# Analysing and Attacking the 4-Way Handshake of IEEE 802.11i Standard

Abdullah Alabdulatif *†, Xiaoqi Ma†, Lars Nolle †
*Department of Computer, College of Sciences and Arts, Qassim University, P.O. Box 53, Al-Rass, Saudi Arabia
{A.Alabdulatif}@qu.edu.sa
†School of Science and Technology, Nottingham Trent University, Nottingham, UK
{N0284284, xiaoqi.ma, lars.nolle}@ntu.ac.uk

***Abstract*—The IEEE 802.11i standard has been designed to enhance security in wireless networks. In the 4-way handshake the supplicant and the authenticator use the pairwise master key (PMK) to derive a fresh pairwise transient key (PTK). The PMK is not used directly for security while assuming the supplicant and authenticator have the same PMK before running 4-way handshake. In this paper, the 4-way handshake phase has been analysed using Isabelle tool to identify a new Denial-of-Service (DoS) attack. The attack prevents the authenticator from receiving message 4 after the supplicant sends it out. This attack forces the authenticator to re-send the message 3 until time out and subsequently to de-authenticate supplicant. This paper has proposed improvements to the 4-way handshake to avoid the Denial-of-Service attack.**

## I. Introduction

One of the great challenges for wireless environments is to provide enough strong protection to the data packages exchanged over WLANs. Eavesdropping attacks can be conducted in WLANs by potential attackers with little effort and suitable radio receivers. So attackers can attack a WLAN with difficult detection or prevention [1]. The wired equivalent privacy protocol (WEP) has been the first attempt proposed to protect the data packages exchanged over WLANs. However, WEP does not provide strong protection to the data packages exchanged over WLANs, especially in the encryption. In June 2004, the IEEE task group $i$ developed a new standard called 802.11i to avoid the weaknesses in WEP and to enhance confidentiality, integrity and mutual authentication [2, 3].

The 802.11i standard involves three entities called supplicant (wireless device), authenticator (access point) and authentication server. All six phases of the 802.11i standard are important to achieve authentication, especially for the 4-way handshake. The 4-way handshake aims to establish a fresh session key between the access point and the wireless device. There are three tasks for the access point and the wireless device to achieve successfully in the 4-way handshake phase. Firstly, establish random nonces to verify the liveness of each other. Then, confirm the existence of the PMK at the access point and the wireless device. Finally, generate the group transient key (GTK) by the access point and transfer the GTK to the wireless device [4].

The 4-way handshake can be analysed using linear temporal logic. Alabdulatif *et al.* have proposed a framework which can be used to investigate and analyse the 4-way handshake [5, 6]. This framework can be classified as a theorem proving method, which is used to analyse all possible behaviours of a protocol to ensure they meet a set of correctness conditions [7]. There are a number of general rules and assumptions in the framework that can be used to analyse many protocols. Isabelle is one of the tools that can be used to implement the framework and to analyse protocols. In this paper, we identify a DoS attack in the 4-way handshake, which prevents message 4 from being received by the access point. Then the access point will re-send message 3 to obtain message 4, but to no avail. All resent messages will be discarded by the wireless device.

The rest of this paper is structured as follows: section II introduces the IEEE802.11i standrad and the 4-way handshake phase. In section III, the framework is adjusted for analysing the 4-way handshake using Isabelle and proving basic properties. Section IV explains in detail the DoS attack and how to fix it in the 4-way handshake on massage 4. The conclusion and future work are in section V.

## II. IEEE802.11i Standard

The 802.11i standard has six sequential phases to achieve authentication among the authentication server, the authenticator (access point) and the supplicant (wireless device). In each phase there are some tasks that should be achieved successfully to meet the security target of the phase. The success of authentication means the wireless device and the access point are identified and verified by each other and a secret key is established for exchanging encrypted data over WLANs. The authentication procedure consists of six phases as follows: a) discover phase, b) authentication and association phase, c) EAP/802.1x/RADIUS authentication, d) 4-way handshake, e) group key handshake, f) secure data communication [8].

### *A. The 4-Way Handshake Phase*

The 4-way handshake is essential in the IEEE 802.11i protocol, aiming to verify that the access point is legitimate to generate the PMK. The wireless device may request to run the 4-way handshake protocol or the access point may start by itself. Figure 1 shows that the 4-way handshake exchanges messages at abstract level, where AA and SPN represent the MAC address of the access point and wireless device, respectively. SNonce represents the access point nonce

978-1-908320-20/9/$25.00©2013 IEEE

and ANonce represents the nonce of the wireless device. The msg1, 2, 3, 4 refer to several message types; $sn$ is sequence number. MIC$_{PTK}\{\}$ refers to the Message Integrity Code (MIC) that uses the fresh PTK to calculate the integrity code of contents between the braces. MIC is used instead of Message Authentication Code (MAC) for cryptography because the meaning of MAC in network is medium access control [9].

The access point and wireless device assume that both have the same PMK before running the 4-way handshake. The PMK is not used for any security operation directly; instead it is used to infer the PTK through pseudo random function with result length $X$ as follow:

PTK = PRF-X(PMK, Pairwise key expansion || Min $\{AA, SPA\}$ || Max $\{AA, SPA\}$ || Min $\{ANonce, SNone\}$ || Max $\{ANonce, SNone\})$

The fresh PTK is divided into three keys. The first key is the Key Confirmation Key (KCK), which is only used to calculate MIC. The second key is the Key Encryption Key (KEK) and the third key is the Temporary Key (TK). The KEK and TK are not used in the authentication process, so they will be ignored in this paper [10].

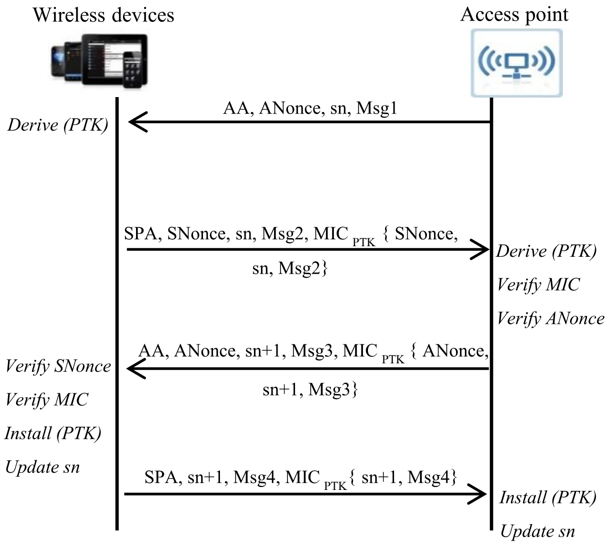

Figure 1. The 4-way Handshake

The wireless device and the access point can discard a message in the 4-way handshake when receiving a message with unexpected sequence number or invalid MIC. Message 1 is unacceptable for the wireless device when it is received after the time interval of successful 802.11i authentication. In this case, the wireless device tries to authenticate with same access point or another one after disassociating and de-authenticating the current access point. On the other side, if the access point has not received a message before time out then it will re-send within configured time intervals. Moreover, the access point will de-authenticate the wireless device if it has never received any reply from the wireless device [9].

## III. A Framework Implemented in Isabelle

Isabelle tool is widely used to reason a formal system based on higher order logic. Bella defined Isabelle tool as “a generic, interactive theorem prover” [11]. Isabelle provides a high level automation, which means human intervention required is lower than many other tools. Paulson is one of the researchers who have used Isabelle to prove the correctness of a number of protocols, such as the internet protocol TLS [12]. In this paper, Isabelle is used to analyse the 4-way handshake prase of 802.11i standard.

### A. Framework for Analysing Protocols Using Linear Temporal Logic

Isabelle can be used to verify and prove the correctness of security protocols. Four steps are followed to analyse protocols using Isabelle tool. First, adjust the framework slightly for the protocol to be verified. The reason is that the framework is a template and requires to accommodate the minor differences amongst various security protocols. Then, model the protocol steps by rewriting the protocol to make it compatible with the language used in the framework. After that, prove basic and essential properties of the protocol, which can be reused for other protocols. Finally, prove security properties of the protocol based on the proof of the basic properties mentioned above. In the next section, we will follow these steps and analyse the 4-way handshake stage using the framework proposed by Alabdulatif *et al* [5, 6].

### B. Framework Adjustment

The framework requires a slight amendment to be appropriate for analysing the 4-way handshake protocol. The access point $AP$ and the wireless device $SP$ are honest agents and the attacker here is called $Spy$. Also, the trusted third party is $TTP$. The definition of $agent$ will be modified as:

datatype $agent = SP \mid AP \mid Spy \mid TTP$

Similarly, four new nonces are used in the 4-way handshake, with SN and SN1 representing the sequence number and the sequence number +1, respectively. The SNonce and ANonce are fresh nonces chosen by agents SP and AP, respectively. Therefore, the defination of $nonce$ will be modified as:

datatype $nonce = SN \mid SN1 \mid SNonce \mid ANonce$

In addition, a new type will be defined for Msg1, Msg2, Msg3 and Msg4. This type will be called $Messages$ and added as follows:

datatype $Masseges = Msg1 \mid Msg2 \mid Msg3 \mid Msg4$

Since the 4-way handshake uses the Message Integrity Code and typed messages, we need to add two constructors for datatype $msg$. They can be defined in $msg$ datatype:

$$\begin{aligned} \text{datatype } msg =& Mag\ Masseges \\ & |MIC\ msg\ key \end{aligned}$$

Besides the type definitions, the analysis requires several new actions to represent their behaviours during the authentication process. The new three actions are added as follows:

$Discard :: agent \rightarrow msg \rightarrow Formula$

$Block :: agent \rightarrow msg \rightarrow Formula$

$RRcv :: agent \rightarrow msg \rightarrow Formula$

The $Discard$ action represents the behaviour of an agent when ignoring received message. The $Block$ action represents the behaviour of an agent when removing the message from the network so that the recipient cannot receive the message. The behaviour of an agent receiving the same message more than once can be represented by the $RRcv$ action.

Since new definitions and actions have been added into the framework, it is necessary to introduce a set of new rules to describe new properties:

Rule 1.1 : $S \models RRcv\ A\ M \wedge (S \prec t) \Rightarrow t \models \Box\ (Discard\ A\ M)$

This rule says that if an agent receives the same message more than once, then the agent will always discard this message.

Rule 1.2 : $(S \models Rcv\ A\ M)\ \wedge\ (t \models Rcv\ A\ M) \Rightarrow (t \models RRcv\ A\ M)\ \wedge\ (S \prec t)$.

This rule says that if an agent receives a message at moment $S$ and receives the same message at moment $t$, then the agent receives the message more than once.

Rule 1.3 : $(S \models Rcv\ Spy\ M) \wedge (S \models Block\ Spy\ M) \Rightarrow \forall X.(S \models (Neg\ (Rcv\ X\ M)))$.

The rule says that if the attacker receives a message and blocks it, then other agents in the network cannot receive this message.

Eavesdropping rule : $(S \models Send\ A\ B\ M) \wedge (S \prec t) \Rightarrow (t \models Rcv\ Spy\ M)$.

The eavesdropping rule says if agent $A$ sends a message to agent $B$ then the attacker can eavesdrop this message.

### C. 4-Way Handshake Model in Isabelle

Normally, a protocol is written in informal language as shown in figure 1. In this part we will therefore formalise the steps of the 4-way handshake as four formal formulas for all honest agents as follows:

1) FHShake1 : $S \models Send\ AP\ SP\ (\{Agent\ AP, \{Nonce\ ANonce, \{Mag\ Msg1,\ Nonce\ (SN)\}\}\})$.
2) FHShake2 : $(S \models Send\ AP\ SP(\{Agent\ AP, \{Nonce\ ANonce, \{Mag\ Msg1,\ Nonce\ (SN)\}\}\})) \Longrightarrow t \models Send\ SP\ AP\ (\{Agent\ SP,\ \{\ Nonce\ SNonce, \{Mag\ Msg2, \{Nonce\ (SN),\ MIC\,\{Nonce\ SNonce, \{Mag\ Msg2,\ Nonce\ (SN)\}\}\ k\}\}\}\}) \wedge (S \prec t)$.
3) FHShake3 : $S \models Send\ SP\ AP\ (\{Agent\ SP, \{\ Nonce\ SNonce,\ \{Mag\ Msg2, \{Nonce\ (SN), MIC\,\{Nonce\ SNonce, \{Mag\ Msg2,\ Nonce\ (SN)\}\}\ k\}\}\}\}) \wedge (S \prec t) \Longrightarrow t \models Send\ AP\ SP\ (\{Agent\ AP, \{Nonce\ ANonce, \{Mag\ Msg3, \{Nonce\ (SN1), MIC\,\{Nonce\ ANonce, \{Mag\ Msg3,\ Nonce\ (SN1)\}\}\,k\}\}\}\})$.
4) FHShake4 : $S \models Send\ AP\ SP(\{Agent\ AP, \{Nonce\ ANonce, \{Mag\ Msg3, \{Nonce\ (SN1), MIC\,\{Nonce\ ANonce, \{Mag\ Msg3, Nonce\ (SN1)\}\}\,k\}\}\}\}) \Longrightarrow t \models Send\ SPAP\ (\{Agent\ SP, \{Mag\ Msg4, \{Nonce\ (SN1), MIC\,\{Mag\ Msg4,\ Nonce\ (SN1)\}\,k\}\}\}) \wedge (S \prec t)$.

The access point will re-send message 1 and message 3 if it did not receive the reply during the per-defined time interval. The access point will continue to re-send and, after timeout, will de-authenticate the wireless device if there is no reply from it. There are two rules for re-sending the message 1 and message 3, as described below:

ReplayMessage1 : $S \models \neg(Rcv\ AP\ (\{Agent\ SP, \{\ Nonce\ SNonce,\ \{Mag\ Msg2, \{Nonce\ (SN), MIC\,\{Nonce\ SNonce, \{Mag\ Msg2,\ Nonce\ (SN)\}\}\ k\}\}\}\})) \wedge (S \prec t) \Longrightarrow S \models Send\ AP\ SP\ (\{Agent\ AP, \{Nonce\ ANonce, \{Mag\ Msg1,\ Nonce\ (SN)\}\}\})$.

ReplayMessage3 : $S \models \neg(Rcv\ SP\ (\{Agent\ SP, \{Mag\ Msg4, \{Nonce\ (SN1),\ MIC\,\{Mag\ Msg4, Nonce\ (SN1)\}\,k\}\}\})) \wedge (S \prec t) \Longrightarrow t \models Send\ AP\ SP\ (\{Agent\ AP, \{Nonce\ ANonce, \{Mag\ Msg3, \{Nonce\ (SN1),\ MIC\,\{Nonce\ ANonce, \{Mag\ Msg3, Nonce\ (SN1)\}\}\,k\}\}\}\})\ \wedge\ (t \prec outtime)\ \wedge\ (intervaltime \prec t)$.

The attacker has the ability to block any messages over the network. So if any agent sends a message, the attacker can block it and the recipient will not be able to receive it. The rule for blocking message 4 can be represented in the framework as follows:

BlockMessage4 : $S \models Send\ SP\ AP\ (\{Agent\ SP, \{Mag\ Msg4, \{Nonce\ (SN1),\ MIC\,\{Mag\ Msg4,$

$Nonce\ (SN1)\}\ k\}\}\}) \Longrightarrow S \models Block\ Spy\ (\{Agent\ SP, \{Mag\ Msg4, \{Nonce\ (SN1),\ MIC\ \{Mag\ Msg4, Nonce\ (SN1)\}\ k\}\}\})$.

### D. Proving Basic Properties

The first basic property is discarding the received messages. The reason for an agent discarding a received message is because the same message is received more than once. So for security reasons the agent should discard the duplicate copies of a message. The first property says that when agent A sends a message to agent B more than once, then agent B will always discard the message:

lemma DiscardReceivedMessage : $(S \models Send\ A\ B\ M) \wedge (S \wedge t) \Longrightarrow (t \models \Box\ (Discard\ B\ M))$

apply (rule Rule 1.1)
apply (rule Rule 1.2)
apply (rule conjI)
apply (rule Rule 8)
apply (auto)
apply (rule Rule 8)
apply (auto)
done.

The second basic property is a special case of the blocked message 4 by the attacker. This property says that when the wireless device sends message 4, then the message may not be received by the access point if it is blocked by the attacker.

lemma NotReceivedMessage4FromSender :
$S \models Send\ SPAP\ (\{Agent\ SP, \{Mag\ Msg4, \{Nonce\ (SN1),\ MIC\ \{Mag\ Msg4,\ Nonce\ (SN1)\}\ k\}\}\})$
$\wedge\ (S \prec t) \Longrightarrow S \models \neg(Rcv\ AP\ (\{Agent\ SP, \{Mag\ Msg4, \{Nonce\ (SN1),\ MIC\ \{Mag\ Msg4,\ Nonce\ (SN1)\}\ k\}\}\}))$

apply(rule allE)
apply(rule Rule 1.3)
apply(auto)
apply(rule Eavesdropping rule)
apply (auto)
apply (rule BlockMessage4)
apply (auto)
done.

## IV. Denial of Service Attack on the Protocol

In a DoS attack, the adversary prevents or inhibits protocols from completing successfully. Simply speaking, it involves disabling or preventing servers who are required to interact with participants. Most protocols have the potential to be attacked by DoS; however, the design of a protocol could improve prevention, or make such attacks more unlikely [13]. It is impossible to fully protect protocols against DoS attacks.

### A. DoS Attack on the 4-way Handshake

The sequence number ($sn$) is a technique used to prevent reply attacks in the 4-way handshake. $sn$ is a counter set to 0 when establishing PMK then incremented with successive messages. The wireless device and the access point assume that they have the same $sn$ value before running the 4-way handshake. During running the 4-way handshake the wireless device should update the $sn$ value when receiving the message 3, while the access point should update the $sn$ value after receiving the message 4 as shown in figure 1. As a result, at the end of the 4-way handshake we assume that they will have the same $sn$ value. If the wireless device and access point have different $sn$ values at the end of the 4-way handshake they will de-authenticate each other and cannot start future sessions.

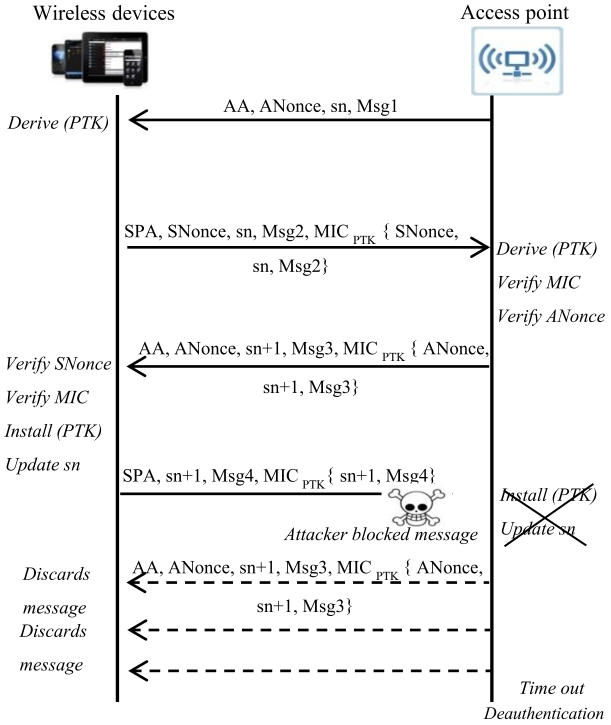

Figure 2. An Attack on the 4-way Handshake

The $sn$ value can be a potential vulnerability in the 4-way handshake. The wireless device and the access point will continue running the 4-way handshake until time out without knowing the attacker having blocked message 4. The access point will re-send message 3 if it does not receive message 4 while the wireless device discards these messages as shown in figure 2. This attack happens because each side has different values of $sn$. A simple effort of the attacker, which blocks message 4 once then lets the protocol run as usual, can destroy the authentication between the wireless device and the access point. It is easy for the attacker to detect message 4 over the network because the 4-way handshake exchanges messages without encryption.

One of the security properties in the 4-way handshake is synchronising the installation of session keys. The wireless device installs the session key after receiving message 3 and the access point will install it after receiving message 4. In Isabelle tool, a lemma shows that the wireless device will discard message 3 resent by the access point. As a result, if the wireless device discards message 3, message 4 will not be received by the access point and the access point cannot install the session keys. The proving scripts of this lemma are as follows:

lemma FindAttackinFourhandshack : $S \models \Box(Discard\ SP$
$(\{Agent\ AP, \{Nonce\ ANonce, \{Mag\ Msg3,$
$\{Nonce\ (SN1),\ MIC\,\{Nonce\ ANonce, \{Mag\ Msg3,$
$Nonce\ (SN1)\}\}\,k\}\}\}\}\})).$

```
apply(rule DiscardReceivedMessage)
apply(rule ReplayMessage3)
apply(rule NotReceivedMessage4FromSender)
apply(rule FHShake4)
apply(rule FHShake3)
apply(rule FHShake2)
apply(rule FHShake1)
done
```

PMK is important to reduce the authentication process costs that occur every time PTK is established or updated. If PMK has already existed they can run the 4-way handshake to obtain new PTK for transferring the data over the network. If the attacker can block message 4 then at the end of the 4-way handshake, PMK will be invalid and 802.1X authentication need to be run every time. As we know, the attacker has the ability to block any messages over the network, therefore it is easy for the attacker to block specifically message 4. Consequently, the attacker can de-authenticate the access point with all wireless devices wanting to connect during the 4-way handshake phase. The DoS attack identified in this paper is easy to implement over the network and it is difficult to prevent. A number of attempts to prevent DoS proposed by some researchers failed to defend against all DoS in the 4-way handshake.

### B. Preventing the DoS Attacks on Message 4

A number of researchers have discussed and proposed solutions to avoid the DoS attack in the 4-way handshake. He and Mitchell provided two solutions to avoid DoS attacks on the wireless device side [14, 15]. In addition, Rango et al. discussed the He and Mitchell solutions and introduced two new solutions to prevent DoS attacks [16]. Unfortunately, all these solutions are unsuitable to prevent the DoS attack identified in this paper. Therefore, we are going to introduce a new solution for this attack.

The $sn$ value plays an important role in preventing replay attacks. The access point usually checks the $sn$ value of received message corresponding to the outstanding message. Whereas, the wireless point checks the $sn$ value used before with current PMK. Moreover, when the access point does not received reply message during the timeout interval, it will keep re-sending the message until time out. The resent message has the same contents as the original message and is valid from the point of view of the access point. The wireless device will likely discard the resent message if it has seen the original message, which is valid for the access point. So there is a contradiction between using $sn$ value and re-sending the original message. In other words, the recipient will be confused with the resent message and the original message where both are valid.

In order to prevent discarding the valid messages by the wireless device, the access point should update the $sn$ value immediately after sending the message. So, when the access point wants to re-send the original message, a new $sn$ value will be used. The $sn$ value makes the resent message different from the original message and therefore the wireless device is not going to discard the resent message. As shown in figure 3, if the access point has not received message 4 during interval time out, then message 3 will be resent with the new $sn$ value. As a result, the simple amendment will reduce the chance of message 4 being attacked. Also, the wireless device is not going to discard the valid messages.

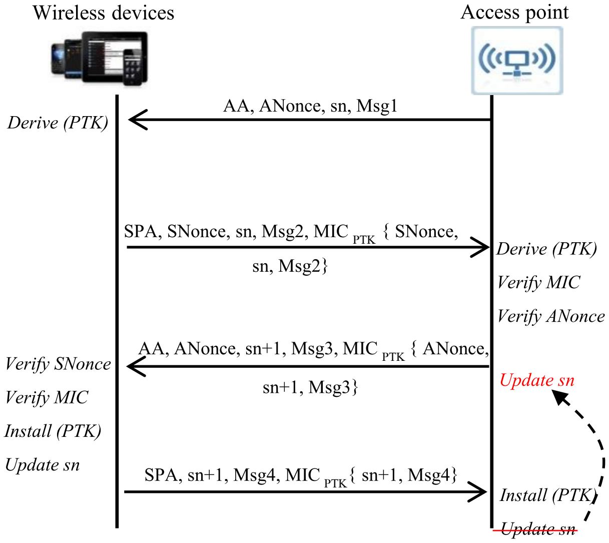

Figure 3. Prove the Updated 4-Way Handshake

### C. Proving the Fix Protocol Using Isabelle

In order to prove that the DoS attack on the 4-way handshake can be prevented the replay message 3 should be modified according to the proposed solution. Suppose that the access point should rename the $sn1$ value to become $sn$ after sending message 3. Meanwhile, the updated $sn$ value will become $sn1$. So, the replayed message 3 will be changed every time it is re-sent; therefore, it is not going to be discarded by the wireless device. The replayed message 3 can be re-write as follows:

ReplayMessage3New : $S \models \neg(Rcv\ SP\ (\{Agent\ SP,$
$\{Mag\ Msg4, \{Nonce\ (SN),\ MIC\,\{Mag\ Msg4,$
$Nonce\ (SN)\}\,k\}\}\})) \wedge (S \prec t) \Longrightarrow t \models Send\ AP\ SP$
$(\{Agent\ AP, \{Nonce\ ANonce, \{Mag\ Msg3,$
$\{Nonce\ (SN1),\ MIC\,\{Nonce\ ANonce, \{Mag\ Msg3,$
$Nonce\ (SN1)\}\}\,k\}\}\}\}) \quad \wedge \quad (t \quad \prec \quad outtime) \quad \wedge$

$(intervaltime \prec t)$.

The following script shows that the access point will keep sending the message 3 until receiving message 4 or the finish time of the session. Whereas, the wireless device is not going to discard the re-sent message 3 because it is not the same as the previous message 3 which has been received.

FixDoSAttack : $t \models Send\ AP\ SP$ $(\{Agent\ AP, \{Nonce\ ANonce, \{Mag\ Msg3, \{Nonce\ (SN1),\ MIC\ \{Nonce\ ANonce, \{Mag\ Msg3, Nonce\ (SN1)\}\}\ k\}\}\}\}) \wedge (t \prec outtime) \wedge (intervaltime \prec t)$

apply(rule ReplayMessage3New)
apply(rule NotReceivedMessage4FromSender)
apply(rule FHShake4)
apply(rule FHShake3)
apply(rule FHShake2)
apply(rule FHShake1)
done

## V. Conclusion and Future Work

The 4-way handshake phase in the IEEE 802.11i standard has been analysed and a DoS attack has been identified. Isabelle tool has been used to implement the linear temporal logic framework. The adjustment of the framework, the modelling of the protocol and the proving of basic properties have been used for analysing the 4-way handshake. More importantly, a new effective DoS attack by blocking message 4 has been identified and analysed.

The protocol uses the $sn$ value to avoid replay attacks in the 4-way handshake. However, the analysis has shown that the $sn$ value will be a flaw if message 4 is not received by the access point. Non-receipt of message 4 can be caused by the attacker or anything else. In this case, the authentication between the wireless device and the access point will fail. Simply updating the $sn$ value after sending message 3 can prevent the attack. Moreover, it is possible for the access point to obtain the reply message for message 3.

## Acknowledgment

This research is supported by Saudi Arabian Cultural Bureau in London, the Ministry of Higher Education in Saudi Arabia and Qassim University.

## References

[1] X. Ma, R. McCrindle, and X. Cheng, "Verifying and fixing password authentication protocol," in *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2006. SNPD 2006. Seventh ACIS International Conference on.* IEEE, 2006, pp. 324–329.

[2] C. He, M. Sundararajan, A. Datta, A. Derek, and J. C. Mitchell, "A modular correctness proof of ieee 802.11 i and tls," in *Proceedings of the 12th ACM conference on Computer and communications security.* ACM, 2005, pp. 2–15.

[3] X. Zha and M. Ma, "Security improvements of ieee 802.11 i 4-way handshake scheme," in *Communication Systems (ICCS), 2010 IEEE International Conference on.* IEEE, 2010, pp. 667–671.

[4] L. DONG, K. F. CHEN, and X. J. LAI, "Formal analysis of authentication in 802.11 i," *Journal of Shanghai Jiaotong University (Science)*, vol. 1, p. 023, 2009.

[5] A. Alabdulatif, X. Ma, and L. Nolle, "A framework for cryptographic protocol analysis using linear temporal logic," in *Information Society (i-Society), 2012 International Conference on.* IEEE, 2012, pp. 525–530.

[6] A. Alabdulatif, X. Ma, and L. Nolle, "A framework for proving the correctness of cryptographic protocol properties by linear temporal logic," *International Journal of Digital Society (IJDS)*, vol. 4, no. 1-2, pp. 749–757, 2013.

[7] C. Boyd and A. Mathuria, *Protocols for authentication and key establishment.* Springer Verlag, 2003.

[8] X. Xing, E. Shakshuki, D. Benoit, and T. Sheltami, "Security analysis and authentication improvement for ieee 802.11 i specification," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE.* IEEE, 2008, pp. 1–5.

[9] "Ieee standard for information technology- telecommunications and information exchange between systems- local and metropolitan area networks- specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 6: Medium access control (mac) security enhancements," 2004.

[10] J. Edney and W. A. Arbaugh, *Real 802.11 security: Wi-Fi protected access and 802.11 i.* Addison-Wesley Professional, 2004.

[11] G. Bella, *Formal correctness of security protocols.* Springer Verlag, 2007.

[12] L. C. Paulson, "Inductive analysis of the internet protocol tls," *ACM Transactions on Information and System Security (TISSEC)*, vol. 2, no. 3, pp. 332–351, 1999.

[13] P. Eronen, "Denial of service in public key protocols," in *Proceedings of the Helsinki University of Technology Seminar on Network Security (Fall 2000).* Citeseer, 2000.

[14] C. He and J. C. Mitchell, "Analysis of the 802.11 i 4-way handshake," in *Proceedings of the 3rd ACM workshop on Wireless security.* ACM, 2004, pp. 43–50.

[15] C. He and J. C. Mitchell, "Security analysis and improvements for ieee802.11i," in *11th Annual Network and Distributed System Security Symposium (NDSS'05).* The Internet Society, Feb 2005.

[16] F. D. Rango, D. C. Lentini, and S. Marano, "Static and dynamic 4-way handshake solutions to avoid denial of service attack in wi-fi protected access and ieee 802.11 i," *EURASIP Journal on Wireless Communications and Networking*, vol. 2006, no. 2, pp. 73–93, 2006.