SECURE AND EFFICIENT KEY EXCHANGE MECHANISM IN IEEE 802.11

A DISSERTATION

Submitted in partial fulfillment of the requirements for the award of the degree of

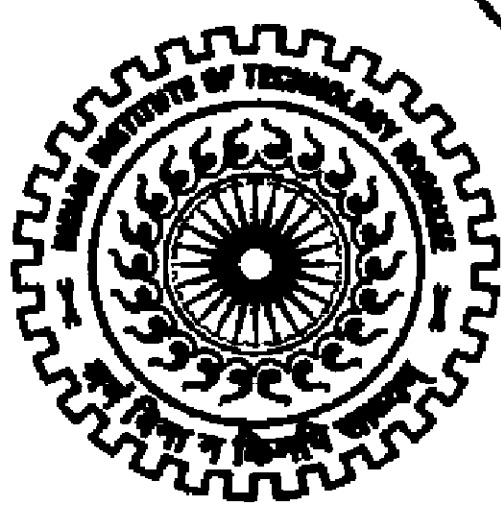
MASTER OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

By

DINESH YADAV



DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE -247 667 (INDIA)

JUNE, 2011

CANDIDATE'S DECLARATION

I hereby declare that the work, which is being presented in the dissertation entitled **"SECURE AND EFFICIENT KEY EXCHANGE MECHANISM IN IEEE 802.11"** towards the partial fulfillment of the requirement for the award of the degree of **Master of Technology in Information Technology** submitted in the Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand (India) is an authentic record of my own work carried out during the period from July 2010 to June 2011, under the guidance of **Dr. Anjali Sardana, Asst. Professor,** Department of Electronics and Computer Engineering, IIT Roorkee.

The matter presented in this dissertation has not been submitted by me for the award of any other degree of this or any other Institute.

Date: 1/6/2011

Place: Roorkee

Dinesh

(DINESH YADAV)

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Anjali Sardana 1/6/11

Date: 1/6/2011

Place: Roorkee

(Dr. Anjali Sardana)

Asst. Professor

Department of Electronics and Computer Engineering

IIT Roorkee.

ACKNOWLEDGEMENTS

First and foremost, I would like to extend my heartfelt gratitude to my guide and mentor **Dr. Anjali Sardana**, Astt.Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, for her invaluable advices, guidance, and encouragement and for sharing her broad knowledge. Her wisdom, knowledge and commitment to the highest standards inspired and motivated me. She has been very generous in providing the necessary resources to carry out my research. She is an inspiring teacher, a great advisor, and most importantly a nice person.

I am greatly indebted to all my friends, who have graciously applied themselves to the task of helping me with ample moral supports and valuable suggestions.

On a personal note, I owe everything to the Almighty and my parents. The support which I enjoyed from my father, mother and other family members provided me the mental support I needed.

DINESH YADAV

Abstract

Since the IEEE 802.11 standard was released in its first version in 1997, IEEE 802.11 based wireless LANs (also called WLANs) quickly evolved to the most commonly used technology to wirelessly connect devices to an IP network. However, due to lack of security mechanisms, major security amendments have been done in the draft of IEEE 802.11i in 2004. In IEEE 802.11i for authentication purpose, Extensible Authentication Protocol (EAP) and for key exchange, 4-way handshake protocol is used. Authentication gives the ability to authenticator and supplicant to check and prove each other's identity. Key exchange provides the facility to exchange shared secret between authenticator and supplicant which is used for encrypting the data or we can say for transmitting the data with confidentiality.

The key exchange mechanism provided in IEEE 802.11i is not secure because of its Message 1 or 1st message sent from authenticator to supplicant. The Message 1 of 4-way handshake protocol does not have any type of encryption and can be forged. This makes it vulnerable to Denial of Service (DoS) attack and Dictionary attack. Due to the existing design flaws, 4-way handshake is incapable in providing the required security and performance.

We propose a new technique for key exchange which is able to provide an enhance security in comparison with 4-way handshake protocol. This enhanced 3-way handshake mechanism is able to provide security against DoS attacks, dictionary attacks and passive attacks. In 3-Way handshake mechanism, three messages are exchanged for generating the pairwise transient key (PTK). The messages which are transmitted in the proposed mechanism are encrypted using New Encryption Key (NEK). This NEK is generated with the help of Pairwise Master Key (PMK) and second Pre-Shared Key (SPK).

The proposed model has been verified analytically and simulated using CPN Tool and results show that in addition with enhanced security, our mechanism performs better and can reduce the communication and computation overheads.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction and Statement of the Problem

Conventional Internet users have been bound to wired connections. Wireless communications, however, have broken this restriction and provide ubiquitous access to the Internet. In addition, increased flexibility strongly motivates wireless network technologies. Today, the deployment of wireless local area networks (WLANs) is sometimes even more economical and efficient than installing wired networks in a whole building.

IEEE 802.11[1] is a set of standards for implementing WLAN. Table 1.1 shows various standards of IEEE 802.11. The IEEE 802.11 standard for WLANs is one of the most widely adopted standards for broadband wireless Internet access because besides mobility and flexibility, it provides quick and easy setup and fast data transfer rates. Today, most of the data transfer is being carried out wirelessly. Wireless network are now being used in real life almost everywhere like in hospitals, universities, airports etc. Security was originally purposefully weak due to export requirements of some governments, and was later enhanced via the 802.11i amendment after governmental and legislative changes in 2004.

Table1.1 IEEE 802.11 Standards

Standard	Release	Frequency(GHz)	Data Rate Max (Mb/sec)	Range(indoor) (meter)	Security
802.11a	1999	5	54	35	WEP
802.11b	1999	2.4	11	38	WEP
802.11g	2003	2.4	54	38	WEP
802.11i	2004	Only Security Amendments			WPA/WPA2
802.11n	2009	2.4/5	72/150	70	WPA/WPA2

1.1 Motivation

As IEEE 802.11 have used in real life everywhere for data transfer like in universities, airports etc. Therefore, lack of security can be harmful for financial data and other network resources. Authentication and key exchange are two important processes for the secure communication. Authentication provides the ability to authenticator and supplicant that they can mutually authenticate themselves to each other or we can say that they can prove their identities using authentication mechanism to each other. In absence of a secure authentication mechanism, any malicious node can pretend to be authenticator or supplicant and steal the data or harm the network. Key exchange gives the shared secret to both the authenticator and supplicant which can be further used for encrypting the data. In absence of secure key exchange mechanism, intruder can easily get the shared key and decrypt the transmitted data in the network.

Security over a wireless environment is more complicated than in a wired environment. Due to the wide open nature of wireless radio, many attacks could make the network insecure. This makes authentication and key exchange a challenging area in case of IEEE 802.11. Initially IEEE 802.11 was providing the security and authentication using Wired Equivalent Privacy (WEP) protocol. But major design flaws have been indicated in [2-7]. Therefore, a Temporal Key Integrity Protocol (TKIP) was introduced by Wi-Fi Alliance to provide better security through Message Integrity Code (MICHAEL), Sequence counter (TSC) and a key mixing function. Instead of using open system authentication or shared key authentication mechanism, a new mechanism called IEEE 802.1X/Extensible Authentication Protocol (EAP) was developed for authentication.

In 2004, IEEE 802.11i [8] was proposed for providing data confidentiality, integrity and replay protection. In it, for authentication purpose combination of IEEE 802.1X authentication method and key management procedure 4-way handshake protocol were used.

This 4-way handshake protocol is not secure against Denial of Service (DoS) and Dictionary attacks. Therefore, a new key exchange mechanism has been proposed here which can provide security against these attacks. These additional features which are

added in the proposed solution can also make the key exchange secure against bogus authenticator or supplicant and passive attacks.

1.2 Statement of the Problem

Problem statement: To design a secure and efficient key exchange mechanism for IEEE 802.11 that addresses security attacks with reduced communication overheads.

This problem can be subdivided into following parts:

1. Design of 3-way handshake mechanism that provides resistance against DoS attacks.

2. Make the Message 1 of the proposed protocol secure to address dictionary attacks using encryption.

3. Simulate and validate the proposed key exchange mechanism, CPN Tool has been used to verify the protocol.

1.3 Organization of the Report

This dissertation report comprises of six chapters including this chapter that introduces the topic and statement of the problem. The rest of the report is organized as follows.

Chapter 2 describes the 4-way handshake protocol and flaws in the 4-way handshake protocol. It also tells about various solutions which has been proposed over the period of time and the research gaps which are still not addressed.

Chapter 3 describes the proposed solution enhanced 3-way handshake protocol for making key exchange in IEEE 802.11 more secure.

Chapter 4 gives the simulation details of the proposed solution, details of experiments performed details of the experimental tool.

Chapter 5 describes the results of simulation and gives a brief discussion over these results.

Chapter 6 concludes the dissertation work and gives suggestions for future work.

Chapter 2
Background and Literature Review

2.1 4-Way Handshake Protocol

The 4-way handshake protocol is executed as shown in the Figure 2.1. In the form of first message the access point sends the random number ANonce. In response, the supplicant generates another random number, SNonce, and sends it to the AP with message integrity code (MIC) using PTK. The PTK is generated with the help of PMK, MAC addresses of AP and the supplicant, and ANonce and SNonce [9].

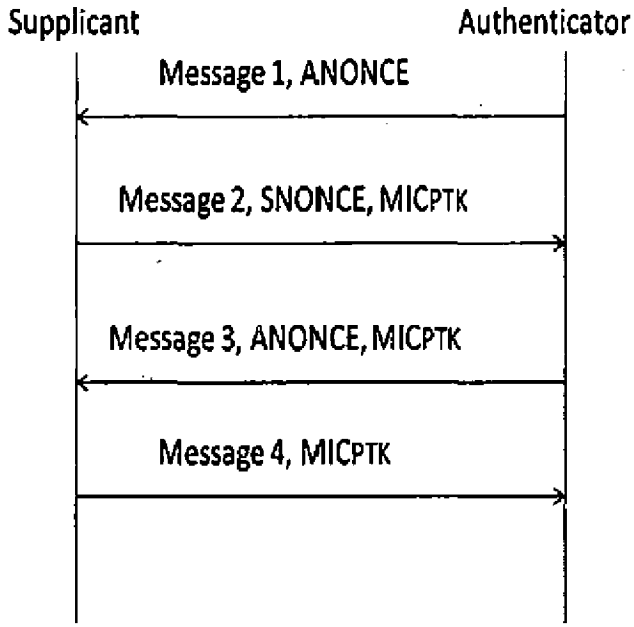


Figure 2.1 4-way handshake protocol

Now a third message is sent by the AP after generating PTK and verifying MIC. Now supplicant verifies MIC of message 3 and sends a MIC and install PTK at supplicant. After receiving message 4, AP also installs PTK. PTK is divided into three parts: Key Confirmation Key (KCK), Key Encryption Key (KEK) and Temporal Key (TK). KCK is used to authenticate message 2, 3 and 4; KEK is used to protect group key distribution and TK is used to provide confidentiality during subsequent data transmissions [10].

The authenticator can refresh the PTK either periodically or upon the request from the supplicant by running another 4-way handshake with the same PMK. Authenticator and

supplicant silently discard the received message having erroneous MIC. When the supplicant does not receive message 1 within the expected time interval after a successful IEEE 802.1X authentication [11,12], it will dissociate, de-authenticate and try another authenticator. On the other hand, the authenticator will timeout and retry the message if it does not receive the expected reply within the configured time interval [13].

2.2 Flaws in 4-way handshake protocol

2.2.1 Vulnerable to DoS attack

In 4-way handshake protocol, message1 is totally unprotected; therefore DoS attacks can be performed [14]. For this purpose an adversary sends a fake message1 with a different ANonce' to the supplicant, before the message3 is sent by authenticator as shown in the Figure 2.2.

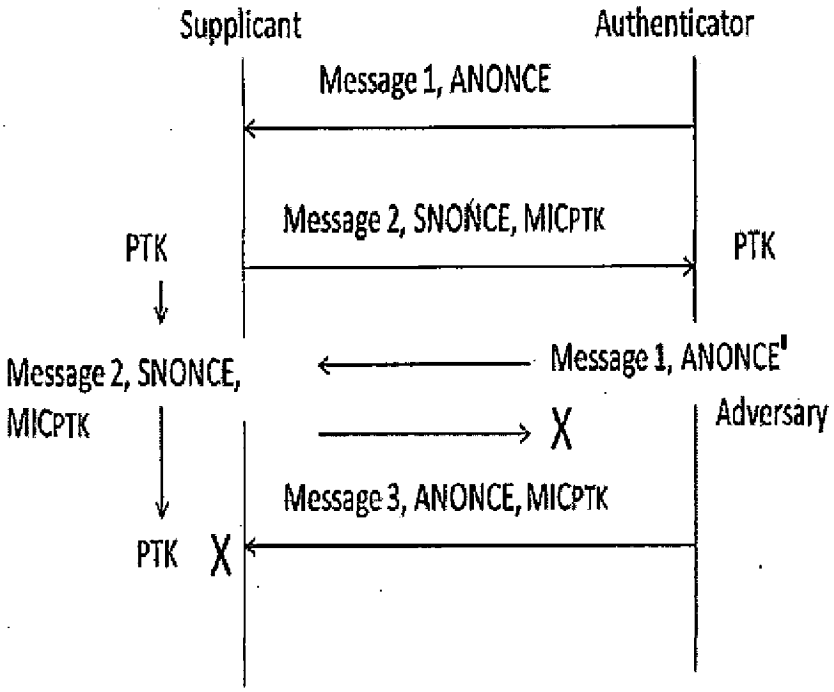


Figure 2.2 DoS attack on 4-way handshake protocol

The supplicant treat it as a retransmission from authenticator and generate a new PTK' using ANonce' and then sends message2' to authenticator. This message2' is discarded and PTK synchronization is disrupted. Now supplicant has PTK' and authenticator is

having PTK, so message3 sent by authenticator is invalidated by PTK' at supplicant and silently dropped and 4-way handshake protocol fails this way.

2.2.2 Vulnerable to dictionary attack

In WPA and WAP2 protocol, 4-way handshake performs the key management role as refreshing the temporal key for data encryption. As the original design, there exists vulnerability in 4-way handshake stage, some attacking tools such as Aircrack can crack the PMK key using dictionary attack [15, 16].

2.3 Proposed Solutions

2.3.1 Message1 Authentication

Since there is already some common secret (PMK) shared between the authenticator and the supplicant, another possible repair is to add a MIC to Message 1, which will prevent the attacker from forging that message. In order to exploit the same hardware or software as in processing other messages, a trivial PTK can be derived based on the PMK and some specific values of nonces (e.g., 0), then calculate the MIC with this derived PTK. Note that after a MIC is added, Message 1 and Message 3 are still distinguishable by the Secure bit [17].

If the PMK is dynamically generated through an 802.1X authentication process, this would solve the problem. However, if a PSK or a cached PMK is used for the current PMK, the authenticated Message 1 is still vulnerable to replay attacks since the PMK is static for a relatively long time. Therefore, the authenticator should keep a monotonically increasing sequence counter to defend against the replay attacks. One global sequence counter per authenticator appears to work for all supplicants. The supplicant can detect the replayed messages by comparing the counter of a received message against the counter of the largest-Numbered previous message.

Fortunately, the requirement that the counter must be monotonically increasing appears feasible since there are apparently 8 octets set aside for this sequence counter. In fact, there appears to be sufficient space in the message format so that clock time could be used as the counter value, eliminating the possible problem of counter rollover. Furthermore, his specific sequence counter is also consistent with its

usage in the group key handshakes and imposes no significant influences on other parts of the standard.

2.3.2 2-way Handshake Protocol

A 2-way handshake protocol is also proposed in [18]. As shown in the Figure 2.3, the authenticator sends a message encrypted by PMK having ANonce, a big random number, RNonce, and other elements as in message1 of the 4-way handshake. After receiving this message, the supplicant generates PTK using ANonce and sends SNonce and RNonce. Then, the authenticator verifies the RNonce and installs PTK.

This method looks perfect but PMK is used for symmetric encryption of first message, which is a big flaw. In [15], it is shown that some tools like aircrack [16] can crack the PMK using dictionary attack, therefore it is not secure to encrypt by PMK.

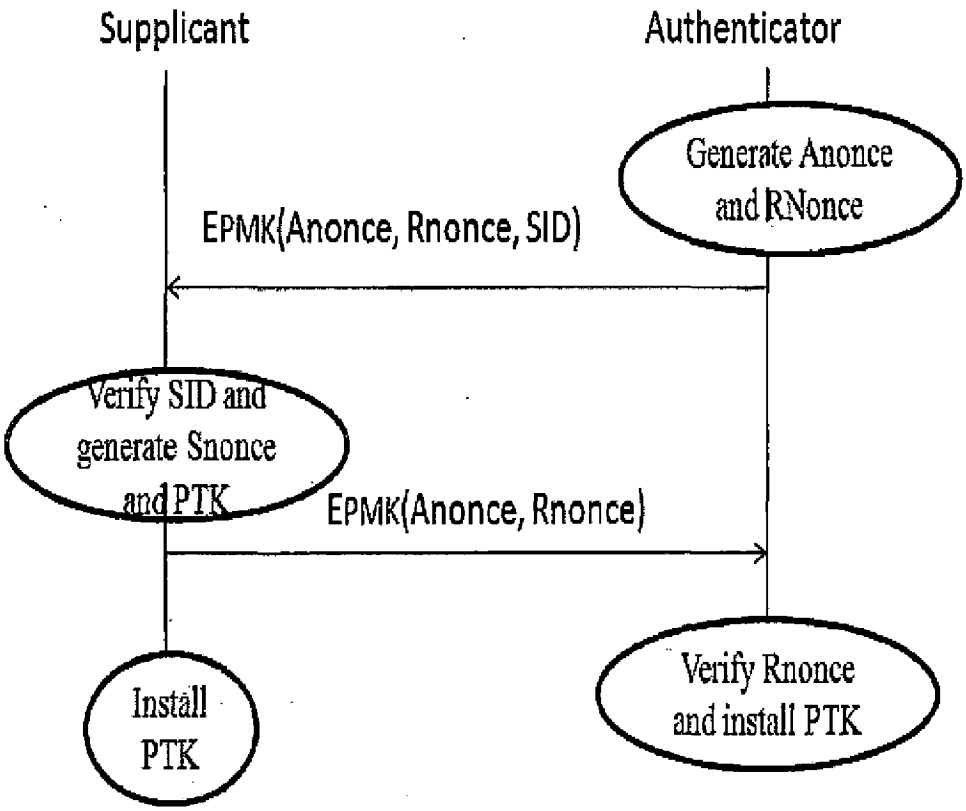


Figure 2.3 2-way handshake protocol

2.3.3 Multi-key Encryption Scheme

Due to dictionary attack flaw, a Multi-Key Encryption (MKE) mechanism was proposed to enhance the key management state in 802.11i [15] as shown in Figure 2.4. In this scheme, SPK (second pre-share key) has been introduced which installed on both authenticator and supplicants and the length is 32 byte just like PMK.

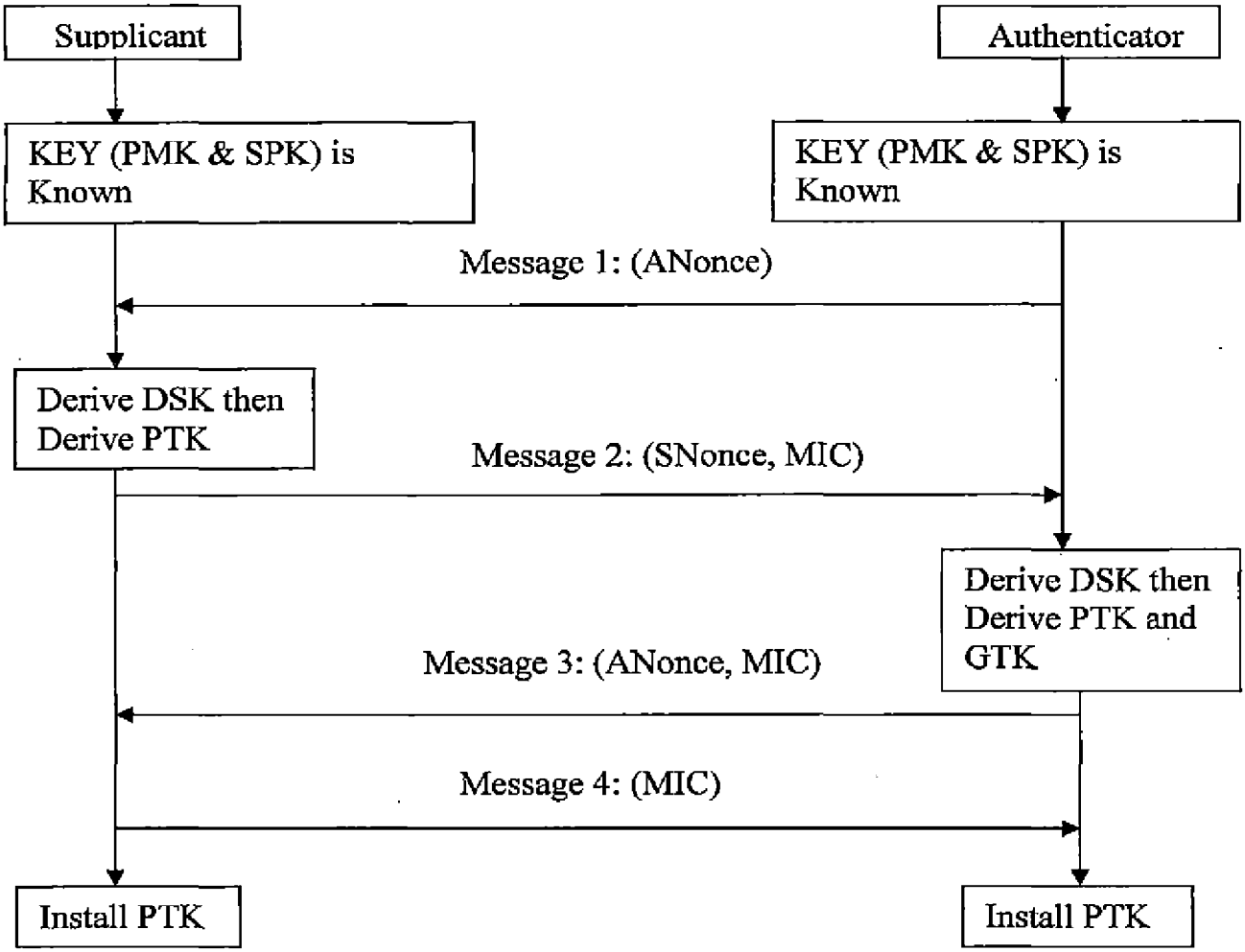


Figure 2.4 4-way handshake protocol using MKE

The procedure of constructing PTK is modified in following equations (where DSK stands for Derived Second pre-share Key):

$DSK = PRF\text{-}256\ (ANonce, SPK, Authenticator_{MAC}, Supplicant_{MAC})$

$PTK = PRF\text{-}384\ (DSK, SNonce, PMK, Authenticator_{MAC}, Supplicant_{MAC})$

All the other process besides generation of PTK is same as in 4-way handshake protocol. But this encryption scheme only address the dictionary attack and the proposed solution is still vulnerable to DoS attacks.

2.3.4 Random - Drop Queue

In this solution, the supplicant can keep a queue of all the initiated, but incomplete, handshake instances. The queue size might be too large for the supplicant; the situation becomes even worse if a longer timeout period or a higher data rate is implemented. Therefore, a feasible improvement would be to implement the queue with a random-drop policy. The supplicant maintains a certain size of queue, say, Q entries to store the states. Once all entries in the queue are filled, one of them is randomly replaced by the new state. when $Q = 1$, the attacker can block the handshake with probability 1 by inserting only one message. When Q increases, the attacker needs to insert more messages in order to block the handshake with a high probability. However, increasing Q could be quite expensive and performance reductive for the supplicant [17].

2.3.5 Nonce Re-use

The third repair is to eliminate the intermediate states on the supplicant side. Specifically, the supplicant can re-use the values of SNonce until a legitimate handshake is completed and a shared PTK is achieved between the supplicant and the authenticator. In other words, the supplicant does not update its nonce responding to each received Message 1 until Message 3 is received and verified. Note that there are no requirements for the authenticator to re-use the values of ANonce, because the legitimate ANonce will ultimately reach the supplicant via a valid Message 3.

In this approach the supplicant only needs to remember one SNonce of its own, which eliminates the memory DoS attack. Although it is still possible for the attacker to send out forged Message 1s with different nonces, the supplicant need not store every received ANonce and the corresponding PTK. It merely derives a PTK from the stored SNonce and the received ANonce, then computes a MIC from the derived PTK and sends out the corresponding Message 2. Upon receiving Message

3, the supplicant will again derive a PTK from the stored SNonce and the received ANonce, then verify the MIC using the derived PTK.

Once the MIC is verified, Message 4 is sent out and the corresponding PTK can be used as the session key. This approach is a robust solution to the memory exhaustion attack; however, it uses more computation on the supplicant side. Specifically, the PTK is calculated twice for each received nonce: the first time when Message 1 is received, and the second time when Message 3 is received. If the computation power is poor for some devices, flooding Message 3 might cause a CPU exhaustion attack, or substantially decrease the performance because the supplicant needs to re-compute the PTK first, then verify the MIC [17].

Of course, the supplicant can store all the received nonces and the derived PTKs to handle the computation load, but then obviously the memory exhaustion attack recurs. There is a tradeoff here that the supplicant needs to make between the memory consumption and the CPU consumption. If the environment is such that most of the messages are expected to be legitimate, the supplicant can store one copy of the derived PTK and received ANonce, and use them to verify the MIC in received Message 3 directly [19]. The supplicant re-computes the PTK only if the nonce in the message does not match the stored ANonce. This combined approach seems to be the most reasonable solution to the 4-Way Handshake problems.

2.4 Research Gaps

Much vulnerability exists in IEEE 802.11 and existing solutions in the 4-way handshake protocol have been discussed in section 3. Table 2.1 presents the research gaps in the existing solutions.

TABLE 2.1. Security Issues and their proposed solutions in the key exchnage process of IEEE 802.11

S. No.	Solution	Issue addressed	Advantages	Disadvantages
1.	Use random drop queues[17]	DoS attack due to unprotected message 1 of 4-way handshake protocol	More messages are needed to block 4-way handshake.	Increasing queue size is quite expensive and performance reductive and vulnerable in high speed networks.
2.	Nonce Reuse[17]		Eliminate memory DoS attacks.	It can cause CPU exhaustion because of recomputation of PTK and MIC verification.
3.	Message 1 Authentication using sequence number[17]	DoS attack due to unprotected message 1 of 4-way handshake protocol, Passive attack and Bogus authenticator or supplicant	It will prevent the attacker from forging the message.	Vulnerable to replay attacks in case of PSK and cached PMK.
4.	2-way handshake instead of 4-way handshake and encrypt the 1st message by PMK[18]		It costs less communication and computation time and more reliable key management.	Cached PMK and PSK is vulnerable to dictionary attack, so still not secure. In case of loss of message 2, it does not provide any mechanism.
5.	Use Multi key encryption scheme[15]	Dictionary Attack on PMK	It makes the PMK guessing almost impossible.	It does not secure message1 of the 4-way handshake protocol, so still vulnerable to DoS attacks.

The existing solutions do not provide complete security. No single solution addresses various attacks namely DoS attack, Dictionary attack, bogus authenticator or supplicant and passive attacks. For example multi key encryption scheme only addresses the dictionary attack, but the 4-way handshake protocol still remains vulnerable to DoS attack [20]. In case of Nonce reuse, memory DoS attack is eliminated, but CPU exhaustion will be a problem because of recomputation of PTK and MIC [21]. Moreover, most of the above solutions have high memory and communication overhead.

Chapter 3

Proposed 3-Way Handshake Mechanism

3.1 Proposed 3-Way Handshake Mechanism

The proposed solution addresses various flaws in the existing 4-way handshake protocol while reducing communication overhead. In the proposed solution two keys are used for generating the new encryption key (NEK) and this NEK is used for encrypting the messages in our protocol [22].

3.1.1 Overall Design

As shown in the Figure 3.1, three messages are exchanged between supplicant and authenticator. Before this 3-way handshake, EAP authentication has been completed and Pairwise master key (PMK) has been generated. Now both the supplicant and authenticator are having PMK and a second pre-shared key (SPK) which are used for generating the New encryption key (NEK) on both the sides. After that Authenticator generated ANonce which is basically a random number used for generating the unique encryption key. This ANonce and Supplicant Identity which is exchnaged earlier are encrypted using NEK and sent to supplicant.

Supplicant which has already generated the NEK decrypt the Msg1 and verify the SID by comparing it with its own identity. Now another random number SNonce is generated and both the nonces are encrypted with the NEK and sent to the authenticator as Msg2. After receiving Msg2, authenticator decrypts it and verify the ANonce and generate Derived Second Pre-shared Key (DSK) using pseudo random function PRF-256 and generate Pairwise transient key (PTK) using pseudo random function PRF-384 and install the PTK to the authenticator side.

Now the Msg3 is sent to supplicant and supplicant after verification of SID and SNonce generates the DSK and PTK and install the PTK.

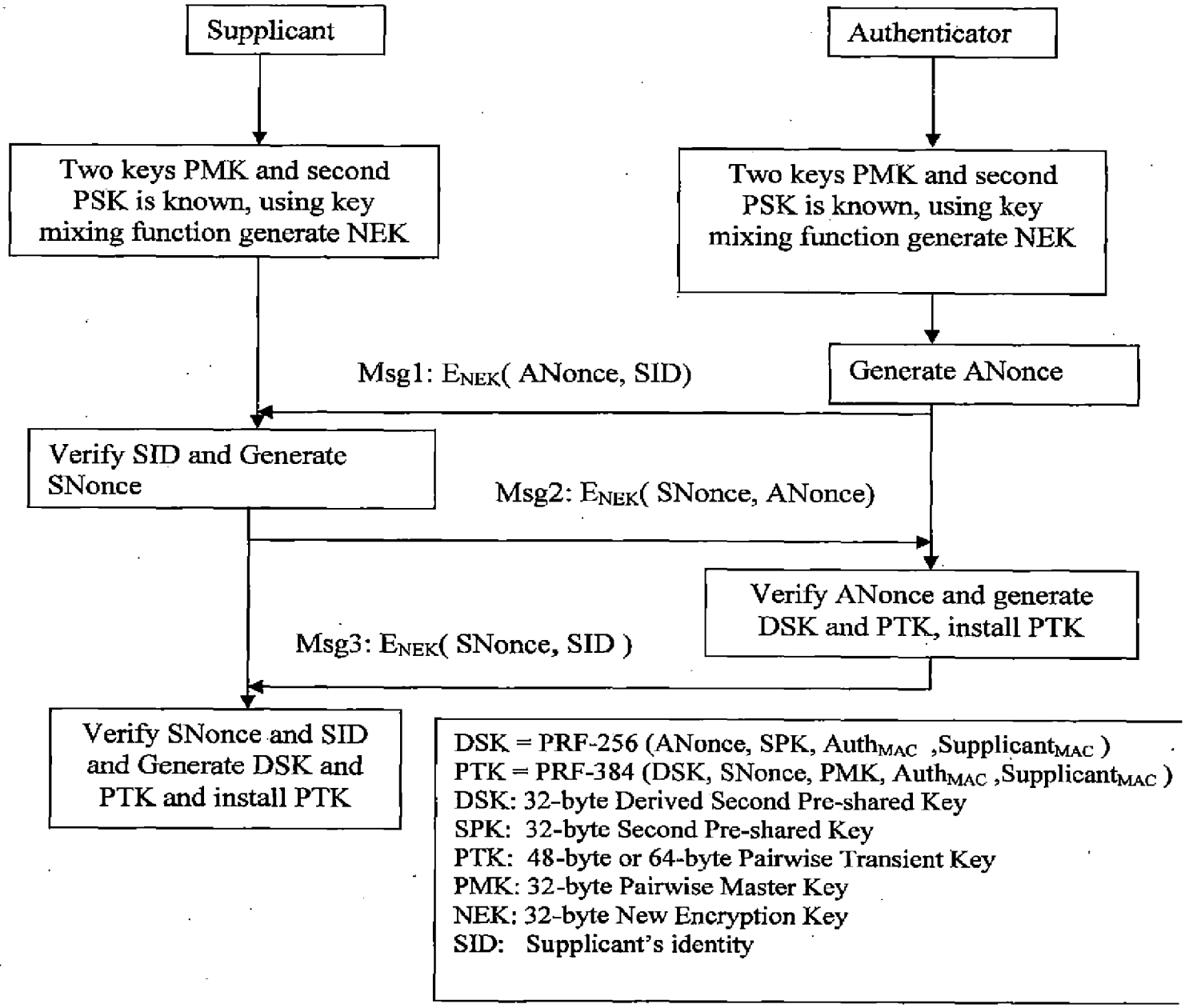


Figure 3.1 3-Way Handshake Protocol

3.1.2 Exchanged Messages

In the proposed mechanism of key exchange total 3 messages are exchanged between Authenticator and supplicant.

Msg1 (Authenticator to Supplicant)

Firstly, the authenticator generates ANonce. The ANonce is same to that used in original 4-way handshake. Secondly, the authenticator picks up a supplicant's identity string, called SID, such as its MAC address. The authenticator has got SID before executing 4-way handshake, for example, the supplicant sent it during EAP authentication phase.

After doing above steps, the authenticator encrypts the ANonce and SID with NEK. Because NEK is 256 bits long, so if we choose AES as cipher algorithm, we should separate NEK into 16 parts and use each part orderly to encrypt the plain materials iteratively. The authenticator packs Msg1 with such cipher materials and other accessorial information, and then sends it to the supplicant.

Msg2 (Supplicant to Authenticator)

On receiving Msg1, the supplicant uses its NEK to decrypt and get ANonce and SID. If SID is not its own identity, the supplicant will judge the authenticator as illegal one and close the connection between them. If identity verification is successful, the supplicant will generate a SNonce.

After doing above steps, the supplicant encrypts the SNonce and ANonce with PMK using the same algorithm as Msg1 encryption. The supplicant packs Msg2 with above cipher materials and other accessorial information, and then sends it to the authenticator.

Msg3 (Authenticator to Supplicant)

On receiving Msg2, the authenticator firstly uses its NEK to decrypt and get SNonce and ANonce. If the ANonce does not equal to the one generated for Msg1, the authenticator will judge the supplicant as illegal one and close the connection between them. If ANonce verification is successful, the authenticator will generate DSK and then PTK based on SNonce, ANonce and other available elements and install PTK.

After doing this, the authenticator encrypts the SNonce and SID with NEK using the same algorithm as Msg1 encryption. The authenticator packs Msg3 with above cipher materials and other accessorial information, and then sends it to the supplicant.

Successful Key Exchange

On receiving Msg3, the supplicant uses its NEK to decrypt and get SNonce and SID. If SID is not its own identity or received SNonce does not match to the SNonce which is generated earlier at supplicant, the supplicant will judge the authenticator as illegal one and close the connection between them. If identity verification is successful, Supplicant generates the DSK and PTK and installs the PTK.

After successful execution of the above mentioned steps, the supplicant and the authenticator verify both identities and generate PTK which is used in the data transmission.

3.2 Analytical analysis of proposed solution

3.2.1 Security analysis

The purpose of original 4-way handshake includes three aspects. First, confirm both sides have the same PMK by checking message integrity code (MIC) of the communicated messages. Second, generate PTK with nonces provided by each side. Third, install PTK to synchronously protect the following process. The proposed 3-way handshake protocol can effectively achieve these purposes without introducing other vulnerability and security weakness.

Moreover it completely addresses various security issues that were partially addressed in existing solution as follows:

1. This proposed 3-way handshake protocol is using two encryption keys for encryption of exchanged messages.
2. In this proposed mechanism unlike 4-way handshake protocol, no message is insecure.
3. In this mechanism, communication overhead is also reduced by reducing the number of exchanged messages.

The above mentioned properties make this protocol secure against following attacks:

1. DoS attack:

Msg1, Msg2 and Msg3 are encrypted, so the intruder cannot forge any message to launch DoS attacks described above.

2. Dictionary attack:

Encryption key is generated using PMK and SPK and PTK is installed using DSK which is created using PMK and SPK, which makes guessing of the keys impossible and as a result dictionary attack is no longer a threat.

3. Passive attack:

All the messages are encrypted during using NEK. After the handshake, the traffic is encrypted using PTK. So there is no information leakage in passive attack.

4. Bogus authenticator or supplicant:

Bogus authenticator or bogus supplicant does not have PMK and SPK, so they must fail in decryption, and cannot get any key information.

3.2.2 Performance Analysis

In aspect of communication time, proposed protocol reduces flows of handshake from 4 ways to 3 ways. As in [23], here we are using the same method of analysis.

Total communication time for 4-way handshake protocol = $2*T_{RTT}$ (3.1)

Total communication time for 3-way handshake protocol = $1.5*T_{RTT}$ (3.2)

Here T_{RTT} = Round trip time of supplicant and authenticator for each communication.

From equation (3.1) and (3.2), total gain (G_{comm}) in communication time is:

$$Gcomm = 0.5*T_{RTT}/(2*T_{RTT}) = 0.25 \sim 25\% \quad (3.3)$$

In aspect of computation time, it avoids MIC calculation and verification which are very time consuming, while symmetric encryption and decryption in our scheme are costless. Furthermore, when mobile supplicant moves from an old authenticator to a new authenticator, the handoff latency caused by re-authentication process will be benefited from these advantages.

3.2.3 Comparison between 3-way and 4-way Handshake Protocols

In the Table 3.1 comparison between messages of 3-way and 4-way handshake protocol is shown. From this table, it can easily be understood that messages of 4-way handshake protocol are vulnerable to DoS and Dictionary attack and produce more communication and computation overhead.

Table 3.1 Comparison between messages of 3-way and 4-way handshake protocol

	4-Way Handshake Protocol	3-Way Handshake Protocol
Msg1	Vulnerable to Dos Attack	Not Vulnerable
Msg2	High Computation and Dictionary Attack possible	Low computation and secure against dictionary attack
Msg3		
Msg4	Just an ACK and increase communication overhead.	Don't exist.

In the Table 3.2, comparison of 3-way handshake protocol with 4-way handshake has been shown. From seeing this comparison, it is explicitly seen that 3-way handshake is being able to provide more security and reliability with better performance.

Table 3.2 Comparison of 3-way Handshake with 4-Way Handshake

	4-way handshake	3-way handshake
Mutual Authentication	Yes	Yes
Key Confirmation	Yes	Yes
Synchronously PTK installation	Yes	Yes
Protection against DoS attack	No	Yes
Protection against Dictionary attack	No	Yes
Computation Overhead	High	Low
Communication Overhead	High	Low

Chapter 4
Simulation Details

In this chapter, we present the simulation details of 4-way handshake protocol and proposed 3-way handshake protocol with and without intruder using colored petrinet (CPN) tool. State space analysis is also done by CPN Tool.

4.1 Overview of Simulation in CPN

For simulation of the proposed solution, CPN Tool has been used because

- CPN Tool has strong formal description capability and well-defined semantics. Its graphical representation and interactive simulation capability help to visually demonstrate concurrency and synchronization of protocol running [24].
- It is a promising tool for describing and studying systems that are characterized as being concurrent, asynchronous, distributed, parallel, and nondeterministic.
- It has many well studied mathematical analysis methods like reachability tree, matrix equations, place and transition invariants. They are used to verify whether a system model could provide structural and behavior properties, such as liveness, boundedness, fairness and home properties [25-27].

Liveness property assumes that if the authenticator sends first message, it will receive last message definitely. It means the handshake executes successfully.

Fairness determines whether the set of transition instances (specified in the list) is impartial or fair.

CPN can test whether the deadlock appears in the modeled system or not. Deadlock means that the protocol will unexpectedly terminate in the case of resource accessing conflict or unlimitedly waiting for acknowledge packets.

Boundedness calculates the maximal or minimal number of tokens on a place.

CPN based simulation and analysis methods consist of these following three steps:

- To give the CPN model of the protocol. Hierarchical CPNs are always used to demonstrate both protocol framework and functional details.
- To give the formal specification of the substitution transitions.

- The last step is to run state space analysis to verify and analyze the results. Passing the verification means the protocol can work according to the given specification. We use a top-down modeling approach. At the highest level of abstraction, an entity is modeled as a substitution transition. Each substitution transition is defined in a separate subpage that provides a lower level description of the behavior of the entity.

Simulation using CPN Tool is consisting of these steps:

1. Create the model of the protocol without an intruder.

- For this purpose, standard modeling language notations are used. Using these notations, we declare the color sets, functions, variables, and constants that will be used in the net inscriptions of the CPN model.
- We build a top-level model for ease of understanding in which the various entities are modeled as substitution transitions.
- Then we define the substitution transitions in the sub pages and connect them to each other and to the top-level page.

2. Create the intruder and add it to the protocol model which is created previously.

- We extend the CPN declarations to include the intruder.
- We add the intruder transition to the top-level model.
- We define the intruder's substitution transition.

3. In this step state space analysis has been done and various properties like liveness, boundedness, fairness and home properties are checked.

4.2 4-Way Handshake Protocol without Intruder

The 4-way handshake protocol, which is standard for key exchange in IEEE 802.11and described in chapter-2, is modeled with the help of CPN Tool. The following figures are showing the hierarchical CPN model of 4-way handshake protocol. In this protocol 4 messages are exchanged between authenticator and supplicant.

4.2.1 The Top Level Model

In the Figure 4.1, the top level page of 4-way handshake protocol is shown. It presents the 4-way handshake protocol in modular way. In CPN, this is implemented by using substitution transitions. First, we focus on the messages exchanged between the protocol entities.

At this level, protocol entities are modeled as transitions. Authenticator and supplicant are shown as substitution transitions and they are connected to each other via places named A, B, C and D. These places are doing the work of interconnection between the authenticator and supplicant sub-pages.

Here the place A represents the Message 1 of 4-way handshake which consists of ANonce and sent by authenticator to supplicant. Place B represents the Message 2 which consists of SNonce and MIC calculated by PTK and sent by supplicant to authenticator. Place C represents the Message 3 containing ANonce and MIC calculated by PTK and sent by authenticator to supplicant. Place D represents the Message 4 which is the acknowledgement of Message 4 and having MIC and this Message 4 is sent by authenticator to supplicant.

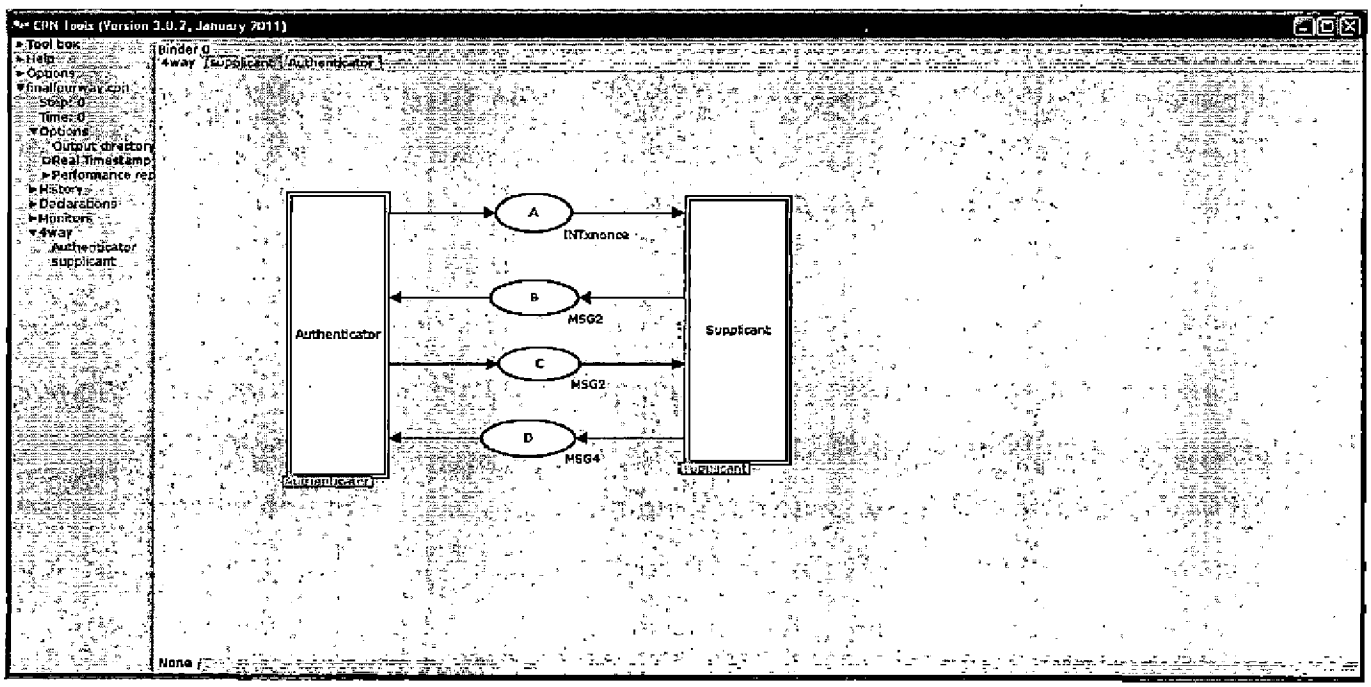


Figure 4.1 Prime page of 4-way handshake CPN Model

4.2.2 Model of Authenticator

In Figure 4.2, the sub-page of authenticator substitution transition has been modeled. It contains three subnets: one model the generation of Message 1 and the process of transferring this message to the supplicant, second is the model of receiving the Message 2 and verifying it and sending the Message 3 to supplicant and third model the receiving of Message 4 and installation of PTK.

In this authenticator model, first, Message 1 is generated and sent with the help of place A. Then at place B, Message 2 is received and authenticator checks the sequence number of this received message at the place seq_ok_for_msg2 and then generates Pairwise Transient Key (PTK) using places anonce, snonce and PMK and calculates the message Integrity code (MIC) for the received message 2 on the place Mic2. Verification of received MIC of message 2 is done at the transition verify_mic2.

After verification of MIC, third message has been generated and MIC of this 3rd message is also calculated and this message 3 is sent to the supplicant via place C. And after receiving and verifying the MIC of message 4 at place D, PTK is installed on the authenticator side.

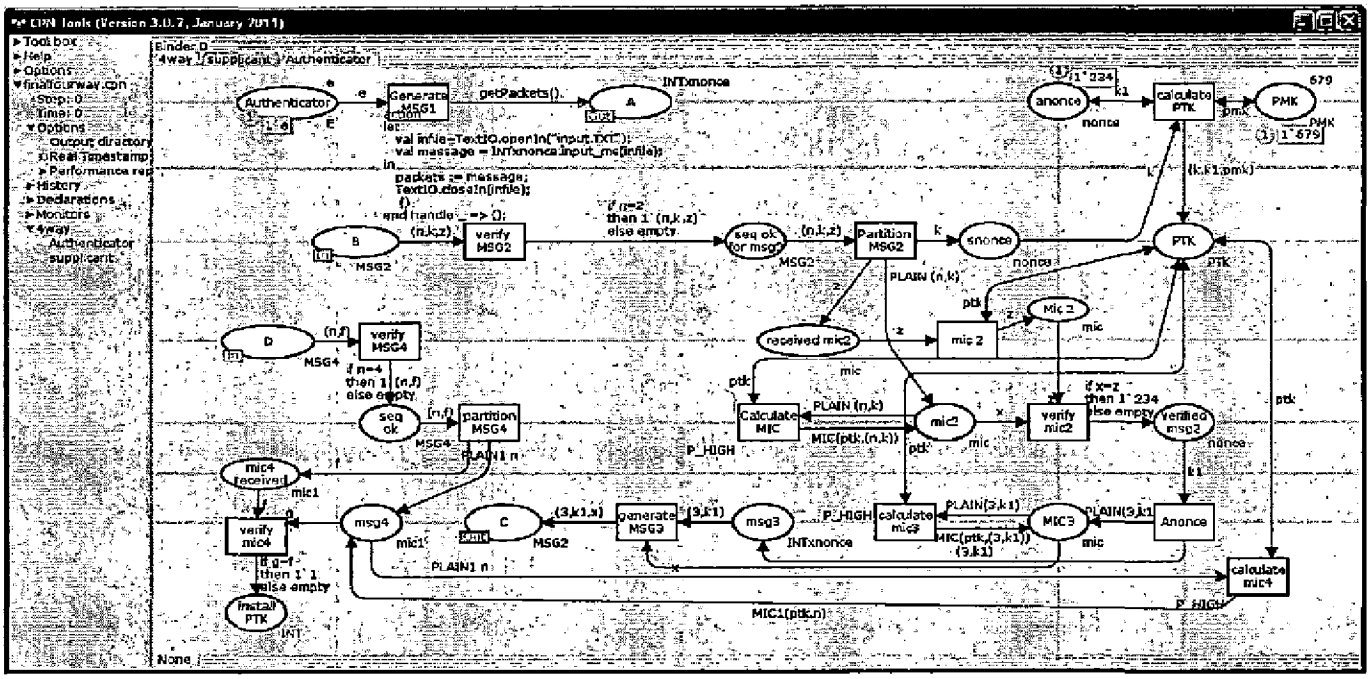


Figure 4.2 Authenticator model of 4-way handshake

4.2.3 Model of Supplicant

In Figure 4.3, CPN model of supplicant is shown. It contains two subnets: one model the receiving of Message 1 and generation and sending process of Message 2, other subnet models the receiving of Message 3. In the second subnet, Message 3 is verified and Message 4 is generated and sent to the authenticator.

Here the supplicant receives the Anonce in the form of first message with the help of place A and generates the snonce and calculate PTK. After this, message 2 is generated at the transition generate MSG2 using snonce and MIC where MIC is calculated with the help of Anonce and Snonce. This message 2 is sent to the authenticator via place B. Then a third message is received at place C and MIC for this third message is calculated verified. After verification, message 4 is generated which is simply an ACK of message 3, but it also contains MIC. This forth message is sent to the authenticator via place D and PTK is installed on the supplicant side.

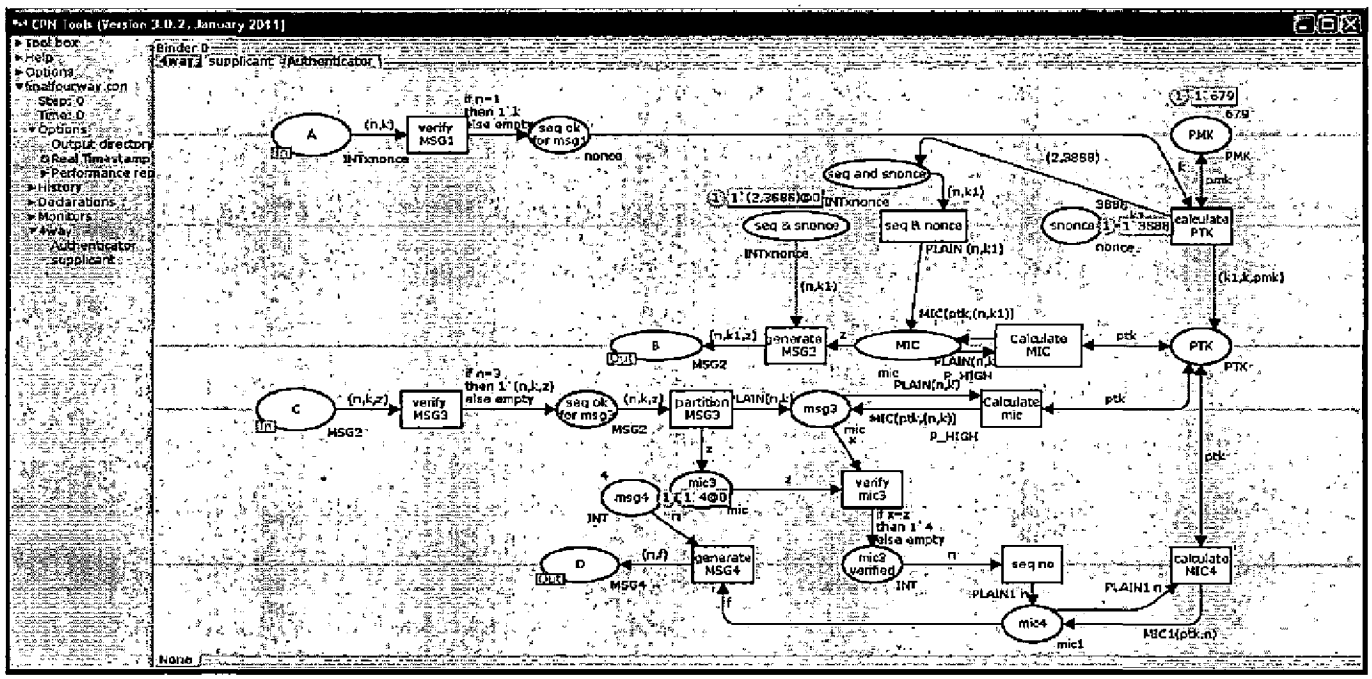


Figure 4.3 Supplicant model of 4-way handshake

4.3 4-Way Handshake Protocol with Intruder

In the Figure 4.4, the prime page of 4-way handshake protocol is shown with an intruder. Here model of intruder is also attached as a substitution transition. Authenticator and supplicant substitution transitions are same as previously described. Here place E represents the fake massage 1 which contains anonce'.

Figure 4.5 is showing the model of the intruder. Here the intruder is generating the fake first message of the 4-way handshake protocol and send it to the supplicant, where supplicant has received the original message 1 and generating the message 2. But, we can see that message 1 of 4-way handshake protocol is not having any type of security and an intruder can easily create this message 1 and sends it to the supplicant.

In supplicant side also, there is no mechanism of verifying this first message. Therefore, supplicant treats this fake message as authentic and generate Message 2, PTK and MIC using the ANonce' (ANonce which is calculated by the fake Message 1) and sends this message 2 to the authenticator which silently discards this message because of failure of MIC verification and 4-way handshake fails.

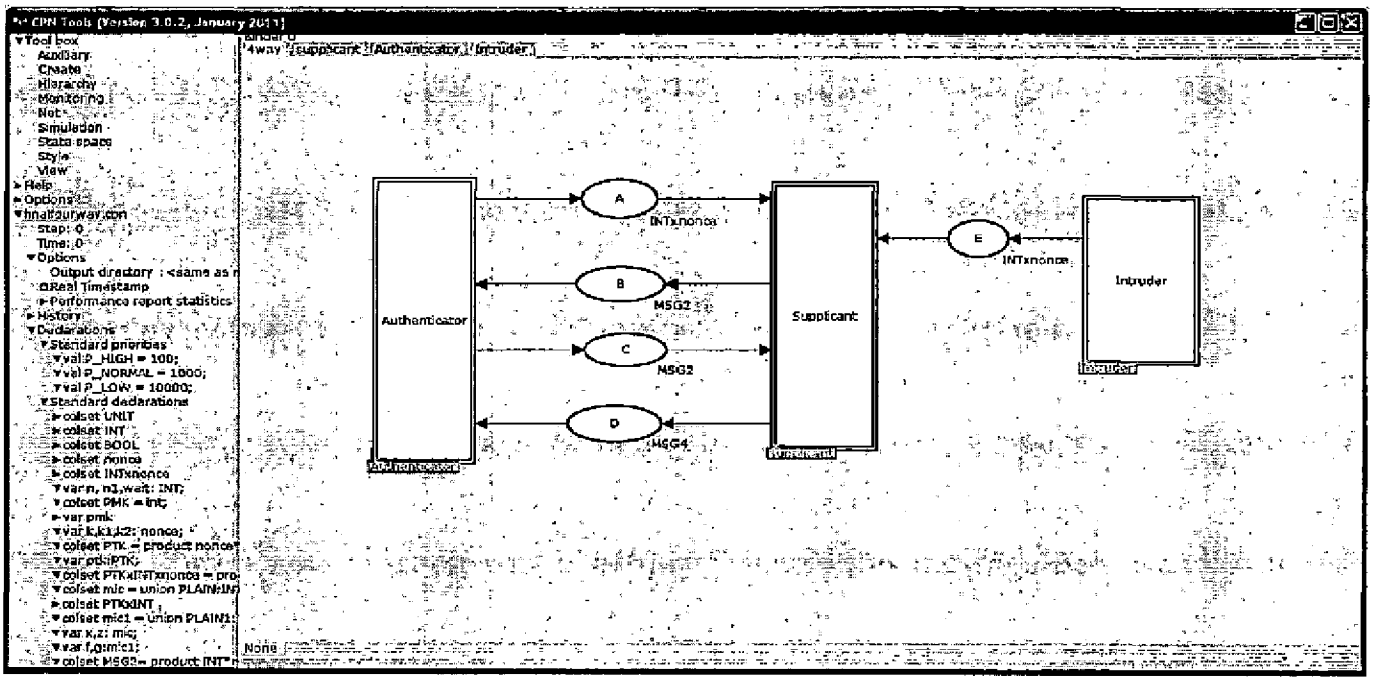


Figure 4.4 Prime Page of 4-way handshake with intruder

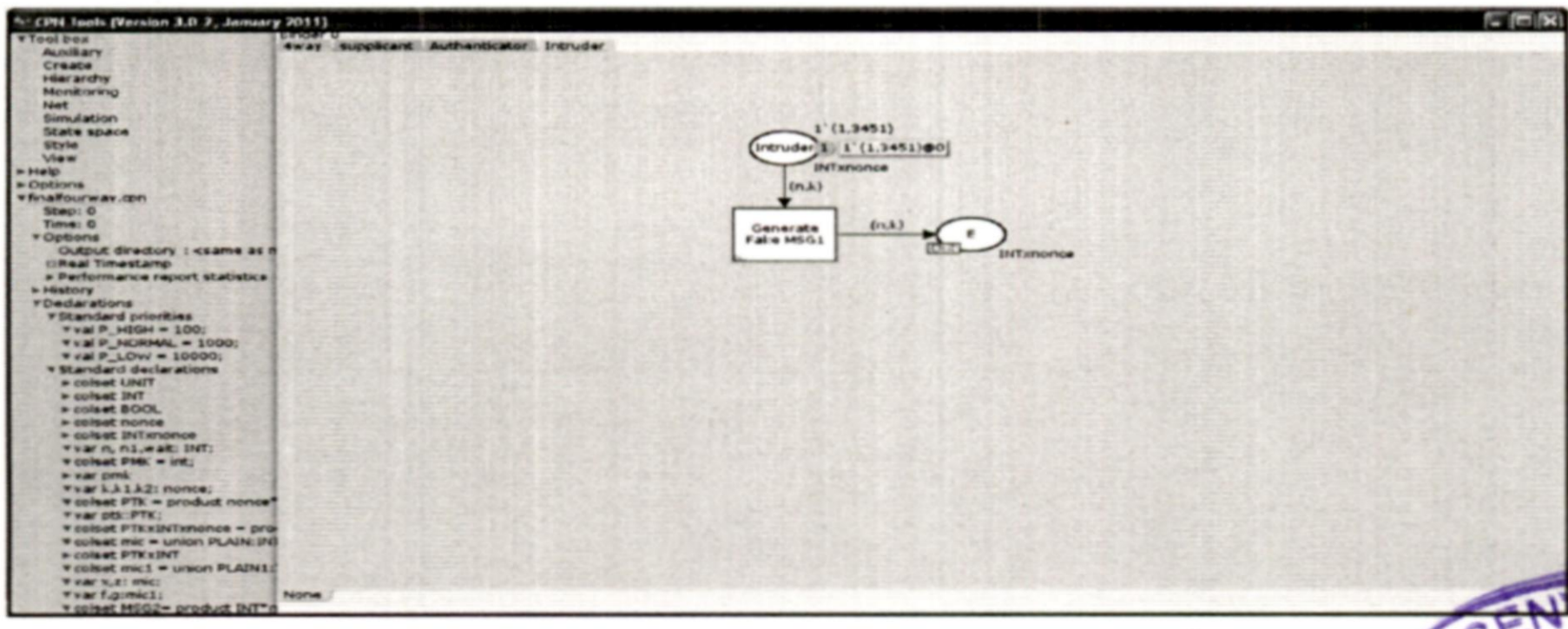


Figure 4.5 Intruder sub-page of 4-way handshake

4.4 Proposed 3-Way Handshake Protocol without intruder

The proposed enhanced 3-way handshake protocol is modeled here using CPN Tool. The proposed protocol provides the better security mechanism compared to the 4-way handshake protocol.

4.4.1 The Top Level Model

In the Figure 4.6, prime page of enhanced 3-way handshake protocol has been created using CPN Tool. Here, supplicant and authenticator are shown as substitution transitions and all the exchanged messages are represented by places A, B and C.

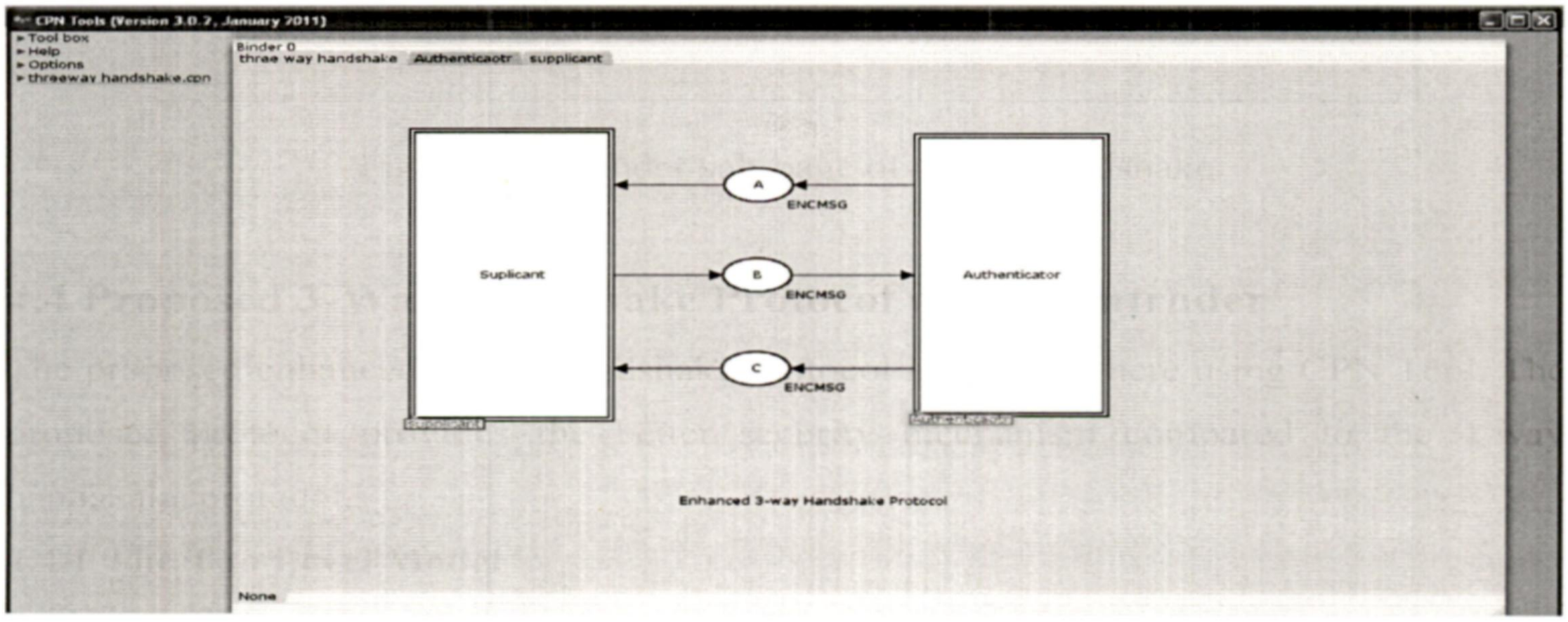


Figure 4.6 Prime page of enhanced 3-way handshake protocol CPN Model

4.4.2 Model of Authenticator

In the Figure 4.7, authenticator transition is shown. It consists of two subnets. The first subnet generates the Message 1 and sends it to the supplicant via place A. Second subnet receives the Message 2 at the place B. After decryption and verification, second subnet generates and sends the Message 3 to the supplicant via place C and installs PTK.

In this page, authenticator generates the anonce and put it with the supplicant identity (SID) which is exchanged already in the EAP authentication process, and encrypt these with the help of New encryption key (NEK). NEK is generated already with the help of Pairwise Master Key (PMK) and a second pre-shared key (SPK) and sends this Message 1 to the supplicant. Now the authenticator receives the Message 2 from supplicant and decrypts it with NEK and gets Anonce and Snonce.

This received Anonce is compared with the Anonce which is generated by authenticator and after verifying Anonce, Message 3 is generated which contains encrypted snonce and SID. This Message 3 is sent to supplicant and PTK is installed on the authenticator side.

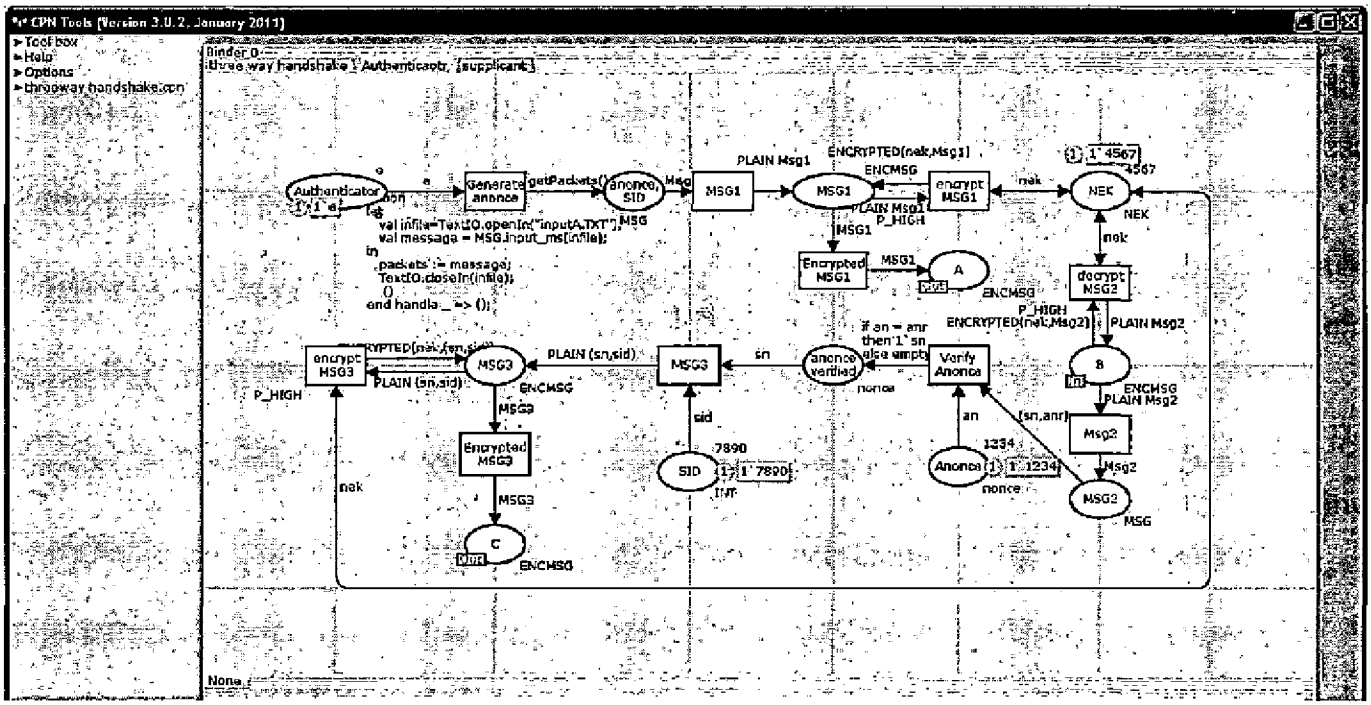


Figure 4.7 Authenticator model of enhanced 3-way handshake protocol

4.4.3 Model of Supplicant

The supplicant sub-page is shown in Figure 4.8. It contains two subnets. First one receives Message 1 at place A and after verification, generates and sends Message 2 via place B to the authenticator. Second subnet receives and verifies the Message 3 and installs PTK.

In the Figure 4.8, supplicant receives the Message 1 from the authenticator and decrypts it with the help of NEK and verifies the SID with its own identity at the transition verify SID. After verification, message 2 is generated at the transition Encrypted MSG2 and sent to the authenticator via place B. Message 2 consists of encrypted form of anonce and snonce. Now supplicant waits for the Message 3 and after receiving this message 3 at place C, supplicant decrypts and verifies the snonce at the transition verify snonce. After successful verification of snonce, PTK is installed at the place Install PTK on the supplicant side.

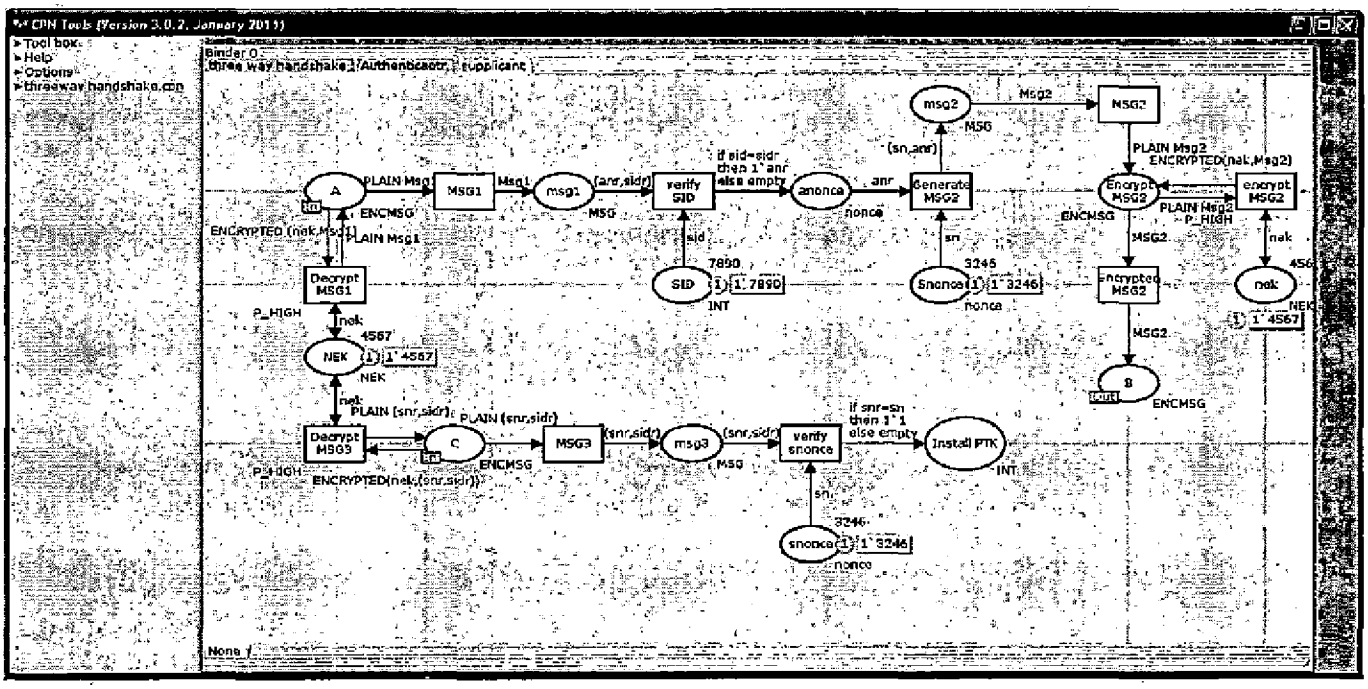


Figure 4.8 Supplicant model of enhanced 3-way handshake protocol

4.5 Enhanced 3-Way Handshake Protocol with intruder

In the proposed 3-way handshake protocol, if we introduce an intruder then also enhanced 3-way handshake protocol will work properly. For proving this, a simulation model has been made as shown in the figure 4.9. Here, an intruder is introduced as a substitution transition which tries to generate Message 1 of the proposed mechanism and sends it to supplicant. However, intruder does not know the NEK, so it will fail in the process of verification in the supplicant.

Here, substitution transition Supplicant and Authenticator are same as previously described in section 4.3. Only the intruder is additional here. Here place A, B and C represents the Message 1, 2 and 3 respectively.

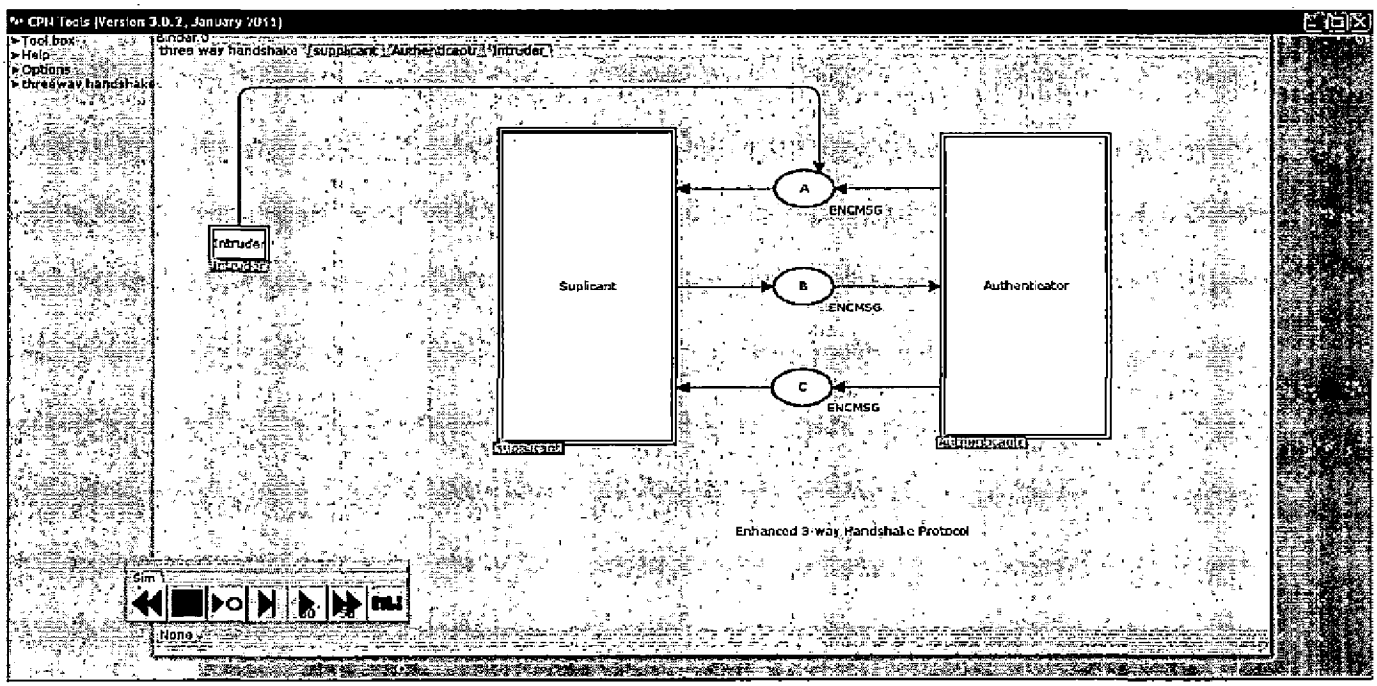


Figure 4.9 Enhanced 3-way handshake protocol CPN Model with intruder

In the Figure 4.10, an intruder is shown which consist of a single subnet. Here the intruder takes SID and ANonce' and encrypts it with some random key and sends it to the supplicant via place D. But in the supplicant side the encryption key is different. Therefore, verification process will fail and this message will be ignored by the supplicant and PTK will install successfully on both the sides.

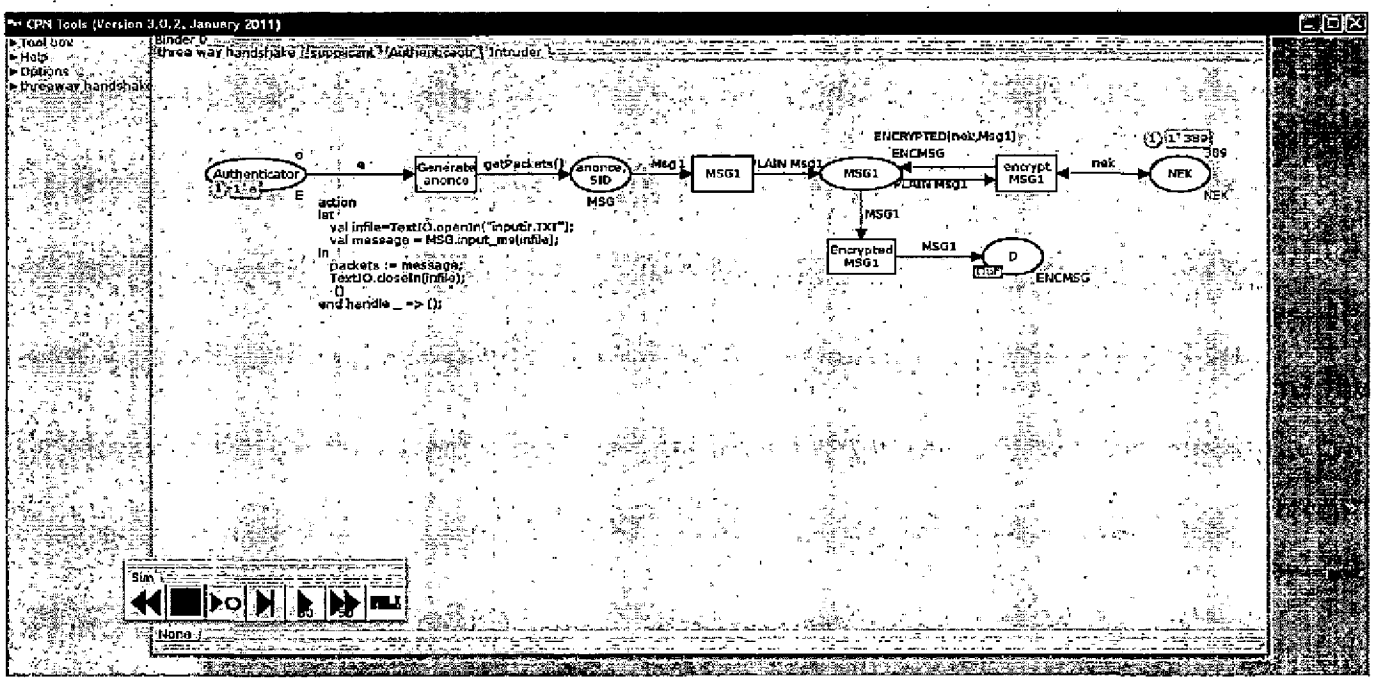


Figure 4.10 Intruder sub-page of 3-way handshake

Chapter 5

Results

5.1 Formal Verification Parameters

The standard formal verification parameters derived from the state space analysis report are:

- Liveness Property: It assumes that if the authenticator sends the first message a protocol, it will receive the last message definitely. In our case, it means the handshake executes successfully. If the execution reaches to the final state then liveness is satisfied otherwise not.
- Fairness: It determines whether the set of transition instances is impartial or fair.
- Deadlock: CPN can test whether the deadlock appears in the modeled system or not. Deadlock means that the protocol will unexpectedly terminate in the case of resource accessing conflict or unlimitedly waiting for acknowledge packets.
- Number of Nodes: The number of nodes is useful to observe the increased number of communicating nodes due to the intruder. With the introduction of intruder the number of nodes increases.
- Number of Arcs: Arcs connects the nodes. As the number of nodes increases the number of arcs also increases.

5.2 Formal Verification

After simulating and finding the flaws of 4-way handshake protocol, the enhanced 3-way handshake protocol has been simulated and state space analysis has been done. The following shows the analysis of the report generated for each scenario.

5.2.1 4-Way Handshake Protocol without intruder

The following is the CPN Tool's state space report for the 4-way handshake protocol without an intruder:

```
Statistics
------------------------------------------------------------------------
-
```

State Space
Nodes: 32
Arcs: 31
Secs: 0
Status: Full

Scc Graph
Nodes: 32
Arcs: 31
Secs: 0

Boundedness Properties

Best Integer Bounds

	Upper	Lower
Authenticator'Authenticator 1	1	0
Authenticator'MIC3 1	1	0
Authenticator'Mic_2 1	1	0
Authenticator'PMK 1	1	1
Authenticator'PTK 1	1	0
Authenticator'anonce 1	1	1
Authenticator'install_PTK 1	1	0
Authenticator'mic2 1	1	0
Authenticator'mic4_received 1	1	0
Authenticator'msg3 1	1	0
Authenticator'msg4 1	1	0
Authenticator'received_mic2 1	1	0
Authenticator'seq_ok 1	1	0
Authenticator'seq_ok_for_msg2 1	1	0
Authenticator'snonce 1	1	0
Authenticator'verified_msg2 1	1	0
fourway'A 1	1	0
fourway'B 1	1	0
fourway'C 1	1	0
fourway'D 1	1	0
supplicant'MIC 1	1	0
supplicant'PMK 1	1	1
supplicant'PTK 1	1	0
supplicant'mic3_received 1	1	0
supplicant'mic3_verified 1	1	0
supplicant'mic4 1	1	0
supplicant'msg3 1	1	0
supplicant'msg4 1	1	0
supplicant'seq 1	1	0
supplicant'seq_and_snonce 1	1	0

```
      supplicant'seq_ok_for_msg1 1
                                1           0
      supplicant'seq_ok_for_msg3 1
                                1           0
      supplicant'snonce 1       1           0

  Best Upper Multi-set Bounds
      Authenticator'Authenticator 1
                                1`e
      Authenticator'MIC3 1
                                1`PLAIN((3,234))++
1`MIC(((3888,234,679),(3,234)))
      Authenticator'Mic_2 1
                                1`MIC(((3888,234,679),(2,3888)))
      Authenticator'PMK 1 1`679
      Authenticator'PTK 1 1`(3888,234,679)
      Authenticator'anonce 1
                                1`234
      Authenticator'install_PTK 1
                                1`1
      Authenticator'mic2 1
                                1`PLAIN((2,3888))++
1`MIC(((3888,234,679),(2,3888)))
      Authenticator'mic4_received 1
                                1`PLAIN1(4)++
1`MIC1(((3888,234,679),4))
      Authenticator'msg3 1
                                1`(3,234)
      Authenticator'msg4 1
                                1`PLAIN1(4)++
1`MIC1(((3888,234,679),4))
      Authenticator'received_mic2 1
                                1`MIC(((3888,234,679),(2,3888)))
      Authenticator'seq_ok 1
                                1`(4,PLAIN1(4))++
1`(4,MIC1(((3888,234,679),4)))
      Authenticator'seq_ok_for_msg2 1
                                1`(2,3888,MIC(((3888,234,679),(2,3888))))
      Authenticator'snonce 1
                                1`3888
      Authenticator'verified_msg2 1
                                1`234
      fourway'A 1               1`(1,234)
      fourway'B 1               1`(2,3888,MIC(((3888,234,679),(2,3888))))
      fourway'C 1               1`(3,234,MIC(((3888,234,679),(3,234))))
      fourway'D 1               1`(4,PLAIN1(4))++
1`(4,MIC1(((3888,234,679),4)))
      supplicant'MIC 1          1`PLAIN((2,3888))++
1`MIC(((3888,234,679),(2,3888)))
      supplicant'PMK 1          1`679
      supplicant'PTK 1          1`(3888,234,679)
      supplicant'mic3_received 1
                                1`MIC(((3888,234,679),(3,234)))
      supplicant'mic3_verified 1
                                1`4
      supplicant'mic4 1         1`PLAIN1(4)++
1`MIC1(((3888,234,679),4))
```

```
     supplicant'msg3 1    1`PLAIN((3,234))++
1`MIC(((3888,234,679),(3,234)))
     supplicant'msg4 1    1`4
     supplicant'seq 1     1`(2,3888)
     supplicant'seq_and_snonce 1
                          1`(2,3888)
     supplicant'seq_ok_for_msg1 1
                          1`234
     supplicant'seq_ok_for_msg3 1
                          1`(3,234,MIC(((3888,234,679),(3,234))))
     supplicant'snonce 1 1`3888

  Best Lower Multi-set Bounds
     Authenticator'Authenticator 1
                          empty
     Authenticator'MIC3 1
                          empty
     Authenticator'Mic_2 1
                          empty
     Authenticator'PMK 1 1`679
     Authenticator'PTK 1 empty
     Authenticator'anonce 1
                          1`234
     Authenticator'install_PTK 1
                          empty
     Authenticator'mic2 1
                          empty
     Authenticator'mic4_received 1
                          empty
     Authenticator'msg3 1
                          empty
     Authenticator'msg4 1
                          empty
     Authenticator'received_mic2 1
                          empty
     Authenticator'seq_ok 1
                          empty
     Authenticator'seq_ok_for_msg2 1
                          empty
     Authenticator'snonce 1
                          empty
     Authenticator'verified_msg2 1
                          empty
     fourway'A 1          empty
     fourway'B 1          empty
     fourway'C 1          empty
     fourway'D 1          empty
     supplicant'MIC 1     empty
     supplicant'PMK 1     1`679
     supplicant'PTK 1     empty
     supplicant'mic3_received 1
                          empty
     supplicant'mic3_verified 1
                          empty
     supplicant'mic4 1    empty
     supplicant'msg3 1    empty
     supplicant'msg4 1    empty
```

```
     supplicant'seq 1     empty
     supplicant'seq_and_snonce 1
                          empty
     supplicant'seq_ok_for_msg1 1
                          empty
     supplicant'seq_ok_for_msg3 1
                          empty
     supplicant'snonce 1 empty


 Home Properties
-------------------------------------------------------------------------
-

  Home Markings
     Initial Marking is not a home marking


 Liveness Properties
-------------------------------------------------------------------------
-

  Dead Markings
     [31,32]

  Dead Transition Instances
     None

  Live Transition Instances
     None


 Fairness Properties
-------------------------------------------------------------------------
-
     No infinite occurrence sequences.
```

In the statistics, the state space shows that the number of nodes and arcs are 32 and 31 respectively. The secs variable shows that the time taken for the execution is zero. The status is full which shows that all the nodes in the state space are fully processed. Here the boundedness property is showing the maximum and minimum number of tokens held by each node.

In Liveness property, the dead marking is [31, 32] which means that the nodes [31, 32] has no further enabled binding elements i.e. the marking of the specified nodes are dead. There are no dead transition instances because every node occurred starting from the initial marking of the state space. In fairness property, it tells us that we do not have an infinite occurrence sequence unless each transition on the page continues to occur.

Thus the standard 4-Way handshake protocol satisfies all the desired properties of fairness, liveness and deadlock-free.

5.2.2 4-Way Handshake Protocol with intruder

The following is the CPN Tool's state space report for the 4-way handshake protocol with an intruder:

```
 Statistics
------------------------------------------------------------------------
-

  State Space
     Nodes:  126
     Arcs:   191
     Secs:   1
     Status: Full

  Scc Graph
     Nodes:  126
     Arcs:   191
     Secs:   0

 Boundedness Properties
------------------------------------------------------------------------
-

  Best Integer Bounds
                              Upper       Lower
     Authenticator'Authenticator 1
                              1           0
     Authenticator'MIC3 1     1           0
     Authenticator'Mic_2 1    1           0
     Authenticator'PMK 1      1           1
     Authenticator'PTK 1      1           0
     Authenticator'anonce 1   1           1
     Authenticator'install_PTK 1
                              1           0
     Authenticator'mic2 1     1           0
     Authenticator'mic4_received 1
                              1           0
     Authenticator'msg3 1     1           0
     Authenticator'msg4 1     1           0
     Authenticator'received_mic2 1
                              1           0
     Authenticator'seq_ok 1   1           0
     Authenticator'seq_ok_for_msg2 1
                              1           0
     Authenticator'snonce 1   1           0
     Authenticator'verified_msg2 1
                              1           0
     Intruder'Intruder 1      1           0
     fourway'A 1              1           0
```

```
      fourway'B 1              1          0
      fourway'C 1              1          0
      fourway'D 1              1          0
      fourway'E 1              1          0
      supplicant'MIC 1         1          0
      supplicant'PMK 1         1          1
      supplicant'PTK 1         1          0
      supplicant'mic3_received 1
                               1          0
      supplicant'mic3_verified 1
                               1          0
      supplicant'mic4 1        1          0
      supplicant'msg3 1        1          0
      supplicant'msg4 1        1          0
      supplicant'seq 1         1          0
      supplicant'seq_and_snonce 1
                               1          0
      supplicant'seq_ok_for_msg1 1
                               2          0
      supplicant'seq_ok_for_msg3 1
                               1          0
      supplicant'snonce 1      1          0

  Best Upper Multi-set Bounds
     Authenticator'Authenticator 1
                              1`e
     Authenticator'MIC3 1
                              1`PLAIN((3,234))++
1`MIC(((3888,234,679),(3,234)))
     Authenticator'Mic_2 1
                              1`MIC(((3888,234,679),(2,3888)))++
1`MIC(((3888,3451,679),(2,3888)))
     Authenticator'PMK 1 1`679
     Authenticator'PTK 1 1`(3888,234,679)
     Authenticator'anonce 1
                              1`234
     Authenticator'install_PTK 1
                              1`1
     Authenticator'mic2 1
                              1`PLAIN((2,3888))++
1`MIC(((3888,234,679),(2,3888)))
     Authenticator'mic4_received 1
                              1`PLAIN1(4)++
1`MIC1(((3888,234,679),4))
     Authenticator'msg3 1
                              1`(3,234)
     Authenticator'msg4 1
                              1`PLAIN1(4)++
1`MIC1(((3888,234,679),4))
     Authenticator'received_mic2 1
                              1`MIC(((3888,234,679),(2,3888)))++
1`MIC(((3888,3451,679),(2,3888)))
     Authenticator'seq_ok 1
                              1`(4,PLAIN1(4))++
1`(4,MIC1(((3888,234,679),4)))
     Authenticator'seq_ok_for_msg2 1
                              1`(2,3888,MIC(((3888,234,679),(2,3888))))++
```

```
1`(2,3888,MIC(((3888,3451,679),(2,3888))))
     Authenticator'snonce 1
                          1`3888
     Authenticator'verified_msg2 1
                          1`234
     Intruder'Intruder 1 1`(1,3451)
     fourway'A 1          1`(1,234)
     fourway'B 1          1`(2,3888,MIC(((3888,234,679),(2,3888))))++
1`(2,3888,MIC(((3888,3451,679),(2,3888))))
     fourway'C 1          1`(3,234,MIC(((3888,234,679),(3,234))))
     fourway'D 1          1`(4,PLAIN1(4))++
1`(4,MIC1(((3888,234,679),4)))
     fourway'E 1          1`(1,3451)
     supplicant'MIC 1     1`PLAIN((2,3888))++
1`MIC(((3888,234,679),(2,3888)))++
1`MIC(((3888,3451,679),(2,3888)))
     supplicant'PMK 1     1`679
     supplicant'PTK 1     1`(3888,234,679)++
1`(3888,3451,679)
     supplicant'mic3_received 1
                          1`MIC(((3888,234,679),(3,234)))
     supplicant'mic3_verified 1
                          1`4
     supplicant'mic4 1    1`PLAIN1(4)++
1`MIC1(((3888,234,679),4))
     supplicant'msg3 1    1`PLAIN((3,234))++
1`MIC(((3888,234,679),(3,234)))
     supplicant'msg4 1    1`4
     supplicant'seq 1     1`(2,3888)
     supplicant'seq_and_snonce 1
                          1`(2,3888)
     supplicant'seq_ok_for_msg1 1
                          1`234++
1`3451
     supplicant'seq_ok_for_msg3 1
                          1`(3,234,MIC(((3888,234,679),(3,234))))
     supplicant'snonce 1 1`3888

  Best Lower Multi-set Bounds
     Authenticator'Authenticator 1
                          empty
     Authenticator'MIC3 1
                          empty
     Authenticator'Mic_2 1
                          empty
     Authenticator'PMK 1 1`679
     Authenticator'PTK 1 empty
     Authenticator'anonce 1
                          1`234
     Authenticator'install_PTK 1
                          empty
     Authenticator'mic2 1
                          empty
     Authenticator'mic4_received 1
                          empty
     Authenticator'msg3 1
                          empty
```

```
Authenticator'msg4 1
                        empty
Authenticator'received_mic2 1
                        empty
Authenticator'seq_ok 1
                        empty
Authenticator'seq_ok_for_msg2 1
                        empty
Authenticator'snonce 1
                        empty
Authenticator'verified_msg2 1
                        empty
Intruder'Intruder 1 empty
fourway'A 1         empty
fourway'B 1         empty
fourway'C 1         empty
fourway'D 1         empty
fourway'E 1         empty
supplicant'MIC 1    empty
supplicant'PMK 1    1`679
supplicant'PTK 1    empty
supplicant'mic3_received 1
                        empty
supplicant'mic3_verified 1
                        empty
supplicant'mic4 1   empty
supplicant'msg3 1   empty
supplicant'msg4 1   empty
supplicant'seq 1    empty
supplicant'seq_and_snonce 1
                        empty
supplicant'seq_ok_for_msg1 1
                        empty
supplicant'seq_ok_for_msg3 1
                        empty
supplicant'snonce 1 empty
```

Home Properties

Home Markings

Initial Marking is not a home marking

Liveness Properties

Dead Markings

[71,125,126]

Dead Transition Instances

None

Live Transition Instances

```
    None

 Fairness Properties
------------------------------------------------------------------------
-
    No infinite occurrence sequences.
```

In the statistics, the state space shows that the number of nodes and arcs are 126 and 191 respectively. The secs variable shows that the time taken for the execution is 0. The status is full which shows that all the nodes in the state space are fully processed. Here the boundedness property is showing the maximum and minimum number of tokens held by each node.

In Liveness property, the dead marking is [71, 125, 126] which means that the node [71, 125, 126] has no further enabled binding elements i.e. the marking of the specified node is dead. There are no dead transition instances because every node occurred starting from the initial marking of the state space. In fairness property, it tells us that we do not have any infinite occurrence sequence unless each transition on the page continues to occur.

In the case when the intruder is sending only one fake Message 1, then we can see that number of nodes and arcs have been increased drastically. And when we increase the number of fake Message 1s then number and arcs also increase. Like in case of two fake Message 1s, the number of nodes and arcs are 468 and 963 respectively. And in the case of further increase in the number of fake Message 1, state space tool has failed in the generation of the report which indicates towards the DoS attack in the form of state space analysis.

5.2.3 3-Way Handshake Protocol without intruder

The following is the CPN Tool's state space report for the proposed 3-way handshake protocol without an intruder:

```
Statistics
------------------------------------------------------------------------
-

  State Space
     Nodes:  21
     Arcs:   20
     Secs:   0
     Status: Full
```

```
Scc Graph
    Nodes:  21
    Arcs:   20
    Secs:   0


Boundedness Properties
------------------------------------------------------------------------
-


Best Integer Bounds
                                 Upper       Lower
    Authenticaotr'Anonce 1       1           0
    Authenticaotr'Authenticator 1
                                 1           0
    Authenticaotr'MSG1 1         1           0
    Authenticaotr'MSG2 1         1           0
    Authenticaotr'MSG3 1         1           0
    Authenticaotr'NEK 1          1           1
    Authenticaotr'SID 1          1           0
    Authenticaotr'anonce 1       1           0
    Authenticaotr'anonce_verified 1
                                 1           0
    supplicant'Encrypt_MSG2 1
                                 1           0
    supplicant'Install_PTK 1
                                 1           0
    supplicant'NEK 1             1           1
    supplicant'SID 1             1           0
    supplicant'Snonce 1          1           0
    supplicant'anonce 1          1           0
    supplicant'msg1 1            1           0
    supplicant'msg2 1            1           0
    supplicant'msg3 1            1           0
    supplicant'nek 1             1           1
    supplicant'snonce 1          1           0
    three_way_handshake'A 1 1    1           0
    three_way_handshake'B 1 1    1           0
    three_way_handshake'C 1 1    1           0

Best Upper Multi-set Bounds
    Authenticaotr'Anonce 1
                          1`1234
    Authenticaotr'Authenticator 1
                          1`e
    Authenticaotr'MSG1 1
                          1`PLAIN((1234,7890))++
1`ENCRYPTED((4567,(1234,7890)))
    Authenticaotr'MSG2 1
                          1`(3246,1234)
    Authenticaotr'MSG3 1
                          1`PLAIN((3246,7890))++
1`ENCRYPTED((4567,(3246,7890)))
    Authenticaotr'NEK 1 1`4567
    Authenticaotr'SID 1 1`7890
    Authenticaotr'anonce 1
                          1`(1234,7890)
```

```
     Authenticaotr'anonce_verified 1
                          1`3246
     supplicant'Encrypt_MSG2 1
                          1`PLAIN((3246,1234))++
1`ENCRYPTED((4567,(3246,1234)))
     supplicant'Install_PTK 1
                          1`1
     supplicant'NEK 1     1`4567
     supplicant'SID 1     1`7890
     supplicant'Snonce 1  1`3246
     supplicant'anonce 1  1`1234
     supplicant'msg1 1    1`(1234,7890)
     supplicant'msg2 1    1`(3246,1234)
     supplicant'msg3 1    1`(3246,7890)
     supplicant'nek 1     1`4567
     supplicant'snonce 1  1`3246
     three_way_handshake'A 1
                          1`PLAIN((1234,7890))++
1`ENCRYPTED((4567,(1234,7890)))
     three_way_handshake'B 1
                          1`PLAIN((3246,1234))++
1`ENCRYPTED((4567,(3246,1234)))
     three_way_handshake'C 1
                          1`PLAIN((3246,7890))++
1`ENCRYPTED((4567,(3246,7890)))

  Best Lower Multi-set Bounds
     Authenticaotr'Anonce 1
                          empty
     Authenticaotr'Authenticator 1
                          empty
     Authenticaotr'MSG1 1
                          empty
     Authenticaotr'MSG2 1
                          empty
     Authenticaotr'MSG3 1
                          empty
     Authenticaotr'NEK 1 1`4567
     Authenticaotr'SID 1 empty
     Authenticaotr'anonce 1
                          empty
     Authenticaotr'anonce_verified 1
                          empty
     supplicant'Encrypt_MSG2 1
                          empty
     supplicant'Install_PTK 1
                          empty
     supplicant'NEK 1     1`4567
     supplicant'SID 1     empty
     supplicant'Snonce 1  empty
     supplicant'anonce 1  empty
     supplicant'msg1 1    empty
     supplicant'msg2 1    empty
     supplicant'msg3 1    empty
     supplicant'nek 1     1`4567
     supplicant'snonce 1  empty
     three_way_handshake'A 1
```

```
                          empty
    three_way_handshake'B 1
                          empty
    three_way_handshake'C 1
                          empty


 Home Properties
-------------------------------------------------------------------------
-

  Home Markings
     Initial Marking is not a home marking


 Liveness Properties
-------------------------------------------------------------------------
-

  Dead Markings
     [21]

  Dead Transition Instances
     None

  Live Transition Instances
     None



Fairness Properties
-------------------------------------------------------------------------
-
     No infinite occurrence sequences.
```

In the statistics, the state space shows that the number of nodes and arcs are 21 and 20 respectively. The secs variable shows that the time taken for the execution is zero. The status is full which shows that all the nodes in the state space are fully processed. Here the boundedness property is showing the maximum and minimum number of tokens held by each node.

In Liveness property, the dead marking is 21 which means that the node 21 has no further enabled binding elements i.e. the marking of the specified node is dead. There are no dead transition instances because every node occurred starting from the initial marking of the state space. In fairness property, it tells us that we do not have any infinite occurrence sequence unless each transition on the page continues to occur.

Thus the proposed 3-way handshake protocol without intruder satisfies all the desired properties of fairness, liveness and deadlock-free.

5.2.4 3-Way Handshake Protocol with intruder

The following is the CPN Tool's state space report for the proposed 3-way handshake protocol with an intruder:

```
 Statistics
------------------------------------------------------------------------
-

  State Space
     Nodes:  95
     Arcs:   156
     Secs:   0
     Status: Full

  Scc Graph
     Nodes:  95
     Arcs:   156
     Secs:   0

 Boundedness Properties
------------------------------------------------------------------------
-

  Best Integer Bounds
                                Upper       Lower
     Authenticaotr'Anonce 1     1           0
     Authenticaotr'Authenticator 1
                                1           0
     Authenticaotr'MSG1 1       1           0
     Authenticaotr'MSG2 1       1           0
     Authenticaotr'MSG3 1       1           0
     Authenticaotr'NEK 1        1           1
     Authenticaotr'SID 1        1           0
     Authenticaotr'anonce 1     1           0
     Authenticaotr'anonce_verified 1
                                1           0
     Intruder'Authenticator 1
                                1           0
     Intruder'MSG1 1            1           0
     Intruder'NEK 1             1           1
     Intruder'anonce 1          1           0
     supplicant'Encrypt_MSG2 1
                                1           0
     supplicant'Install_PTK 1
                                1           0
     supplicant'NEK 1           1           1
     supplicant'SID 1           1           1
     supplicant'Snonce 1        1           0
     supplicant'anonce 1        1           0
```

```
      supplicant'msg1 1        1          0
      supplicant'msg2 1        1          0
      supplicant'msg3 1        1          0
      supplicant'nek 1         0          0
      supplicant'snonce 1      1          0
      three_way_handshake'A 1 2           0
      three_way_handshake'B 1 1           0
      three_way_handshake'C 1 1           0

  Best Upper Multi-set Bounds
      Authenticaotr'Anonce 1
                           1`1234
      Authenticaotr'Authenticator 1
                           1`e
      Authenticaotr'MSG1 1
                           1`PLAIN((1234,7890))++
1`ENCRYPTED((4567,(1234,7890)))
      Authenticaotr'MSG2 1
                           1`(3246,1234)
      Authenticaotr'MSG3 1
                           1`PLAIN((3246,7890))++
1`ENCRYPTED((4567,(3246,7890)))
      Authenticaotr'NEK 1 1`4567
      Authenticaotr'SID 1 1`7890
      Authenticaotr'anonce 1
                           1`(1234,7890)
      Authenticaotr'anonce_verified 1
                           1`3246
      Intruder'Authenticator 1
                           1`e
      Intruder'MSG1 1      1`PLAIN((3467,9090))++
1`ENCRYPTED((389,(3467,9090)))
      Intruder'NEK 1       1`389
      Intruder'anonce 1    1`(3467,9090)
      supplicant'Encrypt_MSG2 1
                           1`PLAIN((3246,1234))
      supplicant'Install_PTK 1
                           1`1
      supplicant'NEK 1     1`4567
      supplicant'SID 1     1`7890
      supplicant'Snonce 1 1`3246
      supplicant'anonce 1 1`1234
      supplicant'msg1 1    1`(1234,7890)
      supplicant'msg2 1    1`(3246,1234)
      supplicant'msg3 1    1`(3246,7890)
      supplicant'nek 1     empty
      supplicant'snonce 1 1`3246
      three_way_handshake'A 1
                           1`PLAIN((1234,7890))++
1`ENCRYPTED((389,(3467,9090)))++
1`ENCRYPTED((4567,(1234,7890)))
      three_way_handshake'B 1
                           1`PLAIN((3246,1234))
      three_way_handshake'C 1
                           1`PLAIN((3246,7890))++
1`ENCRYPTED((4567,(3246,7890)))
```

```
Best Lower Multi-set Bounds
   Authenticaotr'Anonce 1
                        empty
   Authenticaotr'Authenticator 1
                        empty
   Authenticaotr'MSG1 1
                        empty
   Authenticaotr'MSG2 1
                        empty
   Authenticaotr'MSG3 1
                        empty
   Authenticaotr'NEK 1 1`4567
   Authenticaotr'SID 1 empty
   Authenticaotr'anonce 1
                        empty
   Authenticaotr'anonce_verified 1
                        empty
   Intruder'Authenticator 1
                        empty
   Intruder'MSG1 1      empty
   Intruder'NEK 1       1`389
   Intruder'anonce 1    empty
   supplicant'Encrypt_MSG2 1
                        empty
   supplicant'Install_PTK 1
                        empty
   supplicant'NEK 1     1`4567
   supplicant'SID 1     1`7890
   supplicant'Snonce 1 empty
   supplicant'anonce 1 empty
   supplicant'msg1 1    empty
   supplicant'msg2 1    empty
   supplicant'msg3 1    empty
   supplicant'nek 1     empty
   supplicant'snonce 1 empty
   three_way_handshake'A 1
                        empty
   three_way_handshake'B 1
                        empty
   three_way_handshake'C 1
                        empty


Home Properties
------------------------------------------------------------------------
-

  Home Markings
     Initial Marking is not a home marking


Liveness Properties
------------------------------------------------------------------------
-

  Dead Markings
     [95]
```

```
Dead Transition Instances
   Authenticaotr'decrypt_MSG2 1
   supplicant'encrypt_MSG2 1

Live Transition Instances
   None

Fairness Properties
------------------------------------------------------------------------
-
   No infinite occurrence sequences.
```

In the statistics, the state space shows that the number of nodes and arcs are 95 and 156 respectively. Here the increase in the number of nodes and arcs is seen, because the nodes and arcs of the intruder are also included. The secs variable shows that the time taken for the execution is zero. The status is full which shows that all the nodes in the state space are fully processed. Here the boundedness property is showing the maximum and minimum number of tokens held by each node.

In Liveness property, the dead marking is 95 which means that the node 95 has no further enabled binding elements i.e. the marking of the specified node is dead. There are no dead transition instances because every node occurred starting from the initial marking of the state space. In fairness property, it tells us that we do not have any infinite occurrence sequence unless each transition on the page continues to occur.

Thus the proposed 3-way handshake protocol with an intruder also satisfies all the desired properties of fairness, liveness and deadlock-free.

5.3 Comparative Analysis

In the case of the 4-way handshake protocol, there is no deadlock and the fairness and liveness properties are satisfied without intruder. But when an intruder is introduced and it sends the fake Message 1 regularly, then state space analysis has gone in infinite loop and deadlock occurs. And in case of 3-way handshake protocol, intruder does not affect the protocol and the proposed protocol completes successfully and the fake messages are ignored by the supplicant.

The report generated for the four models showing the different desired properties can be consolidated into a single table as shown in Table 5.1.

Table 5.1: Analysis of State Space

Approaches	**Fairness**	**Deadlock**	**Liveness**	**No. of nodes**	**No. of arcs**
3-Way Handshake w/o intruder	yes	no	yes	21	20
3-Way Handshake with intruder	yes	no	yes	95	156
4-Way Handshake w/o intruder	yes	no	yes	32	31
4-Way Handshake with intruder	yes	yes	no	126	191

From analyzing this report, we can say that fairness properties, liveness properties are satisfied for the proposed 3-way handshake protocol with and without intruder. But in case of 4-way handshake protocol, number of nodes and arcs increases very rapidly when an intruder is introduced, which makes it vulnerable and insecure.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

A secure and efficient key exchange mechanism 3-way handshake is proposed here, it can provide security against DoS and Dictionary attacks unlike conventional key exchange protocol 4-way handshake which has been proved insecure against DoS and Dictionary attacks because of its insecure Message 1. In our proposed mechanism, encryption of all the messages by NEK is used. The encryption makes it stronger against DoS attacks and removes the possibility of fake messages. And PTK is also generated by PMK and SPK which makes the guessing of key difficult and can provide the security against dictionary attack. Besides these attacks, various attacks like bogus authenticator or supplicant and passive listening are also removed by using proposed key exchange mechanism.

The proposed key exchange protocol is modelled and tested on CPN tool. The state space analysis report shows that the proposed protocol satisfies the desired properties of liveness, fairness and deadlock-free. In aspect of communication time, this technique is using only three message exchanges, so communication overhead is reduced by 25%. In aspect of computation time, it avoids MIC calculation and verification which is very time consuming, while symmetric encryption and decryption in the proposed technique are costless. Hence, proposed 3-way handshake protocol is more secure and robust in comparison with 4-way handshake protocol.

6.2 Future Work

In the future the improvements can be done in the following areas:

- Real time implementation of the proposed 3-way handshake protocol can be done.
- Effect of proposed 3-way handshake protocol on the mobile supplicant and handover mechanism can be studied.
- The proposed mechanism can be extended to other wireless networks like WIMAX for key exchange.

REFERENCES

[1] G. Hiertz, D. Denteneer, L. Stibor, Y. Zang, X. P. Costa, and B. Walke, "The IEEE 802.11 universe," *IEEE Communications Magazine,* vol. 48, pp. 62-70, 2010.

[2] S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the key scheduling algorithm of RC4," *Selected Areas in Cryptography, Springer,* vol. 2259, pp. 1-24, 2001.

[3] W. A. Arbaugh, N. Shankar, Y. C. J. Wan, and K. Zhang, "Your 802.11 wireless network has no clothes," *IEEE Wireless Communications,* vol. 9, pp. 44-51, 2002.

[4] N. Cam-Winget, R. Housley, D. Wagner, and J. Walker, "Security flaws in 802.11 data link protocols," *Communications of the ACM,* vol. 46, pp. 35-39, 2003.

[5] J. Walker, "Unsafe at any key size; an analysis of the WEP encapsulation," *IEEE document,* vol. 802, pp. 00-362, 2000.

[6] M. E. Manley, C. A. McEntee, A. M. Molet, and J. S. Park, "Wireless security policy development for sensitive organizations," *IEEE Information assurance and security,* vol. 25, pp. 150-157, 2005.

[7] A.Bittau, M. Handley, and J. Lackey, "The final nail in WEP's coffin," *In Proceedings of the 2006 IEEE Symposium on Security and Privacy,* pp. 515-525, 2006.

[8] J. C. Chen, M. C. Jiang, and Y. W. Liu, "Wireless LAN security & IEEE 802.11i," *IEEE Wireless Communications,* vol. 12, pp. 27-36, 2005.

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

[10] R. Moskowitz, "Weakness in passphrase choice in WPA interface," *Online available at: http://www.wifinetnews.com/archive/002452.html,* 2003.

[11] J. Bellardo, and S. Savage. "802.11 Denial-of-Service attacks: real vulnerabilities and practical solutions," In *Proceedings of the USENIX Security Symposium,* pp.15-28, 2003.

[12] Khan, M.A., Hasan, A., "Pseudo Random Number Based authentication to counter denial of service attacks on 802.11," In the proceedings of *5th IFIP International Conference on Wireless and Optical Communications Networks, 2008. WOCN '08.*, Surabaya, pp.1-5, 5-7 May 2008.

[13] D. Chen, J. Deng, and P. K. Varshney. "Protecting wireless networks against a Denial of Service attack based on virtual jamming," In *Poster Session of MobiCom2003*, San Diego, CA, pp. 124-128, 2003.

[14] P. Ding, J. Holliday, and A. Celik. "Improving the security of Wireless LANs by managing 802.1X Disassociation," In *Proceedings of the IEEE Consumer Communications & Networking Conference,* Las Vegas, NV, pp. 198-210, 2004.

[15] Chia-Chen Hung, Eric Hsiaokuang Wu, Cheng-Lin Wu, Ruei-Liang Gau, Yi-Cyuan Chen, "A Multi-Key Encryption Scheme for the Next Generation Wireless Network," *Journal of Computers*, vol.18, No.4, pp. 49-67, 2008.

[16] Aircrack-ng tool, in http://www.aircrack-ng.org/

[17] C. He, C. Mitchell, "Analysis of the 802.11i 4-way Handshake," in *Proceedings of the ACM Workshop on Wireless Security (WiSe'04), Philadelphia,* PA, USA, pp. 43–50,2004.

[18] Jing Liu1, Xinming Ye, Jun Zhang, Jun Li, "Security Verification of 802.11i 4-way Handshake Protocol," *ICC proceedings*, pp. 1642-1647, 2008.

[19] Dinesh Yadav and Anjali Sardana. "Authentication Process in IEEE 802.11: Current Issues and Challenges," 4th *International Conference on Network Security & Applications (CNSA, 2011)*, Chennai, India, 15-17 July-2011. Publisher: LNCS Springer. (Accepted and Registered)

[20] W. Mao, Modern Cryptography: Theory and Practice. Pearson Education, Prentice Hall PTR, 2004.

[21] H. Changhua and J. C. Mitchell, "Security Analysis and Improvements for IEEE 802.11i," in *12th Annual Network and Distributed System Security Symposium,* pp. 90-110, 2005.

[22] Dinesh Yadav and Anjali Sardana. "Enhanced 3-Way Handshake Protocol for Key Exchange in IEEE 802.11i," *3rd IEEE International Conference on*

Electronics Computer Technology (ICECT 2011), vol. 6, pp.132-135, Kanyakumari, India, 8-10 April, 2011.

[23] T.C. Clancy, "Secure handover in enterprise WLANs : CAPWAP, HOKEY, AND IEEE 802.11r", IEEE Wireless Communications, vol-15, pp. 80–85, 2008.

[24] CPN Group at the University of Aarhus. "Design/CPN Online, 2004", Available at http://www.daimi.au.dk/designCPN/.

[25] B. Nieh and S. Tavares. "Modelling and analyzing cryptographic protocols using Petri nets," In *Advances in Cryptology-ASIACRYPT*, vol. 718(LNCS), Springer, pp. 275–295, 1992.

[26] Al-Azzoni, D. G. Down, and R. Khedri, "Modeling and verification of cryptographic protocols using coloured petri nets and design/CPN," *MOMPES'05,* pp. 1-28, 2005

[27] Dresp, Wiebke, "Security Analysis of the Secure Authentication Protocol by means of Coloured Petri Nets," *In proceedings of Communications and Multimedia Security (LNCS),Springer, Berlin*, pp. 230-239, 2005.

LIST OF PUBLICATIONS

[1] **Dinesh Yadav** and Anjali Sardana. "Enhanced 3-Way Handshake Protocol for Key Exchange in IEEE 802.11i," *3rd IEEE International Conference on Electronics Computer Technology (ICECT 2011)*, vol. 6, pp.132-135, Kanyakumari, India, 8-10 April, 2011.

[2] **Dinesh Yadav** and Anjali Sardana. "Authentication Process in IEEE 802.11: Current Issues and Challenges," 4th *International Conference on Network Security & Applications (CNSA, 2011)*, Chennai, India, 15-17 July-2011. Publisher: LNCS Springer. (Accepted and Registered)