國立交通大學

網路工程研究所

碩 士 論 文

在 RFID Gen2 下偵測 Blocker Tag 之研究

A study of Blocker Tag Detection Based on RFID Gen2 Protocol

研 究 生：呂安勝

指導教授：蔡文能　教授

中 華 民 國　九 十 八　年 五 月

在 RFID Gen2 下偵測 Blocker Tag 之研究

A study of Blocker Tag Detection Based on RFID Gen2 Protocol

研 究 生：呂安勝　　Student：An-Sheng Lu

指導教授：蔡文能　　Advisor：Wen-Nung Tsai

國 立 交 通 大 學

網 路 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Network Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

May 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年五月

中文摘要

在 RFID Gen2 下偵測 Blocker Tag 之研究

學生：呂安勝　　指導教授：蔡文能

國立交通大學資訊工程學系（研究所）碩士班

摘　　要

RFID(無線射頻辨識)是一種利用無線射頻傳送識別資料的技術，提供小型且非接觸式通訊，應用層面相當廣泛，被視為未來生活上的基礎設建設。但是，RFID Tag 硬體在先天上的限制，引發許多威脅到安全性的問題產生，如：隱私問題、Tag 竊聽、追蹤或複製等攻擊。針對這些相關安全議題，許多的相關研究被提出來，Blocker Tag 便是其中之一。

Blocker tag是對於隱私權保護的一個解決方案，是一種類似干擾訊號的演算法，利用這個演算法來保護tag不會被讀取。但是，一旦這個方案被人惡意的濫用，人們可以在商店中帶走商品並巧妙的避過結帳掃描等，可能會造成商業或作業損失的缺失，而如何去偵測出是否有存在Blocker tag，即是一種需要解決的問題。

為了解決這個問題，本篇論文設計出SLN-algorithm來偵測存在的Blocker tag，經由模擬實驗結果，證明不僅能有效的偵測Blocker Tag的存在，並且與[1]比較後偵測率和效能都勝出。

英文摘要

A study of Blocker Tag Detection Based on RFID Gen2 Protocol

Student：An-Sheng Lu　　Advisor：Wen-Nung Tsai

Institute of Computer Science and Information Engineering
National Chiao-Tung University

Abstact

Radio Frequency Identification (RFID) is a sensor network technology to automate identification. RFID systems had been applied to many fields and have been considered as a key infrastructure for the ubiquitous society in the future. However, due to the inherent drawbacks, RFID causes various security threats like privacy problems, tag wiretapping, tracking or cloning, etc. To address these issues, lots of solutions had been proposed. The Blocker Tag is one of these.

Blocker tag is a solution for privacy protection. It uses a kind of algorithm which is similar to using jamming signal. However, if this method is used in malicious way, the inventory process of the RFID reader might be stopped, and people in department store may carry some merchandise out without accounting. Find out a way to detect the Blocker Tag could be an important issue on RFID.

To resolve this problem, we proposed a scheme “SLN-algorithm” to detect the existence of Blocker tags. In our experiments, the results have demonstrated that our detection mechanism is more effective than the mechanism proposed in [1] on detecting Blocker Tag.

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，受到了許多人的幫助，在此一一感謝。首先，要感謝的是我的指導教授—蔡文能教授，在碩士班的兩年之中，老師給我許多方面的指導，讓我獲益良多，本身也因此成長了不少。再者，感謝我的父母與兩位兄弟給我一個平穩的環境在背後支持我，讓我能夠專心學習。接著要感謝實驗室的學長—蔡宗易學長和陳文學長，和學長們一起討論與研究，也不辭辛勞的指引我研究的方向，非常感謝。最後要感謝的是實驗室的同學們，大家一起工作、學習，並且互相幫助，讓我有個難忘的碩士回憶。需要感謝的人時在太多了，在我難過或沮喪時，給予我幫助與支持，讓我度過種種難關，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

無線射頻識別(Radio Frequency Identification，RFID)是一種次世代的辨識系統，具有快速辨識的能力，其功能足以取代條碼，成為現今社會最適合的辨識管理系統。RFID 是一種還在發展中的技術，目前在安全性尚存在一些問題有待解決。在本章節中，會先於 1.1 節介紹本論文研究的動機與目標。1.2 節介紹本論文的組織架構。

1.1 研究動機

RFID Tags 在硬體上有所限制，引發許多威脅到安全性的問題產生，如：隱私問題、Tag 竊聽、追蹤或複製等攻擊。為了解決這些相關安全問題，許多的安全防護措施被提出來，其中部分的安全研究是建立在 EPCglobal RFID Class1 Gen2 標準的環境下實施，例如：Back-end hash scheme、RSA Blocker tag…等。其中，我認為 RSA Blocker tag 是比較符合標準且能夠普遍性推廣的防護措施。

RSA Blocker tag 是利用碰撞排程的方式隔絕讀取器讀取 tags 的資訊。當辨識物品在一定範圍內的時候，只要貼附一張 Blocker tag，就能有效的保護物品資訊不被外漏；然而若是有人惡意使用 Blocker tag 隔絕物品並且攜帶離開保護場所，會發生辨識物品在不被察覺的情況下消失，這樣的惡意攻擊並無有效的偵測方法。因此本論文提出一種機制來偵測 Blocker Tag 的惡意攻擊。

因為目前 RFID 在全球共同認可的標準是 EPCglobal Class 1 Gen 2，在此提出的偵測機制是基於 Gen 2 標準的環境上，加強一些控制命令功能，希望能夠以安全且有效的方式偵測出 Blocker Tag 的存在。

1.2 論文架構

本篇論文的組織架構如下:『第一章 緒論』介紹研究動機與目標。『第二章 背景知識』介紹 RFID 系統的運作原理、應用與隱私問題，以及 EPCglobal 所提出的 EPC 網路、Gen2 標準和安全防護功能。『第三章 相關研究』依序介紹最早提出 Blocker Tag 阻擋消費者隱私的論文，接著進化為可彈性阻擋的 Blocker Tag 的論文，最後介紹有人提出偵測 Blocker Tag 系統的論文。『第四章 系統架構』將提出新的偵測方法，有效地判斷 Blocker Tag 的存在。『第五章 分析與實驗結

果』利用模擬實驗量測出來的數據證明本論文提出的新偵測機制不僅可行，並且比 3.1.3 節的論文[1]更安全有效率。『第六章 結論』提出研究機制的整理與總結，以及未來能夠繼續研究的方向。

第二章 背景知識

RFID(Radio Frequency Identification)，顧名思義，是一種利用無線射頻來傳送識別資料的技術，藉此達到身份辨識的目的。此技術的優點在其可提供小型、非接觸式通訊以及加密技術，藉由讀取藏於 RFID 裡多樣且具可辨識的資訊，應用層面廣泛。因此，有一個 EPCglobal 的規範組織為了方便推廣 RFID，制定了 RFID 相關的 EPCglobal 標準。在此章節中，2.1 會介紹 RFID 系統與架構。2.2 介紹 EPCglobal，以及其制定的 RFID 標準協定和安全防護。

2.1 RFID 系統

RFID(無線射頻辨識)是一種有未來發展性的技術，具有自動辨識技術，可以幫助人們利用無線射頻的方式接收並管理訊息。此系統主要由一個無線通訊 IC 以及天線當作標籤，利用讀取器接收標籤傳出的訊號，將訊號傳送至後端伺服器作管理。許多利用 RFID 的應用，例如製造商使用在供應鏈管理系統、停車場管理系統、賣場購物系統和機場行李管控系統…等，所以 RFID 被看好是未來一種無所不在的基礎建設建置在社會上。底下先介紹 RFID 系統的組織架構、運作原理及硬體設備。

2.1.1 RFID 系統元件

RFID 系統的運作方式是先將標籤貼附在需要辨識的物品上，讀取器在固定範圍內發送無線射頻搜尋標籤，並且讀取儲存在標籤裡的訊息，然後將接收到的訊息傳回到後端應用伺服器，最後由伺服器將蒐集的資料應用在各種不同的功能上，例如：倉儲管理的貨物資料、寵物管理的定位系統、電子錢包的帳務系統…等應用。RFID 的系統元件主要分成三部份(圖 1)，分別是：標籤(Tags)、讀取器(Readers)和後端應用伺服器(Back-end Application)。

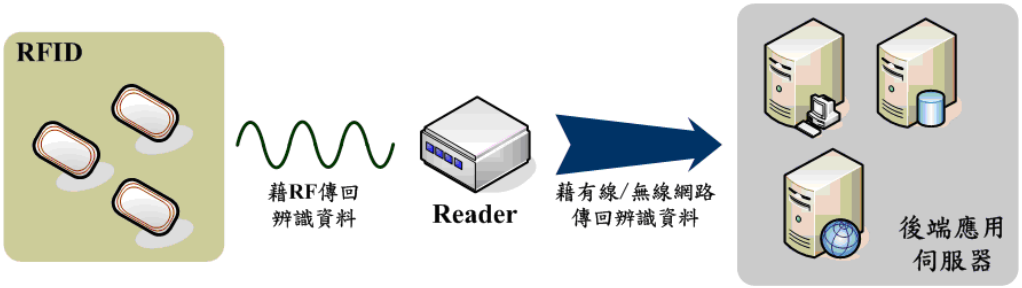

圖 1：RFID 系統

資料來源：[20]

1. RFID 標籤

RFID標籤(Tags)是由晶片和天線所組成的，它能夠儲存物品的訊息，像是：產品種類、產品製造商和產品身分等，當讀取器發出要求的時候，標籤便會將這些訊息傳送出去。標籤根據能量提供上的不同可以分成主動式、半被動式和被動式的標籤，而在EPCglobal標準下是採用被動式標籤。

主動式標籤擁有自己的電池提供能量，通常伴隨著較強大的運算能力和較長的傳輸距離，但是標籤的生命週期被電池所控制，且成本價格比較昂貴。半被動式tag使用電池能量提供給晶片運算能力，溝通的能量則來自於reader所發送的電磁波。

被動式沒有自己的電池，標籤的主要能量來自於讀取器所發出的無線電磁波訊號，標籤接收到波動產生微量電流，透過微量電流讓晶片運算處理，跟主動式標籤比起來，被動式標籤運算能力以及傳輸距離都比較衰弱，被動式標籤只具有低階計算能力，不具有高階計算功能。如：XOR，Hash function、對稱性加密就無法支援在被動式標籤上。優點在於生命週期沒有被電池限制住，成本價格便宜。大部分的RFID應用都是採取被動式標籤，因為價錢上的考量。通常可以接受的標籤成本不能超過 5分錢($USD)。

2. RFID 讀取器

讀取器(Reader)會詢問其附近的RFID 標籤並讀取標籤裡的資訊，然後將資訊再往前傳向後端資料庫提供給更前線的應用。讀取器也能夠透過下達指令的方式，更新標籤裡儲存的資訊，以達到動態更新且重複利用的功能。

3. RFID 後端應用伺服器

後端伺服器儲存的RFID 標籤資訊通常包括：物件名稱、價值、位置、製造商和擁有者…等，根據相關的應用，資料庫會變動標籤的紀錄。舉例來說，當資料庫是在幫助圖書館管理的時候，讀者租借書本將會更新資料庫中該書本標籤的時間紀錄，方便掌控圖書外借的時間；其它像零售商倉儲系統的管理、機場行李運送的管制，都是類似於圖書館的應用方式。這些不同類的工作型態，是依據後端應用伺服器的應用方向而有所不同。

2.1.2 RFID 系統應用

RFID 最主要的功能是在與辨識不同的物品。RFID 技術其本身的優勢，讓 RFID 系統以各種形式出現於日常生活中，大幅提升生活的便利性。目前於市面上的

RFID 系統隨處可見，門禁系統以及悠遊卡算是我們耳熟能詳的應用。首先我們可以看到 2006 所舉辦的世界盃足球賽，國際足球協會(FIFA)與飛利浦半導體合作，將 RFID 導入本次足球賽的售票系統中；2006 暑假台北市新開幕了一家遊樂園－神奇寶貝樂園則在台灣率先將 RFID 導入遊戲中。利用此項技術的特性，不僅可用來杜絕假門票的發生，球場周邊亦設有相關 RFID 設施，觀眾可利用門票儲值，來進行購物付款、停車、寄放物品和搭乘大眾交通工具等的服務，讓門票更有價值，更加肯定了 RFID 系統在有價證券上的應用實力。

近年來出現許多整合型的 RFID 晶片應用，有些校園與金融業者合作，提供電子化的學生證，使學生證除了可以做為進出校園的辨識，也提供許多小額金融服務，或是與金融卡合併的服務。台北市南湖國小即利用主動式 RFID 以及無線網路整合應用，來提升校園安全；透過讓每一位學生身上配戴主動式標籤，並利用讀卡機讀取後將辨識資料經由無線網路送至伺服器進行資料的篩選和處理來確保校園學童的安全。RFID 晶片植入技術近來也漸漸地嶄露頭角，娛樂產業在這股 RFID 的浪潮下也開發新的應用，在不久的未來，RFID 將大量融入社會之中，與人們生活息息相關。

2.1.3 RFID 隱私問題

雖然 RFID 帶給我們許多的方便，但卻也帶來了某些程度上的風險，如遭到惡意探測，就有可能將個人資料或是金融卡的資訊洩漏，如此將帶給使用者不少安全性上的威脅。這些 RFID 系統的應用，其背後卻潛藏許多安全性的風險。近年來也出現越來越多的 RFID 安全性事件發生，如國外就發生破解護照上 RFID 晶片的事件。由此可見，隨著 RFID 系統應用的範圍愈加廣泛，其系統的安全性就備受考驗，RFID 系統的安全性也已經是不容忽視的一項議題。隱私問題(privacy issues)主要分類為資料隱私(data privacy)和位置隱私(location privacy)。

資料隱私的情況，通常是由攻擊者利用讀取器在不被發現的情況下讀取目標特定的標籤資料，攻擊者可以取得目標的商品資訊、物品價值甚至個人的身份證識別編號；位置隱私的威脅，因為人們攜帶 RFID 標籤在身上活動，攻擊者透過持續讀取目標標籤資料來跟蹤目標標籤的行蹤，可以得到目標所有的移動路徑，且人們也很難發現自己已經被鎖定追蹤了。

美國國家標準技術局(The National Institute of Standards and Technology，NIST)便針對 RFID 技術公佈了一些 RFID 安全性導引，並且在此報告中呼籲，設計 RFID 系統時也需要將其安全性加入考量，避免造成許多安全性上的問題。RFID 系統的基本架構中，可以得知除了標籤與讀取器之間的資料傳輸之外，仍然會與後端的伺服器、中介軟體連結，這之中也將潛藏一些原本就存在的網路安全議題。經由以上的分析後，以及所發生的各種 RFID 系統的安全性

問題，可以得知現今的 RFID 系統其安全性依然不足。

2.2 EPCglobal

EPCglobal 是 2003 年 11 月建立的，由全世界一百多個國家，經過六年的研究工作才建立起來的，是對 RFID 技術新的應用。EPCglobal 是一個全球組織，代表了各行業、各部門的利益，目的是開發由用戶主導的新技術，實現多行業、跨全球的產品電子代碼技術。

目前，EPCglobal 在全球有 545 家會員，其中 20%以上的會員企業總部設在亞洲，亞洲也是 EPCglobal 會員增長最快的地區。EPC 是一項能在供應鏈中準確識別單一產品的技術，在信息可辨識方面具有獨特的優勢。EPCglobal 的主要工作是制定標準，以用戶需求為驅動，標準的制定也是為了滿足用戶的要求，是為識別各種商業利益而用的。

EPCglobal 跟 ISO 也有建設性的聯系，目前，EPCglobal RFID Class-1 Generation-2 standard[4]被制定出來，並且 ISO-18000-6 Type-C 以此標準為主制定規格，認可 EPCglobal 標準在 900MHz RFID 頻道。

2.2.1 EPCglobal Network

EPCglobal Network Framework)(圖 2)主要使用 EPC code、RFID 和資訊網絡系統等技術，建立 RFID 全球標準架構，為供應鏈運作、管理與追蹤能夠提高效率與準確性，讓企業間透過電子方式共享資料。在圖 2 的各個元件皆有制定該元件的標準，其標準的細節可以參考網站[17]。

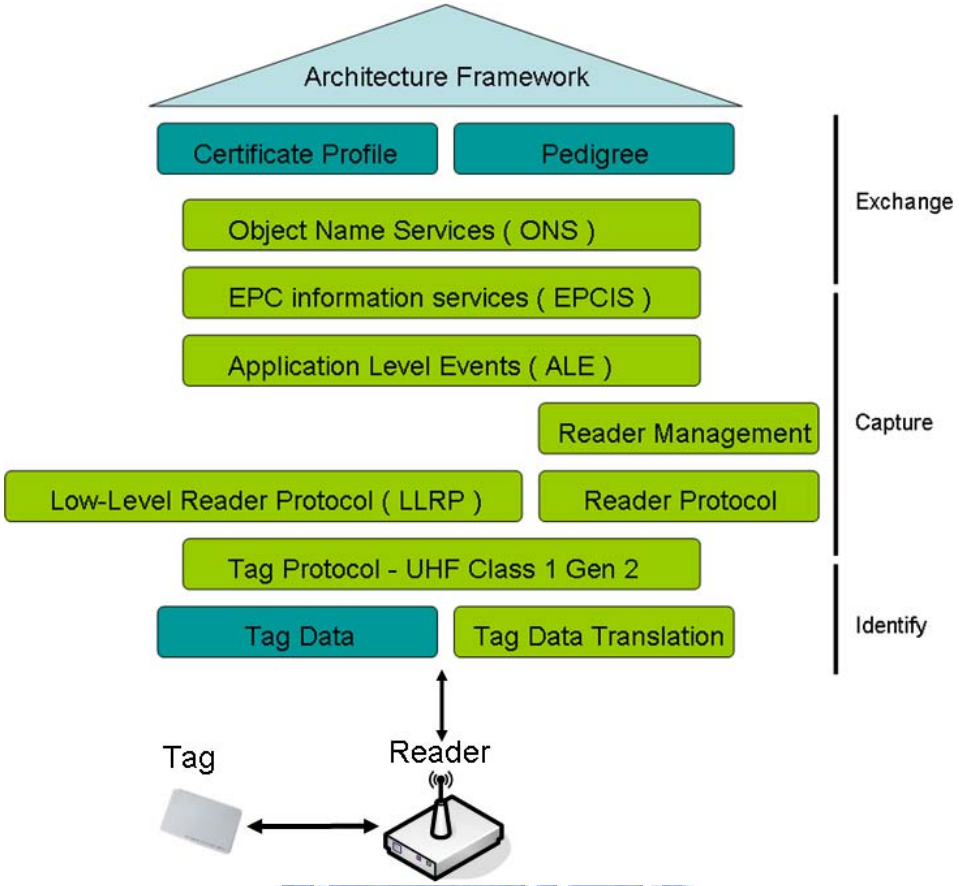

圖 2：EPCglobal Network 架構

在 RFID 系統的架構中，由標籤到讀取器，是以無線電磁波進行資訊交換，並在使用被動式標籤時，需要以電磁波對遠端標籤進行通電啟動其運作，在不同電波操作頻率 135kHz 至 5.8GHz，選擇適當的標籤與讀取器系統以及配備周邊設施，建立一個 RFID 系統進行資訊詢答之區域(Interrogation zone)，這是 RFID 在硬體設備上進行的工作與環境。

詢問區建立之後，散佈的標籤資訊(Tag Data)，經由 EPC 規範之編碼格式(Tag protocol UHF Class1 Gen2)進行標籤資訊轉譯(Tag data translation)，讀取器將詢問區中 Tag 資訊蒐集後，整合為有意義的資訊，作為辨識、查詢、管理等應用之依據，此過程由 EPC architecture framework 建構成三層網路架構：識別(Identify)、擷取(Capture)與資訊交換(Exchange)。

辨識層定義 EPC 實體物件交換標準，確保當某一使用者將實體物件送至另一使用者時，接收方能知道這個實體物件的 EPC 碼並且正確判讀。在擷取層主要定義讀取器資訊讀取解碼格式(Reader protocol & LLRP)，定義重要基礎建設元素需要收集與紀錄的 EPC 資料之介面標準，讓會員能以相容互通的構件配置自己的內部系統。

最後在資訊交換層，使用者藉由 EPCglobal Network 相互交換資料而受惠，定義出 EPC 資料交換標準，讓單一使用者能與另一位使用者之間藉由點對點互動

來共用 EPC 的資料，並得以使用 EPCglobal 核心服務與其他共享服務以增加便利性。經由事件觸發(Application Level Event)後，透過物件命名服務(Object name service)和資訊儲存中心(EPCIS)，進行認證流程(Certificate profile)，流程履歷管理(Pedigree)，或者是其他的系統服務項目(Discovery services)，實現各項 RFID 系統識別應用乃至於感測器網路的發展與智慧生活化應用。

2.2.2 EPCglobal Class1 Gen2 Air Interface protocol

EPCglobal Gen2 協議定義了物理和邏輯要求給被動式散播(passive-backscatter)、詢問者優先(Interrogator-talk-first)、RFID(radio-frequency identification)在 860MHz – 960MHz 的頻率範圍，克服了 EPCglobal 以前 Class0 和 Class1 的很多限制。它具有全面的框架和加強的功能，能夠在高密度讀取器的環境中工作，符合全球管制條例，標簽讀取正確率高，讀取速度快，安全性和隱私功能都有所加強。

從最初 2005 年 1 月 26 日公佈的 UHF Class 1 Gen 2 Standard v. 1.0.0，直到最近 2008 年 5 月 11 日又公佈了最新版本 UHF Class 1 Gen 2 Standard v. 1.2.0，代表著 EPCglobal 的 Gen2 標準一直都在持續性的更新中。V. 1.2.0 擴充了物品階層標籤的能力，增加三種新的特色；當使用者記憶體中有格式化過的資料，現在可以透過指示器的方式顯示出來；附加一個阻擋階層在使用者記憶體當中，用以保護被寫過資料的內容；經過 POS(point of sale)的運作之後，現在能夠重複的使用，此重複使用的動作決定在一個擴充的控制位元。

讀取器可一次讀取多個 RFID，但細看其通訊的過程即可發現，讀取器仍然是以一次讀取單個電子標籤的方式進行。方法大致是由讀取端以切割時間及多次掃描(Multiple scans)的交互配合來達成目標。切割時間的方法是讓電子標籤的辨識資料(ID)在某些位元符合該時槽所指定的位元時，允許該電子標籤可進行資料通信；而多次讀取的目的在於讓每一個電子標籤都有機會被允許通訊並傳送資料。

讀取器在面對一群標籤時，均透過三個階段: select(選擇)-->Inventory(盤點)-->Access(存取)(如圖 3[4])。每一個執行階段都組合了一個或多個命令集，這些階段的定義如下：

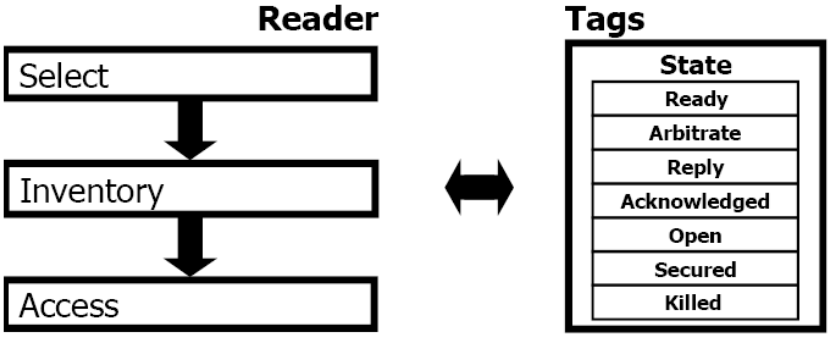

圖 3：讀取器的運作與標籤的狀態

1. Select：

此程序是讓詢問者選擇一特定數量標籤以便接下來進行盤點及存取。尋問者在盤點之前可能會使用一個或多個 *Select* 命令去選擇特定目標的標籤。

2. Inventory：

則是對已被選擇的標籤數量進行識別。每個盤點回合由詢問者在四個 Session 中的一個發出 *Query* 命令開始，然後由一或多個標籤回覆，讀取器偵測到標籤回覆後，再要求標籤回覆資料內容(如 protocol control, EPC 號碼, CRC-16 等資料)，每個盤點回合每次只能在一個且唯一的 session 中進行。本論文所進行的研究便是處在此階段的當中。

Inventory 階段運作的流程，從讀取器發送出 *Query* 命令之後，Inventory 階段就開始了，讀取器送出 *Query* 命令的之後，Tag 會進行傳送資料給讀取器的排程，而在排程當中，可能會產生三種情況，分別為衝突(collision)、單一(single)和無(empty)。下圖圖 4 所示，為讀取器和 tags 之間溝通交流的通訊協定流程，讀取器主要會發送出的命令如：*Query*、*Ack*、*QueryRep* 以及 *QueryAdjust* 等，流程的順序依圖由左到右，讀取器一個訊號發送，tags 便一個訊號傳回，只有當讀取器接收到正常的回覆訊號時，才會進行下一步的命令；在圖中也呈現了排程的三種情況分別會以怎樣的訊號傳回，如圖 4[4]裡的 Single Tag Reply、Cokkided Reply 和 No Reply。

3. Access：

在 select 與 inventory 之後，讀取器使能對單一標籤進行存取(讀取或寫入資料)。在 access 之前每個單獨的標籤必須被獨特且唯一的被辨識。Access 程序是由多樣的命令所組成。

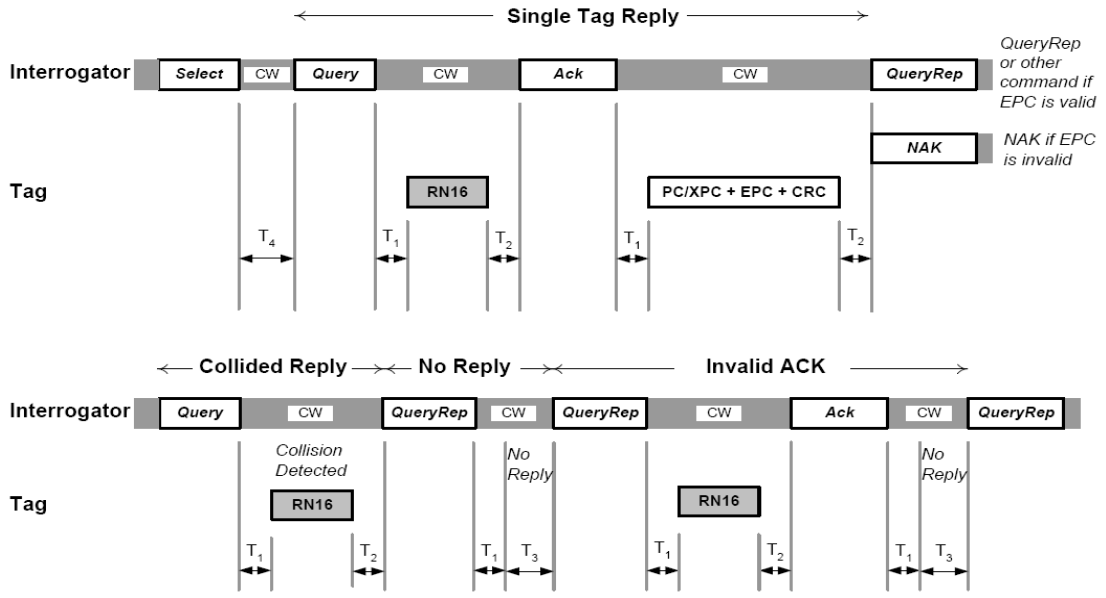

圖 4：Query 命令在 RFID 的通訊協定

在前面圖 3 提到讀取器所進行的三個步驟外，標籤也會依據相對應的命令進行不同狀態的切換，而標籤的狀態分為七種狀態，在標籤的各個狀態切換以及其運作的功能，都在下圖圖 5[4]中。因為研究的重點放在讀取器的 Inventory，所以在此不詳述標籤的狀態運作。

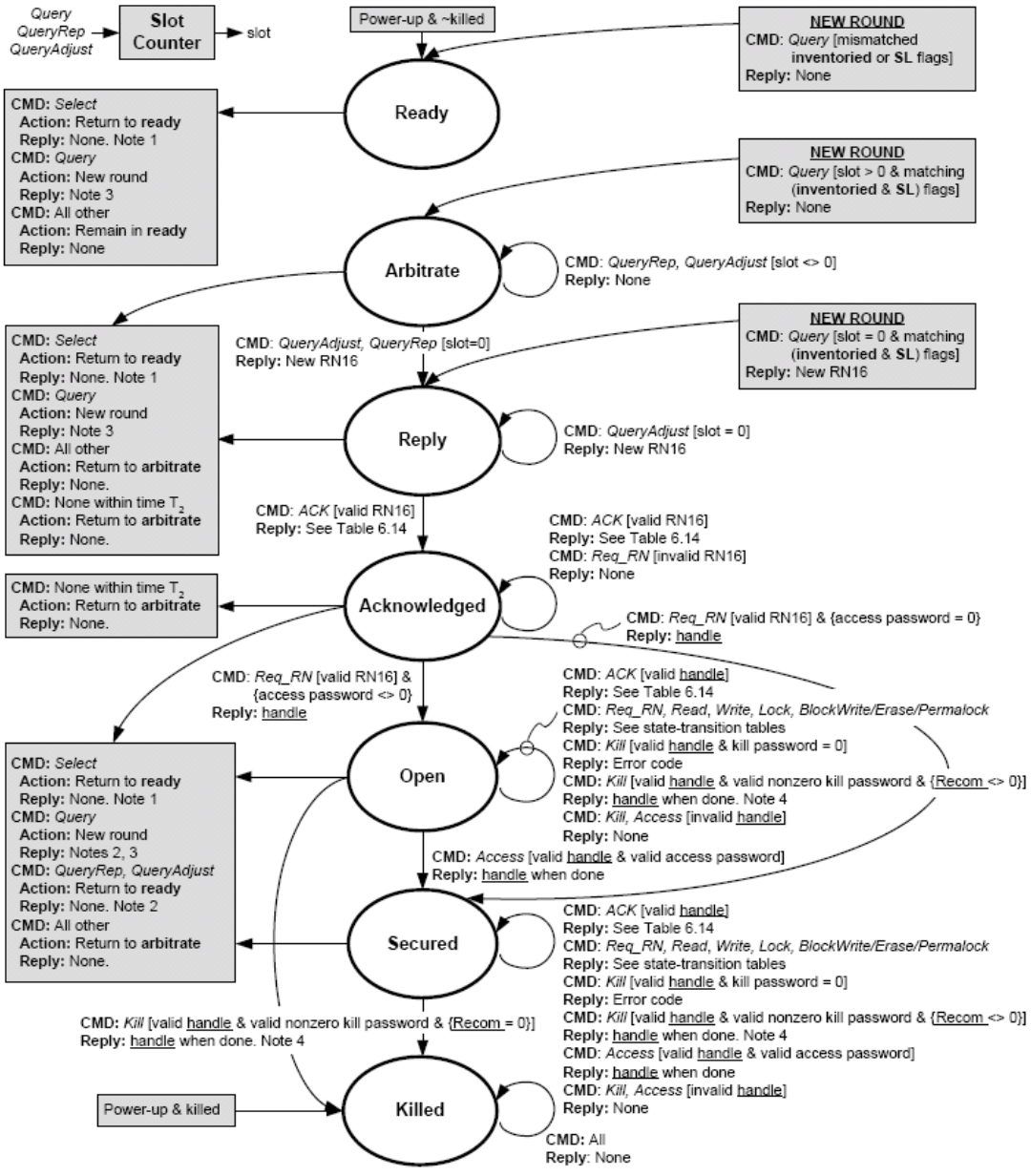

圖 5：標籤的狀態圖

在 Gen2 標準中，除了上述的通訊協定之外，對標籤的硬體規格也有一致的統一，才能方便世界上發展 RFID 系統。Gen2 標籤的記憶體主要由四個儲存庫(bank)組成，如下(圖 6[4])：

- The Reserved memory (Bank 00)，包含了兩個安全訊息，32-bit KILL 和 ACCESS password。
- The EPC memory (Bank 01)，包含 code，PC(Protocol control)和 CRC-16

這些必要訊息。

- The TID memory (Bank 10)包含 32-bit TID，TID 的數值代表著 Tag 模組號碼或是販售者的訊息等資料。
- The User memory (Bank 11)包含使用者的資料，此資料也可以為使用者本身自己的設定。

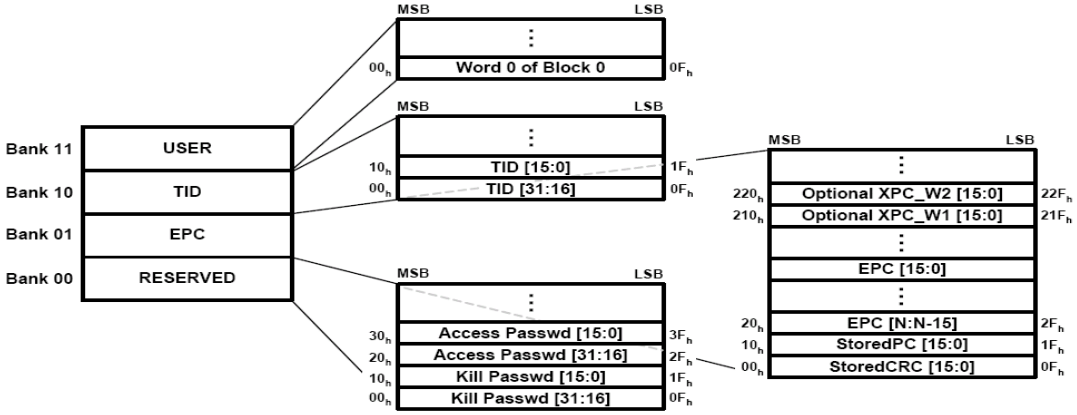

圖 6：標籤的記憶體配置

2.2.2.1 分段式 ALOHA

當讀取器和標籤之間在傳輸資料的時候，因為標籤數量龐大的關係，一定會產生許多的標籤同時傳送資料給讀取器而產生了衝突，當產生衝突的時候，為了解決衝突發生的情況，反碰撞演算法(anti-collision algorithm)就被推出來解決問題，Gen2 標準當然也要一致統一演算法的執行，於是採用了 slotted ALOHA 演算法(如圖 7)，ALOHA 演算法是利用排程的方式使所有的標籤進入時間槽(timeslots)排隊，每次以一個時間槽為計算單位進行讀取，時間槽被讀取器讀取會有三種情況發生，分別是：衝突(collision)、單一(single)和無(empty)，其運作流程如下：

- 讀取器週期性地送出訊號給標籤。
- 標籤接收到訊號之後，選擇某一個時段的時間槽進入，等待讀取器對標籤做讀取資料。
- 當一些標籤處在同一格時間槽中的時候，則讀取器跳過產生碰撞的時間槽，先採用單一存在時間槽中的標籤。
- 讀取器對單一存在時間槽的標籤通知傳送資料。
- 標籤傳送資料給讀取器。然後對剩下的標籤重複之前的動作，直到所有標籤資料讀取完畢。

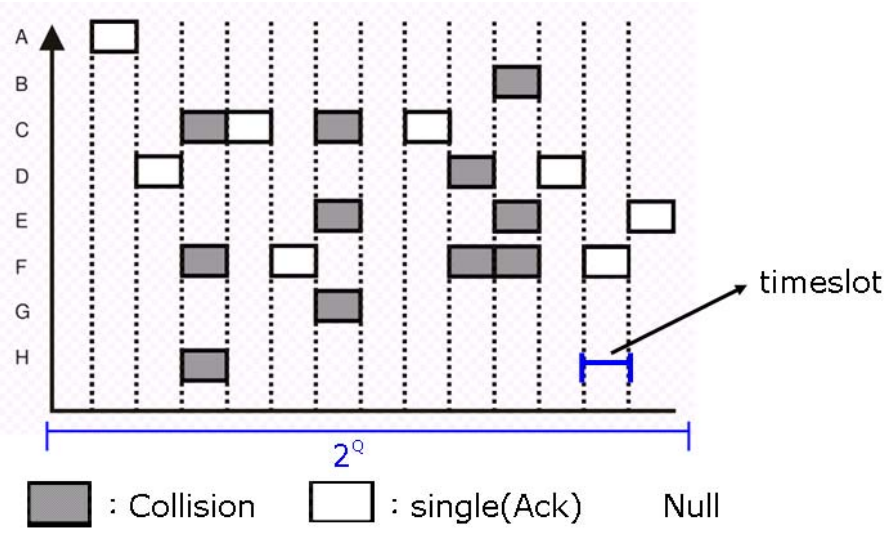

圖 7：分段式 ALOHA 通訊協定

因為衝突的發生是不可避免的，所以在RFID Gen 2標準中，加入了另一個演算法幫助降低衝突的發生。讀取器透過廣播發送出Query命令給所有的標籤，並且給予這些標簽一個變數Q，如果標籤是在ready狀態下獲得了命令，該標籤將會等待一個隨機的時間T，T是一個介於1－2^Q的(即圖 7中看到的時間槽2^Q為最大值)隨機數字，當讀取器偵測出存在衝突的時候，讀取器會增加Q值以擴大時間槽的最大值來降低衝突發生率，而根據衝突的發生紀錄，適當地變動Q值，我們稱此增加Q值的過程為：Q演算法(如圖 8[4])。

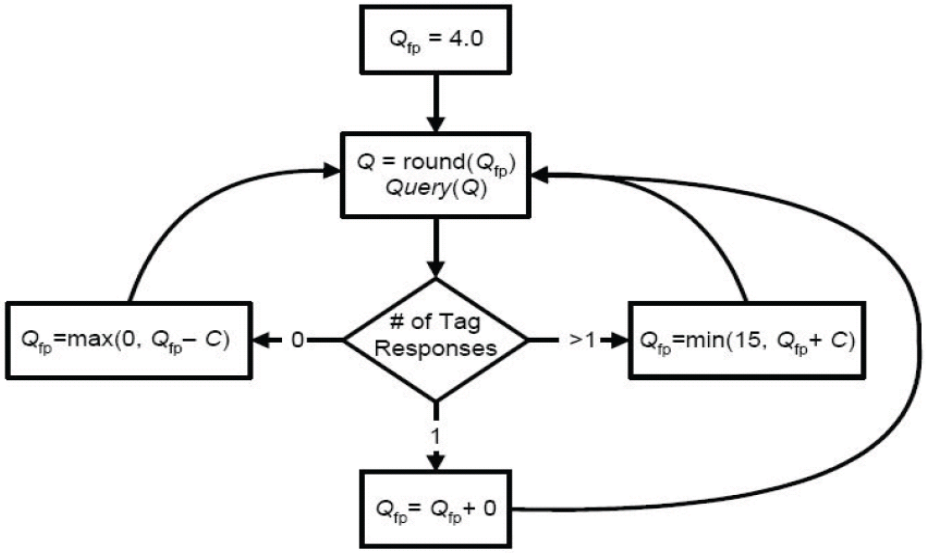

圖 8：Q 演算法

2.2.3 EPCglobal 安全防護

在 Gen2 標準中，除了對硬體以及通訊協定的標準制定外，對於資料的安全保護，有提供兩種安全機制模式：

第一種機制是殺掉 RFID Gen2 標籤，Gen2 標準定義了 KILL password 和 KILL 命令去執行，當下達 KILL 命令的時候，標籤會對讀取器所送出的 KILL password 跟標籤本身的 KILL password 作確認，比對完成且正確時，標籤會進入 KILL 狀態(標籤擁有七種狀態)，之後對於讀取器所下達的任何指令，標籤永遠不會產生相對應的動作。

第二種機制是控制接收 Gen2 的記憶體，Gen2 標準定義了 ACCESS password 和兩個指令去執行，指令分別為 ACCESS 和 LOCK，ACCESS 指令可以命令標籤進入 secured 狀態，當在 secured 狀態的時候，讀取器需要讀取資料時，只能先透過 ACCESS password 由標籤確認之後，才開放權限接受獨取記憶體，通常是搭配 KILL 指令一起傳達至標籤，標籤接受到 LOCK 指令便可以轉成可以讀寫的狀態。

2.2.3.1 KILL Tag

KILL 命令是被設計來取消掉 RFID 標籤接受命令的能力。當 RFID 標籤接受到 KILL 命令的時候，標籤會對讀取器所送出的 KILL password 跟 RFID 標籤本身的 KILL password(密碼儲存如圖 9[4]所示)作確認，比對完成且正確時，RFID 標籤會進入 KILL 狀態(Gen2 標準制定標籤有七種狀態)，之後對於讀取器所下達的任何指令，標籤永遠不會產生相對應的動作且不再接受或傳輸資料。

在 KILL 命令安全防護機制下，其主要的優點在於標籤的資料可以被有效的防護，並且不會被竊聽到資料而被追蹤。舉個例子來說，當一個消費者購買商品之後，其商品的標籤經過 KILL 處理，原本屬於商品資料的內容，就不會再被傳輸出去，不用擔心自己購買的商品會被其他人所知道，有效的保護消費者隱私安全。但是缺點也是相當的明顯，經過 KILL 處理過的標籤，都不可以被再利用的功能，不能回收性地成為一次性拋棄物品。

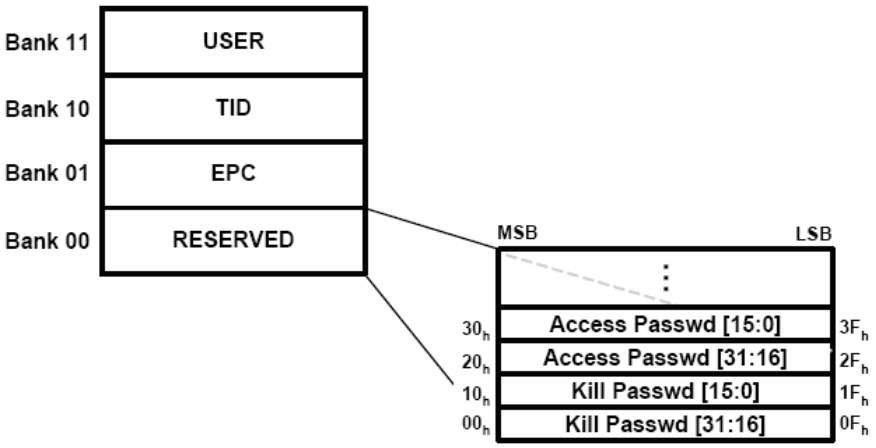

圖 9：標籤中密碼的記憶體配置

2.2.3.2 ACCESS Password

在 Gen2 標準中制定的記憶體配置，其中有 32 位元的空間被提供來存放 Password，稱為 Reserved memory(如上圖圖 9)，並且定義了 ACCESS password 和兩個指令去執行，指令分別為 ACCESS 和 LOCK，ACCESS 指令可以命令標籤進入 secured 狀態；當在 secured 狀態的時候，Reader 需要讀取資料時，只能先透過 ACCESS password 由標籤確認之後，才開放權限接受獨取記憶體。

所以當讀取器需要讀取或是更改標籤資料的時候，讀取器送出的 Password 與標籤儲存的 Password 必須經過標籤做過 XOR 的比對確認，確認過後判斷密碼為正確者，標籤才允許讀取器去讀取或是修改資料。優點在於不知道密碼的外來者，將無法讀取到標籤的資料，能夠有效的防護；最大的缺點在密碼的變化不夠龐大，其內容只有 32 位元的組合，被破解密碼的可能性不低。

在上述的 Gen2 標準裡的安全防護機制，並沒有任何防護有關訊號干擾的問題，假如有攻擊者針對 Inventory 階段進行讀取器與標籤之間的無線訊號干擾，Gen2 標準的防護機制無法有效的保護與偵測攻擊方的威脅，此處在標準的安全當中算是一種安全漏洞，所以針對此處的漏洞，本論文研究提出了一種改善此弱點的防護機制。

Chapter 3 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研究加以改善且引用其方法。3.1 介紹 RSA Blocker tag 的想法以及應用，透過 Blocker tag 的作法改善遭到惡意讀取隱私攻擊的防護；由最初的 Blocker tag 到後來更有彈性的 soft blocker tag，以及如何偵測惡意使用 Blocker tag 的 anti blocker tag 研究。

3.1 RSA Blocker tag

阻絕式攻擊(DDos)是一種在通訊協定中傳統的攻擊方法，這種方法利用在RFID的Air protocol時，被稱做為Blocker tag。這種方法的利用是由RSA率先提倡，藉由攻擊讀取器讀取標籤的通訊協定，Blocker tag可以阻止標籤被讀取；衝突(collision)的發生，是blocker tag的重心，在RFID Class 1 Gen 1是利用同時傳送 '0' 和 '1' ，達到無法正確判斷標籤的識別號碼，而RFID Class 1 Gen 2是利用在時間槽中發生排隊碰撞的情況，做到無法單一傳送訊號。Blocker Tag 的防護示意圖如下圖 10。接下來的小節中，將會詳細的介紹Blocker tag的理論。

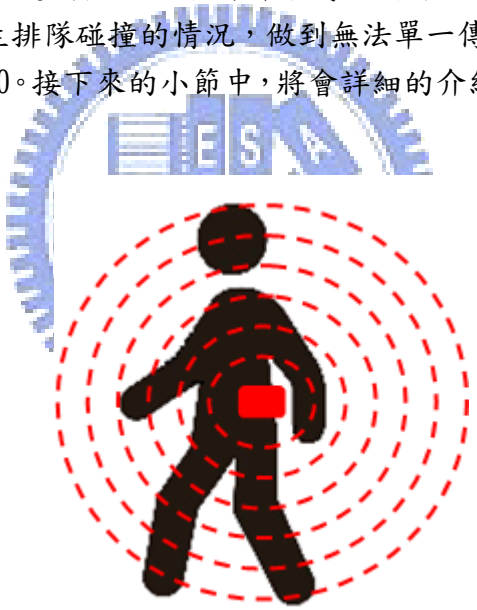

圖 10：Blocker Tag 示意圖

3.1.1 The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy

[6]在此篇論文中提到，Blocker Tag展示如何在盤存階段(Inventory)中去防止RFID讀取器去讀取資料，舉例來說，RFID Class 1 Gen 1使用樹狀演算法，利用產品的ID號碼在樹的樹枝端點位置(如圖 11)；藉由詢問所有tags的ID前序

號碼，如果發送出詢問前序 P且 P的長度是d，則tag將會送回d+1個位元的產品ID給讀取器。

透過重複以上的動作，讀取器最後可以獲得所有現存在的tag ID，但是當一個tag有兩個天線且總是送出位元0和1給讀取器的時候，讀取器將會做一次完整樹的遊走，因為讀取器將會浪費時間且導致盤存列表無效用。

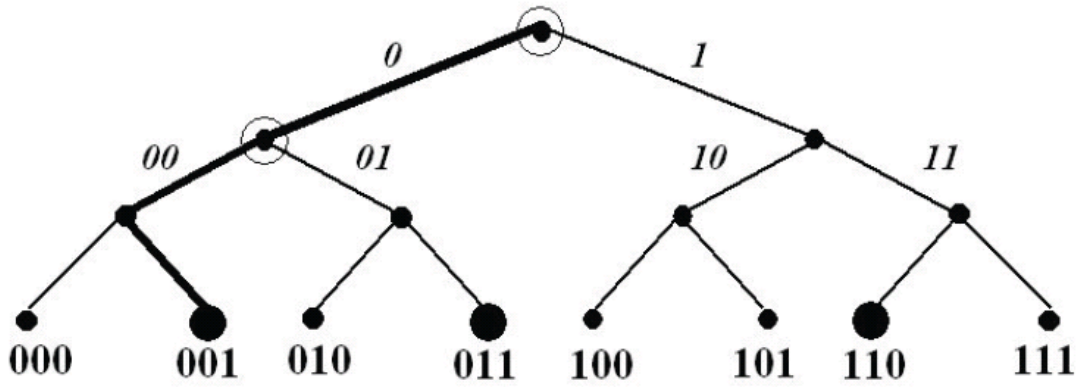

圖 11：樹狀演算法

Blocker tag的基本運作是很簡單的(如下圖圖 12)，它冒充所有的RFID-tag一系列號碼，我們可以稱這種tag是完全阻擋者(full blocker)或是全體的阻擋(universal blocker)。在樹走向演算法(tree-walking algorithm)的結構中，阻擋是很簡單能夠達到地。每當讀取器詢問tag在子樹結點B去尋求下一個位元值，blocker tag則冒充地廣播-和1的位元值(Blocker tag可能需要雙天線去做這件事情)。這種強迫衝突發生，迫使讀取器去深入所有節點，使得讀取器去探勘完全的樹。

如果讀取器有足夠的時間、記憶體和運作能量去執行完整的tree-walking演算法在這種環境下，讀取器可能會輸出所有的編號集合種類。網狀覆蓋是完全的blocker tag阻擋所有在讀取的標籤，這個點子在當一個設備在攻擊對抗盤存階段的控制系統，並且控制系統相信已經被偷的物品仍舊展示在零售環境中；而且，一個blocker tag製造簡單且造價不貴，能夠被有效的製造且廣大應用在社會上。但是，如果blocker tag被人惡意使用在偷竊物品上，反而會造成一些商業上或管理上的威脅。

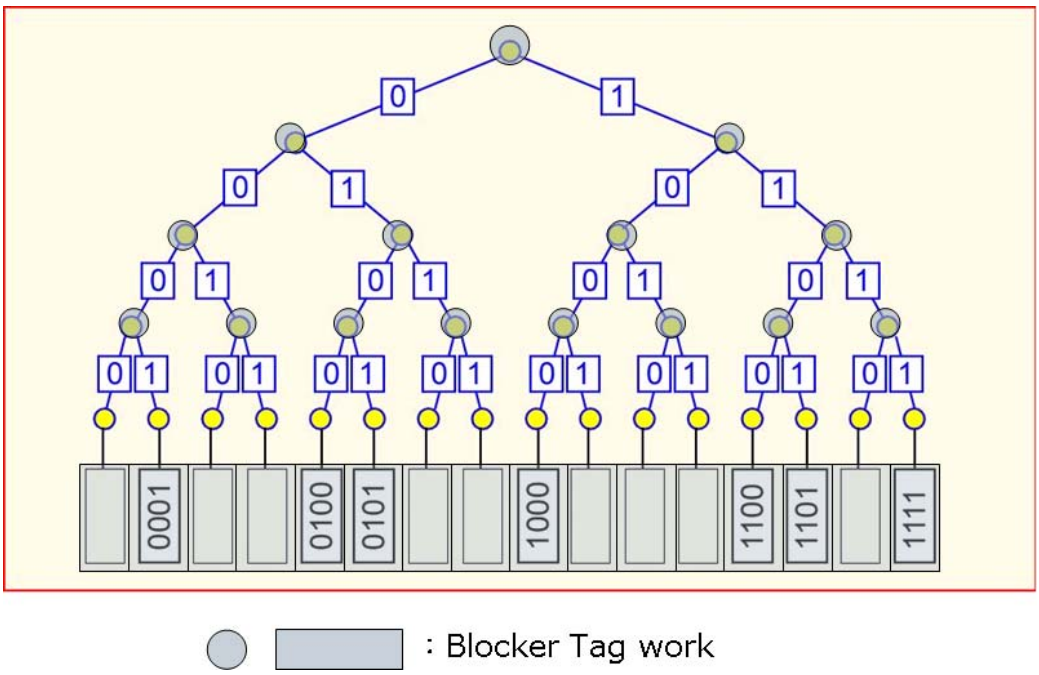

圖 12：Blocker Tag 機制在樹狀演算法

3.1.2 Soft blocking: flexible blocker tags on the cheap

此論文[7]牽涉到軟體的元件，該部份提供了一個不同於平常的Blocker tag。軟性阻擋提供一種弱一點的阻擋隱私方式，該執行是來自Blocker tag本身。這是個有意義的優點，且依靠RFID tags標準。此外，軟性阻擋也提供彈性地隱私機制，並且可能的在部份被列為有關私人的資料tags，讓blocker tag做到警戒。

這裡將介紹軟性阻擋的想法，一個便宜且在原本的Blocker tag機制下提供選擇。Soft Blocker tag不能夠提供相同強力的隱私保證在全阻擋式的Blocker tag。軟性Blocker tag也是在幾乎相同的RFID tags硬體和函式，也支援一個寬廣範圍給隱私方針。

論文中提到，我們可以將tag分為三種不同的類別，分別為blocker、private和public。如果有一個Blocker tag在蒐集的資料當中，它會傳回像是public tag的訊號，以及private的訊號。在public的樹節點中，最後可以看到該tag的ID編號，但是在private的樹節點，卻無法有效的進入到節點位置；因為軟性Blocker tag現在是針對該特定領域內的ID編號做阻擋(如圖 13)，當被阻擋起來的tag資料便是被稱做為private tag，不提供給外在的人讀取資料，而其他的public資料便交由一般的讀取器正常讀取。

一個軟性Blocker tag可以只阻擋目標ID的樹子集，而不是所有的樹。這是一個簡單的方法去保護我們的隱私，而且不需要額外增加硬體設施，但是，如果人們使用Blocker tag在惡意的應用，他們能夠躲過商店行竊的偵測系統。在這

種情況下，有人可以從商店偷東西不經過掃瞄確認，這是一個非常嚴重的問題在RFID結帳系統上。

在上述的情境當中，當我們想要對tag進行分類的時候，會分做重要物品以及非貴重物品，此時候我們便可以透過軟性Blocker tag的方式，好好的分配好運作系統，如此一來，彈性的阻擋特定tag資料，大大提高軟性Blocker tag的價值。舉例來說，一般人在使用悠遊卡的時候，應該是要能夠順利的讓讀取器讀取，且幽遊卡可以放在身上的任何地方，但是自己個人的身份資料如果設定在RFID的tag當中的時候，通常人們會將身分資料卡和悠遊卡放在皮包當中，如果不想影響到悠遊卡的存取且不想讓人偷看到身分卡的資料，這時候軟性Blocker tag就可以被拿來有效的利用。

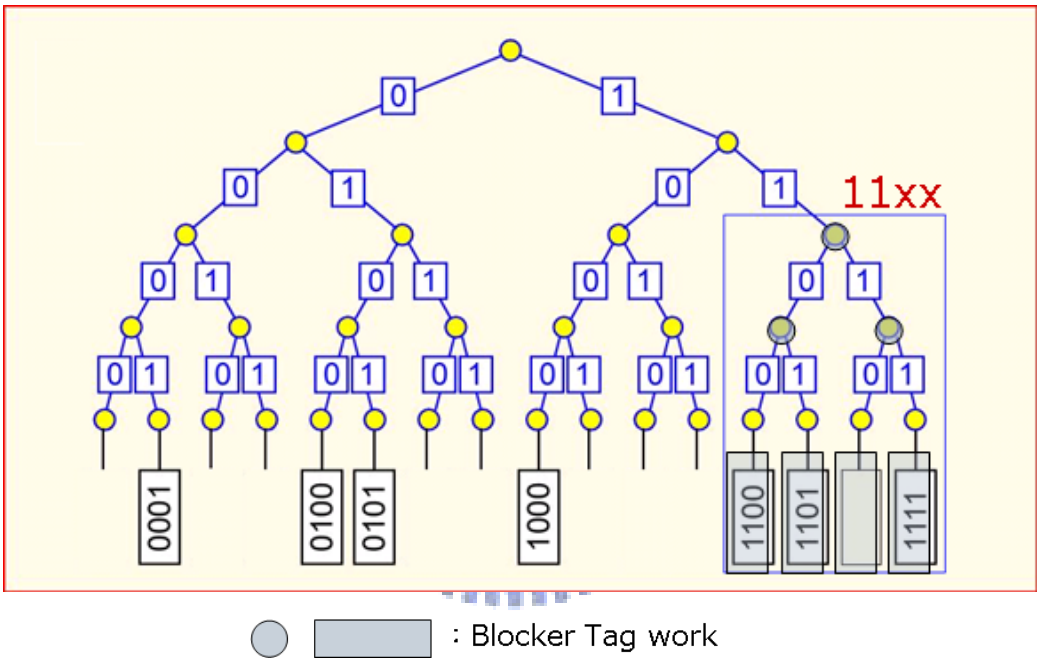

圖 13：軟性 Blocker Tag 機制在樹狀演算法

3.1.3 RFID privacy protect using blocker tag with anti-blocker tag scheme

前兩篇所提到的論文，是Blocker Tag發展的演進，是以較早的Gen 1作為基礎開發的，現在有新開發的RFID標準為：RFID Class 1 Gen 2。不同的標準有著不同的通訊協定，Gen 1使用了樹演算法，Gen 2使用了分段式ALOHA演算法，在此論文提出了如何在Gen 2標準下做到Blocker Tag的方法以及對Blocker Tag提出偵測防護的探討。

1. Anti-blocker tag in RFID Class 1 Gen 1：

根據產品ID的原則，Blocker tag會偽裝產品ID。Blocker tag利用同時回應’0’和’1’的動作，不過不能應用在上述論文提到的軟性Blocker tag，因為我們不能知道哪一部分的產品被遮蓋。在此種情況下，我們可以獲得警戒的優點和我們可以知道哪些ID將永遠不會回應我們的詢問;如果我們發出詢問且一些tag宣稱該警戒ID是存在的，代表必有Blocker tag存在。(如圖 14[1])我們稱之為警戒系統。

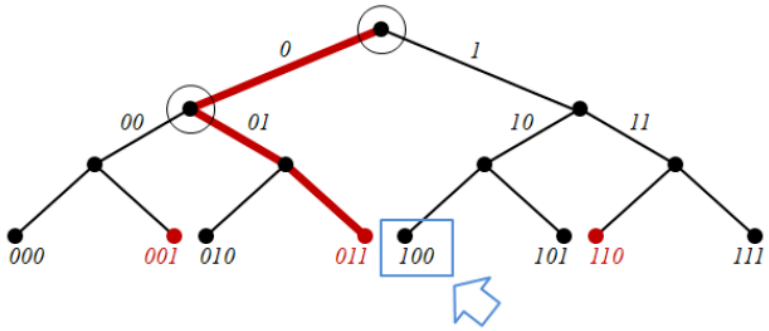

圖 14：間諜標籤機制

在軟性Blocker tag系統中，一個警戒tag將不會總是有用的，我們可以透過先行定義所有的數量大小，假如所有的盤存數量超過最大門檻。我們可以宣稱該處存在Blocker tag。

2. Anti-blocker tag in RFID Class 1 Gen 2：

在RFID Class 1 Gen 2的標準下，是利用分段式ALOHA的演算法作為基礎的通訊協定。利用時間槽的方式產生一連串的排程，再由時間槽內所發生的狀態做是否順利進入盤點的動作。Blocker tag在Gen 2的環境下建立的方法如圖15[1]，雖然產品ID進入時間槽是透過隨機取樣的方法選擇的，但是Blocker tag將所有的時間槽中都存在一個它的ID，便會導致所有的時間槽發生衝突的狀況，讀取器即無法盤點任何一樣產品ID，這就是在Gen 2的標準下做到Blocker tag的演算法。

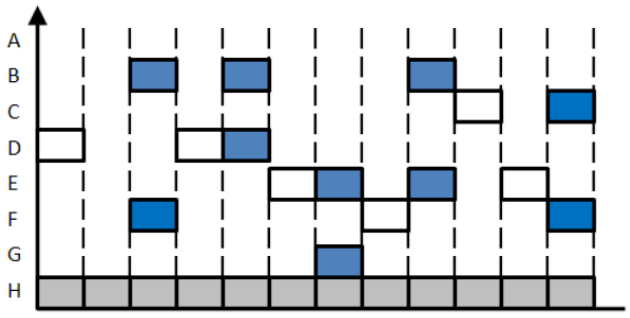

圖 15：Blocker tag 造成的碰撞

警戒系統的方法不能夠應用在RFID Class 1 Gen 2上，因為反碰撞演算法在Gen 2的時候，不能使用產品ID區別所有的產品。我們可以偵測出存在的blocker tag，藉由機率的結構。如果產品貨物中存在Blocker tag，會有很高的衝突比例發生，而我們偵測的方法中主要是想透過不正常的高衝突比例，發現狀況。

一旦Blocker tag存在，它將會無效的增加Q值，當詢問在Q值中選擇一個門檻，且衝突比例可以發現不正常的情況。此方法將總會執行在任何時間。存在一個問題就是我們所要選擇的門檻以及如何加速去發現Blocker tag的存在。

換句話說，我們依然能夠支持警戒系統，只是該警戒系統需要做一點修改。RFID Class 1 Gen 2使用分段式ALOHA演算法去盤存，儘管該演算法不能使用產品的ID去瀏覽所有的tag，我們可以藉由選擇命令中明確指定一個特定的產品ID。舉例來說，我們可以使用選擇命令去選擇某一個產品ID，然後我們使用詢問用我們所選擇的產品，然後我們將獲得預期中的回應，如果我們確定該回應是一個沉默的tag且我們獲得了一個回音，則可以判斷該處必有Blocker tag存在。

在進行實驗之後，我們知道衝突比例不能夠被使用作為門檻(如圖 16[1])。因為衝突率不只會受到Blocker rate影響，還會受到tag數量影響。一旦tag的數量增加，衝突比例也會隨之上升。此外，我們事先知道tag的數量，衝突比例不能代表Blocker tag的存在。

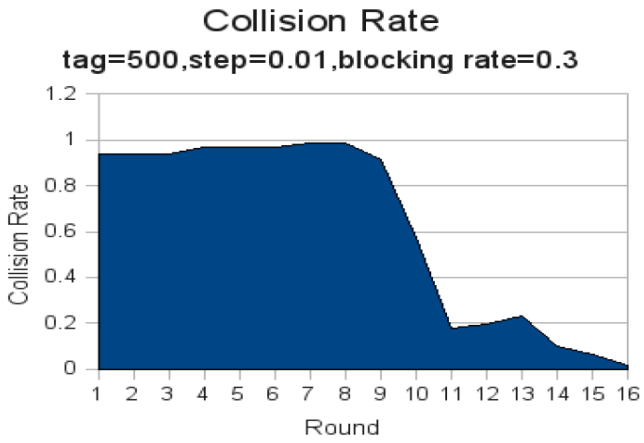

圖 16：時間影響的衝突率

定義了一個比例處在 ACK 的數量和被盤點的 tag 當作 A/Q 比例。在”Query Adjust”的方案下，Q 演算法調節 A/Q 的比例從 0.3 到 0.5。最大的 A/Q 比例是 1 且表示此情況下並沒有衝突的發生；如果 Blocker tag 收到正確的 RN16 的話，一個聰明的 Blocker tag 一定會回應 ACK 給讀取器，因為 blocker tag 送出”ACK”太多的話，會增加 A/Q 的比例。如果 Blocker tag 存在，則 A/Q 比例將會超過某個比例，根據此論文的實驗結果(如圖 17[1])，將此比例訂定為 0.4。

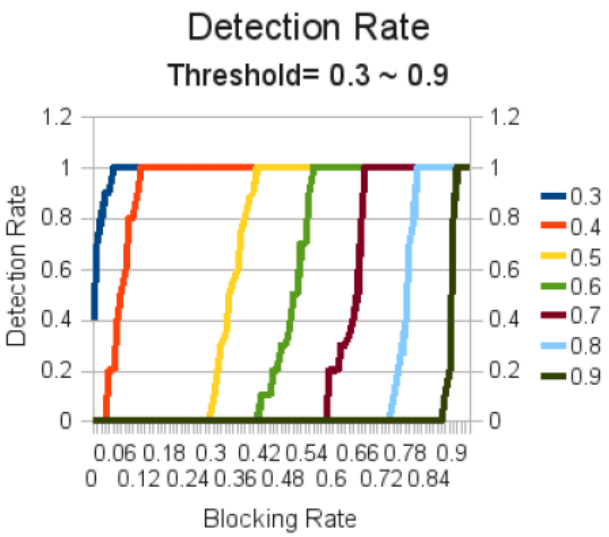

圖 17：ACK 和 Query 的數量比例

一個 Blocker tag 的 blocking rate 代表著有多高比例的時間槽被 Blocker tag 佔據。偵測和傷害是成正比的：如果 blocking rate 較低，則很難去偵測出來；若是 blocking rate 比較高，雖然會受到比較大的傷害，但是較容易偵測出 Blocker tag。傷害的意義以有多大的時間我們需要去將盤點，100%的 blocking rate 將會停止所有的盤點狀態。

但是如果 blocking rate 少於 100%的話，在盤點狀態的時候一定會在一個有限時間內，較大的 blocking rate 會需要更多的時間，我們的工作室製造一個正過的門檻，並且此門檻擁有合理的時間和偵測率。

Chapter 4 系統架構

本論文提供一個新的偵測 Blocker Tag 的演算法。本章節將依序說明偵測機制的流程，首先在 4.1 系統概觀與假設中，我們先介紹整個系統架構的流程，包括如何將標準中原有的命令加強功能的假設，到將下達命令執行動作的內容等等。接著 4.2 節介紹系統設計，把執行步驟的三個部份細分為各小節。4.2.1 初始階段，展示標籤在進行盤點之前的初始化動作。4.2.2 交流階段，讀取器與標籤之間的訊號交流，盤點各個標籤的識別順序，偵測 Blocker Tag 的行為也在此階段當中，詳細介紹本論文偵測 Blocker Tag 的演算法。4.2.3 存取階段，盤點工作完成之後，就是單純的讀取器存取標籤中存的資料。

4.1 系統概觀與假設

在系統的環境架構上，是基於 EPCglobal Gen 2 標準上，讀取器經由三個階段去存取標籤(tags)的資料，其流程和一般的情況沒有任何的不同，差別只在於發送出的命令當中，我們加入了一些額外的資訊在讀取器的命令當中。透過利用控制位元的方式，我們可以操控標籤在時間槽中所處在的位置。觀察並統計每格時間槽中的訊號資料，利用新增加的命令位元使標籤的時間槽位置重現，然後比較該回合和上回合的時間槽資訊是否相同，如果資料比對後有誤差產生，則表示有 Blocker Tag 處在該讀取範圍內。

在此種偵測方法下能夠安全且有效的偵測出 Blocker Tag 的存在，不需要以機率性的方式判斷是否存在，因為機率的方法會產生錯誤判斷的問題，並且機率式錯誤判斷只能以降低發生機率的方式減少錯誤，無法有效的確實偵測；而在此提出的偵測方式，是以非機率性的判斷方法偵測，能夠確切的判斷有惡意攻擊的存在，已達到安全有效的偵測機制。

假設：

因為我們必須加入一些額外的資訊於 EPCglobal Gen2 標準的命令訊框中，例如：slot counter 的配置、RNG 的運作等等，並且不影響到原有的運作，也就是設計的機制與 Gen 2 相容。所以本研究必須利用命令訊框新增欄位定義出新的意義。

1. 所有的標籤可以儲存 slot counter 在 RESERVED 記憶體當中，並且在每次收到 Query 或是 QueryAdjust 命令的時候執行。在前面圖 6 可以看到 EPC 記憶體原有的配置空間，因為 slot counter 是一個 16 位元的數，所以能直接存放到 EPC 記憶體的一個欄位中。

2. 我們會額外增加 Query 以及 QueryAdjust 的命令訊框，而一般的 Gen 2 Query 和 QueryAdjust 命令訊框格式如圖 18 所示，依據命令訊框裡面的控制位元，Tag 可以做不同的起始設定和執行對應運作功能，其中 Query command 的位元長度為 22bits 而 QueryAdjust 的位元長度是 9bits，本論文需要在原有的命令訊框當中，額外增加一格欄位，該欄位主要是控制 tag 的 slot counter 是否需要從 RNG(Random Number Generator)中產生，或者是利用標籤儲存在記憶體當中的記憶體送出。

	Command	DR	M	TRext	Sel	Session	Target	Q	CRC-5
# of bits	4	1	2	1	2	2	1	4	5
description	1000	0: DR=8 1: DR=64/3	00: M=1 01: M=2 10: M=4 11: M=8	0: No pilot tone 1: Use pilot tone	00: All 01: All 10: ~SL 11: SL	00: S0 01: S1 10: S2 11: S3	0: *A* 1: *B*	0–15	

圖 18：Query 命令訊框

資料來源：[4]

	Command	Session	UpDn
# of bits	4	2	3
description	1001	00: S0 01: S1 10: S2 11: S3	110: Q = Q + 1 000: No change to Q 011: Q = Q – 1

圖 19：QueryAdjust 命令訊框

資料來源：[4]

看過一般格式的 Query 命令訊框之後，本偵測演算法將會在 Q 的旁邊加上一個新的參數值，稱之為 SLN。SLN 參數值只佔有一位元的長度，分別為 0 和 1。如圖 20 和圖 21 所示。當 SLN 設為’0’的時候，Tag 收到讀取器送出的 Query 命令，除了依照原本的 Gen 2 標準 Inventory 流程運作外，標籤需要將 RNG 所產生出來的變數放到 slot counter 當中，同時也將變數儲存到記憶體裡，等待以後的利用；當 SLN 設為’1’的時候，Tag 收到 Query 命令後，不會採用 RNG 所產生出來的變數存到 slot counter 當中，而是從記憶體裡取出上一回合 Query 中 RNG 產生且儲存的變數。相同地，QueryAdjust 的命令框架上，在 UpDn 的欄位後端額外增加 SLN 參數欄位，其參數值的功能和 Query 命令一致。

	Command	DR	M	TRext	Sel	Session	Target	Q	SLN	CRC-5
# of bits	4	1	2	1	2	2	1	4	1	5
description	1000	0: DR=8 1: DR=64/3	00: M=1 01: M=2 10: M=4 11: M=8	0: No pilot tone 1: Use pilot tone	00: All 01: All 10: ~SL 11: SL	00: S0 01: S1 10: S2 11: S3	0: *A* 1: *B*	0–15	0: RNG 1: MEM	

圖 20：新 Query 命令訊框

	Command	Session	UpDn	SLN
# of bits	4	2	3	1
description	1001	00: S0 01: S1 10: S2 11: S3	110: Q = Q + 1 000: No change to Q 011: Q = Q – 1	0: RNG 1: MEM

圖 21：新 QueryAdjust 命令訊框

在 Gen 2 標準中原有的命令額外增加其框架的位元數，並不會對 Gen 2 標準的運作以及硬體規格造成任何影響。下圖圖 22 中，展示了 Gen 2 標準命令的命令碼和框架長度，從圖中可以看出命令框架最長的長度達到 59 個位元，最少也有 4 個位元，而 Query 以及 QueryAdjust 的長度分別為 22 個位元和 9 個位元；依照上述所提出的額外增加命令框架的 SLN 參數位元，將其 Query 和 QueryAdjust 各增加一位元數，由原本 22 位元和 9 位元數擴大到 23 位元和 10 位元，其位元數沒有超過 Gen 2 標準裡已規劃命令框架長度的最大值，代表著本論文提出新增加參數位元數的方法，是被包容在 Gen 2 標準中並且可以在 Gen 2 標準環境下執行運作。

Command	Code	Length (bits)
QueryRep	00	4
ACK	01	18
Query	1000	22
QueryAdjust	1001	9
Select	1010	> 44
Reserved for future use	1011	–
NAK	11000000	8
Req_RN	11000001	40
Read	11000010	> 57
Write	11000011	> 58
Kill	11000100	59

圖 22：Gen 2 命令的長度

資料來源：[4]

4.2 系統設計

在系統架構設計上，遵照 Gen2 標準讀取器讀取標籤所執行的步驟，也就是 Select、Inventory 和 Access 三階段，其中，在 Inventory 的階段中，使用本論文所提出的演算法偵測 Blocker tag，當無偵測到任何異常發生且確保並無存在 Blocker Tag 之後，再繼續進行正常的 Q 演算法盤存讀取範圍內的標籤，最後盤存結束，讀取器便可以正常的在 Access 階段存取 tag 的資料。接下來各小節將詳盡的介紹偵測演算法的各個步驟與流程。

4.2.1 Select 階段

依據 EPCglobal Class 1 Gen 2 標準，當讀取器要存取標籤(tags)資料的時候，會依照三個階段運作(其三階段如圖 3 所示)。首先，會進入 Select 階段，在此階段當中讀取器會將標籤分為四個 sessions 以及設定 selected flag 變數值，利用 session 和 flag 變數分配標籤給各個讀取器，以及判別該標籤是否已經進入存取排程被存取過。

Select 階段運作結束之後，便把所有處在讀取範圍內的標籤分配給該存取的讀取器，設定好標籤被讀取器開始存取的初始化設定，標籤設定完成才能夠進入下一個階段：Inventory。本篇論文所提出的演算法沒有更改到 Select 階段原有的運作，在 Inventory 階段才開始進入本論文演算法的執行。

4.2.2 Inventory 階段

我們在進入 Inventory 階段的時候，就可以執行 Blocker Tag 偵測演算法的運作，偵測的運作重心主要在讀取器發送的命令，因為在此提出的 Blocker Tag 偵測演算法與 SLN 的參數有關，所以之後都稱之為 SLN 演算法(SLN-algorithm)。在 Gen 2 標準的運作流程下，讀取器會先將 Select 階段選好的標籤發送 Query 命令，此時我們稱作一個回合的開始，標籤收到訊號之後會自行從 RNG(Random Number Generator)產生隨機數字，產生的數字大小處在 Query 命令中 Q 欄位參數以二為底的指數次方範圍內，標籤利用這數字以倒數的方式依序排隊準備跟讀取器交流訊號，直到在當下 Query 回合中的時間槽都讀取過後，該回合便到此結束。

在每一個回合當中，讀取器會判斷時間槽有三種狀態(2.2.2 小節中介紹到)：衝突(collision)、單一(single)和無(empty)，針對時間槽中的狀態統計，同時執行 Q 演算法，而 Q 演算法的結束都表示該回合的結束；本篇論文的偵測演

算法，是以 Q 演算法做基礎，Q 演算法的結束表示該回合的結束，之後準備進入下一回合的 Q 演算法，我們便在 Q 演算法的輸入與輸出額外進行其它運作與偵測判斷。

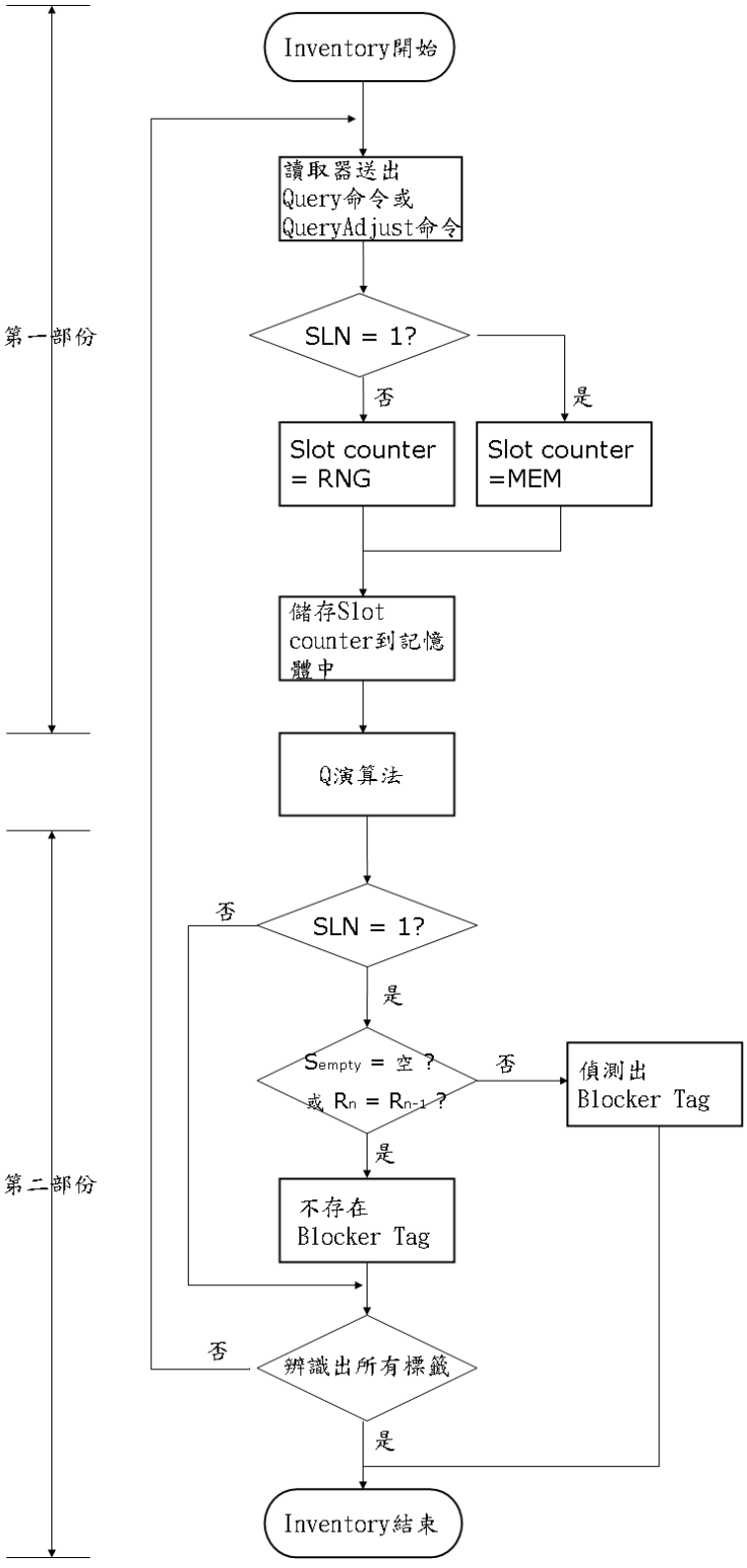

圖 23：SLN 演算法的流程圖

SLN 演算法的流程圖如上圖圖 23。SLN 演算法可以分為兩個部份，分別在 Q 演算法之前與之後。

第一部份：

Inventory 階段開始的時候，讀取器會發送出 Query 命令訊號，代表開啟了一個盤點回合(Inventory round)，在第一次發送完 Query 命令之後，之後每回合開始都會發送 QueryAdjust 的命令，Query 和 QueryAdjust 的命令訊框中有 Q 以及 SLN 的參數，因為只有此兩者命令會修改到 Q 的參數值，所以在此特地對 Query 和 QueryAdjust 命令進行關注動作。

標籤(tags)收到了 Query 或是 QueryAdjust 命令之後，會根據其命令訊框的參數值執行相對應的動作，其中，兩者命令當中 SLN 變數則是 SLN 演算法的運作重心所在。SLN 的參數值只有兩種，分別是 0 和 1 透過一個位元表示之，依據 SLN 參數值的不同，0 和 1 會使得 tag 執行的兩種不同的動作(如圖 24)：當 SLN = 0 的時候，tag 的 slot counter 則由 RNG(Random Number Generotor)產生後給予，另一方面，若 SLN = 1 的時候，slot counter 則由記憶體(RESERVED Bank)中匯出數值給予。Slot counter 的起始數值資料，只能從這兩種來源地方取得，並不會有其他的方法可以修改或刪除 slot counter 的起始數值，而 SLN 的預設參數是 0，表示 slot counter 數值的預設來源是從 RNG 取得。

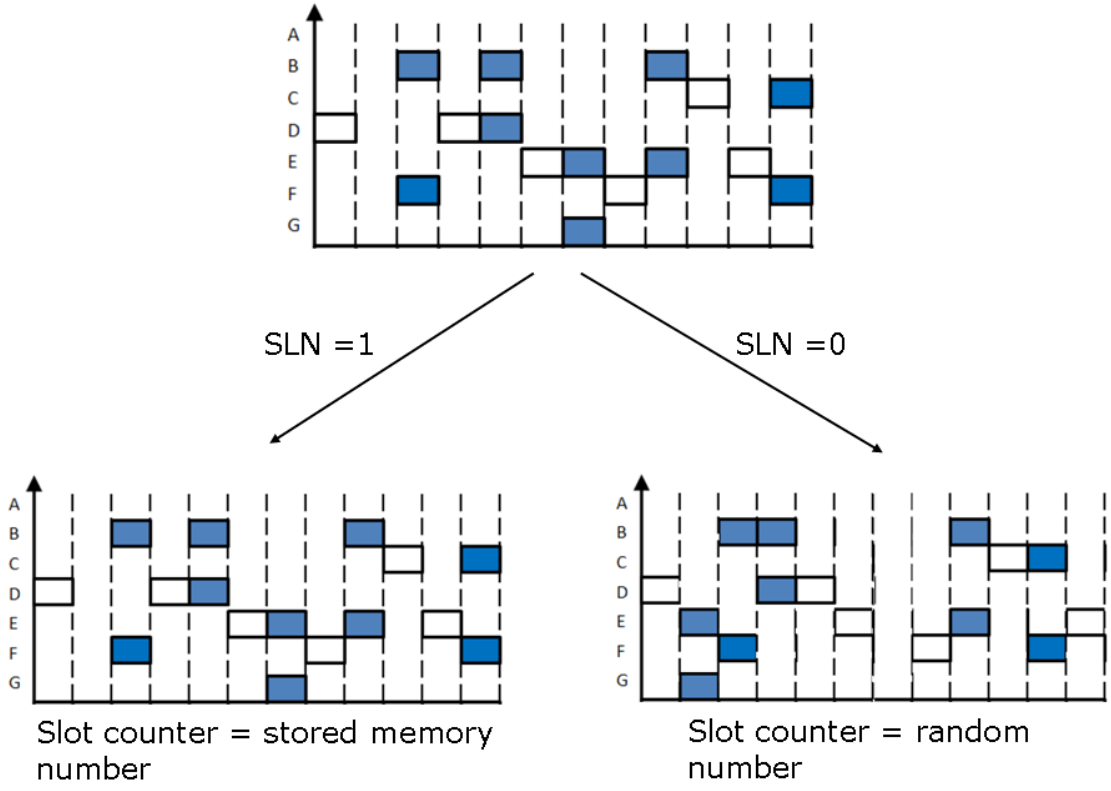

圖 24：SLN 變數控制的運作

當 slot counter 有了起始數值之後，標籤則將當下 clot counter 數值的資料存放在記憶體當中，以備未來作為判斷是否有 Blocker Tag 存在的工具，透過儲存起來的數值資料，可以比對相鄰近盤點回合的識別情況是否相同，因為利用存起來的 slot counter 數值，將兩個盤點回合的 slot counter 設置相同，根據分段式 ALOHA 演算法的運作，兩個盤點回合的盤點結果會是一致的。在標籤儲存好資料之後，即可進入 Gen 2 標準的反碰撞演算法：Q 演算法，進行讀取器識別時間槽中標籤的編碼的動作了。

第二部份：

Q 演算法結束之後，讀取器需要先進行一個判斷動作，便是先判斷該盤點回合的 SLN 值是否為 1，如果是的話，表示該盤點回合是有要進行偵測 Blocker Tag 的動作；若 SLN 沒有為 1 的話，代表該次盤點回合，乃是一次正常執行 Gen 2 辨識標籤編碼的演算法，直接接著判斷是否讀取器已經識別完所有的標籤編碼，假如識別完畢，則可以結束 Inventory 階段準備進入下一個階段；相反地，還沒有辨識完所有的標籤編碼，就可以進入到下一個盤點回合，進行下一次的 SLN-algorithm。

在判斷 SLN 為 1 的情況時，會開始進行判斷是否有 Blocker Tag 干擾讀取器識別標籤編碼(如圖 25)。首先觀察是否有不合法的訊號出現在不合理的時間槽(Sempty)當中，因為 SLN 為 1 的關係，tags 的 slot counter 數值都和上一次盤點回合的數值相同，根據分段式 ALOHA 演算法(slotted ALOHA algorithm)的運作，該次盤點回合的各時間槽訊號會是相同的，所以時間槽的狀態統計資料也會一致，但是該盤點回合的 Q 值則會依據 Q 演算法的影響而變動，因為 Blocker Tag 的存在會影響時間槽的衝突情況增加，所以 Q 會以增大的方式進行，在 Q 比上一次盤點回合還要大且 slot counter 的最大值不變的情況下，上一次時間槽最大值之外的部份，都必須都是屬於無(empty)，如果在這些必須是無的時間槽中回收到標籤的訊號，代表必有干擾讀取的訊號存在，也就是有 Blocker Tag 的存在。

判斷完 Sempty 之後，如果時間槽的狀況一切正常，沒有出現未知的訊號後，統計該次時間槽的狀態資料(Rn)，比對上一次盤點回合時間槽的狀態資料(Rn-1)是否一致，如果統計資料一致，表示讀取器讀取標籤沒有受到干擾，也表示沒有 Blocker Tag 的存在；若並不相同，代表讀取器受到干擾，導致兩次的統計資料不一致，讀取器遭受到 Blocker Tag 的影響。在此多判斷兩個盤點回合的資料是否相同，是為了防止有點聰明的 Blocker Tag 的存在；如果 Blocker Tag 以機率性的方式處在時間槽當中，很湊巧的干擾區域都在上一次盤點回合的時間槽之內，則 Sempty 的方法無法判斷出是否有 Blocker Tag 的存在，必須將上回合時間槽範圍內的資料也進行比對，才可以偵測是否有訊號干擾存在該區域時間槽之內，所以兼具觀察上回合數量的時間槽和此回合新增加的時間槽部位，才是安全完善的偵測方式。

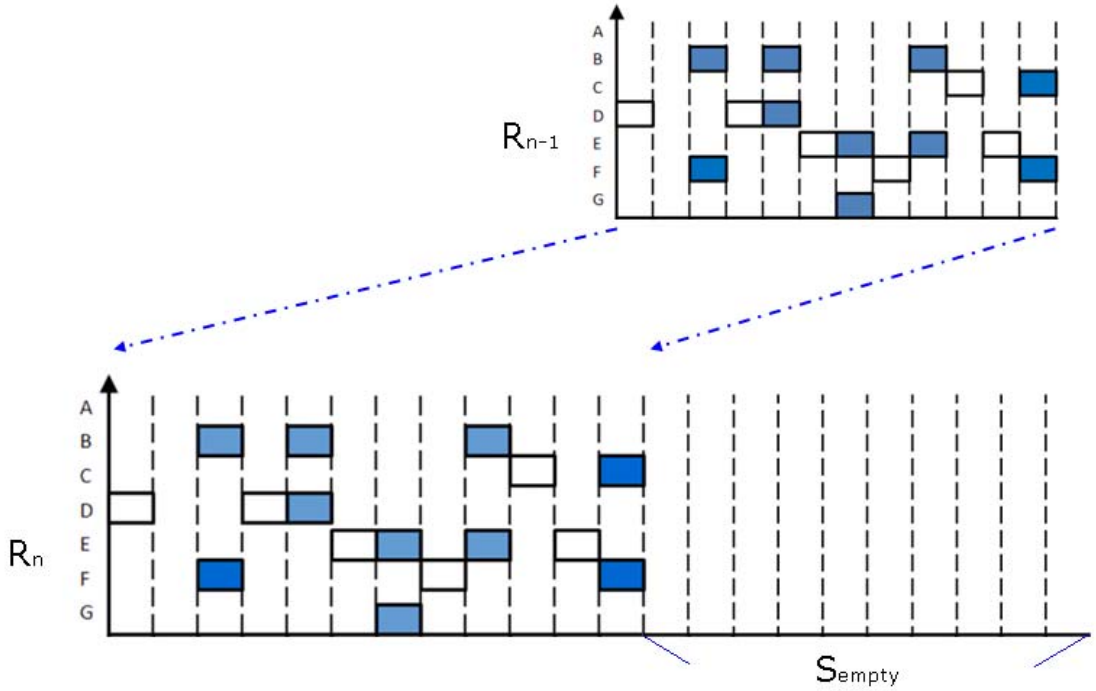

圖 25：SLN=1 時, Rn = Rn-1

以上的情況都沒有發生的話，最後在判斷是否已經辨識完所有的標籤資料，尚未辨識完畢則根據 Q-algorithm 跑完的結果，發送新的 Query 命令給標籤進行下一次的盤點回合；假如所有標籤辨識完畢之後，因為先前也已經判斷過沒有 Blcoker tag 的存在，所以沒有受到干擾而導致缺少讀取標籤的問題，那麼 Inventory 階段的運作便順利完成了，可以結束 Inventory 階段準備進入下一個階段：Access。

4.2.3 Access 階段

進行完 Inventory 階段的動作後，在 Access 階段就可以進行 RFID 系統最主要的功能動作，存取標籤儲存的數據資料，其數據資料可能有：產品訊息、個人身分、溫度、溼度、位置座標，儲值卡金額…等資訊，可以讓使用者方便且快速的存取 tags 資料，且能夠一次存取多張標籤的資料。

Access 階段中，標籤的資料可能會有密碼保護(在 2.2.3 中有介紹到)，當讀取器需要存取標籤資料時，需要先行傳送 ACCESS 密碼給標籤進行比對，標籤便會將收到的資料與記憶體中儲存的 ACCESS 密碼作 XOR 的核對，然後標籤回傳比對密碼成功的訊號給讀取器之後，讀取器便可以取得標籤資料。直到讀取器存取在 Inventory 階段中識別的所有標籤的資料之後，Access 階段便可以結束了，而 RFID 系統裡，讀取器存取標籤資料的行為也到此完成。

Chapter 5 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有效且效能優良的。5.1 節介紹實驗環境，使用 Python 編寫 EPCglobal Class 1 Generation 2 standard 的環境。5.2 節介紹系統運作的模組以及系統元件，在其小節中會呈現系統模擬的元件。5.3 節分析本論文和其他的偵測演算法的效能，實驗會以三種不同情境的 Blocker Tag 測試，並且在其小節中分別以偵測率以及偵測速度分析比較。

5.1 實驗環境

本論文研究的模擬環境如表 1，採用 Python 的程式語言來撰寫模擬環境，因為模擬環境中需要使用數量龐大的 class 才能表示系統環境，而 Python 提供了簡單且強大的動態語言，動態語言的意思表示為直譯式的語言，所以幾乎任何東西都可以在執行時動態決定，這大大地增加了使用上的彈性，也省下了重新編譯的時間，讓開發週期更短，除此之外，Python 也是良好的物件導向程式語言、可讀性佳，綜合以上的原因，導致我在此次研究中使用了 Python 作為開發系統。

表 1：實驗環境

CPU	Intel Core 2 CPU 4400 @ 2.00GHz
RAM	2.00 GB
OS	Windows XP professional SP3
Programming language	Python 2.6.1
IDE	Eclipse SDK Version:3.4.0

在模擬環境的設定上，根據我們希望得到的測試結果資料，影響模擬環境的可操縱變數在系統中有三個，分別是：

1. 正常 Tag 的數量(n_tag)：此變數影響讀取器總共需要存取的標籤(tags)總數，範圍由 50 到 400 且間距 50，50 和 100 的參數選擇是因為作為賣場購買物品的數量為限，100 以上是考慮貨物倉儲管理系統的數量，預期的變化狀態是和時間成正比，數量越大時間越長。

2. 異常 Tag 的數量(Blocker tag)：此變數控制 Blocker Tag 在讀取範圍內所存在的數量，預期的變化狀態是與時間成正比，數量越多時間耗費越長。

3. 阻擋率(blocking rate)：此變數操作 Blocker Tag 的阻擋機率，範圍由 0 到 1 的小數，預期的變化狀態是與時間成正比，機率越高時間越長。

4. SLN 參數(SLN number)：此變數操作是否執行 SLN-algorithm，變數資料只有 0 和 1，預期的變化狀態是在 1 的時候，能夠確實的找出 Blocker Tag，偵測攻擊時間的長短並不受到阻擋率(第三變數)以及數量(第二變數)的影響。

5.2 系統元件

在系統環境上，依據 EPCglobal Class 1 Generation 2 standars 的規範製作，在 Gen 2 標準轉換成模擬環境(如圖 26)，我們將讀取器以及 Tag 規劃為 class，其中，因為 Tag 會扮演兩種角色，分別是一般正常的 Tag(程式取名 NORMAL_TAG)和有阻擋性的 Blocker Tag(程式取名 BLOCKER_TAG)，然後利用函式作為 Gen 2 的環境整合上述的 class 溝通。

在環境的建構上，我們利用靜態變數控制訊號的傳遞，分別代表讀取器發送的訊號以及標籤發送的訊號，而且讀取器會從標籤送出的訊號中觀察是否存在衝突的狀況，相同地，標籤也會從讀取器所送出的訊號切換自身的狀態；在運作的流程上，因為無法將讀取器所發送出的訊號同時傳遞至標籤的身上，所以模擬實驗以循序的方式傳遞讓標籤規律性的運作，如此一來，標籤可以被當作是同時收到訊號，並且同時做完標籤該做的動作。

```
import normal_tag,blocker_tag,reader,sys

def run(Q,step,n_tag,p,t,SL):
    ch_reader=""
    ch_tag=""
    time=0
    global EPC
    EPC=0
    t_list=[]
    for i in range(n_tag):
        t_list.append(normal_tag.NORMAL_TAG(SL))
    b_list=[]
    b_list.append(blocker_tag.BLOCKER_TAG(p))

    r=reader.READER(Q,step,t,SL)
    while time <50000:
        time+=1 #time tick ++

        ch_reader=""
        r.recv(ch_tag)
        r.event()
        ch_reader=r.send(ch_reader)
        ch_tag=""
        for tag in b_list:    # blocker tag
            tag.recv(ch_reader)
            tag.event()
            ch_tag=tag.send(ch_tag)
        inv=1
        for tag in t_list:    #normal tag
            tag.recv(ch_reader)
            tag.event()
            ch_tag=tag.send(ch_tag)
            if tag.invented == 0:
                inv=0
        if inv == 1:
            ch_reader="done"

        if ch_reader=="done":

            break
    return [r.query,r.inventoried]
```

圖 26：環境的函式

5.2.1 標籤

RFID Tag 在 Gen 2 標準中被制定了七種狀態(如圖 3)：Ready、 Arbitrate、 Replay、 Acknowledged、 Open、 Secured 和 Killed，因為 RFID Inventory 階段的通訊協定只包含了前五種狀態，所以我們的模擬環境也只採用了其五種。

在下表表 2 所示，是在 Tag class 當中所處理的函式名稱，Send 和 Recv

是作為標籤送出和接收命令的功能，接收到命令之後，接下來是執行 Event 的函式功能，Event 裡面依據標籤的狀態判斷不同，會有不同的動作，Event 執行完畢之後，最後要將訊息傳送去，所以整體流程的順序是 Recv->Event->Send；newround 是當接收到 Query 的參數之後，會重新產生新的回合，要重新設定新的標籤參數。

表 2：標籤類別中的函式

Event	Tag 處理所有狀態中的命令
Send	傳送回資料
Recv	接收命令
Newround	刷新 Tag 的起始資料

5.2.2 讀取器

讀取器在 RFID 的系統中是扮演發號命令的角色，所以在模擬讀取器的類別(class)程式裡，會依據各種不同的命令進行不同的參數資料，除此之外，程式觀察 RFID 系統是否存在 Blocker Tag 的判斷式，便是處在讀取器(Reader)的類別(class)裡面，因為只有讀取器會搜集到標籤傳送回來的訊號，所以就在收到標籤訊號之後，Reader 程式也同時進行判斷式的運算；在現實環境的時候，判斷式的運算工作也是執行在讀取器當中，這樣才能達到快速有效率的管理，如果需要後端伺服器的運算能力的話，將統計資料送回到伺服器等待運算結果之外，還要等候資料傳送回讀取器，這段時間花費是一個冗長的間隔，會嚴重影響到 RFID 管理系統的效率，所以在程式模擬上，我們也儘量做到運算過程簡單。

在下表表 3 所示，是在 Reader class 當中所處理的函式名稱，其 Send 和 Recv 的功能和 Tag class 的功能相似，差別在於 Reader 送出的是命令，而 Event 是處理不同命令之間如何去做切換的動作，當收到標籤的資料後，根據標籤的資料內容更改當下 Reader 所需要送出的命令資料；因為 Reader 是持續送出命令訊號，所以不存在 newround 這個功能函式。

表 3：讀取器類別中的函式

Event	READER 處理命令間的切換
Send	送出命令資料
Recv	接收 Tag 資料

5.3 分析

在此小節中，將針對三種不同的演算法以及三種不同的情況的 Blocker Tag 做分析與比較。其中三種演算法分別是 Q 演算法[4]、AQ 演算法[1]和本篇論文所提出的 SLN 演算法。

術語：

B_n：第 n^{th} 個 Blocker Tag.（假設有許多的 Blocker Tag 存在）

R ：符合 Class 1 Gen 2 標準的讀取器

T_n ：有 n 個標籤(tags)存在讀取範圍內

依據 Blocker Tag 的能力以及情況差異，我們把 Blocker Tag 規劃為三種案例，以下將說明各種 Blocker Tag 的情況：

個案 1：

在讀取範圍內只存在一個 Blocker Tag，該 Blocker Tag(B_1)是比較原始且愚笨的 Blocker Tag，B_1 會一直對所有的時間槽發送回覆的訊號，我們定義這種情況下的 Blocker Tag 為 B_n^{α}(如圖 27)。

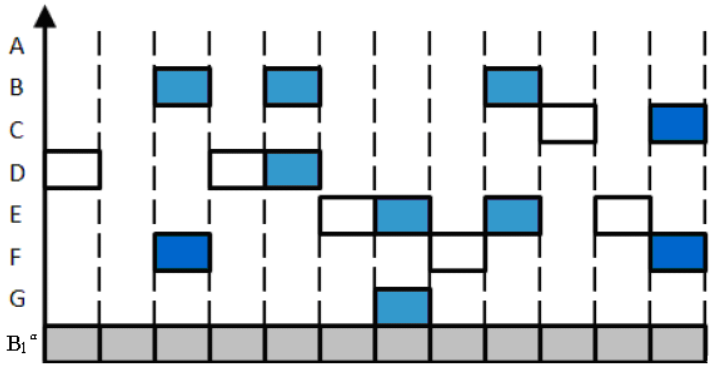

圖 27：Blocker Tag 的案例 1

案例 2：

在讀取範圍內只存在一個 Blocker Tag，該 Blocker Tag(B_1)是一種比較聰明的 Blocker Tag，B_1 只會對某些的時間槽發送回覆的訊號，並且回傳的訊號可以是衝突(collision)的訊號，我們定義這種 Blocker Tag 為 B_n^{β}(如圖 28 和圖 29)。此種情況下的 Blocker Tag 有著兩種特色：

(1)Blocker Tag 所佔的百分比。(阻擋率)

(2)Blocker Tag 的阻擋模式。(回傳訊號型態)

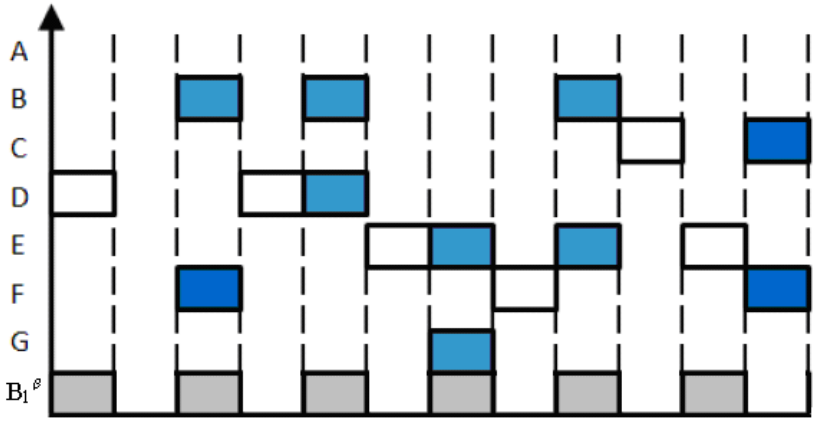

圖 28：Blocker Tag 的案例 2-(1)

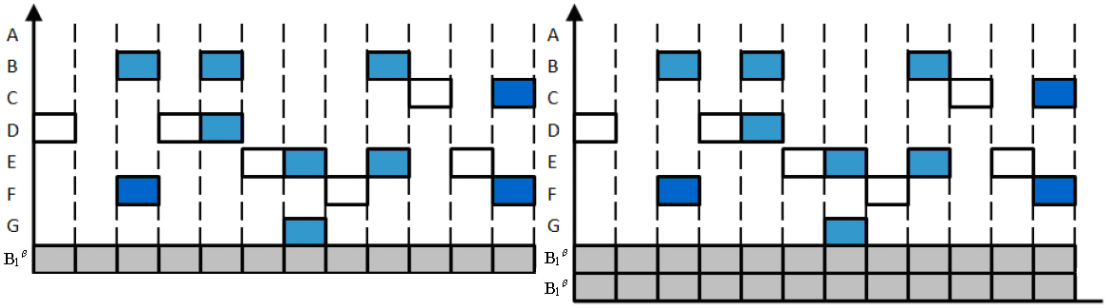

圖 29：Blocker Tag 的案例 2-(2)

案例 3：

在讀取範圍內不只存在一個 Blocker Tag，該 Blocker Tag(B_1)是屬於 B_n^{β} 的 Blocker Tag，n 的數量則不一定(如圖 30)。此種情況下的 Blocker Tag 有著兩種特色：

(1)Blocker Tag 所佔的百分比。(阻擋率)

(2)Blocker Tag 的阻擋模式。(回傳訊號型態)

(3)Blocker Tag 的數量。

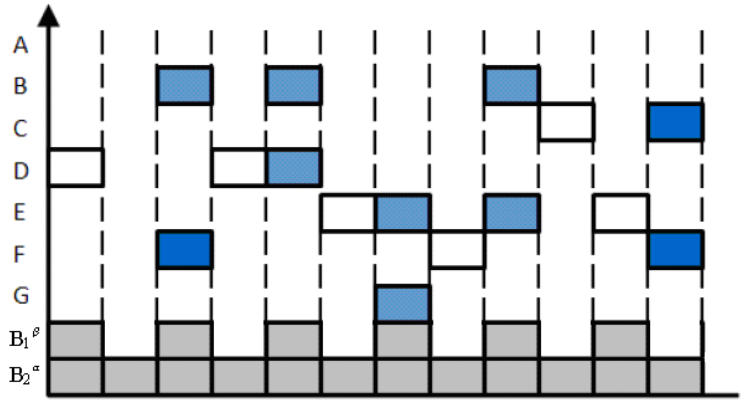

圖 30：Blocker Tag 的案例 3-(1)

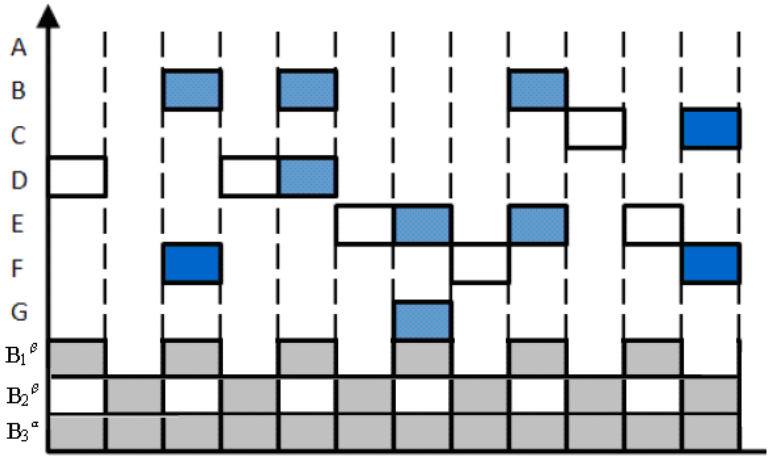

圖 31：Blocker Tag 的案例 3-(2)

5.3.1 偵測率

在此小節中，我們將對三個不同的演算法進行偵測率的比較，在三種不同的 Blocker Tag 情境下，是否能夠順利的偵測出有 Blocker Tag 的存在。

(1)案例 1：

Q 演算法：在案例 1 當中的 Blocker Tag 會持續的對 RFID 時間槽做出發送訊號的動作，而 Q 演算法只會對時間槽中的狀態進行擴大縮小時間槽空間的動作，而不會對衝突率過高產生其他應對措施，所以 Q 演算法只會永無止盡的增加 Q 的值，不能發現異常狀態的訊號攻擊。

AQ 演算法：在案例 1 當中的 Blocker Tag 因為會對所有時間槽發出訊號，所以當 Q 的數量增大的時候，ACK(single)的訊號會持續的增加，因此達到 A/Q 比例超過 0.4，即可有效偵測出 Blocker Tag 存在。

SLN 演算法：在案例 1 當中的 Blocker Tag 因為會對所有時間槽發出訊號，而 SLN

演算法會設定正常的 tags 處在固定範圍內的時間槽，只要發現固定時間槽以外的區域傳回訊號，則表示有 Blocker Tag 的存在，所以在 SLN 演算法發出 SLN 值為 1 的 Query 訊號後，觀察非固定時間槽的區域是否回應訊號，便可以有效偵測 Blocker Tag 的存在。

此次的模擬實驗結果，Q 演算法的模擬程式進入永無止盡的運作，AQ 演算法和 SLN 演算法的模擬結果如圖 33 所示，兩者演算法所提供的偵測都能有效的偵測出 Blocker Tag 的存在，而效能上的探討將會在下段小節中比較。

(2)案例 2：

Q 演算法：在案例 2 當中，Blocker Tag 會將阻擋時間槽的格子以機率挑選，這是為了躲過一些偵測攻擊的安全機制，而 Q 演算法的運作在情境二裡可以完整的讀取到 tags 的編碼，只是在效能執行上花費龐大的時間，但是 Q 演算法仍舊無法有效的偵測出 Blocker tag 的存在。

AQ 演算法：情境 2 中 Blocker Tag 使用的部份阻擋在時間槽，便是為了躲過此演算法的偵測防護。當阻擋率越低的時候，AQ 演算法越不容易偵測出 Blocker Tag，因為 A/Q 比例上升的速度較慢；相反地，阻擋率越高的時候，A/Q 比例上升較快，也較容易偵測出 Blocker Tag 的存在。除此之外，當有點聰明的 Blocker Tag 可以發送出衝突的信號的情況下，AQ 演算法無法偵測出是否有 Blocker Tag，因為在時間槽總是衝突信號的時候，不存在任何的 Ack 訊號。

SLN 演算法：情境 2 的 Blocker Tag 雖然是以機率性的方式阻擋在時間槽中，但是當我們以限定正常 tags 處在某依範圍內的時間槽，雖然 Blocker Tag 有機會能僅存在正常 tags 限定範圍內，並且剛好只落在有產生衝突的時間槽當中，但凡機率越低的時候，會有越高的機會可以躲過偵測，經過實驗的測試後(如圖 37)，在此建議連續執行兩次 SLN=1；除此之外，即使 Blocker Tag 可以發送出衝突的信號，仍舊可以依照 SLN 演算法的機制偵測出 Blocker Tag。

(3)案例 3：

Q 演算法：在情境 3 的狀況上，只要不存在情境 1 的 Blocker Tag，Q 演算法就可以辨認出所有的標籤編碼，只是時間上會很冗長。測試的實驗結果如圖 39。

AQ 演算法：情境 3 的組合上，根據機率的大小進行排列組合，會有不同的情況發生，而 AQ 演算法也能夠有效的偵測 Blocker Tag，但是在一種情況下會發生偵測失效，就是在如果有兩個情境 1 的 Blocker Tag 干擾下，AQ 演算法無法依據 AQ 比例做為判斷式；實驗測試結果如圖 40。

SLN 演算法：在情境 3 的環境中，如果有越多的 Blocker Tag 存在的話，越能偵測出 Blocker Tag 的存在，因為 SLN 演算法所設置的非固定區域，會有其它的 Blocker Tag 會回應，提高了偵測率。

5.3.2 時間性

在此小節裡，將會針對每個情境中，需要辨識所有的標籤須要的時間，比較時間的方式採用 Query 命令的數量為時間單位，包含偵測 Blocker Tag 的時間單位一起，完整的識別標籤編碼的時間：偵測 Blocker Tag 所花費的時間 + 無 Blocker Tag 識別標籤的時間。正常 RFID 系統識別標籤所需要的 Query 數量如圖 32。

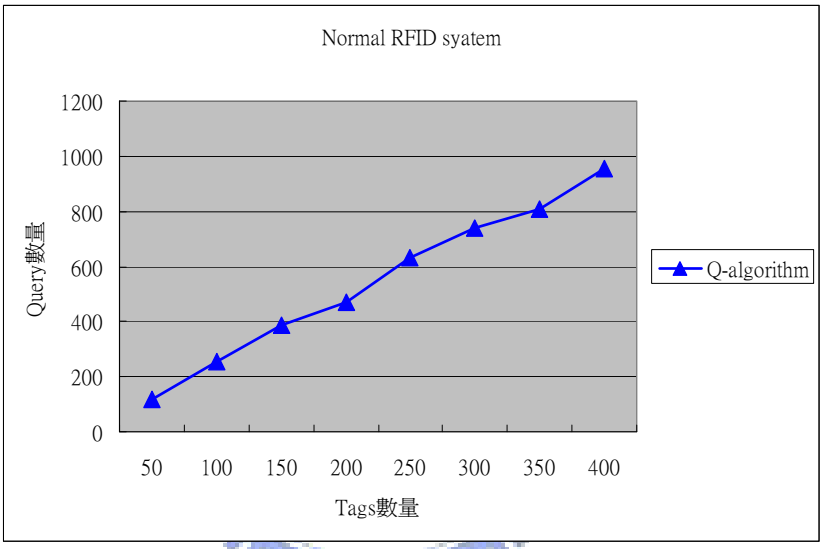

圖 32：正常的 RFID 系統

(1)案例 1：

在案例 1 的環境下，因為是一種比較笨的 Blocker Tag，所以除了 Gen2 標準裡的 Q 演算法無法識別全部的 tags 編碼，AQ 以及 SLN algorithm 其偵測 Blocker Tag 出來的時間如下圖 33。識別完目標範圍內的 tags 所需要的時間：偵測 Blocker Tag 所花費的時間 + 無 Blocker Tag 識別 tags 的時間。

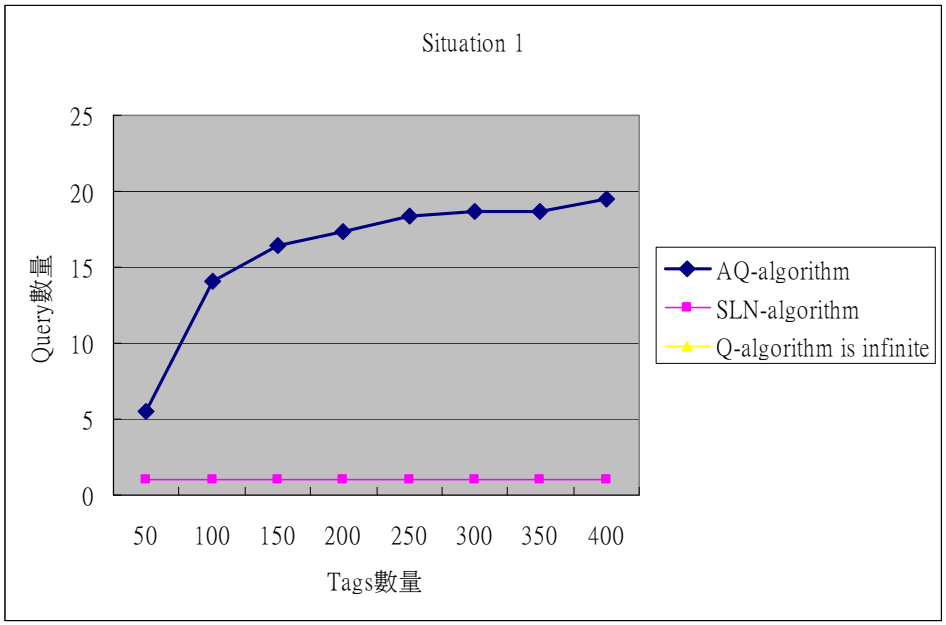

圖 33：笨拙的 Blocker Tag(案例 1)

在案例 1 裡面，因為 Q 演算法無法識別且偵測 Blocker Tag 存在，所以 Q 演算法的時間設定為無窮大。按例 1 的時間比較圖如下圖圖 34。AQ 演算法所花費的時間大於 SLN 演算法，而 SLN 演算法的時間接近於一般情況下的 RFID 系統。

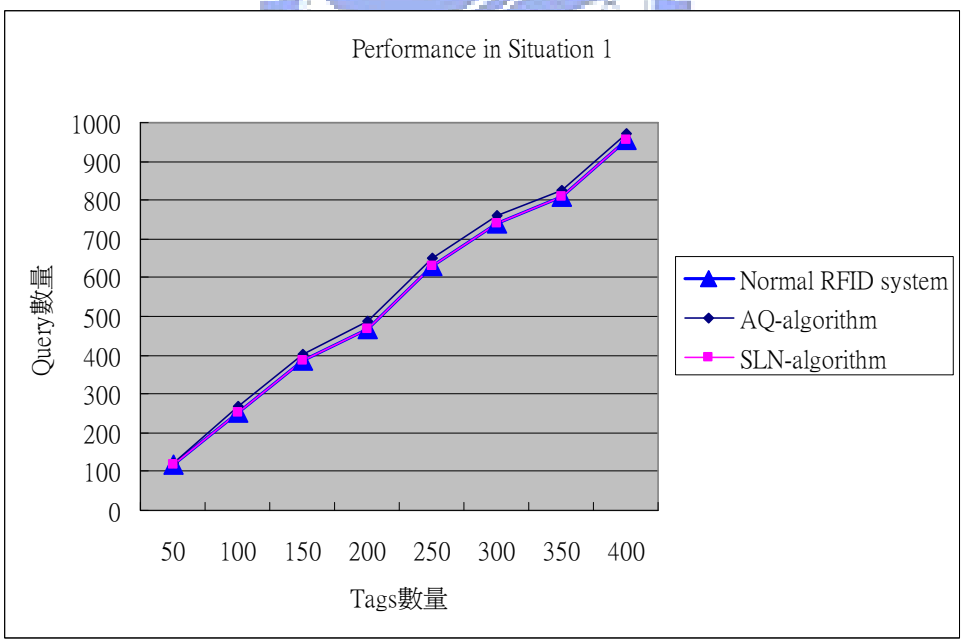

圖 34：各演算法在案例 1 的效能

(2)案例 2：

在情境 2 當中，Q 演算法可以順利的識別所有的標籤編碼，且可以從下圖圖 35 觀察到識別的時間，根據 Blocker Tag 阻擋率的上昇而增加。接著圖 36 是

AQ 演算法在案例 2 中偵測 Blocker Tag 所花費的 Query 數量；然後圖 37 是 SLN 演算法的花費 Query 數量。

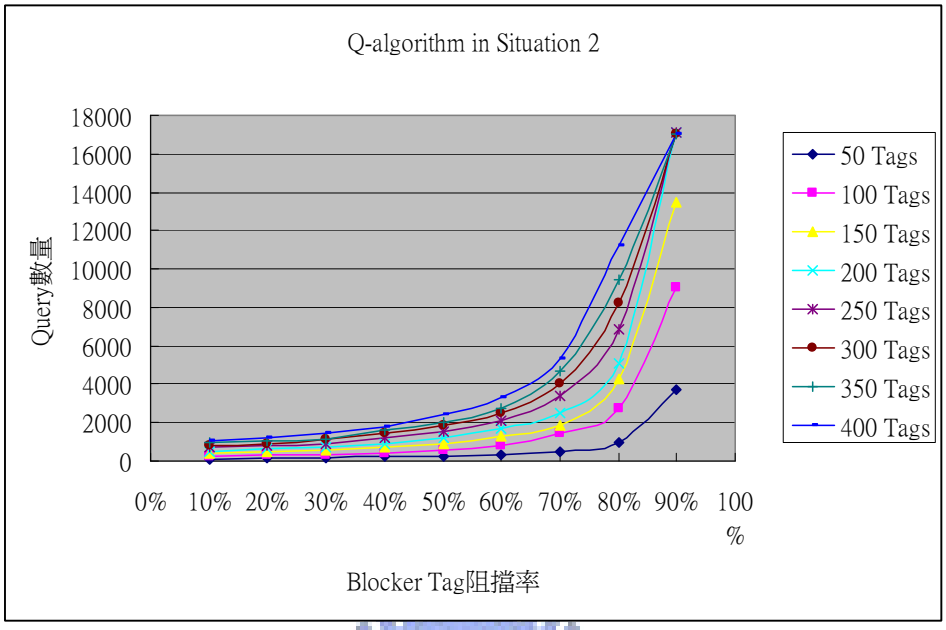

圖 35：Q 演算法在案例 2 的效能

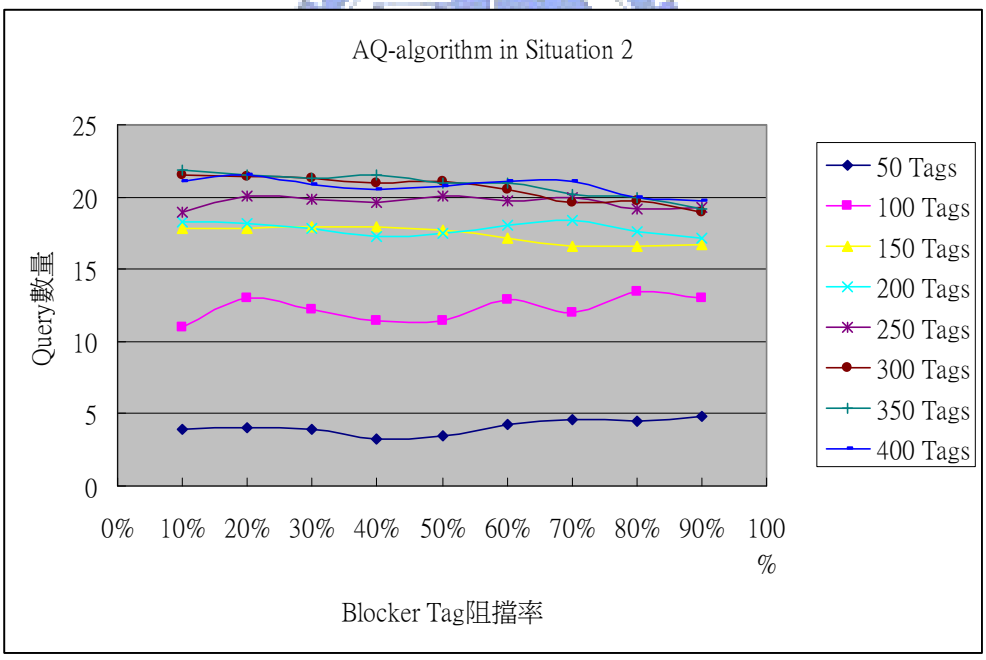

圖 36：AQ 演算法在案例 2 的效能

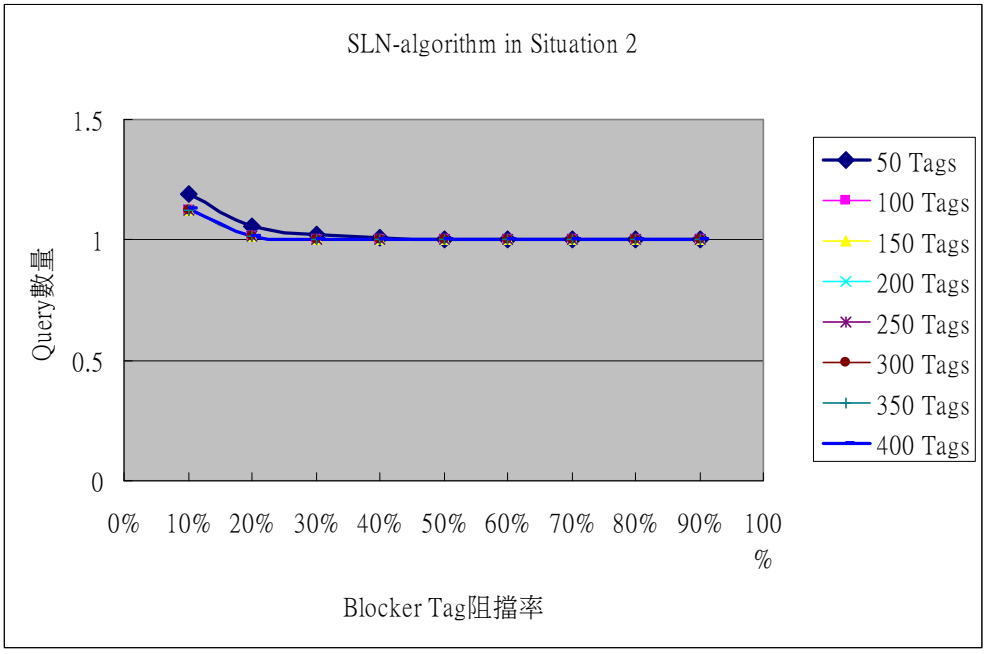

圖 37：SLN 演算法在案例 2 的效能

我們從案例 2 的各實驗結果當中，取出兩個參數值，分別為最大和最小值，作為比較 Query 數量的大小，因為最大最小可以作為最佳和最差。其中，在 AQ 演算法的實驗結果當中，Query 的數量隨著標籤的數量大小增加，被 Blocker Tag 的阻擋率影響比較不大；SLN 演算法在各種阻擋率當中，並不會被影響偵測流程，所以花費的時間和情境 1 相同。下圖是案例 2 的時間比較。

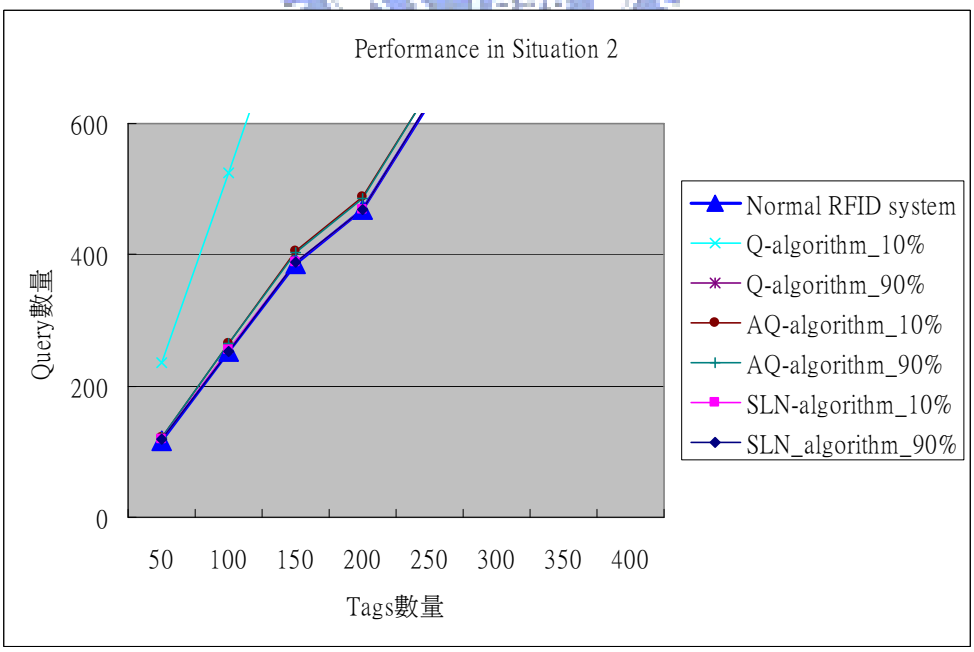

圖 38：各演算法在案例 2 的效能

(3)案例 3：

在案例 3 中，觀察實驗的測試結果(圖 39)，Q 演算法當 Blocker Tags 的阻

擋率處在兩個 Blocker Tag 阻擋率相加為 1 的時候具有最大的攻擊力，因為當相加超過 1 的情況下，根據鴿籠原理，Blocker Tags 本身自己就會進行衝突狀態，而對於相加小於 1 的情況來說，衝突發生的機率沒有相加為 1 來的高，所以使得相加為 1 的時候，Blocker Tags 的攻擊性最強。

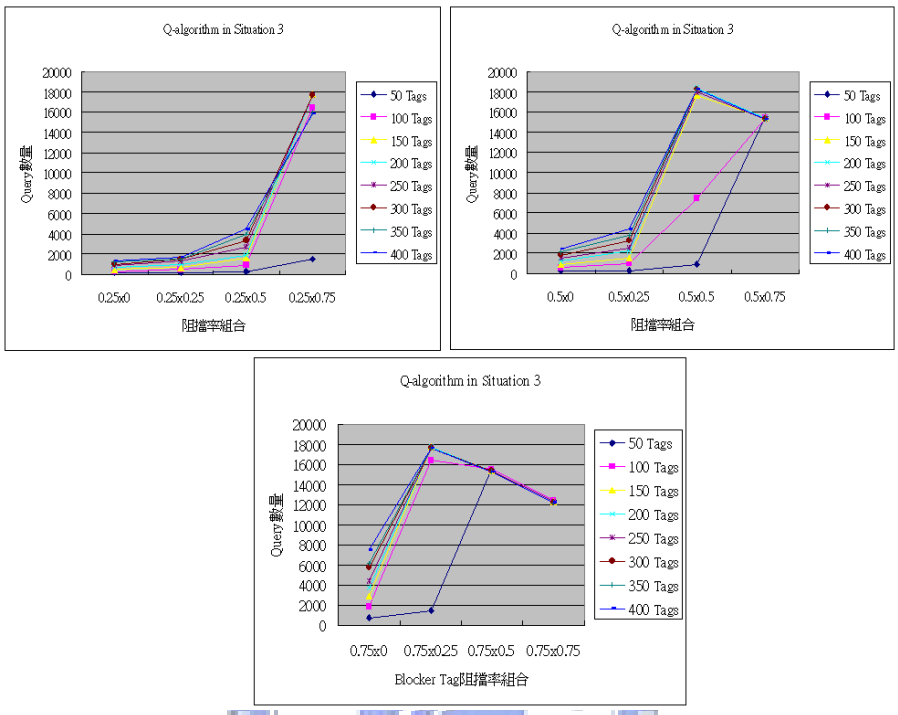

圖 39：Q 演算法在案例的效能

AQ 演算法處在案例 3 的環境中，當有兩個 Blocker Tags 一起進行訊號干擾的情況，實際上是和情境 2 當中的實驗結果相似(如圖 40)，有兩個 Blocker Tags 在一起的時候，會以阻擋率比較高的 Blocker Tag 為主，因為阻擋率比較高的 Blocker Tag 會有比較大的機會傳送回 ACK；但是當兩個 Blocker Tag 的阻擋率都是 1 的時候，就和情境 2 當中的問題一樣，AQ-algorithm 的 ACK 收不到訊號則無法偵測 Blocker Tag 的存在。

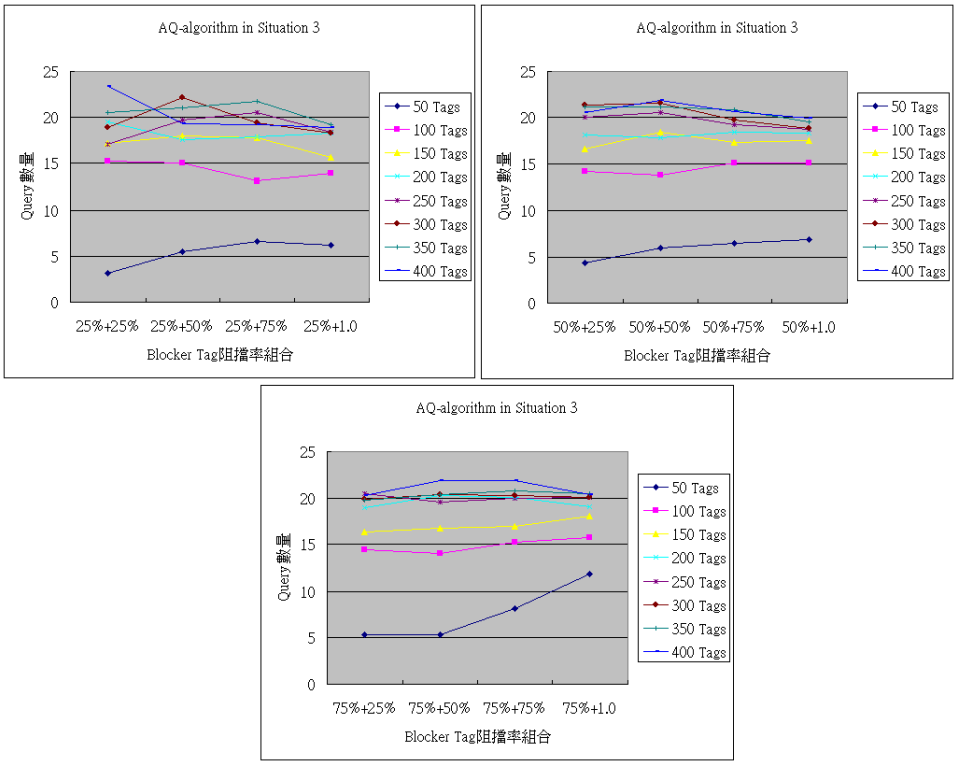

圖 40：AQ 演算法在案例 3 的效能

本篇研究提出的 SLN 演算法，並不會因為受到 Blocker Tag 的數量增加使偵測受到影響，仍舊是使用一個 Query 命令，然後從時間槽中尋找不合法的標籤訊號，但是會有機率產生突發狀況，其情況和情境 2 當中的原因相同，實驗測試結果如下圖圖 41，所以建議進行 SLN 演算法偵測的時候，一次執行兩回合的偵測安全性才能得到保障。

在情境三的測試中，我們不僅採用 2 個 Blocker Tag 進行攻擊，也測試了 3 個 Blocker Tag 同時存在的情況，其測試結果如下圖圖 42 所示，從圖中觀察可以看到，在 Q 演算法的運作時，如果採用的 Blocker Tag 阻擋率是在接近 50%的時候，其讀取的時間花費越久，並且類似於兩個 Blocker Tag 的情況；而 AQ 演算法運行時，如果在阻擋率不是為 100%的情況下，可以順利偵測出 Blocker Tag 的存在，但是在越接近 100%的時候，花費的時間漸漸的增加，其狀況也類似於情境 2 當中的 AQ 演算法，而且受到標籤數量的增加，花費 Query 數量越大；最後在 SLN 演算法的情況中，因為只要有加入的 Blocker Tag 阻擋機率大於 50%，SLN 演算法便會有效的快速偵測出 Blocker Tag 的存在，而在機率小的情況下一般會有誤差發生，但是隨著 Blocker Tag 的數量增加，其 SLN 演算法的偵測效能反而會上升，因為會有更高的機率可以讓 Blocker Tag 暴露它的存在。

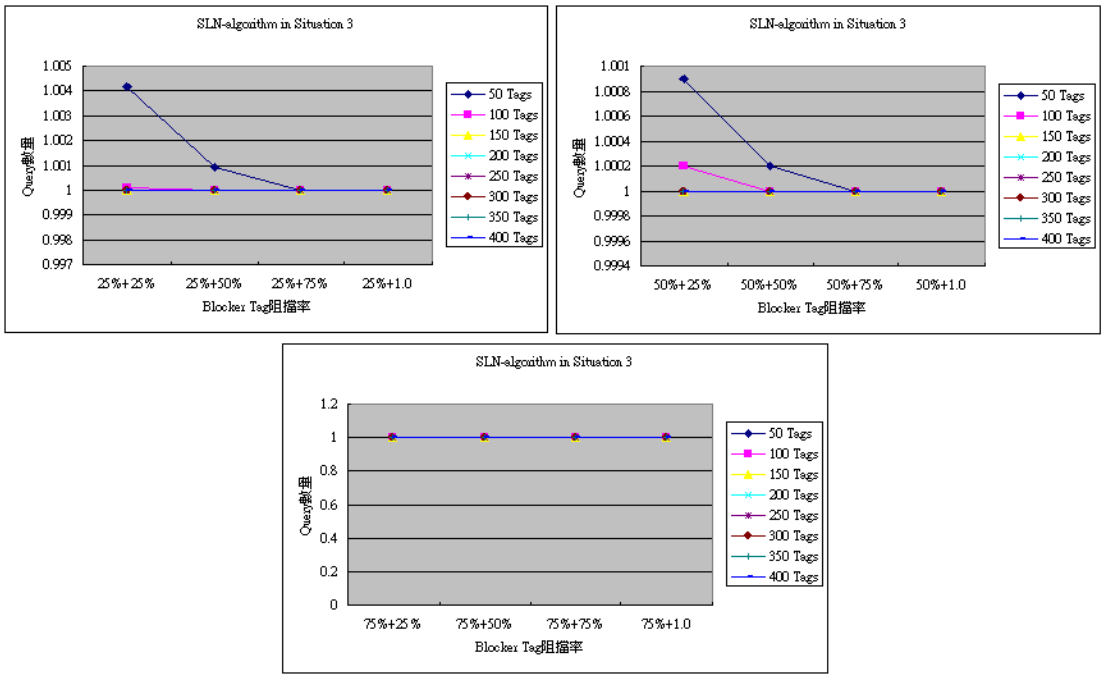

圖 41：SLN 演算法在案例 3 的效能

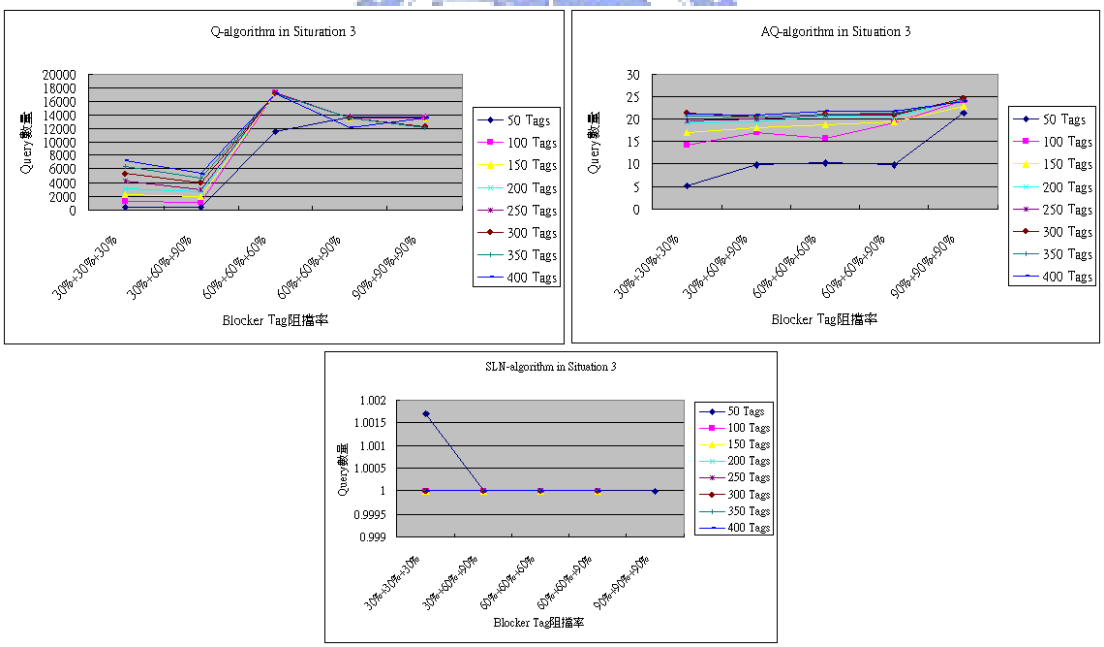

圖 42：各演算法在案例 3-(2)的效能

經過三種案例和三個演算法的測試之後，在下表表 4 是針對此組合的九種狀況做綜合整理。從下表表 4 可以清楚的看到，本研究所提出 SLN 演算法能夠有效的對三種不同情境下的 Blocker Tag 做偵測，而且所花費的時候勝過另外兩種演算法。

表 4：各案例的比較

演算法\案例	案例 1	案例 2	案例 3
Q 演算法	X	△	△
AQ 演算法	O	△	△
SLN 演算法	O	O	O
X：無法偵測 Blocker Tag　O：可以偵測 Blocker Tag　△：有缺陷可以攻擊			

在 RFID 的標準制定的 Q 演算法，只能以時間換取辨識所有標籤的動作，所以在案例 2 和 3 的表格當中，僅以三角形的方式表示之，代表可以在遭到攻擊的情況下運作，但是執行的結果與過程，比一般正常情況下的效能還差；而在其他論文所提到的 AQ 演算法，因為其演算法的動作是先行偵測出是否有攻擊者的存在，而不是同 Q 演算法強行辨識所有的標籤，所以花費的時間上會比較少，但是，AQ 演算法偵測攻擊者的方式有著缺陷的存在，導致攻擊者某些特殊的攻擊模式，AQ 演算法無法有效的偵測，所以在案例 2 和 3 的表格當中，仍舊是以三角形的方式表示，代表有缺陷會被遭受攻擊者攻擊。SLN 演算法也是如同 AQ 演算法的運作，先以偵測出的是否有攻擊者的存在之後，再執行辨識標籤的動作，而且，SLN 演算法完善了 AQ 演算法中無法偵測的缺點，所以在表格當中，給予了 SLN 演算法都是圓圈的評價，能夠有效率的偵測出攻擊者的存在，且在之前的實驗結果中可以發現，偵測出攻擊者的運作，並不會比正常情況下的 RFID 系統花費龐大的額外時間，僅僅需要新增加二、三次命令訊號就可以達到偵測。

Chapter 6 結論

在前面的章節中，我們所提出的 SLN 演算法用來偵測 Blocker Tag 的存在，經由實驗的測試之後，本研究提出的架構確實可以達到效果。在 6.1 節說明本研究所提供的貢獻。在 6.2 節探討未來可以繼續擴充研究的方向。

6.1 貢獻

本研究使用 SLN 演算法進行偵測 Blocker Tag 的機制，防止 RFID 系統讀取範圍內的標籤被訊號遮蔽或干擾。此方法不僅改善了原 EPCglobal class 1 Generation 2 standard 中的缺陷，增進系統在 Inventory 的安全性外，亦為一種新穎、準確且有效率的方式，不像[1]計算 A/Q 數量的比例為門檻，而導致判斷錯誤，使得系統負擔增加，處於持續進行冗長的識別 tags 的 ID。底下列出本論文提出系統的特點：

(1)改進原 EPCglobal class 1 Generation 2 standard[4]的溝通機制
安全性比[4]強韌且能夠簡單地應用在 Gen 2 環境當中，而且是基於 Q 演算法的加強，此外，不需要加強 tags 的運算能力，也不需要新增加任何的命令訊號，便可有效防護 Inventory 的訊號干擾攻擊。
(2)偵測機制有效性
將各種 Blocker Tag 的攻擊情況下，能夠有效地偵測出 Blocker Tag 的存在，比起[4]無法偵測出任何情境的 Blocker Tag，以及[1]無法偵測出情境 2、3 發展出來的 Blocker Tag，本論文的研究可以有效的在情境 1、2 和 3 中偵測出有 Blocker Tag 的存在。此外，本研究所提出的偵測機制，不像[1]以機率的方式判斷 Blocker Tag。
(3)偵測效能快速
不使用大量的統計資料且只需要增加二個命令訊號，所以跟[1]比起持續加大 Q 值以計算 A/Q 比例、統計所有時間槽中的狀態，本研究提出的偵測方法執行快速效能。

6.2 未來工作

本研究提出的機制除了可以解決三種情境中的 Blocker Tag 干擾攻擊外，底下列出幾點未來研究將會應用或擴充的部份：
(1)Blocker Tag 具有接收命令並且進行解析命令的處理能力，當 Blocker Tag 看到偵測機制開始啟動的時候，可以做出躲避其偵測機制的攻擊，比如說

Blocker Tag 只針對上一回合的時間槽範圍內進行攻擊。

(2)Blocker Tag 是否有更多種攻擊 RFID 系統的案例，以各式各樣的案例再進行更深一步的探討與研究，並且針對各案例提出防護的機制。

(3)使用其他有效資訊偵測 Blocker Tag 的存在，例如：隨機抽樣的方式進行判斷，並且以數學算式證明之。

(4)本論文是使用模擬的方式來實驗，未來可以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] Wen Chen and Wen-Nung Tsai, “RFID privacy protect using blocker tag with anti blocker tag scheme,” NCTU 2009

[2] Anne Huang, Jwu-Sheng Hu, Yu-Lun Huang, “The Design and Implementation of a Secure Multi-level Authentication Protocol for RFID Systems,” NCTU 2005

[3] Li-an Lee, Shiuh-Pyng Shieh, “Protecting User Privacy with Dynamic Identity-Based Scheme for Low-cost Passive RFID Tags,” NCTU 2005

[4] EPC global, “EPC™ Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz – 960 MHz Version 1.1.0,” http://www.epcglobalinc.org

[5] Younghwa, An Soohyun Oh, “RFID System for User's Privacy Protection,” Communications, 2005 Asia-Pacific Conference on, pp. 516-519

[6] A. Juels, R. L. Rivest, and M. Szydlo, “The blocker tag: Selective blocking of rfid tags for consumer privacy,” *Atluri, ed. 8th ACM Conference on Computer and Communications Security*, vol. In V, pp. 103--111, 2003.

[7] A. Juels and J. Brainard. “Soft Blocking: Flexible Blocker Tags on the Cheap,” In S. De Capitani di Vimercati and P. Syverson, eds., Workshop on Privacy in the Electronic Society (WPES), pp. 1—7. 2004.

[8] Harald Vogt. “Efficient Object Identification with Passive RFID Tags,”. in Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '02), Hammamet, Tunisia, October 2002

[9] Jing Nie; Wing Shing Wong. “Optimized anti-collision techniques in RFID systems,” Mobile Wireless Communications Networks, 2007 9th IFIP International Conference on 19-21 Sept. 2007 pp:36 - 40

[10] Jaemin Park, Junchae Na and Minjeong Kim, Terminal Application Development Team, Terminal Laboratory, R&D Group, KTF, “A Practical Approach for Enhancing Security of EPCglobal RFID Gen2 Tag,“ in the Future generation communication and networking (fgcn 2007), pp.436-441

[11] Xiaodong Deng, Mengtian Rong, Tao Liu, Yong Yuan, Dan Yu, “Tag Count Frame Slotted Aloha: A Novel Anti-Collision Protocol in RFID Systems,” in: Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE on 1-14 May 2008, pp2666-2670

[12] Jianwei Wang; Dong Wang; Yuping Zhao; Korhonen, T, “Fast Anti-Collision Algorithms in RFID Systems,” in: Mobile Ubiquitous Computing, Systems, Services and Technologies, 2007. UBICOMM '07. International Conference on 4-9 Nov. 2007, pp 75-80

[13] Yong-Sik Choi; Soo Han; Seung Ho Shin. “A design of e-ID authentication protocol in Gen2 environment,” in the Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on Volume 1, 17-20 Feb. 2008, pp246 - 251

[14] Huiyun Lim, Fengqi Yu, Yun Hu, Chinese Univ. of Hong Kong, Shenzhen, “A Solution to Privacy Issues in RFID Item-level Applications,” in the Integration Technology, 2007. ICIT '07. IEEE International Conference on 20-24 March 2007, pp459-464

[15] Jae Sung Choi; Hyun Lee; Daniel W. Engels; and Ramez Elmasri. “Robust and Dynamic Bin Slotted Anti-Collision Algorithms in RFID System,” RFID, 2008 IEEE International Conference on April 16-17, 2008, pp191-198,

[16] Donghwan Lee, Kyungkyu Kim, and Wonjun Lee, “Q+-Algorithm: An Enhanced RFID Tag Collision Arbitration Algorithm,” Springer-Verlag: Lecture Notes in Computer Science, 2007

[17] EPCglobal http://www.epcglobalinc.org/standards/

[18] Platform for Privacy Preferenceshttp://www.w3.org/P3P/

[19] 何丁武, 羅濟群,”用於 Auto-ID 環境下減少碰撞的機制,” NCTU 2004

[20] 曾煜棋，潘孟鉉，林致宇，無線區域及個人網路:隨意及感測器網路之技術與應用，經瑋國際，台北市，2006。