行政院國家科學委員會專題研究計畫　成果報告

自動化標籤(Auto-ID)的相關問題與應用-以電子收費系統為例

計畫類別：個別型計畫
計畫編號：NSC93-2416-H-009-018-
執行期間：93 年 08 月 01 日至 94 年 07 月 31 日
執行單位：國立交通大學資訊管理研究所

計畫主持人：羅濟群

報告類型：精簡報告

處理方式：本計畫可公開查詢

中　華　民　國 94 年 8 月 4 日

行政院國家科學委員會專題研究計畫成果報告

自動化標籤（Auto-ID）的相關問題與應用以電子收費系統為例

The Development of an Auto-ID Based Information System－Using Electronic Toll Collection System as an Example

計畫編號：NSC 93－2416－H－009－018－
執行期限：93 年 8 月 1 日至 94 年 7 月 31 日

主持人：羅濟群　　國立交通大學資訊管理研究所
計畫參與人員：林謂立、鍾尚衛、許堯欽　　國立交通大學資訊管理研究所

中文摘要

自動化標籤(Auto-ID)是MIT 研發出來的一種自動化辨識系統。Auto-ID 是開放式架構，使用低廉成本的識別標籤。然而目前Auto-ID的應用始終侷限於供應鏈管理以及物流管理層面。本研究試圖利用Auto-ID的標準，制定出適合電子收費環境的Auto-ID 架構。然而 Auto-ID 原先設計是使用在靜態環境中，若要使用在高速移動的動態環境，會有碰撞與資訊安全的問題產生。我們也將針對其中可能產生的識別標籤碰撞問題進行解決。

國內高速公路收費系統目前採用人工收費。目前全台灣約有450萬各種車輛，且每年仍成長約45 萬輛。然而，國道一號目前的道路容量已不敷使用，到了例假日更是顯現出擁塞嚴重的情況。觀察高速公路擁塞的主要路段，以收費站附近路段最為明顯頻繁。顯然目前的傳統人工收費方式必須加以改進。而運用Auto-ID來做車輛之識別與收費，將是最可能之解決方案。

在本計畫的研究中，我們將針對車輛識別單元的相關主題，包括資訊安全、碰撞問題取得等進行深入研究，然後採用Auto-ID的標準建立出適用於高速公路的收費架構。並利用數值分析或電腦模擬的方式，驗證這些架構針對高速公路電子收費時所帶來的效能與有效性。

關鍵詞：自動化標籤；電子收費；自動識別；智慧型高速公路

Abstract

Auto-ID is an automatic identification system developed by MIT. It is a technology which is an open Standard, using low cost RFID tags to achieve its goal. Current Auto-ID applications focus on supply chain management and inventory control. This research will use the Auto-ID Standard to design architecture suitable of freeway electronic toll collection systems.

The Auto-ID architecture is originally proposed to work in a Static environment. Using in a high-speed dynamic environment like the freeway, collisions of tags may appear which we will solve it. The freeway capacity is no longer enough to handle increasing traffic each year. The freeway congested that will result in the waste of time and money. From the studies, the most congested areas are around toll-collection Stations. Apparently, current toll collection scheme needs to be improved, and using the Auto-ID technology is a viable solution.

We emphasize on collision problem and Auto-ID tag security issues. Then we make numeric experiments or computer simulations on the architecture designed to conclude our contribution.

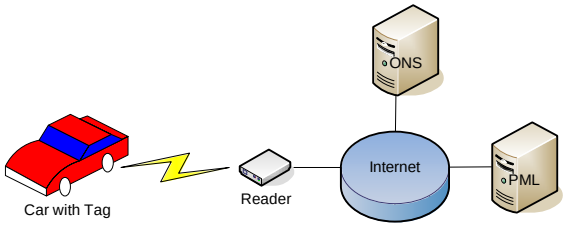
Keywords ： Auto-ID, Electronic Toll Collection, Intelligent Freeway

一、計畫緣由與目的

電子式自動收費(Electronic Toll Collection, ETC)即是將原本收費站採取人工收費的作業過程自動化，車輛不必減速繳費而可快速通過，由於其具有消除收費站附近車輛延滯、節省人工及簡化收費手續等優點，近幾年已經逐漸在歐美各國的收費公路上佈設使用。而自動化標籤(Auto-ID)是一個公開式開放架構，利用低廉成本的識別標籤來取代目前現有的識別機制。

截至目前為止，Auto-ID的應用主要在於以RFID的物件辨識能力加上整體的物件資訊架構應用於供應鏈管理與物流方面。而本篇報告將針對如何應用自動化標籤(Auto-ID)於電子式自動收費，並針對發展高速公路電子收費時所碰到的兩大問題：物件辨識時的資訊安全問題與在高速的情況下辨識車輛所產生的問題進行改進。

圖一為採用Auto-ID架構之電子收費系統示意圖。讀碼器（Reader）與ONS以及PML伺服器之間透過網際網路來連結符合Auto-ID的標準，而讀碼器與應答器之間的通訊為了符合高速公路電子收費系統的需求，必須有所改進。在下段章節會分別對通訊的資訊安全與高速之下所產生的碰撞進行討論。



圖一：高速公路電子收費系統架構圖

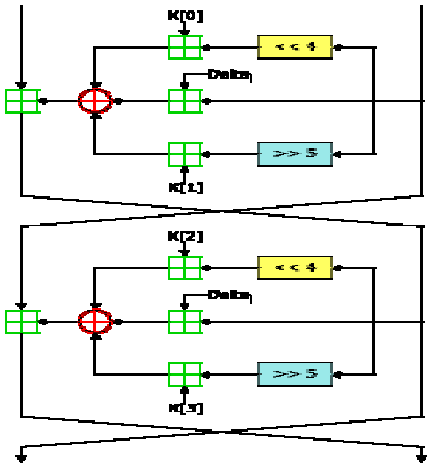
二、文獻探討

2.1 資訊安全

讀碼器（Reader）和應答器（Tag）之間是在無線的環境傳送明文（電子產品碼EPC），因此，使得有心人士可以對此明文EPC作攻擊，使後續作業出現問題。因此，以密文的方式來提高安全性是必須的，但有限的運算能力及記憶體空間，故選擇使用 TEA 作為 Auto-ID 的加解密演算法。

TEA 是由 Cambridge Computer Lab 的 David wheeler 與 Roger Needham 於 1994 年首度發表[4]。 TEA 的特性包括

- 64 Bit 的 block cipher 能力。
- 128Bit 的金鑰
- 具有 Feistel network 的特性。可以一直遞迴加密以增加強度。



圖二：TEA 加密示意圖

而將圖二轉化成數學示。看到 TEA 的每一個回合表示如下

- 明文本身 p 為 64bit 長，均分為二個相等的部份 P1 與 P2。
- 金鑰 K 本身為 128 bit 長，均分為四個相等的部份 k[0],k[1],k[2],k[3]
- Delta[i] 為一任意數，可以任意設置，主要是用來累加作為 key extension 使用，讓每回合所使用的 Key 值會不同。
- 每回合(round)是由二次的循環(cycle)所完成。

2.2 碰撞處理

在 Auto-ID 規格當中，在 UHF 環境下解決碰撞問題的方法就是 Binary-Tree(BT)協定。每次的查詢 Reader 只廣播 0 或 1 給

各 Tag，每個 Tag 得去記憶現在 Reader 所廣播的 ID Index。若 Tag 所收到的 Bit 廣播與其 Index 所指的 Bit 不同，則進入靜止的狀態，亦即在下一輪 Reader 查詢新 Tag ID 前，都不會對 Reader 的廣播有所回應。

當 Tag 所收到的查詢 Bit 與其 Index 所指位置的 Bit 相同，則其回傳 ID 下一個 Bit 給 Reader。在 Auto-ID 規格裡，Tag 傳 0 與 1 是用不同的 Sub-Channel，故 0 與 1 之間不會有碰撞發生，但若很多 Tag 傳 0 或是很多 Tag 傳 1 則會產生碰撞現象。當 Reader 送出相當於 Bit 數的查詢 Bit 後，則一輪即告終止，且有一個 Tag 被辨識出來。

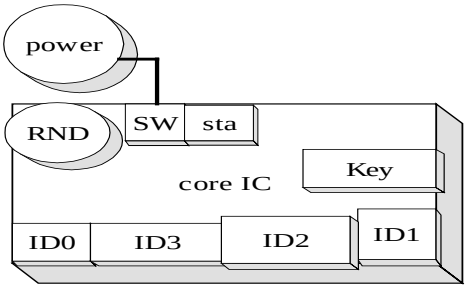
三、研究方法

3.1應答器之設計

本計劃所提出的加密技術，是將 TEA 加密模組晶片加入應答器核心晶片設計。

3.1.1 核心晶片

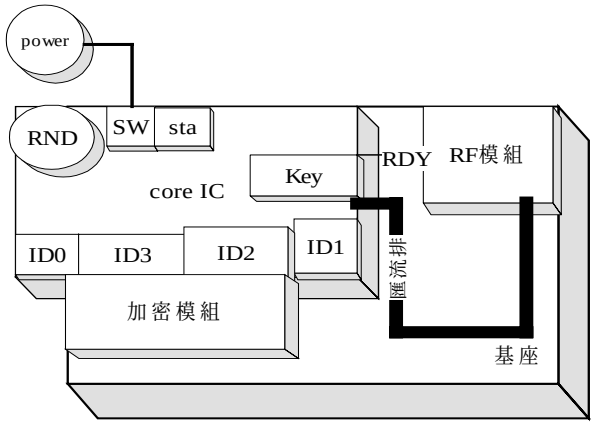
本計劃所提出之系統的核心晶片，設計如下圖三所示。包含了 D0（應答器所定義的亂數，在每次呼叫後會重設）、ID1（一個以該應答器的 EPC 碼為基本所產生的固定值）、ID2（該應答器所含的 ECP 碼）、ID3（本計畫所提出的一個數值，為經加密後的數值，會在每次叫號完成後變動）、RND（亂數產生器）、Sw（外接電源控制開關）、Sta（工作模式暫存器）、Key（加密用的金鑰值）。



圖三：核心 IC 的設計圖

3.1.2 加密電路模組

此為本計劃所提出的特別模組，目地在使用 TEA 演算法加密原有之資料，產生一變動的加密後數值 ID3，在這個模組中，可選擇性備有一外接電源供應線路以預防經由接收讀碼器所產生的電場不足以完成加密電路的運算。此電路開關 Sw 由核心 IC 所控制。在每次完成 ID3 的傳遞後，打開並進行加密的運算。為了密碼強度的設計，TEA 演算法將被會進行 32 個回合來加密。在圖四中，加密模組是直接附著在核心晶片上，並直接將加密後的資料回傳核心 IC。



圖四：整個應答器的組成

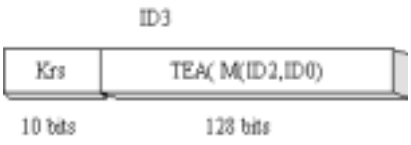
3.1.3 讀碼器的設計

讀碼器負責發送控制命令、依不同的工作模式依 ID 叫號，並接收應答器所回應的資料，與原系統不同的是，本計畫提出的讀碼器多了 Mode X 的工作模式。解碼器依據 ID1 叫號與 ID3 所含的 Key ring serial，找出對應的金鑰並加以解密。

3.1.4 ID3 及 Key 資料結構

- ID3

ID3 是本計畫所提出的系統中獨有的資料。其資料格式如圖五所示：



圖五：ID3 資料格式

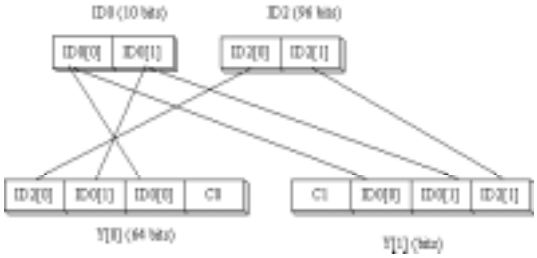
其中，Krs 為 10 bit 的 key ring serial；TEA(M(ID2, ID0))為 ID2 與 ID0 的混合後，經 TEA 演算法使用 Key 值作為金鑰加密 M(ID2,ID0)所產生的 128bit 數值；M(ID2, ID0) 為 ID0 與 ID1 的 128bit 的切割混合函數。

假設輸出值 M(ID2, ID0)=y 為 128bits，且將 y 分為 y[0]y[1]二個 64bits 的值。ID2 的 96 bits 等分為 ID2[0] ID2[1] 二個 48 bits 的值，以及 ID0 10 bit 等分為 ID0[0]ID[1] 二個 5bit 的值 。則

$$Y[0]=ID2[0]||ID0[1]||ID0(0)||C0,$$
$$Y[1]=C1[1]||ID0[0]||ID0[1]||ID2[1]$$

方程式 1：混合方程式 M

其中 C0 為一依據 ID2[0]與 ID0[0]所產生的 6it 的檢查碼。C1 為一依據 ID2[1]與 ID0[1]所產生的 6it 的檢查碼。將這子字串連接起來後，各自形成一個 64 bit 的字串 Y[0]與 Y[1]。而將 Y[0]與 Y[1]連接起來後，就形成了 M(ID2, ID0)。圖六展示了混合函式 M(ID2,ID0)如何將 ID0 與 ID1 加以混合。

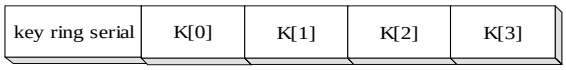


圖六：M(ID2,ID0)

ID3 (138 bits)比 ID2 (96 bits)多了 42 bit . ID3 會成為 128 bit 的原因是因為 TEA 裡每次都是以 64 個 bit 為一個區塊(block)作區塊加密。所以就將原來的 96 個 bit 擴充為 128 個 bit。而混合函式 M 又適當的將亂數 ID0 混合於 ID2 這個固定值，使得 TEA (M (ID2,ID0)) 在每次 ID0 RESET 後都會重新產生。讓每次的叫號後 ID3 值都會不同。

- **key**

Key 值的資料格式如圖七所示為一 128 bit 的值，具有五個部份。Key ring serial 則會依出廠的 tag 批號或是日期的不同而不同。k[0],k[1],k[2],k[3]這四個子金鑰(subkey)為加密使用的金鑰等四分切為各 32bit 的值，目地在提供 TEA 演算法時所需要的 4 個子金鑰值。key 值在每個應答器出廠時就已經燒錄在內。



圖七：Key 值的格式設定

在金鑰的傳遞上是使用金鑰環(Key ring)的觀念。每個 key ring 有一個 key ring serial，而每一 key ring 對應一 1024 欄的 table，裡頭有對應於這 1024 個 ID1 值所使用的金鑰。這些 Key ring 是儲存於讀碼器中，而 key ring 的生成與維護，是由製造商來維護。

當讀碼器完成某一 ID1 的叫號後，將所收集到所有 ID3 值作解析。由於在 ID3 中以明文傳遞了 Key ring serial，讀碼器將比對現有 Key ring serial。如果在讀碼器中有相對應的 key ring serial，則根據叫號的 ID1 值找出相對應的金鑰。如果所收到的 ID3 值所含的 key ring serial 值，並不在讀碼器現存的 key ring serial 清單裡，則讀碼器可以直接，或經由 savant 經由網路取得對應的 key ring。

3.2 安全資料傳輸的運作模式

3.2.1 Mode X 的運作流程

假設一讀碼器與以 Mode X 的方式收集應答器的資料在傳遞的過程中，在圖八中詳細的將 Mode X 的流程進行了描述：

Step1：讀碼機廣播設定運作狀態為 Mode X。在應答器收到此控制指令後。將 IC 中的 Sta 設定為 3 (Sta=3)。

Step2：而讀碼機在廣播工作模式封包後一段時間，就依據某種法則，例如二元

樹，開始叫號(ID1 tree traversal(n)。

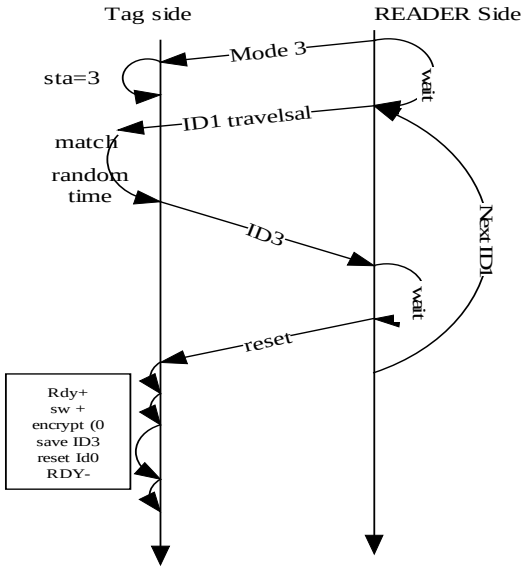
Step3：而應答器在接收到與其 ID1 同號的叫號封包後，延遲一亂數時間後將 ID3 的資料送至讀碼機) (IF ID 1 tree traversal(ID1)= ID1, transmit ID3)；

Step4：讀碼機在收到 ID3 後，將其存入一 Buffer 以供後續的程式進行解密。

Step5：讀碼器完成存入後，對同 ID1 應答器發出 RESET 控制封包(RESET)。

Step6：應答器於收到此重設封包後，重設 ID0，接著開啟 Sw 啟動外接電源(選用)來進行加密電路的演算。產生下一次呼叫的 ID3 值。並將 ID0 與 ID3 存入 IC 的 RAM 區裡。完成以上動作後，關閉 Sw，並重設 RF 單元上的 RDY 腳位 (rdy=1, reset ID0 ,Sw=1 ,generate D3 , save ID0 , ID3, then Sw=0, rdy=0)。

Step7：準備接收下一次的呼叫。而讀碼器在送出 RESET 信號後。就重覆進行下一個 ID1 值的叫號流程。直到所有 ID1 值都叫號完為止。



圖八：Mode X 的通信協定圖

而當讀碼器接收資料後：

Step1：當讀碼器接收了來自應答器的 ID3 資料後，首先經由 ID3 的前 10 個 bit 尋找對應的 key ring。

Step2：並由該對應的 Key table 中，尋找對應 ID1 的金鑰。將 D3 的後 128 位元以所查詢得到的金鑰進行解密。得到 Y = M(ID2,ID0)。

Step3：將 Y 的最前 48bit (ID2[0]) 與最後的 48bit (ID2[1]) 組合就可以得到 ID2。也就是 EPC 碼。

3.1.2 金鑰的傳遞與管理

本系統金鑰的管理是透過有線實體傳遞傳達，在應答器與讀碼器的任何通訊封包中，並未有任何金鑰傳送，所以可以完全防止使用者竊聽得到金鑰。另一個機制是 key ring 的有效時間設計，設定某一 key ring 於何時過期。而 key ring 一旦過期，就算惡意使用者取得了也無效了。

由於 Auto-ID 定義讀碼器或是 SAVANT 必需連接在 TCP/IP 網路上，所以製造商端就必需使用適當的認證程序，以防止未經授權的讀碼機取得 key rings，或是在傳輸的過程中使用 SSL 加密，或是只直接傳送編碼後的 key rings 到讀碼器。

然而，使用者可從應答器上取得所含金鑰，因在同一批號或同一段時間製造的應答器皆使用同一組 key ring。可將應答器作反向工程以取得 Key 值，但反向工程廢時，故當解出金鑰時，該組 key rings 已失效。

3.2 改良的碰撞機制

在 2.2 中，原先 Auto-ID 讀取應答器 ID 資訊的機制是採用 BT 協定法，該協定中讀碼器送出 Bit 0 或 1 來查詢應答器的 ID。若讀碼器要查詢出一個特定的 ID，得至少經過 ID 長度的查詢次數後才能查詢出來。設若有 n 個應答器需要讀碼器查詢，則最少得經過「n * (ID 長度-1)」次的查詢，才能將這 n 個應答器全部辨識出來。若有許多應答器需要讀碼器來做辨識，在一個應答器不會移動的靜態環境底下，最終一定可

以辨識完所有應答器，但是在一個應答器會移動的動態的環境下，可能在讀碼器還未辨識到某一特定的應答器時，該 Tag 已經離開讀碼器的讀取範圍了。所以在此得發展一些方法提升 Auto-ID 的辨識效率。

我們於此提出的二個方法：「基於 CSMA/CA 的機制」與「混合式的 BT 機制」，可以藉由碰撞次數的減少來改善原先辨識機制的效率，使 Auto-ID 更適用於動態的環境。

3.2.1 適用於 Auto-ID 的 CSMA/CA

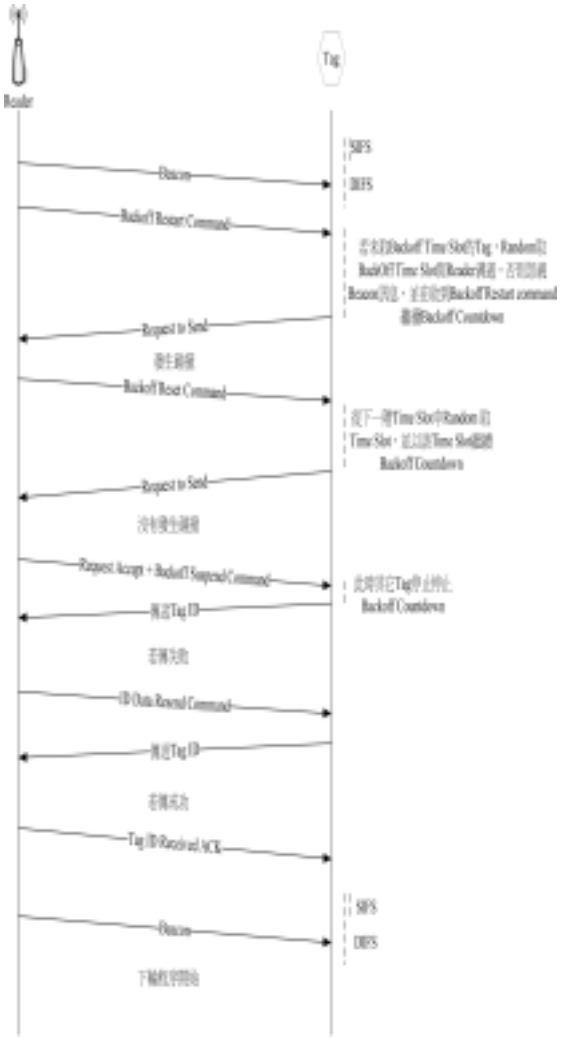
CSMA/CA 原用在 802.11 的環境裡，若要將其應用在 Auto-ID 的環境下，則有下列幾點不能適用的地方：

- Auto-ID 裡各應答器為了低成本的需求，各應答器需被動地聆聽 Reader 所下的指令，以決定接下去的工作，然而在 802.11 裡各工作站卻是得自行聆聽是否有碰撞以及決定何時開始 Backoff。
- Reader 與 Tag 互動最主要的目的在於得到 Tag 96 Bits 的 ID 資訊（一般而言），並沒有其它多餘的資料要傳輸，然而在 802.11 裡各工作站所傳的資料相當大，得將各資料分封處理，並協調需 NAV 的時間。

經過模擬的證實，當應答器數超過 512 個時，會發生辨識有所漏失的現象，也就是有些應答器會永遠無法被讀碼器辨識到。在 802.11 的規格亦建議節點數最好不要超過 512 個，故此機制在同時間於 Reader 讀取範圍內的 Tag 數不大於 512 個時，可以適用；反之，則不適用。整個流程如下圖九：

- 一輪開始時，經過 SIFS（Short Inter Frame Space）的時間，Reader 發送 Beacon Command 給所有 Tag。
- 經過 DIFS（DCF Inter Frame Space）的時間，讀碼器發送 Backoff ReStart Command 給所有應答器。
- 若應答器還沒隨機選過 Backoff Time Slot（其選法依文獻探討裡 CSMA/CA 之 DCF 與延後法則所示），則在接到讀碼器所發送的 Beacon Command 後，開始隨機選取；若已取過，則忽視 Beacon Command。
- 當應答器收到 Backoff ReStart Command 後，開始倒數其所取到的 Time Slot。
- 若 Time Slot 倒數完，則應答器送出 Request to Send 的 Command 給讀碼器。
- 若讀碼器收到一個以上的 Request to Send Command，則表示碰撞發生，此時讀碼器會送出 Backoff Reset Command，要已發送出 Request to Send Command 的應答器再從 2 的下一階次方中隨機取 Time Slot（其選法依文獻探討裡 CSMA/CA 之 DCF 與延後法則所示），做為 Backoff Time Slot。
- 若 Reader 只收到一個 Request to Send Command，表示沒有發生碰撞，Reader 會廣播 Request Accept 加上 Backoff Suspend 的 Command 給應答器。
- 有送出 Request to Send Command 的 Tag 在收到 Request Accept 加上 Backoff Suspend 的 Command 時，開始傳送本身的 ID 資訊，其它未送 Request to Send Command 的應答器，則停止倒數其 Time Slot。
- 若 Tag 成功的傳送本身 ID 資訊給讀碼器，則讀碼器會送出 Received ACK Command 給應答器，否則，則送 ID Data Resend Command 給應答器，要求重送。

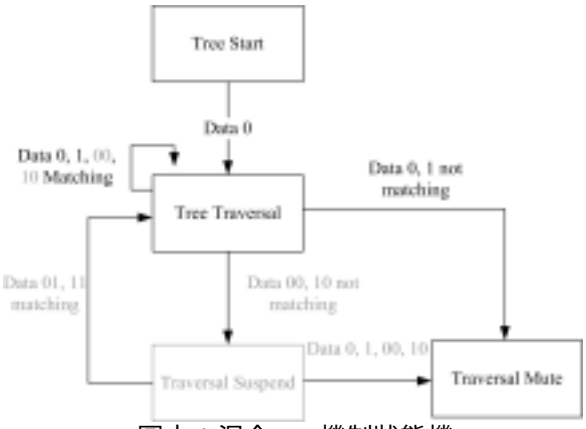
  當讀碼器成功收到應答器所送的 ID 資料後，一輪即結束。



圖九：基於 CSMA/CA 的機制協定圖

3.2.2 混合式 BT 機制

基於 CSMA/CA 的機制並不適用過多 Tag 的狀況，因此當 Tag 數超過 512 個時，得改採用混合式的 BT 機制。雖然經過模擬後的證實，其碰撞次數與媒介總存取次數的減少，並沒有比基於 CSMA/CA 的機制要來得好，可是該機制與原先的 BT 機制相比下，對碰撞次數與媒介總存取次數還是有所改善，且沒有基於 CSMA/CA 機制會發生的辨識漏失問題，亦不必花費任何電子零件的成本。整個流程如下：



圖十：混合 BT 機制狀態機

- 若發生碰撞，則將發生碰撞的 Bit 加上 0,並以此當作 Query Bits 向 Tag 探尋。
- 所有應答器回傳所探尋的位址之下一個 Bit
- 若再發生碰撞, 則再將發生碰撞的 Bit 加上 0,並以此當作 Query Bits 向應答器探尋。
- 在發送 X0(X 為 0 或 1)查詢 Bits 後，若有發生所有 Tag 沒有回應的情況，則傳所發生碰撞的 Bit 加上 1。
- 當應答器接到所發生碰撞的 Bit 加上 1 這樣的查詢 Bits 時，若他處於 Traversal Suspend 狀態，則回到可被 Reader Traversal 的狀態，並回傳下一 Bit 的值給讀碼器，否則即進入 Traversal Mute 狀態。
- 若不再發生碰撞，則程序再依 BT 機制進行，如此反覆。

四、成果報告

4.1 經 TEA 改良後之 Auto-ID

在無線網路的環境下傳輸資料，資料會被有心人士以竊聽、偽裝、重送攻擊、訊息竄改、阻斷服務五種方式加以干擾與破壞。而 Auto-ID 整體架構，在經由 TEA 加解密技術加以保護之後，於原先明文傳輸的 Auto-ID 架構比較，比較如下表所示，本系統所提出的改良式 Auto-ID 架構讓 Auto-ID 在安全上大大的提升。可有效

的解決原有的 auto-ID 在安全上所碰到的問題。

攻擊方式	Auto-ID	改良後 Auto-ID
竊聽	不能防止	可以防止
偽裝	不能防止	可以防止
重送攻擊	不能防止	可以防止
訊息竄改	不能防止	可以防止
阻斷攻擊	部分防止	部分防止

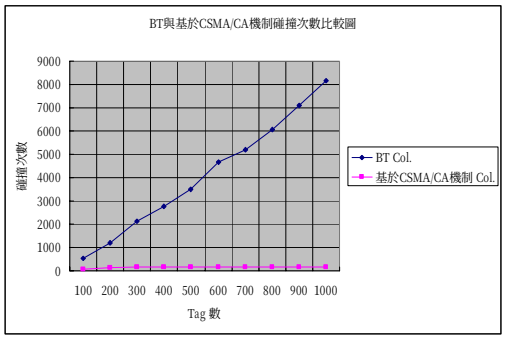
本研究提供了一個改良 Auto-ID 傳輸上安全問題的環境，在此我們針對其原本以明碼傳遞的方式，提出另一個加入以 TEA 為加密演算法的通訊架構，從而定義了該架構使用的資料格式、操作流程。而這個新的架構不但是具有足夠的安全性，更因為它是基於原有架構所提出來的改良。故新的架構仍然與原有的架構相容。

4.2改良過後的碰撞機制處理

在此我們針對基於 CSMA/CA 的機制與 BT 機制做個模擬。下圖是針對基於 CSMA/CA 機制與 BT 機制的模擬比較圖。於圖十一的碰撞比較圖中，可以看到 BT 的碰撞次數會隨著應答器的增加而大幅增加，然而基於 CSMA/CA 機制的碰撞次數並不會隨著應答器的增加而大幅增加，只有小幅的成長。於圖十二的媒介存取次數比較圖中，可以看到基於 CSMA/CA 的機制遠優於 BT 機制，然而在此亦可觀察到一個現象，即基於 CSMA/CA 的機制在 Tag 數大於 500 後，其存取次數的增加趨緩，其最主要的原因在於當應答器數大於 500 時，採用基於 CSMA/CA 的機制有可能會發生應答器沒辨法被辨識到的情況。根據模擬結果，就平均而言，應答器數超過 700，就一定會發生應答器沒辨法辨識到情況。

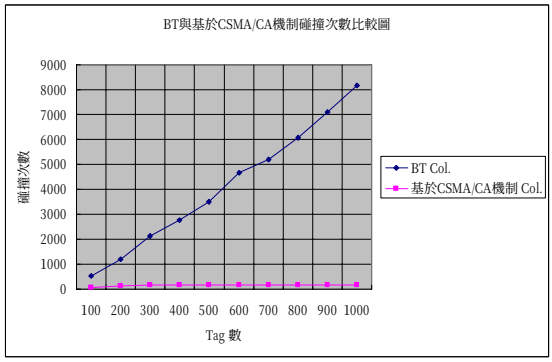
4.2.1CSMA/CA機制的模擬成果與分析

以提升效率的角度來說，從下圖的總存取次數，CSMA/CA 的機制在媒介的總存取次數上的確比原先的好上許多，這也代



圖十一 BT 與基於 CSMA/CA 機制碰撞次數比較圖

表示基於 CSMA/CA 的機制只需少量的總媒介存取數就可辨識應答器，因此在辨識效率上，的確比原先的 BT 機制要來得好。



圖十二 BT 與基於 CSMA/CA 機制存取次數比較圖

4.2.2 混合式 BT 機制的模擬成果與分析

混合式 BT 機制，主要是將 QT 與 BT 兩者結合，去改良 BT 的辨識效能。其與 BT 最大的不同地方在於當發生碰撞時，會將發生碰撞的那個 Bit 加上 0 或 1，而變成跟 QT 一樣的 Query String，去詢問 Tag，然後 Tag 一樣是回下一個 Bit 給 Reader，而非完整的 ID。

以下將針對混合式 BT 與 BT 機制在碰撞次數、存取次數、與 Bit(s)查詢次數上的比較圖，由圖十五可以看出混合式 BT 在碰撞次數上比原先 BT 機制的碰撞次數好上很多，就一般而言，可以減少一半左右的碰撞次數。就混合式 BT 來說因為其將會發生碰撞的 Bit 於後加上 0 或 1，使得原先二個查詢合併成一個查詢，故除了碰撞次數得以減少，亦可讓整個 Bit(s)查詢次數減少，如圖十四，加快應答器被辨識到的時

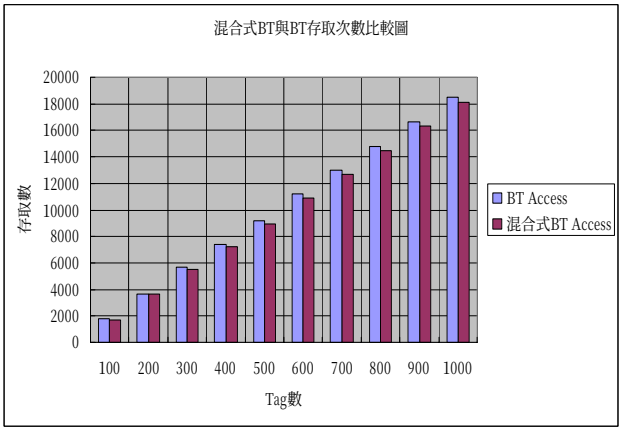
間。在總存取次數上，如圖十三所示，因為當讀取器下 X1（X 為 0 或 1）時，表示之前 X0 的 2 次媒介存取沒有回應, 造成會比原先的 4 次媒介存取數還多 1 的情況，故與原先的 BT 相比改善並不那麼多, 但如前所說，他減少原有 BT 的 Bit(s)查詢次數，與原先查詢次數相比約減少了 1/5，且在應答器的成本上，並不需要額外增加電子零件，故此方法，仍有其可取之處。

為什麼 Bit(s)查詢次數的減少可以改善原有的辨識效率？每要做一次查詢時，除了應答器需要反應時間外，讀碼器亦需因應此次查詢去判斷下一個查詢該是什麼。若能減少查詢次數，除了減少應答器反應的時間外，讀碼器所要做的判斷次數亦會減少，故對辨識上的效率有其改善的功效。

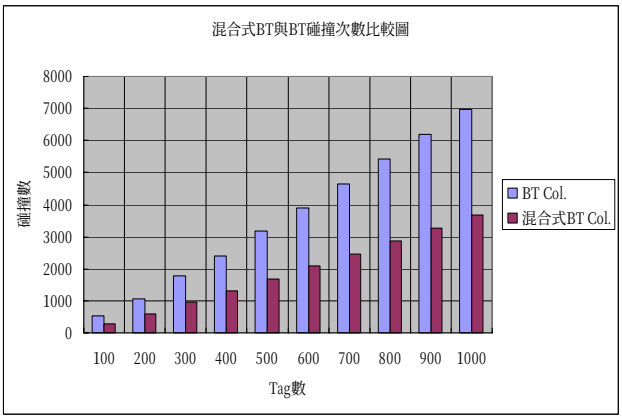
又從下圖可以看得出當應答器數量越多，混合式 BT 在總存取次數與 Bit(s)查詢次數上減少越多，然而在總碰撞次數上仍維持約原先碰撞次數的一半。總存取次數與 Bit(s)查詢次數的減少是因為當應答器越多，碰撞次數增加，混合式 BT 所能改善的碰撞次數亦隨之增加，故減少的總存取次數與 Bit(s)查詢次數會跟著所改善的碰撞次數而增加。

混合式 BT 主要是改善 BT 的碰撞情況，故其它沒有碰撞的情況，其總存取次數與 Bit(s)查詢數是與 BT 相等的。當應答器少時，碰撞亦少，故其能改善的總存取數與 Bit(s)查詢數並不會太多；然而當應答器數量增多時，碰撞發生的機會跟著增加，其能改善的總存取數與 Bit(s)查詢數，亦隨著所改善的碰撞數增加而增加。故改善的總存取數和 Bit(s)查詢數，與改善的碰撞次數息息相關。

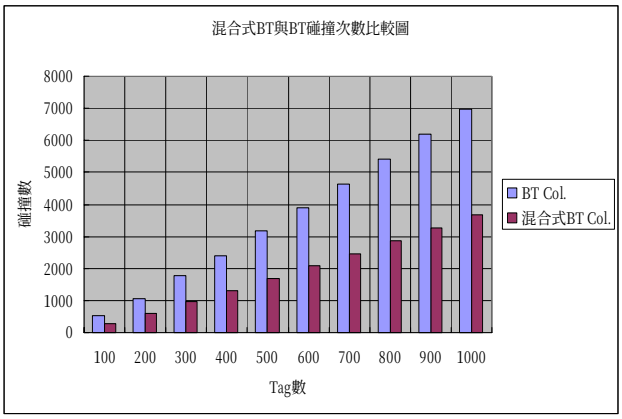
在 Bit(s)查詢次數方面，光是其減少並不代表辨識效能會跟著提高，因若媒介存取次數沒有相對減少的話，辨識效能可能無法有效提升。然而在這經過模擬的結果發現, 混合式 BT 在總媒介存取次數與查詢次數上都比原先的 BT 機制要來的少, 所以可以由此評斷出混合式 BT 在辨識效能上勝於原先的 BT 機制。



圖十三：混合式 BT 與 BT 存取次數比較圖



圖十四：混合式 BT 與 BT Bit(s)查詢數比較圖



圖十五：混合式 BT 與 BT 碰撞次數比較圖

五、計畫成果自評

本計劃之目標為提升國道高速公路之流量，因此導入目前物流界中廣泛採用自動化流程的 Auto-ID，企圖使原本的人工收費流程改成自動化程序，以降低車輛於收費站中停等的時間，進而增加整體道路流量，改善現有國道的效能。然而，原本用於物流的 Auto-ID 架構要能適用於高速移動下的環境，勢必需將原本的架構作部份修改。故在本計劃報告中針對建置自動

化電子收費系統時可能遭遇的兩個問題作為主要研究對象並進行改善，第一是資訊安全問題，讀碼器與應答器在進行資料交換時，由於資料是以明文方式傳送，為了避免有心人士於系統中竊取他人應答器之資訊作為不法用途，故提出符合 Auto-ID 架構的加密機制，防止可能發生的漏洞；第二是因應於高速公路電子收費系統中車輛處於高速移動的狀況下，原本 Auto-ID 在該狀況下會有應答器碰撞的情形，且越高速的狀態下碰撞機率越大，所以針對這樣的環境提出兩種機制降低碰撞次數，其一是修改無線網路的 CSMA/CA 機制並使其符合 Auto-ID 架構；另一種方法則是將原本 Auto-ID 的 Binary Tree 協定加進 Query Tree 協定，使得混合式的 BT 機制即使在大量應答器也能成功減少高速環境下的碰撞次數。

考量目前業界中使用的 Auto-ID 應用在高速公路電子收費系統可能發生上述兩個問題，我們於計劃報告詳加描述、然後提出解決方法，最後以模擬程式得出實驗數據證明成功解決所描述的問題。

六、結論與討論

在本計劃報告中說明現有的 Auto-ID 架構，並將其中應用於高速環境如高速公路電子收費系統可能遭遇到的問題加以解決，提升 Auto-ID 應用於高速公路電子收費系統的可行性，值得未來在建置高速公路電子收費系統時作為參考。

參考文獻

[1] 交通部台灣區國道高速公路局,“高速公路智慧化之整體規劃”，民國九十年二月，台灣，交通部。

[2]D. Brock, “The Electronic Product Code (EPC)”

[3] Oat Systems & MIT Auto-ID Center, “The Object Name Service”

[4]Roger M. Needham and David J. Wheeler., Tea extensions, Technical report, Computer Laboratory, University of Cambridge, October 1997.

[5]何丁武，用於 Auto-ID 環境下減少碰撞的機制，國立交通大學資訊管理研究所碩士論文，民國 93 年

[6]李明橋，以 Auto-ID 為基礎的資訊系統之建置-以電子收費系統為例，國立交通大學資訊管理研究所碩士論文，民國 93 年

[7]陳俊麟，於 Auto-ID 的環境下使用加密傳輸，國立交通大學資訊管理研究所碩士論文，民國 93 年

[8]鄭立群，自動識別應用於供應鏈管理之分析研究，國立交通大學資訊管理研究所碩士論文，民國 93 年