the attacker until finding the correct value for the key. Complexity of this attack against the SCP-M and SCP-O models is considered as follows.

**Problem:** Key recovery attack using the given information

**Goal:** Attacker tries to find a valid F$K'$ such that F$K'$ = $FK$

**Attack complexity:** $2^{k-1}$attempts

Since the length of the keys for the both SCP-M and SCP-O models is the same, to figure out the correct value of the FK-SCP-M and FK-SCP-O, the attacker has $2^{159}$ possible values to examine.

#### 3.6.3. Forgery attack

In order to implement this attack, the attacker tries to determine a valid tag for his new forgery message, which has not already been transmitted to the network. Forgery attacks in terms of second preimage [17] or birthday forgery [18] are carried out by the attackers to find valid tags. The complexity of these attacks is measured as follows.

**Problem:** Second preimage forgery attack using the given information

**Goal:** Attacker by having $(x_t, z_t)$ *where*: $1 \leq t \leq q$ tries to make his message $x$ such that $z = z_t$

**Attack complexity:** $2^{m-1}$attempts

The work factor to make a new SCP matching with a given S-VAF using the second preimage forgery method is $2^{m-1}$. Thus, the complexity of the second preimage attack is about $2^{159}$ and $2^{95}$ for the SCP-O and SCP-M models, respectively.

**Problem:** Birthday forgery attack using the given information

**Goal:** Attacker by having $(x_t, y_t)$ *where*: $1 \leq t \leq q$, tries to make his message $x$ such that $y = y_t$

**Attack complexity:** $2^{m/2}$ attempts

The workload to carry out forgery attack using the birthday method is $2^{m/2}$. Thus, complexity of the birthday forgery attack is about $2^{80}$ and $2^{48}$ for the SCP-O and SCP-M models, respectively.

## 4. Experimental design

We have developed a simulation environment using omnetpp to implement the SCP-M and SCP-O models and evaluate their performance. The simulation environment, experiments methodology, and performance metrics that are used to evaluate the models are further described in the following sections.

### 4.1. Simulation environment

The simulation environment consists of three types of entities, namely attacker station, authorized wireless stations, and authorized AP. In this environment, there are two types of wireless networks, namely, the protected and unprotected. The unprotected wireless network adheres to the current MAC layer of IEEE 802.11 standard, which is known to be vulnerable to DoS attacks. In contrast, the protected wireless network adheres to the proposed models as the secure MAC layers. Two new MAC layers are created and written in C++ to include the required codes for each proposed model separately.

In general, each wireless host (80211-Host) has a wireless NIC (80211-NIC) that includes the PHY and MAC layers. The unprotected wireless network utilizes the current MAC layer (802.11-Current-MAC layer). In contrast, the protected networks utilize two secure MAC layers as the proposed models, namely, the 802.11-SCP-M and 802.11-SCP-O MAC layers. Figure 6 shows these secure and insecure MAC layers.

The structure of the unprotected wireless network and the two protected wireless networks in the simulation

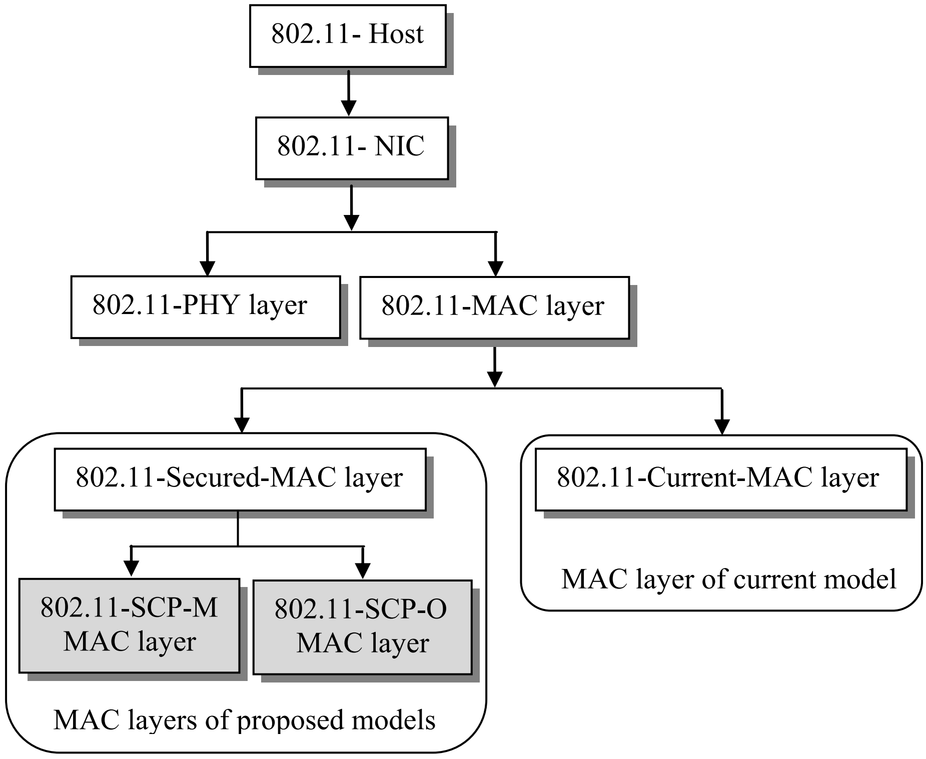


**Figure 6** Secure and insecure MAC layers in the simulation environment.

environment are developed exactly the same to provide fair conditions during evaluation of the models (Figure 7).

The two wireless stations in the unprotected network are wireless station1 (WS1) and wireless station2 (WS2) connected to the AP. All the WS1, WS2, and AP follow the 802.11 standard model.

In the protected wireless network using the SCP-M, the two wireless stations are protected wireless station1 (PWS1) and protected wireless station2 (PWS2) connected to the protected AP (PAP). All the PWS1, PWS2, and PAP follow the SCP-M model.

In the protected wireless network using the SCP-O, the two wireless stations are protected wireless station1 (PWS1) and protected wireless station2 (PWS2) connected to the PAP. All the PWS1, PWS2, and PAP follow the SCP-O model.

In the above environment, the sender transmits 1000 bytes TCP packets with 0.5 s intervals and 56 bytes ICMP packets with 1 s intervals to the AP which in turn transmits them to the receiver. The attacker station is configured to trigger different types of DoS attacks against the three networks using different types of SCP. For all types of the attacks, we consider the attack cycle as 100 false SCP per second (0.01 s attack rate), while the duration field of all these false packets is set to the maximum possible value which is 32767 μs. The reason is a larger duration filed can keep the NAV reserved for a longer time. This consequently denies the channel from the authorized users for a longer time which can extend impact of the attacks as the attackers do in real world.

### 4.2. Experiments methodology

An extensive set of experiments are carried out to implement, evaluate, and compare the performance of the protected wireless networks using the proposed SCP-M and SCP-O models and the unprotected wireless network using the current 802.11 model.

According to the IEEE 802.11, there are two types of communication modes in wireless networks, namely, the enabled and disabled RTS/CTS handshakes. Since the proposed models directly deal with the wireless CP, enabling or disabling the handshake can provide significant differences in the network performance. This motivates us to perform the experiments under two general scenarios as enabled and disabled handshakes. Each of these scenarios includes two more sub-scenarios to evaluate the models performance under DoS attacks and under normal conditions without any attacks.

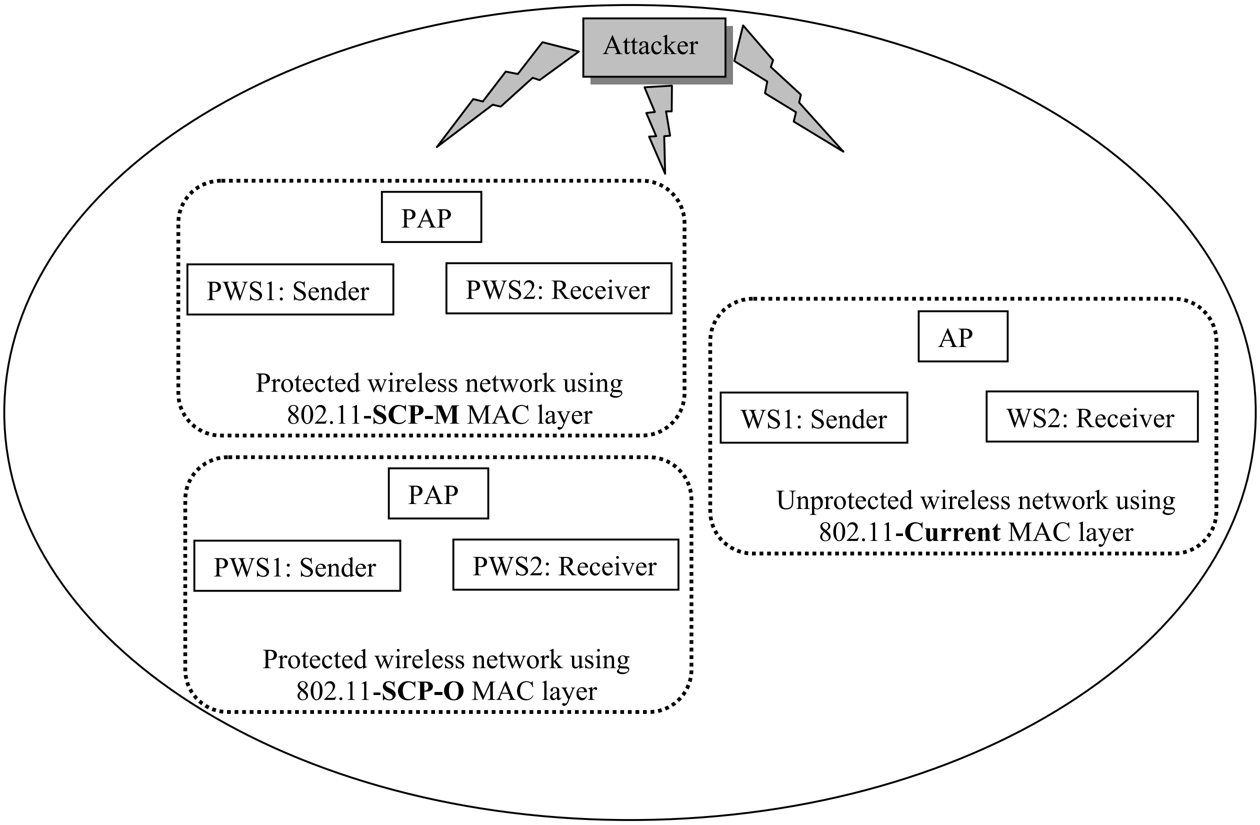


**Figure 7 Structure of the three types of wireless networks in the simulation environment.**

Throughout the experiments, the total implementation time is considered 90 s. In the scenarios under DoS attack, the 90 s is divided in three 30-s time frames to represent duration before attack (B.attack), during attack (D.attack), and after attack (A.attack). For the scenarios under normal conditions with no attacks, these three 30-s time frames are denoted as 0-30, 30-60, and 60-90 s to have identical time frames for comparisons.

### 4.3. Performance metrics

The following network metrics are investigated to evaluate performance of the models.

- **Throughput:** It is computed by dividing the amount of data received by the destination node with the time taken to arrive at this node.
- **Delay:** It is the average amount of time taken by a packet to travel from the originating node until it is successfully received at the destination node.
- **Performance improvement ratio:** It is to show the overall percentage of the throughput improvement in the wireless networks using the SCP-M model as compared to the SCP-O model under DoS attacks and under normal conditions.
- **Security cost:** It is percentage of the network performance degradation caused by the proposed models as compared to the standard model under normal conditions with no attacks.
- **Packet lost ratio (PLR):** It is measured as the number of dropped ICMP packets divided by the total number of sent ICMP packets during the attacks.
- **Round trip response time (RTT):** It is the time required for ICMP packets to travel from the source to the destination and back again.

## 5. Results and analysis

In this section, the results from implementation of the proposed models are presented. We evaluate the performance of the SCP-M and SCP-O models to determine and compare their capabilities. These comparisons are accomplished to assist in selecting the more appropriate model for the wireless networks in order to prevent DoS attacks. By comparing the results and considering the required security level along with the system limitations, capabilities, and requirements, the more efficient model can be applied. Performance of the proposed models is evaluated in the following sections.

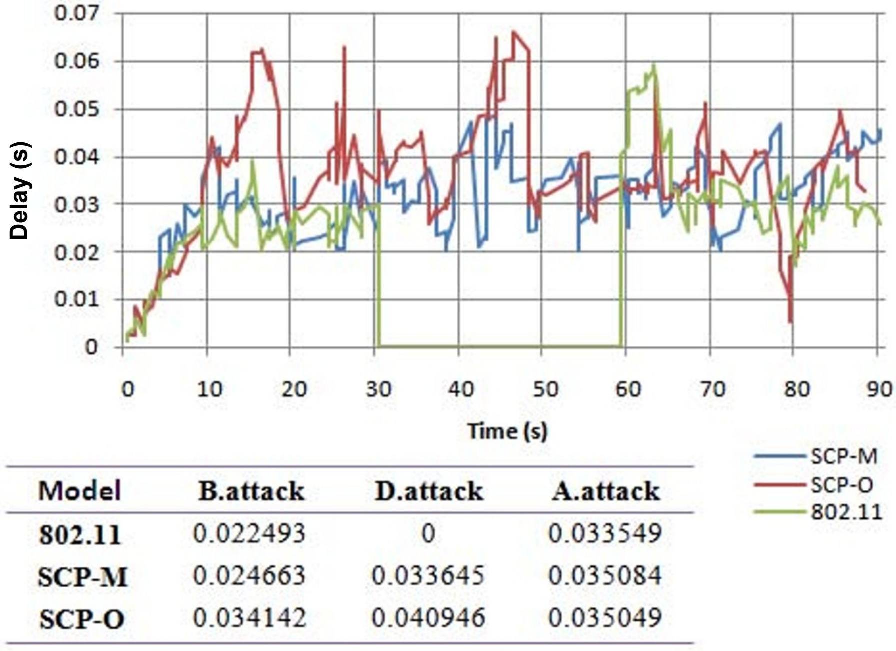
### 5.1. Effect of disabled handshake

This scenario evaluates and compares performance of the protected wireless networks using the proposed SCP-M and SCP-O models and unprotected wireless network using the current IEEE 802.11 model when the RTS/CTS handshake is not involved in the communications.

#### *5.1.1. Performance analysis under DoS attacks*

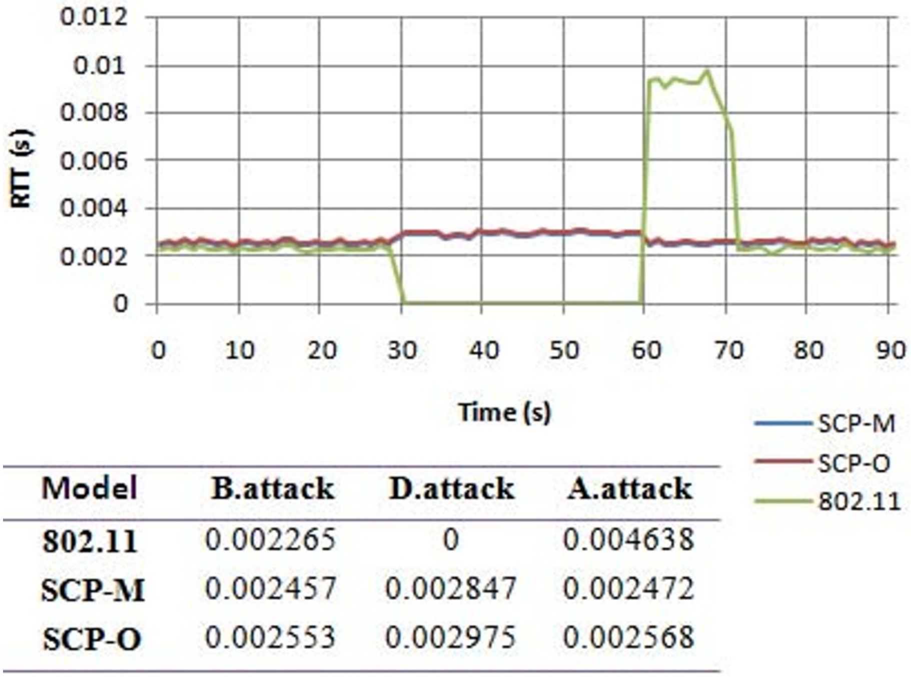
Figures 8 and 9 show the results from the delay and RTT along with the throughput and PLR as presented in Tables 3 and 4, respectively.

According to the above results, we can see that both SCP-M and SCP-O proposed models, unlike the current



| Model | B.attack | D.attack | A.attack |
|---|---|---|---|
| **802.11** | 0.022493 | 0 | 0.033549 |
| **SCP-M** | 0.024663 | 0.033645 | 0.035084 |
| **SCP-O** | 0.034142 | 0.040946 | 0.035049 |

**Figure 8** Delay comparison.



| Model | B.attack | D.attack | A.attack |
|---|---|---|---|
| 802.11 | 0.002265 | 0 | 0.004638 |
| SCP-M | 0.002457 | 0.002847 | 0.002472 |
| SCP-O | 0.002553 | 0.002975 | 0.002568 |

**Figure 9** RTT comparison.

802.11 model, have successfully prevented wireless DoS attacks. Before the attacker begins the attacks, normal traffics are observed in the both protected and unprotected networks. However, immediately after triggering the attacks, the unprotected network is flooded with large number of false CP that consume the available bandwidth. As the result, the network is no longer capable to handle valid requests made from any authorized user. Consequently, the communication between the stations is broken and the network throughput quickly drops to null. The remarkable difference between the null throughput of the current model with the normal throughput of the proposed models during the attacks, as well as the 36% PLR of the standard model as compared to the null PLR of the proposed models show that SCP-M and SCP-O models have successfully prevented the attacks.

After the attacks, we observed a high peak in amount of delay and RTT of the standard model for a short time. The cause of this high peak is related to some already scheduled packets. Before starting the attacks, some packets were in the queue waiting to be transmitted. When the attack starts, the AP is overloaded by the false CP thus dropping all the upcoming new packets. The packets that were in the queue prior to the attack still remain there because the AP is not able to do any more services. When the AP was back to its normal conditions and regained ability to provide services for its users, it began to transmit the queued packets. Therefore, these packets experienced significant amount of delay. After transmitting all these queued packets, the new upcoming packets are transmitted normally. In contrast, because the proposed models are not affected by the attacks, there is no peak in their amount of delay and RTT after the attacks.

Based on the results, we observed 100% performance improvement for the protected wireless networks by adopting the proposed models as compared to the standard model under different types of attacks. The results also show that using the SCP-M in the network will improve the network performance under the DoS attacks, up to 7% compared to the SCP-O model.

**Table 3 Throughput comparison (Mbps)**

| Model | B.attack | D.attack | A.attack |
|---|---|---|---|
| 802.11 | 1.771 | 0 | 1.673 |
| SCP-M | 1.622 | 1.609 | 1.616 |
| SCP-O | 1.576 | 1.502 | 1.604 |

**Table 4 PLR comparison**

| Model | Number of sent | Number of received | Number of lost | Lost ratio % |
|---|---|---|---|---|
| **802.11** | 90 | 56 | 34 | 36 |
| **SCP-M** | 90 | 90 | 0 | 0 |
| **SCP-O** | 90 | 90 | 0 | 0 |

#### *5.1.2. Performance analysis under no attacks*

The results of the delay, RTT, and throughput under normal conditions are presented in Figures 10, 11, and Table 5 respectively.

Based on the above results, we observed that when the wireless network is under normal condition without the presence of the false CP, performance of the proposed models is very close to the standard model. Considering the fact that security comes in price of extra overheads, the small difference between performance of the SCP-M and SCP-O models with the standard model proves that adopting these models does not impose remarkable computational overhead to the wireless networks. The security cost of 9% in SCP-M and 12% in SCP-O can be considered negligible as compared to their outstanding achievements of 100% performance improvement under different types of DoS attacks. The results also showed that adopting the SCP-M model provides 2% performance improvement as compared to the SCP-O model when the handshake is disabled during the normal transmissions.

### 5.2. Effect of enabled handshake

The scenario of enabled handshake evaluates and compares performance of the protected wireless networks using the proposed SCP-M and SCP-O models and unprotected wireless network using the current IEEE 802.11 model when the RTS/CTS handshake is used during the communications between the authorized stations.

#### *5.2.1. Performance analysis under DoS attacks*

The results of delay and throughput are presented in Figure 12 and Table 6, respectively.

The above results prove the fact that enabling the RTS/CTS handshake causes significant influence over the entire network performance regardless if the network is protected or not. In this case, the amount of delay that packets experience is higher than when the handshake is disabled. The reason for this increase is because the handshake process needs extra time to be accomplished. The originator node must wait until the handshake is completed before the actual data transmission takes place. This sending RTS and waiting for CTS before any data transmission is time consuming while at the same time data are waiting in the transmitter buffer to be sent. In contrast, with a disabled handshake, data are sent immediately as soon as they are ready, which in turn decreases the overall amount of delay.

These results are also consistent with the previous results, where the proposed models unlike the standard model have successfully prevented the DoS attacks. Comparing the null throughput of the standard model with the normal throughput of the proposed models during the DoS attacks indicate 100% improvement in the wireless network performance. In addition, adopting the SCP-M model in the wireless networks has resulted in better performance as compared to the SCP-O model in terms of higher throughput, less delay, and RTT. The SCP-M model managed to enhance the overall system performance by up to 11% during the attacks, which is 4% higher than when the handshake is disabled.

#### *5.2.2. Performance analysis under no attacks*

The results of delay and throughput are presented in Figure 13 and Table 7, respectively.

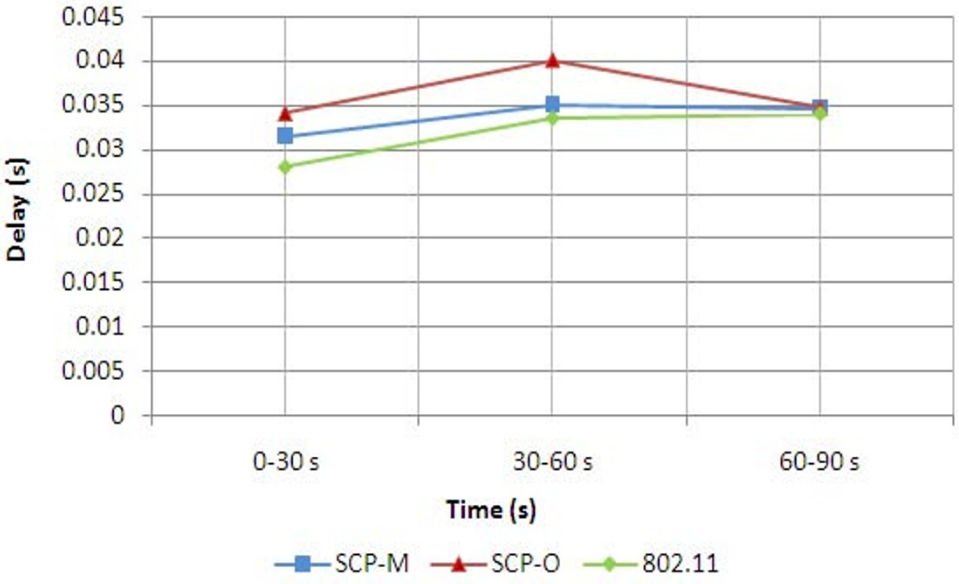
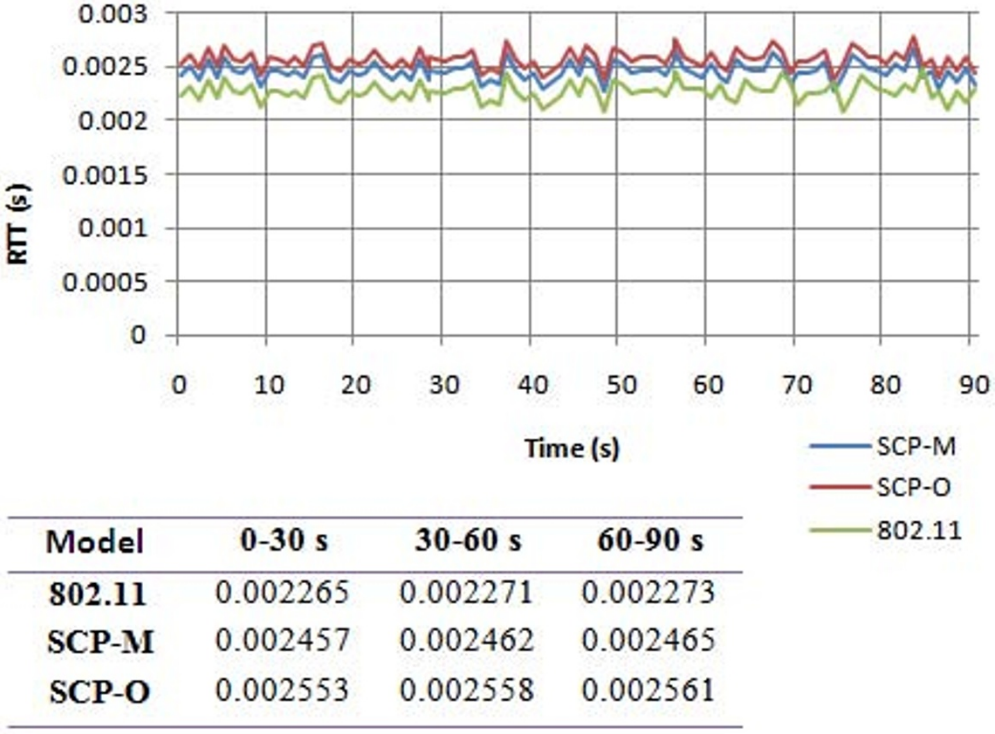


**Figure 10** Delay comparison.



| Model | 0-30 s | 30-60 s | 60-90 s |
|---|---|---|---|
| 802.11 | 0.002265 | 0.002271 | 0.002273 |
| SCP-M | 0.002457 | 0.002462 | 0.002465 |
| SCP-O | 0.002553 | 0.002558 | 0.002561 |

**Figure 11** RTT comparison.

From the above results, we observe that under normal conditions, the performance of the SCP-M model is close to the standard model. The security cost of the SCP-M and SCP-O models are about 13 and 20%, under normal conditions. Thus, these results confirm our previous results as better performance of the SCP-M model in terms of higher throughput, and less delay, and RTT compared to the SCP-O model and thereby prove advantage of the proposed M-hmac over the original O-hmac. Adopting the SCP-M model in wireless network can improve the overall system performance by up to 6% compared to the SCP-O model which is 3% higher than when the RTS/CTS handshake is disabled.

The reason of higher security improvement of the SCP-M model over the SCP-O model when the handshake is enabled compared to a disabled handshake is related to the nature of the handshake. When the handshake is used during the communications, using the SCP-O model imposes extra 8 bytes to the network by each RTS, CTS, and ACK packets. In this case, the total overhead of each successful data transmission using the SCP-O model is about 24 bytes more than the SCP-M model. In contrast, when the handshake is disabled the extra 8 bytes is imposed to the network by only ACK packet which makes the difference between the imposed overhead by the SCP-M model less than the SCP-O model.

### 5.3. Functional analysis of the proposed models

According to the SCP-M and SCP-O models, upon receiving a wireless CP, the recipient must first verify its freshness and then its validity to accept or reject the CP. Thus, the functionality of the models varies depending on value of the TS and AF. In this section, functionality of the proposed models is investigated upon receiving the CP either belong to the attacker or the authorized users. The results are presented for the SCP-M model as follows.

#### *5.3.1. Influence of arriving forgery CP*

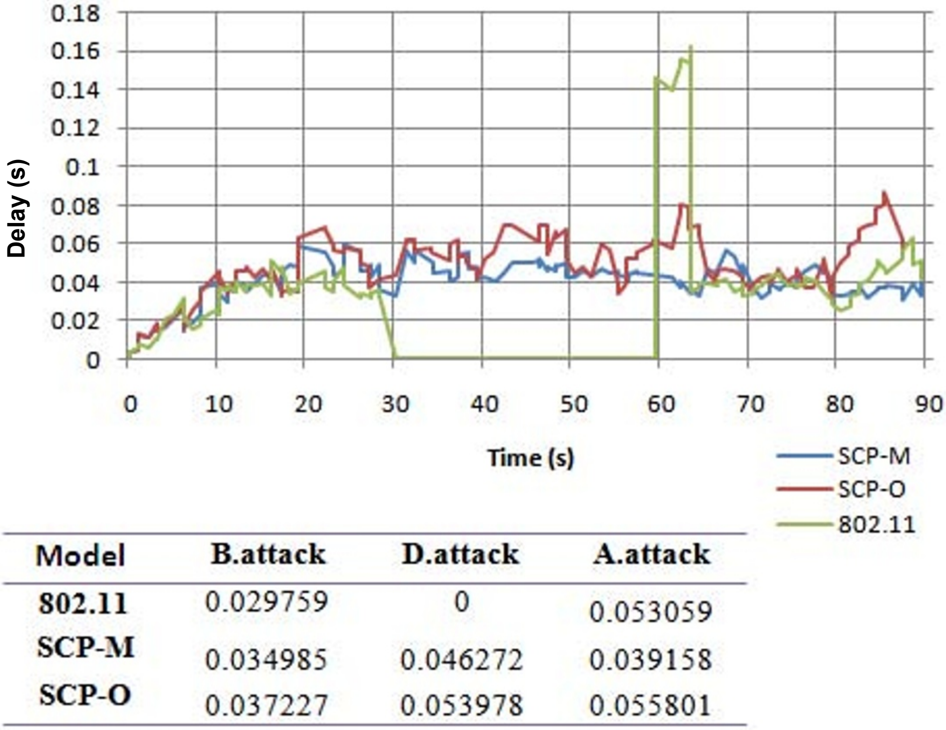
Figure 14 shows how the SCP-M model functions when the DoS attack starts at 30th second and the first

**Table 5 Throughput comparison (Mbps)**

| Model | 0-30 s | 30-60 s | 60-90 s |
|---|---|---|---|
| 802.11 | 1.771 | 1.802 | 1.793 |
| SCP-M | 1.622 | 1.637 | 1.602 |
| SCP-O | 1.576 | 1.557 | 1.573 |

**Table 6 Throughput comparison (Mbps)**

| Model | B.attack | D.attack | A.attack |
|---|---|---|---|
| 802.11 | 1.500 | 0 | 1.397 |
| SCP-M | 1.318 | 1.210 | 1.358 |
| SCP-O | 1.247 | 1.086 | 1.269 |



| Model | B.attack | D.attack | A.attack |
|---|---|---|---|
| **802.11** | 0.029759 | 0 | 0.053059 |
| **SCP-M** | 0.034985 | 0.046272 | 0.039158 |
| **SCP-O** | 0.037227 | 0.053978 | 0.055801 |

**Figure 12** Delay comparison.

forgery CP belong to the attacker is received by the recipient.

As the above figure shows, upon receiving the CP, the secure MAC layer of the SCP-M model (80211-SCP-M) first determines the packet specifications. The above packet is an RTS packet with 36 bytes length, 30 s TS field, and 32767 μs duration value which has been received by the AP. The 80211-SCP-M MAC layer

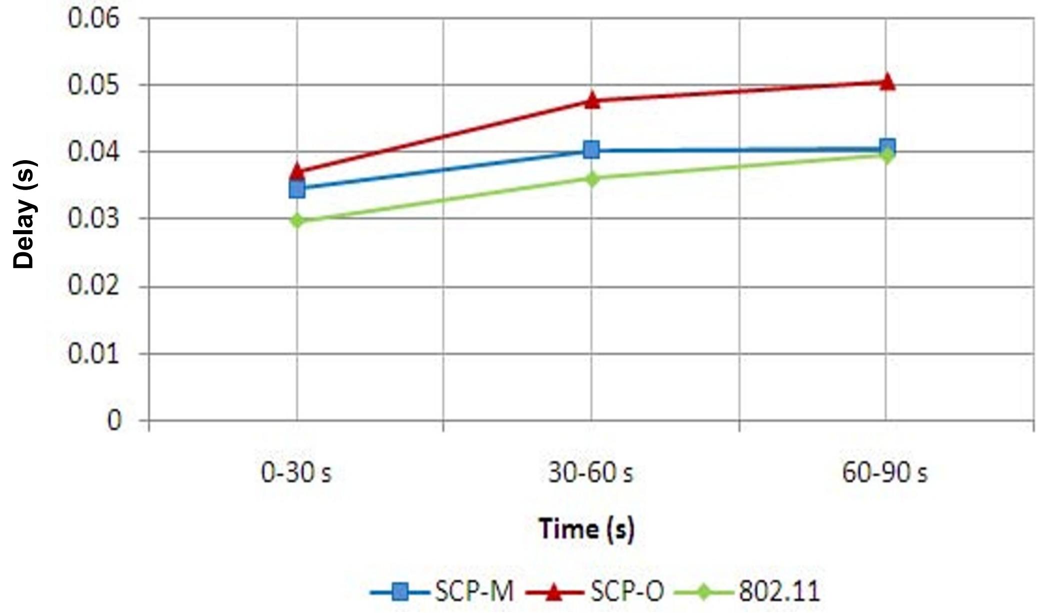


**Figure 13** Delay comparison.

**Table 7 Throughput comparison (Mbps)**

| Model | 0-30 s | 30-60 s | 60-90 s |
|---|---|---|---|
| 802.11 | 1.500 | 1.497 | 1.448 |
| SCP-M | 1.318 | 1.311 | 1.311 |
| SCP-O | 1.247 | 1.271 | 1.231 |

attempts to verify freshness of the received RTS. Therefore, as the figure shows, subtraction of the current clock time and TS is calculated by the replay-preventing mechanism. The result of the calculation is less than the $TO_{RTS}$, therefore although the RTS packet is forgery, it can pass the freshness check.

Then the recipient attempts to verify validity of the AF security field. At this step, since the attacker does not know value of the key, he is not able to set a valid value for the AF field. Thus, the AF field of the received RTS packet is not valid. Consequently, although the packet is fresh, it is discarded as an invalid CP and thwarts the attack.

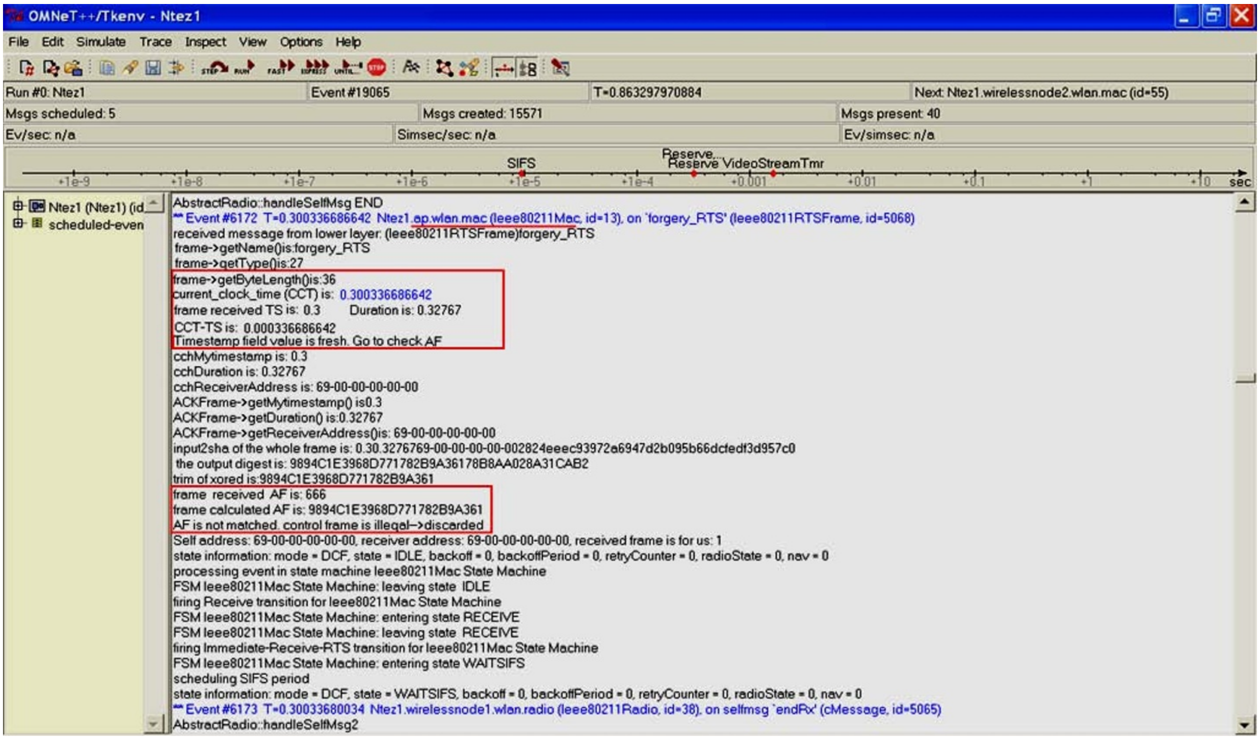
As it is observed, since the first forgery CP considered fresh, for the first forgery packet both TS and AF verifications are carried out. However after that, since the forgery packet is getting old over the time, the second forgery packet cannot pass the freshness check anymore and without even checking the AF the second and all the subsequent forgery CP are discarded. The live capturing of this state is shown in Figure 15.

As the above figure shows, the secure MAC layer of the SCP-M model (80211-SCP-M) determines the packet as an RTS packet with 36 bytes length, 30 s TS, and 32767 μs duration field received by the AP. This packet is regarded as a old CP through the freshness check and is discarded without even checking the AF field. This significantly speeds up the overall process of the proposed models and makes them more efficient to apply in the wireless networks. During the entire attacks time, the results show that only the first forgery CP at the beginning of the attack can pass the freshness check while it is discarded by the proposed models because of the wrong AF field. All the subsequent forgery CP other than the first one are discarded as old CP by the replay preventing mechanism.

### *5.3.2. Influence of arriving authorized CP*

When the destination node receives a valid CP, since the legitimate originator node has the correct value of the key, it is able to make a valid AF value in addition to a fresh value for the TS field. The process of passing both freshness and authentication verifications for legal CP is shown in Figure 16.

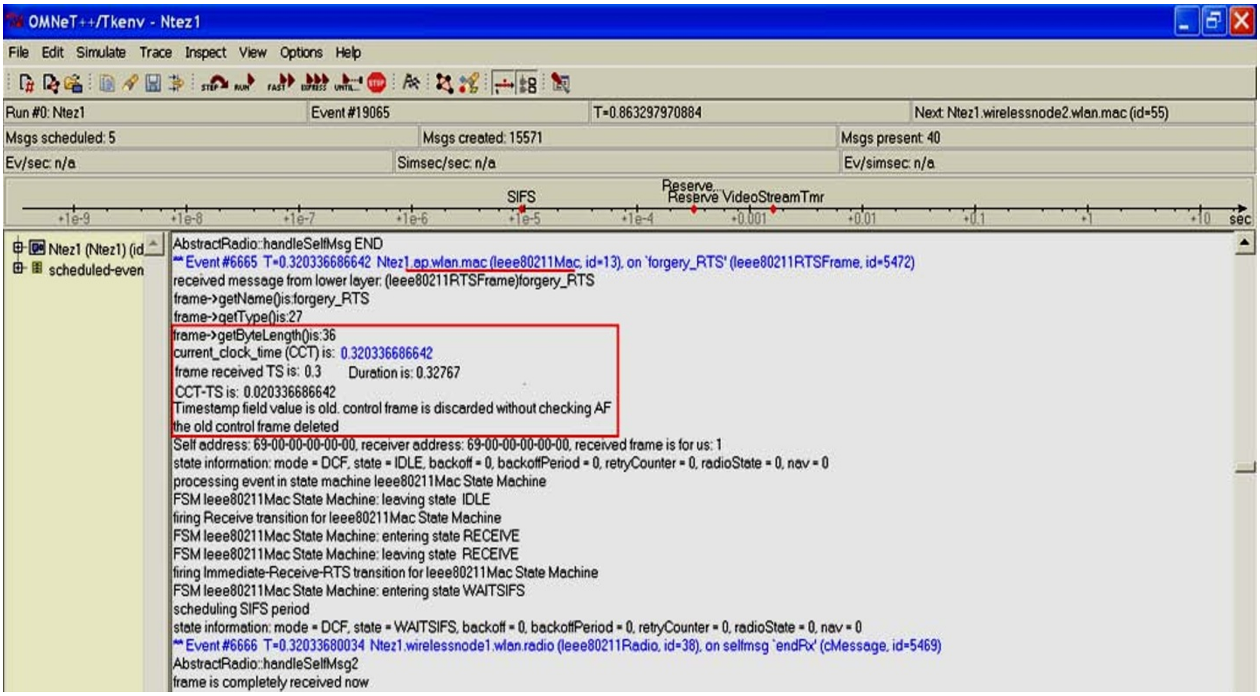
As the above figure shows, after determining the specification of the received CP by the 80211-SCP-M MAC



Run #0: Ntez1 | Event #19065 | T=0.863297970884 | Next: Ntez1.wirelessnode2.wlan.mac (id=55)
Msgs scheduled: 5 | Msgs created: 15571 | Msgs present: 40
Ev/sec: n/a | Simsec/sec: n/a | Ev/simsec: n/a

```
AbstractRadio::handleSelfMsg END
** Event #6172  T=0.300336686642  Ntez1.ap.wlan.mac (Ieee80211Mac, id=13), on 'forgery_RTS' (Ieee80211RTSFrame, id=5068)
received message from lower layer: (Ieee80211RTSFrame)forgery_RTS
 frame->getName()is:forgery_RTS
 frame->getType()is:27
frame->getByteLength()is:36
current_clock_time (CCT) is:  0.300336686642
frame received TS is:  0.3       Duration is: 0.32767
 CCT-TS is:  0.000336686642
Timestamp field value is fresh. Go to check AF
cchMytimestamp is: 0.3
cchDuration is: 0.32767
cchReceiverAddress is: 69-00-00-00-00-00
ACKFrame->getMytimestamp() is0.3
ACKFrame->getDuration() is:0.32767
ACKFrame->getReceiverAddress()is: 69-00-00-00-00-00
input2sha of the whole frame is: 0.30.3276769-00-00-00-00-002824eeec93972a6947d2b095b66dcfedf3d957c0
 the output digest is: 9894C1E3968D771782B9A36178B8AA028A31CAB2
trim of xored is:9894C1E3968D771782B9A361
frame  received  AF is: 666
frame calculated AF is: 9894C1E3968D771782B9A361
AF is not matched. control frame is illegal-->discarded
Self address: 69-00-00-00-00-00, receiver address: 69-00-00-00-00-00, received frame is for us: 1
state information: mode = DCF, state = IDLE, backoff = 0, backoffPeriod = 0, retryCounter = 0, radioState = 0, nav = 0
processing event in state machine Ieee80211Mac State Machine
FSM Ieee80211Mac State Machine: leaving state  IDLE
firing Receive transition for Ieee80211Mac State Machine
FSM Ieee80211Mac State Machine: entering state RECEIVE
FSM Ieee80211Mac State Machine: leaving state  RECEIVE
firing Immediate-Receive-RTS transition for Ieee80211Mac State Machine
FSM Ieee80211Mac State Machine: entering state WAITSIFS
scheduling SIFS period
state information: mode = DCF, state = WAITSIFS, backoff = 0, backoffPeriod = 0, retryCounter = 0, radioState = 0, nav = 0
** Event #6173  T=0.30033680034  Ntez1.wirelessnode1.wlan.radio (Ieee80211Radio, id=38), on selfmsg 'endRx' (cMessage, id=5065)
AbstractRadio::handleSelfMsg2
```

**Figure 14** Discarding forgery CP with fresh TS and invalid AF by the SCP-M model.


OMNeT++/Tkenv - Ntez1
File Edit Simulate Trace Inspect View Options Help
STEP RUN FAST EXPRESS UNTIL STOP
SIFS
Reserve...
Reserve VideoStreamTmr
+1e-9 +1e-8 +1e-7 +1e-6 +1e-5 +1e-4 +0.001 +0.01 +0.1 +1 +10 sec
Ntez1 (Ntez1) (id
scheduled-even

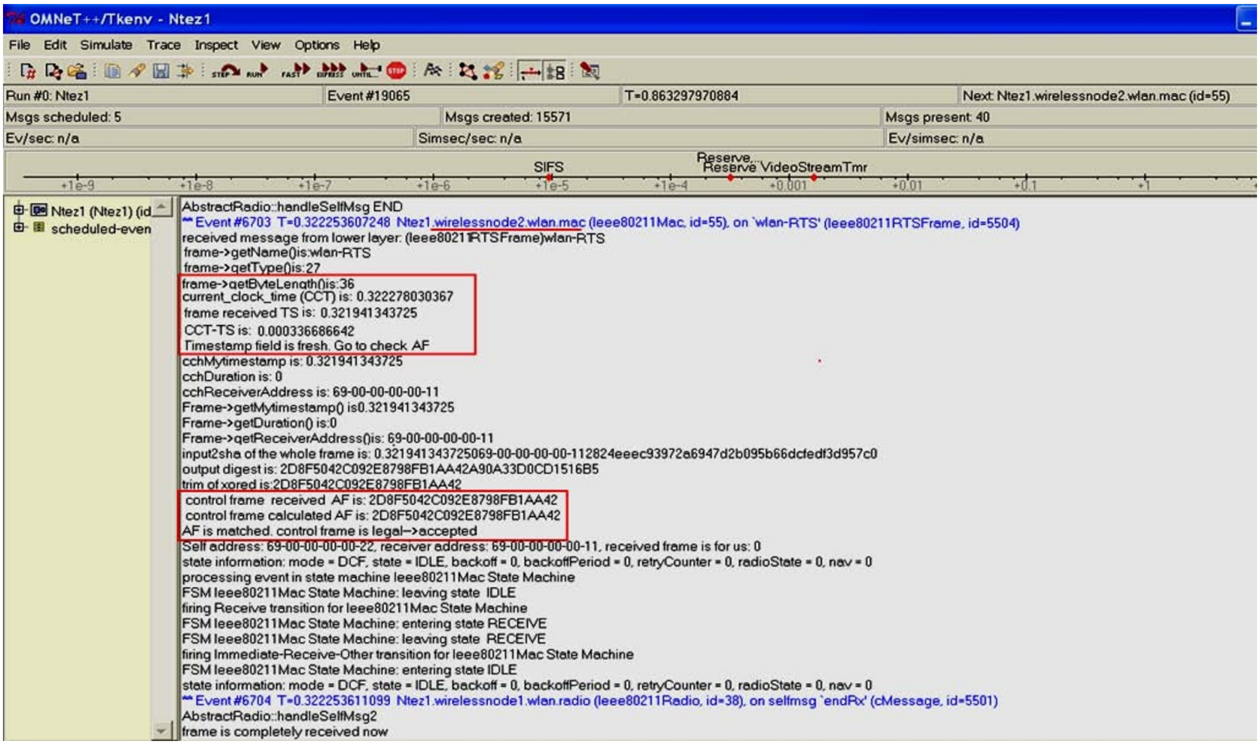
```
Run #0: Ntez1    Event #19065    T=0.863297970884    Next: Ntez1.wirelessnode2.wlan.mac (id=55)
Msgs scheduled: 5    Msgs created: 15571    Msgs present: 40
Ev/sec: n/a    Simsec/sec: n/a    Ev/simsec: n/a
```

```
AbstractRadio::handleSelfMsg END
** Event #6665  T=0.320336686642  Ntez1.ap.wlan.mac (Ieee80211Mac, id=13), on `forgery_RTS' (Ieee80211RTSFrame, id=5472)
received message from lower layer: (Ieee80211RTSFrame)forgery_RTS
frame->getName()is:forgery_RTS
frame->getType()is:27
frame->getByteLength()is:36
current_clock_time (CCT) is:  0.320336686642
frame received TS is:  0.3      Duration is: 0.32767
CCT-TS is:  0.020336686642
Timestamp field value is old. control frame is discarded without checking AF
the old control frame deleted
Self address: 69-00-00-00-00-00, receiver address: 69-00-00-00-00-00, received frame is for us: 1
state information: mode = DCF, state = IDLE, backoff = 0, backoffPeriod = 0, retryCounter = 0, radioState = 0, nav = 0
processing event in state machine Ieee80211Mac State Machine
FSM Ieee80211Mac State Machine: leaving state  IDLE
firing Receive transition for Ieee80211Mac State Machine
FSM Ieee80211Mac State Machine: entering state RECEIVE
FSM Ieee80211Mac State Machine: leaving state  RECEIVE
firing Immediate-Receive-RTS transition for Ieee80211Mac State Machine
FSM Ieee80211Mac State Machine: entering state WAITSIFS
scheduling SIFS period
state information: mode = DCF, state = WAITSIFS, backoff = 0, backoffPeriod = 0, retryCounter = 0, radioState = 0, nav = 0
** Event #6666  T=0.32033680034  Ntez1.wirelessnode1.wlan.radio (Ieee80211Radio, id=38), on selfmsg `endRx' (cMessage, id=5469)
AbstractRadio::handleSelfMsg2
frame is completely received now
```

**Figure 15** Discarding forgery CP with old TS by the SCP-M model.

layer, first the TS field of the packet is verified. Since value of the TS field is recognized fresh, the AF field of the packet is verified. Because the packet is originated from authorized user that knows correct value of the key, the AF field is valid as well and the packet is accepted by the recipient as a valid CP.

During implementation of the models it was observed that, like the standard model, the SCP-M and SCP-O


OMNeT++/Tkenv - Ntez1
File Edit Simulate Trace Inspect View Options Help
STEP RUN FAST EXPRESS UNTIL STOP
SIFS
Reserve...
Reserve VideoStreamTmr
+1e-9 +1e-8 +1e-7 +1e-6 +1e-5 +1e-4 +0.001 +0.01 +0.1 +1
Ntez1 (Ntez1) (id
scheduled-even

```
Run #0: Ntez1    Event #19065    T=0.863297970884    Next: Ntez1.wirelessnode2.wlan.mac (id=55)
Msgs scheduled: 5    Msgs created: 15571    Msgs present: 40
Ev/sec: n/a    Simsec/sec: n/a    Ev/simsec: n/a
```

```
AbstractRadio::handleSelfMsg END
** Event #6703  T=0.322253607248  Ntez1.wirelessnode2.wlan.mac (Ieee80211Mac, id=55), on `wlan-RTS' (Ieee80211RTSFrame, id=5504)
received message from lower layer: (Ieee80211RTSFrame)wlan-RTS
 frame->getName()is:wlan-RTS
 frame->getType()is:27
frame->getByteLength()is:36
current_clock_time (CCT) is: 0.322278030367
 frame received TS is:  0.321941343725
 CCT-TS is:  0.000336686642
 Timestamp field is fresh. Go to check AF
cchMytimestamp is: 0.321941343725
cchDuration is: 0
cchReceiverAddress is: 69-00-00-00-00-11
Frame->getMytimestamp() is0.321941343725
Frame->getDuration() is:0
Frame->getReceiverAddress()is: 69-00-00-00-00-11
input2sha of the whole frame is: 0.321941343725069-00-00-00-00-112824eeec93972a6947d2b095b66dcfedf3d957c0
output digest is: 2D8F5042C092E8798FB1AA42A90A33D0CD1516B5
trim of xored is:2D8F5042C092E8798FB1AA42
 control frame  received  AF is: 2D8F5042C092E8798FB1AA42
 control frame calculated AF is: 2D8F5042C092E8798FB1AA42
AF is matched. control frame is legal-->accepted
Self address: 69-00-00-00-00-22, receiver address: 69-00-00-00-00-11, received frame is for us: 0
state information: mode = DCF, state = IDLE, backoff = 0, backoffPeriod = 0, retryCounter = 0, radioState = 0, nav = 0
processing event in state machine Ieee80211Mac State Machine
FSM Ieee80211Mac State Machine: leaving state  IDLE
firing Receive transition for Ieee80211Mac State Machine
FSM Ieee80211Mac State Machine: entering state RECEIVE
FSM Ieee80211Mac State Machine: leaving state  RECEIVE
firing Immediate-Receive-Other transition for Ieee80211Mac State Machine
FSM Ieee80211Mac State Machine: entering state IDLE
state information: mode = DCF, state = IDLE, backoff = 0, backoffPeriod = 0, retryCounter = 0, radioState = 0, nav = 0
** Event #6704  T=0.322253611099  Ntez1.wirelessnode1.wlan.radio (Ieee80211Radio, id=38), on selfmsg `endRx' (cMessage, id=5501)
AbstractRadio::handleSelfMsg2
frame is completely received now
```

**Figure 16** Accepting valid CP with fresh TS and valid AF by the SCP-M model.

models do not discard any valid CP which proves that under normal conditions the models correctly follow the IEEE 802.11 standard. Based on the obtained results, under normal conditions without presence of the attackers, both freshness and authenticity of the authorized CP must be verified which can consume system resources. To avoid this issue and to get the best performance of the wireless networks, enabling the SCP-M and SCP-O models after detection of the DoS attacks can reduce the overall overheads and maintain maximum network throughput. In this case, since the proposed models are only used when the DoS attacks have occurred, they are light on resource usages which make the models significantly efficient for the limited resources wireless networks.

## 6. Conclusion

The unprotected CP are capable to make the entire wireless networks vulnerable to DoS attacks and result in serious damages in the critical areas where constant availability of the networks, resources, and services is a prime priority. In this study, we proposed two new distinct models, SCP-M and SCP-O, to prevent wireless DoS and replay attacks by protecting the CP. The proposed SCP-M and SCP-O models were evaluated through extensive set of scenarios and experiments. The obtained results proved that while the standard model failed against the attacks, both the proposed models have successfully prevented the wireless DoS and replay attacks. The results also showed that the best performance of the both models is obtained when the RTS/CTS handshake is disabled. In this case, the performance of the SCP-M and SCP-O models is considerably close to the standard 802.11 model with a negligible security cost. Results of comparison between SCP-M and SCP-O models also proved that by adopting the SCP-M in wireless networks, the network performance is enhanced. This proves that the proposed M-hmac as underlying authentication algorithm of the SCP-M model provides higher efficiency to access to the system resources and services while maintaining a sufficient level of security.

### Competing interests

The authors declare that they have no competing interests.

Received: 8 October 2010 Accepted: 28 November 2011
Published: 28 November 2011

### References

1. A Rachedi, A Benslimane, Impacts and solutions of control packets vulnerabilities with IEEE802.11 MAC. Wiley InterSci Wirel Commun Mob Comput. **9**(4):469–488 (2008)
2. MA Khan, A Hasan, Pseudo random number based authentication to counter denial of service attacks on 802.11. *IEEE 5th International Conference on Wireless and Optical Communications Networks, WOCN '08.* (Surabaya, Indonesia, 2008), pp. 1–5
3. K Bicakci, B Tavli, Denial-of-Service attacks and countermeasures in IEEE 802.11 wireless networks. ACM J Comput Stand Interface. **31**(5):931–941 (2009). doi:10.1016/j.csi.2008.09.038
4. TS Celebi, Design and FPGA implementation of hash processor. (Thesis, Graduate School of Natural and Applied Sciences of Middle East Technical University, 2007 in press)
5. FF Yao, YL Yin, Design and analysis of password-based key derivation functions. IEEE Trans Inf Theory. **51**(9):3292–3297 (2005). doi:10.1109/TIT.2005.853307
6. P Oechslin, Making a faster cryptanalytic time-memory trade-off. Springer Lecture Notes in Computer Science (LNCS). 617–630 (2003)
7. L Buttyan, L Csik, Security analysis of reliable transport layer protocols for wireless sensor networks. *IEEE Workshop on Sensor Networks and Systems for Pervasive Computing (PerSeNS).* (Mannheim, Germany, 2010)
8. H Krawczyk, HMAC-based Extract-and-expand key derivation function (HKDF). Request for comments, RFC 5869. (2009)
9. W He, X Liu, H Nguyen, K Nahrstedt, A cluster-based protocol to enforce integrity and preserve privacy in data aggregation. *IEEE 29th International Conference on Distributed Computing Systems (ICDCS'09).* (Montreal, Canada, 2009), pp. 14–19
10. S Blake-Wilson, M Nystrom, D Hopwood, J Mikkelsen, T Wright, Transport Layer Security (TLS) Extensions. Request for comments RFC3546. (2003)
11. L Green, K Balmy, M Emmelmann, IEEE doc 802.11-06/0928r2: Theoretical Throughput Limits. IEEE 802.11 publications. (2006)
12. M Malekzadeh, AA Abdul Ghani, S Subramaniam, Design and implementation of a lightweight security model to prevent IEEE 802.11Wireless DoS attacks. EURASIP J Wirel Commun Netw. **2011**, 1–16 (2011)
13. E Heilman, Attacks against permute-transform-Xor compression functions and spectral hash. http://eprint.iacr.org/2009/415.pdf (2009)
14. D Man, Y Wu, Y Yang, A method based on global attack graph for network hardening. *IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'08).* (Dalian, China, 2008), pp. 1–4
15. Q Dang, Recommendation for applications using approved hash algorithms. (NIST Special Publication SP, 2009), pp. 800–107
16. B Preneel, *MAC Algorithms: State of the Art and Recent Developments.* (ANCE Tunis, School on Cryptography, Tunis, Tunisia, 2008)
17. K Jia, X Wang, Z Yuan, G Xu, Distinguishing attack and second-preimage attack on the CBC-like MACs. Springer Lecture Notes in Computer Science (LNCS). **5888**, 349–361 (2009). doi:10.1007/978-3-642-10433-6_23
18. H Handschuh, B Preneel, Key-recovery attacks on universal hash function based MAC algorithms. Springer Lecture Notes in Computer Science (LNCS). **5157**, 144–161 (2008). doi:10.1007/978-3-540-85174-5_9
19. ZI Qureshi, B Aslam, A Mohsin, Y Javed, Using randomized association ID to detect and prevent spoofed PS-Poll based denial of service attacks in IEEE 802.11 WLANs. ACM WSEAS Trans Commun. **7**(3):170–179 (2008)
20. W Chen, D Chen, G Sun, Y Zhang, Defending against jamming attacks in wireless local area networks. Springer Lecture Notes in Computer Science (LNCS). **4610**, 519–528 (2007). doi:10.1007/978-3-540-73547-2_53
21. K Sugantha, S Shanmugavel, Anomaly detection of the NAV attack in MAC layer under non-time and time-constrained environment. *IEEE International Conference on Wireless and Optical Communications Networks (IFIP'06).* (Bangalore, India, 2006), pp. 1–5
22. K Sugantha, S Shanmugavel, A Statistical approach to detect NAV attack at MAC layer. *Proceedings of the International Workshop on Wireless Ad-Hoc Networks.* (London, UK, 2005)
23. Z Zhang, J Wu, J Deng, M Qiu, Jamming ACK attack to wireless networks and a mitigation approach. *Proceeding of IEEE Global Telecommunications Conference, Wireless Networking Symposium (GLOBECOM'08).* (New Orleans, USA, 2008), pp. 1–5
24. R Negi, A Rajeswaran, Dos analysis of reservation based MAC protocols. IEEE International Conference on Communications (ICC'05). **5**, 3632–3636 (2005)
25. D Chen, J Ding, PK Varshney, Protecting wireless networks against a denial of service attack based on virtual jamming. *ACM 9th International Conference on Mobile Computing and Networking (MobiCom'03).* (San Diego, USA, 2003)

26. J Bellardo, S Savage, 802.11 denial-of-service attacks: real vulnerabilities and practical solutions. *Proceedings of 12 USENIX Security Symposium*. (Berkeley, USA, 2003)

doi:10.1186/1687-417X-2011-4
**Cite this article as:** Malekzadeh *et al.*: Protected control packets to prevent denial of services attacks in IEEE 802.11 wireless networks. *EURASIP Journal on Information Security* 2011 **2011**:4.