雲端服務架構下的行動使用者身份認證

陳金鈴　　張耀中
朝陽科技大學資工系　朝陽科技大學資工系
副教授　　研究生
clc@mail.cyut.edu.tw　cyc200@gmail.com

摘要

雲端架構下的公共雲是以推出軟體服務大眾為目的，使用者會以不同的設備登入網頁使用公共雲所提供的服務。雲端服務中的 SaaS（Software as a service）的主要優點是在不同設備之中支援開啟 web 瀏覽器之軟體，透過標準格式就可以使用雲端所提供的服務。雲端的架構是需要受到保護的，否則使用者與內部伺服器溝通伺服器必須承擔很大的風險。

使用者提出身份證明給雲端的過程中，使用者的訊息是需要受到保護的。透過雲端伺服器安全架構與密碼學互相輔助下，才能有效率地保護使用者的資訊安全。本篇論文所提出的協定是使用者以行動裝置的角度思考使用雲端服務，使得記取簡單的使用者身份、密碼就可以安全的證明使用者之身份。

關鍵詞：雲端、認證、行動裝置、安全、攻擊

Abstract

The public cloud of the cloud architecture offers the software service of serving public access. User can login to access the resource via various devices. The main advantage of the SaaS (Software as a service) of the cloud service is to support different software and devices to open web browser through the standard format. However, the user's identity needs to be authenticated; otherwise the communication message will suffer from attacks between user and cloud server.

The process of the users present the identity proof to cloud needs to be protected. Thus, the information security can be guaranteed efficiently. The process of the users present the identity proof to cloud needs to be protected. Thus, the information security can be guaranteed efficiently via the cloud server's secure architecture and the protection of the cryptography mechanism. In this paper, we focus on using the mobile device to access the cloud service via proposing the user's password to identify the user's identity.

1. 介紹

1.1 雲端架構：

使用者透過設備與網路連線，只要將資訊放在雲端上，可以在任何時間、地點隨時隨地使用雲端服務。使用者不需要知道雲端是使用何種架構（如 Grid computing, Distribution computing, Cluster computing…），使用服務只需將需求傳送給雲端，雲端會自動對設備進行最有效的計算與運用。

雲端早期目標是透過網際網路結合很多台電腦作分散式架構的運算，將執行序切割成小的執行序再交由多台電腦分散去運算，將結果傳回來，可以在短時間處理單一台電腦要花費很久的運算量。

雲端的發展漸漸的轉向以服務使用者為導向，使用者的需求透過雲端的特性：無時無刻、運算快速、開發環境簡單…等簡單的步驟就可以使用雲端所提供的服務。

目前雲端服務可區分三類：

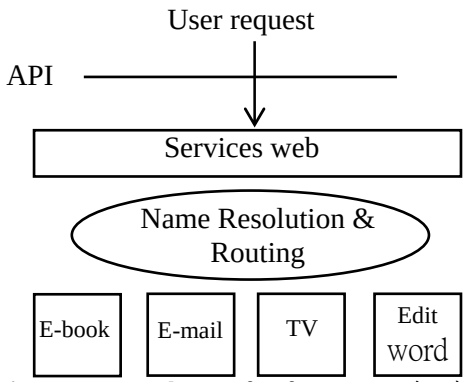
(1) SaaS(Software as a Service)
使用者不需要下載或安裝任何程式，透過瀏覽器使用雲端所提供的服務(例如：Google docs)。

(2) PaaS(Platform as a Service)
提供運算平台，使用者將撰寫好的程式傳送至雲端所提供的平台，該平台會編譯程式與呈現執行結果(例如：Google app Engine)。

(3) IaaS(Infrastructure as a Service)
提供運算、儲存、網路等資源，使用

者對所需要的資源作評估進行租借(例如：Amazon EC2)。

雲端架構上，早期是使用者向雲端所提供的不同服務設備進行溝通，而雲端服務的溝通方式必須加強雲端架構的安全。使用者的要求在最上層服務首頁會將訊息轉交給內部架構作處理，使用者不會直接與雲端內部架構作溝通，可以保護雲端內部架構安全，此種方式稱為 object-to-object[2]而不是早期 host-to-host 架構的觀念，以上所介紹的架構如圖一所示。



圖一、Networking of Information 架構[2]

雲端業者所提供的服務不可能包括使用者所有的需求，所以使用者會使用不同雲端業者所提供的網路服務。在 2007 年，業者所推動的 OpenID[6]驗證規範 2.0 與屬性交換規範 1.0。OpenID 目的是讓使用者方便在不同雲端提供者進行身份認證，使用者只需要註冊一次 OpenID 就可以在其他網頁進行登入身份確認。當然 OpenID 聯盟必須確保使用者的安全性，必須監督雲端業者是否違法，否則使用者的隱私訊息很容易被惡意的攻擊者冒用。

在 2000 年 2 月初，亞馬遜網站遭受到 DDOS 的攻擊造成嚴重的損失，藉由這次的攻擊，使用者與伺服器溝通之前會先由防火牆過濾封包，判斷該封包是否是異常行為。雲端系統的私鑰不會儲存在使用者設備裡，使用者需要透過安全的加密方式（如 PKI、SSL）將使用者的訊息傳送至雲端認證，而伺服器端有驗證表儲存使用者的身分資料藉以確認使用者。

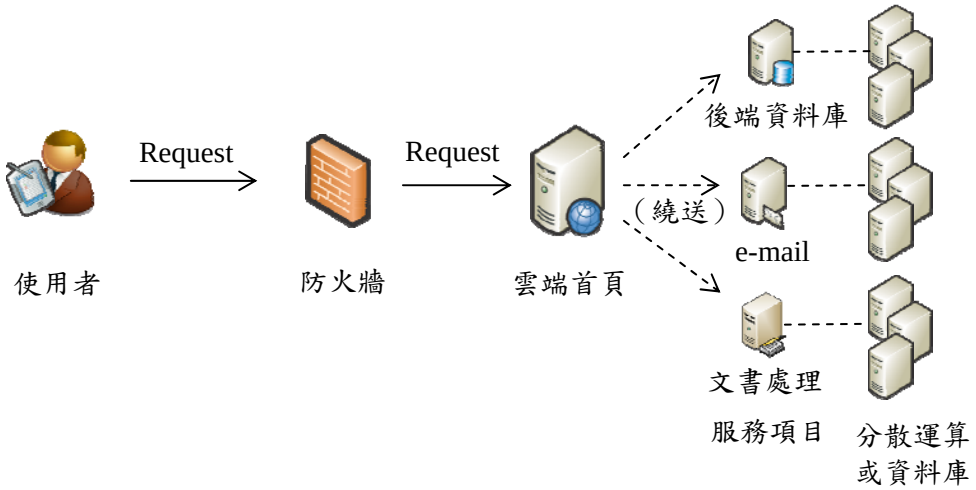
1.2 行動裝置分析：

隨著智慧型手機的興起，行動設備在這幾年有很大的成長空間，可以在接收到電信業者所提供的訊號範圍下使用網路服務。目前行動裝置 PDA 或平板電腦與雲端服務是廠商所主打的溝通平台，強調在外地可以隨時收發 e-mail 與使用網路資源。

使用者會使用不同設備裝置使用雲端服務，PDA 與平板電腦是目前使用無線上網的良好溝通平台。但行動裝置目前也有些硬體上的限制[3]，我們整理了以下常見的限制：

(1) 頻寬的限制
(2) 連線穩定度
(3) 低運算量
(4) 有限的電池容量
(5) 容量不大的儲存功能

以行動使用者的觀點而言，使用雲端所提供的服務之前，必須先向雲端提出使用者的身份證明。不同於智慧卡的是並非每一台設備都可以讀取智慧卡、安全機制都有所不同。再者不同的作業系統對於週邊設備的限制也有所不同（如 iPad）雖然可以瀏覽網頁，但該設備沒有提供一般標準介面（如 USB）提供讀取智



圖二、我們所提出的架構

慧卡功能。

使用者存放訊息至雲端裡，在不同設備進行存取時必須先向雲端提供者證明使用者身份，一般使用者透過電腦使用 SSL 的 PKI 機制與憑證方式安全證明使用者身份，但這是因為使用個人電腦的運算量可以應付這些運算量，頻寬量夠大也夠穩定，但行動裝置具有低運算量與通訊成本昂貴等限制，不適合運算量大的機制。以 RSA 機制而言[4]，若所選取的質數如果長度，則很容易可找出破解方式，在 2002 年 RSA-158 被成功的因數分解，2009 年 RSA-768 也被成功地破解，所以一般認為安全的 RSA 金鑰長度應該要在 1024-2048 bits 或以上。

如果使用者所選取的質數不夠大，使用者所傳送的加密訊息很可能會被破解，所以 RSA 這種運算需求不適用於行動設備。在另一方面由於行動設備運算能力受限，為了保護由行動設備傳送訊息的安全性，因此就必須另外設計考量適合的安全機制。

綜合了以上的說明，為了達到需要的安全等級，我們結合了行動設備與雲端服務。讓使用者在電信業者所提供的訊號範圍下透過行動裝置使用網路資源，安全認證使用者身份與使用雲端所提供的服務。

2. 我們所提出的認證協定

使用者作身份確認的同時，產生與雲端伺服器溝通的對話密鑰，雲端是必須儲存使用者身份與密碼，使用者在傳送訊息作身份確認時是不會曝露密碼。提出身份證明時還必須傳送國際行動設備代碼（International Mobile Equipment Identification, IMEI），使用雲端服務除了會談金鑰可認證使用者身份之外，還必須確認溝通的裝置是否為之前認證過的同一個人所使用。我們所提出的架構如圖二所示。

使用者的訊息會先經過防火牆確認封包是否異常。使用者與雲端伺服器作身份確認，其目的是產生會談金鑰，使用者之後對雲端的請求是透過會談金鑰的保護，雲端伺服器可以確認使用者之身份也確認使用者的行動裝置，使用者不用擔心該訊息到底要如何送達到雲端內部處理，當雲端內部對使用者作回應，會透過同一頁面回應該使用者的需求。以下是我們在本文中所使用到的符號。

符號

$\oplus$：	互斥或閘運算
$\|\|$：	串接運算
ID：	使用者身份代碼
PW：	使用者密碼
$IMEI$:	國際行動設備代碼
N_u, N_s：	使用者與伺服器產生的亂數
SK:	雲端與行動裝置間的會談金鑰
API_{req}：	使用者對雲端提出服務需求
$RESP$：	雲端回應使用者之內容需求
$E_{SK}(m)$：	以 SK 對訊息 m 作對稱式加密
$D_{SK}(m)$：	以 SK 對訊息 m 作對稱式解密
$A \stackrel{?}{=} B$：	比較 A 和 B 是否相等
$h(.)$:	單向雜湊函數

(1) 註冊階段：使用者透過安全通道提出身份代碼 ID 及密碼 pw 向雲端註冊。雲端將使用者所提出的相關訊息儲存在驗證表裡。

(2) 認證階段：圖三為我們所提出的認證協定流程。

步驟 1：使用者向雲端提出身份證明輸入使用者身份代碼 ID 及密碼 pw 並計算：
$C_1 = h(h(pw) \oplus IMEI) \oplus N_u$，
$C_2 = h(ID \| h(pw) \| h(N_u \oplus IMEI))$ ，將 $(ID, IMEI, C_1, C_2)$ 訊息傳送至雲端作身份確認。

步驟 2：雲端首先檢查 ID，再使用 ID 對照驗證表取出相對應密碼 pw'，對 C_1 計算出 $N_u' = C_1 \oplus h(h(pw') \oplus IMEI)$ ，並驗證 N_u' 之正確性
$h(ID \| h(pw') \| h(N_u' \oplus IMEI)) \stackrel{?}{=} C_2$ 。
若驗證成功後產生 N_s，伺服器計算會談金鑰 $SK = h(N_u \| N_s) \oplus IMEI$ 。
接下來傳送 $h(h(pw) \| IMEI \| N_u) \oplus N_s$
及 $h(IMEI \| N_u \| N_s)$ 給使用者。

步驟 3：使用者計算出 N_s'
$N_s' = h(h(pw) \| IMEI \| N_u)$
$\oplus h(h(pw) \| IMEI \| N_u) \oplus N_s$
並檢查 N_s' 之正確性

1.1 $C_1 = h(h(pw) \oplus IMEI) \oplus N_u$
1.2 $C_2 = h(ID \| h(pw) \| h(N_u \oplus IMEI))$
1.3 $(ID, IMEI, C_1, C_2)$ →

2.1 檢查ID
2.2 $N_u' = C_1 \oplus h(h(pw') \oplus IMEI)$
2.3 $h(ID \| h(pw') \| h(N_u' \oplus IMEI)) \stackrel{?}{=} C_2$
2.4 產生N_s
2.5 $SK = h(N_u \| N_s) \oplus IMEI$
2.6 $h(h(pw) \| IMEI \| N_u) \oplus N_s, h(IMEI \| N_u \| N_s)$

3.1 $N_s' = h(h(pw) \| IMEI \| N_u)$
$\oplus h(h(pw) \| IMEI \| N_u) \oplus N_s$
3.2 $h(IMEI \| N_u \| N_s) \stackrel{?}{=} h(IMEI \| N_u \| N_s')$
3.3 $SK = h(N_u \| N_s) \oplus IMEI$

圖三、我們所提出的認證協定流程

$h(IMEI\|N_u\|N_s) \stackrel{?}{=} h(IMEI\|N_u\|N'_s)$。

這時使用者與雲端完成了交互認證之程序，使用者即可和雲端通訊服務訊息，同時使用者也產生了下次溝通之會談金鑰 $SK = h(N_u\|N_s) \oplus IMEI$ 。

(3) 服務需求階段：圖四為使用者對雲端提出服務需求流程。

步驟 1：使用者選擇雲端服務 API 並向雲端提出要求回應，利用先前驗證使用者身份產生的 SK 作對稱式加密保護訊息 $C_3 = E_{SK}(\text{API}_{req})$，再產生 N_{u+1}，並計算：

$C_4 = h(h(pw) \oplus IMEI) \oplus N_{u+1}$

$C_5 = h(ID\|h(pw)\|h(N_{u+1} \oplus IMEI))$，將 ID, C_3, C_4, C_5 傳送至雲端。

步驟 2：雲端收到後先對加密的訊息進行對稱式解密 $\text{API}_{req} = D_{SK}(C_3)$，再對訊息做出回覆 $(RESP)$。對 C_4 計算出 N'_{u+1}，$N'_{u+1} = C_4 \oplus h(h(pw') \oplus IMEI)$ 並驗證 N'_{u+1} 之正確性

$h(ID\|h(pw')\|h(N'_{u+1} \oplus IMEI)) \stackrel{?}{=} C_5$ 。

若驗證成功後產生 N_{s+1} 並計算出本次傳輸所需之會議金鑰 SK_{new}，$SK_{new} = h(N_{u+1}\|N_{s+1}) \oplus IMEI$。使用者對雲端要求回應的內容需求經過對稱式加密的保護計算出 C_6，$C_6 = E_{SK_{new}}(RESP)$，並連同

$h(h(pw)\|IMEI\|N_{u+1}) \oplus N_{s+1}$，

$h(IMEI\|N_{u+1}\|N_{s+1})$ 傳送至使用者。

步驟 3：行動使用者收到訊息後先計算出 N'_{s+1}

$$N'_{s+1} = h(h(pw)\|IMEI\|N_{u+1}) \oplus h(h(pw)\|IMEI\|N_{u+1}) \oplus N_{s+1}$$

並驗證其正確性

$h(IMEI\|N_{u+1}\|N_{s+1}) \stackrel{?}{=} h(IMEI\|N_{u+1}\|N'_{s+1})$。驗證成功後再利用將先前產生的 N_{u+1} 計算出 SK_{new}，$SK_{new} = h(N_{u+1}\|N_{s+1}) \oplus IMEI$ 。在對加密訊息 C_6 進行對稱式解密 $(RESP) = D_{SK_{new}}(C_6)$，這時使用者就可以存取先前對雲端做出要求的訊息內容 $(RESP)$。

在我們提出的協定下，使用者使用雲端服務時，經過服務需求階段的流程，可以與持續的與雲端保持安全的溝通。

3. 安全性分析

以下分析是我們提出的雲端架構與行動使用者身份認證協定對於各種攻擊的安全性分析。

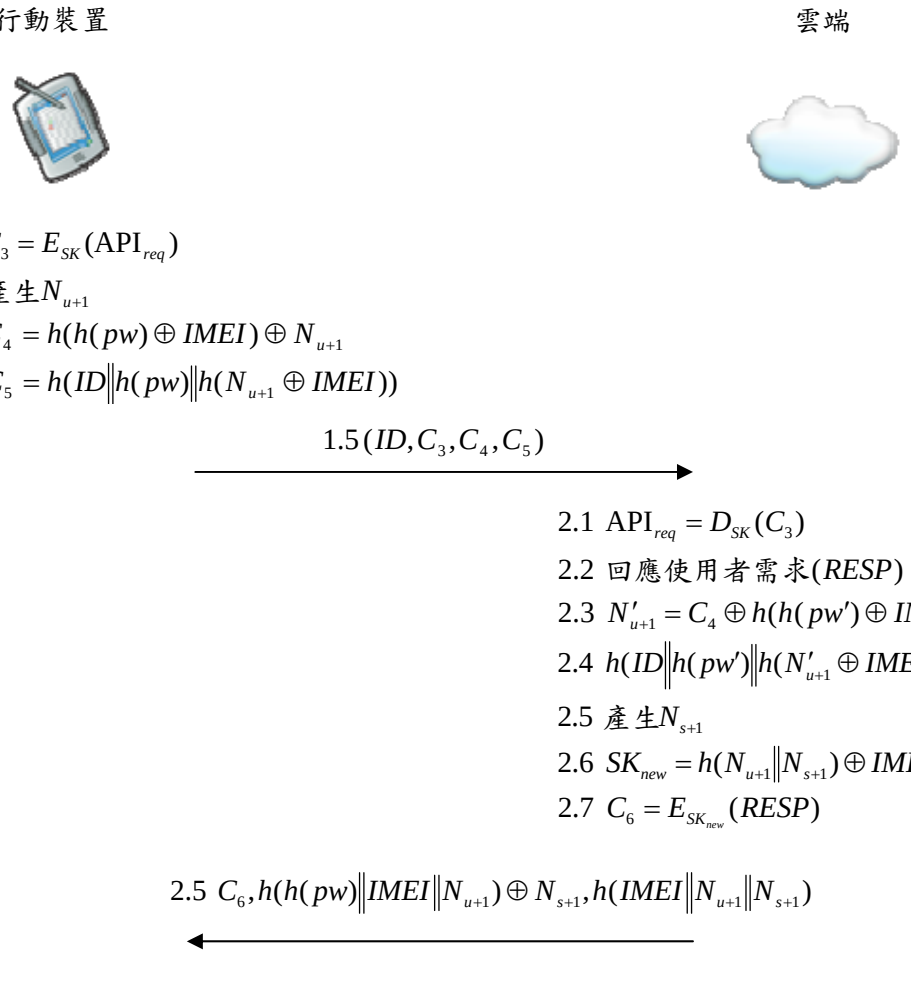
(1) DOS、DDOS 攻擊

以亞馬遜雲端架構網站為例[5]，使用者與伺服器溝通時需先經過防火牆過濾訊息，為了減輕不正常的惡意攻擊還必須同步使用者的瀏覽器 cookies，限制使用者對雲端的連接數。使用者如果不使用瀏覽器對雲端進行惡意攻擊，該伺服器會自動鎖定同一個使用者的使用行為。例如同一個 IP 在一秒內連續要求回應 1000 次訊息，伺服器視為該使用者為惡意攻擊封鎖該 IP 服務。

(2) 密碼猜測攻擊

使用者的行動裝置不會儲存使用者任何資料，攻擊者無法對手機進行離線密碼猜測攻擊。

雲端伺服器在保護使用者帳戶時，無論是攻擊者或合法使用者，密碼輸入錯誤連續數次就會封鎖該帳號，並對該使用者密碼作更改後



圖四、使用者對雲端提出服務需求流程

再寄給使用者註冊時所填寫的信箱。所以攻擊者沒辦法使用線上密碼猜測攻擊，攻擊者或使用者也不會知道先前所設定的密碼。

(3) 內部攻擊

“有價值”的雲端系統[5]會定期要求使用者更換密碼伺服器也會定期更換私鑰，伺服器儲存密碼時不會直接儲存使用者密碼，會先經由單向雜湊函數保護儲存使用者密碼。以 linux 為例[1]內部密碼使用 MD5 單向雜奏函數與 linux 私鑰加密($h(h(pw)\otimes x)$)，攻擊者必須先擁有最高權限(root)在竊取使用者密碼，當攻擊者使用暴力猜測攻擊猜到使用者密碼時與分辨出伺服器的私鑰，伺服器已經更換私鑰，使用者也更換密碼了。

(4) 重送攻擊

在我們的架構中每次溝通的 N_u, N_s 都不一樣即 $C_1 = h(h(pw) \oplus IMEI) \oplus N_u$ 和 $h(h(pw)\|IMEI\|N_u) \oplus N_s$，攻擊者無法利用所攔截溝通過的訊息，再一次使用攔截的訊息對使用者或伺服器作溝通。

(5) 假冒攻擊

每次溝通都會記錄 *ID*、*IMEI*，溝通過程中攻擊者無法假冒使用者，溝通過程中會對 *IMEI* 作比對。攻擊者也無法假冒伺服器，伺服器存有使用者的密碼，當使用者作登入時計算 $C_1 = h(h(pw) \oplus IMEI) \oplus N_u$ 因為密碼 *pw* 是經過 *IMEI* 及單向雜湊函數及 N_u 所保護，使用者的密碼是很難猜測出來的。

(6) 中間者攻擊

每次傳送訊息中都有兩個未知數 N_X 和 $h(pw)$ 加以保護，因此攻擊者攔截訊息 $C_1 = h(h(pw) \oplus IMEI) \oplus N_u$，$h(h(pw)\|IMEI\|N_u) \oplus N_s$ 要對訊息作竄改是不可能的。

(7) 平行會議攻擊

使用者與雲端伺服器溝通時，無論是使用者對雲端溝通 (C_1, C_2, C_4, C_5) 或者雲端對使用者溝通 $h(h(pw)\|IMEI\|N_u) \oplus N_s$，$h(IMEI\|N_u\|N_s)$，使用加密方式都不同，不可能以使用者與雲端伺服器溝通的參數解出雲端伺服器與使用者溝通的值。

4. 結論

我們提出的協定可以解決使用者利用行動裝置使用雲端服務時的安全性，使用者之行動設備不需要儲存個人隱私訊息，也不會儲存伺服器的私鑰。在通訊過程中，使用計算量低的對稱式加密、單向雜湊函數、互斥或閘運算、串接運算，減低移動設備的計算量與通訊量，解決雲端服務架構下的行動使用者的身份認證問題。

參考文獻

[1] 鳥哥的 linux 私房菜, “http://linux.vbird.org”.

[2] B. Ohlman, A. Eriksson, R. Rembarz, “What Networking of information Can Do for Cloud Computing,” the 18th ***IEEE International Workshops on Enabling Technologies : Infrastructures for Collaborative Enterprises,*** pp. 78-83, 2009.

[3] C. L. Chen,”A secure and traceable E-DRM system based on mobile device,” ***Expert systems with applications,*** Vol. 35,No. 3, pp. 878-886, 2008.

[4] R. Rivest, Shamir A., Adleman L., “A Method for Obtaining Digital Signatures and Public Key Cryptosystems,” ***Communications of the ACM,*** Vol. 21, No. 2, 1978.

[5] S. Subashini, V. Kavitha, “A survey on security issues in service delivery models of cloud computing,” ***Journal of Network and Computer Applications,*** Vol. 34, No. 1, In press.

[6] Wiki web site, “http://zh.wikipedia.org/zh-tw/OpenID”