可限制驗證次數之匿名驗證系統

楊尚光
中興大學資訊科學與工程學系
alex19911118@gmail.com

陳國璋
中興大學資訊科學與工程學系
s9756013@csmail.nchu.edu.tw

洪國寶
中興大學資訊科學與工程學系
gbhorng@cs.nchu.edu.tw

簡漢民
中國科技大學數位多媒體設計系
hmchien@cute.edu.tw

摘要

在雲端服務、電子商務情境中，使用者的身分隱私已成為一個重要議題，在一些網路情境下，使用者並不想洩漏真實身分。例如在網站上購買商品，或者是使用 E-Tag 支付高速公路的通行費，都不想被別人、服務供應者識別自己的真實身份，以避免消費內容或行蹤被窺探

本論文參考 Kerberos 協定，提出可以由使用者自行產生權證，同時也限制使用者可以通過驗證次數之匿名驗證。本方法可應用在車載網路，智慧電網，電子商務等。達到使用者負擔輕量化，並兼顧伺服器管理效率。

關鍵詞：匿名驗證，身分驗證，權證，Kerberos 協定。

Abstract

Due to the popularity of cloud services, user privacy has become an important issue. In this paper, we propose an anonymous authentication based on the structure of Kerberos protocol. The scheme can be used in the context such as VANETs, smart grid and E-commerce.

Keywords: Anonymous Authentication, Identity Authentication, Token-Based Authentication, Kerberos Protocol.

1. 前言

隨著雲端運算服務的普及，透過網路可以使用軟體、平台、系統服務等，例如 Amazon EC2、AWS 提供的 IaaS、PaaS 架構，或是 SaaS 如 PC-cillin 的雲端掃毒服務、Spotify 的音樂串流服務。我們可以透過線上付費刷卡的機制，藉以「租用」一些昂貴的設備一段時間，或是在有限的期間，使用軟體、串流的服務，皆是有別於以往軟體銷售採取的「買斷」機制。這樣的模式將會越來越盛行，因為租用較買斷提供更好的售後服務、更好的維護與升級。

然而，直接付費給服務供應商的機制存在一些問題，如付費者「使用者身分隱私」洩漏，另外對於軟體服務使用存在著「計費公平性」問題…即不論使用者掃毒幾次、聽多少音樂，明明消耗的資源不同，卻必須付一樣的錢。這樣人人使用服務吃到飽的機制，勢必造成資源上的浪費，各種可能之原因造成的負面影響如，因為使用者付給 KKBOX 一個月新台幣 300 元，於是使用者的消費者心理，則是不論有沒有真的想要聽音樂都一直播放音樂，始終將串流服務開啟；倘若每位使用者都佔用資源，則會造成伺服器的負擔繁重。這使我們必須採用一個較為公平、合理的計費模式，使用者使用了多少服務，供應商就收多少錢。

而在使用智慧卡付款的機制下，即使是不記名的卡片、電子零錢包，亦無法完全顧及使用者的隱私，在不可連結性的部分，一旦使用者使用同一張卡片兩次以上，商家即可知道這些記錄來自於相同使用者，使得消費者的購物隱私洩漏，當今資料探勘使用氾濫，使用者未明確授權商家蒐集購買紀錄，於是商家可因為商業考量，窺視購物記錄分析每位使用者的消費習慣，對使用者極為不友善。而在一些更為嚴格的匿名驗證，需保證使用者身份不可被揭露之情境，如電子投票[1]。

本論文對於使用者對服務供應商隱藏身份、服務供應商控制使用者通過驗證次數等目的，而設計出一套可行的方法，即可計算驗證次數之匿名驗證。在常見的匿名驗證，通過驗證者得到憑證(Certificate)或是權證(Token)，藉由驗證者(Verifier)替身分合法性背書，而伺服器可以有效地管控權證的有效性、被驗證者的進行

驗證次數；一旦服務伺服器和使用者發生消費糾紛時，本系統可以有條件的揭露使用者的真實身份。

在付費的匿名驗證情境，如車載網路、E-Tag 等情境下，使用者都不希望洩漏自己的真實身份，但使用者必須為送出的訊息負責。中國學者 Xiaoyan Zhu 等人[2]提出可應用在車載隨意網路的隱私保留驗證方法。

另外應用於智慧電網中，電力公司必須對使用者收取電費，可以合理得知用戶用電情形。但是使用者不希望幾點幾分正在用電，其用電的特徵可能洩漏正在使用的家電情況。所以將整個智慧電表分為兩部分，即時用電資訊屬於敏感資料必須匿名，且授權電力公司可以對這樣的資訊進行分析，好作為發電的依據。Efthymiou 等學者針對了這樣的情境提出解套方案[3]，將電表內設置 HFID 和 LFID，分別用來處理不同的資料流，以假名的方法達到匿名是一種簡單的方法。

本論文的組成主要分成四個主要的部份。在第二章，我們對於相關研究做簡要的介紹；第三章則為本論文提出之可限制驗證次數之匿名驗證方法；第四章則針對本論文的安全性進行分析，並和其他方法比較；第五章為本論文之結論。

2. 預備知識與相關研究

本章節對於相關研究以及所需要使用到的預備知識作簡要的介紹。

2.1 非對稱式密碼系統

本論文使用非對稱金鑰系統實作，並可以考慮使用者之不同裝置運算能力，決定使用的密碼系統簽章、加密。因此，在裝置擁有較佳的運算能力時，則採用橢圓曲線密碼系統；相反地，我們假設在物聯網情境下，嵌入式裝置所較適用 RSA 或是 ElGamal 的加密、簽章系統。

故本論文在透過非對稱式金鑰加密、簽章的情境下，採用 ElGamal 的方式產生公私鑰配對，其流程為，傳送者 A 對接收者 B，由 A 先挑選一個大質數p，並在Z_p^*中任取p的一個原根g當作底數，並在Z_{p-1}中任取一數k作為指數。其中，k為 A 的私鑰、y則為對應之公鑰，並公開$\{y, g, p\}$等參數。

$$y = g^k \bmod p$$

對於這種方法的安全性則是建構在分解離散對數是困難的。而在簽章的階段，傳送者 A 任選一個和$p-1$互質的數r，對先前取用的原根g作指數運算後模p，計算出a如下。

$$a = g^r \bmod p$$

並且採用 Yi Mu 等學者提出對於 ElGamal 簽章的修改[1]，計算出s，和a組合成為整個簽章。

$$s = k^{-1}(ma - r) \bmod (p-1)$$

因此當接收者 B 收到 A 所簽署的訊息時，其驗證式如下，當等式成立時，則簽章為有效的。

$$y^a a \bmod p =? \ g^{ma} \bmod p$$

2.2 匿名驗證

關於匿名驗證的實作，由 Lindell 等人所提出的兩個方法[4]。第一個方法，在驗證伺服器存有所有合法使用者的公鑰。驗證伺服器在驗證的時候產生一個訊息M，使用所有合法使用者的金鑰加密此訊息，產生了和可通過驗證使用者數量相同之密文，並將這些密文回送給每一個被驗證者。倘若被驗證者可在這些密文中解開其中一個，表示該使用者具有可通過驗證使用者集合中的私鑰，雖然不知他的真實身分，但是可以相信他是合法使用者的一員。

當合法使用者解開了對應於自己的密文，此使用者仍需以其他合法使用者的公鑰加密自己解開的訊息M_i後，為了避免伺服器藉著傳送不同內容的訊息藉以追蹤使用者身分，因此使用其他合法使用者的公鑰再次加密此訊息，從密文的集合中尋找是否有相同密文，以避免驗證伺服器不誠實。

$$Encrypt_{pk_j}(M_i) =? \ Cipher_j$$

第二個方法，則是可註銷的匿名驗證方法，本情境建立在透過可信第三方為使用者身分合法性背書，但使用者不對提供服務的伺服器洩漏身分，因而採取智慧卡輔助運算，使用具法律效力之智慧卡如自然人憑證…須避免非法使用者變造公鑰，和先前方法類似機制，唯當使用者解開密文、得到明文後，必須再次以其公鑰加密訊息，此時為驗證此使用者使用公鑰是否和憑證管理中心(Certificate Authority，

簡稱 CA)管理的一樣，於是 CA 要求使用者變造M_i中，指定 bit 數量成為M_i'，使用者再加密，回傳密文作驗證。

2.3 Kerberos 協定

Kerberos 協定，是由美國麻省理工學院學者所研發制定，其名源自希臘神話中 Hades 的一隻三頭護衛神犬，協定中包含了三個主要的角色：使用者、驗證伺服器、服務伺服器。[5]

在此架構下的使用者，在使用服務前必須先進行身分的驗證，流程主要包含了六個部分，以下說明詳細驗證流程。

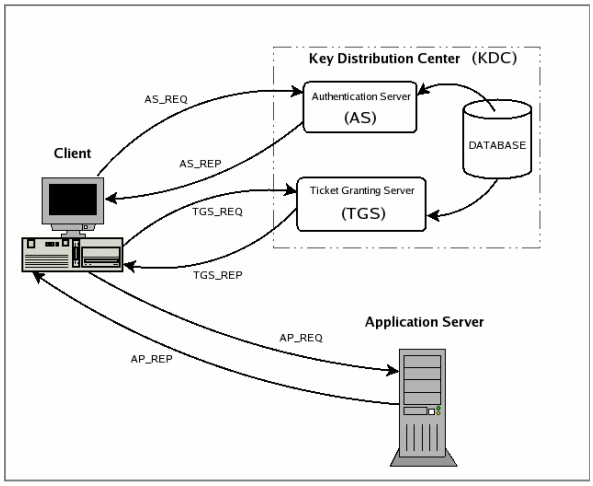

圖 1 Kerberos 架構 [5]

(1) 使用者向驗證伺服器提出身分驗證請求，僅向驗證伺服器送出 ID。

(2) 驗證伺服器查詢該使用者的 ID 是否存在。若存在，則回送兩個訊息：通訊金鑰(Session Key)和票卷伺服器票卷(Ticket Granting Ticket，TGT)。通訊金鑰用以加密使用者和票卷伺服器(Ticket Granting Server)的連線；而 TGT 由 TGS 的金鑰加密而產生，內含 Session Key、使用者 ID、使用者網路位置、TGT 時效等內容；使用者不能自行解開 TGT。

(3) 使用者向 TGS 送出前一階段所獲得的 TGT 和欲使用服務伺服器 ID 及以 Session Key 加密含用戶 ID、時戳之訊息，該訊息稱作認證符(Authenticator)。

(4) 當票卷伺服器(TGS)收到票卷伺服器票卷(TGT)後，先從 KDC 資料庫中確認使用者請求服務伺服器是否存；並解開票卷伺服器票卷(TGT)，因此得到使用者和票卷伺服器連線使用之連線金鑰，TGS 再使用此連線金鑰解開驗證符(Authenticator)，和伺服器票卷(TGT)內的使用者 ID 比較，以避免重送攻擊。完成驗證兩訊息內容後，票卷伺服器回送票卷給使用者(Service Ticket)、使用者和服務伺服器連線金鑰(Client/Server Session Key)。使用者收到透過服務伺服器金鑰所加密之票卷(Ticket)，其內含使用者/服務伺服器的連線金鑰(Client/Server Session Key)、用戶 ID、網路位置、有效期。

(5) 當使用者欲使用服務伺服器資源時，使用者對其送出票卷。

(6) 透過使用者傳來的票卷訊息，決定回覆使用者內容。

3. 可計算驗證次數之匿名驗證

在此章節中，我們將本論文環境設定在類似 Kerberos 架構下，於是我們可以大致將整個方法切割成四個階段：(1)初始階段/權證產生階段(2)身份驗證階段(3)重新產生權證階段(4)身份揭露階段。

在初始階段，由一個公正第三方伺服器負責收費、並分配使用者用以產生權證之「種子變數」，由使用者自行產生權證。身份驗證階段，使用者無須向服務伺服器表明身份但必須證明自己是合法使用者的過程；重新產生權證階段，則是當使用者欲再次付費購買權證的議題；而身份揭露階段，為預留之方法，以解決爭議。

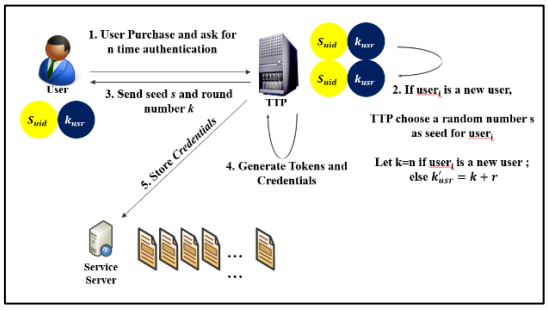

圖 2 本系統架構圖

3.1 初始階段

本階段為初始階段(Initialize Phase)，亦為權證產生階段(Token Generation Phase)。情境為當使用者U_i付費，成功購買了 n 次驗證的資格

後，TTP 為該使用者產生亂數r_{usr}，TTP 對 r_{usr} 簽章成為s_{uid}後傳送給使用者，作為使用者和 TTP 產生權證的變數。

$$s_{uid} = Sign_{TTP-sk}(r_{usr})$$

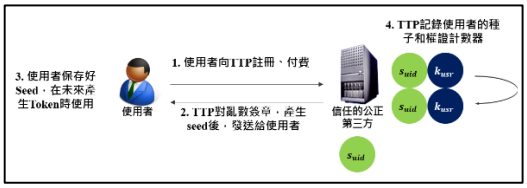

圖 3 初始化階段—產生使用者變數

此時當使用者和 TTP 都具有對應的權證產生種子 s，取其和權證計數連結後，由雜湊函式製作出，如下方運算式所表示，因此使用者和 TTP 都可以自行計算出每一個 Token 的值如下

$$H_{TokGen}(s_{uid} \| round_{usr}) = Token_{usr-round}$$

$$H_{TokGen}(s_{uid} \| 1) = Token_{usr-1}$$
$$H_{TokGen}(s_{uid} \| 2) = Token_{usr-2}$$
$$\cdots$$
$$H_{TokGen}(s_{uid} \| n) = Token_{usr-n}$$

一些較不安全的作法，是直接把這些權證擺放到服務伺服器中。一旦權證外流，任何持有權證者皆可持該權證通過驗證。另外，使用者購買 n 個權證，全部放在 TTP 中，除了耗用空間，亦具有安全性的問題。所以最理想的方法，是由在本架構下負責收費的 TTP，基於安全性考量、節省空間等好處，只儲存使用者的 ID、使用者所購買的權證(Token)數量k、使用者的種子s_{uid}。紀錄 s_{uid} 是因為考量揭露使用者身份的需求；否則，每當使用者購買權證時，應視為不同的使用者，並使用不同的種子變數。

並且避免將產生出來的權證儲存在 TTP，TTP 僅需在有需求的時候可以自行推算出權證。因此，TTP 將每一個被產生出的權證，透過(1)單向雜湊函式，將權證運算成為不可反推出身分、彼此間亦無連結性之憑據(Credential)後，再儲存至服務伺服器中。或者是經由(2)由 TTP 的公鑰加密這些權證，得到僅 TTP 私鑰可解開的憑據後，再儲存到服務伺服器中。

$$H_{CredGen}(Tokens_{usr-round}) = Credential_i$$
or
$$E_{TPP-pk}(Tokens_{usr-round}) = Credential_i$$

$$SEND_{TTP \to SS}(Credential_i)$$

3.2 身份驗證階段

在驗證階段(Authentication Phase)，為某一使用者$user_i$向服務伺服器(Service Server)發出請求，並向該伺服器送出權證（$Token$）；而當服務伺服器收到此權證後，(1)將其做單向雜湊轉換為憑據（$Credential$），並在服務伺服器中找尋是否先前存入權證是否有相同的值，一旦找尋到相同的值，則表示使用者通過驗證，且因透過傳送憑據驗證的手段，是一個簡單且容易限制使用者的驗證次數方法。

$$SEND_{Client \to SS}(Token_j)$$

$$H_{CredGen}(Tokens_j) =? Credential_{stored}$$

當在服務伺服器中，找到相同的憑據，使用者即通過驗證，而服務伺服器(1)必須刪除此驗證過的憑據、此方法不存在揭露身份功能；或是採取(2)較為嚴謹的方法，將當前的時間戳記(timestamp)、此驗證過的憑據記錄下來…往後如果發生爭議時，則可由此紀錄得知。

3.3 重新產生權證階段

當某使用者所持有的權證即將用罄時，繼續採用先前的種子變數產生的權證。所以使用者必須重新向 TTP 付費購買後續 m 次權證，TTP 因為持有該用戶的種子變數s_{usr}，因此可以產生後續權證($round: k+1 \sim k+m$)，權證產生方法如前一小節敘述。唯這次 TTP 無須使用者重新計算種子變數s_{usr}，TTP 僅需更新該使用者的種子計數k，此計數除了是物流座位下一次續購權證時的起始位置計算，也確保某使用者權證之最大數量

$$k' = k + m$$

但是如果某使用者更換 ID 使用本系統，自然是無法避免掉 TTP 為其重新產生種子變數、變數傳送、s_{usr} 及 k_{usr} 額外儲存的成本。

3.4 揭露身分階段

在付費情境下，可能存在消費糾紛，一旦

服務伺服器和使用者發生糾紛時，服務伺服器有權利揭露使用者的真實身份。本系統之揭露身份方法。當某惡意使用者要求服務伺服器資源時，該惡意使用者對服務伺服器送出不當持有之權證(可能是竊取他人權證盜用，或者是變造的權證)。因此在前述初始階段，憑據若是透過 TTP 公鑰加密而產生，則可以在本階段還原回權證，因為 TTP 本身存有所有使用者的種子變數，故在執行加密演算法 $n \times k$ 次的時間內可以找到該權證的擁有者 ID。

$$D_{TPP-sk}(Credential_i) = Tokens_{usr-round}$$

3.5 交換權證階段

本階段之方法用於避免好奇的 TTP 追蹤某個特定使用者的權證是否已經被使用。根據本章所提及之初始階段，任何權證都是可以由 TTP 對每個不同使用者所產生的特定種子產生，所以一個好奇的 TTP 因為具有服務伺服器的資料庫寫入的權限，因此可以去試探該憑證的存留與否。雖然 TTP 知道每個權證是對應到哪個使用者的，但是對服務伺服器來說，因為不知道種子的關係，所以無法區別不同權證之間的來源是否出自同一個使用者。在這樣的條件下，當使用者 A 向使用者 B 提出權證交換的情況下，A 和 B 必須同時向服務伺服器確認對方是否具有合法可通過驗證的權證。一旦服務伺服器確認雙方皆各持有一個合法的權證後，則將存於服務伺服器中的這兩個權證刪除。並由服務伺服器各回送一個「重新取得一次權證」的許可訊息，該訊息中記錄流水號碼，並由服務伺服器簽章後，再以 TTP 公鑰加密之。

$$SEND_{SS \to Client}(Enc_{TTP}(Sign_{SS}(serial)))$$

使用者可以持本訊息，向 TTP 要求一個新的權證。

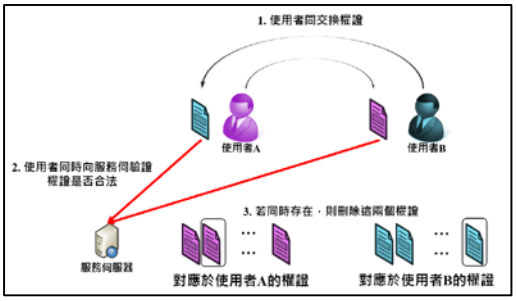

圖 4 使用者交換權證階段

4. 分析

本章節針對安全性和效能兩大層面，與相關研究所提出的方法做比較。在安全性分析的章節，我們考慮相關研究的特點，列舉出可能被攻擊的方法，並提出對於本論文提出方法之安全性證明。而在效能性分析上，也與不同架構之匿名驗證評比。

4.1 安全性分析

在安全性分析的章節，我們首先證明本方法的正確性，即證明本方法所採用的單向雜湊和非對稱式金鑰密碼系統的正確性。匿名驗證須兼顧匿名性及不可連結性。匿名性表示不洩漏真實身份；不可連結性則是攻擊者無法得知相異的連線或身份驗證是來自同一使用者。

(1) 破解雜湊函式：
而單向雜湊的正確性則必須證明不容易「發生碰撞」，證明「不容易」是非常困難的。故先前於 2004 年時，中國學者張小雲發表的文章指出，對於 MD4、MD5、HAVAL-128 及 RIPEMD 的碰撞探討[6]，故在雜湊的挑選上應避免使用一些有疑慮的方法。[7]

(2) 匿名性與不可連結性：
針對匿名性的分析，因使用者將唯一能夠代表身份的變數 s 通過單向雜湊函式運算，所以在該雜湊演算法未被破解之前，本論文所提出的方法所產生的憑據是無法被 TTP 以外之成員逆推回去。
針對不可連結性的分析，因為使用者使用本系統付費後，「批次」地將憑據存入服務伺服器時，伺服器可以直接推測這些憑據是有關連性的，所以在傳送上應避免一次的傳送僅含一個或是少數的使用者憑據；或基於安全考量，累計到一定的數量的憑據再和其他使用者的憑據混和傳送。

(3) 重送攻擊：
本系統所採取權證式匿名驗證系統，在驗證通過後，刪除系統上先前存放的權證雜湊值，所以當使用者持相同的權證再次向伺服器請求驗證時，無法再度通過，因此可抵禦重送攻擊。

(4) 服務伺服器遭入侵：
本系統所提出的架構著重在收費、發

行權證之伺服器本身，故對於採納本驗證機制之服務伺服器本身之安全性無法直接予以補強，因此假設無誤伺服器存在被入侵的風險。因本論文在第三章所描述之初始化階段，TTP 放置於服務伺服器中的權證雖已經被單向雜湊式轉換，故攻擊者無法竊取可以通過驗證的權證。但攻擊者可能取得服務伺服器的寫入權限，因此將偽造的權證植入，導致持有偽造權證者可以通過驗證。所以服務伺服器中所存的權證必須由 TTP 簽章，避免偽造攻擊者植入偽造的簽章。

(5) Semi-Trusted TTP：
就使用者角度而言，若非一次性、不記名的系統，而是在一個需要註冊付費的系統，使用者的身分被記錄在 TTP 端，有可能被揭露身分。而 TTP 被視為是好奇、欲窺探使用者隱私的。使用者若不希望自己的付費資訊、購買服務等行為被記錄、追蹤，必須避免洩漏自己的真實身分給 TTP，如此，在付費的過程中必須避免洩漏自己的信用卡卡號、避免註冊帳號等行為。本論文提出一個新穎的方式以避免自己的真實身分被追蹤，此方法即是透過實作權證交換的機制，即可避免。

4.2 效能分析

在效能分析上，我們將論文所提出的方法，針對「憑證產生」、耗用「儲存資源」、「資料傳送」等方面，和相關研究做效能比較。

(1) Hash 和 Hash Chain 比較：
我們不考慮採用 Hash Chain 的理由有幾個，特別是在效能的方面。在 Kemal Bicakci 等學者所提出的方法[8]，雖然達到了可以限制 Hash 長度可以實現我們用來產生特定數量權證的目的，但是我們的方法更適合套用在物聯網的情境。因為若是我們的系統採用 Hash Chain 產生權證，每個權證都是互相關聯的，也就是一次性的就把所有權證做出來放著，又或者為了產生下一次的權證而必須保留先前的權證。
本方法的最大好處，是不需透過 Hash Chain，只需擁有種子變數以及合法授權之權證編號，即可以產生想要的、特定次序的權證，此外，還可以將本權證送給別人使用。我們不怕別人用這個權證推敲出其他的有效權證。

(2) Kerberos 和 X.509 比較：
由 A. Moralis 等學者提出的文章[9]，可以直接看到，在實務應用層面上比較 Kerberos 和 X.509 系統產生權證。其中，採用 Kerberos 的驗證架構所產生出的權證效率較採用 X.509 的權證高出 28%，應用在網路情境上，以 Kerberos 為優。
另外在 Marvin A. Sirbu 等學者提出的方法[10]，該系統在 Kerberos 的架構下實作分散式驗證，但該系統驗證憑證方式仍需透過 Kerberos-Server 向 Key Distribution Center 取得金鑰。

(6) 變色龍雜湊
與 Song Guo 等學者採用變色龍雜湊達隱私保留的方法[11]，應用在車載網路的架構上，每次取權證都驗證身份(驗證車載裝置的合法性)。這樣的架構，使用者/車載裝置向 CA 洩漏身分、對 RSU 匿名完成一次驗證後，必須一次拿很多個權證回來存，因為權證非由車輛自己產生，故耗用儲存空間且提高通訊成本。

5. 結論

現今的網路幾乎無所不及，幾乎任何裝置及物品皆可透過連上網路的方式被使用者使用。當透過網路取得資源，除了付費問題外，如何保護使用者的身份隱私也是安全議題的重要考量。本論文貢獻，不僅讓使用者不洩漏身分即可存取資源外，最大的貢獻是提出了一套可以控制使用者通過驗證次數的匿名驗證。系統的設計情境，適用於收費環境，保護使用者在網路情境下取用資源時的真實身份。

本方法用於物聯網的使用情境，使用者僅須保存很少的資訊，即可產生很多次可通過驗證的權證，在輕量化的運算方面，也適用於計算能力不高的連網裝置如付費租用腳踏車，以 RFID 搭配動態權證使用[12]，達到對使用者身份隱私更友善的環境。

致謝

本研究接受科技部編號：MOST 104-

2221-E-005-047 研究計畫經費補助。

參考文獻

[1] Yi Mu; Varadharajan, V., "Anonymous secure e-voting over a network," ***Computer Security Applications Conference, 1998. Proceedings. 14th Annual***, pp.293-299, 7-11 Dec 1998

[2] Xiaoyan Zhu; Shunrong Jiang; Liangmin Wang; Hui Li, "Efficient Privacy-Preserving Authentication for Vehicular Ad Hoc Networks," ***Vehicular Technology, IEEE Transactions***, vol.63, no.2, pp.907-919, Feb. 2014

[3] Efthymiou, C.; Kalogridis, G., "Smart Grid Privacy via Anonymization of Smart Metering Data," ***Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference,*** pp.238-243, 4-6 Oct. 2010

[4] Yehuda Lindell, "Anonymous Authentication", ***Journal of Privacy and Confidentiality***, Vol. 2, No. 2, pp. 35-63, 2010.

[5] Fulvio Ricciardi, "Kerberos Protocol", ***MIT Kerberos Consortium - Protocol Tutorial***, http://web.mit.edu/kerberos/

[6] Xiaoyun Wang, Dengguo Feng, Xuejia Lai, Hongbo Yu, "Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD", ***Cryptology ePrint Archive, Report 2004/199***

[7] Valerie Aurora, "Lifetimes of cryptographic hash functions", ***http://valerieaurora.org/hash.html***

[8] Bicakci, K.; Baykal, N., "Infinite length hash chains and their applications," ***Infrastructure for Collaborative Enterprises, 2002. WET ICE 2002. Proceedings. Eleventh IEEE International Workshops***, pp.57-61, 2002

[9] Moralis, A.; Pouli, V.; Grammatikou, M.; Papavassiliou, S.; Maglaris, V., "Performance Comparison of Web Services Security: Kerberos Token Profile Against X.509 Token Profile," ***Networking and Services, 2007. ICNS. Third International Conference on,*** pp.28-28, 19-25 June 2007

[10] Sirbu, M.A.; Chuang, J.C.-I., "Distributed authentication in Kerberos using public key cryptography," **Network and Distributed System Security, 1997. Proceedings., 1997 Symposium,** pp.134-141, 10-11 Feb 1997

[11] Song Guo; Deze Zeng; Yang Xiang, "Chameleon Hashing for Secure and Privacy-Preserving Vehicular Communications," ***Parallel and Distributed Systems, IEEE Transactions***, vol.25, no.11, pp.2794-2803, Nov. 2014

[12] Min Chen; Shigang Chen, "An Efficient Anonymous Authentication Protocol for RFID Systems Using Dynamic Tokens," ***Distributed Computing Systems (ICDCS), 2015 IEEE 35th International Conference***, pp.756-757, June 29 2015-July 2 2015