國立交通大學

資訊科學與工程研究所

碩 士 論 文

安 全 且 快 速 的 R F I D 雙 向 認 證 協 定

Secure and Efficient Mutual Authentication Protocol for RFID systems

研 究 生：廖偉志

指導教授：蔡文能　教授

中 華 民 國　壹 佰　年 六 月

安全且快速的 RFID 雙向認證協定

Secure and Efficient Mutual Authentication Protocol for RFID systems

研 究 生：廖偉志　　Student：Wei-Chih Liao

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 Network Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2010

Hsinchu, Taiwan, Republic of China

中華民國壹佰年六月

安全且快速的 RFID 雙向認證協定

學生：廖偉志　　指導教授：蔡文能

國立交通大學資訊科學與工程研究所碩士班

摘要

RFID 是一種非接觸式的自動識別技術，具有快速且無方向性要求的辨識能力，應用層面相當廣泛，被視為未來社會中的基礎建設之一。但是，因為 RFID 標籤先天上硬體資源的限制，使得 RFID 仍然存在著許多安全問題，例如: 隱私問題、竊聽、標籤複製、標籤追蹤等。因此，許多的研究相繼提出各種方法，試圖解決 RFID 的安全問題，而 RFID 認證協定即為其中一種。

RFID 認證協定的作法是讓讀取器和標籤在進行存取之前，雙方皆必須檢驗對方的身分，只有當雙方的身分皆為合法時，讀取器和標籤之間才能夠做進一步的存取動作，藉此方法來確保 RFID 應用的安全。目前的 RFID 認證協定常見的問題有安全性不足、標籤運算負擔過重和後端伺服器查詢標籤時間過長等，使得 RFID 認證協定無法有效的解決 RFID 的安全問題。

因此，本論文將提出兩個 RFID 認證協定，針對上述常見的 RFID 認證協定問題做出改善，確保 RFID 的安全。經由分析與比較的結果，證實本論文所提出的兩個 RFID 認證協定，不僅具有足夠的安全性，可以抵抗多種 RFID 攻擊，在效能方面，不但降低標籤的運算負擔，且大幅的減少後端伺服器查詢標籤所需的時間。

Secure and Efficient Mutual Authentication Protocol for RFID systems

Student：Wei-Chih Liao　　Advisor：Wen-Nung Tsai

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

Abstract

Radio Frequency Identification (RFID) is a non-contact 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, because RFID tag on the inherent limitations of hardware resources, RFID has various security threats like privacy problem, eavesdropping, tag cloning, tag tracing, etc. To address these problems, lots of solutions had been proposed. RFID authentication protocol is one kind of these solutions.

The point of RFID authentication protocol is to enforce the security policy between legal RFID tags can the legal RFID readers. RFID authentication protocols provide a mutual authentication mechanism to protect RFID system. But most of them have some drawbacks. These problems include the security not strong enough to resist various RFID attacks, the computation overhead of RFID tag is too high, and the back-end server spends too much time for searching RFID tag data, and etc.

In this thesis, we proposed two RFID authentication protocols to overcome these drawbacks. In our analysis, the results have demonstrated that our RFID authentication protocols can resist most common RFID attacks and are more effective than most RFID authentication protocols.

誌謝

首先，對於自己能夠進入交通大學就讀資訊工程研究所，我深深的感到萬分榮幸，我想感謝在準備研究所考試時，陪伴在我身邊的家人與朋友，家人的寄望和朋友的支持是我能夠努力不懈的動力來源，沒有你們，我就沒有這份榮耀，謝謝你們。

在碩士班的生涯中，我覺得自己很幸運可以進到蔡文能教授的網路應用與網路安全實驗室，感謝蔡文能老師的指導，讓我在這兩年中，學習到非常非常多的專業知識和待人處事的態度；感謝實驗室的學長、學姐，當我有疑難雜症時，可以提供我寶貴的意見；感謝實驗室的同學，碩一一起在實驗室打拼的日子，將成為我在交大最珍貴的回憶；感謝實驗室的學弟，為實驗室帶來了新的朝氣；感謝我的好朋友們，小宏、大吉、頹哥、大Ｂ、豬豬、治明、享岳、蓓蓓、吳郁聆、點點，在空暇之餘能夠開開心心的與你們到處出遊，度過各種節日，是我每天最期待的事情，謝謝你們讓我的研究生活增添了許多許多的歡樂。

一就學就開始擔心的畢業論文，在這兩年的努力之下，突破各種困難與問題後，終於完成了！ 誠心感謝蔡文能老師的指導，在您的指引之下，我才能夠完成這份畢業論文，另外，感謝實驗室的同學，謝謝你們提供的意見，有了你們的意見，才能為我的畢業論文思考出更多改善的方法。

想感謝的人太多太多了，真心的感謝在我開心時能夠與我分享快樂，在我難過時能夠為我加油鼓勵的人們，謝謝你們的陪伴與幫助，感恩。

目錄

圖目錄

表目錄

第一章 緒論

無線射頻識別(Radio Frequency Identification，RFID)，是一種非接觸式的自動識別技術，利用射頻信號自動識別目標物品並獲取其相關資料進而完成工作，在此過程中不需要任何人員介入，可以應用在惡劣的環境底下。RFID 具有快速且無方向性要求的辨識能力，足以取代傳統的條碼(Barcode) 成為現今社會最適合的辨識管理系統。但是，RFID 是一種發展中的技術，目前在安全性方面尚存在一些問題有待解決。在本章節中，將先於 1.1 節介紹本論文的研究動機，接著在 1.2 節介紹本論文的研究目標，最後在 1.3 節介紹本論文的組織架構。

1.1 研究動機

RFID 標籤在硬體資源方面有所限制，因此引發許多威脅到安全性的問題產生，例如:隱私問題、竊聽、標籤複製、標籤追蹤...等。為了解決這些問題，陸陸續續有各種安全防護措施被提出來，其中，有一類型的安全防護措施是基於 EPCglobal RFID Class1 Gen2 標準底下的 RFID 認證協定，這類型的安全防護措施目的是在於建立一套安全的認證流程，提供 RFID 系統抵抗各種 RFID 攻擊的能力。

RFID 標籤為了要降低生產成本，在電路設計上不會提供太複雜的運算，因此大部分的 RFID 認證協定的特色在於標籤方面使用簡單且快速的運算並且可以達到期望的安全需求，但是並沒有考慮到後端伺服器查詢標籤資料的負擔。在大部分的 RFID 認證協定中，後端伺服器查詢標籤資料的方式都是憑藉著一筆標籤所給定資料到資料庫中查詢，此時後端伺服器會將該筆資料與資料庫中每一筆項目的某個欄位內容逐一比較，檢查是否相同，如果相同才代表找到該標籤的相關資料。若考量到同時有數以百計的標籤進行存取或標籤數量眾多的環境，例如:賣場、圖書館、倉儲...等，此時後端伺服器查詢標籤資料所花費的時間將會變成 RFID 系統效率的關鍵之一。

其中有少部分的 RFID 認證協定考量到後端伺服器查詢標籤資料的負擔，並且提出方法減少後端伺服器在查詢標籤資料時所需的時間，但是不幸地，該方法卻引發其他的安全問題。

1.2 研究目標

本論文將提出一套 RFID 認證協定，在安全性方面，提供 Reader-to-Tag 和 Tag-to-Reader 的雙向認證機制，確保能夠抵抗各種 RFID 攻擊以達到安全性需求；在效能方面，考量到標籤的運算量負擔，標籤在認證過程中只需要使用 XOR 運算即可快速且安全的完成認證，此舉將降低標籤電路設計的複雜度進而減少標

籤的生產成本，另外，考量到後端伺服器查詢標籤資料的負擔，後端伺服器不需要再憑藉一筆標籤所給定的資料與資料庫中的所有項目一一比對才能找到標籤的相關資料，而是藉由標籤所給定的 Index 直接到資料庫該標籤所在的項目中取得該標籤的相關資料，此舉將大幅地減少後端伺服器查詢標籤資料所花費的時間。

1.3 論文架構

本論文的組織架構如下:「第一章 緒論」介紹本論文的研究動機、研究目標和論文架構。「第二章 背景知識」介紹 RFID 系統、RFID 標準和 RFID 的安全議題。「第三章 相關研究」將介紹六個 RFID 認證協定，並且依照後端伺服器查詢標籤資料所花費的時間將其分類，先介紹 O(N) Database 查詢時間的 RFID 認證協定，再介紹 $O(N/2^m)$ Database 查詢時間的 RFID 認證協定，最後介紹 O(1) Database 查詢時間的 RFID 認證協定，其中 N 為 RFID 系統中的標籤數量，m 有待第三章詳細說明。「第四章 RFID 認證協定設計」介紹本論文所提出的 RFID 認證協定，Secure and Efficient Mutual Authentication Protocol for RFID systems (SEMAP)，並且基於安全與效率的考量，提出兩種版本，分別為 Secure and Efficient Mutual Authentication Protocol for RFID systems version 1(SEMAPv1)和 Secure and Efficient Mutual Authentication Protocol for RFID systems version 2(SEMAPv2)。「第五章 分析與比較」先針對安全性進行分析與比較，再針對效能進行分析與比較。「第六章 結論」先為本論文做個結論，最後以探討未來的研究工作內容作為結尾。

第二章 背景知識

RFID(Radio Frequency Identification)，是一種利用無線射頻信號傳送識別資料的技術，藉此達到目標物品身分辨識的目的。RFID 相較於傳統的條碼可以更加快速且方便的完成身分識別的工作，基於 RFID 的各項優點使得 RFID 被廣泛的應用在各種領域之中。因此，EPCglobal 組織為了推廣 RFID，制定了許多 RFID 相關的標準。在本章節中，將先於 2.1 節介紹 RFID 系統，包含 RFID 系統元件、RFID 運作流程和 RFID 應用環境，接下來在 2.2 節介紹 EPCglobal 組織，包含 EPCglobal Network、EPCglobal Class1 Gen2 Air Interface protocol 和 EPCglobal 所提供的安全防護機制，最後在 2.3 節探討 RFID 相關的安全議題，其中包括隱私問題和常見 RFID 攻擊種類。

2.1 RFID 系統

RFID 是一種具有未來發展性的技術，此項技術改善了接觸式系統的缺點，利用無線射頻信號的方式傳送資料，因此感應器不需要與接收器有所接觸即可完成資料的交換，且此種傳輸方式無方向性的需求，這項特性帶來了偌大的便利性，因此，RFID 被視為是未來一種無所不在的社會基礎建設。

2.1.1 RFID 系統元件

RFID 系統元件主要分成三個部分，分別是: 標籤(Tag)、讀取器(Reader)和後端伺服器(Back-end Server)，接下來將依序介紹此三個元件。

1. 標籤

圖 1: 悠遊卡

資料來源: 維基百科

標籤是由晶片和天線所組成，其中晶片包含邏輯控制單元、記憶體和收發器，可以用來進行存取過程中的相關運算；而天線用於接收讀取器發送的無線射頻資

料和傳送本身的識別資料給讀取器。另外，具備儲存裝置可以儲存物品的識別資料，例如: 產品身分、產品種類、產品製造商...等，當讀取器發出要求時，標籤便會將這些識別資料傳送回去。標籤之樣本實體，可參照上圖 1，其目的是作為搭乘台北大眾捷運的繳費工具。

標籤根據電力提供方式的不同可以分成主動式、半主動式和被動式三種型態，而在 EPCglobal 標準下是採用被動式標籤。以下分別介紹此三種不同型態的標籤:

(1) 主動式標籤:

主動式標籤具備電池提供能量，可以利用自有電力在標籤周圍形成有效活動區，主動偵測周遭有無讀取器發射的呼叫信號，並將自身的資料回送給讀取器。這類型的標籤通常具有較強大的運算能力和較長的傳輸距離，但是標籤的生命周期受到電池所控制，在生產成本上的價格也較為昂貴。主要用於軍事、醫療和工業上。

(2) 半主動式標籤:

半主動式標籤也內建電源，可以利用內部能量偵測標籤周圍環境，但是內部能量不足以用來通訊，必須仰賴讀取器提供的無線電磁波，標籤收到無線電磁波後轉換成自身電力才能回送訊號給讀取器。其運算能力和傳輸距離皆介於主動式標籤和被動式標籤之間。主要用於監測周遭環境溫度或是震盪情況。

(3) 被動式標籤:

被動式標籤沒有自己的電池提供能量，標籤的主要能量來自於讀取器所發出的無線電磁波，標籤收到波動後產生微量電流，透過此微量電流讓晶片有運算的能力並且回送訊號給讀取器。這類型標籤的運算能力較弱且傳輸距離也較短，但是標籤的生命周期不會受電池控制，在生產成本上也較為便宜。主要用於動物晶片、智慧卡、防盜管理...等。

	主動式	被主動式	被動式
電力來源	內部電池	1.內部電池 2.電磁感應	電磁感應
傳輸距離	5~100 公尺	---	5 公尺以內
記憶體容量	64k~228k bits	---	64~8k bits
生命週期	電池壽命	---	長達十年
生產成本	最高	中	最低

表 1: 三種不同型態標籤的優缺比較

2. 讀取器

圖 2: RFID 讀取器
資料來源: adcnordic.com

讀取器可以透過無線電波的對標籤進行讀取或者是寫入的動作。讀取器由收發器、微處理器和記憶體所組成，其中收發器由發送器和接收器組成，負責發送信號給標籤以及接收標籤的回應；微處理器負責相關運算的處理；記憶體則負責儲存讀取器配置的相關參數和讀取到的標籤資料。讀取器之樣本實體，可參照上圖 2。

3. 後端伺服器

後端伺服器提供資料庫負責儲存標籤的相關資料，通常包括: 名稱、價格、位置、製造商、擁有者...等，依照應用需求的不同，資料庫中所存放的標籤資料也有所不同，舉例來說，當 RFID 應用在圖書館環境時，後端伺服器的資料庫主要是紀錄圖書館書本的租還時間以便於書本管理；而 RFID 應用在賣場環境時，後端伺服器的資料庫主要是紀錄賣場商品的價格以便於商品結帳。

2.1.2 RFID 運作流程

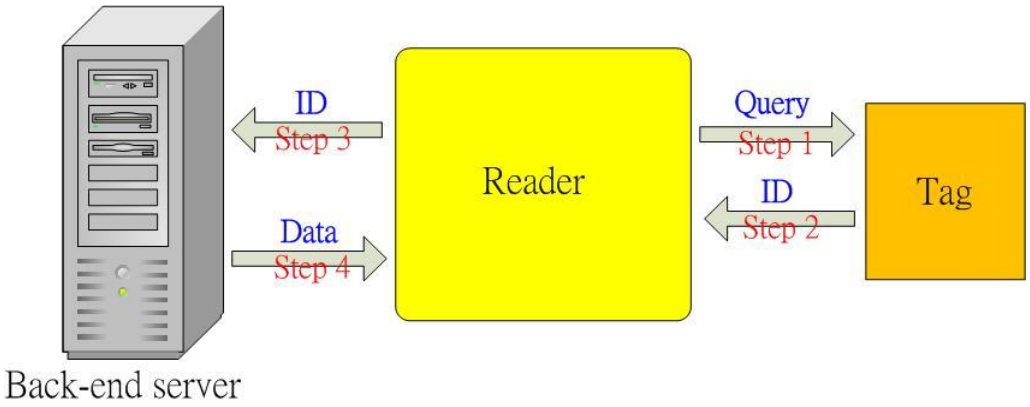

圖 3: RFID 運作流程

上圖 3 為 RFID 運作流程的示意圖，接下來將初步的介紹 RFID 的運作流程:

Step 1: 讀取器透過天線發出一定頻率的無線電波給周遭的標籤，其所代表的意義為讀取器向標籤發出存取的請求。

Step 2: 位於感應範圍內的標籤收到信號後，回送自身的識別資料給讀取器，例如: ID。

Step 3: 讀取器收到標籤送回來的識別資料 ID 後，即傳送回後端伺服器處理。

Step 4: 後端伺服器藉由讀取器所傳送的識別資料 ID 到資料庫中查詢，若查到該標籤的相關資料，則將該資料傳送給讀取器。

如此一來，讀取器就可以辨識物品的身分，並取得物品的相關資料進而完成工作。

2.1.3 RFID 應用環境

RFID 具有快速且方便的身分識別能力，能夠改善傳統的物品管理機制並且大大的提升管理效率，因此已被廣泛地應用在各種領域之中。目前常見的 RFID 應用如下:

門禁管制

無論是學校、公司和住家等環境都需要徹底掌握進出人員的身分，RFID 在這項應用底下可以作為電子鑰匙，只有具有合法標籤的人員才能進出該場所以達到人員控管的工作。2006 德國世界杯足球賽建入 RFID 標籤作為世足賽的門票，成為全球首件應用此項技術的大型體育賽事，而建入 RFID 標籤的世足賽門票還可兼做電子錢包用途，觀眾可以用來購買食物、寄放物品和停車等加值服務。

聯合票證

以台北捷運的悠遊卡而言，除了作為搭乘捷運的繳費工具之外，還可以搭乘臺鐵、公車、計程車...等交通工具，另外還可以在特約商店中使用，例如: 7-11、全家、萊爾富...等商店，而使用的範圍持續在擴大之中，使得悠遊卡的功能越來越多元，增添許多便利性。

倉儲管理

在圖書館或大賣場的環境底下，商品的數量難計其數，傳統的管理方式在這種情況下就顯得相當耗費人力與時間，而 RFID 的快速且方便的識別能力恰好能改善這項缺點，提升商品管理效率的同時意味著減少人力資源的耗費，可以帶來更多的產出。

防盜應用

RFID 在資料更新、儲存容量、使用壽命和安全性方面都比傳統的條碼更具優勢，此項特性可以用來作為店家商品的防盜機制，例如在每項商品中都建入 RFID 標籤，只有結完帳的商品經過店家更新標籤內容後才可帶出商店外，如此一來，尚未結帳的商品要離開時，門口的讀取器偵測到就會通知店家達到防盜的效果。

交通運輸

以高速公路電子收費系統(ETC)而言，就是利用 RFID 快速且無方向性需求的辨識能力，當車輛在限定的速度內通過高速公路收費站時，RFID 讀取器會辨識車上掛載的 RFID 標籤並完成該車輛繳費的工作，如此一來可以減少傳統高速公路收費站的人力收費時間以利車輛快速通過，進而達到高速公路暢通的目的。

寵物晶片

寵物晶片代表寵物獨一無二的身分證，透過 RFID 的特色可以實現寵物識別的自動化，對於寵物走失、流浪狗管理等方面都有相當大的幫助。晶片還可以根據動物的使用部位，可以分為耳環、腳環和項圈等標籤。

2.2 RFID 標準

EPCglobal 的前身為麻省理工學院所成立的自動識別中心(Auto-ID Center)，此中心致力於研發適用於供應鏈的自動識別系統，目的是創造出如 Internet 般無國界且便利的使用環境，並且將 RFID 技術應用在 EPC 系統上。在 2003 年 10 月，Auto-ID Center 正式宣告結束，並將 EPC 系統轉交給 EPCglobal Inc.，由 EPCglobal 接手繼續 EPC 的技術研發與全球推廣工作，其中所代表的意義為 EPC 正式從學術研究領域進入商業應用領域。

EPCglobal Inc.負責 EPC 的註冊、管理和導向 EPC 發展成為全球通用的標準。而另一項重要的任務為結合 GS1 全球會員組織共同推廣 EPC 標準，藉由 EPC 標準的導入與應用，提升交易夥伴使用 RFID 的效益，並透過持續發展 EPC 網路標準的相關構件，開放企業參與，促使全球各地的產業共同採用 EPC；而原自動識別中心實驗室(Auto-ID Labs)歸劃在 EPCglobal 組織之下，聯合世界上六大著名研究學府，分別為美國麻省科技學會(Massachusetts Institute of Technology)、英國的劍橋大學(The University of Cambridge)、澳洲阿德萊德大學(the University of Adelaide)、日本慶應大學(Keio University)、中國復旦大學以及瑞士聖迦南大學(the University of St. Gallen)，負責 EPC 相關前瞻技術的研究。

2.2.1 EPCglobal Network

EPCglobal Network 又稱為物聯網，是利用現有的網際網路架構，在全球建立一個巨大的物品資訊交換網路，在此網路底下流通的物品皆具有唯一的產品電

子碼，藉由 EPCglobal 網路架構相關元件，使這些具有產品電子碼的物品可以在網路上準確的定位及追蹤，並且為每一個物品建造一份電子履歷，使得偽造物品無法流通。

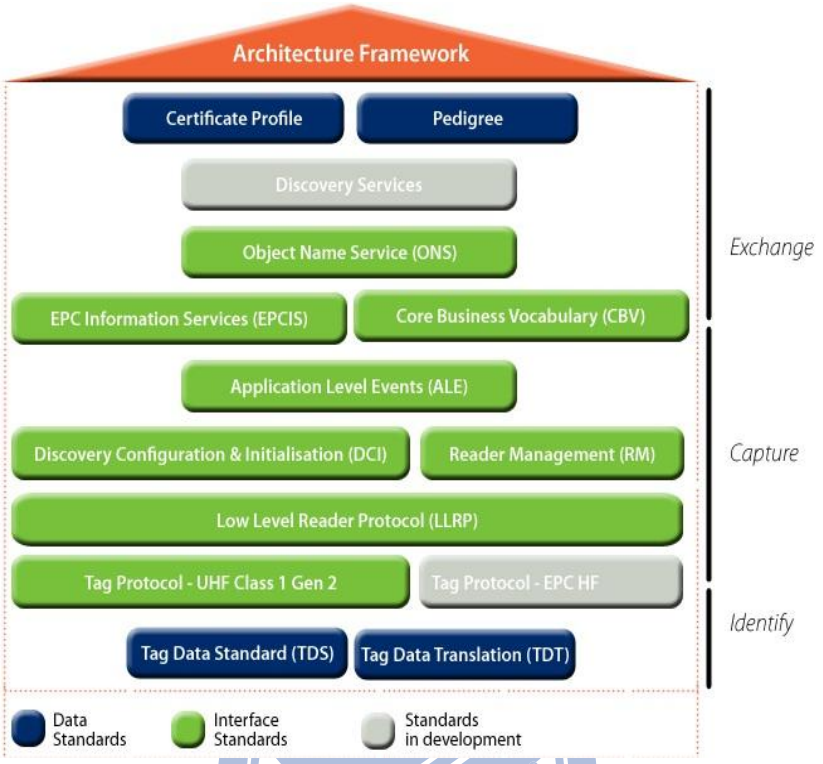

圖 4: EPCglobal Network 架構圖

資料來源: http://www.gs1tw.org

上圖 4 為 EPCglobal Network 的架構圖，其中各個元件皆有相關的制定標準，其標準的細節資料可以參考來源網站。整個 RFID 系統的運作過程由 EPC 結構框架建構成三層網路架構，分別是辨識(Identify)、擷取(Capture)和資訊交換(Exchange)，分別介紹如下:

辨識層:

辨識層定義 EPC 實體物件的交換標準，確保當某一使用者將實體物件送至另一使用者時，接收者能夠知道此實體物件的電子產品碼並且正確的判讀。

擷取層:

擷取層主要定義讀取器資訊讀取解碼格式(Reader Protocol、Low Level Reader Protocol)，以及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。

資訊交換層:

資訊交換層定義 EPC 資料交換標準，讓單一使用者與另一位使用者之間可

以藉由 P2P 互動來共用 EPC 資料，並得以使用 EPCglobal 核心服務與其他共享服務以增添便利性。

2.2.2 EPCglobal Class1 Gen2 Air Interface protocol

EPCglobal Class1 Gen2 標準是由全球 60 多家廠商共同開發的超高頻(UHF)開放式新標準，整合了 Class0、Class1、ISO18000-6°和 ISO18000-6B 四個標準，於 2004 年底通過 EPCglobal 審核，並於 2006 年 6 月通過國際標準組織(ISO)認定，納入 ISO18000-6C 標準。此一新標準具有較全面的框架結構和較強的功能，能夠在高密度讀取器的環境中工作，且符合全球管制條例，在標籤讀取正確率、標籤讀取速度、安全性和隱私性上都有所提升，而 Gen2 的主要特點如下:

開放式的標準

IP 協議的單位免收專利費，允許廠商生產基於該標準的商品，如讀取器和標籤，意味著更多的技術供應商可以在免專利費的情況下根據此標準生產符合供應商、製造商和用戶端所需要的產品，也減少了用戶端佈署 RFID 系統的費用，可以吸引更多用戶使用 RFID 技術。

標籤尺寸小但儲存量大

標籤的尺寸縮小到現有版本的一半至三分之一，提升了標籤的實用性並且可以滿足各種應用需求，例如可以更容易的縫在衣服的接縫裡、夾在紙板之中、成型在塑膠內和整合在顧客的包裝設計中。另外，標籤的儲存能力增加，標籤中可以存放更多的資料，藉由這些資料可以提供更安全的保護機制。

不同廠商產品的兼容性

目前 RFID 有兩個技術標準陣營，一個是總部設在美國麻省理工學院的 Auto-ID Center，另一個是日本的 Ubiquitous-ID Center。日本 UID 標準和歐美的 EPC 標準在無線頻率和應用領域等地方皆不相同，Gen2 標準保證了不同製造商的設備之間具有良好的相容性。

設置了 Kill 指令

Gen2 標準讓使用者有控制標籤的能力，即使用者可以透過下達 Kill 指令讓標籤自行永久失效，如此一來可以有效的防止標籤被非法的存取，增加了安全性，也減輕使用者的隱私困擾。

更廣泛的頻譜和射頻分布

Gen2 協議的頻譜和射頻分布比較廣泛，減少了與其他無線電設備的干擾問題。另外，也解決了 RFID 在不同國家使用不同頻譜的問題。

另外，Gen2 標準也定義了標籤的硬體規格，標籤的記憶體由四個儲存庫(bank)所組成，分別如下(下圖 5 為標籤的記憶體配置圖):

Bank 00 – The Reserved memory:
包含兩個安全訊息，分別為 32-bits Kill password 和 32-bits Access password。

Bank 01 – The EPC memory:
包含 EPC code、Protocol control 和 CRC-16 等必要訊息。

Bank 10 – The TID memory:
包含 32-bits TID，此 TID 的數值代表著標籤相關的識別資料。

Bank 11 – The User memory:
包含使用者的資料，此資料也可以由使用者本身自己設定。

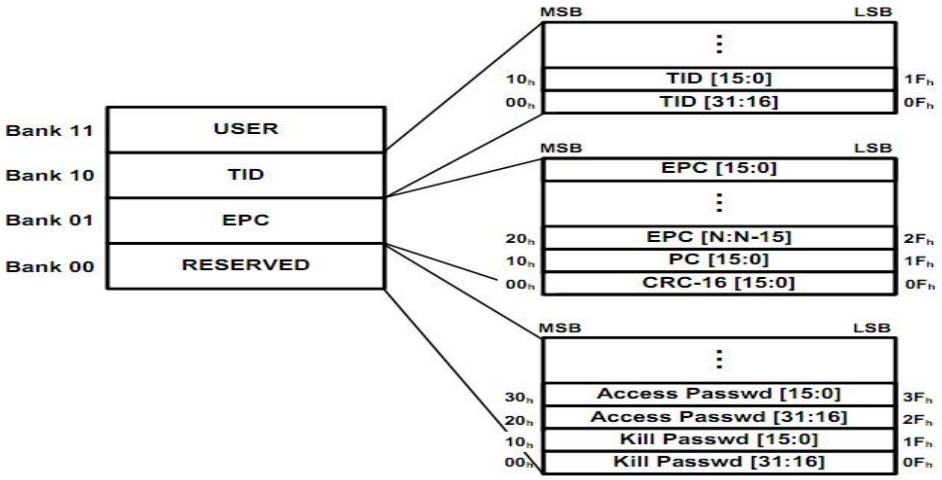

圖 5: 標籤的記憶體配置圖
資料來源: http://www.epcglobalinc.org

Gen2 標準中允許讀取器可以一次讀取多個標籤，但實際的運作過程是以一次讀取單一個標籤的方式進行，藉由讀取端以切割時間和多次掃描的交互配合下達到一次讀取多個標籤的功能。切割時間的目的是讓當標籤的識別資料中的某些位元符合該時間槽(Timeslot)所指定的位元時，允許該標籤通訊並傳輸資料；多次掃描的目的是讓每一個標籤都有機會允許通訊並且傳送資料。

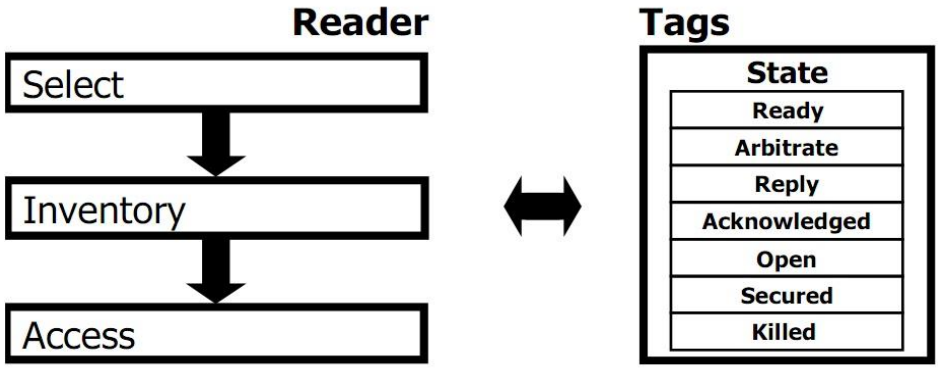

圖 6: 讀取器運作的三個階段
資料來源: http://www.epcglobalinc.org

上圖 6 描述讀取器在存取一群標籤時必須經過三個階段，依序是選擇(Select)、盤點(Inventory)和存取(Access)，每一個階段都是由一個或多個命令所構成，其定義如下:

Select 階段:

此過程的目的是讓詢問者選擇特定數量的標籤以便接下來進行盤點和存取。詢問者可以使用一個或多個 Select 命令去選擇特定目標的標籤。

Inventory 階段:

此過程的目的是對已被選擇的標籤進行識別。此階段的運作流程從讀取器送出 Query 命令開始，標籤會傳送資料給讀取器進行排程，而在排程當中，可能會發生三種情況，分別為碰撞(collision)、單一(single)和無(empty)。

Access 階段:

在經過 Select 和 Inventory 階段之後，讀取器便能夠對單一標籤進行存取。此階段由多樣的命令所組成。

除了讀取器所歷經的三個階段之外，標籤也會依據讀取器發送出的不同命令進行相對應狀態的轉換，而標籤的狀態共分為七種，下圖 7 描述了標籤狀態的切換過程及運作功能，由於本研究的重點不在於此，所以在此不詳細的說明標籤狀態運作的細節。

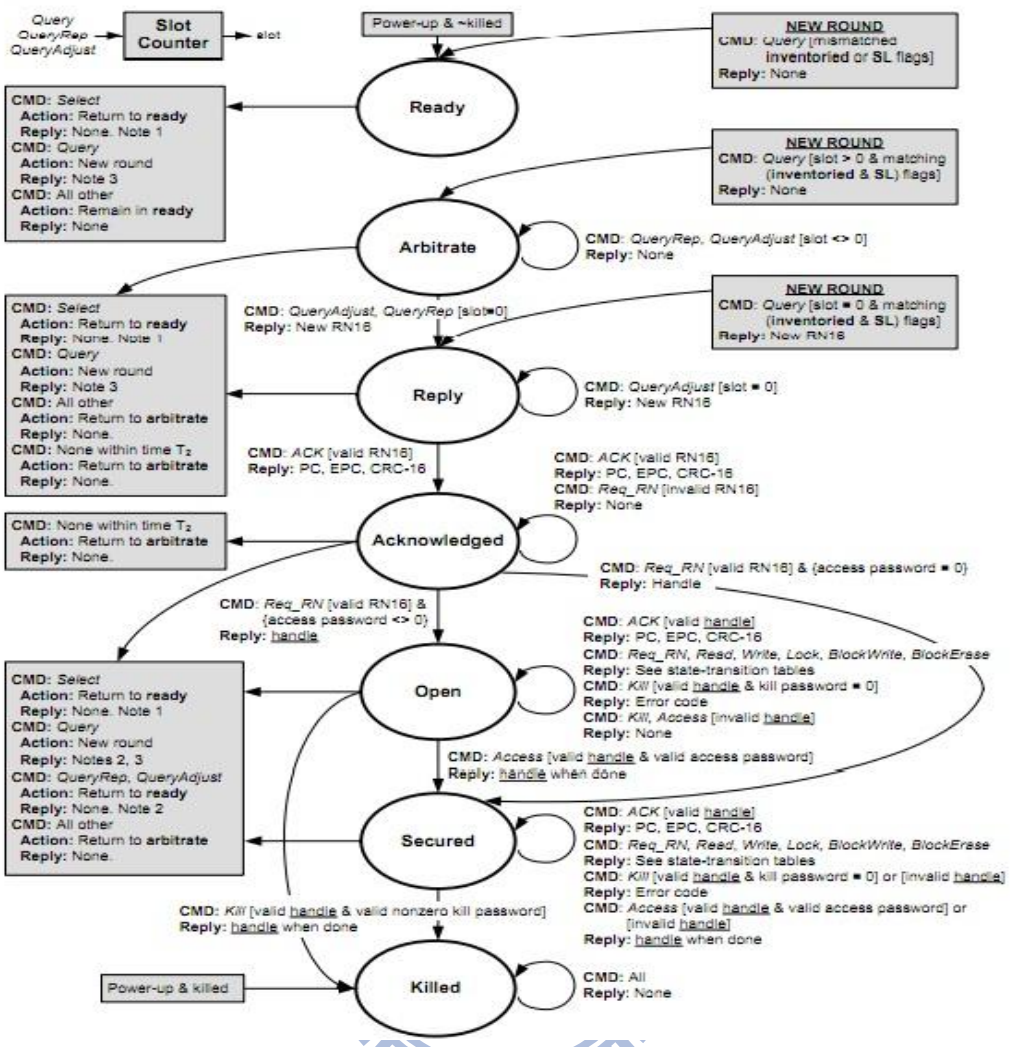

圖 7: 標籤狀態圖

資料來源: http://www.epcglobalinc.org

當讀取器和標籤在傳輸資料的時候，若標籤的數量龐大，則很有可能會因為多個標籤同時要傳送資料給讀取器而發生了碰撞的情況，此時，就需要使用反碰撞演算法(anti-collision algorithm)來解決這些碰撞，而 Gen2 標準為了要統一反碰撞演算法，於是採用了 slotted ALOHA 演算法，下圖 8 即為 slotted ALOHA 協定，利用排程的方式讓所有的標籤進入時間槽中排隊，每次以一個時間槽為單位進行標籤的讀取，其運作流程如下:

Step 1: 讀取器周期性的發送訊號給標籤。

Step 2: 標籤收到訊號之後，選擇某一個時段的時間槽進入，等待讀取器對標籤讀取資料。

Step 3: 當多個標籤處於同一個時間槽時，即發生了碰撞，此時讀取器跳過該時間

槽，優先讀取單一情況的時間槽。

Step 4: 讀取器通知單一情況時間槽的標籤傳送資料。

Step 5: 單一情況時間槽的標籤傳送資料給讀取器。對剩下的標籤重複之前的動作，直到所有的標籤皆讀取完畢。

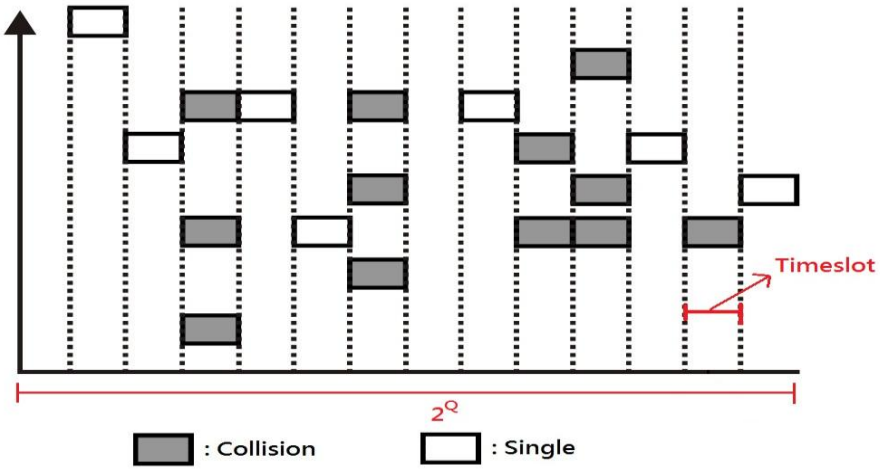

圖 8: slotted ALOHA 協定

資料來源: 維基百科

因為碰撞是無法避免的，所以 Gen2 標準中加入了 Q algorithm 來幫助降低碰撞的發生，下圖 9 即為 Q algorithm。讀取器透過廣播發送 Query 命令給所有的標籤，並且給予標籤一個變數 Q，如果標籤是在 Ready 的狀態下收到命令，則該標籤將會等待一個隨機時間 T 後進行讀取，T 是介於 $1\sim2^{Q}$(2^{Q} 為時間槽的最大值，如圖 8 所示)的隨機數字，當讀取器偵測到有碰撞發生時，讀取器會增加 Q 值以擴大時間槽來降低碰撞的發生率。因此，根據碰撞的發生情況，適度的調整 Q 值來降低碰撞的發生，此即為 Q algorithm 的目的。

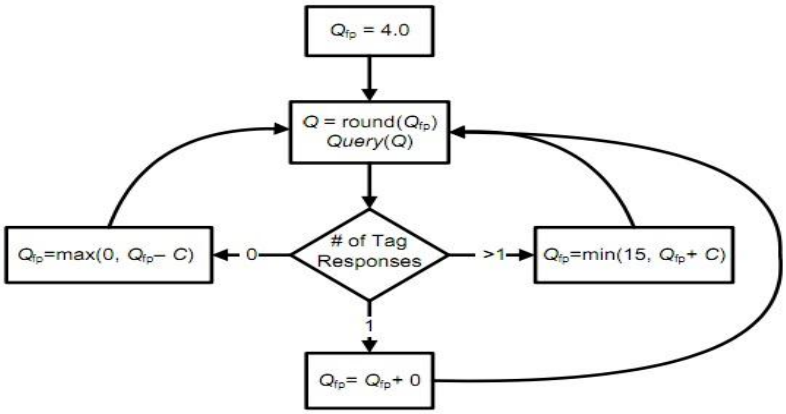

圖 9: Q algorithm

資料來源: http://www.epcglobalinc.org

2.2.3 EPCglobal 安全防護機制

在 Gen2 標準中，除了制定硬體和通訊協定的標準之外，對於資料的安全防護也有提供兩種安全防護機制，分別是 Kill password 和 Access password，詳述如下:

Kill password

Gen2 定義的標籤記憶體中存放了一組 Kill password，如圖 5 所示。Kill 命令是設計來取消標籤接受命令的能力。當標籤收到 Kill 命令時，標籤會比對讀取器所送來的 Kill password 與自己的 Kill password 是否相同，如果相同，則標籤會進入 Killed 狀態，從此之後對於讀取器所下達的任何命令，標籤永遠都不會產生相對應的回應且不再接受或傳送資料。

此機制的優點在於可以有效的保護標籤資料，而不會造成資料被竊聽或追蹤等情況發生。舉例來說，當一個消費者進入賣場購買某項商品之後，商品結帳後，標籤經過 Kill 處理，此時，商品的資料內容不會再傳送出去，消費者也就不用擔心自己購買的商品會被別人知道，且該商品也不會被追蹤，可以有效的保護消費者的隱私安全。但是缺點也顯而易見，即經過 Kill 處理過的標籤無法再被重新利用，造成該標籤變成一次性拋棄物品。

Access password

Gen2 定義的標籤記憶體中存放了一組 Access password，如圖 5 所示，並定義了兩個命令，分為為 Access 和 Lock。Access 命令可以讓標籤進入 Secured 狀態，當標籤處於 Secured 狀態時，讀取器若想要存取標籤資料，則讀取器必須先發送 Access password 給標籤，標籤比對讀取器所送來的 Access password 與自己的 Access password 是否相同，如果相同，才允許讀取器存取資料。

此機制的優點在於標籤資料再被存取之前會先做認證機制，只有知道 Access password 的讀取器才能存取標籤資料，否則無法進行存取，可以有效的保護標籤資料。但缺點則是 Access password 只有 32 位元，使用暴力法破解密碼的可能性不低，因此，此機制無法確保標籤資料的安全性。

2.3 RFID 安全議題

RFID 帶給我們許多的方便，但在某些程度上也帶來了風險，如遭到惡意探測，可能會造成個人資料或者金融卡資訊外洩的情況，如此一來，將帶給 RFID 使用者安全性上的威脅。RFID 系統的應用，其背後皆潛藏著許多安全性的風險，隨著 RFID 系統應用越來越廣泛，此一安全性的問題就越來越嚴重，如今，RFID 系統的安全性已經是不容忽視的一項議題。

雖然 Gen2 標準中有提供兩種安全防護機制，但此兩種安全防護機制並沒有提供讀取器與標籤之間溝通訊號的防護功能，即若攻擊者對 Inventory 階段的讀

取器與標籤之間的無線訊號進行干擾，則 Gen2 標準的安全防護機制無法有效的偵測且達到保護的效果，因此，RFID 技術中仍存在著有待解決的安全問題，以下依序介紹 RFID 的隱私問題和常見的 RFID 攻擊種類。

2.3.1 RFID 隱私問題

RFID 隱私問題主要分為兩種，分別為資料隱私(Data privacy)和位置隱私(Location privacy)，介紹如下:

資料隱私

資料隱私外洩的威脅，通常發生在攻擊者利用讀取器在不被發現的情況下讀取特定物品的標籤資料，如此一來，攻擊者即可獲得該項物品的商品資訊，如商品價格、產品製造商、身分證識別號碼等，不同的商品資訊會對使用者造成不同程度上的損害。

位置隱私

位置隱私外洩的威脅，因為人們身上所攜帶的物品之中，可能包含了具有標籤的物品，如此一來，攻擊者透過持續對該項具有標籤的物品進行讀取的動作，並且根據標籤所回應的資料進行跟蹤，即可以達到對特定人物的跟蹤效果，且被跟蹤的人幾乎不會發現自己已經被跟蹤了。

2.3.2 RFID 攻擊種類

RFID 的攻擊手段非常多樣化，從被動式的竊聽到主動式的干擾，都會對 RFID 系統造成程度不等的傷害。接下來要介紹現今在 RFID 系統中常見的攻擊手段，並且依據攻擊者是否需要取得標籤實體才可展開攻擊分類，將這些攻擊區分成兩種類型，分別是 Non-physical attack 和 Physical attack。

Non-physical attack

此類型攻擊所代表的意義是攻擊者不需要取得標籤的實體即可對 RFID 系統展開攻擊。在此類型的攻擊中，將依序介紹 Eavesdropping、Tag cloning、Tag tracing、Impersonation、Replay attack 和 Desynchronization attack 此六種攻擊手段。

1. Eavesdropping

由於讀取器與標籤是藉由無線電波的方式進行通訊，故任何人都可以藉由硬體設備竊聽讀取器與標籤的通訊過程，從中擷取讀取器和標籤所送出的訊息。此攻擊所造成的傷害性取決於訊息的重要性，如果訊息的內容皆經過加解密或混淆訊息的處理，則攻擊者無法取得真正的資料，此種情況下的傷害程度較低；但如

果訊息的內容無經過任何加解密或混淆訊息的處理，則攻擊者取得相關的資料後，若該訊息包含了個人隱私的資料，則侵犯到個人的隱私權，另外，擷取到的資料可以用來作為其他後續攻擊的用途，此種情況下的傷害程度就不容小覷。

2. Tag cloning

在此攻擊中，要探討的情況是當攻擊者手邊沒有目標標籤的實體時，即攻擊者並不知道目標標籤的資料，此時，攻擊者是否能夠藉由竊聽讀取器與目標標籤的通訊過程，分析此過程中雙方所傳輸的訊息，進而收集到目標標籤的相關資料，並且利用收集來的資料偽造出另一個標籤，使得偽造的標籤與原本的目標標籤具有相同的資料，如此一來，攻擊者持有偽造的標籤如同持有原本的目標標籤，並可以利用此偽造的標籤進行合法的交易。

3. Tag tracing

在 RFID 的認證過程中，一開始讀取器會送出 Query 給標籤，標籤收到後會有所回應，此時，若標籤每次回應的訊息內容皆相同，攻擊者便可以利用這點達到追蹤標籤的效果，即攻擊者使用讀取器不斷的發出 Query 給目標標籤，因為目標標籤只會回應固定內容的訊息，攻擊者可以藉此判斷目前所進行追蹤的標籤是否為目標標籤。直覺來看，追蹤標籤好像不會造成重大的傷害，但是，如果考慮到人們會帶著具有標籤的物品活動時，攻擊者就可以透過追蹤目標標籤達到追蹤目標人物的效果，如此一來，不但關係到人的隱私權問題，甚至可能造成人員生命財產的安全問題。

4. Impersonation

此攻擊所要探討的是當 RFID 認證協定不完善時，即沒有做到 Tag-to-Reader 的認證機制，攻擊者可以使用任意一個讀取器，即使該讀取器不是合法的，但是仍然可以讓標籤誤以為它是合法的讀取器，並且允許該讀取器存取標籤的資料。此項攻擊可能會造成標籤內容遭到竄改，造成重大的損失，舉例來說，當標籤作為商店的儲值卡用途，且標籤內容包含該使用者的儲值金額，則若攻擊者透過此項攻擊修改標籤內容中的儲值金額大小，則會對店家造成重大的損害。

5. Replay attack

若攻擊者企圖仿造標籤並且成功的蒙騙過讀取器或者是仿冒合法的讀取器並且成功的蒙騙過標籤，其中的關鍵在於必須傳送正確的訊息給對方，讓對方相信你是合法的。但如何知道什麼是正確的訊息就涉及到 RFID 的認證協定，也就是說攻擊者必須對該 RFID 系統的認證協定有一定的了解，才有辦法知道要傳送怎樣的訊息讓對方相信。而 Replay attack 可以讓攻擊者在對 RFID 認證協定不知情的情況，傳送正確的訊息讓對方相信，以偽造的標籤要蒙騙讀取器的角度說明，首先攻擊者必須竊聽讀取器與標籤之間的通訊過程，並且記錄下標籤送給讀取器

的所有訊息，接下來，攻擊者使用偽造的標籤去跟讀取器做認證，每當讀取器發送訊號給偽造的標籤時，攻擊者就送回之前所竊聽到的且相對於該讀取器訊號的標籤訊息給讀取器，由於該標籤訊息是從合法標籤所傳送的訊息中竊聽來的，當 RFID 認證協定沒有做到此項攻擊的相關防禦措施時，則讀取器就可能會因為認可該訊息而認為攻擊者的偽造標籤是合法的標籤，如此一來，即達到偽造標籤並成功蒙騙讀取器的效果，將對 RFID 系統造成一定程度上的損害。

6. Desynchronization attack

此攻擊所要探討的對象是每一回合認證結束後，後端伺服器和標籤都會做資料更新的 RFID 認證協定。採用此種認證協定的 RFID 系統中，後端伺服器和標籤皆會儲存著與認證相關的重要資料，而基於安全考量，這些資料在每一回合認證結束後都必須更新，無論更新的順序為何，RFID 系統無法確保標籤和後端伺服器的更新可以同步化。一般來說，後端伺服器會先做更新資料的動作，而標籤必須等待收到讀取器所送來的訊息才做資料更新的動作，當然，此訊息也是後端伺服器做完更新動作後傳送給讀取器的，此時，攻擊者可以利用 DoS(Denial of Service)或者其他干擾方法，迫使標籤沒有收到讀取器所送來的訊息而沒有做更新的動作，如此一來，就會造成後端伺服器的資料是更新過的，但是標籤的資料是尚未更新過的，此種情況即為後端伺服器與標籤的資料不一致，若該 RFID 認證協定沒有針對此項攻擊提供相關的補救措施，則有可能會造成標籤無法再被辨識的報廢情況發生。

Physical attack

此類型攻擊所代表的意義是攻擊者已取得某個標籤的實體，並且經由各種方法的分析，獲取標籤記憶體中的資料，利用這些資料對 RFID 系統展開攻擊。在此類型的攻擊中，將依序介紹 Compromising attack、Forward security 和 Backward security 此三種攻擊手段。

1. **Compromising attack**

在此攻擊探討的情況是當攻擊者取得了標籤的實體，並且得知該標籤的資料，而整個 RFID 系統，包含所有標籤，會不會因為攻擊者知道了這些資料而造成損害。舉例來說，當 RFID 系統的標籤皆儲存了一把 shared key，假設此 key 是認證過程中，後端伺服器與標籤用來對傳輸資料做加解密之用，因此，後端伺服器與標籤都必須儲存著這把 key，因為 key 是儲存在標籤之中，只要認證的通訊過程中，此 key 不會以明文的方式傳送出去，就不會有外漏的情況，但是現在探討的情況是攻擊者取得標籤，得知標籤的資料，即攻擊者知道了這把 key，如此一來，攻擊者不但可以利用此 key 對該標籤所傳送的訊息做解密動作得知標籤所傳送的資料為何，還可以利用同樣的手段得知其他標籤所傳送的訊息內容。故此攻擊對 RFID 系統所造成的傷害取決於標籤中所儲存的共享資料之重要性。

2. Forward security

此項攻擊的重點在於，攻擊者拿到標籤的實體，也知道標籤的資料，另外還有一份 RFID 系統的通訊訊息紀錄，若 RFID 認證協定沒有做到抵抗此項攻擊的相關防禦措施，則攻擊者可能可以利用自標籤中取得的資料，與通訊訊息記錄進行比對，分析出哪些通訊訊息是關於此標籤的，進而觀察出該標籤過去的交易行為。舉例來說，攻擊者取得某人的儲值卡，並且取得該儲值卡中的標籤資料，另外，攻擊者也有某個賣場的 RFID 通訊訊息記錄，若該賣場的 RFID 系統沒有做到抵抗此項攻擊的相關防禦措施，則攻擊者即可藉由已知標籤的資料與通訊訊息記錄做比對，找出該儲值卡的原主人在該賣場過去的交易行為，如此一來，即侵犯了個人的隱私權。

3. Backward security

此項攻擊與 Forward security 的原理相同，重點都在於攻擊者可否透過已知的標籤資料與交易紀錄進行比對，找出該標籤的交易行為，而差別在於，以取得標籤實體的時間為基準點，Forward security 是針對找出該時間點之前，該標籤的交易行為；Backward security 則是針對找出該時間點之後，該標籤的交易行為。此攻擊同樣也是侵犯到個人的隱私權。

第三章 相關研究

本章將介紹六個 RFID 認證協定，並且依照後端伺服器查詢標籤資料所花費的時間為依據，將其分成三類，分別是 O(N)、O(N/2^m)和 O(1) Database searching time RFID authentication protocol，其中 O(N)類型為較常見的 RFID 認證協定，而 O(N/2^m)和 O(1)類型的則是有使用一些小技巧來減少後端伺服器查詢標籤資料所花費的時間。首先，將在 3.1 節介紹 O(N) Database searching time RFID authentication protocol，在此節中，將介紹兩個屬於此類型的 RFID 認證協定，接下來在 3.2 節介紹 O(N/2^m) Database searching time RFID authentication protocol，在此節中，將介紹一個屬於此類型的 RFID 認證協定，最後在 3.3 節介紹 O(1) Database searching time RFID authentication protocol，在此節中，將介紹三個屬於此類型的 RFID 認證協定。

3.1 O(N) Database searching time RFID authentication protocol

此節將介紹後端伺服器查詢標籤資料時間為 O(N)的 RFID 認證協定。3.1.1 介紹 Jie Li, 2010，3.1.2 介紹 Xiaoyun Chen, 2010。

3.1.1 Jie Li, 2010

這是由 Jie Li 等人在 2010 年所提出的 RFID 認證協定[3]。接下來將依序介紹此 RFID 認證協定的認證流程、安全性分析和效能分析。

認證流程

H:	a hash function
G:	a hash function
ID:	tag identification code
K:	secret value shared by the server and the tag
M:	how many times the tag had not update ID and K
ID_{old}:	ID which is used in current communication by the server after success authentication
ID_{new}:	ID which will be used in the new communication by the server after success authentication
K_{old}:	K which is used in current communication by the server after success authentication
K_{new}:	K which will be used in the new communication by the server after success authentication
Rr:	random numbers generated by the reader
R:	random numbers generated by the tag
$\|\|$:	concatenation operator
$\oplus$:	XOR operator
=:	assignment operator

圖 10: Jie Li, 2010 符號說明

資料來源: [3]

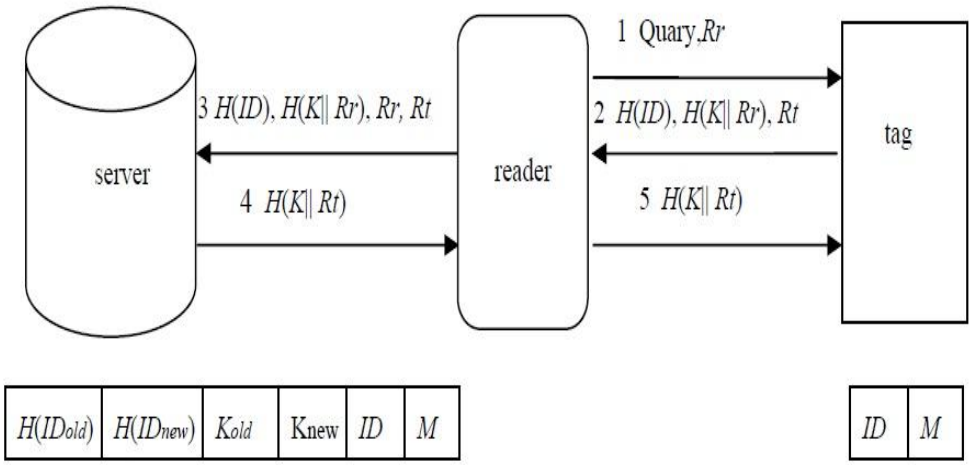

圖 11: Jie Li,2010 認證流程

資料來源: [3]

圖 10 為此 RFID 認證協定所用到符號的說明，圖 11 為此 RFID 認證協定的流程。接下來將一步一步說明:

Step 0: 初始化動作。標籤在一開始必須先儲存著與後端伺服器共享的兩筆資料，ID 和 K；後端伺服器在資料庫中必須儲存著每一個標籤的認證資料，如下:

(1) ID: 標籤 ID。

(2) $H(ID_{new})$: 標籤的新 ID 經過雜湊函數 H( )計算後的值。初始值為標籤 ID 經過雜湊函數 H( )計算後的值。

(3) $H(ID_{old})$: 標籤的舊 ID 經過雜湊函數 H( )計算後的值。初始值為空。

(4) K_{new}: 標籤的新 key，初始值為 K。

(5) K_{old}: 標籤的舊 key，初始值為空。

(6) M: 記錄著該標籤多少回合沒有做 ID 和 K 的更新動作。

Step 1: 讀取器產生一個隨機亂數 Rr，發出 Query 給標籤，並且連同 Rr 一起傳送給標籤。

Step 2: 標籤收到後，產生一個隨機亂數 Rt，並且計算 H(ID)和 H(K || Rr)，最後將 H(ID)、H(K || Rr)和 Rt 傳送給讀取器。

Step 3: 讀取器收到後，將收到的 H(ID)、H(K || Rr)和 Rt，連同 Rr 一起傳送給後端伺服器。

Step 4-1: 後端伺服器收到後，將收到的 H(ID)與資料庫中每一個項目的 $H(ID_{new})$和 $H(ID_{old})$做比對，比較是否相同，此時會有下列三種情況:

i. 沒有任何一個 $H(ID_{new})$或 $H(ID_{old})$與 H(ID)相同，此時代表後端伺服器在資料庫中找不到該標籤的資料，則終止此回合的通訊。

ii. 找到一個 $H(ID_{new})$與 H(ID)相同，此時代表後端伺服器在資料庫中找到該標籤的資料，且該標籤所儲存的資料與資料庫同步，此時，後端伺服器自該標籤所在的項目中取出 K_{new}，計算 $H(K_{new} \| Rr)$，並將計算結果與剛收到的 $H(K \| Rr)$做比對，若不相等，則終止此回合的通訊；若相等，則代表此標籤通過 Reader-to-Tag 的認證，為合法標籤，接下來，後端伺服器會更新該標籤在資料庫中的相關資料，更新方式為 $H(ID_{old}) = H(ID_{new})$、$ID = G(ID)$、$H(ID_{new}) = H(ID)$、$K_{old} = K_{new}$、$K_{new} = ID \oplus Rr \oplus Rt$。

iii. 找到一個 $H(ID_{old})$與 H(ID)相同，此時代表後端伺服器在資料庫中找到該標籤的資料，且該標籤所儲存的資料與資料庫不同步，此時，後端伺服器自該標籤所在的項目中取出 M，檢查 M 值是否超過規定的最大值，若超過，則意味此標籤太久未更新，為了安全考量必須終止此次通訊；若沒超過，則後端伺服器再取出 K_{old}，計算 $H(K_{old} \| Rr)$，並將計算結果與剛收到的 $H(K \| Rr)$做比對，若不相等，則終止此回合的通訊；若相等，則代表此標籤通過 Reader-to-Tag 的認證，為合法標籤，但後端伺服器在此種情況下不做標籤資料更新的動作。

Step 4-2: 若標籤以 Step 4-1 ii 的情況通過 Reader-to-Tag 認證，則後端伺服器會計算 $H(K_{new} \| Rt)$，並將 $H(K_{new} \| Rt)$傳送給讀取器；若標籤 Step 4-1 iii 的情況通過 Reader-to-Tag 認證，則後端伺服器會計算 $H(K_{old} \| Rt)$，並將 $H(K_{old} \| Rt)$傳送給讀取器。

Step 5-1: 讀取器收到後，直接將 $H(K_{new} \| Rt)$或 $H(K_{old} \| Rt)$傳送給標籤。

Step 5-2: 標籤收到後，計算 $H(K \| Rt)$，並將計算結果與收到的 $H(K_{new} \| Rt)$或 $H(K_{old} \| Rt)$做比對，若不相等，則終止此回合的通訊；若相等，則代表該讀取器通過 Tag-to-Reader 的認證，為合法讀取器，接下來，標籤會更新自己的資料，更新方式為 $ID = G(ID)$、$K = ID \oplus Rr \oplus Rt$。

安全性分析

此 RFID 認證協定有做到一般常見 RFID 攻擊的相關防禦措施，因此，大部分的 RFID 攻擊都無法有效的造成傷害。另外，因為此 RFID 認證協定在每一回合認證結束後，後端伺服器和標籤都必須做資料更新的動作，這類型的 RFID 認證協定就必須考量到 Desynchronization attack，一旦遭受此攻擊，必須提供相關的補救措施讓 RFID 系統能夠正常的運作，若沒有提供任何相關的補救措施，則可能造成標籤無法再被識別的後果。而在此 RFID 認證協定當中，後端伺服器在資料庫中有額外的保留舊的標籤資訊，$H(ID_{old})$、K_{old} 和 M，這些舊的標籤資料就是此 RFID 認證協定為了防禦 Desynchronization attack 的補救措施，當後端伺

服器與標籤發生資料不一致的情況時(後端伺服器資料已更新，標籤資料尚未更新)，後端伺服器收到標籤所送來的資料時，若標籤是合法的，即使標籤所送來的資料是舊的，後端伺服器仍然可以自資料庫中找到 $H(ID_{old})$和 K_{old}，其中 $H(ID_{old})$與標籤所送來的 H(ID)相等，$H(K_{old} \| Rr)$與標籤所送來的 H(K || Rr)相等，而完成標籤的認證，因此，在此 RFID 認證協定當中，不會發生因為遭受 Desynchronization attack 而導致標籤無法再被識別的下場。

但在 Tag tracing 方面，此 RFID 認證協定利用每一回合認證結束後會對標籤的 ID 做更新動作及隨機亂數 Rr 和 Rt，使得標籤傳送回讀取器的訊息: H(ID), H(K || Rr), Rt，每一回合都不相同，藉此方法來抵抗 Tag tracing。但由於標籤必須等到收到認證協定中最後一個訊息，並且判斷對方是合法的讀取器後，才會做資料更新的動作，也就是說，H(ID)只有在經過合法的交易回合後，才會因為資料更新而不同。考慮到下列情況，若攻擊者正在對目標標籤做 Tag tracing，且該目標標籤在短時間內都沒有發生合法的交易，因此標籤每一次所傳送回來的 H(ID)皆相同，攻擊者就可以利用 H(ID)對目標標籤進行追蹤，在這種情況下，此 RFID 認證協定有著 Tag tracing 的風險。

效能分析

此 RFID 認證協定使用了雜湊函數和 XOR 運算，無論在後端伺服器或是標籤都必須使用到雜湊函數，因此，雖然後端伺服器在認證過程中的運算量相較於其他 RFID 認證協定不算高，但就標籤而言，使用到雜湊函數，其運算量相較於其他 RFID 認證協定不算低。

另外，考慮到後端伺服器在資料庫中查詢標籤資料所花費的時間，因為後端伺服器會憑藉著標籤所傳送來的資料，H(ID)，與資料庫中每一個項目的 $H(ID_{new})$和 $H(ID_{old})$做比對，相同才代表找到該標籤的資料，此查詢時間的花費為 O(N)，其中，N 為 RFID 系統的標籤數量。考慮到標籤數量眾多或交易次數頻繁的 RFID 系統環境，此一時間的花費會造成後端伺服器效能的負擔。

3.1.2 Xiaoyun Chen, 2010

這是由 Xiaoyun Chen 等人在 2010 年所提出的 RFID 認證協定[4]。接下來將依序介紹此 RFID 認證協定的認證流程、安全性分析和效能分析。

認證流程

符號	說明
RR	後端伺服器產生的隨機亂數。
TID_K	標籤 ID，即標籤辨識碼。
R_i	標籤中儲存著 Random list，RL，其中包含 n 個隨機亂數，$R_0, R_1, ..., R_{n-1}$。R_i 即為此 RL 中第 i 個隨機亂數。
TR_i	標籤與讀取器進行第 i 次交易時，標籤所使用的隨機亂數。

圖 12: Xiaoyun Chen, 2010 符號說明
資料來源: [4]

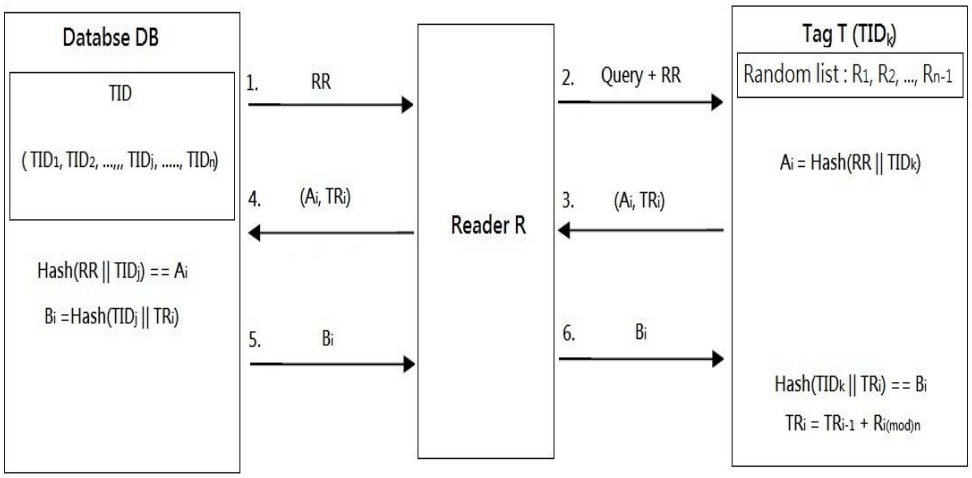

圖 13: Xiaoyun Chen, 2010 認證流程
資料來源: [4]

圖 12 為此 RFID 認證協定所用到符號的說明圖，圖 13 為此 RFID 認證協定的流程。接下來將一步一步說明:

Step 0: 初始化動作。標籤在一開始會儲存著標籤識別碼 TID_k，另外，還會儲存著 Random List，RL，RL 中有 n 個隨機亂數，R_0、R_1、…、R_{n-1}；後端伺服器在資料庫中必須儲存著每一個標籤的認證資料，TID。

Step 1: 後端伺服器產生一個隨機亂數，RR，傳送給讀取器。

Step 2: 讀取器發送 Query 給標籤，並且連同 RR 一起傳送給標籤。

Step 3: 標籤收到後，計算 $A_i = Hash(RR \| TID_k)$，計算完後將 A_i 和 TR_i 一起傳送給讀取器。

Step 4: 讀取器收到後，直接將 A_i 和 TR_i 一起傳送給後端伺服器。

Step 5: 後端伺服器收到後，逐一檢查資料庫中每一個項目，檢查方式為取出每個項目中的 TID 欄位資料，假設為 TID_j，計算 $A_i' = Hash(TID_j \| RR)$，並且比對 A_i'與剛收到的 A_i 是否相同，若不相同，則代表此為不合法的標籤，終止此次通訊；若相同，則代表此為合法的標籤，通過了 Reader-to-Tag 的認證，接下來，利用 TID_j 和剛收到 TR_i，計算 $B_i = Hash(TID_j \| TR_i)$，再將 B_i 送給讀取器。

Step 6: 讀取器收到後，直接將 B_i 傳送給標籤。

Step 7: 標籤收到後，計算 $B_i' = Hash(TID_k \| TR_i)$，並且比對 B_i'與剛收到的 B_i 是否相同，若不相同，則代表此為不合法的讀取器，終止此次通訊；若相同，則代表該讀取器通過 Tag-to-Reader 的認證，為合法的讀取器，接下來，標籤會做更新 TR_i 的動作，更新方法為 $TR_{i+1} = TR_i + R_{i+1(mod)n}$。

安全性分析

此 RFID 認證協定有做到一般常見 RFID 攻擊的相關防禦措施，因此，大部分的 RFID 攻擊都無法有效的造成傷害，接下來以數個 RFID 攻擊為例，簡單的說明此 RFID 認證協定如何抵抗這些攻擊。在 Tag tracing 方面，標籤每次傳送回讀取器的內容 A_i 和 TR_i 都不相同，攻擊者無法進行追蹤，故此 RFID 認證協定可以抵抗這類型的攻擊；在 Replay attack 方面，標籤與讀取器之間的訊息皆有做到 Challenge-response 認證，攻擊者無法利用過去標籤與讀取器之間的合法認證訊息試圖再次認證成功，故此 RFID 認證協定可以抵抗這類型的攻擊；在 Desynchronization attack 方面，因為此 RFID 認證協定在每回合認證結束時，只有標籤需要做更新 TR_i 的動作，而後端伺服器不需要做任何更新的動作，所以不會有標籤與後端伺服器資料不一致的問題發生，故此 RFID 認證協定不需要顧慮這類型的攻擊。

效能分析

此 RFID 認證協定在標籤方面使用 Random list 取代隨機亂數產生器來產生隨機亂數，如此一來，標籤不需要設計隨機亂數產生器的電路，可以減少標籤的生產成本，但 Random list 必須儲存在標籤之中，因此，標籤需要更多的儲存空間來存放 Random list，使得原本減少的標籤生產成本又因此提升。

此 RFID 認證協定使用了雜湊函數，無論在後端伺服器或是標籤都必須使用到雜湊函數。在標籤方面，因為使用了雜湊函數，故其運算量相較於其他 RFID 認證協定並不算低；在後端伺服器方面因為大量的使用雜湊函數，且使用次數與 RFID 系統的標籤數量成正比關係，使得後端伺服器的運算量相較於其他 RFID 認證協定高出不少。

另外，考慮到後端伺服器在資料庫中查詢標籤資料所花費的時間，後端伺服器憑藉著標籤所送來的 $A_i = Hash(RR \| TID_k)$，自資料庫中逐一取出每一個項目

中的 TID 欄位資料，令為 TID_j，計算 $A_i' = Hash(TID_j || RR)$，計算完後，再比對 A_i 與 A_i'是否相同，相同才代表找到該標籤的資料，此查詢時間的花費為 O(N)，其中，N 為 RFID 系統的標籤數量，其中還包含了 N 次的雜湊函數運算量。雖然此 RFID 認證協定之所以安全，有大部份的原因是歸咎於後端伺服器額外的運算量，但考慮到標籤數量眾多或交易次數頻繁的 RFID 系統，後端伺服器查詢標籤資料所花費的時間恐怕會造成該 RFID 系統效能上的負擔。

3.2 O(N/2^m) Database searching time RFID authentication protocol

此節將介紹後端伺服器查詢標籤資料時間為 O(N/2^m)的 RFID 認證協定，3.2.1 介紹 Chiu C. Tan, 2008。

3.2.1 Chiu C. Tan, 2008

這是由 Chiu C. Tan 等人在 2008 年所提出的 RFID 認證協定[5]。接下來將依序介紹此 RFID 認證協定的認證流程、安全性分析和效能分析。

認證流程

CA	Trusted party, responsible for authenticating readers and deploying tags
R_i	RFID reader i
r_i	id for RFID reader R_i
L_i	access list for RFID reader R_i
n	number of entries in L_i
T_i	RFID tag i
id_i	id for RFID tag T_i
t_i	secret for RFID tag T_i
$h(x)$	one-way hash function
$f(x, y)$	Concatenate x and y, then applying $h(.)$, $h(x\|\|y)$
l	number of bits of hash $h(.)$
m	CA defined number of bits, $m < l$

圖 14: Chiu C. Tan, 2008 符號說明

資料來源: [5]

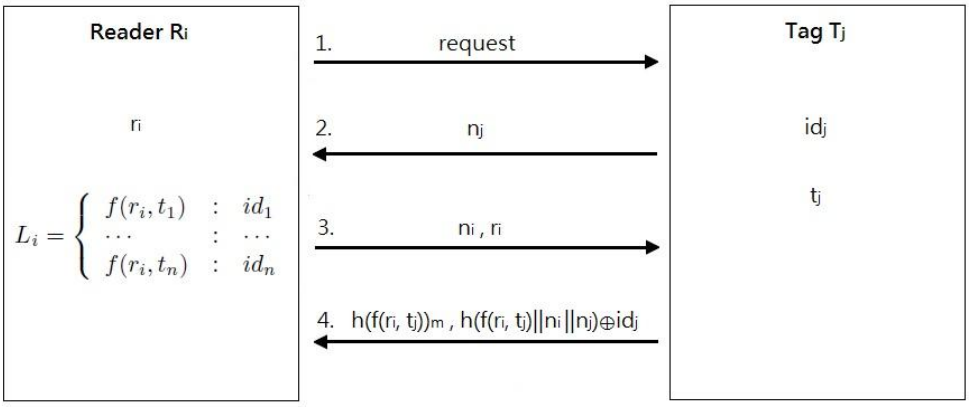

圖 15: Chiu C. Tan, 2008 認證流程

資料來源: [5]

圖 14 為此 RFID 認證協定所用到符號的說明圖，圖 15 為此 RFID 認證協定的流程。接下來將一步一步說明:

Step 0: 初始化動作。此 RFID 認證協定考量到當讀取器無法與後端伺服器進行連線時，讀取器仍然要有能力做到標籤的辨識，如以下情況: 送貨員被指派到某個偏僻的倉庫做物品盤點的工作，且該倉庫無法與後端伺服器連線，此時，送貨員必須單靠讀取器完成物品的盤點工作。因此，此 RFID 認證協定將原本的後端伺服器功能整合到讀取器中，故圖 15 的認證流程中，沒有後端伺服器元件。此 RFID 認證協定中，有一個 Certificate authority，CA，負責讀取器和標籤的註冊申請，每一個讀取器都必須向此 CA 註冊成為該 RFID 系統的合法讀取器，每一個標籤也必須向此 CA 註冊成為該 RFID 系統的合法標籤。每一個合法的讀取器會儲存著 r_i 和 L_i，其中 r_i 代表讀取器 R_i 的 ID，L_i 為 R_i 所能存取的標籤名單，其形式如圖 15 所示；每一個合法的標籤會儲存著 id_j 和 t_j，其中 id_j 代表標籤 T_j 的 ID，t_j 為 T_j 與 CA 共享的資訊，只有 T_j 和 CA 知道。另外，此 RFID 認證協定會使用到雜湊函數 h( )，令 l 代表 h( )輸出的長度，則 CA 會定義 m，$m < l$，且該 RFID 系統中的所有讀取器和標籤都知道 m。

Step 1: 讀取器發出 request 給標籤。

Step 2: 標籤收到後，產生一個隨機亂數 n_j，並將 n_j 傳送給讀取器。

Step 3: 讀取器收到後，產生一個隨機亂數 n_i，連同 r_i 一起傳送給標籤。

Step 4-1: 標籤收到後，計算 $h(f(r_i, t_j))$，但僅留下結果的前 m 位元，令為 $h(f(r_i, t_j))_m$，再計算 $h(f(r_i, t_j) \| n_i \| n_j) \oplus id_j$，計算完後，將 $h(f(r_i, t_j))_m$ 和 $h(f(r_i, t_j) \| n_i \| n_j) \oplus id_j$ 傳送給讀取器。

Step 4-2-1: 讀取器可以事先組織 L_i，計算 L_i 中的 $h(f(r_i, t_k))$，$1 \le k \le n$，並且將結果前 m 位元相同者劃分到同一個群組中。讀取器收到標籤送來的訊息後，先利用 $h(f(r_i, t_j))_m$ 與每一個群組的前 m 位元比對，若找不到任何一個群組的前 m 位元與 $h(f(r_i, t_j))_m$ 相同，則代表該標籤不在此讀取器的存取名單之中，終止此次通訊；若找到某個群組的前 m 位元與 $h(f(r_i, t_j))_m$ 相同，再進行 **Step 4-2-2**。此步驟的意義為只有經過 CA 授權可以存取該標籤的合法讀取器才能在 L_i 中找到該標籤所在的群組，因此，可以視為 Tag-to-Reader 的認證。

Step 4-2-2: 讀取器找到該標籤所在的群組後，逐一檢查該群組中的每一個項目，檢查方法為取出每一個項目中的 $f(r_i, t_k)$，計算 $h(f(r_i, t_k) \| n_i \| n_j) \oplus id_k$，並將計算結果與收到的 $h(f(r_i, t_j) \| n_i \| n_j) \oplus id_j$ 進行比對，若有找到一個項目符合，則代表

該標籤通過 Reader-to-Tag 的認證，為合法標籤；否則，終止此次通訊。

安全性分析

此 RFID 認證協定將後端伺服器的功能整合到讀取器中，解決了當讀取器所在的環境不能與後端伺服器進行連線時，讀取器就無法繼續辨識標籤的情況，但也因為這樣，讀取器必須儲存著所有標籤與認證相關的資料，此時，若標籤的秘密皆儲存在讀取器中，一旦讀取器落入某個攻擊者手中，則攻擊者就有可能得知所有標籤的秘密，並且利用這些資訊偽造出假的標籤。此 RFID 認證協定有考量到此種情況，因此，標籤的秘密，t_j，只有標籤自己和 CA 知道，而儲存在讀取器中的 $f(r_i, t_j)$，為標籤認證所需的資料，但此資料經過了雜湊函數 h( )的處理，即使讀取器落入攻擊者手中，攻擊者也無法利用 $f(r_i, t_j)$逆推得到 t_j，標籤的秘密就不會因此洩漏，攻擊者就無法偽造標籤。另外，在 forward/backward security 方面，因為標籤傳送回讀取器的訊息內容皆為與讀取器 ID，r_i，做完運算的結果，當標籤與不同的讀取器進行交易時，所回傳的訊息內容皆不相同，因此，即使攻擊者取得標籤實體，也拿到 RFID 系統的通訊紀錄，但因為標籤之前可能跟各個不同的讀取器進行交易，故攻擊者無法在歷史紀錄中觀察出該標籤過去的交易行為為何。

在此 RFID 認證協定之中，使用了將標籤分群組織化的技巧減少查詢標籤資料所花費的時間，但是此技巧卻帶來了 Tag tracing 的風險，原因在於標籤傳送給讀取器的 $h(f(r_i, t_j))_m$，此筆資料是讓讀取器用來找出該標籤所在的群組位置，但是此筆資料在當標籤與同一個讀取器交易時都是相同的值，也就是說，當攻擊者使用同一個讀取器 R_X 一直發出 request 給目標標籤 T_y 時，該標籤所回傳的 $h(f(r_x, t_y))_m$ 都是相同的值，因此，攻擊者可以利用此筆資料進行目標標籤的追蹤。而此 RFID 認證協定也有談論到如何避免 Tag tracing 發生的方法，就是將 m 值調小，因為每個群組中的標籤數為 $n/2^m$，當 m 值變小時，群組中的標籤數就會增加，而同一群組中的標籤有著相同的 $h(f(r_i, t_j))_m$，其意味著有更多的標籤傳送回讀取器的 $h(f(r_i, t_j))_m$ 都相同，此時，攻擊者就無法分辨現在是哪一個標籤在傳送 $h(f(r_i, t_j))_m$，因此，可以降低 Tag tracing 發生的機率。但把 m 值調小，同一群組中的標籤數就會增加，此舉又會讓查詢標籤資料所花費的時間上升。m 值的大小，就像是安全與效率的雙面刃，只可擇一，無法兼顧。

效能分析

此 RFID 認證協定使用了雜湊函數和 XOR 運算，無論在後端伺服器或是標籤都必須使用到雜湊函數。在標籤方面，因為使用了雜湊函數，故其運算量相較於其他 RFID 認證協定並不算低；在後端伺服器方面因為使用多次雜湊函數，且使用次數與群組中的標籤數量成正比關係，使得後端伺服器的運算量略高於其他 RFID 認證協定。

另外，考慮到讀取器查詢標籤資料所花費的時間，讀取器利用 $h(f(r_i, t_j))_m$ 找

到該標籤所在的群組位置，再逐一檢查該群組中所有的標籤，檢查方法為取出每一個項目中的 $f(r_i, t_k)$，計算 $h(f(r_i, t_k) \| n_i \| n_j) \oplus id_k$，並將計算結果與收到的 $h(f(r_i, t_j) \| n_i \| n_j) \oplus id_j$ 進行比對，若相同，則代表找到該標籤的資料，此查詢時間的花費為 $O(n/2^m)$。此 RFID 認證協定有做到降低查詢標籤資料所花費的時間，因此，在標籤數量眾多或交易次數頻繁的 RFID 系統中，其系統效能不會因為查詢標籤資料而低落。

3.3 O(1) Database searching time RFID authentication protocol

此節將介紹後端伺服器查詢標籤資料時間為 O(1)的 RFID 認證協定，3.3.1 介紹 Yanfei Liu, 2008，3.3.2 介紹 FLMAP, Alireza Sadighian, 2008，3.3.3 介紹 Jianqing Fu, 2010。

3.3.1 Yanfei Liu, 2008

這是由 Yanfei Liu 在 2008 年所提出的 RFID 認證協定[6]。接下來將依序介紹此 RFID 認證協定的認證流程、安全性分析和效能分析。

認證流程

S	: The authentication server
R_i	: An RFID reader
T_j	: An RFID tag
$h(.)$	: A one-way hash function
$f_k(.)$	: A one-way function with a secret key k
$\oplus$	: Bitwise XOR operation

圖 16: Yanfei Liu, 2008 符號說明

資料來源: [6]

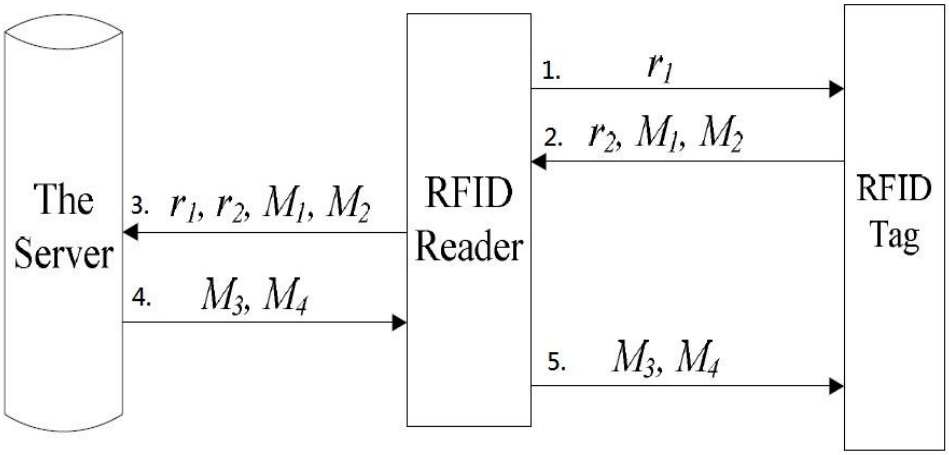

圖 17: Yanfei Liu, 2008 認證流程

資料來源: [6]

圖 16 為此 RFID 認證協定所用到符號的說明圖，圖 17 為此 RFID 認證協定的流程。接下來將一步一步說明:

Step 0: 初始化動作。標籤 T_j 一開始會儲存著 x_j、y_j 和 $h(k)$，其中 x_j 為標籤 T_j 向後端伺服器 S 註冊成功之後，後端伺服器產生的一個隨機亂數；$y_j = f_k(x_j)$，是由後端伺服器計算後配置給標籤 T_j 的，k 為後端伺服器的密鑰(secret key)，作為此標籤的 key；$h(k)$也是由後端伺服器計算後配置給標籤 T_j 的。而後端伺服器 S 會儲存著 k，並且在資料庫中儲存著每一個標籤的認證資料。

Step 1: 讀取器產生一個隨機亂數 r_1，向標籤發出 Query，並連同 r_1 一起傳送給標籤。

Step 2: 標籤收到後，產生一個隨機亂數 r_2，並且計算 $M_1 = x_j \oplus h(h(k) \oplus r_2)$和 $M_2 = h(y_j \oplus r_1 \oplus r_2)$，計算完後，將 r_2、M_1 和 M_2 傳送給讀取器。

Step 3: 讀取器收到後，將收到的 r_2、M_1 和 M_2，連同 r_1 一起傳送給後端伺服器。

Step 4-1: 後端伺服器收到後，利用收到的 M_1 和 r_2 計算 $x_j = M_1 \oplus h(h(k) \oplus r_2)$，計算出 x_j 後，再利用 x_j 計算 $y_j = f_k(x_j)$，接下來計算 $h(y_j \oplus r_2 \oplus r_1)$，並將計算結果與收到的 M_2 進行比對，若不相同，則終止此次通訊；若相同，則代表該標籤通過 Reader-to-Tag 的認證，為合法標籤。

Step 4-2: 後端伺服器更新 x_j 和 y_j，更新方式為 $x_j^* = h(x_j \oplus y_j \oplus r_1 \oplus r_2)$、$y_j^* = f_k(x_j^*)$。另外，後端伺服器會為該標籤保留舊的 key y_j，直到下次與該標籤進行認證時，確認了該標籤的 key 為新的 key y_j^*時，才將舊的 key y_j 丟棄。此動作的目的是要抵抗 Desynchronization attack。

Step 4-3: 後端伺服器計算 $M_3 = y_j^* \oplus h(x_j^* \oplus y_j)$和 $M_4 = h(x_j^* \oplus y_j^*)$，並找出該標籤的相關資料，將 M_3、M_4 和該標籤的相關資料傳送給讀取器。

Step 5-1: 讀取器收到後，保留該標籤的相關資料，並將 M_3、M_4 傳送給標籤。

Step 5-2: 標籤收到後，計算 $x_j^* = h(x_j \oplus y_j \oplus r_1 \oplus r_2)$，利用收到的 M_3 計算 $y_j^* = M_3 \oplus h(x_j^* \oplus y_j)$，接下來計算 $h(x_j^* \oplus y_j^*)$，並將計算結果與收到的 M_4 進行比對，若不相同，則終止此次通訊；若相同，則代表該讀取器通過 Tag-to-Reader 的認證，為合法讀取器，更新 $x_j = x_j^*$，$y_j = y_j^*$。

安全性分析

此 RFID 認證協定有做到一般常見 RFID 攻擊的相關防禦措施，因此，大部分的 RFID 攻擊都無法有效的造成傷害。此 RFID 認證協定使用了一些技巧讓後端伺服器再經過數次雜湊函數的運算後就可以找到標籤的資料，而不用逐一比對資料庫中的每一個項目，大幅的減少查詢標籤資料所花費的時間，更重要的是，此 RFID 認證協定所使用的技巧不會為 RFID 系統帶來安全性的問題，如 Tag tracing。另外，此 RFID 認證協定在每一回合認證結束後，後端伺服器與標籤都必須做資料更新的動作，因此，必須顧慮到 Desynchronization attack，而此 RFID 認證協定的補救措施為後端伺服器會儲存標籤舊的 key y_j，當後端伺服器與標籤發生資料不一致的情況時(後端伺服器的資料已更新、標籤的資料未更新)，該標籤在下一次進行認證時，後端伺服器還是能用該標籤舊的 key y_j 完成標籤的認證。

效能分析

此 RFID 認證協定使用了 HMAC(Hash-based Message Authentication Code) $f_k(\ )$、雜湊函數 h( )和 XOR 運算。在標籤方面，因為使用了多次的雜湊函數，故其運算量相較於其他 RFID 認證協定較為高一些；在後端伺服器方面，雖然使用了 HMAC 和雜湊函數，但使用次數固定，並不會與 RFID 系統中的標籤數量成正比關係，在標籤數量眾多的 RFID 系統環境底下，後端伺服器的運算量低於其他 RFID 認證協定。

另外，考慮到後端伺服器查詢標籤資料所花費的時間，後端伺服器利用收到的 M_1 和 r_2 計算 $x_j = M_1 \oplus h(h(k) \oplus r_2)$，計算出 x_j 後，再利用 x_j 計算 $y_j = f_k(x_j)$，接下來計算 $h(y_j \oplus r_2 \oplus r_1)$，並將計算結果與收到的 M_2 進行比對，若相同，則代表該標籤為合法標籤，並藉此可以直接找到該標籤的相關資料，此查詢時間的花費為 O(1)，大幅的減少後端伺服器查詢標籤資料所花費的時間，因此，在標籤數量眾多或交易次數頻繁的 RFID 系統中，後端伺服器不需花費太多時間在查詢標籤資料上，有助於提升 RFID 系統效能。

3.3.2 FLMAP, Alireza Sadighian, 2008

這是由 Alireza Sadighian 和 Rasool Jalili 在 2008 年所提出的 RFID 認證協定[7]。接下來將依序介紹此 RFID 認證協定的認證流程、安全性分析和效能分析。

認證流程

符號	說明
T	RF tag, or transponder.
R	RF tag reader, or transceiver.
B	Back-end server, which has a database.
DB	Database of Back-end server.
h()	One-way hash function.
PRNG	Pseudo Random Number Generator.
⊕	XOR operation.
Λ	AND operation.
V	OR operation.

圖 18: FLMAP, Alireza Sadighian, 2008 符號說明
資料來源: [7]

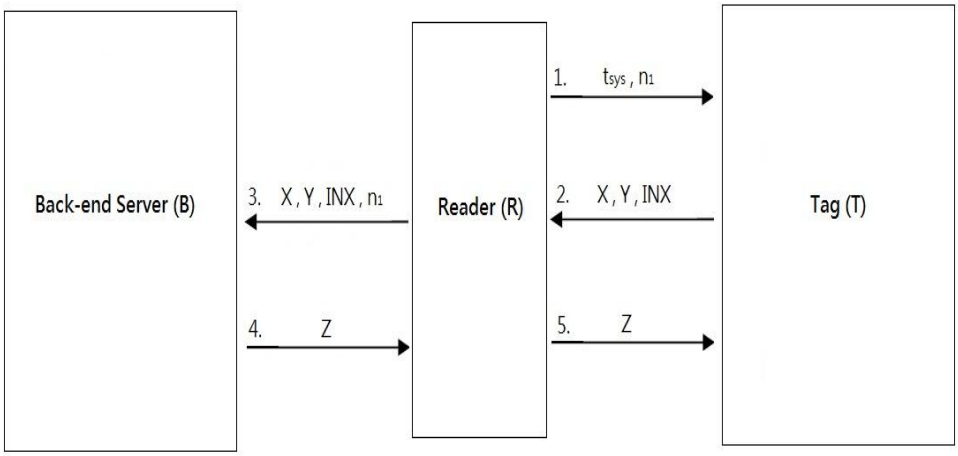

圖 19: FLMAP, Alireza Sadighian, 2008 認證流程
資料來源: [7]

圖 18 為此 RFID 認證協定所用到符號的說明圖，圖 19 為此 RFID 認證協定的流程。接下來將一步一步說明:

Step 0: 初始化動作。一開始標籤會儲存著 ID、K、t_{tag}、t_{max} 和 INX，其中 ID 為標籤的識別碼；K = K_1 || K_2，K_1 和 K_2 皆為 96 bits，為標籤的 key；t_{tag} 用來儲存標籤上一次收到合法 Query 的時間點，初始值設為標籤的出廠時間即可；t_{max} 也是儲存著某個時間點，可以視為標籤的使用期限；INX 為標籤在後端伺服器資料庫中的索引。後端伺服器在資料庫中必須儲存著每一個標籤的認證資料，ID、K 和 K_{last}。

Step 1: 讀取器產生一個隨機亂數 n_1，發出 Query 給標籤，將 t_{sys} 和 n_1 傳送給標籤，其中 t_{sys} 為目前 RFID 系統的時間。

Step 2: 標籤收到後，檢查 t_{sys} 是否大於 t_{tag}，若不是，則代表此為非法的 Query，標籤產生一個隨機亂數 n_2，傳送給讀取器，由於 n_2 對讀取器而言無實質意義，此動作意義如同終止此次通訊；若是，則再檢查 t_{sys} 是否小於 t_{max}，若不是，則代表此標籤已經超過使用期限，標籤產生一個隨機亂數 n_2，傳送給讀取器，由於 n_2 對讀取器而言無實質意義，此動作意義如同終止此次通訊；若是，標籤產生一個隨機亂數 n_2，並且計算 $X = K_1 V (n_1 \oplus n_2)$和 $Y = K_1 \oplus ID \oplus n_2$，將 X、Y 和 INX 傳送給讀取器。

Step 3: 讀取器收到後，將收到的 X、Y 和 INX，連同 n_1 一起傳送給後端伺服器。

Step 4: 後端伺服器收到後，利用 INX 自資料庫中找到該標籤的資料，取出該標籤的 ID、K_1 和 K_2。利用 K_1 與收到的 X 和 n_1 做運作萃取出 n_2，再計算 $Y' = K_1 \oplus ID \oplus n_2$，比對 Y'與收到的 Y 是否相同，若不同，則終止此次通訊；若相同，則代表該標籤通過 Reader-to-Tag 的認證，為合法標籤，接下來，後端伺服器計算 $Z = K_2 \oplus (ID \wedge n_2)$，計算完後將 Z 傳送給讀取器，最後，後端伺服器會更新該標籤的資料，K_{last}、K 和 INX，更新方式為 $K_{last} = K$、$K_1 = K_2 \wedge (n_1 \oplus n_2)$、$K_2 = K_1 \wedge (n_1 \oplus n_2)$、$INX = (INX \vee ID) \oplus n_2$。

Step 5-1: 讀取器收到後，將 Z 傳送給標籤。

Step 5-2: 標籤收到後，計算 $Z' = K_2 \oplus (ID \wedge n_2)$，比對 Z'與收到的 Z 是否相同，若不相同，則終止此次通訊；若相同，則代表該讀取器通過 Tag-to-Reader 的認證，為合法的讀取器，接下來，標籤會做更新資料的動作，更新方式為 $t_{tag} = t_{sys}$、$K_1 = K_2 \wedge (n_1 \oplus n_2)$、$K_2 = K_1 \wedge (n_1 \oplus n_2)$、$INX = (INX \vee ID) \oplus n_2$。

安全性分析

此 RFID 認證協定有做到一般常見 RFID 攻擊的相關防禦措施，因此，大部分的 RFID 攻擊都無法有效的造成傷害。此 RFID 認證協定使用時間類型的 Challenge-response 認證，標籤只有在收到合法的 Query，$t_{tag} < t_{sys} < t_{max}$，才會回應有意義的訊息給讀取器，若攻擊者使用 Replay attack 傳送上一回合的 t_{sys} 給標籤，則標籤會因為 $t_{sys} < t_{tag}$ 而認為此為非法 Query，並且回應無意義的訊息給讀取器，此方法可以有效的抵抗 Reply attack。

但在 Tag tracing 方面，此 RFID 認證協定利用每一回合認證結束時會對標籤的 INX 和 K 做更新動作及隨機亂數 n_1 和 n_2，使得標籤傳送給讀取器的訊息，X、Y 和 INX，每一回合都不相同，藉此方法來抵抗 Tag tracing。但由於標籤必須等到收到認證協定中最後一個訊息，並且判斷對方是合法的讀取器後，才會做資料更新的動作，也就是說，INX 只有在經過合法的交易回合後，才會因為資料更新而不同。考慮到下列情況，若攻擊者正在對目標標籤做 Tag tracing，且該目標標

籤在短時間內都不會發生合法的交易，因此，標籤每一次所傳送回來的 INX 皆相同，攻擊者就可以利用 INX 對目標標籤進行追蹤，在這種情況下，此 RFID 認證協定有著 Tag tracing 的風險。另外，此 RFID 認證協定在每一回合認證結束後，後端伺服器和標籤都必須做資料更新的動作，因此，必須顧慮到 Desynchronization attack，而此 RFID 認證協定的補救措施為後端伺服器會額外儲存標籤舊的 key K_{last}，目是是希望當後端伺服器與標籤發生資料不一致的情況時(後端伺服器的資料已更新、標籤的資料未更新)，該標籤下一次進行認證時，後端伺服器還是能用該標籤舊的 key K_{last} 完成標籤的認證，但是，此認證流程在每一回合認證時，無論該回合所使用的 key 是新的還是舊的，在回合結束時，後端伺服器皆會進行 key 更新的動作，而此種處理方式，在以下的例子中，仍然會造成標籤無法再被辨識的報廢情況發生:

回合 1: 認證結束。後端伺服器更新 key K 為 K'、K_{last} 為 K。發生 Desynchronization attack，標籤沒有做更新 key K 的動作，即標籤的 key K 為 K。

回合 2: 標籤仍可使用舊的 key K 完成認證，認證結束。後端伺服器更新 key K' 為 K"、K_{last} 為 K'。發生 Desynchronization attack，標籤沒有做更新 key K 的動作，即標籤的 key K 仍為 K。

回合 3: 標籤無法使用舊的 key K 通過認證，即標籤無法再被辨識。

由上例可以看出，在此種情況下，此 RFID 認證協定有著 Desynchronization attack 的風險存在。

效能分析

此 RFID 認證協定在後端伺服器和標籤方面僅使用了 AND、OR 和 XOR 運算。在標籤方面，因為僅使用 AND、OR 和 XOR 運算，故其運算量低於其他 RFID 認證協定；在後端伺服器方面，因為僅使用 AND、OR 和 XOR 運算，且使用次數固定，並不會與 RFID 系統中的標籤數量成正比關係，故後端伺服器的運算量低於其他 RFID 認證協定。

另外，考慮到後端伺服器查詢標籤資料所花費的時間，後端伺服器可以利用 INX 直接找到該標籤的資料，省去在資料庫中逐一查詢的動作，查詢時間的花費為 O(1)，大幅的減少查詢標籤資料所花費的時間，因此，在標籤數量眾多或交易次數頻繁的 RFID 系統中，後端伺服器不需花費太多時間在查詢標籤資料上，故可以提升系統的效能。

3.3.3 Jianqing Fu, 2010

這是由 Jianqing Fu 等人在 2010 年所提出的 RFID 認證協定[8]。接下來將依序介紹此 RFID 認證協定的認證流程、安全性分析和效能分析。

認證流程

符號	說明
K	Key of tag, shared between tag and the server
K_s	Secret key of server
IDT	Index of tag
IDTA	Alias of tag
H( )	Hash function
E_{Ks}( )	Symmetric encryption with key K_s

圖 20: Jianqing Fu, 2010 符號說明
資料來源: [8]

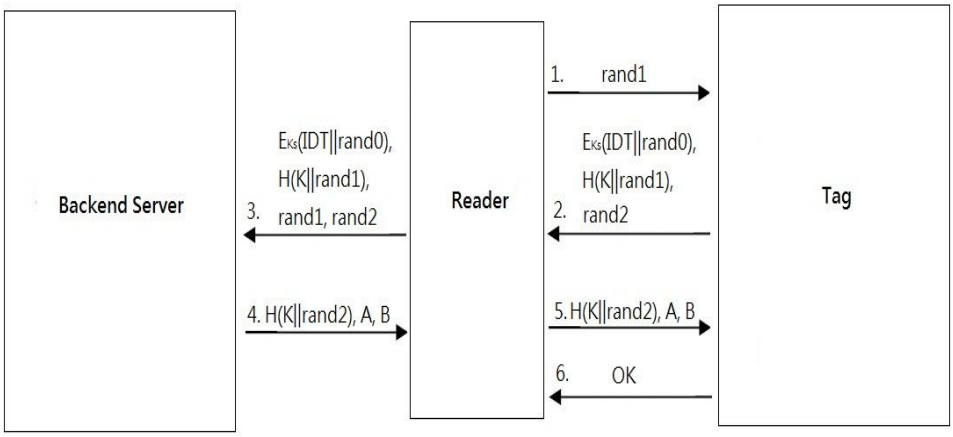

圖 21: Jianqing Fu, 2010 認證流程
資料來源: [8]

圖 20 為此 RFID 認證協定所用到符號的說明圖，圖 21 為此 RFID 認證協定的流程。接下來將一步一步說明:

Step 0: 初始化動作。一開始標籤會儲存著 IDT、K 和 IDTA，其中 IDT 為標籤的識別碼，96 bits；K 為標籤的 key，96 bits；IDTA = E_{Ks}(IDT||rand0)，為標籤識別碼的別名，96 bits，每一回合認證結束都會進行更新的動作，由後端伺服器產生並配置給標籤。後端伺服器儲存著自己的密鑰 K_s，並在資料庫中儲存著每一個標籤的認證資料，K。

Step 1: 讀取器產生一個隨機亂數 rand1，發出 Query 給標籤，並且連同 rand1 一起傳送給標籤。

Step 2: 標籤收到後，產生一個隨機亂數 rand2，並且利用收到的 rand1 和自己的 key K，計算 H(K||rand1)，計算完後將 IDTA、H(K||rand1) 和 rand2，即

E_{Ks}(IDT||rand0) 、H(K||rand1)和 rand2，傳送給讀取器。

Step 3: 讀取器收到後，將收到的 E_{Ks}(IDT||rand0) 、H(K||rand1)和 rand2 連同 rand1 一起傳送給後端伺服器。

Step 4: 後端伺服器收到後，使用自己的密鑰 K_s 對收到的 E_{Ks}(IDT||rand0)進行解密的動作，解密完成後，取得 IDT，利用 IDT 到資料庫中找到該標籤的資料，取出該標籤的 key K，利用 K 和收到的 rand1 計算 H(K||rand1)，比對計算結果與收到的 H(K||rand1)是否相同，若不相同，則終止此次通訊；若相同，則代表該標籤通過 Reader-to-Tag 的認證，為合法標籤，接下來，後端伺服器會產生一個隨機亂數 rand0'，並計算 H(K||rand2)、IDTA' = E_{Ks}(IDT||rand0')、A = IDTA' ⊕H(K||rand1||rand2)和 B = IDTA' ⊕H(K||rand2||rand1)，最後將 H(K||rand2)、A 和 B 傳送給讀取器。

Step5-1: 讀取器收到後，將 H(K||rand2)、A 和 B 傳送給標籤

Step 5-2: 標籤收到後，計算 H(K||rand2)，比對計算結果與收到的 H(K||rand2)是否相同，若不相同，則終止此次通訊；若相同，則代表該讀取器通過 Tag-to-Reader 的認證，為合法讀取器，接下來，標籤使用收到的 A 和 B 計算 IDTA1 = A⊕H(K||rand1||rand2)和 IDTA2 = A⊕H(K||rand2||rand1)，計算完後，比對 IDTA1 和 IDTA2 是否相同，若相同，則標籤進行更新 IDTA 的動作，更新方式為 IDTA = IDTA1。

安全性分析

此 RFID 認證協定有做到一般常見 RFID 攻擊的相關防禦措施，因此，大部分的 RFID 攻擊都無法有效的造成傷害。在此 RFID 認證中，每一回合認證結束時，後端伺服器會計算 IDTA' = E_{Ks}(IDT||rand0')，並且傳送給標籤，讓標籤進行 IDTA 的更新動作，由於這只是標籤單方面的資料更新，即使發生 Desynchronization attack 使得標籤沒有做 IDTA 的更新動作，該標籤在下一次認證時，會傳送 IDTA = E_{Ks}(IDT||rand0)給後端伺服器，後端伺服器仍然能夠使用自己的密鑰 K_s 對 IDTA 進行解密取得 IDT，並且順利的完成標籤的認證，因此，此 RFID 認證協定不用顧慮 Desyncrhonization attack。

但在 Tag tracing 方面，此 RFID 認證協定利用每一回合認證結束時會對標籤的 IDTA 做更新動作及隨機亂數 rand1 和 rand2，使得標籤傳送給讀取器的訊息，IDTA、H(K||rand1)和 rand2，每一回合都不相同，藉此方法來抵抗 Tag tracing。但由於標籤必須等到收到認證協定中最後一個訊息，並且判斷對方是合法的讀取器後，才會做資料更新的動作，也就是說，IDTA 只有在經過合法的交易回合後，才會因為資料更新而不同。考慮到下列情況，若攻擊者正在對目標標籤做 Tag

tracing，且該目標標籤在短時間內都不會發生合法的交易，因此標籤每一次所傳送回來的 IDTA 皆相同，攻擊者就可以利用 IDTA 對目標標籤進行追蹤，在這種情況下，此 RFID 認證協定有著 Tag tracing 的風險。

效能分析

此 RFID 認證協定使用到對稱型加密演算法 $E_{Ks}(\ )$、雜湊函數 H( )和 XOR 運算。在標籤方面，因為使用到雜湊函數，故其運算量相較於其他 RFID 認證協定不算低；在後端伺服器方面，使用了對稱型加密演算法和雜湊函數，但使用次數固定，並不會與 RFID 系統中的標籤數量成正比關係，且後端伺服器可以利用 IDT 快速找到標籤的資料，省去在資料庫中逐一查詢的動作，因此，在標籤數量眾多的 RFID 系統環境底下，後端伺服器雖然使用到對稱型加密演算法，但其運算量相較於其他 RFID 認證協定並不算高。

另外，考慮到後端伺服器查詢標籤資料所花費的時間，後端伺服器可以利用 IDT 直接找到該標籤的資料，省去在資料庫中逐一查詢的動作，查詢時間的花費為 O(1)，大幅的減少查詢標籤資料所花費的時間，因此，在標籤數量眾多或交易次數頻繁的 RFID 系統中，後端伺服器不需花費太多時間在查詢標籤資料上，有助於提升 RFID 系統的效能。

第四章 RFID 認證協定設計

本章將介紹本論文所提出的 RFID 認證協定，Secure and Efficient Mutual Authentication Protocol(SEMAP)。首先在 4.1 節將概括性的介紹 SEMAP，包含系統架構和設計重點，接下來在 4.2 節將說明 SEMAP 的環境假設，4.3 節將說明 SEMAP 所用到的符號其所代表的意義為何，最後在 4.4 節將介紹 SEMAP 的認證流程，而基於效率與安全的考量，會提出兩種版本的 SEMAP，分別為 Secure and Efficient Mutual Authentication Protocol version 1(SEMAPv1) 和 Secure and Efficient Mutual Authentication Protocol version 2(SEMAPv2)，在此節中將詳細地說明 SEMAPv1 和 SEMAPv2 的認證流程。

4.1 SEMAP 概述

在此節中，首先將在 4.1.1 中介紹 SEMAP 系統架構，並在 4.1.2 介紹 SEMAP 設計重點。此節的目的主要是先概括性的介紹 SEMAP，讓讀者先對 SEMAP 有了些初步的認識，稍後在詳細說明 SEMAP 的認證流程時，可以幫助讀者更易於理解 SEMAP 的設計理念。

4.1.1 SEMAP 系統架構

在系統架構上，SEMAP 遵照 Gen2 所制定的標準，在 Gen2 標準中，讀取器存取標籤必須經過三個階段，分別為 Select、Inventory 和 Access 階段，如圖 6，接下來依序介紹各個階段的過程:

1. Select 階段:

在此階段中，讀取器會將過程分為四個 sessions 以及設定 Selected flag 變數值，利用 session 和 flag 變數將標籤分配給讀取器，並且判別該標籤是否已經進入存取排程。此階段運作結束之後，所有處在讀取器存取範圍內的標籤皆會分配給該讀取器，當標籤完成要被讀取器存取前的初始化設定後，才能進入下一個階段。而 SEMAP 沒有更改到此階段原有的運作。

2. Inventory 階段:

在此階段中，將對 Select 階段所選取的標籤進行識別，SEMAP 將負責此一識別工作，通過 SEMAP 認證的標籤即為合法標籤，獲准進入下一個階段；否則為非法標籤或者遭受攻擊，必須終止通訊。因此，本論文所提出的 RFID 認證協定主要是加強 Inventory 階段的標籤識別工作，防止任何不合法的存取行為，並且防禦各種 RFID 攻擊，以確保個人隱私的保密性以及 RFID 系統整體的安全性。

3. Access 階段:

此階段為 RFID 系統最主要的功能，存取標籤資料，其資料可能為商品資訊、個人身分、儲值卡金額等各種訊息，適 RFID 所應用的環境而定。當讀取器存取完通過 Inventory 階段的所有標籤資料後，Access 階段便可結束，也代表著讀取器完成存取標籤資料的動作。而 SEMAP 沒有更改到此階段原有的運作。

4.1.2 SEMAP 設計重點

RFID 可以為人們帶來方便，在許多領域之中可以大幅的提升產能，但卻因為潛藏且尚未解決的安全問題而無法全面性的佈署及使用，因此，本論文為此提出了新的 RFID 認證協定，SEMAP，以下分為兩個層面，分別為安全層面和效率層面，說明 SEMAP 所要解決的問題和 SEMAP 的設計重點:

安全層面

雖然 Gen2 標準中有提供基本的安全防護機制，但單靠 Gen2 的安全防護機制無法有效且徹底的保障個人隱私和 RFID 系統安全，因此，許許多多的 RFID 認證協定相繼被提出來，為的就是要解決這些安全問題，而 SEMAP 在安全層面的目的也是如此。在個人隱私方面，SEMAP 必須確保資料不會外洩，並且要能夠抵抗 Tag tracing、Forward security 和 Backward security 等 RFID 攻擊，讓 RFID 系統不會因為遭受此類型的攻擊，造成個人隱私洩漏的情況；在系統安全方面，SEMAP 要能夠抵抗 Eavesdropping、Tag cloning、Impersonation、Replay attack、Desynchronization attack 和 Compromising attack 等 RFID 攻擊，讓 RFID 系統不會因為遭受此類型的攻擊，造成系統無法正常運作的情況。

除了僅由讀取器單方面的驗證標籤是否合法的單向認證外，我們希望 SEMAP 可以做到讀取器可以驗證標籤是否合法(Reader-to-Tag)，並且標籤也可以驗證讀取器是否合法(Tag-to-Reader)的雙向認證機制，加強 SEMAP 認證過程中的安全性。

效率層面

在效率方面，再分為兩個部分探討，分別為標籤和後端伺服器。在標籤部分，由於生產成本上的考量，標籤的電路設計不會提供太多的功能，而這項硬體限制，使得標籤本身無法使用過於複雜的運算，如對稱/非對稱型加密演算法、雜湊函數等，因此，RFID 認證協定必須要讓標籤僅使用較簡單且快速的運算而能安全的完成認證，在此考量下，SEMAP 的標籤，在認證過程中僅使用了 XOR 運算即可完成認證；在後端伺服器方面，大部分的 RFID 認證協定會憑藉著一筆標籤所給定資料到後端伺服器查詢，此時後端伺服器會將該筆資料與資料庫中每一筆項目的某個欄位內容做比較，逐一比對是否相同，如果相同則代表找到該標籤的相關資料，此查詢時間為 O(N)，其中，N 為 RFID 系統的標籤數量，而考量到大多的 RFID 應用環境，其標籤數量眾多，如賣場、圖書館等，此種花費時

間與 RFID 系統的標籤數量成正比的查詢標籤資料方式會使得 RFID 系統的效率低落，而在此考量下，SEMAP 使用 IDX 技巧來降低後端伺服器查詢標籤資料所花費的時間，使此查詢時間為 O(1)，與 RFID 系統的標籤數量無關，讓 SEMAP 更加適合在標籤數量眾多的 RFID 系統中使用。

4.2 SEMAP 環境假設

此節將說明本論文所提出的 RFID 認證協定能夠正常運作的前提環境假設，如下:

1. 讀取器與標籤之間的通訊是處於開放且不安全的環境底下，通訊過程中所傳送的訊息可能會遭到攻擊者竊聽或修改等可能。

2. 讀取器與後端伺服器之間的通訊處於安全的環境底下，通訊過程中所傳送的訊息不會遭到攻擊者竊聽或修改等可能。

3. RFID 系統中的每一個標籤都有足夠的 ROM 能夠儲存後端伺服器配置給標籤的識別資料。

4. RFID 系統中的每一個標籤都有足夠 NVRAM 能夠儲存每一次認證結束後標籤必須更新的認證資料。

5. RFID 系統中的每一個標籤都有產生隨機亂數的能力並且能夠使用 XOR 運算。

6. 後端伺服器中有一張亂數表，此表共有 65536 個項目，每個項目的內容皆為一個 32 bits 的隨機亂數，由後端伺服器自行產生。

7. 後端伺服器能夠使用單向雜湊函數(One-way hash function) h( )，且此 h( )為安全的單向雜湊函數，即滿足下列兩個性質:
   i. 假設 y = h(x)，在不知道 x 的前提之下，給定 y 無法逆推得到 x。
   ii. 給定 x 和 h(x)，則找不到 y 使得 h(y) = h(x)。

其中，第四點和第七點僅為 SEMAPv2 的環境假設，SEMAPv1 不需要此兩項環境假設。

4.3 SEMAP 符號說明

此節將說明本論文所提出的 RFID 認證協定，所用到的符號其實際代表的意義為何，如下:

1. r_i: 16 bits 隨機亂數。SEMAP 認證過程中共會使用到三個隨機亂數，分別為 r_1、r_2 和 r_3。

2. X: 16 bits 整數，代表後端伺服器亂數表的索引。

3. Y: 16 bits 整數，代表後端伺服器亂數表的索引。

4. [ ]: 代表後端伺服器亂數表中某個項目的內容，也就是一個 32 bits 的隨機亂數。下圖 22 為舉例說明此符號的用法。

圖 22: 舉例說明[ ]的用法

5. ⊕: XOR 運算子。

6. ||: 串接(concatenator)運算子。

7. ß: 後端伺服器預設的 16 bits 整數，以 2 進位表示為 1010101010101010。

8. h( ): 單向雜湊函數(One-way hash function)。

9. IDX: 80 bits 整數，由 3 個 16 bits 整數和 1 個 32 bits 整數串接而成。代表標籤在後端伺服器資料庫中的索引的別名，其中隱藏著該標籤真正在後端伺服器資料庫中的索引。此資料由後端伺服器產生後，並且配置給標籤。
例: IDX = X || Y || X⊕Y⊕ß || Index⊕[X]⊕[Y]⊕[X⊕Y⊕ß]，其中，Index 即為該標籤真正在後端伺服器資料庫中的索引。

10. K: 32 bits 整數，$K = K_1 \| K_2$，其中，K_1 和 K_2 皆為 16 bits 整數。代表標籤的 key，每個標籤的 key 皆不相同。此資料由後端伺服器產生後，並且配置給標籤。

11. K_{old} : 32 bits 整數，$K_{old} = K_{1old} \| K_{2old}$，其中，$K_{1old}$ 和 K_{2old} 皆為 16 bits 整數。代表標籤舊的 key，即標籤上一次認證時所使用的 key。

4.4 SEMAP 認證協定

此節將詳細介紹本論文所提出的 RFID 認證協定。依據後端伺服器與標籤在認證結束後，是否要做資料更新動作的與否，提出兩種 RFID 認證協定，分別是 SEMAPv1 和 SEMAPv2，其中，SEMAPv1 是屬於認證結束後，後端伺服器與標籤皆不需要做資料更新動作的 RFID 認證協定，而 SEMAPv2 是屬於認證結束後，後端伺服器與標籤皆需要做資料更新動作的 RFID 認證協定。因為在每一次認證後，後端伺服器與標籤更新認證所需資料的動作會對 RFID 認證協定帶來一定程度上的好處和一定程度上的壞處，故此兩種認證協定有著各自的優點和缺點，且呈現互補的情況。接下來將在 4.4.1 介紹 SEMAPv1 的認證流程，接著在 4.4.2 介紹 SEMAPv2 的認證流程。

4.4.1 SEMAPv1

現在將介紹本論文所提出的第一個 RFID 認證協定，Secure and Efficient Mutual Authentication Protocol version 1(SEMAPv1)，是屬於認證完成後，後端伺服器與標籤皆不需要做資料更新動作的 RFID 認證協定。此類型的 RFID 認證協定，因為後端伺服器與標籤皆不需要顧慮資料是否已同步更新的問題，故其認證過程較為簡單、快速，且整個過程所需的運算量也較低，但若 RFID 系統出現偽造標籤時，此類型的 RFID 認證協定沒有讓偽造標籤失效，而讓原本的合法標籤能夠繼續使用的能力，故在安全性上較為不足。接下來將在 4.4.1.1 說明 SEMAPv1 的前置作業要做哪些事情，接著在 4.4.1.2 詳細說明 SEMAPv1 的認證流程。

4.4.1.1 SEMAPv1 前置作業

此階段將說明 SEMAPv1 的初始化設定，分成兩個部分，分別是後端伺服器和標籤，以下將依序介紹此兩部分初始化設定的內容:

後端伺服器:

在此部分的初始化設定有三件事，分別是 ß、後端伺服器亂數表和標籤資料庫，依序介紹如下:

1. ß: 後端伺服器必須先設定好 16 bits 整數 ß 之值，以 2 進位表示為

1010101010101010。

2. 後端伺服器亂數表: 後端伺服器必須先設定好此亂數表的內容，即後端伺服器要自行產生 65536 個 32 bits 隨機亂數，並依序填入亂數表的項目之中，亂數表的每一個項目之內容即為一個 32 bits 隨機亂數。

3. 標籤資料庫: 此資料庫會存放標籤所有的相關資料，包括認證時所需要的資料。每當一個標籤向後端伺服器註冊成為該 RFID 系統的合法標籤後，後端伺服器會配置一把 32 bits key K 給該標籤，其中 $K = K_1 \| K_2$，可視為兩個 16 bits 整數 K_1 和 K_2，且每一個標籤的 key 皆不能相同，而後端伺服器必須將此 key K 存放於資料庫中，作為該標籤未來認證之用。

標籤:

在此部分的初始化設定有兩件事，分別是 IDX 和 K，依序介紹如下:

1. IDX: 當標籤向後端伺服器註冊成為合法標籤後，後端伺服器會產生 IDX，並且將 IDX 配置給此標籤。IDX 為一個 80 bits 整數，由 3 個 16 bits 整數和 1 個 32 bits 整數串接而成。
   例: IDX = X ‖ Y ‖ X⊕Y⊕ß ‖ Index⊕[X]⊕[Y]⊕[X⊕Y⊕ß]，以此例說明 IDX 的產生過程:
   i. X: 後端伺服器產生的 16 bits 隨機亂數，目的是作為存取後端伺服器亂數表的索引之用。
   ii. Y: 後端伺服器產生的 16 bits 隨機亂數，目的是作為存取後端伺服器亂數表的索引之用。
   iii. X⊕Y⊕ß: 後端伺服器計算 X⊕Y⊕ß，其運算結果為一個 16 bits 整數，目的也是作為存取後端伺服器亂數表的索引之用。
   iv. Index⊕[X]⊕[Y]⊕[X⊕Y⊕ß]: 後端伺服器計算 Index⊕[X]⊕[Y]⊕[X⊕Y⊕ß]，其中 Index 為該標籤在後端伺服器標籤資料庫中的索引，而[X]、[Y]、[X⊕Y⊕ß]為 3 個後端伺服器亂數表中的隨機亂數，用法可以參考圖 22。
   最後將上述四個步驟的結果串接起來，即將 X、Y、X⊕Y⊕ß 和 Index⊕[X]⊕[Y]⊕[X⊕Y⊕ß]串接起來成為 IDX。

2. K: 當標籤向後端伺服器註冊成為合法標籤後，後端伺服器會產生 K，並且將 K 配置給此標籤，作為此標籤的 key。K 為 32 bits 整數，可視為兩個 16 bits 整數 K_1 和 K_2 的串接，即 $K = K_1 \| K_2$，且每一個標籤的 key 皆不能相同。

下圖 23 為 SEMAPv1 初始化設定後所配置的內容。

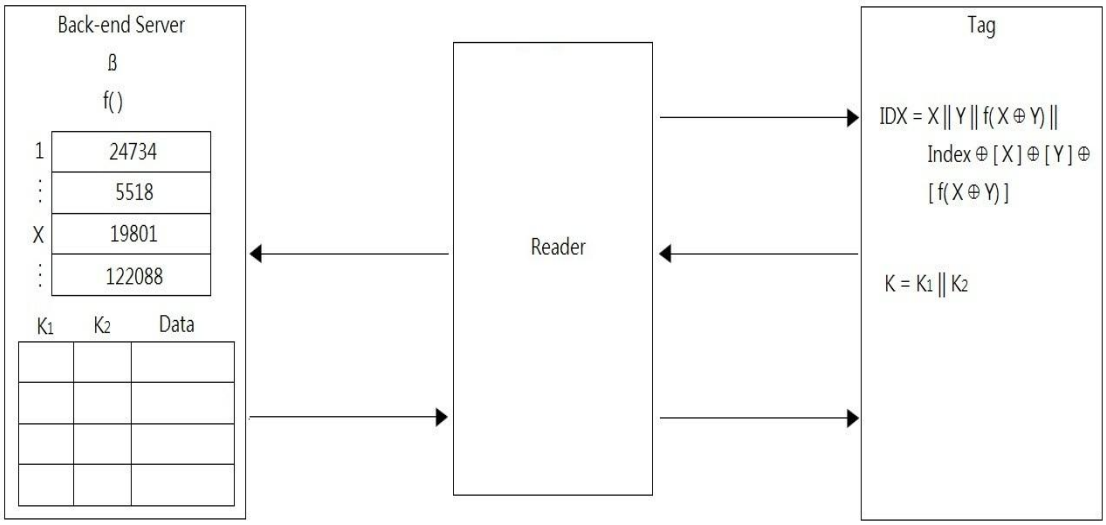

圖 23: SEMAPv1 初始化設定內容

4.4.1.2 SEMAPv1 認證流程

此階段將介紹 SEMAPv1 的認證流程，下圖 24 為 SEMAPv1 的認證流程圖，接下來將詳細的說明每個步驟:

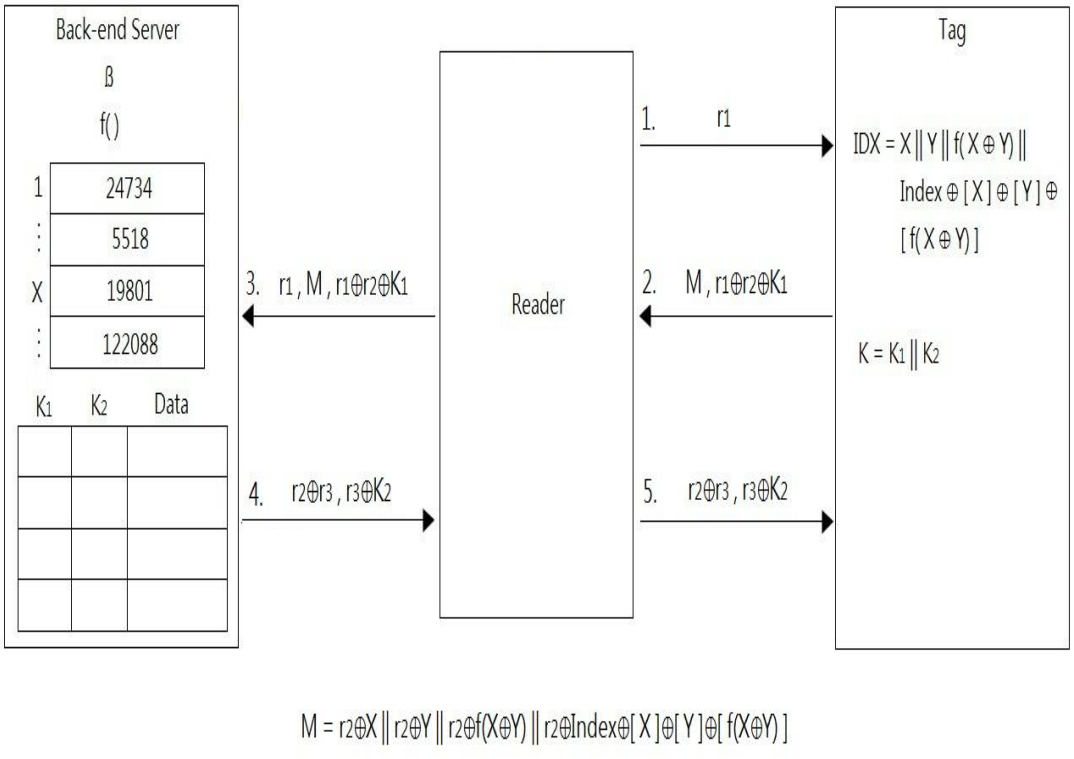

圖 24: SEMAPv1 認證流程

Step 1: 讀取器產生一個隨機亂數 r_1，發出 Query 給標籤，並且連同 r_1 一起傳送給標籤。

Step 2: 此步驟的目的為標籤產生要傳送給後端伺服器的驗證訊息。標籤收到後，產生一個隨機亂數 r_2，接下來利用自己的 IDX 產生 M，作法為逐一取出 IDX 的每一個部分，分別與 r_2 做 XOR 運算，如下:

i. 取出 IDX 的第一個部分 X 與 r_2 做 XOR 運算，產生 $r_2 \oplus X$。
ii. 取出 IDX 的第二個部分 Y 與 r_2 做 XOR 運算，產生 $r_2 \oplus Y$。
iii. 取出 IDX 的第三個部分 $X \oplus Y \oplus ß$ 與 r_2 做 XOR 運算，產生 $r_2 \oplus X \oplus Y \oplus ß$。
iv. 取出 IDX 的第四個部分 $Index \oplus [X] \oplus [Y] \oplus [X \oplus Y \oplus ß]$ 與 r_2 做 XOR 運算產生 $r_2 \oplus Index \oplus [X] \oplus [Y] \oplus [X \oplus Y \oplus ß]$。

將上述四個步驟的結果串接起來，即將 $r_2 \oplus X$、$r_2 \oplus Y$、$r_2 \oplus X \oplus Y \oplus ß$ 和 $r_2 \oplus Index \oplus [X] \oplus [Y] \oplus [X \oplus Y \oplus ß]$ 串接起來成為 M，即 $M = r_2 \oplus X \| r_2 \oplus Y \| r_2 \oplus X \oplus Y \oplus ß \| r_2 \oplus Index \oplus [X] \oplus [Y] \oplus [X \oplus Y \oplus ß]$。再來取出自己 key K 的上半部 K_1，計算 $r_1 \oplus r_2 \oplus K_1$，最後將計算結果與 M 一起傳送給讀取器。

Step 3: 讀取器收到後，將收到的 M 和 $r_1 \oplus r_2 \oplus K_1$ 連同 **Step 1** 產生的隨機亂數 r_1，一起傳送給後端伺服器。

Step 4-1: 此步驟的目的為找出該標籤的 Index 以便下一步驟的認證。後端伺服器收到後，利用 ß 和收到的 M 進行下列的運算處理後，找出此標籤的 Index:

i. 取出 M 的第一個部分 $r_2 \oplus X$ 和 M 的第二個部分 $r_2 \oplus Y$，將 $r_2 \oplus X$ 與 $r_2 \oplus Y$ 做 XOR 運算取得 $X \oplus Y$，即 $(r_2 \oplus X) \oplus (r_2 \oplus Y) = X \oplus Y$。
ii. 計算 $X \oplus Y \oplus ß$，即取得 $X \oplus Y \oplus ß$。
iii. 取出 M 的第三個部分 $r_2 \oplus X \oplus Y \oplus ß$，將其與 ii.取得的 $X \oplus Y \oplus ß$ 做 XOR 運算取得 r_2，即 $(r_2 \oplus X \oplus Y \oplus ß) \oplus X \oplus Y \oplus ß = r_2$。
iv. 利用 iii.取得的 r_2，分別與 M 的第一個部分 $r_2 \oplus X$ 和 M 的第二個部分 $r_2 \oplus Y$ 做 XOR 運算取得 X 和 Y，即 $r_2 \oplus (r_2 \oplus X) = X$、$r_2 \oplus (r_2 \oplus Y) = Y$。
v. 利用 iv.取得的 X、Y 和 ii.取得的 $X \oplus Y \oplus ß$ 作為索引到後端伺服器亂數表中取得 $[X]$、$[Y]$ 和 $[X \oplus Y \oplus ß]$。
vi. 取出 M 的第四個部分 $r_2 \oplus Index \oplus [X] \oplus [Y] \oplus [X \oplus Y \oplus ß]$，將其與 iii.取得的 r_2 和 v.取得的 $[X]$、$[Y]$ 和 $[X \oplus Y \oplus ß]$ 做 XOR 運算取得 Index，即 $(r_2 \oplus Index \oplus [X] \oplus [Y] \oplus [X \oplus Y \oplus ß]) \oplus r_2 \oplus [X] \oplus [Y] \oplus [X \oplus Y \oplus ß] = Index$。

Step 4-2: 此步驟的目的為 Reader-to-Tag 認證。後端伺服器利用 **Step 4-1** 取得的 Index 作為索引到標籤資料庫中取出該索引所指標籤的 key K 上半部 K_1，取得後，計算 $r_1 \oplus r_2 \oplus K_1$，並將計算結果與收到的 $r_1 \oplus r_2 \oplus K_1$ 進行比較，若不相同，

則代表該標籤為非法標籤或者遭受 RFID 攻擊，故終止此次通訊；若相同，則代表該標籤通過 Reader-to-Tag 認證，為合法標籤，並進行下一步驟。

Step 4-3: 此步驟的目的為後端伺服器產生要傳送給標籤的驗證訊息。後端伺服器此時會產生一個隨機亂數 r_3，計算 $r_2 \oplus r_3$，並自標籤資料庫中取出該標籤的 key K 下半部 K_2，計算 $r_3 \oplus K_2$，最後將 $r_2 \oplus r_3$ 和 $r_3 \oplus K_2$ 一起傳送給讀取器。

Step 5-1: 讀取器收到後，直接將收到的 $r_2 \oplus r_3$ 和 $r_3 \oplus K_2$ 傳送給標籤。

Step 5-2: 此步驟的目的為 Tag-to-Reader 認證。標籤收到後，利用自己在 **Step 2** 所產生的 r_2 與收到的 $r_2 \oplus r_3$ 做 XOR 運算取得 r_3，即 $r_2 \oplus (r_2 \oplus r_3) = r_3$，再利用取得的 r_3 與收到的 $r_3 \oplus K_2$ 做 XOR 運算取得 K_2，即 $r_3 \oplus (r_3 \oplus K_2) = K_2$，將計算得出的 K_2 與自己的 key K 下半部 K_2 進行比較，若不相同，則代表此為非法讀取器，故終止此次通訊；若相同，則代表該讀取器通過 Tag-to-Reader 認證，為合法讀取器。

4.4.2 SEMAPv2

接下來將介紹本論文所要提出的第二個 RFID 認證協定，Secure and Efficient Mutual Authentication Protocol version 2(SEMAPv2)，是屬於認證結束後，後端伺服器與標籤皆需要做資料更新動作的 RFID 認證協定。此類型的 RFID 認證協定，當發現 RFID 系統中有偽造標籤的存在時，可以利用資料更新的動作，同時修改標籤和該標籤儲存在後端伺服器標籤資料庫中的認證資料，如此一來，能夠讓原本的合法標籤繼續使用，偽造標籤會因為沒有更新認證資料的關係而失效，此功能可以提升 RFID 系統的安全性，但更新資料的動作需要花費更多的運算來完成，故此類型的 RFID 認證協定整體所需的運算量通常會較高，另外，後端伺服器必須在標籤資料庫中儲存額外的資訊，作為後端伺服器和標籤發生資料不一致時的補救措施，故此類型的 RFID 認證協定所需要的儲存空間也較大。接下來將在 4.4.2.1 說明 SEMAPv2 的前置作業要做哪些事情，接著在 4.4.2.2 詳細說明 SEMAPv2 的認證流程。

4.4.2.1 SEMAPv2 前置作業

此階段將說明 SEMAPv2 的初始化設定，分成兩個部分，分別是後端伺服器和標籤，以下將依序介紹此兩部分初始化設定的內容:

後端伺服器:

在此部分的初始化設定有四件事，分別是 ß、h( )、後端伺服器亂數表和標籤資料庫，依序介紹如下:

1. ß: 後端伺服器必須先設定好 16 bits 整數 ß 之值，以 2 進位表示為 1010101010101010。

2. h( ): 後端伺服器必須準備一個安全的雜湊函數 h( )。

3. 後端伺服器亂數表: 後端伺服器必須先設定好此亂數表的內容，即後端伺服器要自行產生 65536 個 32 bits 隨機亂數，並依序填入亂數表的項目之中，亂數表的每一個項目之內容即為一個 32 bits 隨機亂數。

4. 標籤資料庫: 此資料庫會存放標籤所有的相關資料，包括認證時所需要的資料。每當一個標籤向後端伺服器註冊成為該 RFID 系統的合法標籤後，後端伺服器會配置一把 32 bits key K 給該標籤，其中 $K = K_1 \| K_2$，可視為兩個 16 bits 整數 K_1 和 K_2，且每一個標籤的 key 皆不能相同，而後端伺服器必須將此 key K 存放於資料庫中，作為該標籤未來認證之用，並且在每一次認證結束後，更新此 key K，另外，還需要兩個欄位 K_{1old} 和 K_{2old} 存放此標籤舊的 key K_{old}，即上一次認證時所用的 key，此部份的目的是作為當發生後端伺服器與標籤資料不一致時的補救措施，用來抵抗 Desynchronization attack。

標籤:

在此部分的初始化設定有兩件事，分別是 IDX 和 K，依序介紹如下:

1. IDX: 當標籤向後端伺服器註冊成為合法標籤後，後端伺服器會產生 IDX，並且將 IDX 配置給此標籤。IDX 為一個 80 bits 整數，由 3 個 16 bits 整數和 1 個 32 bits 整數串接而成。
   例: IDX = X ‖ Y ‖ X⊕Y⊕ß ‖ Index⊕[X]⊕[Y]⊕[X⊕Y⊕ß]，以此例說明 IDX 的產生過程:
   i. X: 後端伺服器產生的 16 bits 隨機亂數，目的是作為存取後端伺服器亂數表的索引之用。
   ii. Y: 後端伺服器產生的 16 bits 隨機亂數，目的是作為存取後端伺服器亂數表的索引之用。
   iii. X⊕Y⊕ß: 後端伺服器計算 X⊕Y⊕ß，其運算結果為一個 16 bits 整數，目的也是作為存取後端伺服器亂數表的索引之用。
   iv. Index⊕[X]⊕[Y]⊕[X⊕Y⊕ß]: 後端伺服器計算 Index⊕[X]⊕[Y]⊕[X⊕Y⊕ß]，其中 Index 為該標籤在後端伺服器標籤資料庫中的索引，而[X]、[Y]、[X⊕Y⊕ß]為 3 個後端伺服器亂數表中的隨機亂數，用法可以參考圖 22。
   最後將上述四個步驟的結果串接起來，即將 X、Y、X⊕Y⊕ß 和 Index⊕[X]⊕[Y]⊕[X⊕Y⊕ß]串接起來成為 IDX。

2. K: 當標籤向後端伺服器註冊成為合法標籤後，後端伺服器會產生 K，並且將 K 配置給此標籤，作為此標籤的 key。K 為 32 bits 整數，可視為兩個 16 bits 整數 K_1 和 K_2 的串接，即 $K = K_1 \| K_2$，且每一個標籤的 key 皆不能相同。在每一次認證結束後，標籤會更新此 key K。

下圖 25 為 SEMAPv2 初始化設定後所配置的內容。

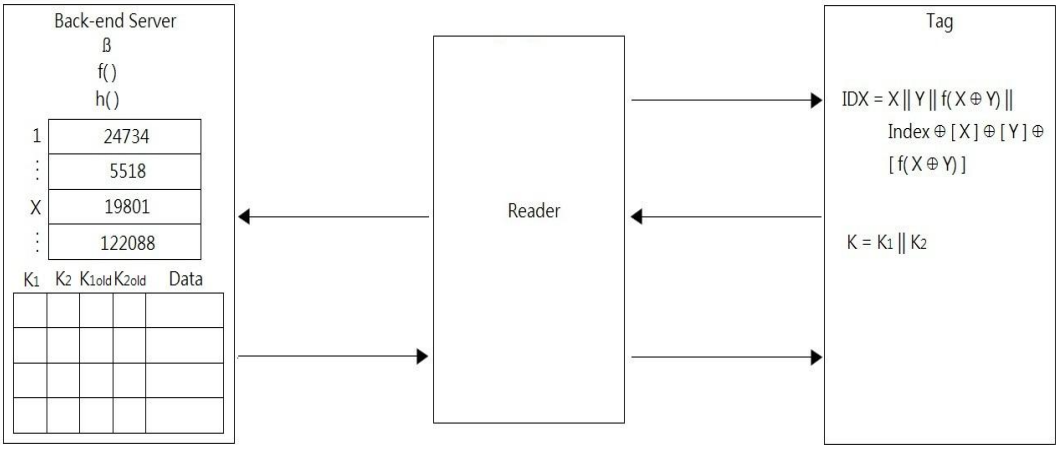

圖 25: SEMAPv2 初始化設定內容

4.4.2.2 SEMAPv2 認證流程

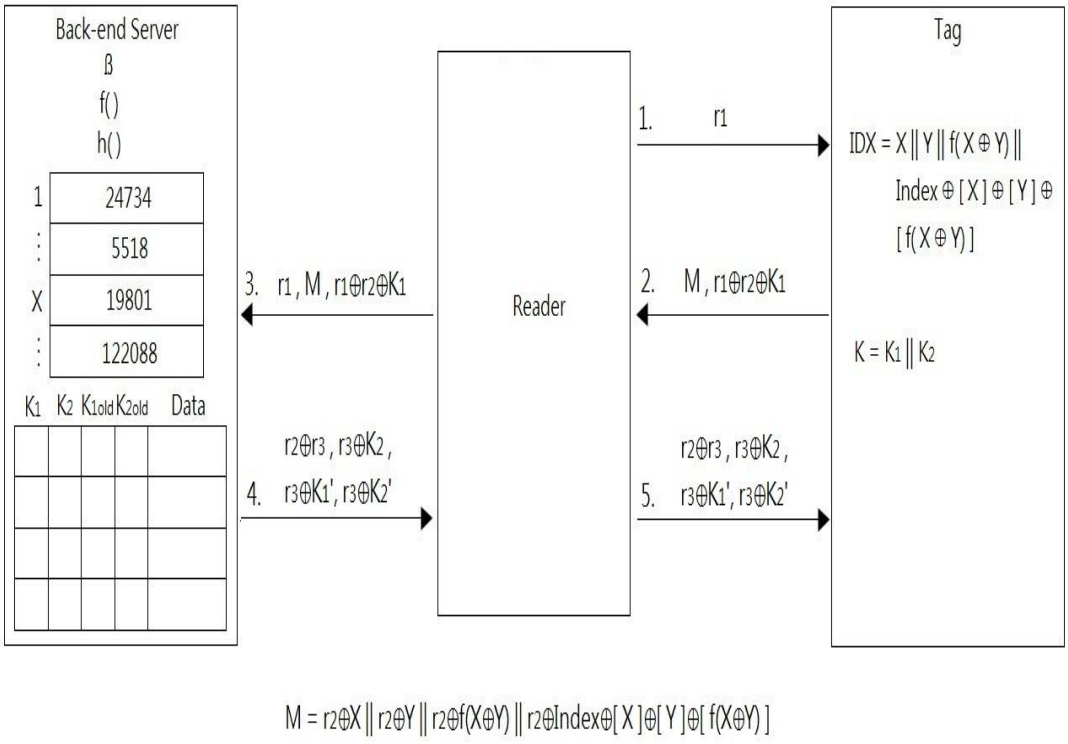

圖 26: SEMAPv2 認證流程

此階段要介紹 SEMAPv2 的認證流程，上圖 26 為 SEMAPv2 的認證流程圖，接下來將詳細的說明每個步驟:

Step 1: 讀取器產生一個隨機亂數 r_1，發出 Query 給標籤，並且連同 r_1 一起傳送給標籤。

Step 2: 此步驟的目的為標籤產生要傳送給後端伺服器的驗證訊息。標籤收到後，產生一個隨機亂數 r_2，接下來利用自己的 IDX 產生 M，作法為逐一取出 IDX 的每一個部分，分別與 r_2 做 XOR 運算，如下:

i. 取出 IDX 的第一個部分 X 與 r_2 做 XOR 運算，產生 $r_2 \oplus X$。
ii. 取出 IDX 的第二個部分 Y 與 r_2 做 XOR 運算，產生 $r_2 \oplus Y$。
iii. 取出 IDX 的第三個部分 $X \oplus Y \oplus ß$ 與 r_2 做 XOR 運算，產生 $r_2 \oplus X \oplus Y \oplus ß$。
iv. 取出 IDX 的第四個部分 $Index \oplus [X] \oplus [Y] \oplus [X \oplus Y \oplus ß]$ 與 r_2 做 XOR 運算產生 $r_2 \oplus Index \oplus [X] \oplus [Y] \oplus [X \oplus Y \oplus ß]$。

將上述四個步驟的結果串接起來，即將 $r_2 \oplus X$、$r_2 \oplus Y$、$r_2 \oplus X \oplus Y \oplus ß$ 和 $r_2 \oplus Index \oplus [X] \oplus [Y] \oplus [X \oplus Y \oplus ß]$ 串接起來成為 M，即 $M = r_2 \oplus X \| r_2 \oplus Y \| r_2 \oplus X \oplus Y \oplus ß \| r_2 \oplus Index \oplus [X] \oplus [Y] \oplus [X \oplus Y \oplus ß]$。再來取出自己 key K 的上半部 K_1，計算 $r_1 \oplus r_2 \oplus K_1$，最後將計算結果與 M 一起傳送給讀取器。

Step 3: 讀取器收到後，將收到的 M 和 $r_1 \oplus r_2 \oplus K_1$ 連同 **Step 1** 產生的隨機亂數 r_1，一起傳送給後端伺服器。

Step 4-1: 此步驟的目的為找出該標籤的 Index 以便下一步驟的認證。後端伺服器收到後，利用 ß 和收到的 M 進行下列的運算處理後，找出此標籤的 Index:

i. 取出 M 的第一個部分 $r_2 \oplus X$ 和 M 的第二個部分 $r_2 \oplus Y$，將 $r_2 \oplus X$ 與 $r_2 \oplus Y$ 做 XOR 運算取得 $X \oplus Y$，即 $(r_2 \oplus X) \oplus (r_2 \oplus Y) = X \oplus Y$。
ii. 計算 $X \oplus Y \oplus ß$，即取得 $X \oplus Y \oplus ß$。
iii. 取出 M 的第三個部分 $r_2 \oplus X \oplus Y \oplus ß$，將其與 ii.取得的 $X \oplus Y \oplus ß$ 做 XOR 運算取得 r_2，即 $(r_2 \oplus X \oplus Y \oplus ß) \oplus X \oplus Y \oplus ß = r_2$。
iv. 利用 iii.取得的 r_2，分別與 M 的第一個部分 $r_2 \oplus X$ 和 M 的第二個部分 $r_2 \oplus Y$ 做 XOR 運算取得 X 和 Y，即 $r_2 \oplus (r_2 \oplus X) = X$、$r_2 \oplus (r_2 \oplus Y) = Y$。
v. 利用 iv.取得的 X、Y 和 ii.取得的 $X \oplus Y \oplus ß$ 作為索引到後端伺服器亂數表中取得 $[X]$、$[Y]$ 和 $[X \oplus Y \oplus ß]$。
vi. 取出 M 的第四個部分 $r_2 \oplus Index \oplus [X] \oplus [Y] \oplus [X \oplus Y \oplus ß]$，將其與 iii.取得的 r_2 和 v.取得的 $[X]$、$[Y]$ 和 $[X \oplus Y \oplus ß]$ 做 XOR 運算取得 Index，即 $(r_2 \oplus Index \oplus [X] \oplus [Y] \oplus [X \oplus Y \oplus ß]) \oplus r_2 \oplus [X] \oplus [Y] \oplus [X \oplus Y \oplus ß] = Index$。

Step 4-2: 此步驟的目的為 Reader-to-Tag 認證。後端伺服器利用 **Step 4-1** 取得的 Index 作為索引到標籤資料庫中分別取出該索引所指標籤的 key K 上半部 K_1 和該索引所指標籤舊的 key K_{old} 上半部 K_{1old}，取得後，計算 $r_1 \oplus r_2 \oplus K_1$ 和 $r_1 \oplus r_2 \oplus K_{1old}$，並將此兩個計算結果分別與收到的 $r_1 \oplus r_2 \oplus K_1$ 進行比較，結果分成三種情況:

i. 皆不相等: 此種情況發生時，代表該標籤為非法標籤或者遭受 RFID 攻擊，故終止此次通訊。

ii. 計算的 $r_1 \oplus r_2 \oplus K_1$ 與收到的 $r_1 \oplus r_2 \oplus K_1$ 相同: 此種情況發生時，代表標籤的資料與後端伺服器標籤資料庫中的資料一致，即上一次認證結束後，後端伺服器和標籤皆有完成 key 更新的動作。此時代表該標籤通過 Reader-to-Tag 認證，為合法標籤。接下來要進行更新 key 的動作，首先，後端伺服器自標籤資料庫中取出該標籤的 key K 上半部 K_1 和下半部 K_2，取出後，利用 h( ) 計算 $K_1' = h(K_1)$、$K_2' = h(K_2)$，並且更新該標籤的 key，更新方式為:

   a. $K_{1old} = K_1$、$K_{2old} = K_2$。

   b. $K_1 = K_1'$、$K_2 = K_2'$。

   注意: 為了方便表示起見，在之後步驟中出現的 K_2 為未更新前的內容，即更新後的 K_{2old}。

iii. 計算的 $r_1 \oplus r_2 \oplus K_{1old}$ 與收到的 $r_1 \oplus r_2 \oplus K_1$ 相同: 此種情況發生時，代表標籤的資料與後端伺服器標籤資料庫中的資料不一致，即上一次認證時，可能遭受 Desynchronization attack 或者其他因素，使得後端伺服器有做 key 的更新動作，但標籤沒有做 key 的更新動作。此種情況仍然代表該標籤通過 Reader-to-Tag 認證，為合法標籤。由於目前後端伺服器標籤資料庫所存放該標籤的 key 是更新過的，而標籤本身存放的 key 是未更新過的，故後端伺服器不需要再做更新 key 的動作。

Step 4-3: 此步驟的目的為後端伺服器產生要傳送給標籤的驗證訊息和 key 更新訊息。後端伺服器此時會產生一個隨機亂數 r_3，計算 $r_2 \oplus r_3$、$r_3 \oplus K_2$、$r_3 \oplus K_1'$和 $r_3 \oplus K_2'$，最後將 $r_2 \oplus r_3$、$r_3 \oplus K_2$、$r_3 \oplus K_1'$和 $r_3 \oplus K_2'$一起傳送給讀取器。

Step 5-1: 讀取器收到後，直接將收到的 $r_2 \oplus r_3$、$r_3 \oplus K_2$、$r_3 \oplus K_1'$和 $r_3 \oplus K_2'$傳送給標籤。

Step 5-2: 此步驟的目的為 Tag-to-Reader 認證。標籤收到後，利用自己在 **Step 2** 所產生的 r_2 與收到的 $r_2 \oplus r_3$ 做 XOR 運算取得 r_3，即 $r_2 \oplus (r_2 \oplus r_3) = r_3$，再利用取得的 r_3 與收到的 $r_3 \oplus K_2$ 做 XOR 運算取得 K_2，即 $r_3 \oplus (r_3 \oplus K_2) = K_2$，將計算得出的 K_2 與自己的 key K 下半部 K_2 進行比較，若不相同，則代表此為非法讀取器，故終止此次通訊；若相同，則代表該讀取器通過 Tag-to-Reader 認證，為合法讀取器，接下來要進行更新 key 的動作，更新方式為:

i. 利用 r_3 與收到的 $r_3 \oplus K_1$’做 XOR 運算取得 K_1’，即 $r_3 \oplus (r_3 \oplus K_1') = K_1'$。

ii. 利用 r_3 與收到的 $r_3 \oplus K_2$’做 XOR 運算取得 K_2’，即 $r_3 \oplus (r_3 \oplus K_2') = K_2'$。

iii. $K_1 = K_1'$、$K_2 = K_2'$。

第五章 分析與比較

在上一章中，本論文依據效率與安全的考量，提出了兩個不同類型的 RFID 認證協定，分別是 SEMAPv1 和 SEMAPv2，而此兩個 RFID 認證協定各自有各自的優點和缺點，但其目的皆是為了要解決 RFID 的安全問題，同時顧及到後端伺服器與標籤此兩端的效能負擔。在此章節中，將一一分析 SEMAPv1 和 SEMAPv2 在安全性和效能上的表現，並且與第三章相關研究中所提及到的六個 RFID 認證協定在安全性和效能上分別做出比較。首先在 5.1 節將分析 SEMAPv1 和 SEMAPv2 的安全性，並且與第三章相關研究中所提及到的六個 RFID 認證協定在安全性上做出比較，接著在 5.2 節將分析 SEMAPv1 和 SEMAPv2 的效能，並且與第三章相關研究中所提及到的六個 RFID 認證協定在效能上做出比較。

4.1 安全性分析與比較

在此節中，將分析 SEMAPv1 和 SEMAPv2 的安全性，分析的方法為根據 2.3.2 RFID 攻擊種類中所提及的 9 種 RFID 攻擊，一一分析 SEMAPv1 和 SEMAPv2 是否有能力能夠抵抗這些攻擊，使得 RFID 系統能夠得到完善的保護，並且在最後與第三章相關研究中所提及的六個 RFID 認證協定在安全性上做出比較。接下來將在 5.1.1 做出安全性分析，最後在 5.1.2 做出安全性比較。

5.1.1 安全性分析

根據 2.3.2 RFID 攻擊種類，提到了 9 種常見的 RFID 攻擊，並且依據攻擊者是否需要取得標籤實體才能展開攻擊，區分成兩種類型，分別是 Non-physical attack 和 Physical attack，其中 Non-physical attack 代表的是攻擊者不需要取得標籤實體即可展開攻擊的攻擊類型，包含了六種 RFID 攻擊: Eavesdropping、Tag cloning、Tag tracing、Impersonation、Replay attack 和 Desynchronization attack；而 Physical attack 代表的是攻擊者需要取得標籤實體後，才能展開攻擊的攻擊類型，包含了三種 RFID 攻擊: Compromising attack、Forward security 和 Backward security。接下來，將依序分析 SEMAPv1 和 SEMAPv2 是否能夠抵抗此 9 種 RFID 攻擊:

Eavesdropping

因為讀取器與標籤之間的通訊是藉由無線電波的方式進行，故任何人都可以藉由硬體設備來竊聽讀取器與標籤之間的通訊過程，從中擷取資料，因此，RFID 認證協定有沒有能力抵抗這項攻擊的重點在於，即使攻擊者擷取到讀取器與標籤之間的通訊訊息，攻擊者也因為無法理解其中的內容，而無法取得重要的資料。

根據圖 24: SEMAPv1 認證流程，SEMAPv1 的讀取器與標籤之間的通訊訊息內容包括 Step 1: r_1、Step 2: M, $r_1 \oplus r_2 \oplus K_1$ 和 Step 5: $r_2 \oplus r_3$, $r_3 \oplus K_2$，接下來會一一說明攻擊者擷取到這些通訊訊息後，能不能夠從中取得重要資料:

i. r_1: 隨機亂數。非重要資料，攻擊者取得後不會造成資料外洩的問題。
ii. M: $M = r_2 \oplus X \parallel r_2 \oplus Y \parallel r_2 \oplus X \oplus Y \oplus ß \parallel r_2 \oplus Index \oplus [\ X\ ] \oplus [\ Y\ ] \oplus [\ X \oplus Y \oplus ß\ ]$。標籤的重要資料，但四個串接的整數分別都是與 r_2 做完 XOR 運算的結果，如同分別對此四個串接的整數做加密的效果，而 r_2 是由標籤每回合所產生的隨機亂數，只有標籤自己知道，攻擊者即使取得 M，但因為不知道 r_2，故無法對 M 進行解密獲得重要資料，即攻擊者取得後不會造成資料外洩的問題。
iii. $r_1 \oplus r_2 \oplus K_1$：$K_1$ 為標籤的重要資料，但經過和 $r_1 \oplus r_2$ 做完 XOR 運算後，如同經過加密的效果，而 r_2 是由標籤每回合所產生的隨機亂數，只有標籤自己知道。攻擊者即使取得 $r_1 \oplus r_2 \oplus K_1$，雖然 r_1 可以藉由竊聽得知，但因為不知道 r_2，故無法對 $r_1 \oplus r_2 \oplus K_1$ 進行解密取得 K_1，即攻擊者取得後不會造成資料外洩的問題。
iv. $r_2 \oplus r_3$: 隨機亂數。非重要資料，攻擊者取得後不會造成資料外洩的問題。
v. $r_3 \oplus K_2$: K_2 為標籤的重要資料，但經過和 r_3 做完 XOR 運算後，如同經過加密的效果，而 r_3 是由後端伺服器每回合所產生的隨機亂數，只有後端伺服器自己知道。攻擊者即使取得 $r_3 \oplus K_1$，但因為不知道 r_3，故無法對 $r_3 \oplus K_1$ 進行解密取得 K_2，即攻擊者取得後不會造成資料外洩的問題。

經過以上的分析，可以看出 SEMAPv1 即使遭到竊聽也不會有重要資料外洩的問題，故 SEMAPv1 的確是具有抵抗 Eavesdropping 攻擊的能力。

根據圖 26: SEMAPv2 認證流程，SEMAPv2 的讀取器與標籤之間的通訊訊息內容包括 Step 1: r_1、Step 2: M, $r_1 \oplus r_2 \oplus K_1$、Step 5: $r_2 \oplus r_3$, $r_3 \oplus K_1$, $r_3 \oplus K_1'$, $r_3 \oplus K_2'$，接下來會一一說明攻擊者擷取到這些通訊訊息後，能不能夠從中取得重要資料。其中，只有 Step 5 的 $r_3 \oplus K_1'$和 $r_3 \oplus K_2'$為 SEMAPv1 的讀取器與標籤之間所沒有的通訊訊息，其他訊息皆相同，故不再重複分析:

i. $r_3 \oplus K_1'$: K_1'為標籤的重要資料，但經過和 r_3 做完 XOR 運算後，如同經過加密的效果，而 r_3 是由後端伺服器每回合所產生的隨機亂數，只有後端伺服器自己知道。攻擊者即使取得 $r_3 \oplus K_1'$，但因為不知道 r_3，故無法對 $r_3 \oplus K_1'$進行解密取得 K_1'，即攻擊者取得後不會造成資料外洩的問題。
ii. $r_3 \oplus K_2'$: K_2'為標籤的重要資料，但經過和 r_3 做完 XOR 運算後，如同經過加密的效果，而 r_3 是由後端伺服器每回合所產生的隨機亂數，只有後端伺服器自己知道。攻擊者即使取得 $r_3 \oplus K_2'$，但因為不知道 r_3，故無法對 $r_3 \oplus K_2'$進行解密取得 K_2'，即攻擊者取得後不會造成資料外洩的問題。

經過以上的分析，可以看出 SEMAPv2 即使遭到竊聽也不會有重要資料外洩的問題，故 SEMAPv2 的確是具有抵抗 Eavesdropping 攻擊的能力。

Tag cloning

攻擊者若想要偽造合法的標籤，讓此偽造的標籤能夠蒙騙讀取器通過認證，則攻擊者需要標籤在認證過程中所傳送的正確認證資料，而此一資料可以藉由兩種方式取得，第一種是取得標籤實體，再從此標籤實體中分析其所儲存的資料內容為何，但此方法攻擊者需要取得標籤實體，不在討論的範圍內；第二種是竊聽讀取器與標籤的通訊過程，從中取得標籤送出的認證資料。因此，RFID 認證協定有沒有能力抵抗這項攻擊的重點在於，必須做到 Reader-to-Tag 認證，即每個標籤都有自己的身分識別資料，後端伺服器可以藉由檢查此一資料來判斷該標籤是否合法，並且保證該身分識別的資料不會被攻擊者所竊取，如此一來，即可防止攻擊者偽造標籤。

在 SEMAPv1 中，每個標籤向後端伺服器註冊成功後，後端伺服器都會配置一個 32 bits 的 key K 給該標籤，作為識別該標籤身分的資料。在 Step 4 中，後端伺服器會檢查標籤所傳送的 key K 上半部 K_1 是否與標籤資料庫中所存放的資料相符，若相同才代表該標籤通過認證。因此，攻擊者必須取得 RFID 系統中某一個合法標籤的 key K 後，才能偽造該標籤，但是 SEMAPv1 具有抵抗 Eavesdropping 攻擊的能力，即攻擊者無法藉由竊聽讀取器與標籤的通訊過程得知此 key K，故攻擊者無法偽造任何標籤，這也說明了 SEMAPv1 的確是具有抵抗 Tag cloning 攻擊的能力。

在 SEMAPv2 中，每個標籤向後端伺服器註冊完成後，後端伺服器也會配置一個 32 bits 的 key K 給該標籤，作為該標籤的身分識別資料。在 Step 4 中，後端伺服器會檢查標籤所傳送的 key K 上半部 K_1 是否與標籤資料庫中所存放的資料相符，若相同才代表該標籤通過認證，且此 key K 在每一回合認證結束後都會進行更新的動作。因此，攻擊者必須取得 RFID 系統中某一個合法標籤的 key K 後，才能偽造該標籤，但是 SEMAPv2 具有抵抗 Eavesdropping 攻擊的能力，即攻擊者無法藉由竊聽讀取器與標籤的通訊過程得知此 key K，故攻擊者無法偽造任何標籤，這也說明了 SEMAPv2 的確是具有抵抗 Tag cloning 攻擊的能力。

Tag tracing

當讀取器送出 Query 給標籤後，若標籤每次回應的訊息內容皆相同，則攻擊者即可藉由不斷的發出 Query 給要進行追蹤的目標標籤，並且判斷標籤回應的訊息內容是否與之前所收到的相同，若相同則代表目前所追蹤的對象即為目標標籤，達到標籤追蹤的效果。因此，RFID 認證協定有沒有能力抵抗這項攻擊的重點在於，標籤每一次回應讀取器 Query 的訊息內容是否相同。

根據圖 24: SEMAPv1 認證流程，在 SEMAPv1 中，標籤回應讀取器 Query 的訊息內容為 Step 2: M, $r_1 \oplus r_2 \oplus K_1$，接下來將分別說明此訊息內容在每一回合是否相同:

i. M: $M = r_2 \oplus X \,||\, r_2 \oplus Y \,||\, r_2 \oplus X \oplus Y \oplus ß \,||\, r_2 \oplus Index \oplus [\,X\,] \oplus [\,Y\,] \oplus [\,X \oplus Y \oplus ß\,]$。M 由四個整數串接而成，且每一個整數皆是和 r_2 做完 XOR 運算的結果，

而 r_2 是標籤在每一回合認證時，所產生的隨機亂數，每一回合皆不相同，這也意味著每一回合的 M 皆不相同。

ii. $r_1 \oplus r_2 \oplus K_1$: r_1 是讀取器在每一回合認證時，所產生的隨機亂數，每一回合皆不相同，r_2 是標籤在每一回合認證時，所產生的隨機亂數，每一回合也都不相同，因此 $r_1 \oplus r_2 \oplus K_1$ 在每一回合也都不會相同。

經過以上的分析，可以看出 SEMAPv1 中，標籤每次回應讀取器 Query 的訊息內容皆不相同，攻擊者無法對某個特定的標籤進行追蹤，故 SEMAPv1 的確是具有抵抗 Tag tracing 攻擊的能力。

根據圖 26: SEMAPv2 認證流程，在 SEMAPv2 中，標籤回應讀取器 Query 的訊息內容為 Step 2: M, $r_1 \oplus r_2 \oplus K_1$，與 SEMAPv1 中，標籤回應讀取器 Query 的訊息內容相同，故不再重複分析。因此，SEMAPv2 中，標籤每次回應讀取器 Query 的訊息內容皆不相同，攻擊者無法對某個特定的標籤進行追蹤，故 SEMAPv2 的確是具有抵抗 Tag tracing 攻擊的能力。

Impersonation

攻擊者若想要使用任意一個讀取器，即使該讀取器不是合法的，但是仍然可以讓標籤誤以為它是合法的讀取器，並且允許該讀取器存取標籤的資料，則攻擊者需要讀取器在認證過程中所傳送的正確認證資料，而此一資料可以藉由竊聽讀取器與標籤的通訊過程取得。因此，RFID 認證協定有沒有能力抵抗這項攻擊的重點在於，必須做到 Tag-to-Reader 認證，即讀取器必須傳送正確的認證資料讓標籤相信該讀取器是合法的，並且此認證資料不會被攻擊者所竊取，如此一來，即可防止攻擊者使用非法讀取器存取標籤資料。

在 SEMAPv1 中，標籤在 Step 5 檢查讀取器所送來的認證資料，標籤的 key K 下半部 K_2，是否正確，因為只有合法的讀取器才能從標籤資料庫中取得該標籤的 key K 下半部 K_2。因此，攻擊者必須取得 RFID 系統中某一個合法標籤的 key K 後，才能欺騙該標籤，讓他相信此一非法的讀取器為合法讀取器，但是 SEMAPv1 具有抵抗 Eavesdropping 攻擊的能力，即攻擊者無法藉由竊聽讀取器與標籤的通訊過程得知此 key K，故攻擊者無法使用非法讀取器存取標籤資料，這也說明了 SEMAPv1 的確是具有抵抗 Impersonation 攻擊的能力。

在 SEMAPv2 中，情況與 SEMAPv1 相同，標籤在 Step 5 檢查讀取器所送來的認證資料，標籤的 key K 下半部 K_2，是否正確，因為只有合法的讀取器才能從標籤資料庫中取得該標籤的 key K 下半部 K_2。因此，攻擊者必須取得 RFID 系統中某一個合法標籤的 key K 後，才能欺騙該標籤，讓他相信此一非法的讀取器為合法讀取器，但是 SEMAPv2 具有抵抗 Eavesdropping 攻擊的能力，即攻擊者無法藉由竊聽讀取器與標籤的通訊過程得知此 key K，故攻擊者無法使用非法讀取器存取標籤資料，這也說明了 SEMAPv2 的確是具有抵抗 Impersonation 攻擊的能力。

Replay attack

攻擊者若想要使用 Replay attack 仿冒標籤騙取讀取器或者仿冒讀取器騙取標籤，則必須確保上一回合中，讀取器與標籤之間的通訊訊息，這一回合仍然可以使用。因此，RFID 認證協定有沒有能力抵抗這項攻擊的重點在於，必須做到 Challenge-response 認證，即無論是標籤或讀取器，其所傳送給對方的認證資料，皆必須使用到對方先前所傳送來的訊息內容，如此一來，上一回合的通訊訊息，就無法在此回合中使用。

在 SEMAPv1 中，有兩種遭受 Replay attack 的情況，第一種是攻擊者竊取上一回合中 Step 2 所傳送的訊息，並於這一回合中再次傳送給讀取器，企圖仿冒標籤騙取讀取器而通過認證，第二種是攻擊者竊取上一回合中 Step 5 所傳送的訊息，並於這一回合中再次傳送給標籤，企圖仿冒讀取器騙取標籤而通過認證，以下分別探討此兩種情況:

i. 仿冒標籤騙取讀取器: 標籤在 Step 2 中送給讀取器的訊息內容為 M, $r_1 \oplus r_2 \oplus K_1$，其中用來驗證標籤身分的資料為 $r_1 \oplus r_2 \oplus K_1$，此一認證資料使用到讀取器在該回合 Step 1 所傳送給標籤的 r_1。假設攻擊者竊取了回合 N 中 Step 2 所傳送的訊息，令為 $r_1^N \oplus r_2^N \oplus K_1$，並於回合 N+1 的 Step 2 再次傳送給讀取器，但由於回合 N+1 中，標籤的正確認證資料為 $r_1^{N+1} \oplus r_2^{N+1} \oplus K_1$，故攻擊者所傳送的訊息無法使仿冒的標籤通過認證。

ii. 仿冒讀取器騙取標籤: 讀取器在 Step 5 中送給標籤的訊息內容為 $r_2 \oplus r_3$, $r_3 \oplus K_2$，其中用來驗證讀取器身分的資料為 $r_3 \oplus K_2$，而標籤必須利用 r_2 與 $r_2 \oplus r_3$ 做 XOR 運算才有辦法取得 r_3，即 $r_2 \oplus (r_2 \oplus r_3) = r_3$。假設攻擊者竊取了回合 N 中 Step 5 所傳送的訊息，令為 $r_2^N \oplus r_3^N$, $r_3^N \oplus K_2$，並於回合 N+1 的 Step 5 再次傳送給標籤，但由於回合 N+1 中，讀取器的正確認證資料為 $r_2^{N+1} \oplus r_3^{N+1}$, $r_3^{N+1} \oplus K_2$，故攻擊者所傳送的訊息無法使仿冒的讀取器通過認證。

經過以上的分析，可以看出 SEMAPv1 有做到 Challenge-response 認證，上一回合的認證訊息，下一回合無法繼續使用，故 SEMAPv1 的確是具有抵抗 Replay attack 攻擊的能力。

在 SEMAPv2 中，與 SEMAPv1 類似，有兩種遭受 Replay attack 的情況，第一種是攻擊者竊取上一回合中 Step 2 所傳送的訊息，並於這一回合中再次傳送給讀取器，企圖仿冒標籤騙取讀取器而通過認證，第二種是攻擊者竊取上一回合中 Step 5 所傳送的訊息，並於這一回合中再次傳送給標籤，企圖仿冒讀取器騙取標籤而通過認證，以下分別探討此兩種情況:

i. 仿冒標籤騙取讀取器: 標籤在 Step 2 中送給讀取器的訊息內容為 M, $r_1 \oplus r_2 \oplus K_1$，其中用來驗證讀取器身分的資料為 $r_1 \oplus r_2 \oplus K_1$，與 SEMAPv1 相同，不再重複分析，即攻擊者所傳送的訊息無法使仿冒的標籤通過認證。

ii. 仿冒讀取器騙取標籤: 讀取器在 Step 5 中送給標籤的訊息內容為 $r_2 \oplus r_3$, $r_3 \oplus K_2$, $r_3 \oplus K_1'$, $r_3 \oplus K_2'$，其中用來驗證讀取器身分的資料為 $r_3 \oplus K_2$，與 SEMAPv1 相同，不再重複分析，即攻擊者所傳送的訊息無法使仿冒的讀取器通過認證。

經過以上的分析，可以看出 SEMAPv2 有做到 Challenge-response 認證，上一回合的認證訊息，下一回合無法繼續使用，故 SEMAPv2 的確是具有抵抗 Replay attack 攻擊的能力。

Desynchronization attack

對於每一回合認證結束時，後端伺服器與標籤都必須做資料更新動作的 RFID 認證協定而言，大部份的情況都是後端伺服器會先進行資料更新的動作，而標籤會等待直到收到後端伺服器所送達該回合中最後一個訊息時，才進行資料更新的動作，若此訊息因為遭受 DoS 攻擊或者其他因素而無法送達時，標籤就不會做資料更新的動作，此時就會發生後端伺服器與標籤之間資料不一致的問題，即後端伺服器所存放該標籤的資料為已更新過的，但標籤所存放的資料為尚未更新的，若該 RFID 認證協定沒有提供相關的補救措施時，就會發生該標籤無法再被辨識的報廢情況。因此，RFID 認證協定有沒有能力抵抗這項攻擊的重點在於，若該 RFID 認證協定在每一回合認證結束時，後端伺服器與標籤都必須做資料更新動作，則該 RFID 認證協定必須提供當後端伺服器與標籤發生資料不一致時的補救措施，讓該資料不一致的標籤在下一回合認證時，可以使用舊的認證資料通過認證，不會發生標籤報廢的情況。

對於 SEMAPv1 而言，因為 SEMAPv1 在每一回合認證結束時，後端伺服器與標籤都不需要做資料更新的動作，因此，SEMAPv1 不需要顧慮 Desynchronization attack，即 SEMAPv1 具有抵抗 Desynchronization attack 攻擊的能力。

對於 SEMAPv2 而言，因為 SEMAPv2 在每一回合認證結束時，後端伺服器與標籤都需要做資料更新的動作，因此，SEMAPv2 必須顧慮 Desynchronization attack，當後端伺服器與標籤之間發生資料不一致的情況時，提供相關的補救措施，讓 RFID 系統能夠正常運作。由圖 25: SEMAPv2 初始化設定內容中可以看到後端伺服器的標籤資料庫中多了兩個欄位，分別為 K_{1old} 和 K_{2old}，目的是用來來存放標籤舊的 key K_{old}，即上一回合認證時所使用的 key，如此一來，即使標籤在上一回合認證結束時沒有做更新 key 的動作，在下一回合認證時，後端伺服器仍然可以利用 K_{old} 辨識出此標籤是合法的。另外，在認證流程方面也必須做些判別的動作，即在 SEMAPv2 Step 4-2 中，必須判斷在這一回合當中，標籤是否使用舊的 key 進行認證，若是，則代表遭受 Desynchronizatin attack，使得後端伺服器與標籤之間資料不一致，此時，該回合後端伺服器不能夠做更新 key 的動作，而標籤必須做更新 key 的動作，藉此方法讓後端伺服器與標籤所存放的 key 為同一把，即讓後端伺服器與標籤之間不一致的資料一致。由此看來，雖然 SEMAPv2 必須顧慮 Desynchronizatin attack，但是 SEMAPv2 有提供完整的補救措施，即使後端伺服器與標籤之間發生資料不一致，但 RFID 系統仍然能夠正常運作，且不一致的資料在下一回合認證中會修正為一致，故 SEMAPv2 的確是具有抵抗 Desynchronizatin attack 攻擊的能力。

Compromising attack

當攻擊者取得標籤實體後，可以從該標籤的記憶體中取得該標籤所存放的所有資料，包含用來認證的資料，因此，若標籤存放了其他標籤認證時也會使用到的重要資料，且該資料沒有經過任何加密處理，攻擊者取得後即可使用，則對其他標籤會造成危害。因此 RFID 認證協定有沒有能力抵抗這項攻擊的重點在於，標籤中不能存放所有標籤會共享的認證資料，或者該共享的認證資料必須經過加密處理，使得攻擊者即使拿到也無法使用。

在 SEMAPv1 中，由圖 23: SEMAPv1 初始化設定內容中可以看到標籤所存放的資料為 IDX 和 K，此兩筆資料為標籤自己私有的資料，不會與其他標籤共享，也不會與其他標籤相同，因此，即使攻擊者取得某一個標籤的實體，攻擊者無法從該標籤的記憶體中取得其他標籤認證所需的資料，即不會對其他標籤造成危害。故 SEMAPv1 的確是具有抵抗 Compromising attack 攻擊的能力。

在 SEMAPv2 中，由圖 25: SEMAPv2 初始化設定內容中可以看到標籤所存放的資料為 IDX 和 K，與 SEMAPv1 相同，此兩筆資料為標籤自己私有的資料，不會與其他標籤共享，也不會與其他標籤相同，因此，即使攻擊者取得某一個標籤的實體，攻擊者無法從該標籤的記憶體中取得其他標籤認證所需的資料，即不會對其他標籤造成危害。故 SEMAPv2 的確是具有抵抗 Compromising attack 攻擊的能力。

Forward security

當攻擊者手邊有一份某個賣場的通訊訊息紀錄和一個在該賣場交易過的標籤實體時，攻擊者可以從該標籤的記憶體中取得該標籤所存放的所有資料，再利用這些資料與通訊訊息紀錄做比對，藉此方法分析出該標籤曾經在該賣場中的交易行為，而此舉已侵犯了該標籤持有人的隱私權。因此 RFID 認證協定有沒有能力抵抗這項攻擊的重點在於，標籤所存放的資料與標籤所傳送給讀取器的訊息之間的關係不能過於簡單，避免攻擊者取得標籤實體得知標籤所存放的資料後，利用這些資料與通訊訊息紀錄做比對，分析出該標籤過去的交易行為。

在 SEMAPv1 中，由圖 23: SEMAPv1 初始化設定內容中可以看到標籤所存放的資料為 IDX 和 K，由圖 24: SEMAPv1 認證流程中可以看到標籤傳送給讀取器的訊息為 Step 2: M, $r_1 \oplus r_2 \oplus K_1$。即使攻擊者取得標籤實體，得知 IDX 和 K 後，分成下列兩種情況討論:

i. IDX 與 M 之間的關係:

IDX = X || Y || X⊕Y⊕ß || Index⊕[X]⊕[Y]⊕[X⊕Y⊕ß]

M = $r_2 \oplus$X || $r_2 \oplus$Y || $r_2 \oplus$X⊕Y⊕ß || $r_2 \oplus$Index⊕[X]⊕[Y]⊕[X⊕Y⊕ß]

其中 r_2 是每一回合認證時，標籤產生的隨機亂數，攻擊者無法得知，因此攻擊者無法藉由 IDX 找出相對應的 M，也無法藉由 M 找出相對應的 IDX。

ii. K 與 $r_1 \oplus r_2 \oplus K_1$ 之間的關係: 雖然 r_1 可以藉由竊聽取得，但 r_2 是每一回合認證時，標籤產生的隨機亂數，攻擊者無法得知，因此攻擊者無法藉由 K 找

出相對應的 $r_1 \oplus r_2 \oplus K_1$，也無法藉由 $r_1 \oplus r_2 \oplus K_1$ 找出相對應的 K。

從以上的分析中可以看出攻擊者即使取得標籤實體，得知標籤資料後，也無法利用這些資料與通訊訊息紀錄做比對，分析出該標籤過去的交易行為，故 SEMAPv1 的確是具有抵抗 Forward security 攻擊的能力。

在 SEMAPv2 中，由圖 25: SEMAPv2 初始化設定內容中可以看到標籤所存放的資料為 IDX 和 K，由圖 26: SEMAPv2 認證流程中可以看到標籤傳送給讀取器的訊息為 Step 2: M, $r_1 \oplus r_2 \oplus K_1$，與 SEMAPv1 的情況相同，即攻擊者取得標籤實體，得知 IDX 和 K 後，無法找出 IDX 與 M 之間的關係，也無法找出 K 與 $r_1 \oplus r_2 \oplus K_1$ 之間的關係，因此，攻擊者即使取得標籤實體，得知標籤資料後，也無法利用這些資料與通訊訊息紀錄做比對，分析出該標籤過去的交易行為。另外，SEMAPv2 在每一回合認證結束後，會進行 key K 的更新動作，使得攻擊者更難分析出標籤所存放的 K 與通訊訊息紀錄中的 $r_1 \oplus r_2 \oplus K_1$ 之間的關係。故 SEMAPv2 的確是具有抵抗 Forward security 攻擊的能力。

Backward security

此攻擊與 Forward security 原理類似，差別在於，當攻擊者手邊取得標籤實體，並且從該標籤的記憶體中取出標籤所有的資料後，將來攻擊者可以利用已取得的標籤資料與通訊訊息紀錄做比對，分析出該標籤之後的交易行為，此舉同樣也是侵犯了標籤持有人的隱私權。因此 RFID 認證協定有沒有能力抵抗這項攻擊的重點在於，標籤所存放的資料與標籤所傳送給讀取器的訊息之間的關係不能過於簡單，避免攻擊者取得標籤實體得知標籤所存放的資料後，利用這些資料與通訊訊息紀錄做比對，分析出該標籤之後的交易行為。

在 SEMAPv1 中，由圖 23: SEMAPv1 初始化設定內容中可以看到標籤所存放的資料為 IDX 和 K，由圖 24: SEMAPv1 認證流程中可以看到標籤傳送給讀取器的訊息為 Step 2: M, $r_1 \oplus r_2 \oplus K_1$。即使攻擊者取得標籤實體，得知 IDX 和 K 後，分成下列兩種情況討論:

i. IDX 與 M 之間的關係:

IDX = $X \| Y \| X \oplus Y \oplus ß \| Index \oplus [\ X\ ] \oplus [\ Y\ ] \oplus [\ X \oplus Y \oplus ß\ ]$

M = $r_2 \oplus X \| r_2 \oplus Y \| r_2 \oplus X \oplus Y \oplus ß \| r_2 \oplus Index \oplus [\ X\ ] \oplus [\ Y\ ] \oplus [\ X \oplus Y \oplus ß\ ]$

其中 r_2 是每一回合認證時，標籤產生的隨機亂數，攻擊者無法推算得知，因此攻擊者無法藉由 IDX 找出相對應的 M，也無法藉由 M 找出相對應的 IDX。

ii. K 與 $r_1 \oplus r_2 \oplus K_1$ 之間的關係: 雖然 r_1 可以藉由竊聽取得，但 r_2 是每一回合認證時，標籤產生的隨機亂數，攻擊者無法推算得知，因此攻擊者無法藉由 K 找出相對應的 $r_1 \oplus r_2 \oplus K_1$，也無法藉由 $r_1 \oplus r_2 \oplus K_1$ 找出相對應的 K。

從以上的分析中可以看出攻擊者即使取得標籤實體，得知標籤資料後，也無法利用這些資料與通訊訊息紀錄做比對，分析出該標籤之後的交易行為，故 SEMAPv1 的確是具有抵抗 Backward security 攻擊的能力。

在 SEMAPv2 中，由圖 25: SEMAPv2 初始化設定內容中可以看到標籤所存

放的資料為 IDX 和 K，由圖 26: SEMAPv2 認證流程中可以看到標籤傳送給讀取器的訊息為 Step 2: M, $r_1 \oplus r_2 \oplus K_1$，與 SEMAPv1 的情況相同，即攻擊者取得標籤實體，得知 IDX 和 K 後，無法找出 IDX 與 M 之間的關係，也無法找出 K 與 $r_1 \oplus r_2 \oplus K_1$ 之間的關係，因此，攻擊者即使取得標籤實體，得知標籤資料後，也無法利用這些資料與通訊訊息紀錄做比對，分析出該標籤之後的交易行為。另外，SEMAPv2 在每一回合認證結束後，會進行 key K 的更新動作，使得攻擊者更難分析出標籤所存放的 K 與通訊訊息紀錄中的 $r_1 \oplus r_2 \oplus K_1$ 之間的關係。故 SEMAPv2 的確是具有抵抗 Backard security 攻擊的能力。

5.1.2 安全性比較

本論文在第三章相關研究中提及了六個 RFID 認證協定，而此六個 RFID 認證協定的安全性程度也有所不同，能夠抵抗的攻擊種類也不盡相同。在此將會對本論文所提出的兩個 RFID 認證協定，分別是 SEMAPv1 和 SEMAVPv2，與第三章相關研究中所提及的六個 RFID 認證協定，針對 2.3.2 RFID 攻擊種類介紹的九種 RFID 攻擊，做出安全性的比較。下表 2 為安全性比較的結果。

	O(N) 3.1.1	O(N) 3.1.2	$O(N/2^m)$ 3.2.1	O(1) 3.3.1	O(1) 3.3.2	O(1) 3.3.3	SEMAP v1	SEMAP v2
Eavesdropping	O	O	O	O	O	O	O	O
Tag cloning	O	O	O	O	O	O	O	O
Tag tracing	Δ	O	Δ	O	Δ	Δ	O	O
Impersonation	O	O	O	O	O	O	O	O
Replay attack	O	O	O	O	O	O	O	O
Desynchroniz-ation attack	O	O	O	O	X	O	O	O
Compromising attack	O	O	O	O	O	O	O	O
Forward security	O	O	O	O	O	O	O	O
Backward security	O	O	O	O	O	O	O	O

O: 可以抵抗　　Δ: 無法完全抵抗　　X: 無法抵抗

表 2: RFID 認證協定之安全性比較

由上表 2 中可以看出進行比較的八個 RFID 認證協定，除了 Tag traing 這項攻擊有為數一半的 RFID 認證協定無法完全抵抗和 Desynchronization attack 有一個 RFID 認證協定無法抵抗之外，其他攻擊皆不會對此八個 RFID 認證協定造成傷害。接下來，將提出數個重點來探討表 2 RFID 認證協定之安全性比較的結果:

Tag tracing 難以抵抗的原因

從表 2 中可以看到其中有四個 RFID 認證協定無法完全抵抗 Tag tracing 的攻擊，分別是 3.1.1 Jie Li, 2010、3.2.1 Chiu C. Tan, 2008、3.3.2 FLMAP, Alireza Sadighian, 2008 和 3.3.3 Jianqing Fu, 2010，其原因在於，這些 RFID 認證協定設計用來抵抗 Tag tracing 的方法為在每一回合認證結束後，會對認證資料做更新的動作，使得下一回合認證時，標籤傳送給讀取器的認證資料有所不同，藉此讓攻擊者無法進行追蹤，但此方法卻忽略了攻擊者再進行特定標籤追蹤時，只會使用讀取器對該標籤發出 Query 並且等待標籤的回應，只要當該標籤在被追蹤的過程中沒有再與其他讀取器完成認證，攻擊者就可以順利的追蹤下去，因此，在此種情況下，此四個 RFID 認證協定皆有著 Tag tracing 的風險，而無法完全的抵抗此項攻擊。

另外，3.3.2 FLMAP, Alireza Sadighian, 2008 和 3.3.3 Jianqing Fu, 2010 皆使用了一些小技巧，其目的是要減少後端伺服器用來查詢標籤資料所花費的時間，而這些小技巧雖然降低了後端伺服器查詢標籤資料所花費的時間，但卻帶來了 Tag tracing 的風險，這也是本論文在設計 RFID 認證協定時所要克服的困難之一。

Desynchronization attack 無法抵抗的原因

從表 2 中可以看到 3.3.2 FLMAP, Alireza Sadighian, 2008 無法抵抗 Desynchronization attack 的攻擊，其原因在於該 RFID 認證協定雖然有針對 Desynchronizatio attack，在後端伺服器的標籤資料庫中額外儲存了標籤舊的 key K_{last}，作為補救措施，目的是希望即使後端伺服器與標籤之間發生資料不一致的問題時，標籤仍然可以使用舊的 key 完成認證。但是該 RFID 認證協定在每一回合認證時，無論該回合所使用的 key 是新的還是舊的，在回合結束時，後端伺服器皆會進行 key 更新的動作，而此種處理方式在當連續兩個認證回合，後端伺服器皆有做 key 更新的動作而標籤皆沒有做 key 更新動作時，就會因為標籤所存放的 key K 與後端伺服器所存放的 key K 和 K_{last} 皆不相同，使得該標籤在下一回合認證時，無法順利通過認證，即發生該標籤無法再被辨識的報廢情況，因此，在此種情況下，此 RFID 認證協定有著 Desynchronization attack 的風險，而無法抵抗此項攻擊。

SEMAPv1 與 SEMAPv2 在安全性上的差異

從表 2 中可以看出 SEMAPv1 和 SEMAPv2 皆能抵抗本論文所提及到的 9 種 RFID 攻擊，代表此兩個 RFID 認證協定已具有足夠的安全性，若再深入的探討

SEMAPv1 和 SEMAPv2 的安全性，則 SEMAPv2 的安全性會較 SEMAPv1 來的高，原因在於 SEMAPv2 對於某些 RFID 攻擊有著更高的防禦能力。

SEMAPv2 每一回合皆會對認證資料 key K 做更新的動作，此舉對 Forward security 和 Backward security 攻擊而言，會大幅的增添攻擊者在分析上的難度，使得攻擊者更無從找出該標籤過去或者之後的任何交易行為，因此， SEMAPv2 對此兩種 RFID 攻擊有著較 SEMAPv1 高的防禦能力。另外，對 Tag cloning 而言，若 RFID 系統中發現偽造的標籤，SEMAPv2 可以藉由更新合法標籤 key K 的動作，使得偽造的標籤會因為其所存的 key K 不正確而無法通過認證，即合法的標籤可以繼續使用而偽造的標籤無法再被使用，此結果比起 SEMAPv1 只能同時將合法標籤和偽造標籤一同報廢的下場，SEMAPv2 對 Tag cloning 攻擊有著較高的修復能力。

5.2 效能分析與比較

在上一節中，分析了 SEMAPv1 和 SEMAPv2 的安全性，從分析的結果看來，SEMAPv1 和 SEMAPv2 具有足夠的安全性，能夠抵抗本論文所提及的 9 種 RFID 攻擊，而在此節中，將針對數個影響 RFID 認證協定效能的主因，分析 SEMAPv1 和 SEMAPv2 是否具有快速且有效率的認證能力使得 RFID 系統能夠快速辨識物品的身份，並且在最後與第三章相關研究中所提及的六個 RFID 認證協定在效能上做出比較。接下來將在 5.2.1 做出效能分析，最後在 5.2.2 做出效能比較。

5.2.1 效能分析

在效能分析這部分，我們會分為三個部分個別分析，分別為標籤、後端伺服器和標籤與讀取器之間。在標籤部分，所要探討的重點在於標籤每一回合認證所需的運算量和標籤所需的儲存空間；在後端伺服器部分，所要探討的重點在於每一回合認證所需的運算量、後端伺服器所需的儲存空間和後端伺服器花在查詢標籤資料所需的時間；在標籤與後端伺服器之間的部分，所要探討的重點在於標籤與讀取器每一回合認證所需的通訊次數和每一回合認證所需的通訊傳輸量。以下將針對上述所提的重點，一一做出分析:

標籤-運算量

為了要降低生產成本，標籤的硬體設備往往無法支援較為複雜的運算，如對稱型加密、非對稱型加密、雜湊函數等，因此，大部分的 RFID 認證協定都盡可能的降低標籤在認證過程時所需要的運算量，或者僅使用較簡單且快速的運算，讓標籤即使在硬體的限制下，仍然可以使用這些簡單的運算，安全的完成認證。也就是說，對於評估 RFID 認證協定的效能和可行性而言，標籤的運算量是一大重要因素。

對於 SEMAPv1 而言，由圖 24: SEMAPv1 認證流程中可以看出，標籤只有在 Step 2 和 Step 5 需要運算，以下會詳細計算此兩個步驟所做的運算:

i. Step 2: 首先，標籤利用自己的 IDX 產生 M，此部分的作法為將 IDX 的四個部分分別與該回合所產生的隨機亂數 r_2 做 XOR 運算，最後在串接成 M，故需要 4 個 XOR 運算和產生 1 個隨機亂數；接下來，標籤利用自己的 key K 的上半部 K_1 與收到的 r_1 和 r_2 做 XOR 運算，即 $r_1 \oplus r_2 \oplus K_1$，故需要 2 個 XOR 運算。此步驟總共需要 6 個的 XOR 運算和產生 1 個隨機亂數。

ii. Step 5: 首先，標籤利用該回合產生的隨機亂數 r_2 與收到的 $r_2 \oplus r_3$ 做 XOR 運算取得 r_3，即 $r_2 \oplus (r_2 \oplus r_3) = r_3$，故需要 1 個 XOR 運算；接下來，標籤利用取得的 r_3 與收到的 $r_3 \oplus K_2$ 做 XOR 運算取得 K_2，即 $r_3 \oplus (r_3 \oplus K_2) = K_2$，故需要 1 個 XOR 運算。此步驟共需要 2 個的 XOR 運算。

由上述分析可以看出，SEMAPv1 的標籤在每一回合認證所需的運算量為 8 個 XOR 運算和產生 1 個隨機亂數。

對於 SEMAPv2 而言，由圖 26: SEMAPv2 認證流程中可以看出，標籤也是只有在 Step 2 和 Step 5 需要運算，以下會詳細計算此兩個步驟所做的運算:

i. Step 2: 此部分所做的運算與 SEMAPv1 相同，總共需要 6 個的 XOR 運算和產生 1 個隨機亂數。

ii. Step 5: 首先，標籤利用該回合產生的隨機亂數 r_2 與收到的 $r_2 \oplus r_3$ 做 XOR 運算取得 r_3，即 $r_2 \oplus (r_2 \oplus r_3) = r_3$，故需要 1 個 XOR 運算；接下來，標籤利用取得的 r_3 與收到的 $r_3 \oplus K_2$ 做 XOR 運算取得 K_2，即 $r_3 \oplus (r_3 \oplus K_2) = K_2$，故需要 1 個 XOR 運算；最後是標籤更新 key 的部分，利用 r_3 與收到的 $r_3 \oplus K_1'$ 和 $r_3 \oplus K_2'$ 分別做 XOR 運算取得 K_1' 和 K_2'，即 $r_3 \oplus (r_3 \oplus K_1') = K_1'$ 和 $r_3 \oplus (r_3 \oplus K_2') = K_2'$，故需要 2 個 XOR 運算。此步驟總共需要 4 個的 XOR 運算。

由上述分析可以看出，SEMAPv2 的標籤在每一回合認證所需的運算量為 10 個 XOR 運算和產生 1 個隨機亂數。

標籤-儲存空間

為了要盡量降低標籤的生產成本，標籤用來儲存認證資料的記憶體花費也是考量因素之一，因此，RFID 認證協定會盡可能的降低標籤所需要儲存的認證資料來減少記憶體的使用量。

對於 SEMAPv1 而言，由圖 23: SEMAPv1 初始化設定內容可以看出標籤只需要儲存 IDX 和 K 即可，其中 IDX 為 80 bits，K 為 32 bits，且因為 SEMAPv1 的標籤在每一回合認證結束後，不需要做更新 key K 的動作，故只需要 112 bits 的 ROM。

對於 SEMAPv2 而言，由圖 25: SEMAPv2 初始化設定內容可以看出標籤也只需要儲存 IDX 和 K 即可，其中 IDX 為 80 bits，K 為 32 bits，但因為 SEMAPv2 的標籤在每一回合認證結束後，需要做更新 key K 的動作，故需要 80 bits 的 ROM

和 32 bits 的 NVRAM。

後端伺服器-運算量

雖然後端伺服器不像標籤具有硬體資源的限制，可以執行各種運算，但就評估 RFID 認證協定的效能而言，後端伺服器的運算量也可以作為一種考量因素。

對於 SEMAPv1 而言，由圖 24: SEMAPv1 認證流程中可以看出，後端伺服器只有在 Step 4 需要運算，分別為 Step 4-1 找出標籤的 Index、Step 4-2 Reader-to-Tag 認證和 Step 4-3 產生要傳送給標籤的驗證訊息，以下會詳細計算此三個步驟所做的運算:

i. Step 4-1: 首先，後端伺服器利用收到的 M，將 M 的第一個部分和 M 的第二個部分做 XOR 運算取得 $X \oplus Y$，即 $(r_2 \oplus X) \oplus (r_2 \oplus Y) = X \oplus Y$，故需要 1 個 XOR 運算；接著利用 $X \oplus Y$，計算 $X \oplus Y \oplus ß$，故需要 1 個 XOR 運算；再利用 M 的第三個部分和 $X \oplus Y \oplus ß$ 做 XOR 運算取得 r_2，即 $(r_2 \oplus X \oplus Y \oplus ß) \oplus X \oplus Y \oplus ß = r_2$，故需要 1 個 XOR 運算；再利用 r_2，分別與 M 的第一個部分 $r_2 \oplus X$ 和 M 的第二個部分 $r_2 \oplus Y$ 做 XOR 運算取得 X 和 Y，即 $r_2 \oplus (r_2 \oplus X) = X$、$r_2 \oplus (r_2 \oplus Y) = Y$，故需要 2 個 XOR 運算；最後利用 X、Y 和 $X \oplus Y \oplus ß$ 作為索引到後端伺服器亂數表中取得[X]、[Y]和[$X \oplus Y \oplus ß$]，取出 M 的第四個部分與[X]、[Y]和[$X \oplus Y \oplus ß$]做 XOR 運算取得標籤的 Index，即 $(r_2 \oplus \text{Index} \oplus [X] \oplus [Y] \oplus [X \oplus Y \oplus ß]) \oplus r_2 \oplus [X] \oplus [Y] \oplus [X \oplus Y \oplus ß] = \text{Index}$，故需要 4 個 XOR 運算。此步驟總共需要 9 個 XOR 運算。

ii. Step 4-2: 後端伺服器利用 Index 找出該標籤的 key K 上半部 K_1，與收到的 r_1 和 r_2 做 XOR 運算，即 $r_1 \oplus r_2 \oplus K_1$，並將計算結果與收到的 $r_1 \oplus r_2 \oplus K_1$ 進行比對來驗證標籤是否合法，故需要 2 個 XOR 運算。

iii. Step 4-3: 後端伺服器產生 1 個隨機亂數 r_3，並取出該標籤的 key K 下半部 K_2，計算 $r_2 \oplus r_3$ 和 $r_3 \oplus K_2$，再將計算結果傳送給讀取器，故需要 2 個 XOR 運算和產生 1 個隨機亂數。

由上述分析可以看出，SEMAPv1 的後端伺服器在每一回合認證所需的運算量為 13 個 XOR 運算和產生 1 個隨機亂數。

對於 SEMAPv2 而言，由圖 26: SEMAPv2 認證流程中可以看出，後端伺服器也是只有在 Step 4 需要運算，分別為 Step 4-1 找出標籤的 Index、Step 4-2 Reader-to-Tag 認證與更新標籤 key K 和 Step 4-3 產生要傳送給標籤的驗證訊息與更新訊息，以下會詳細計算此三個步驟所做的運算:

i. Step 4-1: 此步驟所做的運算與 SEMAPv1 相同，總共需要 9 個 XOR 運算。

ii. Step 4-2: 首先，後端伺服器利用 Index 找出該標籤的 key K 上半部 K_1，與收到的 r_1 和 r_2 做 XOR 運算，即 $r_1 \oplus r_2 \oplus K_1$，並將計算結果與收到的 $r_1 \oplus r_2 \oplus K_1$ 進行比對來驗證標籤是否合法，故需要 2 個 XOR 運算；接下來，後端伺服器會進行標籤 key K 的更新動作，作法為取出該標籤的 key K 上半部 K_1 和下半部 K_2，取出後，利用 h( )計算 $K_1' = h(K_1)$、$K_2' = h(K_2)$，並且更新該標

籤的 key，更新方式為: $K_{1old} = K_1$、$K_{2old} = K_2$、$K_1 = K_1'$、$K_2 = K_2'$，故需要 2 個 h( )運算。此步驟總共需要 2 個 XOR 運算和 2 個 h( )運算。

iii. Step 4-3: 後端伺服器產生 1 個隨機亂數 r_3，計算 $r_2 \oplus r_3$、$r_3 \oplus K_2$、$r_3 \oplus K_1'$和 $r_3 \oplus K_2'$， 再將計算結果傳送給讀取器，故需要 4 個 XOR 運算和產生 1 個隨機亂數。

由上述分析可以看出，SEMAPv2 的後端伺服器在每一回合認證所需的運算量為 2 個 h( )運算、15 個 XOR 運算和產生 1 個隨機亂數。

後端伺服器-儲存空間

在此部分所要探討的重點在於，扣除原先後端伺服器用來儲存物品資料的儲存空間外，後端伺服器另外需要用來儲存標籤認證資料的儲存空間花費為何。

對於 SEMAPv1 而言，由圖 23: SEMAPv1 初始化設定內容可以看出後端伺服器需要在標籤資料庫中儲存所有標籤的 key K，並且需要一張存放著 65536 個 32-bits 隨機亂數的亂數表，才能確保 SEMAPv1 可以順利進行。

對於 SEMAPv2 而言，由圖 25: SEMAPv2 初始化設定內容可以看出後端伺服器需要在標籤資料庫中儲存所有標籤的 key K 和舊 key K_{old}，並且需要一張存放著 65536 個 32-bits 隨機亂數的亂數表，才能確保 SEMAPv2 可以順利進行。

後端伺服器-查詢標籤時間

若後端伺服器查詢標籤資料的方式為憑藉著一筆標籤所給定資料到標籤資料庫中查詢，將該筆資料與資料庫中每一筆項目的某個欄位內容做比較，逐一比對是否相同，如果相同則代表找到該標籤的相關資料，此種查詢方式所花費的時間為 O(N)，其中，N 為 RFID 系統的標籤數量。故當 RFID 認證協定應用在標籤數量眾多的 RFID 系統中時，後端伺服器查詢標籤的時間花費可視為 RFID 認證協定效能好壞的評估因素之一。

SEMAPv1 和 SEMAPv2 在 Step 4-1 中，皆會對標籤所送來的 M 做一連串的運算，抽絲剝繭後，取出標籤的 Index，而此 Index 即為該標籤在標籤資料庫中的索引，可以幫助後端伺服器立即找到該標籤的資料，略過逐一比對標籤資料庫項目的動作，而此種查詢方式的時間為 O(1)。

標籤與讀取器-通訊次數

對 SEMAPv1 而言，由圖 24: SEMAPv1 認證流程可以看出每一回合認證，標籤與讀取器所需的通訊次數為 3 次，分別為 Step 1、Step 2 和 Step 5。

對 SEMAPv2 而言，由圖 26: SEMAPv2 認證流程可以看出每一回合認證，標籤與讀取器所需的通訊次數也是 3 次，分別為 Step 1、Step 2 和 Step 5。

標籤與讀取器-通訊傳輸量

對 SEMAPv1 而言，由圖 24: SEMAPv1 認證流程可以看出標籤與讀取器之

間的通訊傳輸量為 Step 1、Step 2 和 Step 5 傳輸資料量的加總，以下分別計算各個步驟的傳輸資料量:

i. Step 1: 傳送的訊息為 r_1，其中 r_1 = 16 bits。
ii. Step 2: 傳送的訊息為 M, $r_1 \oplus r_2 \oplus K_1$，其中 M = 80 bits、$r_1 \oplus r_2 \oplus K_1$ = 16 bits。總共為 96 bits。
iii. Step 5: 傳送的訊息為 $r_2 \oplus r_3$, $r_3 \oplus K_2$，其中 $r_2 \oplus r_3$ = 16 bits、$r_3 \oplus K_2$ = 16 bits。總共為 32 bits。

由上述分析可以看出，每一回合認證，標籤與讀取器之間的通訊傳輸量為 144 bits。

對 SEMAPv2 而言，由圖 26: SEMAPv2 認證流程可以看出標籤與讀取器之間的通訊傳輸量為 Step 1、Step 2 和 Step 5 傳輸資料量的加總，以下分別計算各個步驟的傳輸資料量:

i. Step 1: 傳送的訊息為 r_1，其中 r_1 = 16 bits。
ii. Step 2: 傳送的訊息為 M, $r_1 \oplus r_2 \oplus K_1$，其中 M = 80 bits、$r_1 \oplus r_2 \oplus K_1$ = 16 bits。總共為 96 bits。
iii. Step 5: 傳送的訊息為 $r_2 \oplus r_3$, $r_3 \oplus K_2$, $r_3 \oplus K_1'$, $r_3 \oplus K_2'$，其中 $r_2 \oplus r_3$ = 16 bits、$r_3 \oplus K_2$ = 16 bits、$r_3 \oplus K_1'$ = 16 bits、$r_3 \oplus K_2'$ = 16 bits。總共為 64 bits。

由上述分析可以看出，每一回合認證，標籤與讀取器之間的通訊傳輸量為 176 bits。

5.2.2 效能比較

本論文在第三章相關研究中提及了六個 RFID 認證協定，而此六個 RFID 認證協定的效能也因為認證手法的不同而有所差異。在此將會對本篇論文所提出的兩個 RFID 認證協定，分別是 SEMAPv1 和 SEMAVPv2，與第三章相關研究中所提及的六個 RFID 認證協定，針對標籤運算量、後端伺服器運算量、後端伺服器查詢標籤時間這幾個重點，做出效能的比較。下圖 27 為下表 3 所使用符號的說明，下表 3 為效能比較的結果。

符號	說明
Crypto	Cryptography algorithm
HMAC	Hash-based message authentication code
Hash	Hash function
Bitwise	Bitwise operation
Search	Searching time

圖 27: 表 3 所使用符號之說明

		O(N) 3.1.1	O(N) 3.1.2	$O(N/2^m)$ 3.2.1	O(1) 3.3.1	O(1) 3.3.2	O(1) 3.3.3	SEMAP v1	SEMAP v2
標籤	Crypto	0	0	0	0	0	0	0	0
	HMAC	0	0	0	0	0	0	0	0
	Hash	4	3	2	5	0	4	0	0
	Bitwise	2	0	1	10	12	2	8	10
後端伺服器	Crypto	0	0	0	0	0	2	0	0
	HMAC	0	0	0	2	0	0	0	0
	Hash	4	O(N)	$O(N/2^m)$	6	0	4	0	2
	Bitwise	2	0	$O(N/2^m)$	10	12	2	13	15
	Search	O(N)	O(N)	$O(N/2^m)$	O(1)	O(1)	O(1)	O(1)	O(1)

表 3: RFID 認證協定之效能比較

由上表 3 中可以看出進行比較的八個 RFID 認證協定，標籤和後端伺服器在進行認證時，其所需要的運算種類和運算量皆不相同。接下來，將提出數個重點來探討表 3 RFID 認證協定之效能比較的結果:

標籤運算量

從表 3 中可以看到本論文所提出的 RFID 認證協定，SEMAPv1 和 SEMAPv2，在標籤方面只需要使用到 Bitwise operation，相較於表 3 中其他的 RFID 認證協定，SEMAPv1 和 SEMAPv2 更為符合標籤硬體資源上的限制，僅使用簡單且快速的運算就可以讓標籤安全的完成認證。

後端伺服器查詢標籤時間

從表 3 中可以看到本論文所提出的 RFID 認證協定，SEMAPv1 和 SEMAPv2，在後端伺服器查詢標籤時間方面為 O(1)，即後端伺服器查詢標籤資料所花費的時間不會因為 RFID 系統中的標籤數量增加而上升，更加適用於標籤數量眾多的 RFID 系統環境，更重要的是 SEMAPv1 和 SEMAPv2 不會因為使用 IDX 技巧減少後端伺服器查詢標籤資料時間而引發安全上的問題。

SEMAPv1 與 SEMAPv2 在效能上的差異

從表 3 中可以看出 SEMAPv2 的運算量，無論是在後端伺服器還是標籤方面，都較 SEMAPv1 高，其中的原因是因為 SEMAPv2 在每一回合認證結束後，後端伺服器和標籤都必須做資料更新的動作，但 SEMAPv1 不需要此一資料更新的動作，故 SEMAPv2 的運算量會較 SEMAPv1 高。

第六章 結論

經由第五章在安全性與效能的分析後，證實本論文所提出的兩個 RFID 認證協定，SEMAPv1 和 SEMAPv2，皆具有足夠的安全性和有效率的認證能力，能夠抵抗 RFID 攻擊並且適用於標籤數量眾多的 RFID 系統環境中。在本章節中，將先在 6.1 節討論本論文的研究成果，接著在 6.2 節探討未來的研究方向。

6.1 結論與討論

為了改善 RFID 安全問題，本論文提出了兩個 RFID 認證協定，分別是 SEMAPv1 和 SEMAPv2，其中，SEMAPv2 在每一回合認證結束後，後端伺服器與標籤都必須做資料更新的動作，而此一動作使得 SEMAPv2 的安全程度較為 SEMAPv1 高，但其所付出的運算負擔則讓 SEMAPv1 的效能較 SEMAPv2 高。

SEMAPv1 和 SEMAPv2 的安全基礎皆是建立在後端伺服器使用一張儲存著亂數的表，利用表中的亂數和標籤所傳送的訊息做 XOR 運算，藉此達到如同對通訊訊息做加密般的效果，即使攻擊者竊聽取得通訊訊息，但因為攻擊者手中沒有後端伺服器的亂數表，故攻擊者無法經由解密取得有實質意義的訊息內容，而此方法經過 5.1.1 安全分析後，證實具有足夠的安全性。在效能方面，考量到標籤硬體資源上的限制，在 SEMAPv1 和 SEMAPv2 的認證過程中，標籤只需要使用 XOR 運算即可完成認證，減少標籤在運算上的負擔，並且顧慮到標籤數量眾多的 RFID 系統環境，使用 IDX 技巧讓後端伺服器可以直接在標籤資料庫中，立即找到該標籤的資料，大幅降低後端伺服器用來查詢標籤資料所花費的時間。

6.2 未來展望

在未來研究的方向，為了要讓本論文所提出的 RFID 認證協定更加具有足夠的安全性保護 RFID 系統，並且能夠更有效率的完成標籤的認證過程，列出以下兩點作為未來繼續研究的重點:

1. **探討更多的 RFID 攻擊**

   本論文在 5.1.1 安全分析中，只針對 9 種 RFID 攻擊進行分析，在未來的研究中，我們可以針對更多不同的 RFID 攻擊進行分析，藉由分析的結果檢驗 SEMAPv1 和 SEMAPv2 的安全性，並且做出改善。

2. **改善 IDX 技巧**

   本論文使用 IDX 技巧，減少後端伺服器查詢標籤資料所花費的時間，但 IDX 總長為 80 bits，使得標籤需要較大的儲存空間存放 IDX，且讀取器與標籤之間的通訊傳輸量也因為要傳送 M 而較大，其中 M 為 IDX 經由計算所產生。因此，在未來的研究中，可以針對 IDX 技巧，提出簡化的方法。

參考文獻

[1] Wen Chen and Wen-Nung Tsai, “RFID privacy protect using blocker tag with anti blocker tag scheme,” NCTU 2009.

[2] An-Sheng Lu and Wen-Nung Tsai, “A study of Blocker Tag Detection Based on RFID Gen2 Protocol,” NCTU 2009.

[3] J. Li, Y. Wang, B. Jiao and Y. Xu, “An Authentication Protocol for Secure and Efficient RFID Communication,” Logistics Systems and Intelligent Management, 2010 International Conference on 9-10 Jan. 2010, pp. 1648 .

[4] X. Chen, Y. Su, H. Xiong, Y. Yao, G. Liu and M. Yue, “An Improved Authentication Approach to Enhance Security and Privacy in RFID System,” Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2010 2nd International Conference on 26-28 Aug. 2010 , pp. 217.

[5] C. C. Tan, B. Sheng and Q. Li, “Secure and Serverless RFID Authentication and Search Protocols,” Wireless Communications, IEEE Transactions on April 2008, pp. 1400.

[6] Y. Liu, “An Efficient RFID Authentication Protocol for Low-cost Tags,” Embedded and Ubiquitous Computing, 2008. EUC '08. IEEE/IFIP International Conference on 17-20 Dec. 2008, pp. 180.

[7] A. Sadighian and R. Jalili, “FLMAP: A Fast Lightweight Mutual Authentication Protocol for RFID Systems,” Networks, 2008. ICON 2008. 16th IEEE International Conference on 12-14 Dec. 2008, pp. 1.

[8] J. Fu, C. Wu, X. Chen, R. Fan and L. Ping, “Scalable Pseudo Random RFID Private Mutual Authentication,” Computer Engineering and Technology (ICCET), 2010 2nd International Conference on 16-18 April 2010, pp. V7-497.

[9] D. Henrici and P. Muller, “Hash-based Enhancement of Location Privacy for Radio-Frequency Identification Devices using Varying Identifiers,” Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on 14-17 March 2004, pp. 149.

[10] T. Dimitriou, “A Lightweight RFID Protocol to protect against Traceability and Cloning attacks,” Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on 05-09 Sept. 2005, pp. 59.

[11] S. A. Weis, S. E. Sarma, R. L. Rivest and D. W. Engels, “Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems,” First International Conference on Security in Pervasive Computing, 2003, pp. 50−59.

[12] D. Molnar and D. Wagner, “Privacy and Security in Library RFID Issues,

Practices, and Architectures," Proc.11th ACM conference on Computer and Communications Security, Washington DC., Oct. 2004, pp. 210–219.

[13] A. Mitrokotsa, M. R. Rieback and A. S. Tanenbaum, "Classification of RFID Attacks," In Proceedings of the 2nd Int'l Workshop on RFID Technol., 2008, pp. 73–86.

[14] EPCglobal, "EPC™ Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz – 960 MHz Version 1.2.0,"

[15] EPCglobal: http://www.epcglobalinc.org

[16] GS1 TAIWAN: http://www.gs1tw.org/