國立交通大學

資訊科學與工程研究所

碩 士 論 文

改良式隨機位元認證機制抵禦 802.11

無線網路阻絕式攻擊

Enhanced Random Bitsream Authentication to Defend IEEE

802.11 DoS Attacks

研 究 生：黃柏翰

指導教授：蔡文能　教授

中 華 民 國　九 十 六　年 七 月

良式隨機位元認證機制抵禦 802.11 無線網路阻絕式攻擊

Enhanced Random Bitsream Authentication to Defend IEEE 802.11 DoS Attacks

研 究 生：黃柏翰　　Student：Po-Han Huang

指導教授：蔡文能　　Advisor：Wen-Nung Tsai

國 立 交 通 大 學

資 訊 科 學 與 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

July 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年七月

中文摘要

改良式隨機位元認證機制抵禦 802.11 無線網路阻絕式攻擊

國立交通大學資訊工程學系（研究所）碩士班

摘　　要

網路已經成為許多人日常生活的一部份，尤其無線網路 IEEE 802.11 更是蓬勃發展，但 802.11 卻因本身特性與協定的問題，而存在許多的安全漏洞。

802.11(a,b,g)以 WEP 為其安全的機制，但已被證實存在許多弱點而容易被破解。IEEE 為了解決 802.11 安全問題，因此制定了 802.11i 為新一代的無線網路標準。雖然 802.11i 解決了完整性(integraty)與機密性(confidentiality)的問題，但在驗證性（Authentication）與可及性(Availability)上卻有嚴重的設計缺陷，使得 802.11 系列的無線網路還是容易遭受阻絕服務(Denial of Service，DoS)攻擊。

[1]提出了「SND（Sequence Number Deviation）與隨機位元串」兩層過濾機制防禦 DoS 攻擊，本篇論文以[1]為延伸，針對其未完整說明與可改進部份，修改成本研究的「DSN（Dynamic Sequence Number）與隨機位元串」兩層過濾機制。「DSN 與隨機位元串」兩層過濾主要概念是在認證（(de)authentication）及連結（(dis)association）訊框中，加入變動的 SN 值與數個隨機位元，作為擷取點（AP）和無線工作站（STA）雙方溝通的認證機制。本研究設計的「DSN 與隨機位元串」兩層過濾 DoS 攻擊機制，經實作與實驗模擬後證明不僅能有效的抵禦 802.11 無線網路阻絕式攻擊，並且與[1]比較後安全與效能都大幅提昇。

英文摘要

Enhanced Random Bitsream Authentication to Defend IEEE 802.11 DoS Attacks

Institute of Computer Science and Information Engineering

National Chiao Tung University

Abstract

Networks have already become a major part of people's life with IEEE 802.11 networks growing the most. However, 802.11 has considerable security issues. WEP is the security mechanism in 802.11 specification, but was proved vulnerable and easy to be cracked.

802.11i is an enhanced version of 802.11. It is concentrated on improving integrity and confidentiality of 802.11 frames. But, the availability of 802.11(i) network is not handled properly. The problem lies in the lack of any key-based authentication mechanism for the protection of the 802.11(i) management frames, As a consequence, 802.11(i) networks are vulnerable to Denial of Service (DoS) attacks.

To solve the problem, Chien et al. [1] proposed a two-phase filtering mechanism consisting of Sequence Number Deviation (SND) and Random Bitstream (RBS) to mitigate the effect of DoS under 802.11 networks. Our research is extended from [1] by implementing a random bitstream generator and enhancing its SND phase. We also considered the synchronization problem in exchanging the authenticating random bitstream. Our proposed scheme thus becomes a new Dynamic Sequence Number (DSN) and RBS, two-phase filtering mechanism. In our experiments, it was demonstrated that our new two-phase filtering mechanism is more effective at defending IEEE 802.11 networks from DoS attacks than the mechanism proposed in [1].

致　　謝

在兩年的努力之下，終於完成了我的畢業論文，中間遇到了許多的困難與問題，也受到許多人的幫助，在此一一感謝。首先要感謝的是我的指導教授—蔡文能教授，在碩士班兩年的之中，他給我許多方面的指導，讓我受益良多，也因此成長了不少。再者，感謝父母與兩個姐姐給我一個平穩的環境在背後支持著我，讓我能夠專心學習。接著要感謝的是實驗室的學長姐與同學們，大家一起工作、學習，並且互相幫忙，讓我有個難忘的碩士回憶。需要感謝的人實在太多了，在我難過或沮喪時，他們都能給我幫助與支持，讓我繼續支持下去，真心感謝曾經幫助過我的人，謝謝你們。

目錄

圖目錄

表目錄

第一章　緒論

從早先的撥接，到 ADSL/Cable 寬頻上網，網路已經成為許多人日常生活的一部份，無線網路 IEEE 802.11 更是漸漸成為主流。雖然無線網路架構出一個無論何時何地都能快速連上網路的環境，但是安全問題卻遠遠比有線網路嚴重許多。802.11 就因為本身的特性與協定制定的問題，而存在許多安全漏洞。所以在此章節中，會先於 1.1 節介紹本論文欲解決的安全問題。1.2 節介紹研究的目標。1.3 節介紹本論文的組織架構。

1.1 研究動機

無線網路的使用十分方便，並且已經慢慢成為主流急速地成長中，根據國科會資科中心的報告，從 2000 年到 2006 年間無線網路的使用率大幅的成長（如圖 1 所示）。但是相對地，水能載舟，亦能覆舟，由於無線電波的特性，使得無線網路比有線網路多了許多安全性的考量，例如：任何人皆能輕易地存取無線網路，使得在無線網路上傳輸的資料無法受到保障。再者協定標準（Protocol）與安全機制亦具有嚴重的漏洞存在，例如：IEEE 802.11 中定義的 WEP（Wired Equivalent Privacy）在 2001 年 Fluhrer、Mantin 和 Shamir 發表的一篇相關研究論文後，網路上就出現破解 WEP 的程式。此外攻擊者也可能使用您的無線網路來進行對外的一些攻擊行為。

無線網路中有四種安全考量，分別是資料的保密性(Confidentiality)、資料的完整性(Integrity)、驗證性（Authentication），與可及性(Availability)。IEEE 為了解決 802.11 安全性的問題，因此制定了 802.11i 為新一代的無線網路標準。雖然 802.11i 解決了保密性和完整性的問題，但是對驗證性與可及性卻有嚴重的設計缺陷，使得 802.11 系列的無線網路還是容易遭受阻絕服務(Denial of Service，DoS)

攻擊。所以本研究將設計一個有效且負擔輕的機制來預防 802.11 系列容易遭受阻絕服務攻擊的問題。

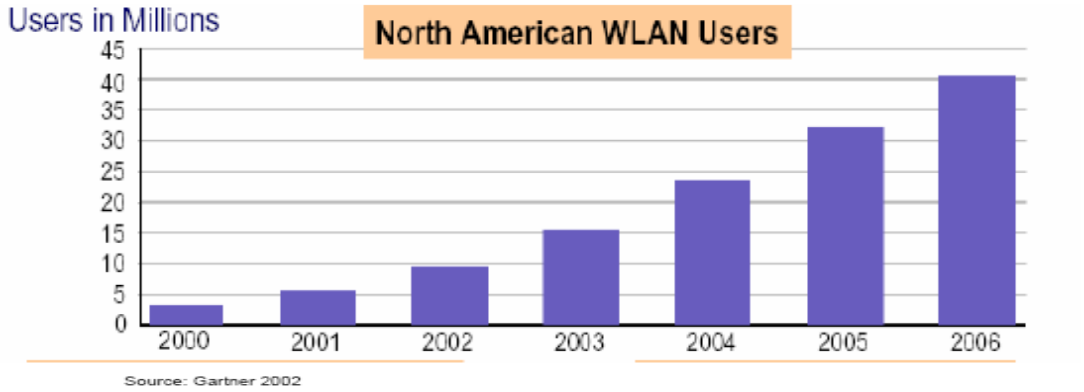


圖 1 無線網路在北美的成長率

1.2 研究目標

本篇論文研究目標分成下列四個部份：

第一，本研究首先讓無線工作站（Station）與擷取點（Access point）利用Diffie-Hellman金鑰交換演算法以產生相同的金鑰（Shared Key）。之後再利用此把共同金鑰輸入本研究所設計的認證隨機位元產生器（Authentication Random Bitstream Generator，ARBG）中，以便產生用來互相認證的相同隨機位元串。此外在設計ARBG時，我們必須要確保此認證隨機位元串是攻擊者（attacker）難以分析出來的，亦無法從其推算出無線工作站與擷取點的共同金鑰。

第二，在無線工作站與擷取點產生相同的認證隨機位元串之後，利用 802.11 在 MAC 層中的訊框主體（Frame Body）欄位，於認證（(de)authentication）及連結（(dis)association）訊框中加入動態的順訊控制值（Dynamic Sequence Number）與數個隨機位元串，以作為擷取點和無線工作站雙方溝通的認證機制，如此達到有效過濾偽造的阻絕式攻擊封包的兩層防禦。

第三，設計一個有效率的隨機位元串同步演算法(Synchronization Algorithm)，因為在無線網路的環境下，封包遺失(Lost)或遭到噪音(Noise)的干擾是常有的事情。所以必須讓無線工作站與擷取點指到相同的位元串上，才能用來做互相驗證的工作。同步演算法的設計是非常重要的問題，因為演算法的設計優劣會直接影響到我們所提出的機制的效能，在本篇論文我們將提出一個簡單、有效率的方式來達成同步機制。

第四，我們會經由分析與實驗證明本研究所提出的兩層防禦機制，不僅是可行、有效率(efficient)且負擔輕(lightweight)，並可與 IEEE 802.11 (a,b,g) 向後相容(Backward Compatible)。

1.3 論文架構

本篇論文的組織架構如下：『第一章 緒論』介紹研究動機目標。『第二章 背景知識』介紹無線網路 IEEE 802.11 的運作原理、特性、802.11 家族與安全問題、阻絕服務攻擊、以及本論文中應用到的相關知識。『第三章 相關研究』依序介紹研究 IEEE 802.11 阻絕服務攻擊的相關論文、現階段提出防禦 IEEE 802.11 阻絕服務攻擊機制的相關論文、本篇論文的兩層防禦機制想法來源，最後再介紹論文中會應用到的相關技巧。『第四章 系統架構』本論文將以 3.3.2 節中的[1]為延伸，將在此章中修改[1]論文中有缺陷的部份，並完成作者沒有說明完整的地方。『第五章 分析與實驗結果』利用實驗證明本論文提出的新防禦機制不僅可行，且比[1]安全且有效率。『第六章 結論』提出結論與未來方向。

第二章 背景知識

此章節將簡介一些無線網路的背景知識，以及為何 802.11 容易遭受阻絕服務攻擊的原因。2.1 簡介線網路，包含其運作、特性、家族規格的演進。2.2 說明 IEEE 802.11 安全問題，以及 IEEE 802.11i 為何沒有解決組絕服務攻擊的原因。2.3 介紹服務阻絕攻擊的種類。2.4 介紹如何利用 Diffie-Hellman 金鑰交換演算法讓無線工作站與擷取點產生共同的金鑰，以便用來產生相同的驗證隨機位元串。

2.1 無線區域網路簡介

本研究探討的問題是有關無線網路安全，但要了解問題時，必須先對無線區域網路的架構有初步認識。所以底下先介紹 IEEE 802.11 的運作原理、特性、及 IEEE 訂定的一系列相關標準。

2.1.1 無線區域網路運作原理

無線網路就是不使用實體線路所構成的網路，目前大部分的無線網路技術都是使用無線電波為傳輸媒體，另外有一小部分則是使用紅外線或藍牙為傳輸媒體。所以，我們也可以說無線網路就是以無線電波或紅外線做為傳輸媒介，而構成的通訊網路。目前常用的協定為 IEEE 802.11，其主要目的是要製定一套適合在無線區域網路環境下作業的通訊協定，依架構可以分為兩種，如下圖 2：

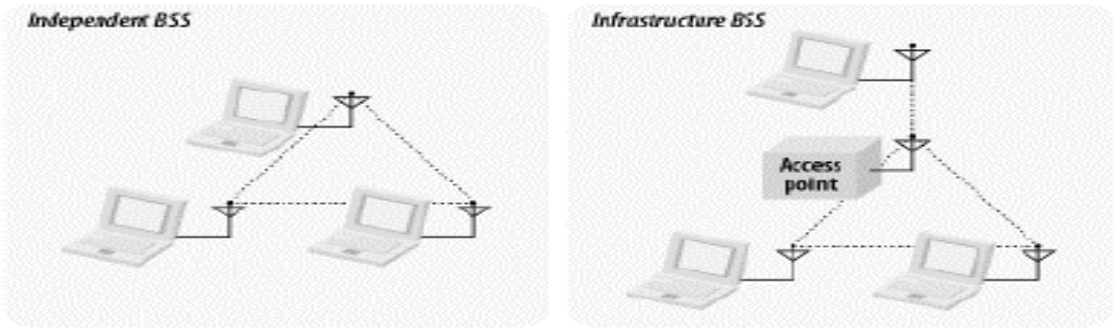


圖 2 獨立基礎架構與基礎架構[31]

獨立基礎架構(Independent BSS（Basic Service Set），亦稱 Ad-Hoc Mode)：

在這個架構下沒有擷取點（Access Point），無線工作站只要彼此在乎互相的範圍內，就可以退過點對點的方式直接互相通訊，而不需透過有線的架構來輔助。

基礎架構(Infrastrucutre BSS，亦稱 Infrastructure Mode)：

在這個架構下是靠有線的架構來輔助無線工作站上網，通常是以擷取點作為溝通的橋樑，擷取點同時擁有有線網路介面與無線網路介面，所以任兩台無線工作站如果要彼此互相通訊，或是無線工作站要與有線網路通訊，都必須透過擷取點來傳送資料。

2.1.2 無線網路的特性

無線區域網路與有線網路不僅在架構上有很大的不同，協定與封包格式也不相同。所以無線區域網路的特性也與有線網路有很多的不同之處。底下為無線網路與有線網路的差別：

（1）無線電波傳輸: 利用無線電波來做為資料的傳遞，它與有線網路的用途相似，兩者最大不同的地方在於傳輸資料的媒介不同，一個使用無線電波，另一個

使用實體線路。由於它是無線電波來做為資料傳遞，因此在硬體架設或使用之機動性均比有線網路要方便許多。

（2）易受干擾：如上所述，因為在空氣中利用無線電波傳送，所以會牽涉到訊號強弱的問題，距離的遠近、無線工作站的位置、週遭是否有其他相同頻段的裝置（例如：微波爐、無線電話等等）在正使用，都會干擾到資料的傳送。所以無線區域網路在資料傳送的可靠性來講，比有線網路來得不穩定。

（3）無線工作站可以漫遊移動（roaming）：無線工作站可以漫遊於 BSS 或 ESS（Extended Service Set）中。所以無線網路的目的位址通常不等於目的位置。在有線網路裡，一個位址通常就代表一個固定的位置，然而在無線網路裡，這件事不一定成立，因為在無線網路中，事先被給定位址的電腦，隨時都有可能會移動到不同的地方。

（4）安全性比有線網路低：由於無線電波是在空氣中傳送的，所以只要在傳輸範圍內有任何人都可以收到訊號，而且訊號的傳輸範圍難以界定，要防止監聽相當困難，所以無線區域網路的安全威脅比有線網路嚴重許多。

2.1.3 802.11 家族的演進

802.11 家族為 IEEE 發展無線區域網路而訂定的一系列規格，在 802.11 家族中，IEEE 針對了不同的問題而定義了不同的規格，例如: 802.11n(速度提昇)、802.11e（Quality of Service，QoS）、802.11i（安全問題）等等。

802.11b：

802.11b 又被稱為 Wi-Fi，用於 2.4GHz 頻帶，其最大速率可達 11Mbps，實際速率 5Mbps，為目前常用的傳輸協定之一。

802.11a：

802.11a 為 802.11b 的後續標準，用於 5GHz 頻帶，最大速率可達 54Mbps，實際傳輸速率為 25Mbps。該標準採用正交頻分多路複用(OFDM)調制技術，此外 802.11a 的 5GHz 頻段無法與 802.11b 相容。

802.11g：

802.11g 向後相容 802.11b，用於 2.4GHz 頻帶。但它與 802.11a 一樣擁有 54Mbps 的傳輸速率，可滿足 802.11b 更大的資料傳輸。

802.11e：

802.11e 標準提供了兩種提高服務保障的新機制，即訊息通道爭用期的增強分散式協調功能(EDCF)與非訊息通道爭用期的混合點協調功能(HCF)來增加無線區域網路的服務品質。

802.11i：

802.11i 主要針對 802.11 安全性方面強化，訂定了強健性安全網路（Robust Security Network, RSN）與過渡性安全網路（Transitional Security Network, TSN）。

RSN 主要是為了加強資料加密與認證性能而訂定，並針對 WEP 加密機制的各種缺陷改進。TSN 主要是因為現有無線網路卡無法升級支援，以至於運算能力不足。因此 Wi-Fi 聯盟提出 WPA，作為要變更到 802.11i 的一種過渡方案。

802.11w：

802.11w 主要是針對管理訊框中沒有受到加密保護的機制，造成攻擊者可以利用的缺陷做改進，目前還在訂定當中。

2.2 IEEE 802.11 安全問題

一般來說，不管是無線或有線網路的安全機制，大致上可以從四個角度分析，分別為機密性（Confidentiality）、完整性（Integrity）、驗證性（Authentication）與可及性（Availability）。根據以上的四個標準來看，IEEE 802.11 安全問題可歸咎為三個原因：

（1）無線通訊的特性：

由於無線網路先天設計便是以無線電技術為基礎，使得攻擊者得以無線電波涵蓋的範圍內進行通訊內容的監聽。如果使用者未將傳送的資訊適當的進行加密，則入侵者很容易的便可以竊取所有的通訊內容。另外，由於無線通訊只要電波收訊範圍內即可使用，也造成了管控上的麻煩，管理者無法完全的進行存取控制。

（2）加密機制設計錯誤：

在 IEEE 802.11 的標準中訂定了 WEP 的標準，希望透過這種加密技術能讓使用獲得更好的資料安全性，但是由於某些設計上的錯誤，使得 WEP 無法保護資料內容。此外亦沒有考慮到金鑰管理的問題，因此如果有一個很大的無線區域網路的話，金鑰的修改及配送亦會是一個很大的管理問題。

（3）驗證性與可及性問題

在 802.11 中有三種訊框，分別為管理訊框（Management Frame）、控制訊框（Control Frame）與資料訊框（Data Frame）。控制訊框是用來作為頻道的宣告或載波感測的狀態維護（例如：RTS、CTS）。管理訊框是用來執行管理功能（例如：Authentication/Deauthentication、Association/Diassociation）。資料訊框則是用來傳送資料的。

802.11 為什麼會有可及性的問題呢？問題出在管理訊框與協定上面，因為無線區網不依靠實體佈線來傳輸資料，那麼無線區域網路要怎麼樣來維持擷取點與無線工作站之間的虛擬網線路呢？答案是靠管理封包來建立連結(Association)。建立這個虛擬線路主要有兩大步驟，第一個是先做身分認證的動作(Authentication)，第二個就是建立連結(Association)的動作。那既然有建立的動作就會有相對應的解除動作，在無線區網所用的通訊協定裡這兩個動作靠的是 Deauthentication 與 Disassociation 兩個封包。然而，設計者在訂定這樣的通訊協定時，並未將身分認證或加密機制加進這些管理虛擬線路的訊框裡，而造成任何人都可以偽造這些管理訊框，然後任意切斷正常使用者的連線（或是干擾網路服

務)，而造成服務阻絕攻擊（如圖 3 所示）。

經過上面的詳述，我們可以知道 IEEE 802.11 系列的無線網路，有上述三種安全問題。因此 IEEE 制定了 802.11i 以改善 802.11(a,b,g)無線網路資料傳送的完整性及機密性的安全問題，但在驗證性與可及性卻沒有嚴謹的考量與設計，也就是 802.11i 繼承了 802.11（a,b,g）先天上的設計缺陷（i.e.容易遭受到阻絕式攻擊的特性)。本研究將在第四章中，針對阻絕服務攻擊造成的驗證性與可及性問題提出解決的機制。

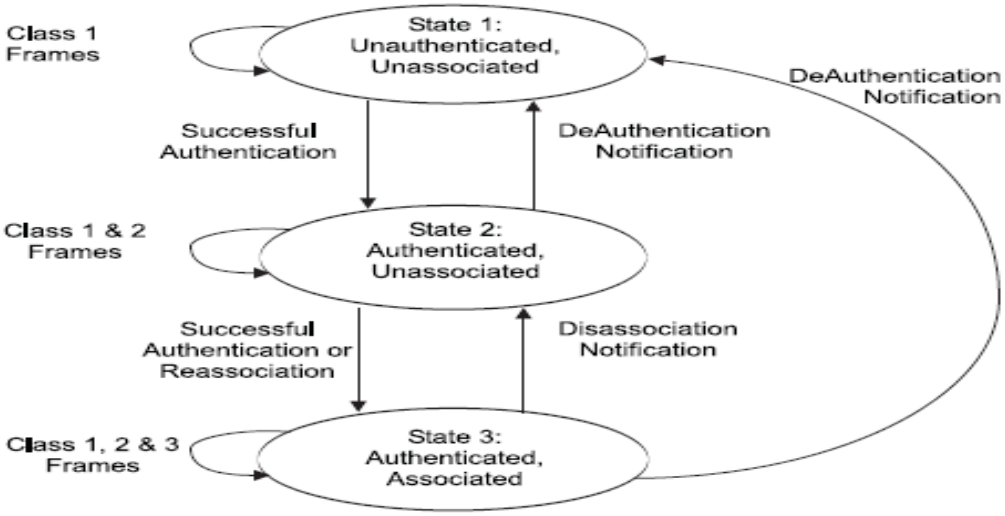


圖 3 802.11 有限狀態機圖

2.3 服務阻絕攻擊（Denial of Service Attack）

本研究是以解決 802.11 系列中容易遭受到服務阻絕攻擊為目的，所以在此小節中，將對服務阻絕攻擊的定義、種類與攻擊手法做介紹，讓讀者對服務阻絕攻擊有更進一步的認識。

2.3.1 服務阻絕攻擊定義

服務阻絕（Denial of Service，DoS）攻擊是指故意攻擊協定的漏洞，或直接使用暴力法耗盡攻擊目標的資源(CPU、網路、硬碟空間等等)，目的不是取得系

統使用的權力，而是讓目標電腦或網路無法提供正常的服務，甚至系統崩潰（Crash）。而另一阻絕服務攻擊的特例為分散式阻絕服務攻擊（Distributed Denial of Service，DDoS），駭客利用多台機器同時攻擊來達到妨礙正常使用者使用服務。分散式服務組絕攻擊非常難以抵擋，因為駭客可能一次控制高達上百台電腦展開攻擊，例如：在西元 2000 年，Yahoo 與 eBay 就遭受到了分散式阻絕服務攻擊，不僅網站癱瘓，並且因而損失達上千萬元。

2.3.2 服務阻絕攻擊種類

要發動服務阻絕攻擊的方法很多，也有很多種軟體可以達成。不過大致可以分為以下三種目的，來達成服務阻絕的目的：

（1）消耗頻寬：駭客利用一些指令去消耗可用的頻寬。例如：smurf 攻擊，直接對網路進行廣播，造成網路很快地充滿垃圾封包而中斷。smurf 會不斷地將小量偽造的 icmp 要求封包送給 IP 廣播位址，然後廣播位址會傳回大量的 icmp 回應封包給目標主機。這種 smurf 的攻擊方式除了攻擊特定目標主機，也能在網路上塞滿 icmp 的要求封包與回應封包而造成網路中斷，稱為加強型的 smurf 攻擊。

（2）消耗系統資源：利用網路主機處理封包的特性進行攻擊，使得受害主機無法處理正常封包，而將資源一直用在主處理垃圾封包而當機，例如：Ping of Death 產生超過 IP 協定所能允許的最大封包。若是沒有檢查機制的系統收到這些過量的封包時，則有可能會造成系統當機。

（3）利用協定缺陷

常見的攻擊利用 TCP/IP 通訊協定中詰問－回應模式的漏洞進行攻擊。例如：SYN-Flood，主要是利用 TCP 連結時的三向交握訊息來造成的。駭客惡意地產生一些大量的 SYN 封包給目標主機，每個 SYN 封包都要求目標主機系統回應一個 SYN-ACK 封包，但是駭客並不產生任何 ACK 封包給目標主機，因此目標主機的系統佇列裡面會暫存大量的 SYN-ACK 封包，目標主機卻收不到 ACK 的

封包，所以目標主機的 SYN queue 會因為儲存太多正在等待連結的資訊而超過其容許量，這些封包必須等到收到對方的 ACK 封包或是超過逾時時間之後才會被移除，所以系統會因為充滿了 SYN-ACK 封包而造成無法服務。本研究中的 802.11 阻絕服務攻擊即屬於此類，細節將於第三章 3.1 節中說明。

2.4 Diffie-Hellman 金鑰交換演算法

在本研究中，因為無線工作站與擷取點需要互相驗證對方的隨機位元串，所以彼此必須要擁有一把相同的金鑰，才能產生相同的驗證隨機位元串。因此我們必須有一套機制可以讓兩個素不相識的無線工作站與擷取點取得相同的金鑰，我們採用的機制是 Diffie-Hellman 金鑰交換演算法。

最早發表的公開鑰匙演算法是由 Diffie 和 Hellman 所提出來的，不少的商用軟體都使用這種金鑰交換技術。這個演算法容許兩個使用者透過某個不安全的交換機制，來共享金鑰以提供後續的加密或應用，而不需要首先就某些祕密值達成協定。Diffie-Hellman 演算法有兩個公開的系統參數，其中一個大質數 P，另一個為 P 的原根 G，$\{G^1, G^2, \ldots, G^{p-1}\} \bmod P$ 可以生成從 1 到 P-1 之間任何一個數。首先，A 與 B 各自生成一個隨機的私有值，即 a 和 b。然後，A 與 B 使用公共參數 P 和 G 以及它們特定私有值 a 或 b 透過一般公式 $G^a \bmod P$ 或 $G^a \bmod P$ 來產生公共值。然後，他們交換這些公共值。最後，一個人計算 $K_{ab} = (G^b)^a \bmod p$，另一個人計算 $K_{ba} = (G^a)^b \bmod p$，則 A 與 B 共享的秘鑰為 $K_{ab} = K_{ba} = K$（如圖 4）。上述 Diffie-Hellman 金鑰交換的方式是基於數學上的要計算離散對數（Discrete Logarithm）是非常困難的，因此其安全是可以信任的。

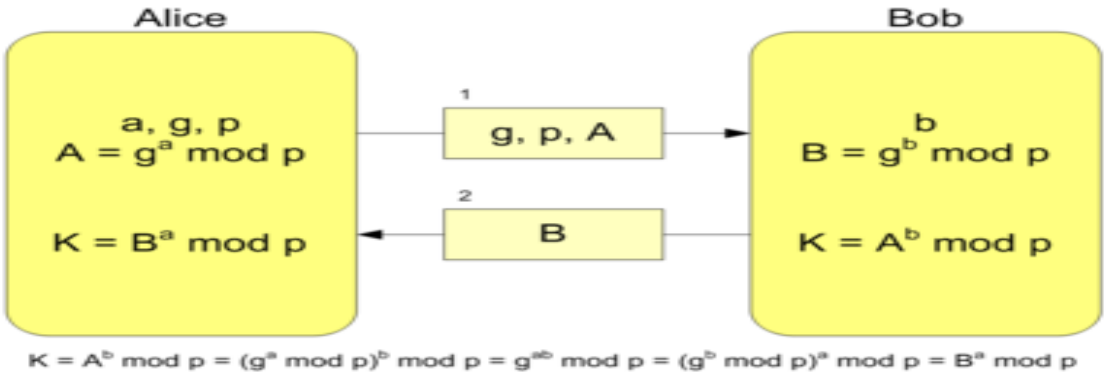


圖 4 Diffie-Hellman 金鑰交換法[37]

Diffie-Hellman 金鑰交換協定可能會遭受到中間人攻擊攻擊（middle person attack）。如果 A 和 B 正在執行交換金鑰，但第三個人 C 有心介入 A 與 B 的金鑰交換。於是 A 認為初始的私有值傳送到了 B。但事實上被 C 攔截，然後向 B 傳送了一個假的私有值，而 B 傳給 A 的私有值也遭受同樣的攻擊（i.e. B 亦以為它的私有值傳送到了 A。這樣的攻擊導致了 A 與 C 擁有共同金鑰，而 B 與 C 亦擁有共同金鑰，但 A 與 B 卻渾然不知。因此 C 就可以在從中攔截 A 到 B 交換的訊息，然後使用 A 與 C 金鑰解密，修改它們，再使用 B 與 C 金鑰轉信到 B，B 到 A 的過程與此相反，而 A 和 B 都將無法察覺。

為了防止這種攻擊，1992 年 Diffie 開發了經認證的 Diffie-Hellman 金鑰協定。在這個協定中，必須使用現有的私密金鑰/公開金鑰對以及與公開金鑰元素的相關數位簽章，由數位簽章驗證交換的初始公共值。不過，本論文中間人攻擊不在的討論中，在此只提供讀者參考。

第三章 相關研究

本章將介紹與本論文相關的一些研究，並於第四章中針對這些缺點加以改善。3.1 介紹 IEEE 802.11（a,b,g）為何會遭到阻絕服務攻擊，與 IEEE 802.11i 為何沒有改善遭到阻絕服務攻擊的相關論文。3.2 介紹最早使用隨機位元串的想法來源，不過這些研究是用於驗證資料方面，而非用於 802.11 阻絕服務攻擊。3.3 介紹目前防禦 802.11 阻絕服務攻擊的機制有哪些，以及分析其優缺點。3.4 介紹串流加密，與本研究隨機位元串的產生器有很大的關聯。

3.1 阻絕服務攻擊於 IEEE 802.11 規格

本論文重點在於防禦 IEEE 802.11 阻絕服務攻擊，而 802.11(a,b,g)與 802.11i 可能會遭受阻絕服務的原因有兩個:（1） S-Band ISM 頻段的干擾。（2）IEEE 802.11 協定本身設計的缺陷。第一個原因是因為 ISM（Industrial Scientific and Medical）頻段是 FCC（Federal Communications Commission）訂定給不需執照的裝置使用。然而目前這個頻段是非常擁擠的，例如:家用無線電話、監視器、X10 攝影機等等，所以使得 IEEE 802.11 容易遭到干擾，造成封包遺失和損毀，以至服務中斷，造成阻絕式攻擊。

我們想要解決的是第二個原因，也就是協定設定不良造成的缺陷問題(第一個原因則是屬於 FCC 頻段規劃使用的問題，本篇論文不討論)。底下 3.1.1 與 3.1.2 進一步介紹 802.11 與 802.11i 設計上的缺陷，以及造成服務阻絕攻擊的原因。

3.1.1 IEEE 802.11 Deauthentication 與 Disassociation 氾濫式攻擊（flooding attacks）

如圖 5 所示，在 IEEE 802.11 系列中，無線工作站與擷取點利用幾個特殊的

管理封包來建立連結。建立這個虛擬線路主要有兩大步驟，第一個是先做身分認證的動作(Authentication)，第二個就是建立連結(Association)的動作。然而，既然有建立的動作，就會有相對應的解除動作。在無線區網所用的通訊協定裡這兩個動作靠的是 Deauthentication 與 Disassociation 兩個訊框。所以攻擊者可以利用這兩個訊框，讓正常使用者無法正常得網路服務。

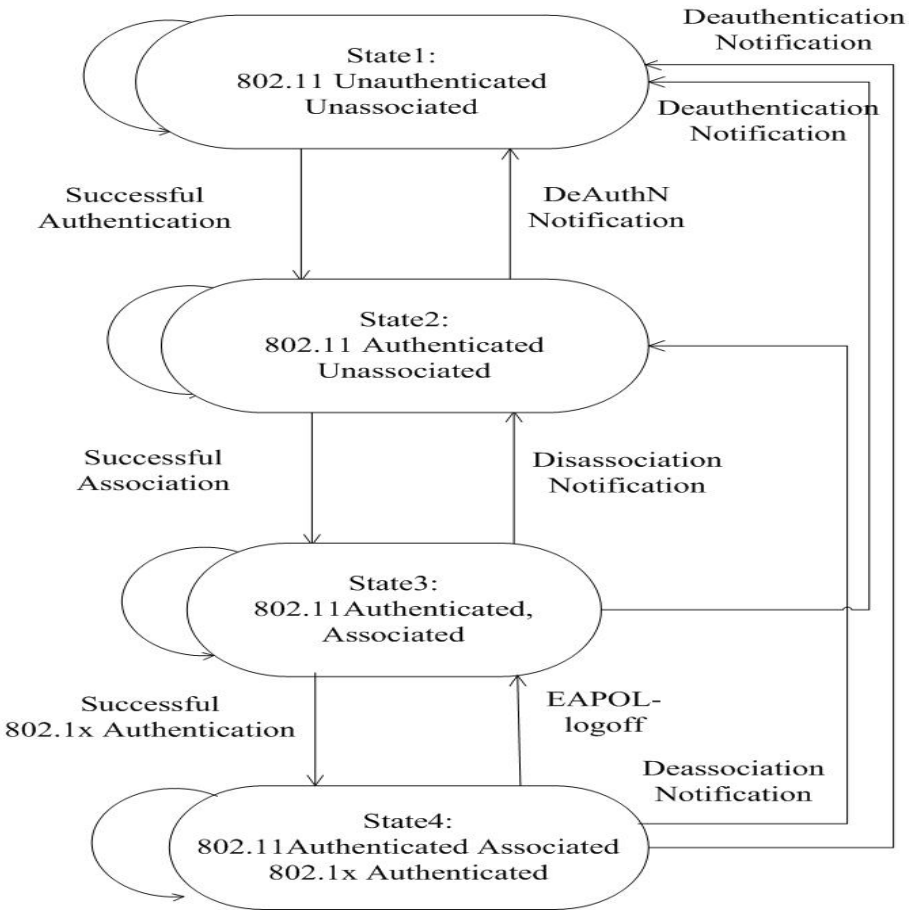


圖 5 802.11 與 802.1x 連結上網協定的有限狀態機 [11][13]

IEEE 802.11 的設計者為了效率問題，所以在設計 IEEE 802.11 的通訊協定時，並未在這些管理虛擬線路的訊框裡面加入互相驗證來源性或加密的機制。所以任何人都可以偽裝成無線工作站或擷取點，發出這些管理訊框後，切斷正常使用者的連線或讓網路處於不穩的狀態，進而達到阻絕式攻擊（如圖 6）。

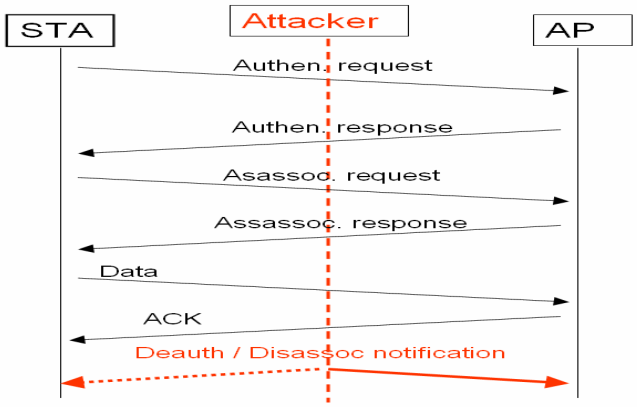


圖 6　802.11 阻絕服務攻擊模式[14]

3.1.2　IEEE 802.11i EAPOL-Failure 與 EAPOL-Logoff 氾濫式攻擊（flooding attacks）

IEEE 為了解決 802.11 安全上的問題，而提出了 802.11i 的標準。不過在[2][3]中提出 802.11i 在防禦類似 802.11（a,b,g）先天協定上的缺陷並沒有改善，也就是 802.11i 繼承了 802.11（a,b,g）的容易遭受阻絕式攻擊的特性，而且 802.11i 的情況比 802.11（a,b,g）更嚴重。802.11i 中的 EAPOL-Logoff、EAPOL-start、EAPOL-Success 與 EAPOL-Failure 訊框都沒有互相驗證來源性或加密的機制，使得有心的人（攻擊者）只要偽裝成擷取點或無線工作站，發出假的訊框，就可以讓使用者斷線，或者無法正常的使用網路服務[2][3][4][5]。

802.11i 協定流程如圖 7 所示，紅色框框中表示傳送的訊框未使用任何金鑰加密或驗證來源性的機制加以保護，所以攻擊者可以利用這些紅色的框框中的訊框發出假造的訊息，如此就可以騙過系統，讓正常使用者斷線，或是做任何會危及系統安全的事。

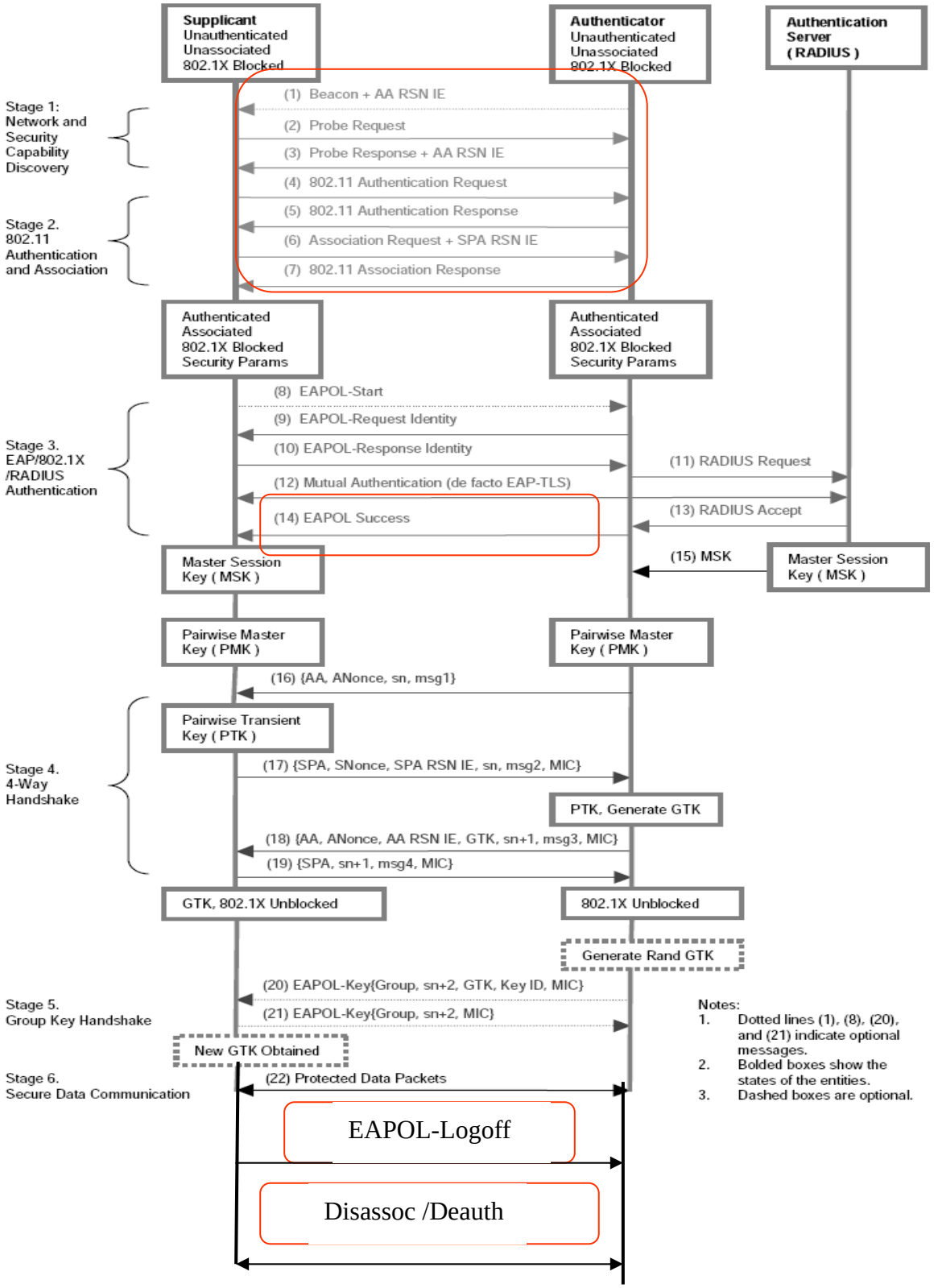


圖 7 802.11i 協定流程（紅色區塊表未加密)[3]

3.2用於 IEEE 802.11 的輕負擔驗證機制（Lightweight Authentication on IEEE 802.11）

[6][7][8][9][10]利用隨機位元串的方式驗證來源性，防止未經授權存取或修改，其不需複雜加密的驗證，主要是用於資料傳輸的存取控制（Access Control）方面，以下 3.2.1 與 3.2.2 介紹使用隨機位元串的方法與其演進過程。

3.2.1 IEEE 802.11 單一隨機位元驗證（One-bit Lightweight Authentication on IEEE 802.11）

在此小節中，要介紹的是[6][7][8][9]四篇。首先[6][7]提出SOLA（Statistical One-bit Lightweight Authentication）身分認證協定，主要的想法是先假設在傳送端與接收端彼此擁有相同的金鑰、會期金鑰與ASG（Authentication Stream Generator，ASG），利用會期金鑰帶入ASG中算出相同的隨機位元串。然後從這相同隨機位元串中，選取一個位元加入到資料訊框（Data Frame）中未用到或保留的欄位，例如：MAC或IP表頭中未用到或保留的位置。如圖8所示，傳送端在把訊框傳送給接收端，接收端收到後會先去比對隨機位元串。如果比對相符的話，接收端會回覆ACK-Success（代表為合法使用者所傳送的資料）給傳送端；反之，接收端會回覆ACK-failed（代表不合法使用者所傳送的資料）給傳送端。

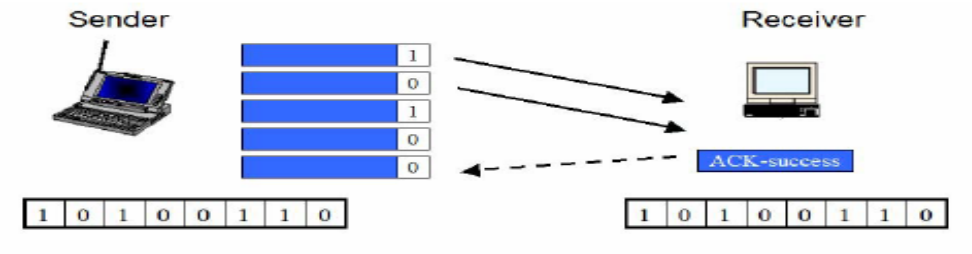


圖 8 SOLA 協定運作

然而在無線網路的環境下，封包的遺失或遭受到干擾是很容易發生的，所以必須確保傳送端與接收端必須指到同一個位元位置上，如此傳送端送給接收端的隨機位元才會相符。底下表1為[6][7]論文提出的同步演算法，圖9則為在最佳情況下的同步過程：

```
Algorithm for AP
1.  // AP receives data packet with Bit[α]
2.       if Bit[α] == Bit[β] then
3.            β++
4.            AP → STA: Packet{ACK, success}
5.       else if Bit[α] ≠  Bit[β] then
6.            β = β + opposite bit + 1
7.            AP → STA: Packet{ACK, failed}
End Of Algorithm

Algorithm for STA
1. //   STA receives ACK packet with success or
         failed bit from STA
2.       if bit == success then
3.            α++
4.       else if bit == failed then
5.            α = α + opposite bit + 1
End Of Algorithm
```

表 1 同步演算法[6][7]

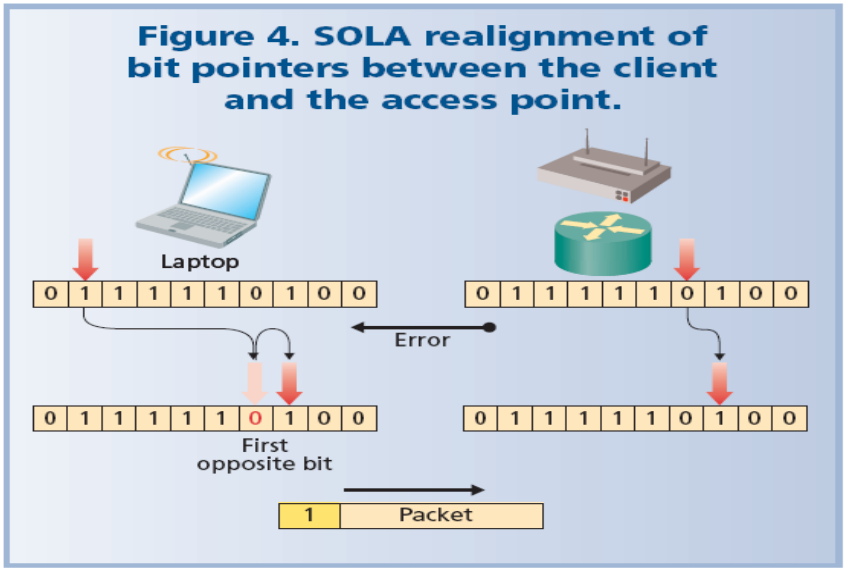


圖 9 同步演算法實例[6]

值得一提的是，在[6]中提到不能使用訊框表頭中的順序控制（Sequence Number）來當作同步隨機位元串的方法。因為當傳送端傳送封包給接收端時，

若接收端回傳給傳送端的封包遺失，則傳送端會重新發送一個封包給接收端，此時的順序控制的參數是不變的，但是因為剛剛接收端有收到封包，所以必須要往前指到下一個隨機位元串上，如此就會造成不同步。所以我們不能使用順訊控制來達到同步的過程。

[8][9]中分析在這樣的同步演算法下，若要使得傳送端與接收端恢復同步的話，則最多需要NSI（non-synchronization index，傳送端與接收端所指到位元串位置的差）次，傳送端與接收端才可恢復同步（如圖10所示）。

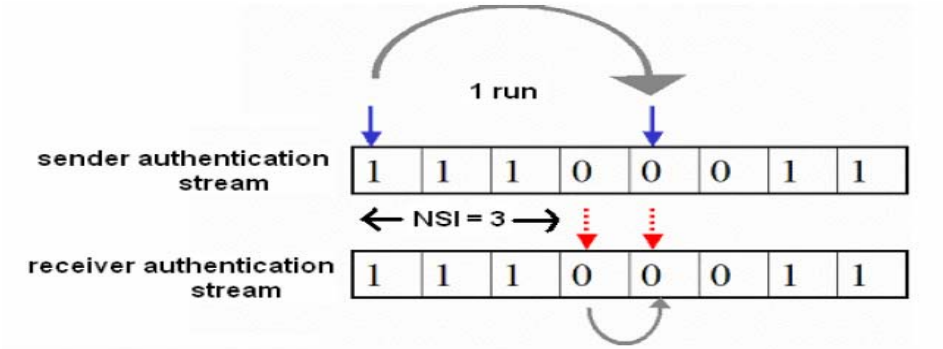


圖 10 同步過程（Best Case）[8][9]

SOLA協定特點：

簡單且低負擔：不需加密或解密的計算過程，而不像WEP則必須在每個訊框中使用RC4來做加解密的動作。而且對一個無線工作站來說必須考量到電力問題，所以簡單的計算量代表有效率，且可以長時間運作。

有效率：在同步的環境下，因為只需檢查一個位元，所以可以很快的完成驗證過程。

實做細節：

要如何把[6][7][8][9]提出的機制加入現有的IEEE 802.11中，並且能向後相容呢？如前所述，將「隨機位元串」與需要的新定義欄位，訂定於原IEEE 802.11訊框中未使用或保留的欄位，如此就可達到向後相容的目的。如表2所示，[8]利用管理訊框的型態（Type）與子型態（Subtype）定義出代表隨機位元串0與1的

格式欄位，以及代表ACK-Success與ACK-Failure的格式欄位。

Frame	Type	Subtype
DATA-(bit=1)	10	1000
DATA-(bit=0)	10	1001
ACK-success	01	0001
ACK-failure	01	0010

表 2 利用保留訊框位置定義出的新型態

3.2.2 IEEE 802.11 加強隨機位元驗證（Enhanced Lightweight Authentication on IEEE 802.11）

在此小節中，要介紹的是[10]這一篇，其提出了一個增強版的存取控制驗證協定（圖 11），加強隨機位元串的部份。原本在[6][7][8][9]都是使用單一位元隨機位元串的方式，來達到資料存取控制的目的。也就是每次傳送端要傳送資訊訊框給接收端的時候，原本[6][7][8][9]都是加入一個隨機位元串，而[10]則改成在一個資料訊框中，一次加入三個隨機位元串，這樣做的好處是可以增加防禦的強度。在每次只使用一個位元加入訊框的情況下，攻擊者使用暴力法攻擊（Brute Force），也有 0.5 的機率會猜中（因為不是 0 就是 1）。然而，若在每次使用三個位元加入訊框中的情況下，只有 0.125 的機率才會猜中，所以防禦的強度也因此而提昇。

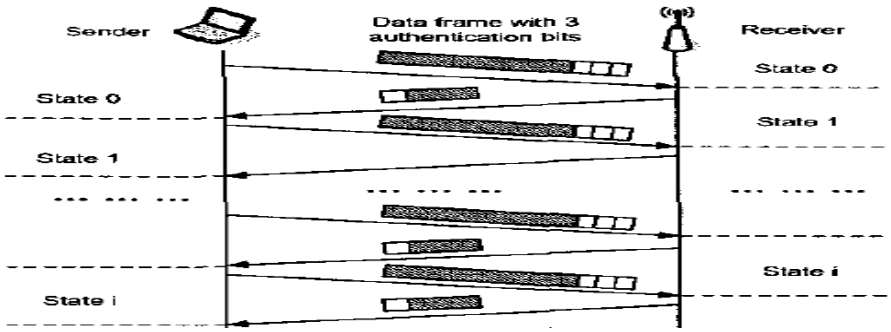


圖 11 增強版的存取控制驗證協定[10]

由 3.2.1 與 3.2.2 小節可知，在無線網路環境下相關的安全防禦是必須的，但是必須考量到無線工作站的負擔與能力，所以[6][7][8][9][10]提出一個負擔輕、高效率、每個傳送出去的資料訊框都可以驗證來源性的機制。

不過，在[6][7][8][9]論文中提出的同步演算法是有問題的。因為若要使得傳送端與接收端恢復同步的話，最多可能要 NSI 次才會讓傳送端與接收端恢復同步。我們覺得這樣的演算法是很沒有效率的，因為無線網路的特性容易遺失訊框或遭受攻擊，在這種情況下很可能會讓系統一直在做同步演算法。所以我們將在第四章的 4.2.5 節中，介紹我們提出的隨機位元串同步演算法，將大幅度地改善同步的效能問題，而其餘相關系統的細節，將於第四章各小節中詳細說明。

3.3 阻絕服務攻擊（DoS）防禦機制

3.1 節中分析了 802.11 為何容易遭受阻絕服務攻擊之因，所以在本節中，將為大家介紹目前防禦 802.11 阻絕服務攻擊機制有哪些，以及它們的優缺點。本論文將在第四章以 3.3.2 小節中的[1]為延伸，幫作者完成研究不足或有缺陷的部份。

3.3.1. 目前防禦 802.11 Deauthentication 與 Disassociation 阻絕服務（DoS）的機制

在[11]中提出有以下四種可能的方式可用來防禦 IEEE 802.11 Deauthentication/Disassociation 阻絕服務攻擊：

(1)停止 IEEE 802.11 Deauthentication/Disassociation 訊框功能：

此種方法是不實際的，因為 Deauthentication/Disassociation 訊框皆有存在的必要，例如：若擷取點為攻擊者偽裝或是有惡意的，正常使用者可以使用 Deauthentication/Disassociation 訊框來離開假的擷取點。

(2)在 Deauthentication/Disassociation 訊框中加入驗證：

在 Deauthentication/Disassociation 訊框中加入驗證機制，也就是說限制只有特定人士可以使用 Deauthentication/Disassociation 訊框。例如：PEAP 利用 keyed integrity check 來保護 Deauthentication/Disassociation 攻擊，不過卻需要花費相當的計算與資源。而在[12]提出 per-frame 的驗證機制，每個訊框在接收的時候都

要先經過驗證，然後才會執行相對應的動作。

在以上兩種 PEAP、[12]或其他複雜的驗證機制下，都有可能會引發其他的阻絕服務攻擊。例如：攻擊者可以從網路抓到一堆監聽的驗證封包，大量的發給擷取點，而擷取點因為需要花費大量的計算量與資源做驗證的工作，使得原本要預防組絕服務攻擊的美意，卻引發另一種阻絕服務攻擊的機會。

(3) 修改協定：

修改 IEEE 802.11 中的管理訊框使其能支援加密機制，但是這樣的方式是沒有效率的且不實用的機制。在[13]設計了一個 CM（Central Manager）的機制來防禦 802.1x 遭受阻絕式攻擊的問題。CM 是在後端（Back-end）的伺服器，取代 802.1x Radious Server，主要是用來管理一群擷取點和無線工作站。也就是除了負責 802.1x 驗證的工作外，還需負責避免遭到阻絕式攻擊。

擷取點從無線工作站收到 Disassociation 訊框後，擷取點會把這個訊框傳送給 CM。CM 收到此訊框後，會送要求訊框給無線工作站，問無線工作站是否真的想要斷線。無線工作站收到來自擷取點問是否真的要斷線的訊框後，會回覆 CM 確認訊息（Confirmation Message）或否認訊息（Denial Message，表無線工作站根本沒有發這樣的訊框）。CM 收到無線工作站的確認回覆之後，CM 會發 Disassociation-continue 訊框給擷取點，擷取點收到 Disassociation-continue 後，才會真的讓無線工作站斷線。反之，若擷取點收到 Disassociation-igonre，會忽略原本的 Disassociation 訊息，原本的無線工作站則不會斷線。

採用 CM 這種架構有三個缺點，（1）若攻擊者偽裝成擷取點，則攻擊會成功，因為真正的擷取點與 CM 無法收到 Disassociation 訊框。（2）需要更改認證伺服器（Authentication Server）的架構，也就是需要對 802.11 的規格作修改（3）認證伺服器損壞（Single Server Failure），則無法提供驗證服務。

(4) 利用佇列：

利用佇列方式來判斷使用者是真的有發出 Deauthentication/Disassociation 訊框是由攻擊者所發出。在[14]中 J. Bellardo and S. Savage 利用佇列機制來防禦服務阻絕攻擊。擷取點會讓 Deauthentication/Disassociation 訊框停佇（queuing）5

到 10 秒，如果等一下從無線工作站來的訊框是資料訊框的話，則放棄上次的 Deauthentication/Disassociation 訊框。因為合法的使用者既然要繼續使用網路的話，是不太可能連續發送 Deauthentication/Disassociation 訊框，所以極有可能是攻擊者所發送的。反之，無線工作站也可以借由佇列的機制，來觀察是法真的是從擷取點所發的 Deauthentication/Disassociation 訊框。J. Bellardo and S. Savage 是使用 FTP 來設計實驗，雖然成功的防禦了阻絕服務攻擊，但是有一些缺點，如下：

1. 延遲現象(Delay）：擷取點則必須等待 5 到 10 去檢查是否為真的 Deauthentication/Disassociation 訊框，所以會導致系統延遲的問題。

2. 實驗使用 FTP 測試：雖然於 FTP 測試上實驗證明可以防禦阻絕服務攻擊，但假若測試環境為使用者在瀏覽網頁時，則無法有效防禦。因為使用者通常會停留一段時間觀看網頁，此時並不會有資料的傳送，若攻擊者在這時候發 Deauthentication/Disassociationn 訊框，則可以成功達到攻擊目的。再者攻擊者可能會在發了 Deauthentication/Disassociation 訊框後，在發假的資料訊框給擷取點，亦可以逃過截取點的過濾。

3.3.2. 新穎的防禦 802.11 Deauthentication 與 Disassociation 阻絕服務（DoS）的機制

在 3.3.1 防禦機制中，缺點為成效不彰、複雜加密運算或需要修改協定。為了要讓無線網路能夠有效率地運作，[1]提出了一個新穎、負擔輕（Lightweight）的方法，其不是利用上面四種機制來預防阻絕服務攻擊，而是使用 3.1 節中[6][7][8][9][10]中的隨機位元串驗證機制。在 3.1 節中的研究主要是將隨機位元串用於 IEEE 802.11 中資料訊框（Data Frame）的存取驗證；而[1]則是將隨機位元串應用於防止 IEEE 802.11 管理訊框（Management Frame）的偽造問題，進而防止阻絕服務攻擊。

[1]在共有金鑰的假設下，在認證（(de)authentication）及連結（(dis)association）

封包中，以隨機方式加入 3 到 4 個位元於 802.11 在 MAC 層的封包標頭結構中，以便讓擷取點和無線工作站可以互相溝通認證，並配合 MAC 層封包標頭中的訊框控制（Sequence Counter，SN）欄位值為連續正整數的特性，設計有效過濾偽造的阻絕式攻擊封包的機制（如圖 12,13 所示）。

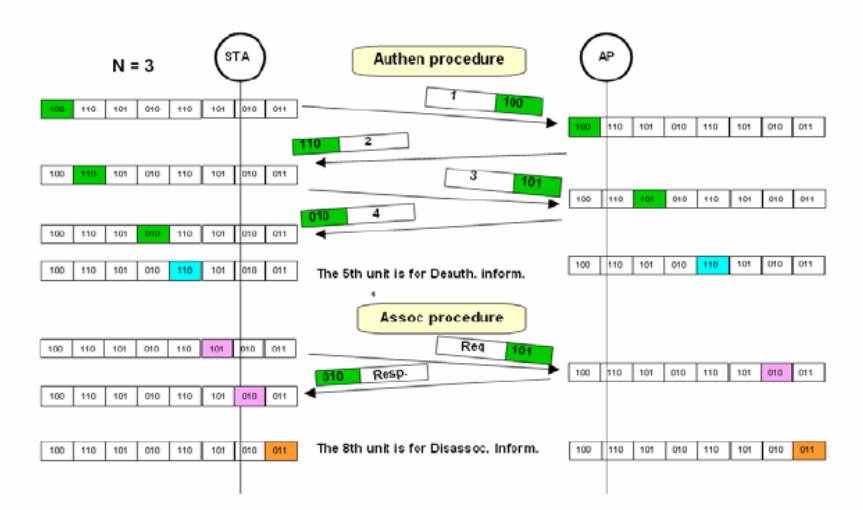


圖 12 隨機位元串的劇本（未遭受到攻擊）

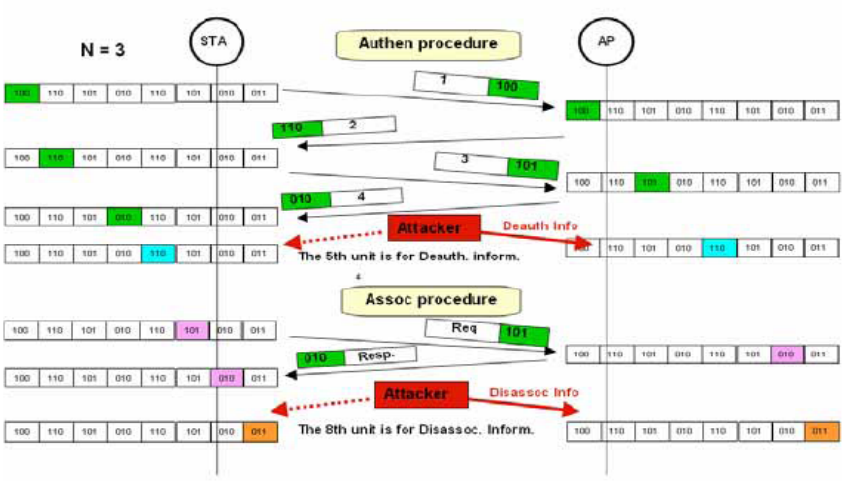


圖 13 隨機位元串的劇本（遭受到攻擊）

[1]研究設計的抵禦無線網路阻絕式攻擊機制，經過作者的實作與實驗模擬後，得到兩種方式可以有效的抵禦 IEEE 802.11 無線網路阻絕式攻擊：

（1）只使用隨機位元串驗證的方式，需要 8 個以上的隨機位元串加入訊框中，才可以有效防禦阻絕服務攻擊。或是只用 SND（$SN_i - SN_{i-1}$，Sequence Number）

防禦，需設 SND 為 16 以上才可以防禦阻絕服務攻擊。

（2）結合 SND 與隨機位元串機制，則使用 3 個隨機位元串，與 SND=24 可以有效的防禦 DoS 攻擊。而使用 4 個隨機位元串，與 SND=12，則也有相同效果（如表 3）。

RBN \ SND / Duration		4	8	12	16	20	24	28	32
3	**Deauth flooding attack**	58.3	49	43.8	42.8	41.8	39.1	39.1	38.9
	Disassoc flooding attack	58.1	47.7	43.6	43.2	41.1	39.7	39.2	38.8
	Delay (1)	19.8	10.6	5.4	4.4	3.4	0.7	0.7	0.4
	Delay (2)	19.6	9.2	5.1	4.7	2.6	1.2	0.7	0.3
4	**Deauth flooding attack**	47.7	43.2	39.1	37.7	38.0	37.8	37.7	37.9
	Disassoc flooding attack	44.9	44.3	39.4	37.9	38.1	37.9	38.0	37.7
	Delay (3)	9.3	4.8	0.7	-0.8	-0.5	-0.7	-0.8	-0.6
	Delay (4)	6.4	5.8	0.9	-0.6	-0.4	-0.6	-0.5	-0.8

表 3　[1]提出的結合 SN 與隨機位元串的防禦機制

雖然[1]經過實驗證明使用 SND 與隨機位元串兩層過濾機制，抵禦阻絕服務攻擊是有效的，但是[1]作者在研究中某些部份是沒有完整說明，或有缺陷的，所以本論文將在第四章以[1]為延伸，把底下列出五個部份做修改與擴充：

（1）加入隨機位元串於訊框中的訊框分析不完整

（2）如何讓傳送端與接收端預先共享金鑰

（3）相同的隨機位元串要如何產生

（4）SND（Sequence Number Deviation）防禦機制有缺陷

（5）隨機位元串的同步演算法

3.4 串流加密（Stream Cipher）

本研究因為需要讓無線工作站與擷取點擁有相同的隨機位元串，所以會需要用到串流加密的技巧來產生相同的隨機位元串，而串流加密又與虛擬亂數有很大的關係，所以在此3.4.1小節中會先介紹虛擬亂數。之後3.4.2小節中再介紹我們產生隨機位元串的想法來源。

3.4.1 虛擬亂數產生器（Pseudorandom Number Generator）

在密碼學或網路安全中，大部分是使用虛擬亂數產生器來產生亂數。這些虛擬亂數是經過一個明確的（Deterministic）演算法過程而產生的，所產生的數字串在統計上並不是隨機且不可預測的。但是只要所使用的演算法夠好，其所產生的虛擬亂數通過很多數學測試[16]（Full Period、統計分佈、效能），就可以用在密碼學上（至少有一定的安全度）。最被廣泛使用的虛擬亂數產生器是 Lehmer[18] 所提出的線性同餘法（Linear Congruential Method）演算法，此方程式有四個參數：a、b、m 與 R_{k-1}，如下方程式所示：

$$R_k = (aR_{k-1} + b) \bmod m$$

只要適當地選取 a，b 與 m（例如：a 為 16807，b 為 0，m 為 $2^{31}-1$），則可以通過[16][17]中的安全測試。如表 4 所示，表中的 a，b 與 m 為經過測試的適當選擇，就可以讓 PRNG 產生很大週期。

Table 16.1
Constants for Linear Congruential Generators

Overflow At:	a	b	m
2^{20}	106	1283	6075
2^{21}	211	1663	7875
2^{22}	421	1663	7875
2^{23}	430	2531	11979
	936	1399	6655
	1366	1283	6075
2^{24}	171	11213	53125
	859	2531	11979
	419	6173	29282
	967	3041	14406
2^{25}	141	28411	134456
	625	6571	31104
	1541	2957	14000
	1741	2731	12960
	1291	4621	21870
	205	29573	139968
2^{26}	421	17117	81000
	1255	6173	29282
	281	28411	134456
2^{27}	1093	18257	86436
	421	54773	259200
	1021	24631	116640
	1021	25673	121500
2^{28}	1277	24749	117128
	741	66037	312500
	2041	25673	121500
2^{29}	2311	25367	120050
	1807	45289	214326
	1597	51749	244944
	1861	49297	233280
	2661	36979	175000
	4081	25673	121500
	3661	30809	145800
2^{30}	3877	29573	139968
	3613	45289	214326
	1366	150889	714025
2^{31}	8121	28411	134456
	4561	51349	243000
	7141	54773	259200
2^{32}	9301	49297	233280
	4096	150889	714025
2^{33}	2416	374441	1771875
2^{34}	17221	107839	510300
	36261	66037	312500
2^{35}	84589	45989	217728

表 4 線性同餘法的參數選擇[19]

3.4.2 加密串流的產生

在[20]中提出一個簡單且有效的 SPRiNG 協定，其利用 Lehmer 所提的虛擬亂數產生器來動態產生金鑰流（Keystream），以改善 WEP 中重覆使用同一把金鑰產生金鑰流的缺點。

原本 WEP 中是利用固定的金鑰與初始向量結合，之後在輸入到ＲＣ4 演算法中，得出用來加密的金鑰流，如圖 14。但是如前所述，ＷＥＰ的缺點就是因為重覆使用金鑰與初始向量，所以[20]利用虛擬亂數產生器（線性同餘）的輸出與 RC4 演算法結合，以動態地產生金鑰流與資料加密。如此就可改善 WEP 中金鑰與初始向量重複的缺點，如圖 15。

雖然有很多其它的機制[21][22][23]提出如何改善 WEP，但是因為無線工作站的能力有限，所以必須要考量到硬體以及電力的問題。SPRiNG 因為簡單，所以可以在任何的無線裝置上面有效地執行，而且提供比 WEP 強的安全度。

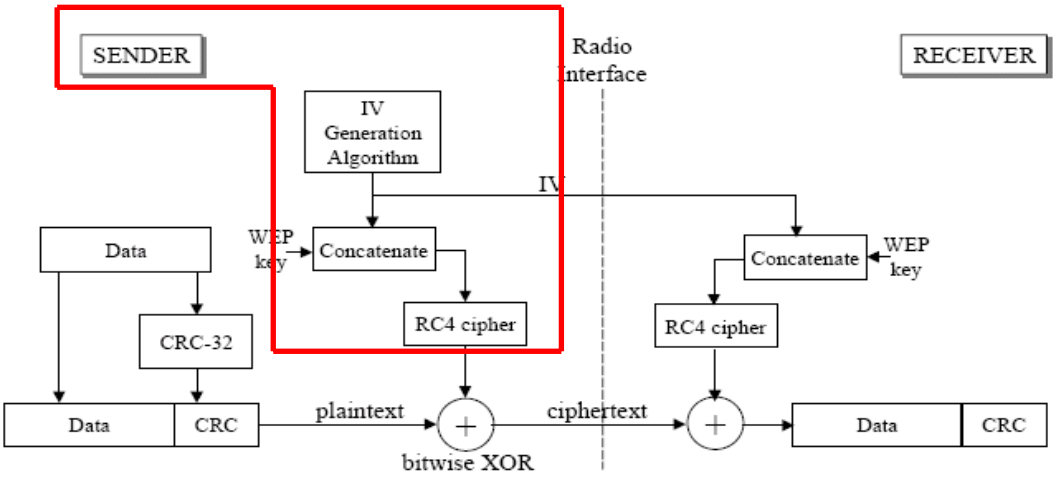


圖 14 仜色框框為 WEP 容易遭受破解主因（重複使用金鑰）

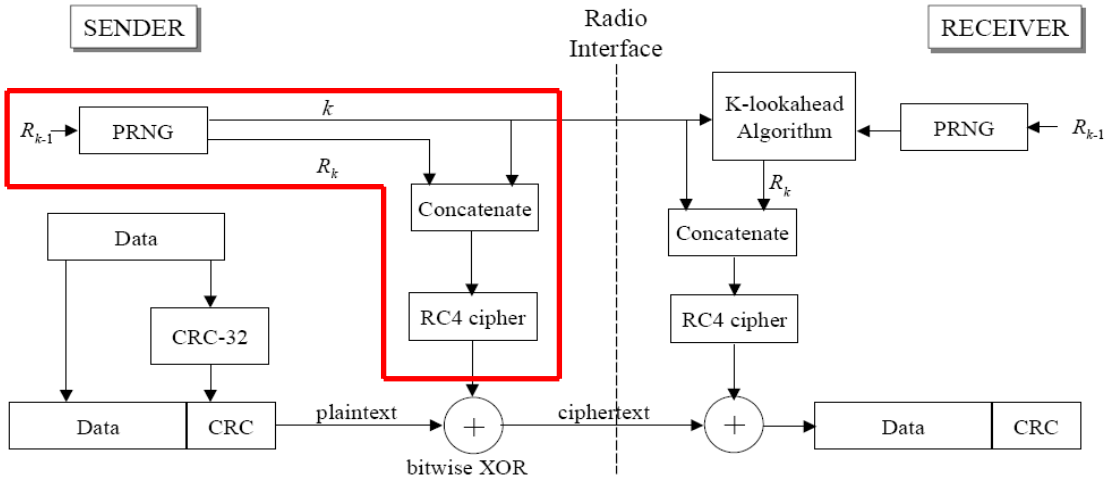


圖 15 仜色框框為[20]所改善 WEP 缺點之處

SPRiNG 的安全性取決於虛擬亂數產生器與 RC4 演算法。我們假設虛擬亂數產生器中的 a、b 與 m 是公開的，雖然只要其中一個 R_k 洩漏出去，整個隨機序列就會被攻擊者知道了，但是 R_k 的資訊隱藏在 RC4 演算法後面，攻擊者很難分析出來。如果要絕對地評估 SPRiNG 的安全性，唯一的方式就是證明 RC4 是難以逆推的函數[24]，所以 SPRiNG 在不僅在安全性上比 WEP 要強的許多（因為虛擬亂數產生器的輸出序列期間比初始向量長，所以可以動態地產生加密流），也較有效率。

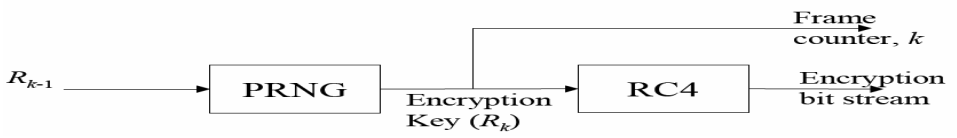


圖 16 SPRiNG 安全分析圖

在本研究中隨機位元串的產生就是透過 SPRiNG 的想法產生的，如圖 16 所示，以便動態地產生我們需要的驗證隨機位元串。這樣的方式不僅安全，且效率亦很理想。不過我們會在第四章做一些變更，讓隨機位元串的產生更安全更適我們的系統（Lightweight），其餘細節部份我們留在第四章，4.2.3 節驗證隨機位元串產生器中討論。

第四章 系統架構

本論文將以 3.2.2 節中的[1]為延伸，其為使用「SND 與隨機位元串」來防禦阻絕服務攻擊的兩層過濾機制。本章節將繼續完成[1]作者研究中未說明分析或有缺陷的部份，首先在 4.1 系統概觀，我們先介紹整個系統架構的流程，包括從擷取點與無線工作站的金鑰交換，到互相驗證隨機位元串等等。接著 4.2 節介紹系統設計，包含 4.2.1 訊框分析，分析 IEEE 802.11 訊框中哪些是可以使用的位置（例如：保留或未使用欄位）。4.2.2 金鑰產生與交換，提出四種可行的金鑰交換方式於本系統中。1.1.4.2.3 討論隨機位元串要如何產生，我們會設計一個 ARBG（Authentication Random Bitstream Generator）模組，以便讓無線工作站與擷取點可以彼此互相驗證的隨機位元串。1.1.4.2.4 修改[1] 中第一層的 SND（Sequence Number Deviation）過濾機制為 DSN（Dynamic Sequence Number），以便增強系統的防禦功能，之後結合「DSN 與隨機位元串」形成新的兩層過濾防禦機制。4.2.5 隨機位元串同步演算法，討論驗證的隨機位元串不同步時，要如何有效率地讓擷取點與無線工作站恢復同步，以便後續的驗證。

4.1 系統概觀

首先，先介紹一般 IEEE 802.11 無線網路上網的程序，因為我們將在此程序中的訊框加入一些額外資訊。如 17 圖，一開始無線工作站必須掃描（Scan）有無擷取點的存在，可以分為被動或主動式掃描。被動式掃描無線工作站會依照頻率順序，依序調整停留接收擷取點所週期發出的訊標（Beacon）訊框，裡面包含了擷取點的資訊（例如：時間同步，群組的識別碼，支援的速率等等）。主動式掃描是由無線工作站發出試探要求（Probe Request）訊框，來尋找週遭的擷取點，

若有擷取點收到此訊框後，就會發送試探回應（Probe Response）給無線工作站，而試探回應訊框裡面也包含了擷取點的相關資訊。

當無線工作站掃描完周遭的擷取點並且收集到相關資訊後，無線工作站會選擇加入其中一個適合的擷取點。選定後，無線工作站會改變自己的狀態（例如：速率、加密支援等等），然後向擷取點表明身份，也就是做身份認證（Authentication）的工作，無線工作站會送出認證要求（Authentication Request）訊框。擷取點收到認證要求後，會回傳認證回覆（Authentication Response）訊框，裡面包含是否認證通過與否。完成身份認證後，無線工作站必須還要與擷取點做連結的動作，才可以要求傳送資料服務。無線工作站會送連結要求（Association Request）訊框，擷取點收到後會送連結回覆（Association Response）訊框，如此才算完成整個 IEEE 802.11 上網流程。

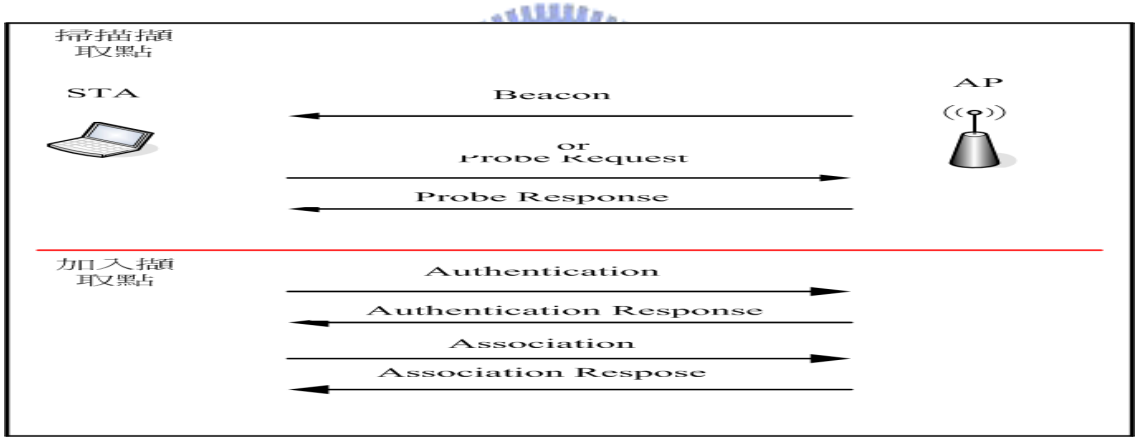


圖 17 一般 IEEE 802.11 無線區域網路上網流程

而在本研究所提出的架構中（如圖 18、19 所示），因為要讓無線工作站與擷取點擁有相同的隨機位元串，所以雙方必須先擁有共同的金鑰（或稱種子），才能帶入共通的 ARBG（Authentication Random Bitstream Generator）模組中產生相同的驗證隨機位元串。但是無線工作站與擷取點一開始並不認識對方，既無法事先分配金鑰，亦不知道互相的資訊，那麼要如何造出共同的金鑰呢？幸運地，我們可以使用 1976 年由 Diffie 與 Hellman 所發明的金鑰交換演算法（Diffie-Hellman）來解決這個問題。無線工作站與擷取點一開始把要產生共同

金鑰的相關資訊，加入試探要求訊框與試探回覆訊框中保留或為用到的位置，如此雙方就可以取得相同的金鑰，進而得出要驗證用的隨機位元串（圖 18）。

大致上，系統概觀就如上述。接下來我們在 4.2 節中對本研究的系統設計有更深入的介紹，依序為分析隨機位元串與額外資訊要插入訊框中的位置、Diffie-Hellman 金鑰演算法、認證隨機位元串產生器（ARBG）、修改 SND 為 DSN 過濾機制，與隨機位元串同步演算法。

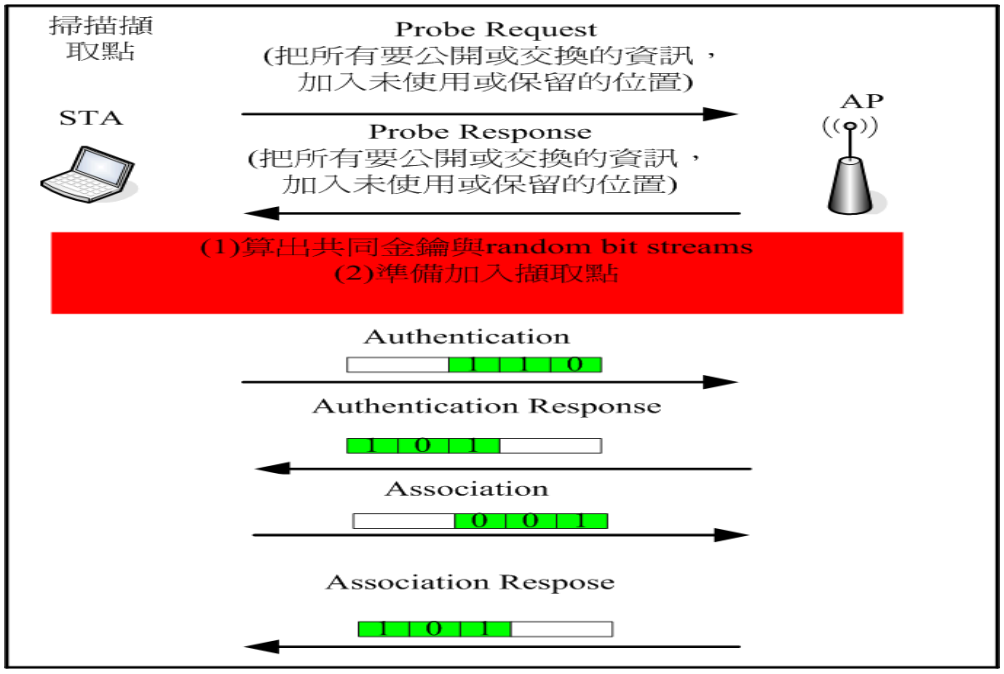


圖 18 本研究提出的上網流程方法一

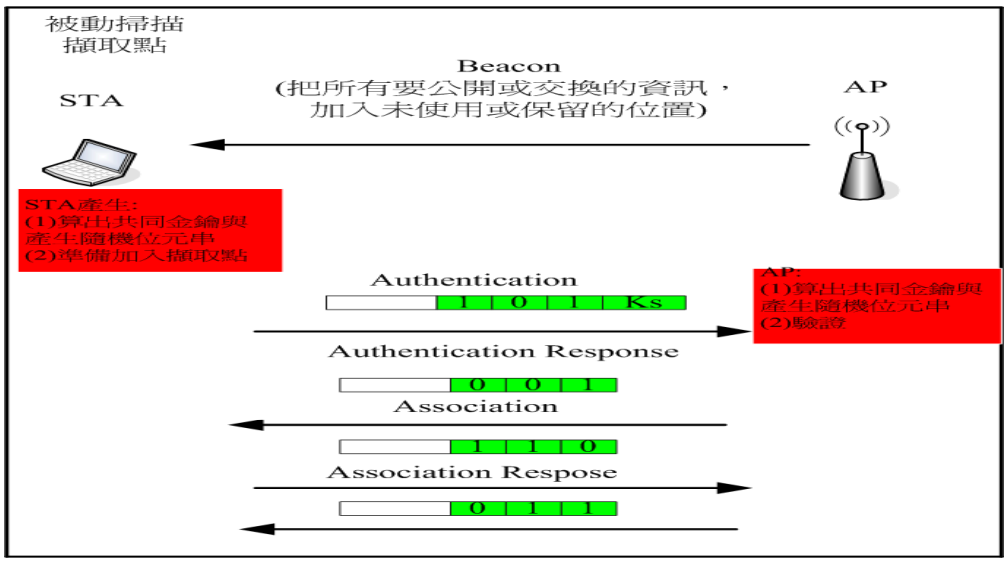


圖 19 本研究提出的上網流程方法二

4.2 系統設計

此小節將介紹將完成[1]作者研究中未說明部份，並修改有缺陷之處，底下列出五個部份：

（1）訊框分析不完整

（2）如何讓無線工作站與擷取點擁有共享金鑰

（3）相同隨機位元串要如何產生

（4）SND（Sequence Number Deviation）防禦機制的修改

（5）隨機位元串同步演算法

4.2.1. 訊框分析

我們必須加入一些額外的資訊於IEEE 802.11訊框中，例如：金鑰的交換、隨機位元串資訊等等，並且不能影響到原有的802.11協定運作，也就是設計的機制必須與802.11向後相容（Backward Compatible）。所以本研究必須利用訊框中沒有使用到或保留的位置，為其欄位定義出新的意義。

一般典型的802.11訊框格式如圖20所示，依據型態（Type）的不同可以分為管理訊框（Management Frame）、控制訊框（Control Frame）以及資料訊框（Data Frame）三種，而每種型態以子型態來定義所需的功能，如表5列出的管理訊框型態與子型態的對應功能表。不是每個訊框都包含圖20中的所有欄位，而是當有需求時才會包含某一欄位，因此我們可以利用這種特性加入額外資訊，但是卻不會影響系統運作。

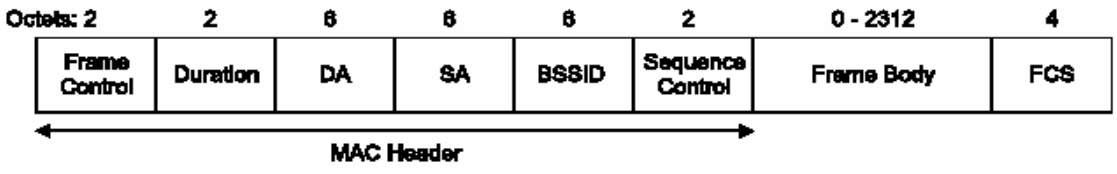


圖 20 一般 802.11 MAC 訊框格式

管理訊框（型態 = 00）	
子型態	功能
0000	連結要求
0001	連結回覆
0010	重新連結要求
0011	重新連結回應
0100	試探要求
0101	試探回覆
1000	訊標
1001	ATM
1010	連結終止
1011	身份認證
1100	身份認證終止

表 5 管理訊框型態與子型態功能對照

接下來我們將分析管理訊框中訊框表頭（Mac Header）與訊框主體（Frame Body）中有哪些位置是可以被我們使用的。

訊框表頭分析：

在[1][7][8][9][10][1]中都是將額外資訊加在訊框表頭中，不過在此處所受的限制很大，因為只有些許的位置是未使用到的。例如：如圖 21 所示，在[1]中將額外資訊加入訊框控制（Frame Control）中，訊框控制在型態為管理訊框的情形下，可能只有 B8、B9、B10、B12、B13、B15 這些位置是可以使用的。這些可用的位置數目對本研究所提的架構是不夠的，所以把額外的資訊加在訊框表頭中並不適用於我們的系統，底下列出加入額外資訊於此處的缺點：

（1）位置不夠（除了隨機位元串至少要三個位置，還有金鑰交換資訊等等）

（2）沒有使用相同格式管理，效率不好

（3）無彈性

所以本研究中的額外資訊都將加入訊框主體（Frame Body）中，如此則可以決上述（1）（2）（3）中的缺點，下一段將為大家說明。

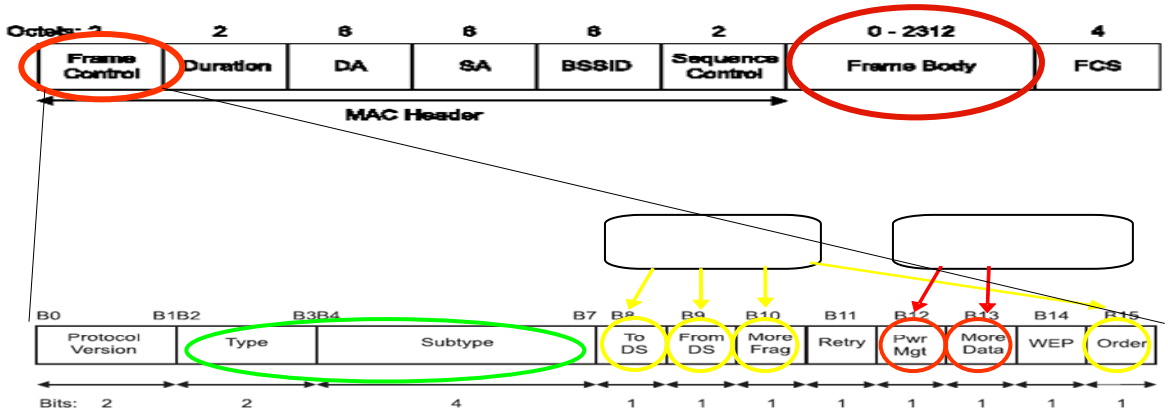


圖 21訊框控制中可以使用的位置分析[1]

訊框主體分析：

根據上述說明，我們知道把額外的資訊加在訊框表頭中並不恰當。所以接下來會把本研究架構所需的額外資訊都加在訊框主體中的，如此不僅可以解決位置不夠多的問題，亦因使用相同的格式來管理而使得效率提昇。管理訊框中的訊框主體定義了許多用來提供資訊的欄位，以便告知對方（擷取點或無線工作站）本身資訊的欄位。例如：訊標訊框中含有擷取點 SSID 的資訊項目等等。要告知對方自己本身相關資訊的欄位可以分為兩種，一種是固定欄位，另一種「資訊項目」欄位。固定欄位並不適合本研究（因為位置不夠），所以我們將使用「資訊項目」欄位，加入本研究所需額外資訊。

資訊項目欄位如表 6 所示，可以看到編號 7~15 與 32~255 都是屬於保留區段，所以我們可以使用這些保留的位置來定義出新的項目資訊。定義方式如圖 22 所示，管理訊框中所攜帶的資訊項目都使用相同的格式來表達，一個位元組的「項目編號」欄位，一個位元組的「長度」欄位，以及不固定長度的「項目表示內容」欄位。

項目編號	項目表示內容
0	服務組編號（SSID）
1	支援速率
2	FH 參數集
3	DS 參數集
4	CF 參數集
5	流量指示圖譜（TIM）
6	IBSS 參數集
7~15	保留未使用
16	挑戰字串
17~31	保留作為挑戰字串的延伸
32~255	保留未使用

表 6 管理訊框的項目內容

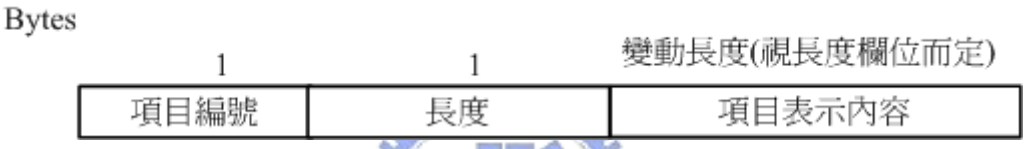


圖 22 項目資訊格式

至於要定義出哪些新的欄位呢？與我們需要在訊框中加入哪些額外資訊有關。根據我們提出的機制，需要定義的資訊項目有 Diffie-Hellam 金鑰交換（大質數 P、P 的原根 G、無線工作站產生的 K_s、擷取點產生的 K_a）、虛擬亂數產生器（a、b、m）、隨機位元串、回覆成功/失敗、計數器。各個新定義資訊項目的功能，會在本章各節中會有更進一步的介紹。

圖 23 無線工作站與擷取點為了利用 Diffie-Hellman 金鑰交換演算法得到共同金鑰，所需的大質數 P 與 P 的原根 G。

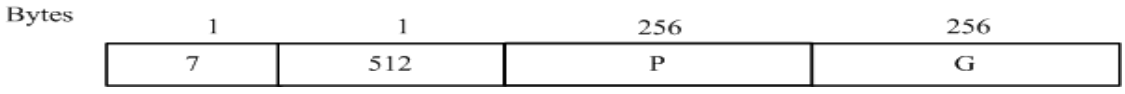


圖 23 金鑰交換所需大質數 P 與原根 G 資訊項目

圖 24 為雙方知道公開的 P 與 G 後，各自產生的一半金鑰資訊（Ka/Ks），加入試探要求與試探回覆（Probe Request/Response）中互相交換，即可算出同一把金鑰，細節於 4.2.2 討論

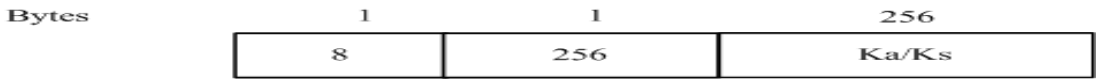


圖 24 擷取點與無線工作站各自產生一半的金鑰交換資訊項目

圖 25 是利用線性同餘（Linear Congruential）方式當虛擬亂數產生器時，所需的公開值，用於 4.2.3 中認證隨機位元串產生器中，細節將於 4.2.3 中討論。

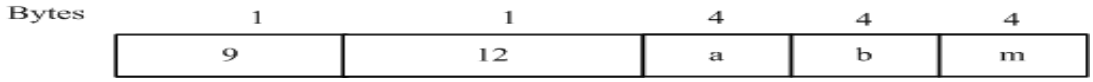


圖 25 虛擬亂數產生器的資訊項目

圖 26 為擷取點或無線工作站插入驗證的隨機位元串欄位，擷取點或無線工作站收到管理類型訊框後，會去檢驗這個欄位的值是否相符，之後才會有續動作。

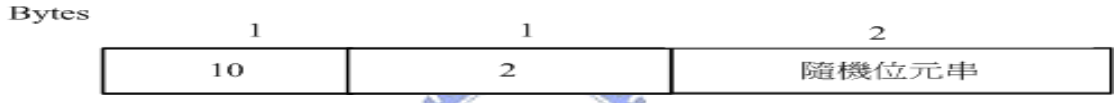


圖 26 加入驗證的隨機位元串資訊項目

圖 27 中的回覆成功或失敗欄位，是接收端用來回覆傳送端驗證的隨機位元串是否相符。若相符，則回覆成功；反之，則回覆失敗。

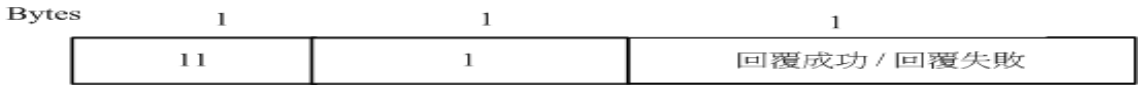


圖 27 同步演算法需要的回覆成功或失敗資訊項目

圖 28 的計數器欄位，是用調整擷取點與無線工作站的隨機位元串不同步的問題，細節於 4.2.5 討論。

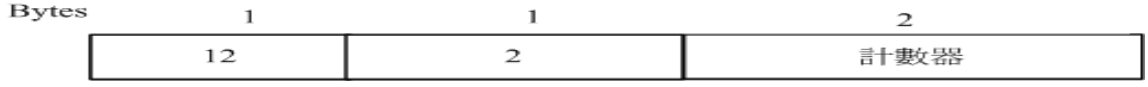


圖 28 同步演算法需要的計數器資訊項目

經過以上的分析之後，我們就可以利用新定義於訊框主體的新欄位加入我們需要的額外資訊於管理訊框中，以下各小節會有更詳盡的介紹本研究的機制是如何運作，包括金鑰交換、隨機位元串的產生、兩層過濾的機制、隨機位元串同步演算法。

4.2.2. 金鑰產生與交換

本研究中，因為擷取點與無線工作站需要擁有相同的隨機位元串，才能互相驗證對方是否為合法使用者，進而達到防止服務阻絕攻擊的攻擊。因此擷取點與無線工作站彼此會先透過某種機制而擁有相同的「金鑰」，之後再根據這把「金鑰」造出隨機位元串以便互相驗證溝通。在本研究提出的系統架構中，有以下四種可行的方式：

(1) 預先分配金鑰（Pre-Shared Key, PSK）：

「預先分配金鑰」是在要使用金鑰前就先分配好，通常使用在對稱式的加密演算法。例如：在 WiFi（IEEE 802.11b）中的 WEP 或 WPA 加密方式，都是採用事先分配金鑰的方式。不過，這種方式主要是應用在比較小型或是變化不大的環境下。

若在本研究中採用「預先分配金鑰」方式，則不需要執行交換金鑰資訊的動作，只要事先分配完金鑰後，擷取點與無線工作站就可以帶入 ARBG 中產生隨機位元串，進行連結上網所需的互相驗證工作。此外金鑰的組成不管在長度與字母的組合都必需要符合密碼學上安全（Cryptographically Secure）的金鑰，並且必須定時更換金鑰，否則將造成金鑰資訊容易被分析出來。

(2) Diffie-Hellman 金鑰交換演算法：

上述預先分配金鑰的方式缺點就是需要管理者手動方式輸入密碼，但是我們希望提出的系統架構能夠更有彈性。所以我們可以利用在第二章背景知識 2.4 節所描述的 Diffie-Hellman 金鑰交換演算法，讓彼此互相不認識的擷取點與無線工作站能夠擁有相同的金鑰。

如圖 29，無線工作站與擷取點只要把所需的金鑰產生資訊，加入試探要求與試探回應訊框的中保留位置（4.2.1 小節中新定義於訊框主體的欄位），彼此收到利用資訊後，就可以利用 Diffie-Hellman 金鑰交換演算法取得共同金鑰 K_{as} 之後，帶入 ARBG 中產生隨機位元串，進行連結上網所需的互相驗證工作。

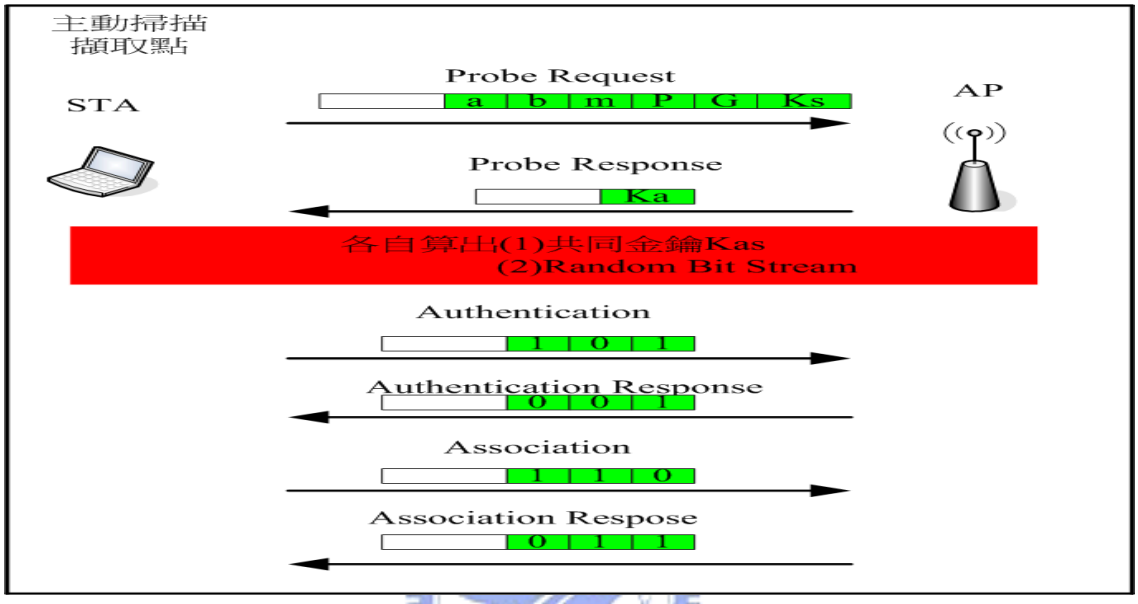


圖 29 本研究提出的主動金鑰交換流程

如圖 30，無線工作站把所需的金鑰產生資訊，加入訊標訊框的保留位置，無線工作站收到訊標訊框加入的額外資訊後，就可以產生金鑰 K_s，之後再利用 Diffie-Hellman 金鑰交換演算法算出 K_{as} 並代入 ARBG 中產生隨機位元串。接著無線工作站準備連結上網時，會加入隨機位元串與 K_s 於驗證訊框中（以便讓擷取點可以產生相同的金鑰）。擷取點收到 K_s 後會先產生 K_{as}，接著代入 ARBG 中產生隨機位元串後，才會進行驗證的工作。

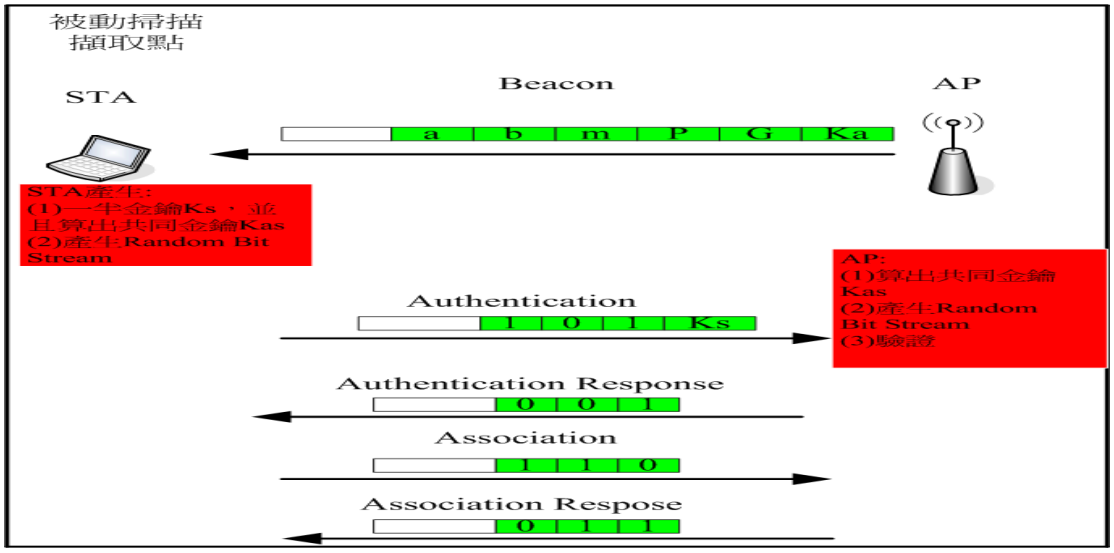


圖 30 本研究提出的被動金鑰交換流程

(3) IEEE 802.11i 交換金鑰：

802.11 TGi 為了改善 WEP 中使用同一把金鑰的問題，而提出了 TKIP 演算法，讓傳收端與接收端可以交換而得到暫時的成對金鑰（PTK），以避免長時間使用同一把金鑰的問題。金鑰產生與交換過程如圖 7 所示，當無線工作站與擷取點取得相同的 PTK 或 GTK 之後，就可以在每次會期溝通時帶入 ARBG 產生驗證的隨機位元串，彼此互相驗證。利用 802.11i 產生金鑰的方式，在安全的考量上的確比較理想。但是有一點要注意的是，因為 802.1x 認證程序也有類似 802.11 容易遭受 DoS 攻擊問題（如：EAPOL-Failure），所以必須等到認證過後才能獲取金鑰的話，中間可能會遭受攻擊者利用 EAPOL-Failure 攻擊，造成使用者不能順利通過認證。

此外，現行的無線工作站計算能力還不足已達到 802.11i 金鑰交換程序，因為其金鑰產生的函數過程太複雜，所以並不適用於無線裝置上。再者電力消耗也是一大考量問題。所以本研究在現階段並不會採取 802.11i 交換金鑰的方式，不過當軟體與硬體的升級支援之後，採取此方式是可行的。

(4) PKI（Public Key Infrastructure）：

擷取點與無線工作站亦可在在 PKI 的架構下交換金鑰。擷取點可以先向 CA（Certification Authority）取得無線工作站的公開金鑰（Public Key），把雙方要產生隨機位元串的金鑰利用無線工作站的公開金鑰加密後，送給無線工作站。無線工作站收到後，就可以利用自己的私密金鑰（Private Key）解開，如此雙方就擁有可以產生相同隨機位元串的共同金鑰。雖然 PKI 是可行的方式，不過建立 CA 的成本太高，再者無線工作站的運算能力與是否有憑證都是需要考量之處。所以，本研究在現階段並不會採取 PKI 交換金鑰的方式，不過當建立 CA 的成本降低，並且由可信任的第三者執行的話，則亦為可行的方式。

在上述四個所提的金鑰交換可行方式中，（3）與（4）的方式雖然可行，但是需要軟體與硬體上的升級支援才可以。所以在現階段中，本研究將會以（1）或（2）的方式來實作，以便讓無線工作站與擷取點可以擁有共同金鑰。在雙方擁有了共同的金鑰之後，會基於這把金鑰下造出會期金鑰，加入我們下一節所提的 ARBG 模組，以便讓無線工作站與擷取點產生相同的驗證隨機位元串。

4.2.3. 驗證隨機位元串產生器（ARBG, Authentication Random Bitstream Generator）

無線工作站與擷取點利用上節 4.2.2 所討論的四種方式之一取得共同金鑰後，會將金鑰帶入雙方共有的 ARBG 模組（Module）以便輸出驗證的隨機位元串，如此擷取點與無線工作站才可以彼此互相驗證。那 ARBG 模組要如何實現呢？實作的方式可以使用硬體或軟體兩種方式來實現。

硬體實現：

ARBG可以使用LFSR（Linear Feedback Shift Registers）的方式來產生。如圖31所示，利用正反器（Flip-Flop）與Feedback Function（例如：XOR）組成的方

式，輸出隨機位元串。理論上長度為n的LFSR，最多可以產生2^{n-1}個位元長度的隨機位元串（最大週期，也就是需要2^{n-1} 才會重複）。若想要增加安全強度，則可以利用多個LFSRs一起帶入函式F(x)做運算，進而得到更大的週期長度（如圖32）。LFSR的優點就是速度快，缺點就是硬體花費比較高且需廠商的支援。

在本研究因為受到裝置的限制，所以在實現ARBG模組時會以軟體方式為主。未來若有無線網卡廠商支援本研究的協定，則可以把ARBG加入硬體中，效能應會比軟體好，所以在此先提出硬體實作的概念。

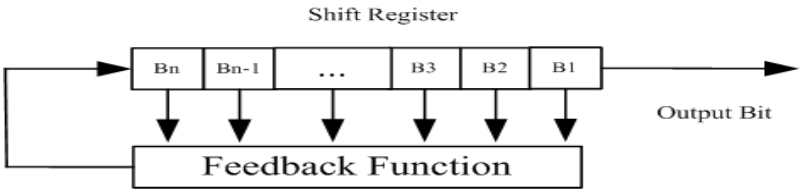


圖 31 Feedback shift register

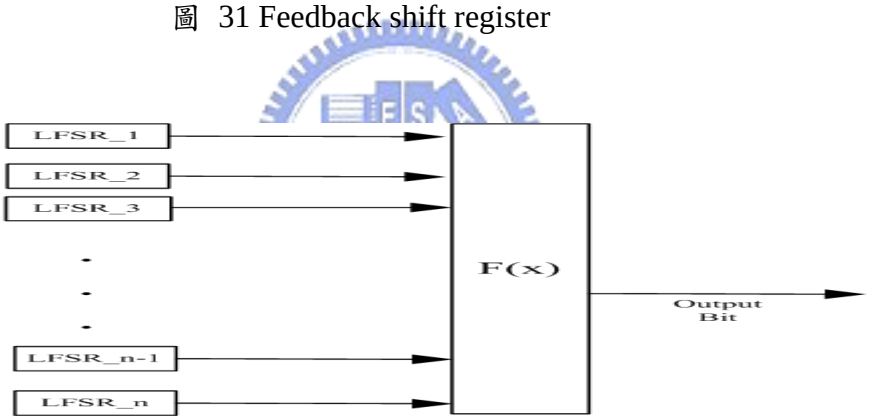


圖 32 多個 Feedback shift register

軟體實現：

用軟體來產生隨機位元串的方式很多，但是必須考量到安全與效能。因為在無線網路的環境下，因為無線裝置的能力有限，所以不能花費太多時間在計算量上，否則可能會造成另一種阻絕服務攻擊。所以在軟體實現上，一定要設計一個在安全與效能兩個因素上，都能達到一定效果的隨機位元串生器。

在此我們應用第三章 3.4 節中[11]所提出的概念(圖 16)來實作 ARBG 模組。

本研究一開始的想法如圖 33 所示，利用會期金鑰帶入 RC4 演算法來造出驗證的隨機位元串，但是會期金鑰要如何產生?本研究利用[11]，David L. Pepyne 所提出的 SPRiNG 協定中的 PRNG（Pseudo Random Number Generator）概念，來當作我們會期金鑰產生的方式。[11]中的 PRNG 是為了改善 WEP 中重複使用相同金鑰與 IV（初始向量）的問題（見第三章 3.4.2 與圖 15）。而在本研究中亦是為了避免每次產生相同的隨機位元串問題（因為金鑰會重複使用，如 PSK 架構），所以加入 PRNG 的概念讓 ARBG 可以動地產生每個會期時所需的隨機位元串，如此不僅會期金鑰產生速度快，也可以動態地改變金鑰值。

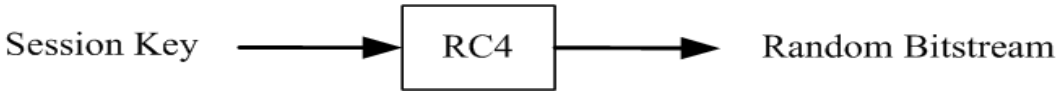


圖 33 ARBG 原始想法

本系統的 ARBG 模組如圖 34，欲驗證的雙方都擁有相同的這個 ARBG 模組，運作方式可以分為三個階段：

（1）一開始時將雙方共同金鑰帶入 PRNG

（2）PRNG 的輸出當作演算法（Algorithm）的輸入，以輸出隨機位元串

（3）將認證的隨機位元串分成數個區段，擷取區段後才當成驗證的隨機位元串

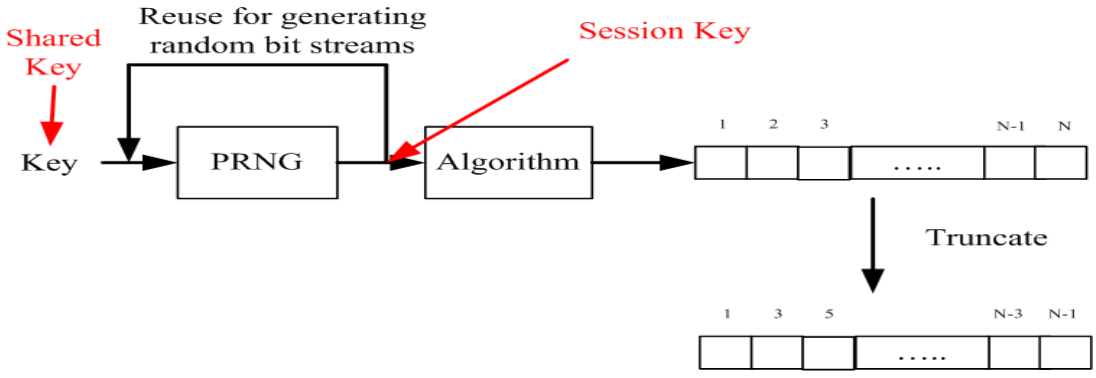


圖 34 ARBG 模組架構

Increase the complexity：$2^{\left\lfloor \frac{N-1}{2} \right\rfloor}$

底下分別說明圖 34 中，PRNG、ALGO 與 Truncate 的用途：

（1）PRNG：

PRNG在ARBG模組的功能與WEP中的初始向量（Initial Vector, IV）相同，WEP使用IV的目的就是要避金鑰重複使用（密碼學上最基本的安全條件就是不能長時間使用相同的金鑰加密）。一般來說IV越長，安全度越高（WEP的IV只有24bit太短），因此我們利用PRNG來替代初始向量的這個概念，以便提供週期更長且動態地變更輸入Algorithm之值，如此每次會期需要的隨機位元串就會不相同（攻擊者也就難以分析），優點如下：

1. 若無線工作站與擷取點是預先分配金鑰，雙方可以先溝通好下一次要用的種子，如此就可利用PRNG產生每次會期的隨機位元串，而不會每次都使用相同的隨機位元串（因為金鑰是固定且預先分配的）。

2. 若為其他金鑰交換方式，亦可有效且動態地變更輸入Algorithm中之值，以便讓每次會期的隨機位元串都不相同（無線工作站與擷取點會先溝通好下一次要用的種子）。

此外，使用PRNG可能會有兩個讓無線工作站與擷取點發生不同步的情況，不過都可以輕易恢復：

1. 當隨機位元串用完時，無線工作站或擷取點會先利用共同金鑰加密現在要輸入到PRNG的種子為何，加密完後傳送給對方。如此雙方就會恢復同步，使用相同的隨機位元串。

2. 結束目前的會期時，無線工作站與擷取點會於結束之前先溝通好下一次會期要用的PRNG輸入種子，將其加密後傳送給對方。

由上述可知PRNG可以減少金鑰交換的額外負擔（如：Diffie-Hellman），只要先溝通好下一次會期要用的PRNG輸入種子為何，則下一次會期就不需要執行金鑰交換的動作，並且可以產生與上次不同的驗證隨機位元串。

本研究以線性同餘的方式實作 PRNG（見 3.4.1 小節），亦可使用其他較安全的亂數演算法實作，例如：Blum Blum Shub。但是若選用其他的安全性較高的

PRNG 時，必須要注意效能問題（因為效能耗費太高會造成另一種阻絕服務攻擊）與一般的無線裝置是否能力能夠支援複雜的運算。

（2）Algorithm：

Algorithm在ARBG模組的作用是產生驗證的隨機位元串，其選用亦可依需求而有所不同，但亦必須考慮效能與安全問題。本研究使用RC4來當作Algorithm，將會期金鑰輸入RC4中，輸出的1024個位元後，需做擷取的動作才能當作驗證的隨機位元串。選擇以RC4的來產生隨機位元串的原因如底下說明：

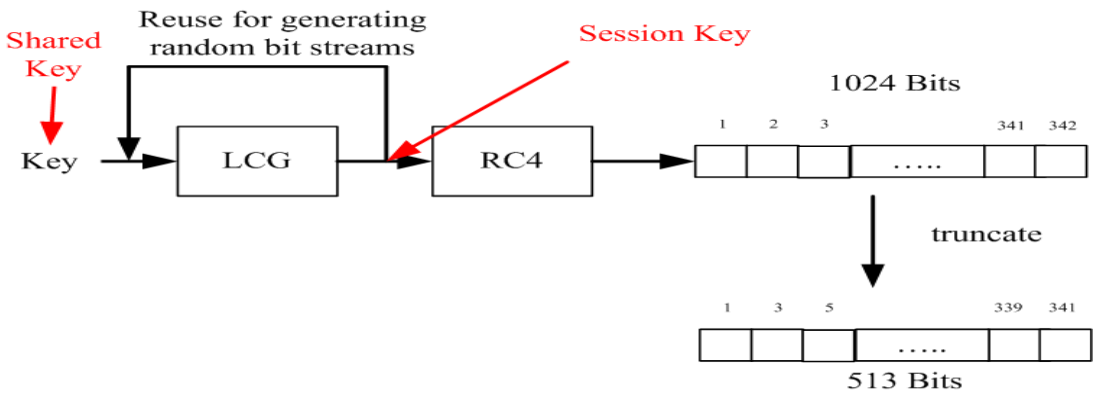
A.演算速度快，對硬體或軟體的要求很低：RC4 產生串流位元的速度很快，並且適合用於軟體上，所以一般的無線裝置都可以負擔其運算複雜度。

B.安全性足夠，且可與效率達到平衡：根據[11] Pepyne, D,L 與[22]的分析，RC4不管在安全性或效能上，都可以達到一定水準。

（3）擷取（Truncate）

擷取的作用是為了讓攻擊者無法收集到完整的隨機位元串資訊。如圖34所示，我們把輸出的隨機位元串分成N區段，但是我們不全部拿來使用，從中擷取幾段作為我們的認證隨機位元串，如此攻擊者將無法收集到完整資訊，亦就無法有效分析。

如圖35，RC4輸出1024個位元，我們把它分成342個區段，每段含三個位元（根據[1] 的實驗結果，利用SND與隨機位元串的機制，只需要三個就可以達到防禦阻絕服務攻擊效果），第342個除外。我們只選用1、3、5、…、349、341的區段，則攻擊者無法蒐集到2、4、6、…、340、342區段，也就是原本1024個位元，我們只取513個位元，則分析的複雜度多了2^{511}。攻擊者收集不到足夠的資訊，所以可以保證安全強度上是足夠的。



Increase the complexity： 2^{511}

圖 35 ARBG 實作細節項目

4.2.4. 結合變動順序控制（Dynamic Sequence Number, DSN）與隨機位元串之防禦阻絕服務攻擊機制

第三章相關研究中介紹過[1]利用兩個階段模式來防禦阻絕服務攻擊：

（1）SND（Sequence Number Deviation）

（2）隨機位元串

第一階段過濾差距太小的SND值，作者假設攻擊者發動阻絕服務攻擊是連續的發送Deauthentication/Disassociation訊框，因為正常使用者不會一直連續發送這類的訊框，所以SND差距太近為不正常現象。第一階段通過後，才會到第二階段隨機位元串比對檢查，否則會直接丟棄該訊框。根據[1]實驗結果使用3個隨機位元串與SND=24時，可以有效的防禦DoS攻擊；而使用4個隨機位元串與SND=12，亦有相同效果。

但是遺憾地，[1]中使用的SND並不能有效地達到防禦效果。因為攻擊者只要更改Deauthentication/Disassociation訊框中的SN值，就可以輕易躲過第一層過濾，直接進入第二階段，底下列出使用SND的缺點：

（1）攻擊者可以輕易改變攻擊模式，就可躲過第一階段過濾（例如：更改訊框中的SN值，使Deauthentication/Disassociation訊框中的SN值不照順序發送）。

（2）因為需要佇列技巧，所以會有延遲（Delay）現象。再者系統的SND值一定

不會設的太大（因為設太大延遲會更嚴重），攻擊者可以利用這個弱點把發送Deauthentication/ Disassociation訊框的差距拉大就可以繞過。此外利用此法也會造成系統空間資源的浪費。

（3）若攻擊模式為DDoS的話，多個攻擊者針對特定使用者一起發動攻擊的話，每個攻擊者只要不斷地發送Deauthentication/Disassociation訊框，就可以繞過第一階段。

為了改善上述缺點，所以本研究提出的防禦機制為：

（1）DSN（Dynamic Sequence Number）

（2）隨機位元串

將第一階段的過濾方式更改成 DSN（原本為 SND），第二階段的過濾方式則保持[1]中的方式。系統流程如圖 36 所示，一開始無線工作站使用隨機位元串與擷取點互相驗證，連結上網。若其中一方接收到 Deauthentiatiom/Disassociation 訊框，則會進入 Defense Mechanism，進行 DSN 與 RBS 過濾。

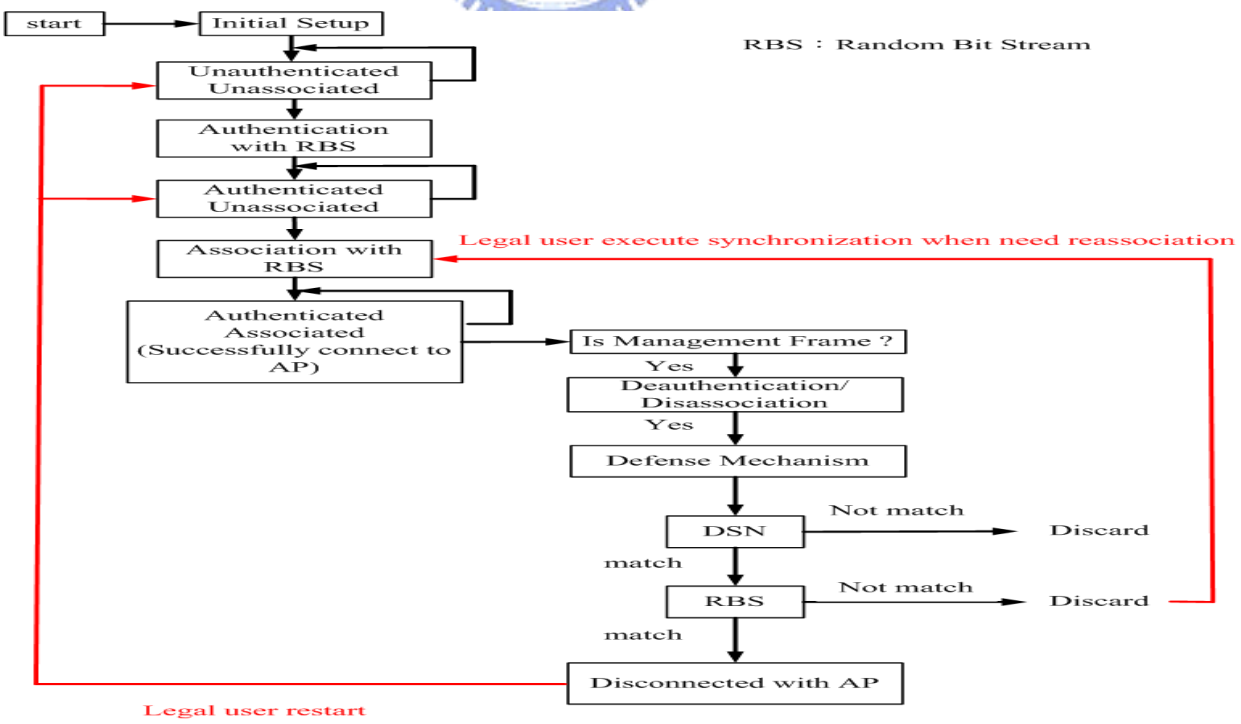


圖 36 本研究提出的防禦機制流程

DSN 的概念就是當傳送端要發送 Deauthntication 或 Disassociation 訊框時，其中的 SN 值是只有傳送端與接收端知道的，而不是依照順序來傳送。若 SN 相符才進入第二階段比對隨機位元串，否則丟棄該訊框。那要如何讓變動的 SN 值只有無線工作站與擷取點知道呢?在此我們利用到一次通行密碼（One-time password， OTP）中的技巧，即可變動的產生 SN 值。

本研究利用 OTP 中時間同步（time-synchronized）的方式來實作 DSN 機制。時間同步是指使用者端與伺服器端在進行登入的同時，需利用「系統的時間」與「雙方共同持有的種子」，計算一次性的密碼進行登入。所以本系統的 DSN 模組可以分成以下兩個步驟：

（1）時序同步：

如圖37，我們可以使用IEEE 802.11管理訊框中訊框主體裡定義的時間標籤（Timestamp）欄位，來讓擷取點與無線工作站達到時序同步（原用於省電模式）。時序同步的運作在基礎架構下，基本上以擷取點為集中的協調中心。每一個無線工作站都有一個時序同步功能（Timing Synchronization Function，TSF）的計時器，此計時器每震盪一次為1us（即每秒10^6）。同步運作方式是由擷取點將TSF的時間擺在時間標籤欄位，再利用訊標訊框或探測回覆訊框送出去給無線工作站，無線工作站收到後取出時間標籤欄位的TSF時間後，再加上一小段Offset（擷取點發送到無線工作站之間的時間，大約需多少us），即可與擷取點同步（如圖38）。如此產生變動SN值所需的「共同金鑰（如同上述的種子）」與「相同時間」就都具備了。

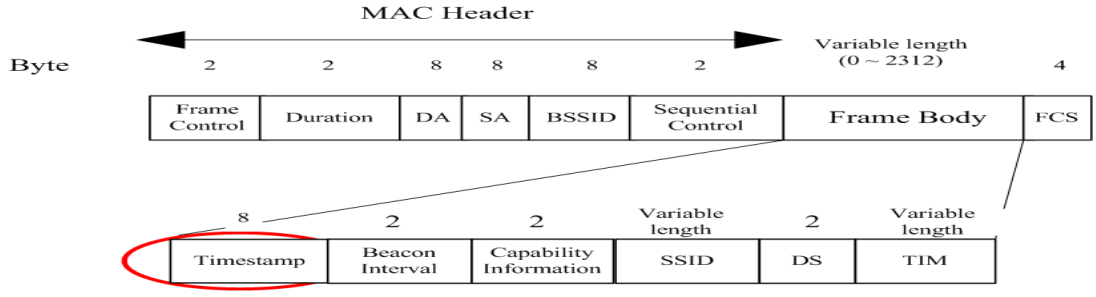


圖 37 訊標訊框中 Timestamp 欄位

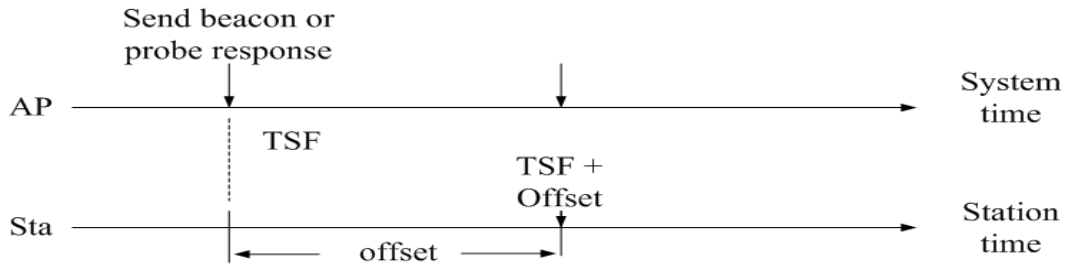


圖 38 時序同步方式

（2）算出所需的變動 SN:

如圖 39，當傳送端要發送 Deauthentication/Disassociation 訊框時，必須利用雙方擁有的相同時序與金鑰帶入所選用的演算法，算出要當作 Deauthentication/Disassociation 訊框的變動 SN 值。接收端收到後，會去比對 SN 是否相符，相符則會通過第一階段過濾（進到第二階段比對隨機位元串），不相符則不理會並且丟棄該訊框。

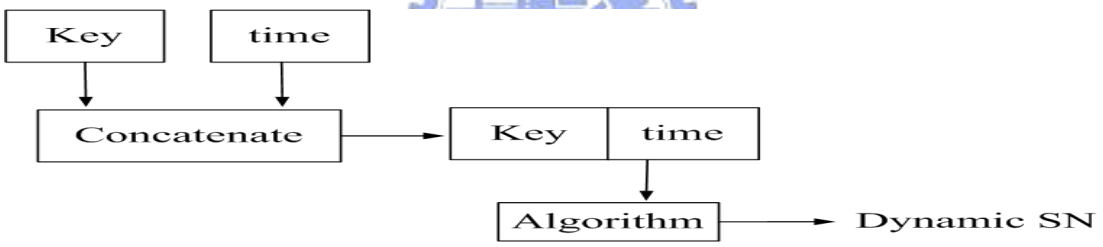


圖 39 變動 SN 產生方式

使用 DSN 的好處如下：

（1）第一階段有效防止攻擊者進入第二階段，因為原本[1]的 SND 設計，攻擊者可以輕易躲過第一階段。 若攻擊者可輕易到第二階段，則可能會使用暴力法讓系統的隨機位元串一直處於不同步的狀況，而因此浪費了驗證的隨機位元串。

（2）不需使用停佇封包，所以不會有延遲現象，並且不會浪費系統空間資源。

（3）安全性高，攻擊者無法在短時間猜出每次變動的 SN 值。

以上就是修改 SND 為 DSN 的加強防禦機制，我們只修改第一階段的防禦機制，第二階段則保持與[1]相同的運作模式。我們會在第五章證明經由修改 SND 變為

DSN 後，證明防禦效果大大地提升，並且不會帶給系統額外的負擔。

4.2.5. 隨機位元串同步演算法

無線工作站與擷取點依據相同的隨機位元串，來驗證對方是否為合法的使者，但是雙方必須要指到相同的隨機位元串區段上，彼此互相送出去的隨機位元串才會相符（match），這也就是所謂的同步問題。

同步演算法的設計會對我們所提出的機制有很大的影響，在第三章3.1.1節中我們介紹過[1][7][8][9]提出的同步演算法，但是其效能有很大的問題。因為此法最多需要NSI次才能恢復同步（在最壞的情況下需要NSI次）。在無線網路的環境下，封包遭到干擾或遺失的機率很大，所以可能會常發生不同步的狀況。若同步演算法每次都可能要NSI次才可以恢復同步的話，是很沒有效率。再者攻擊者很可能利用這個的缺陷，造成系統一直在做同步演算法，進而變成另一種阻絕服務攻擊。

本研究所提的演算法，將利用管理訊框中訊框主體的保留欄位，定義一個計數器（Counter）欄位，此計數器指向現在隨機位元串所使用到的位置，因此傳送端與接收端不同步的情形，只需要一次就可以恢復同步，讓整體的效率大大提升。演算法的設計如表7、8所示，無線工作站與擷取點各自擁有一個獨立的計數器，用來記錄現階段所使用的隨機位元串所指到的區段位置（Index）。當傳送端（通常為無線工作站）送封包給接收端（通常為擷取點）時，不管傳送端送過來的隨機位元串是否相符，接收端的計數器都會加一。若隨機位元串相符，則接收端會回傳回覆成功（Response_Success）與驗證的隨機位元串（以便驗證來源性）給傳送端；反之，若隨機位元串不相符，接收端則會回傳正確的計數器位置、驗證的隨機位元串與回覆失敗（Resonse_Failure）給傳送端。

```
/* RBS (Random Bit Stream) */

/* Receiver receives the Request from Sender */
if ( Receiver_random_bit_stream ==
Sender_random_bit_stream )
{
    Receiver_Counter++;
    Reply station with "RBS and Response_Success";
    Receiver_Counter++
}
else
{
    Receiver_Counter++;

    /*  Receiver回傳RBS可以驗證是否爲真的
      AP回傳的Response_Failure  i.e表明來源性  */
     Reply station with  "Receiver_Counter +1 , RBS
     , and Response_Failure"  ;

     Receiver_Counter++;
}
```

表 7 本研究提出的同步演算法-接收端（擷取點）

```
/* Sender receives the Response from Receiver */
if ( Response  ==  "Response_Success" )
{
   Sender_Counter++;
   if ( Receiver_random_bit_stream  ==  Sender_random_bit_stream )
   {
      Sender_Counter++;
      Continue to Execute the Next Process; //繼續執行下一個步驟
   }
}

 else if ( Response  ==  "Response_Failure" )
{
  /*  Sender驗證Receiver的RBS   i.e. 驗證來源性  */
  if( Receiver_random_bit_stream[Receiver_counter-1] ==
       Sender_random_bit_stream[Receiver_counter-1]  )
  {
      Sender_Counter = Receiver_Counter;
      Continue to Execute the Next Process;
  }
  else
  {
     Discard the Response_Failure;
     Continue to Execute the Next Process;
  }
```

表 8 本研究提出的同步演算法-傳送端（無線工作站）

根據本演算法的特性只有在回覆遺失的情況或遭受攻擊的時候才會發生不同步的情形，圖 40、41 與 42 分別說明本演算法皆可在任何情況下以 O（1）時間輕鬆恢復同步（不像[1][7][8][9]提出的同步演算法會受到 0 與 1 位元位置交錯而有影響）。圖 41 為無線工作站端傳送要求訊框遺失之同步情形。圖 42 為擷取點回覆訊框遺失之同步情形。圖 43 則是無線工作站與擷取點在不同步的情形下（可能因遭受攻擊所造成），恢復同步的過程。表 9 為本研究與[1][7][8][9]的同步演算法在不同情況下的效能比較表。

	本研究提出的同步演算法	[3][4][5][6]提出的同步演算法
Best Case	O(1)	O(1)
Worse Case	O(1)	O(N)
Average Case	O(1)	O(N)

表 9 同步演算法比較圖

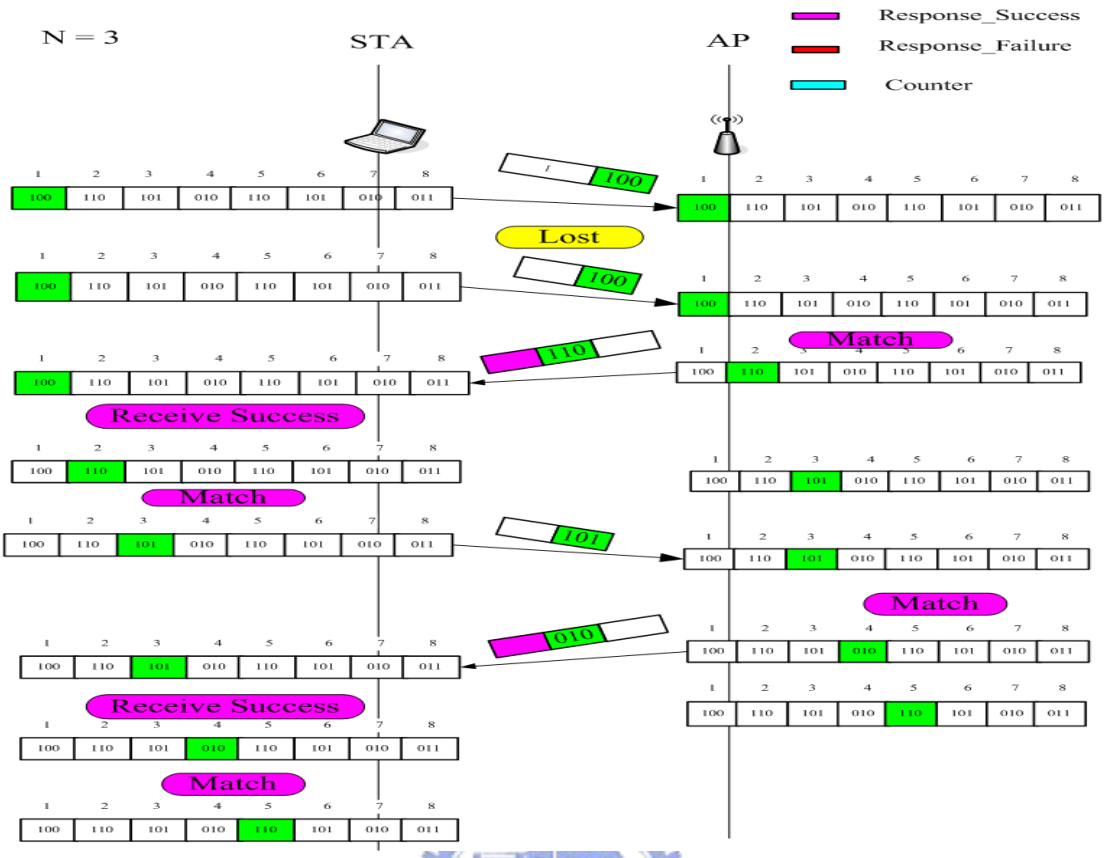


圖 40 無線工作站要求訊框遺失之同步演算法過程

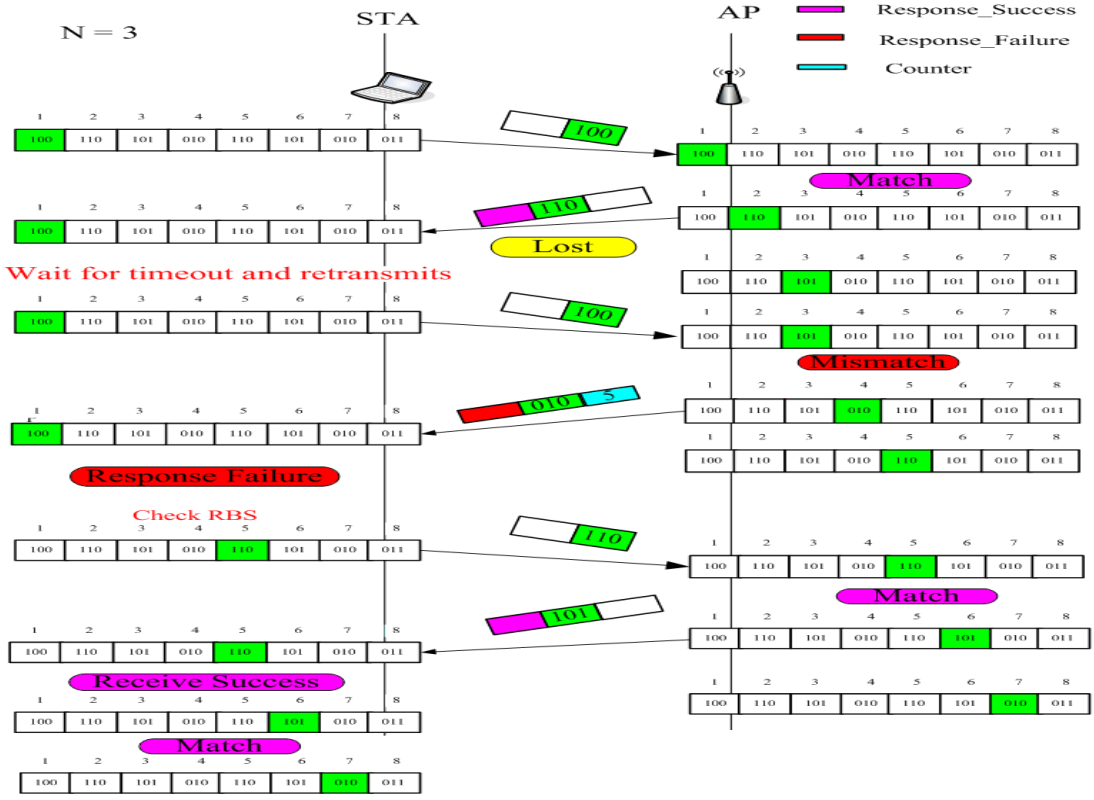


圖 41 擷取點回覆遺失之同步演算法過程

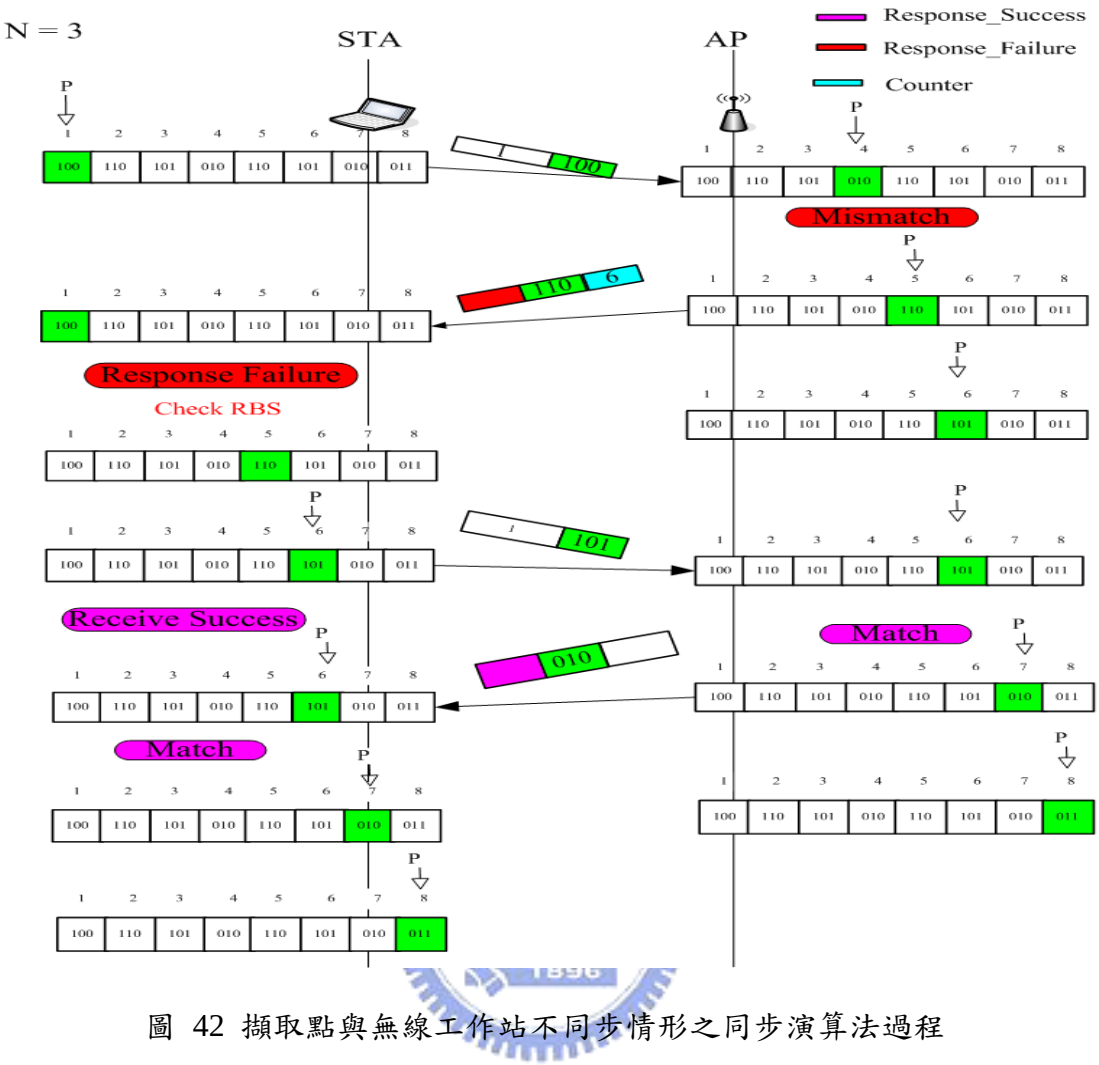


圖 42 擷取點與無線工作站不同步情形之同步演算法過程

第五章 分析與實驗結果

在此章節中，我們將利用實驗來證明本研究提出的系統架構是可行且有效率。5.1 節介紹實驗環境，採用網路模擬器的方式進行實驗測試。5.2 節套入本研究提出的 Diffie-Hellman 金鑰交換、ARBG 與隨機位元串同步演算法於[1]的兩層「SND 與隨機位元串」防禦中，測試系統實際的效能。5.3 節測試本論文改進的「DSN 與隨機位元串」防禦效能，並與上小節 5.3 中[1]提出的「SND 與隨機位元串」互相比較，以證實本研究不僅在安全性上大幅改進，並且未增加系統負擔。5.4 節利用統計模型從「失敗的驗證訊框數目」，察覺是否有潛在的攻擊或非法使用者，提供網管者一個參考依據，看是否要更進一步採取其他防禦行動。

5.1 實驗環境

本研究研究環境如表 10，採用 NCTUns 3.0 網路模擬器來模擬本論文提出的系統架構。NCTUns 3.0 是由交通大學所研發的一套網路模擬軟體，為目前世界上常用的網路模擬器之一，其採用 C++開發，大約寫了十三萬行的程式，擁有非常高的整合性、友善的介面、直覺性、靈活和可擴充性，且提供 GUI 介面讓使用者可於短時間內設定好模擬的網路架構，模擬後的情形亦可以圖形化顯示出來，讓使用者容易分析及清楚的看到實驗結果與情形。

此外相關的密碼方法採用 Crypto++ 5.5.1，採用 C++標準寫成自由軟體，提供專業的密碼學函式庫，例如：區段加密（block cipher）、串流加密（stream cipher）、公開金鑰加密、雜湊函式…等等功能。1995 年 6 月發佈了 1.0 版本，目前的最高版本為 2007 年 5 月所發佈的 Crypto++ 5.5.1。

CPU	Intel Pentium 1.7GHz
RAM	256M bytes
OS	Red Hat Linux 9.0 (Kernel 2.6.11-nctuns-20060503)
Simulator	NCTUNS 3.0
Cryptography	Crypto++ 5.5
Diffie-hellman	rfc2875 , draft-ietf-ipsec-skip-06

表 10 模擬環境

實驗模擬的網路架構如圖 43 所示，有兩台無線工作站，一台為正常使用者（ID 為 3），另外一台為攻擊者（ID 為 4），以及一個擷取點（ID 為 2）與擷取點連結的有線端電腦（ID 為 1）。測試方式為正常使用者利用 FTP 的方式傳送一測試檔案給有線網路端，檔案大小為 30,106,820 Bytes，攻擊者發動大約十秒的 Deauthentication/Disassociation 氾濫攻擊，分為兩種攻擊模式：

（1）SN 值為連續(sequent)增加的方式不斷地發送。

（2）SN 值為跳躍(Jump)的方式不斷地發送，例如：發一個 Deauth/ Disassoc 訊框後，中間發數個 association/authentication 訊框後，才繼續發下一個 Deauth/ Disassoc 訊框，i.e. SND>=10 或 SND>=30，但 SND 不超過 100）

我們將分析各種防禦機制的延遲時間並且互相比較（各項實驗測試十次，比較平均值），最後證明本研究是可行、有效率且安全度較高的。

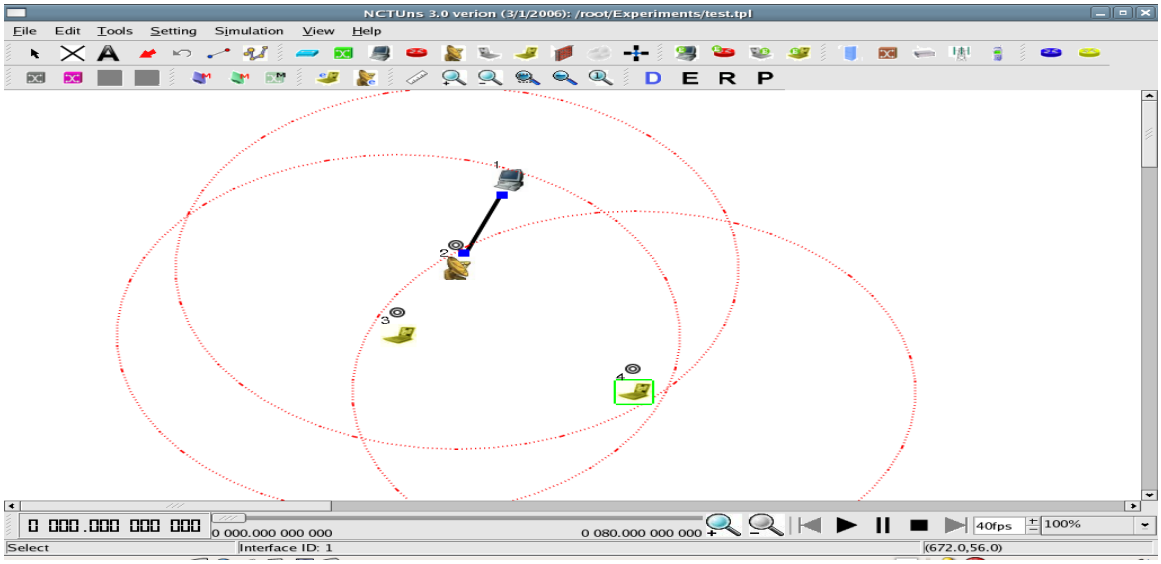


圖 43 模擬網路架構圖

5.2[1]提出的 SND（Sequence Number Deviation）與隨機位元串兩層過濾機制效能

在此小節中，我們會套入本研究提出的 Diffie-Hellman 金鑰交換、ARBG 與隨機位元串同步演算法於[1]的「SND 與隨機位元串」兩層防禦中（先測試只單獨使用隨機位元串與 SND 的效能如何，之後再測試合併為兩層過濾的機制），以測試系統實際的效能，之後與下一節本論文提出的「DSN 與隨機位元串」兩層防禦互相比較。

連結上網所需時間測試：

首先，先測試本研究提出的金鑰交換、隨機位元串產生（ARBG）、身份認證與連結各階段所花的時間。從表 11 可以看出各個階段都只需些許的時間，而加總的時間也只有約 0.022 秒，所以對系統的負擔是很輕的（Lightweight）。此外從表中可以看出，各階段以 Diffie-Hellman 金鑰交換所需的時間最多。若無線工作站與擷取點已經擁有同金鑰，則下一次會期就不需使用 Diffie-Hellman 金鑰交換，可直接利用 ARBG 中的 PRNG 就可產生這次會期需要的隨機位元串（只有第一次連結需要執行金鑰交換動作，見 4.2.3 節），如此就可以減少更多時間，更符合負擔輕（Lightweight）的精神（如表 12）。

	Diffie-Hellman (1024 bits)	ARBG	Authentication and Association	Total
時間（秒）	0.0187937	0.002283	0.0009222	0.0219989

表 11 金鑰交換、ARBG、身份認證與連結所需時間

表 12 顯示從無線工作站要使用網路前，分別掃描各個頻道的擷取點到連結上至某一擷取點所需的時間。可以從表中看出，本研究提出的架構大約只比原一般上網連結的方式多出約 0.026 秒。若不需 Diffie-Hellman 金鑰交換，則需要的時間就更少了，只比原一般上網連結的方式多出約 0.006 秒。

次數	1	2	3	4	5	6	7	8	9	10	平均	平均值差
一般連結上網	0.6512211	0.6531321	0.6513281	0.6522189	0.6527214	0.6512627	0.6512283	0.6509989	0.6518343	0.6512234	0.65171692	
本研究連結上網（包含金鑰交換 1024bits 與隨機位元串產生）	0.6737902	0.6795717	0.6811002	0.6786998	0.6785922	0.6791642	0.6732912	0.6774093	0.6798629	0.6805824	0.67820641	0.02648949
本研究連結上網（省略金鑰交換步驟）	0.6592133	0.6592133	0.6587343	0.6601121	0.6601209	0.6587024	0.6571324	0.6554122	0.6561347	0.6561143	0.65808899	0.00637207

表 12 一般連結上網與使用本研究連結上網時間比較

一般傳送時間與失敗攻擊的頻寬消耗：

表 13 顯示在一般與攻擊皆失敗兩種情況下，所消耗的頻寬的傳送時間，我們後續研究將以此表的時間作為各項測試的基準，以便比較各種防禦的效能。

次數	1	2	3	4	5	6	7	8	9	10	平均值	平均值差
一般傳送時間	45.23	46.38	45.96	45.89	46.34	46.57	46.12	45.94	45.72	45.81	45.996	0
Deauth 攻擊皆失敗的傳送時間	49.83	49.62	49.74	48.97	49.18	49.85	48.93	49.21	48.88	49.31	49.352	3.356
Disassoc 攻擊皆失敗的傳送時間	49.13	49.27	48.89	49.54	49.83	48.74	49.32	49.33	49.37	48.99	49.241	3.245

表 13 正常情況與皆攻擊失敗的傳送時間比較

隨機位元串過濾防禦機制測試：

測試只有使用隨機位元串一層過濾的情況下，至少需要多少個隨機位元串數才可以達到防禦的效果。測試環境如同 5.1 節所述，攻擊者以 SN 值連續增加的方式不斷地發送，其餘不變。表 14 可以看出隨機位元串大約需要 10 個以上才會達到令人滿意的效果，而可看出 Deauthentication 攻擊會比 Disassociation 攻擊來的嚴重，因為需要多一個階段才能恢復連結（見圖 5）。此處的結果與[1]有些許的不同（[1]測試出來的結果為 8 個以上，可能原因為[1]使用機率的方式實驗），

但此處在一次證實隨機位元串的確可以用於防禦阻絕服務攻擊。此外更重要的是，從這裡可以看出本研究的隨機位元串同步演算法的效能表現的相當好，可以很快的恢復同步而不會增加系統太多時間。

隨機位元串數	0	1	2	3	4	5	6	7	8	9	10	11
Deauth 氾濫攻擊	70.63	70.01	69.71	68.97	67.16	65.93	63.29	61.08	58.73	51.89	49.22	48.03
Deauth 延遲時間（與一般時間比較）	24.634	24.014	23.714	22.974	21.164	19.934	17.294	15.084	12.734	5.894	3.224	2.034
Deauth 延遲時間（與攻擊皆失敗比較）	21.278	20.658	20.358	19.618	17.808	16.578	13.938	11.728	9.378	2.538	-0.132	-1.322
隨機位元串數	0	1	2	3	4	5	6	7	8	9	10	11
Disassoc 氾濫攻擊	68.29	68.19	67.89	67.31	65.82	63.21	60.83	58.29	57.17	49.36	47.32	47.12
Disassoc 延遲時間（與一般時間比較）	22.294	22.194	21.894	21.314	19.824	17.214	14.834	12.294	11.174	3.364	1.324	1.124
Disassoc 延遲時間（與攻擊皆失敗比較）	19.049	18.949	18.649	18.069	16.579	13.969	11.589	9.049	7.929	0.119	-1.921	-2.121

表 14 隨機位元串防禦效果（攻擊模式為 SN 值為連續增加的方式）

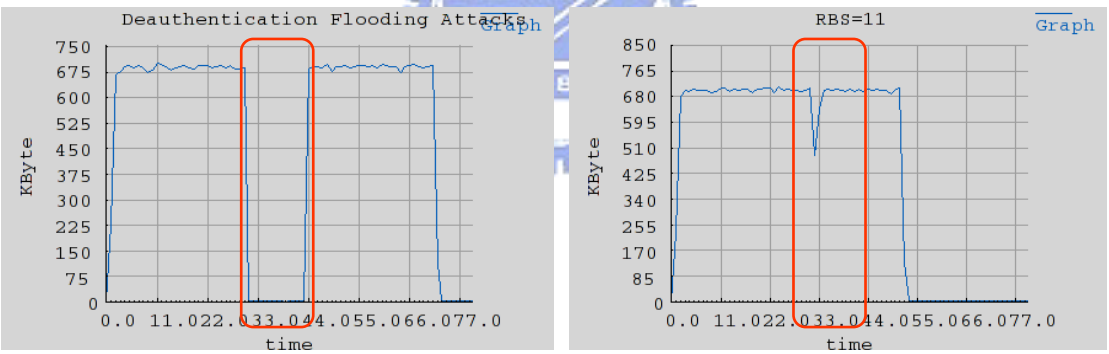


圖 44 左圖為未使用任何防禦機制，右圖為隨機位元串數為 11 時的防禦效果（攻擊模式為 SN 值連續增加的方式）

SND 過濾防禦機制測試：

接著測試只使用 SND 一層過濾的防禦效果，在此攻擊模式為 SN 連續增加與跳躍 SN 值兩種，其餘測試環境相同。表 15 為 SN 連續增加的結果，可以發現防禦效果在 SND 值為 8 時，即可以產生不錯的效果。如果要達到令人滿意地效果的話，可以將 SND 值設為 16 以上。

不過，若攻擊者以跳躍 SN 值的方式發送（SND>=10 或 SND>=30），則效果會大打折扣。如表 16（SND>=10）所示，可以看到必須要設定 SND 值為 80 以上才會有令人滿意地效果。至於表 17（SND>=30）的情況就更嚴重了，SND 值要設到 112 才可以有效防禦。之前章節（見 4.2.4）分析過 SND 之值不可以設定太大，因為會導致系統有延遲的問題發生，所以由此處可以看出 SND 的過濾機制其實是有很大的缺陷存在的。

SND	1	4	8	16	24	32
Deauth 氾濫攻擊	70.82	63.23	52.03	48.67	47.92	47.38
Deauth 延遲時間（與一般時間比較）	24.824	17.234	6.034	2.674	1.924	1.384
Deauth 延遲時間（與攻擊皆失敗比較）	21.468	13.878	2.678	-0.682	-1.432	-1.972
SND	**1**	**4**	**8**	**16**	**24**	**32**
Disassoc 氾濫攻擊	70.79	60.93	49.72	47.78	47.23	47.11
Disassoc 延遲時間（與一般時間比較）	24.794	14.934	3.724	1.784	1.234	1.114
Disassoc 延遲時間（與攻擊皆失敗比較）	21.549	11.689	0.479	-1.461	-2.011	-2.131

表 15 SND 防禦效果（攻擊模式為 SN 值連續增加的方式下）

SND	8	16	24	32	64	80
Deauth 氾濫攻擊	70.23	68.32	65.87	61.84	58.27	51.79
Deauth 延遲時間（與一般時間比較）	24.234	22.324	19.874	15.844	12.274	5.794
Deauth 延遲時間（與攻擊皆失敗比較）	20.878	18.968	16.518	12.488	8.918	2.438
SND	**8**	**16**	**24**	**32**	**64**	**80**
Disassoc 氾濫攻擊	70.38	70.14	67.92	65.09	59.34	49.07
Disassoc 延遲時間（與一般時間比較）	24.384	24.144	21.924	19.094	13.344	3.074
Disassoc 延遲時間（與攻擊皆失敗比較）	21.139	20.899	18.679	15.849	10.099	-0.171

表 16 SND 防禦效果（攻擊模式為 SN 值跳躍的方式，SND>=10）

SND	28	32	64	80	96	112
Deauth 氾濫攻擊	70.23	70.19	62.29	58.81	55.42	52.96
Deauth 延遲時間（與一般時間比較）	24.234	24.194	16.294	12.814	9.424	6.964
Deauth 延遲時間（與攻擊皆失敗比較）	20.878	20.838	12.938	9.458	6.068	3.608
SND	28	32	64	80	96	112
Disassoc 氾濫攻擊	70.21	69.94	61.03	56.97	53.04	51.37
Disassoc 延遲時間（與一般時間比較）	24.214	23.944	15.034	10.974	7.044	5.374
Disassoc 延遲時間（與攻擊皆失敗比較）	20.858	20.588	11.678	7.618	3.688	2.018

表 17 SND 防禦效果（攻擊模式為 SN 值加大跳躍的方式，SND>=30）

「SND 與隨機位元串」兩層過濾防禦測試：

最後我們合併 SND 與隨機位元串為兩層機制，攻擊者攻擊模式為跳躍與加大跳躍 SN 值的方式（SND>=10 與 SND>=30，其餘環境保持相同）。

從表 18（SND>=10）當中可以看出，當隨機位元串為 3（或 4）時，SND 需設定 20（或 12）以上才會產生一定程度的效果。此外此處亦可看出從連結上網到傳送完一檔案，本研究的提出的金鑰交換、ARBG 與隨機位元串同步演算法幾乎不會增加系統太多的額外時間（與表 13 一般時間相比）。但「SND 與隨機位元串」兩層過濾真的有效嗎?下一段我們將測試 SND>=30 的情況，檢測防禦效果是否會受到攻擊者改變攻擊模式而有所影響。

RBS=3 SND		4	8	12	16	20	24	28	32
3	Deauth 氾濫攻擊	68.51	68.82	67.37	62.17	54.68	51.93	49.81	48.03
	Disassoc 氾濫攻擊	67.49	60.95	58.12	52.87	49.03	48.72	48.89	47.67
	Deauth 延遲時間（與一般時間比較）	22.514	22.824	21.374	16.174	8.684	5.934	3.814	2.034
	Disassoc 延遲時間（與一般時間比較）	21.494	14.954	12.124	6.874	3.034	2.724	2.894	1.674
		4	8	12	16	20	24	28	32
4	Deauth 氾濫攻擊	64.98	65.71	63.33	59.98	52.02	49.34	48.81	47.64
	Disassoc 氾濫攻擊	61.92	57.76	49.31	49.12	48.87	48.63	47.52	47.21
	Deauth 延遲時間（與一般時間比較）	18.984	19.714	17.334	13.984	6.024	3.344	2.814	1.644
	Disassoc 延遲時間（與一般時間比較）	15.924	11.764	3.314	3.124	2.874	2.634	1.524	1.214

表 18 SND 與隨機位元串的兩層過濾效果（攻擊模式為 SN 值跳躍的方式，SND>=10）

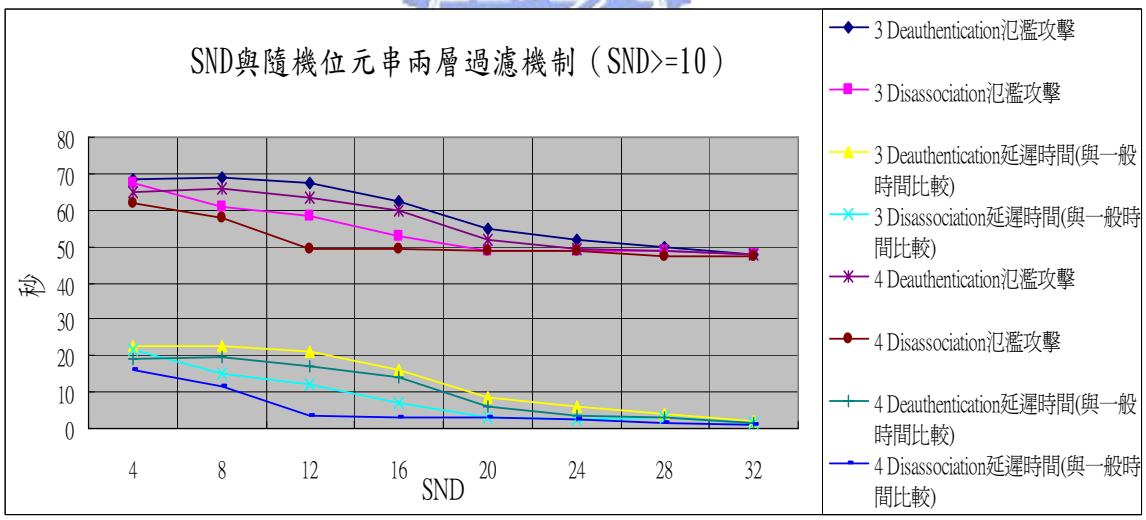


圖 45 SND 與隨機位元串的兩層過濾效果（攻擊模式為 SN 值跳躍的方式，SND>=10）

若攻擊者攻擊模式改為加大跳躍 SN 值（SND>=30）的話，則第一層 SND 防禦幾乎不會產生任何的作用（因為攻擊者刻意拉大每個 Deauthentication /Disassociation 訊框的 SN 值，造成 SND 機制無法過濾出假的訊框）。如表 19 所示，我們可以看到當隨機位元串為 3 或 4 時，SND 即使設為 32，都不會產生良好的防禦效果（比一般時間多出約 20 秒）。對於類似這種改變攻擊模式的情況，我們將在 5.4 節展現將 SND 修改為 DSN 後，DSN 可以有效地防禦。

RBS=3 \ SND		4	8	12	16	20	24	28	32
3	Deauth 氾濫攻擊	69.49	69.77	69.47	68.09	67.58	67.91	67.44	66.53
	Disassoc 氾濫攻擊	67.94	68.21	67.29	65.74	65.33	64.72	64.77	64.37
	Deauth 延遲時間（與一般時間比較）	23.494	23.774	23.474	22.094	21.584	21.914	21.444	20.534
	Disassoc 延遲時間（與一般時間比較）	21.944	22.214	21.294	19.744	19.334	18.724	18.774	18.374
		4	8	12	16	20	24	28	32
4	Deauth 氾濫攻擊	67.89	66.79	67.45	66.98	67.02	66.94	66.12	65.03
	Disassoc 氾濫攻擊	65.92	65.64	65.97	66.04	65.73	64.63	64.19	64.08
	Deauth 延遲時間（與一般時間比較）	21.894	20.794	21.454	20.984	21.024	20.944	20.124	19.034
	Disassoc 延遲時間（與一般時間比較）	19.924	19.644	19.974	20.044	19.734	18.634	18.194	18.084

表 19 SND 與隨機位元串的兩層過濾效果（攻擊模式為 SN 值加大跳躍的方式，SND>=30）

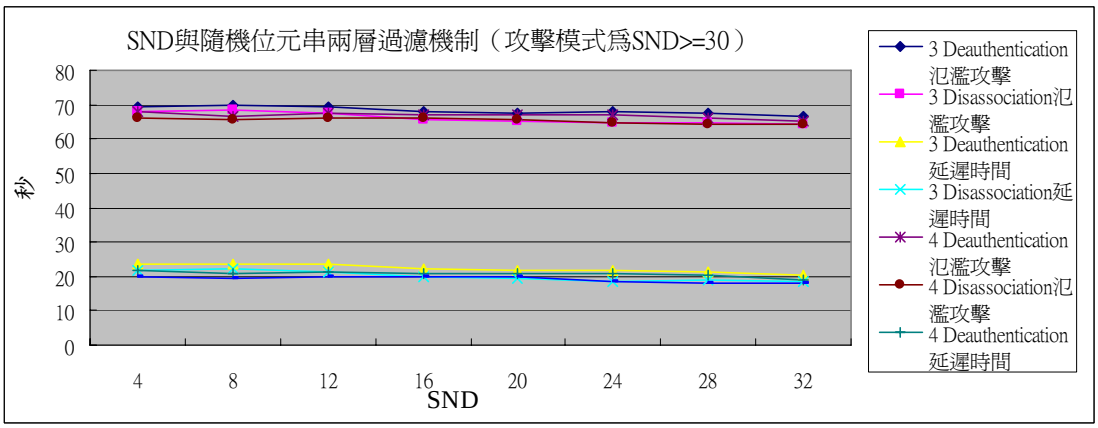


圖 46 SND 與隨機位元串的兩層過濾效果（攻擊模式為 SN 值加大跳躍的方式，SND>=30）

5.3本研究提出的 DSN（Dynamic Sequence Number）與隨機位元串兩層過濾機制效能

此節將測試（測試環境與上節相同）本論文提出的「DSN 與隨機位元串」兩層過濾機制，並與上小節 5.3 中[1]提出的「SND 與隨機位元串」兩層過濾機制互相比較，印證將 SND 修改為 DSN 在安全性上的確比較強韌，且不會帶給系統額外的負擔。

DSN 過濾防禦機制測試：

首先我們測試 DSN 的防禦能力，攻擊模式為跳躍 SN 值(SND>=10 與 SND>=30）的方式發送攻擊訊框，其餘測試環境保持相同。測試結果如表 20、21 所示，DSN 過濾防禦機制幾乎不會受到攻擊所帶來的影響，並且不會因為攻擊者加大 SND 的差距而躲過過濾，造成防禦效能的下降。此外再與表 16、17 比較，可看出 DSN 的防禦比 SND 來安全許多，底下列出 DSN 與 SND 的比較：

（1）從表 20、21 可以看出，因為 DSN 值一直在變動，所以就算攻擊者改變攻擊模式（SND>=10 變為 SND>=30）亦無法成功。其所需的傳送時間皆約 49 秒，與表 13 中的「一般傳送時間」或「Deauth/Disassoc 攻擊皆失敗的時間」比較後，可看出幾乎攻擊者的攻擊都不成功，只有攻擊時所消耗的頻寬會有所影響。

(2)從表 16、17 可以看出，SND 會受到攻擊模式改變的影響。如表 16，在 SND>=10 的攻擊下，SND 必須設到 80 以上才會有效果。然而如表 17，在 SND>=30 的攻擊下，SND 值就更大了，要射到 112 以上才會有效果，但所需時間還須約 52 秒（Deatuehentication 氾濫攻擊下）。此外 SND 值不可能設太大，因為會導致系統延遲的問題，所以 SND 防禦是不切實際的。

DSN	1	2	3	4	5	6	7	8	9	10	平均
Deauth 氾濫攻擊	49.67	49.64	49.71	49.41	49.66	49.39	48.92	49.07	49.02	49.13	49.362
Deauth 延遲時間（與一般時間比較）	3.674	3.644	3.714	3.414	3.664	3.394	2.924	3.074	3.024	3.134	3.366
Deauth 延遲時間（與攻擊皆失敗比較）	0.318	0.288	0.358	0.058	0.308	0.038	-0.432	-0.282	-0.332	-0.222	0.01
	1	2	3	4	5	6	7	8	9	10	平均
Disassoc 氾濫攻擊	49.41	48.63	49.23	49.27	48.44	49.32	49.26	48.77	48.73	49.01	49.007
Disassoc 延遲時間（與一般時間比較）	3.414	2.634	3.234	3.274	2.444	3.324	3.264	2.774	2.734	3.014	3.011
Disassoc 延遲時間（與攻擊皆失敗比較）	0.169	-0.611	-0.011	0.029	-0.801	0.079	0.019	-0.471	-0.511	-0.231	-0.234

表 20 DSN 防禦效能（攻擊模式為 SN 值加大跳躍的方式，SND>=10）

DSN	1	2	3	4	5	6	7	8	9	10	平均
Deauth 氾濫攻擊	49.72	49.69	49.74	49.17	49.82	48.99	49.76	49.15	48.92	49.27	49.423
Deauth 延遲時間（與一般時間比較）	3.724	3.694	3.744	3.174	3.824	2.994	3.764	3.154	2.924	3.274	3.427
Deauth 延遲時間（與攻擊皆失敗比較）	0.368	0.338	0.388	-0.182	0.468	-0.362	0.408	-0.202	-0.432	-0.082	0.071
	1	2	3	4	5	6	7	8	9	10	平均
Disassoc 氾濫攻擊	49.33	48.71	49.34	49.42	48.66	49.29	48.86	49.47	48.93	49.84	49.185
Disassoc 延遲時間（與一般時間比較）	3.334	2.714	3.344	3.424	2.664	3.294	2.864	3.474	2.934	3.844	3.189
Disassoc 延遲時間（與攻擊皆失敗比較）	0.089	-0.531	0.099	0.179	-0.581	0.049	-0.381	0.229	-0.311	0.599	-0.056

表 21 DSN 防禦效能（攻擊模式為 SN 值加大跳躍的方式，SND>=30）

「DSN 與隨機位元串」兩層過濾防禦測試：

接著合併測試 DSN 與隨機位元串為兩層機制，攻擊者的攻擊模式為加大跳躍 SN 值（i.e. SND>=30）的方式不斷地發送，其環境餘保持相同。表 22 為「DSN 與隨機位元串」的防禦效能，可以看出不管在何種情況下皆只需 49 秒的時間。反觀表 19「SND 與隨機位元串」在 Deauthentication 氾濫攻擊下，需要 66 秒才

可以完成傳送。如此互相比較表 19 與 22，驗證了本研究的「DSN 與隨機位元串」兩層過濾機制的確比[1]的「SND 與隨機位元串」安全，而且對系統不會有額外的負擔。

在短時間的攻擊下，只用 DSN 就可以達到有效防禦（如表 20、21 所示）。但是隨機位元串還是有存在的必要，因為不僅可以當作互相驗證之用（確保彼此來源性），並且還是可以防禦當攻擊者意外猜到 DSN 值的情況下，還有隨機位元串的第二層防禦。此外，DSN 的機制另一個好處就是可以減少隨機位元串的浪費，因為有些攻擊者會故意一直發失敗的訊框，讓系統的隨機位元串處於不同步的情況，因而造成隨機位元串的浪費。所以，本研究還是保留隨機位元串的機制，變成「DSN 與隨機位元串」的兩層過濾機制。

DSN / 隨機位元串數=3	1	2	3	4	5	6	7	8	9	10	平均
Deauth 氾濫攻擊	49.63	49.92	49.84	49.84	49.28	49.49	48.97	49.83	49.54	49.27	49.561
Deauth 延遲時間（與一般時間比較）	3.634	3.924	3.844	3.844	3.284	3.494	2.974	3.834	3.544	3.274	3.565
Deauth 延遲時間（與攻擊皆失敗比較）	0.278	0.568	0.488	0.488	-0.072	0.138	-0.382	0.478	0.188	-0.082	0.209
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**平均**
Disassoc 氾濫攻擊	49.26	49.39	48.94	49.62	48.89	49.18	48.97	49.34	49.23	49.45	49.227
Disassoc 延遲時間（與一般時間比較）	3.264	3.394	2.944	3.624	2.894	3.184	2.974	3.344	3.234	3.454	3.231
Deauth 延遲時間（與攻擊皆失敗比較）	-0.092	0.038	-0.412	0.268	-0.462	-0.172	-0.382	-0.012	-0.122	0.098	-0.125

表 22 DSN 與隨機位元串兩層機制防禦效能（攻擊模式為 SN 值加大跳躍的方式，SND>=30）

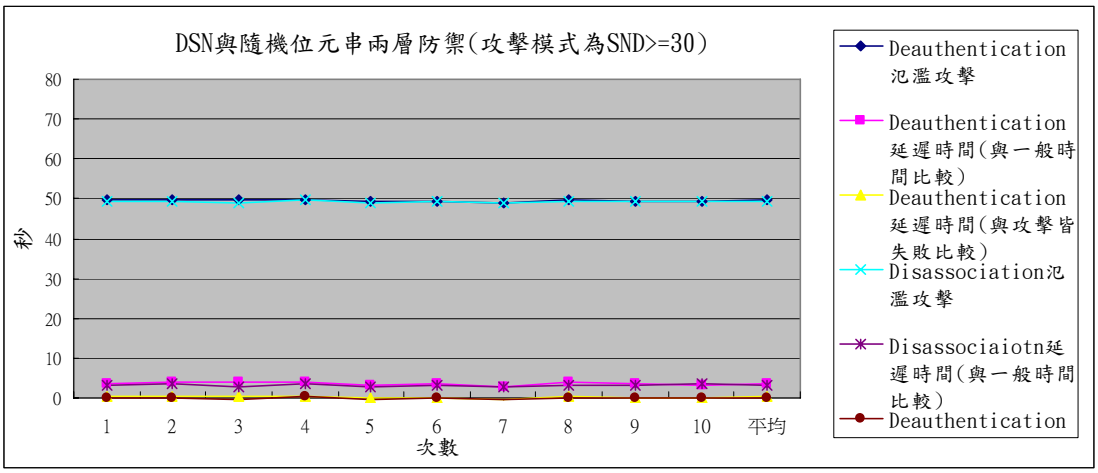


表 23 DSN 防禦效能（攻擊模式為 SN 值加大跳躍的方式，SND>=30）

DDoS 攻擊測試：

最後，測試「DSN 與隨機位元串」是否可以有效防禦 DDoS 攻擊，模擬環境與 5.1 節有些許不同。如圖 47，ID 3 為正常使用者、ID 2 為擷取點、ID 1 為擷取點連結有線端的電腦，其餘 ID 4~ID 6 的為攻擊者，每個攻擊者於相同時間內（30~40）發動大約十秒的 Deauthentication 氾濫攻擊(攻擊模式為 SND>=30)。

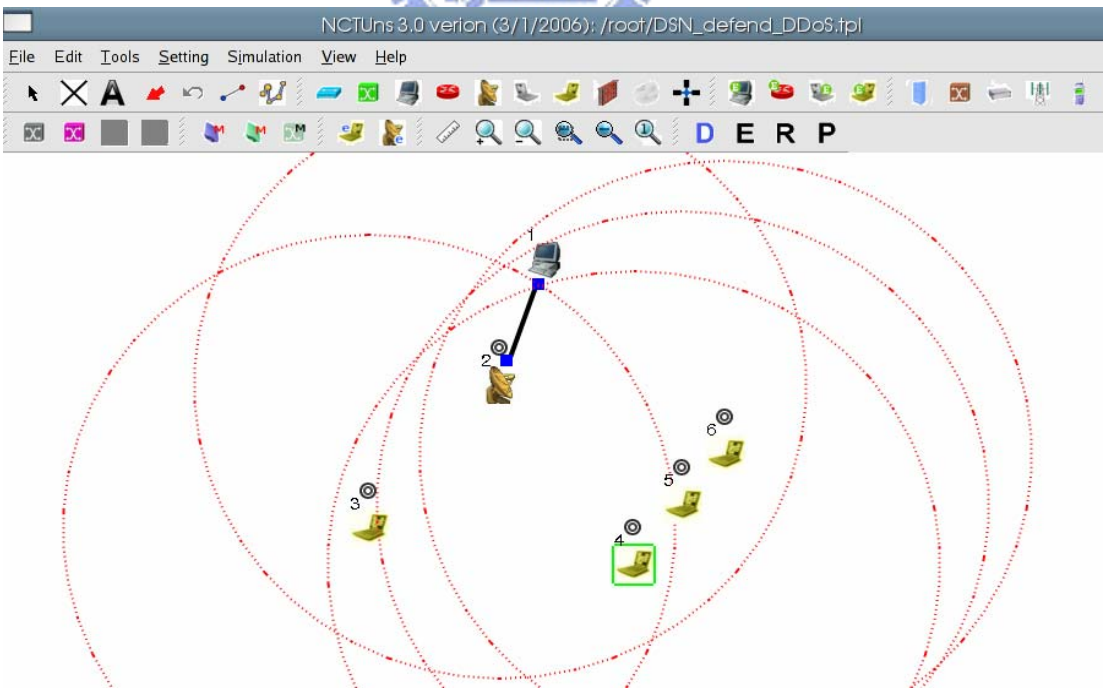


圖 47 DDoS 攻擊示意圖(攻擊模式為 SN 值加大跳躍的方式，SND>=30)

表 24 顯示「DSN 與隨機位元串(3 個)」與「SND(設為 32)與隨機位元串(3 個)」兩種機制在 DDoS 攻擊下的比較表，可以看出本研究的「DSN 與隨機位元串」比原先的「SND 與隨機位元串」快了 16.66 秒，所以在一次驗證「DSN 與隨機位元串」的機制的安全性比原本的「SND 與隨機位元串的機制」要強許多。

從圖 48 可以看出即使在 30~40 秒間遭受 DDoS 攻擊，「DSN 與隨機位元串的機制」還是可以維持每秒約 450000 Bytes 的傳輸量。反之，從圖 49 可以看出「SND 與隨機位元串的機制」則完全無法抵禦攻擊而維持穩定的傳輸。

次數	1	2	3	4	5	6	7	8	9	10	平均值	平均值差
DSN 與隨機位元串(3 個)	53.84	54.43	54.31	53.96	53.81	53.92	54.02	54.41	53.92	54.13	54.075	16.662
SND(設為 32)與隨機位元串(3 個)	70.49	70.62	70.67	70.59	70.66	71.09	70.82	71.03	70.92	70.48	70.737	

表 24 「DSN 與隨機位元串」與「DSN 與隨機位元串」防禦 DDoS 比較圖（攻擊模式為 SN 值加大跳躍的方式，SND>=30）

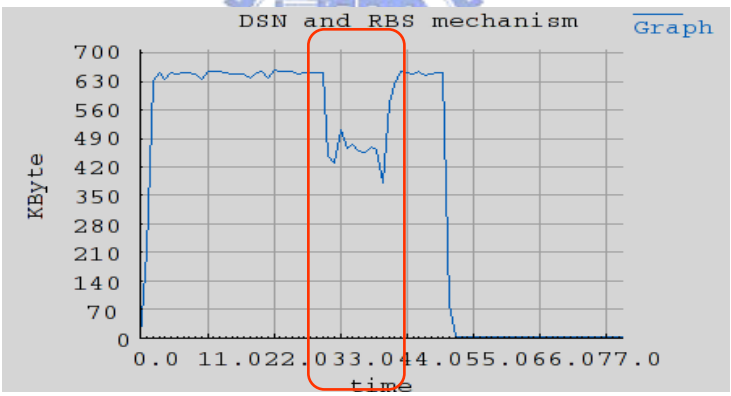


圖 48 DSN 與隨機位元串防禦 DDoS 的吞吐量

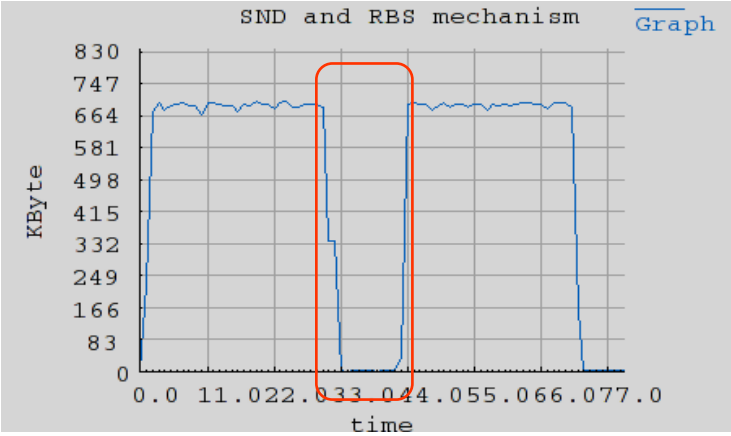


圖 49 SND 與隨機位元串防禦 DDoS 的吞吐量

5.4驗證失敗率門檻（Authentication Failure Rate Threshold）

最後我們利用統計 模型從「失敗的驗證訊框數目」中，察覺是否有非法使用者存在。因為在攻擊者無法猜到無線工作站與擷取點所共用的隨機位元串的情況下，攻擊者發動服務組絕攻擊一定是發出大量的假訊框，所以若驗證失敗的訊框數目太高，則表示網路處於不正常狀況；反之，若驗證失敗的訊框數目低於門檻，則表示網路目前處於正常狀況。

根據本研究提出的隨機位元串機制特性，系統可以藉由檢驗收到的隨機位元串，進而偵測出是否有潛在的攻擊。我們利用貝氏定理（Bayer Formula）假設正常使用者（Legal）與攻擊者（Attacker）出現機率各為 50%的假設下，每個訊框驗證的隨機位元串數為 N、驗證視窗大小為 w、回覆訊框遺失率為 l、驗證失敗的訊框數 f，回覆訊框的遺失數 m，可以推導出下列公式。

$$P(Attac\ker \mid w, f) = \frac{\binom{w}{f} * (1-2^{-N})^{f} * (2^{-N})^{w-f}}{\binom{w}{f} * (1-2^{-N})^{f} * (2^{-N})^{w-f} + \binom{w}{\frac{m}{2}} * l^{\frac{m}{2}} * (1-l)^{w-\frac{m}{2}}}$$

$$P(Legal \mid w, f) = \frac{\binom{w}{\frac{m}{2}} * l^{\frac{m}{2}} (*1-l)^{w-\frac{m}{2}}}{\binom{w}{f} * (1-2^{-N})^{f} * (2^{-N})^{w-f} + \binom{w}{\frac{m}{2}} * l^{\frac{m}{2}} * (1-l)^{w-\frac{m}{2}}}$$

底下為數學說明：

依據貝氏定理：我們可以推導出（1）

$$P(attac\ker \mid w, f)$$

$$= \frac{P(w, f \mid attac\ker) * P(attac\ker)}{P(w, f \mid attac\ker) * P(attac\ker) + P(w, f \mid legal) * P(legal)} \quad \text{----（1）}$$

$$= \frac{P(w, f \mid attac\ker)}{P(w, f \mid attac\ker) + P(w, f \mid legal)}$$

攻擊者的機率：假設攻擊者猜不到隨機位元串的情行下，根據隨機位元串 N 的設定可知攻擊的失敗機率為$(1-2^{-N})$，所以可以推導出（2）

$$P(w, f \mid attac\ker) = \binom{w}{f} * (1-2^{-N})^{f} * (2^{-N})^{w-f} \quad \text{---（2）}$$

正常使用者的機率：唯一會造成驗證失敗的訊框數 f 增加的機率，為回覆遺失的情形下，假設回覆訊框遺失率 l 與回覆訊框的遺失數 m，根據本研究同步演算法的特性，可推導出下列式子

$$P(w, m \mid legal) = \binom{w}{m} * (l)^{m} * (1-l)^{w-m}$$

而又因為若有一次的回覆訊框遺失，會產生兩個無法通過驗證的訊框(也就是 1 個驗證失敗訊框為 0.5 個回覆遺失所造成，依據本演算法特性)

$$\Rightarrow P(w, m \mid legal) = P(w, 2f \mid legal)$$

$$\Rightarrow P(w, f \mid legal) = P(w, \frac{m}{2} \mid legal) = \binom{w}{\frac{m}{2}} * (l)^{\frac{m}{2}} * (1-l)^{w-\frac{m}{2}} \quad \text{---（3）}$$

所以綜合（1）（2）（3）可得（4）（5）

$$P(Attac\ker \mid w, f) = \frac{\binom{w}{f} * (1-2^{-N})^{f} * (2^{-N})^{w-f}}{\binom{w}{f} * (1-2^{-N})^{f} * (2^{-N})^{w-f} + \binom{w}{\frac{m}{2}} * l^{\frac{m}{2}} * (1-l)^{w-\frac{m}{2}}} \quad \text{---（4）}$$

$$P(Legal \mid w, f) = \frac{\binom{w}{\frac{m}{2}} * l^{\frac{m}{2}} (*1-l)^{w-\frac{m}{2}}}{\binom{w}{f} * (1-2^{-N})^{f} * (2^{-N})^{w-f} + \binom{w}{\frac{m}{2}} * l^{\frac{m}{2}} * (1-l)^{w-\frac{m}{2}}} \quad \text{---（5）}$$

所以根據[1]，本研究只需隨機位元串 3（N=3）的情況下，可以導出（6）

$$P(Attac\ker \mid w, f) = \frac{\binom{w}{f} * (0.125)^{w-f} * (0.875)^{f}}{\binom{w}{f} * (0.125)^{w-f} * (0.875)^{f} + \binom{w}{\frac{m}{2}} * l^{\frac{m}{2}} * (1-l)^{w-\frac{m}{2}}} \quad \text{---（6）}$$

當網路異常時，網路管理者可以依據（5）判斷是因為受到無線網路的容易遺漏封包的特性造成的影響，還是因為有潛在攻擊者存在。例如圖 50，可以看出在 w=8、f=4、l=0. 01 時，攻擊者的機率為 79.2%。網管者此時就需要採取適當的措施，以防止正常使用者受到影響。

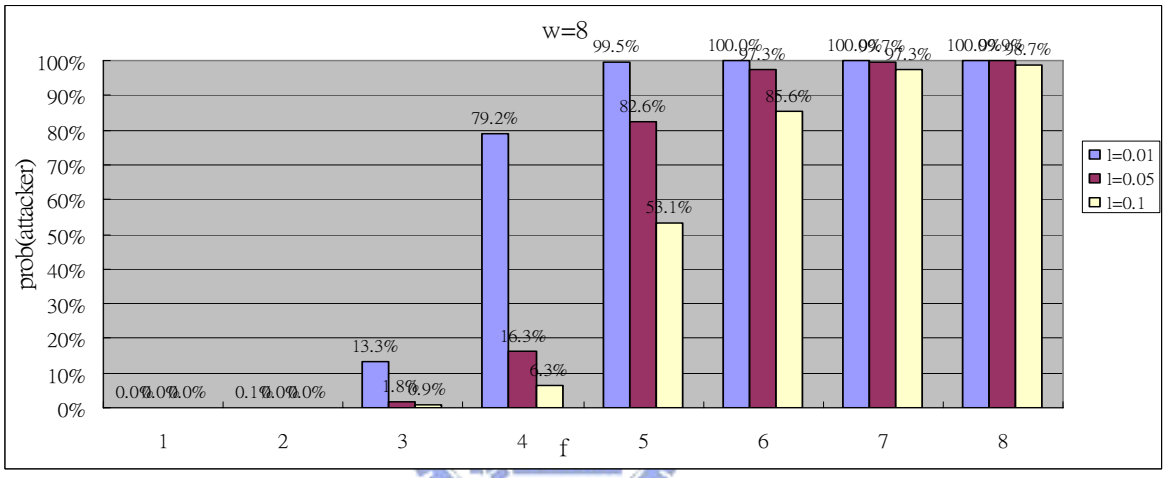


圖 50 在 w=8 時，攻擊者的機率

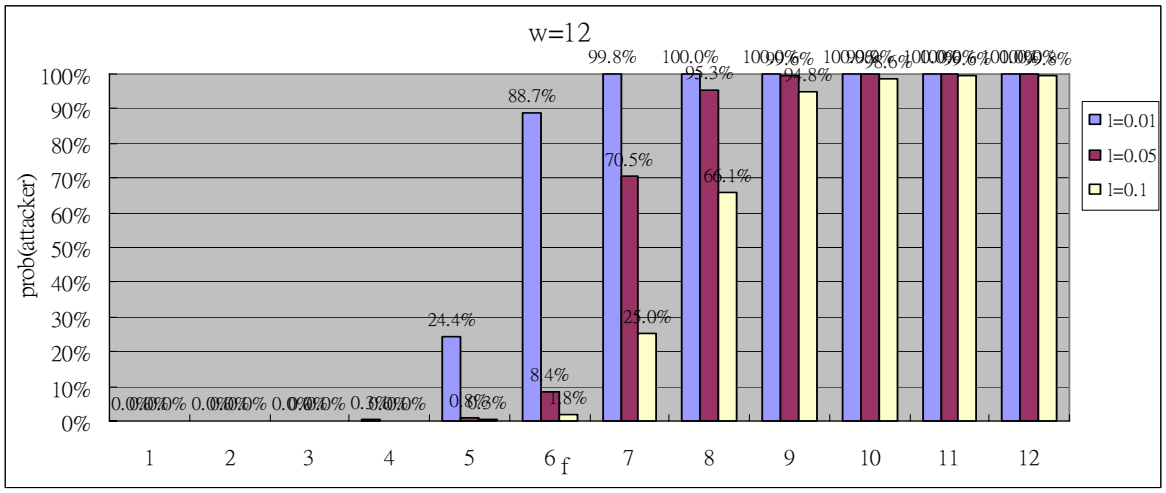


圖 51 在 w=12 時，攻擊者的機率

第六章 結論

在前面的章節中，我們提出兩層的預防機制來防禦阻絕服務攻擊，經由實驗的測試後，本研究提出的架構確實可以達到效果。在 6.1 節提出本研究的結論。6.2 節提出三點未來可以繼續擴充研究的方向。

6.1 結論

本研究使用 DSN（Dynamic Sequence Number）與隨機位元串（Random Bitstream Authentication）兩層過濾機制，防止服務阻絕式攻擊。此法不僅更改原[1]中 SND（Sequence Number Deviation）中的缺陷，增進系統的安全性外，亦為一種新穎、負擔輕且有效率的方式，不像加密防禦機制因複雜計算量，而導致負擔過重，使得系統處於引發另一種阻絕式攻擊的危機下。底下列出本論文提出系統的特點：

（1）改進的兩層過濾機制比原[1]中的兩層機制

安全性比[1]強韌且不會增加系統額外負擔，此外 DSN 可有效抵禦 DDoS（Distributed Denial of Service）攻擊。

（2）同步演算法效能大幅改進

將任何情況下恢復同步所需的時間都降為 O(1)，比起[1][7][8][9]的效能可能會受到不同情況的影響下，大幅改進系統效能。

（3）有效率且負擔輕（Lightweight）

不使用複雜的加密方式，所以比起其它用加密機制來的有效率，不需花費系統太多資源，更不會導致另一種阻絕服務攻擊。

（4）與 802.11(a,b,g)向後相容

只使用到利用訊框主體中保留的位置，所以是符合向後相容的。

6.2 未來工作

本研究提出的機制除了可以解決 IEEE 802.11 的 Deauthentication/ Disassociation 氾濫式攻擊外，底下列出三點未來研究將會應用或擴充的部份：

（1）應用於 802.11i 網路上的 EAPOL-Failure 與 EAPOL-Logoff 阻絕服務攻擊上，因為這兩個訊框都是未加密且以明文傳送，所以攻擊者亦可以利用這個缺陷讓正常使用者無法取得網路服務。

（2）此外本研究的機制亦可用於省電模式攻擊（Power Saving mode attacks）方面，因為在省電模式中 TIM（Traffic Indication Map）欄位亦是為加密的，攻擊者可以假造假的 TIM 後傳給受害者，使得受害者無法收到自己的資料。本研究的機制可以驗證來源性，所以未來將應用於這個攻擊上。

（3）使用隨機位元串來防禦探測/身份認證/連結要求氾濫攻擊（Probe/Authentication/Association request flooding attacks）。

（4）本論文是使用模擬的方式來實驗，未來將會進行實體裝置的測試，我們大膽預測實驗結果會與本研究模擬結果相符。

Reference

[1] S.D. Chien, “Using Random Bit Authentication to Defend IEEE 802.11 DoS Attacks” ,簡先得碩士論文 理學院網路學習學程 國立交通大學, May 2006

[2] Changhua He, John C. Mitchell. “Analysis of the 802.11i 4-way handshake”. In Proceedings of the 2004 ACM workshop on wireless security, ACM Press, New York, USA, 2004, Pages: 43–50.

[3] Changhua He, John C Mitchell. “Security analysis and improvements for IEEE 802.11i”. Network and Distributed System Security Symposium Conference Proceedings, 2005, http://www.isoc.org/isoc/conferences/ndss/05/proceedings/papers/NDSS05-1107.pdf.

[4] Ferreri F., Bernaschi M., Valcamonici L. “Access points vulnerabilities to DoS attacks in 802.11 networks”. Wireless Communications and Networking Conference, Vol. 1, March 2004, pages: 634–638

[5] Boland, H.; Mousavi, H. “Security issues of the IEEE 802.11b wireless LAN”. Electrical and Computer Engineering, Canadian Conference, Vol 1, May 2004 Pages: 333-336

[6] Felix Wu, Henric Johnson, and Arne Nilson, “SOLA：Lightweight Security for Access Control in IEEE 802.11” IT Professional Volume 6, Issue 3, May-June 2004 Page(s):10-16 Digital Object Identifier 10,1109/MITP,2004,21

[7] Henric Johnson, Arne Nilsson, Judy Fu, S, Felix Wu, Albert Chen and He Huang, “SOLA: A one-bit identity authentication protocol for access control in IEEE 802.11”, GLOBECOM, IEEE Global Telecommunications Conference, vol, 21, no, 1, November 2002, pages: 777–781

[8] H, Wang, A, Velayutham, and Y, Guan, ”A Lightweight Authentication Protocol for Access Control in IEEE 802.11”, In Proceedings of IEEE Globecom 2003, San Francisco, CA, December 1-5, 2003

[9] Wang, H,; Aravind Velayutham, “An enhanced one-bit identity authentication protocol for access control in IEEE 802.11” Military Communication Conference, 2003,MILCOM 2003 IEEE Volume 2, 13-16 Oct, 2003 Page(s):839 - 843 Vol,2 Digital Object Identifier 10,1109/MILCOM,2003,1290221

[10] Kui Ren, Hyunrok Lee, Kyusuk Han, Park J, Kwangjo Kim, “An Enhanced Lightweight Authentication Protocol for Access Control in Wireless LANs” Networks, 2004, (ICON 2004), Proceedings, 12th IEEE International Conference onVolume 2, 16-19 Nov, 2004 Page(s):444 - 450 vol,2 Digital Object Identifier 10,1109/ICON,2004,1409206

[11] Chibiao Liu,”802.11 Disassociation Denial of Service（DoS）attacks” Scholl of CTI DePaul University

[12] Ge, Wenfeng, S. Sampalli, "A Novel Scheme For Prevention of Management Frame Attacks on Wireless LANs", March 29, 2005

[13] Ping Ding, JoAnne Holliday, Aslihan Celik. “Improving the Security of Wireless LANs by Managing 802.1X Disassociation”, In Proceedings of the IEEE Consumer Communications and Networking Conference, Las Vegas, NV, January 2004

[14] J. Bellardo, and S. Savage. “802.11 Denial-of-Service attacks: real vulnerabilities and practical solutions”. In Proceedings of the 12th USENIX Security Symposium, Washington, D.C., August 4-8, 2003

[15] Mohammed, L.A.; Issac, B., “DoS Attacks and Defence Mechanisms in Wireless Networks” Mobile Technology Application and System,2005 2nd International Conference on 15-17 Nov. 2005 Page(s):8 pp

[16] William Stallings, “Cryptography and Network Security：Principles and Practice, Second edition” Published 1999 ISBN：0-13-869017-0

[17] .S.K. Park and K.W. Miller, “Random Number Generators Good ones are Hard to Find,” *Communications of the ACM* Vol. 31, No. 10, pp. 1192-1201, 1988

[18] Lehmer, D “Mathmatical Method in Large-Scale Computings” Proceeding, 2nd Symposium on Large-Scale Digital Calculating Machinery,Cambridge：Harvard University Press, 1951

[19] BRUCE SCHNEIER “APPLIED CRYPTOGRAPHY Protocols, Algorithms,and Source Code in C second edition” 1996

[20] Pepyne, D,L,; Yu-Chi Ho; Qinghua Zheng, “SPRiNG：synchronized random numbers for wireless security” Wireless Communications and Networking, 2003, WCNC 2003, 2003 IEEE Volume 3, 16-20 March 2003 Page(s):2027 - 2032 vol,3 Digital Object Identifier 10,1109/WCNC,2003,1200698

[21] Jesse R. Walker, "Unsafe at any key size: an analysis of the WEP encapsulation, " Tech. Rep. 03628E, IEEE 802.11 committee, March2000.

[22] J. Walker, “802.11 Key Management Series: Part I: Key Management for WEP and TKIP,” available on-line

[23] Stubblefield, A., Ioannidis, J., and Rubin, A. “Using the Fluhrer, Mantin, and Shamir attack to break WEP”. In Proceedings of the 2002 Network and Distributed Systems Security Symposium, 2002, pages: 17–22

[24] M. Luby, *Pseudorandomness and Cryptographic Applications*, Princeton University Press1996

[25] IEEE Standard 802.11i. “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”. IEEE Std 802.11i-2004

[26] IEEE Standard 802.11. “Local and metropolitan area networks- Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control (MAC) Security Enhancements”. ANSI/IEEE Std 802.11, 1999 Edition.

[27] Joshua Wright. “Detecting Wireless LAN MAC Address Spoofing”. GCIH, CCNA, 2003, http://home.jwu.edu/jwright/papers/wlan-mac-spoof.pdf.

[28] Jodi Haasz. “Re: P802.11w - Amendment to Standard [FOR] Information Technology-Tel ecommunications and Information Exchange between systems-Local and Metropolitan netw orks-Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and P hysical Layer (PHY) specifications: Protected Management Frames”, IEEE 802.11w appro ved letter, March 22 2005, http://standards.ieee.org/board/nes/projects/802-11w.pdf

[29] DB Faria, DR Cheriton. “DoS and authentication in wireless public access networks” Pro ceedings of the ACM workshop on Wireless security, 2002 - portal.acm.org

[30] D. Chen, J. Deng, and P. K. Varshney, "Protecting Wireless Networks against a Denial o f Service Attack Based on Virtual Jamming," The Ninth ACM Annual International Conf erence on Mobile Computing and Networking (MobiCom 2003)

[31] Mattbew S.Gast , 802.11 Wireless Networks: The Definitive Guide , O’REILLY 2002

[32] Jesse Walker. Status of Project IEEE 802.11 Task Group w: Protected Management Fram es. http://grouper.ieee.org/groups/802/11/Reports/tgw_update.htm

[33] RFC 2875, http://www.ietf.org/rfc/rfc2875.txt

[34] NCTUns 3.0 , http://nsl.csie.nctu.edu.tw/nctuns.html

[35] Crypto++ Library, http://www.cryptopp.com/benchmarks.html

[36] Wikipedia, the free encyclopedia. “Denial-of-Service attack” http://en.wikipedia.org/wiki/ D enial-of-Service attack

[37] Wikipedia, the free encyclopedia. “Diffie-Hellman key exchange” http://en.wikipedia.org/wi ki/Diffie-Hellman