聯宏科技股份有限公司
PAYSECURE TECHNOLOGY CO., LTD.

MOTP 身份認證管理工具
實 習 教 材

台北市內湖路一段 91 巷 17 號 10 樓之 1 WWW.PAYSECURE.COM.TW
TEL:(02)2657-1187 FAX:(02)2657-1205

PaySecure Technology

版本紀錄

版本	日期	修訂章節	說明	備註
1.0	2008/10/6		Initial	

目錄

壹、 系統簡介 1
一、 一次性密碼(OTP)簡介 1
二、 OTP 的種類 2
三、 OTP 標準介紹 6
(一)、 HMAC 演算法 6
(二)、 OTP 標準-OATH HOTP(Event Based) 7
(三)、 OTP 標準-OATH TOTP(Time Based) 8
四、 雙因素認證 9
五、 練習 10
貳、 MOTP 應用及設定 11
一、 遠端登入 11
(一)、 遠端登入 Radius 協定 11
(二)、 遠端登入 MOTP 主機設定 12
(三)、 遠端登入應用端設定 16
二、 Windows Logon 23
三、 網站內容過濾器 Filter 25
(一)、 IIS Filter 25
四、 練習 26
參、 MOTP 網頁整合應用 27
一、 JSP 網頁 27
(一)、 OTP 驗證整合 27
(二)、 新增 OTP 使用者 30
(三)、 OTP 註冊整合 32
(四)、 OTP 同步整合 34
(五)、 練習 36
二、 ASP 網頁 36
(一)、 OTP 驗證整合 37
(二)、 新增 OTP 使用者 38
(三)、 OTP 註冊整合 40
(四)、 OTP 同步整合 42
(五)、 練習 44
三、 PHP 網頁 45
(一)、 OTP 驗證整合 45
(二)、 新增 OTP 使用者 47

(三)、 OTP 註冊整合 49
(四)、 OTP 同步整合 51
(五)、 練習 53
肆、 加值應用 54
一、 網路銀行應用 54
(一)、 行員代註冊網頁 54
(二)、 銀行 OTP 登入網頁 56
(三)、 行員代同步網頁 58
(四)、 練習 59
二、 線上遊戲登入應用練習 59
(一)、 初始登入 60
(二)、 下載 OTP 程式並啟用 MOTP 機制 62
(三)、 玩家註冊手機 64
(四)、 驗證 OTP 開通遊戲 66
(五)、 練習 68
伍、 參考書籍及網站 69

壹、系統簡介

本文將介紹一次性密碼(OTP)身份認證技術原理及全景軟體的 OTP 產品：MOTP 行動動態密碼的整合技術。

一、一次性密碼(OTP)簡介

隨著網際網路盛行，許多使用者皆欲憑藉網際網路取得可用資源與服務，而身分識別是網路服務最基本的基礎建設。

網路銀行被盜轉帳？線上遊戲寶物被盜？隨著電子商務及 Web 服務的蓬勃發展並漸趨成熟，伴隨著資訊安全風險的增加，民眾開始重視自身個人資料的安全性，其實這些資安事件背後都指向一個逐漸被重視的問題—使用者身份識別驗證(Identity Authentication)。

現在不只是 B2C 或是 B2B 之類的網路服務重視使用者身份驗證，就連企業內部系統登入或是電腦本身的權限控管等，都有使用者身份驗證的需求。業務人員在外面可能會透過 VPN 連回公司做收信...等的公司事務，就連員工平常在公司內部收發 Email、登入公司內部網站.....等，都需要通過使用者身份驗證。根據調查，企業在資訊系統上所受到的攻擊，已經有 50%是來自於內部攻擊，除了自身員工之外，也有可能是內部電腦遭到駭客入侵，進一步進行竊取資料的行為。

以往我們所專注防範的外部威脅防禦，也因此逐漸轉為內部權限控管的需求，所以如何做到內部系統權限與資訊存取的安全控管，反而成為目前企業資訊安全上比較被關注的議題。

以往我們登入網路服務網站(網路銀行、線上遊戲...)或是公司內部系統(VPN、Outlook…...)，傳統作法都是利用使用者帳號/密碼(username/password)來做使用者登入驗證的動作。可是這樣的驗證使用者方式，隨著駭客跟木馬、鍵盤側錄(keylogger)、後門程式技術的進步，單純利用使用者名稱/密碼已經不再安全了。因為一旦駭客利用木馬、後門程式或是竊聽網路資料，乃至於目前最常聽到的釣魚(Phishing)網站，都是駭客可以竊取使用者帳號/密碼的方法，而駭客就可以利用所竊得的使用者帳號/密碼來登入網站或公司內部系統，小則造成使用者個人損失，大則可能讓公司機密資料外洩，造成公司更大的損失。

為了防止這些入侵情況的發生，並加強使用者身份驗證的安全性，目前市面上有運用 PKI (Public Key Infrastructure)技術跟一次性密碼(OTP，One-Time Password)技術的兩大類解決方案推出。

PKI 解決方案主要是利用憑證與公私鑰等 PKI 技術，來達到安全的使用者身份驗證，每一位使用者都有其專屬的憑證，該憑證可以用來做身份認證及數位簽章，由於公開金鑰演算法的安全性及金鑰長度都較一般的密碼長，駭客也難以破解憑證。此外，一般的憑證都可以搭配其他的硬體設備來做保護，如 IC 智慧卡(Smart Card)和 USB 載具，使用此類的硬體設備也進一步提高了憑證本身的安全性。以目前來說，PKI 機制的安全性是最高的。不過 PKI 的缺點就是建置成本太高，使得推廣上較為困難。

OTP (One Time Password)，以字面上的解釋就是一次性密碼，也就是這個密碼只會使用一次，當使用者登入過後，當次 OTP 所產生的密碼就變得無效了，而下次要使用的時候就必須重新產生一組數字來作驗證。為什麼說這樣的雙因數認證方式(密碼+OTP)會比較安全呢？因為如果只是再增加一個固定不變的驗證資料(如身份證字號或使用者代碼)，還是無法防止被駭客竊取的問題，而 OTP 的驗證資料每次都會隨機變動，用完即無效，就算是被駭客竊走了這次傳送的 OTP 資料，也無法再次用來做使用者身份的驗證。在系統或網站加入 OTP 功能後，當使用者要登入系統的時候，必須將 OTP Token 所產生的一次性密碼跟使用者帳號/密碼一起輸入，利用這些資料來作使用者身份驗證的動作。

二、OTP 的種類

目前市面上可以看到的 **OTP** 解決方案大概有四種：

甲、 傳統 **OTP** 載具**(Token)**：主要就是使用者會有一個像隨身碟大小一樣的 OTP Token,當使用者需要使用 OTP 的時候,使用者只要隨著時間的改變或按 OTP Token 上的按鈕，以產生一個 OTP 亂數來當做身份驗證的資料。此類載具的外觀樣式千變萬化，有計算機型、鑰匙吊飾型、信用卡片式、隨身碟式及晶片卡式等。是目前最常見的 OTP 種類。

OTP Token 目前常見的產品有 RSA SecurID、Verisign Unified Authentication、Aladdin eToken、Authenex A-Key.....等，除了銀行業及券商在網路銀行或網路下單外，還有企業內部開始導入 OTP 加強身份認證的安全性。

乙、 簡訊 **OTP**：在傳統的 OTP 模式當中，之所以推廣不易且較不能被使用者接受，其中最重要的因素，在於使用者需要攜帶一個硬體 OTP Token,隨著使用者的增加，所需要的 OTP 數量也會隨之增加。不論此 OTP Token 的成本由企業負擔或是使用

者自行負擔，都是在建置 OTP 系統時所必須要考量到的成本問題。此外，使用者必須隨身帶著 OTP Token，如果沒有 OTP Token 的話，就無法進行使用者驗證。

為解決傳統 OTP 的成本問題，最近我們可以看到所謂簡訊 OTP 服務的推出。簡訊 OTP 運作流程就是當你要進行系統使用者驗證的時候，你的手機會接收到一封由系統發出的簡訊，訊息內容就是一個 OTP 的驗證資料，輸入此密碼之後就可以進行驗證。簡訊 OTP 主要在於可以省掉硬體 OTP Token 的成本，取而代之的使用媒介是現代人手邊不可或缺的手機，因為現在幾乎都是人手一機，大家也都會隨身攜帶手機，所以利用手機當做媒介，可以減少要多帶一個 OTP Token 的困擾，也降低忘記帶的可能性。此類 OTP 有另一種變型，採用 Email 通知 OTP。

而簡訊 OTP 因為是被動式接收，所以當使用者遇到釣魚網站的時候，並沒有 OTP 資訊可以輸入，恰巧可以避開釣魚網站的攻擊模式；不過日前曾發生過有駭客假冒使用者跟電信公司謊報 SIM 卡遺失，而從電信公司重新申請同樣號碼的 SIM 卡，以接收 OTP 資訊，並進一步假冒使用者身份登入銀行所提供的網路銀行服務 ，造成使用者的損失。

簡訊 OTP 對於建置的企業而言，多了傳送手機簡訊的成本負擔，而且取得 OTP 的方式為被動方式，使用者必須要 on-line 等待簡訊的傳送，如果在收訊不良的地方，如地下室...等，則會影響簡訊的取得即時性。另外由於採用的是被動式認證，也就是系統必須先知道使用者的身份，才能透過手機系統傳送 OTP 密碼給使用者進行驗證，這樣的模式較適合用在一般網站設計 ，使用者先採用帳號/密碼登入系統，等到要進行更機密的行為時，如證券下單、轉帳、存取機密文件.....等，再向系統要求一組簡訊 OTP 密碼，並輸入做身份驗證。這種兩階段的驗證方式，對於使用在 Windows Logon、VPN、Outlook 的驗證，這類的情形則較不適合使用簡訊 OTP。目前國內已經有銀行業者將簡訊 OTP 機制應用在他們的網路銀行上，提供給網銀使用者來使用。

丙、 密碼查表式 **OTP**：此類 OTP 又可分成兩大派，一個是用實體密碼卡，一個是虛擬密碼卡，所謂的密碼卡是一個 M*N 的表格，每一格都有不同的數字或密碼，使用者根據當時主機提供的訊息來對應密碼卡上所記載的密碼，將它輸入密碼欄位登入系統。

舉一個實體密碼卡的例子來說，使用者手上擁有一張印有以下表格的紙卡 。

	1	2	3	4	5	6	7	8	9
A	1	6	2	7	5	2	6	3	1
B	0	2	1	5	6	9	4	8	4
C	2	1	3	8	9	0	3	5	4
D	3	8	5	4	1	3	8	9	5
E	4	9	2	0	6	7	0	3	7
F	2	1	7	9	0	9	6	7	4

當網頁出現登入畫面時，會同時出現(A5, B7, F9, D1, E3, C2)的一個數列。此時，使用者就必須對照密碼卡的相對位置，輸入 544321 這個密碼來登入網站。當然此例子太過簡單，攻擊者只要收集足夠的資訊即可破解該密碼卡，實際的密碼卡會更複雜一些，比如說加入顏色辨別、每一欄的數字不只一碼…等。

另一種虛擬密碼卡，則是相反，使用者要自行記住座標，例如：(A5, B7, F9, D1, E3, C2)，而在使用者要登入的網頁上顯示上面的密碼表，而該密碼表欄位中的每個數字會每次變動，使用者利用他自己記憶的座標找到對應的數字 544321，然後輸入該密碼登入系統。

不管是哪一派的密碼卡系統，因為必須修改網頁介面，使用的情境都只能應用在自行開發的網頁上，無法應用在其他系統上。而且，由於它並非使用密碼學系統來演算，因此，只要攻擊者收集的資料夠多，比較容易被破解。

丁、 軟體式行動裝置 **OTP**：所謂的行動 OTP，就是利用手機、PDA 各式的行動裝置當作 OTP 使用者媒介，取代傳統的 OTP Token 來做 OTP 密碼資訊的產生。使用者可以根據自己手邊既有的行動裝置，選擇自己最常用的行動裝置，並將 OTP 程式安裝在行動裝置上，以便於日後做為身份驗證之用。為何不直接裝在 PC 上? 軟體式 OTP 其實也可以如同一般軟體一樣裝在使用者的 PC 或 NB 中，但由於木馬及病毒的盛行，若 OTP 的 Token 裝在 PC 或 NB 中，一但中木馬後，因為駭客同時也取得該 OTP Token

的使用權，則 OTP 的機制等同失效，因此，建議使用其他的裝置來產生 OTP 是較安全的做法。

行動 OTP 除了有第二項簡訊 OTP 的優點之外，保留傳統 OTP 的主動式產生 OTP 密碼，當使用者需要作身份驗證時，由使用者自行產生 OTP 密碼來做驗證，而不像簡訊 OTP 採用被動式認證模式，使用者只能等待 OTP 密碼簡訊傳送到手機之後，再進行驗證動作。因為使用的是 off-line(離線)產生驗證亂數，使用者不用擔心無法收到簡訊或簡訊被竊聽的問題，並且不需要簡訊傳送費用，可降低企業的建置成本。

另外由於行動 OTP 所採用的是軟體安裝方式，因此可以將多個 OTP 程式安裝在同一個行動裝置上。而對於傳統 OTP 來說，如果 OTP 系統更換，就需要重新更換 OTP Token，當使用多種 OTP 系統時，就必須要使用多個 OTP Token，造成使用者的不便性，也增加企業的成本支出。而且行動 OTP 除了一般網站的身份驗證之外，還支援各種系統的整合建置，如 Outlook、Windows Logon、Radius Server、VPN...等。

上述介紹的幾種 OTP 的應用，「簡訊式」及「密碼表式」的 OTP，都不需使用密碼學演算法，只需要一個亂數產生器，即可實作這些功能。而「硬體 Token 式」或「軟體行動裝置 OTP 式」的 OTP，則是使用密碼學的演算法 HMAC 來達到 OTP 不可預測的功能。以下介紹幾個常見的 OTP 演算法標準。

三、OTP 標準介紹

(一)、HMAC 演算法

HMAC, keyed-Hash Message Authentication Code，是一種帶有金鑰的訊息驗證碼(MAC)，它的底層可以是任何一種雜湊函式(Hash Function)，如：SHA1 或 MD5，它的安全性也是根據它所使用的雜湊演算法的強度 。

一般的雜湊演算法可以將很大的一串資訊，映射成一個固定長度的資料，以做為訊息資料的檢查碼，當原始訊息有被修改時，該資料的雜湊值將會跟著改變，因此被用來當成訊息傳遞的驗證碼(MAC)，但是 MAC 也可以被有心人士給修改，因此，HMAC 導入了金鑰的概念。訊息的收送雙方共享一把金鑰，在製做訊息驗證碼的同時，加入了金鑰的參數，讓其他不知

道金鑰的人，就算修改了原始訊息，但無法製做出相同的驗證碼，因此可以保護資料的完整性。

只要傳送者與接收者共用一個秘密金鑰，雜湊式訊息驗證碼 (HMAC) 就可以用來判斷透過不安全通道傳送的訊息是否遭到竄改。傳送者會計算原始資料的雜湊值，並且將原始資料和 HMAC 一起當做單一訊息傳送。接收者會重新計算所接收訊息的雜湊值，並且檢查計算出的雜湊值是否與傳輸的雜湊值相符。

HMAC 可以與任何密碼編譯雜湊函式 (例如 MD5 或 SHA-1) 一起使用，並與秘密共用金鑰組合。HMAC 的密碼編譯強度取決於基礎雜湊函式的屬性。

對資料或雜湊值所做的任何變更都將導致結果不相符，因為必須具有秘密金鑰的資訊，才能變更訊息並重新產生正確的雜湊值。因此，如果原始雜湊值和計算出的雜湊值相符，此訊息便通過驗證

由於該演算法有每次不同、不可預測、亂度夠、有金鑰機制等特性，非常適合用在 OTP 的領域，於是被拿來當成 OTP 的標準演算法。

最簡單的 OTP 演算法為，一開始隨機產生第一個 OTP，及約定金鑰 key，第二個 OTP 為第一個 OTP 的 HMAC 值，再取 OTP 值，演算法如下：

$$OTP_{n+1} = HMAC(key, OTP_n)$$

使用者及認證端採用同一把金鑰來產生同樣的 OTP 序列，即可用來實作一次性密碼的基礎。這是最早也是最簡單的 OTP 演算法。但此方式有缺點，若雙方的 OTP 不同步時，很難直接找到正確的 OTP 來同步，必須從頭開始算。因此，出現了以下的標準。

(二)、OTP 標準-OATH HOTP(Event Based)

OATH，Open Authentication Organization，一個制定認證標準的國際組織，在 2005 年訂出了一個基於 HMAC 的標準 HOTP 演算法，改良了舊式的 OTP 演算法，在輸入端由上一個 OTP 修改為目前的 Counter 值(OTP 的產生次數)。再加強 OTP 位數的選擇方式 (Truncate)，可以是 6~8 碼的 OTP，其演算法如下：

$$OTP_c = HOTP(Key, Counter) = Truncate(HMAC - SHA1(Key, Counter))$$

要計算第 n 個 OTP 值，直接將 n 帶入 Counter 參數，即可算出該 OTP，不需從頭算起。而 Truncate 函式主要在處理將 Hash 函式處理完的資料轉換成 6~8 碼的 10 進位數字。

當使用者及認證端用同一把金鑰，同步一樣的 counter 即可驗證該使用者所產生的動態密碼。

由於此演算法必須同步 counter 值，若使用者一直產生 OTP 而不到認證端使用該 OTP，會造成使用者與主機的 counter 相差太多，系統通常會設定一個容許值(threshold)，而超過此容許值的 OTP 將不被認可，必須經過再同步的程序，將兩邊的 Counter 設定成一樣的數值。

Counter	Client Key	Server
0	913450	913450
1	158394	158394
2	548977	548977
3	749503	749503
.	560042	560042
.	.	.
.	.	.
.	.	.
.	.	.
n	758043	758043

(三)、OTP 標準-OATH TOTP(Time Based)

另一種 OTP 的產生方式，將 OTP 的產生因子改成用時間來計算，也就是將 HOTP 的 counter 參數改為 time 參數，此系統將根據時間的不同產生不一樣的 OTP。嚴格來說，此

方式不算一次性密碼，因為，在認可的時間內，該密碼都有效，並無使用一次的特性，但由於它的密碼有時效性，所以很多 OTP 的系統也採取以時間為基礎的 OTP 演算法。

而此演算法如下：

$$OTP_t = HOTP(Key, Timer) = Truncate(HMAC - SHA1(Key, Timer))$$

使用此類 OTP 前，必須先讓使用者端的時間，與認證端的時間能夠同步，若使用者端使用的時間無法支援 UTC 的話，也會產生一些問題。

time	Client	Key	Server
200803101600	913450		913450
200803101601	158394		158394
200803101602	548977		548977
200803101603	749503		749503
.	560042		560042
.	.		.
.	.		.
.	.		.
.	.		.
20080310160n	758043		758043

四、雙因素認證

既然 OTP 如此神奇，那就所有的認證都只用 OTP 就好了嗎?以安全的觀點來看，OTP 載具有遺失風險，而且只有 6~8 位的數字，密碼強度稍微薄弱，禁不起暴力攻擊。因此，在安全需求度較高的地方，還是會建議使用者要搭配固定密碼(6~8 碼，文數字交錯)，配合一次性密碼 OTP，如此幾乎可防止所有的帳號密碼攻擊 。

五、練習

請利用亂數產生器設計一個隨機式的一次性密碼 OTP 產生器，此產生器可產生 6 位數字的一次性密碼。(可使用各式語言，如： C, JAVA…)

請針對上述亂數產生器加入一個驗證機制 。提示:把之前產生的 OTP 記錄起來。

貳、MOTP 應用及設定

OTP 的應用可以用在所有需要帳號密碼登入的系統中 ，但在某些條件限制下(須修改系統)，較常見應用 OTP 的系統，如下：

1. 遠端登入
2. Windows 登入
3. 網頁過濾器 filter

因為這幾個應用可透過既有的通訊協定 Radius，或已開發好的安裝套件，直接應用到系統上，所以廣為大眾所接受，接下的本節將介紹以上的幾種應用及其設定方式 。

一、遠端登入

遠端登入，指的是由企業外 Internet，透過一些安全通道，如：SSL VPN、Firewall 或 Windows PPTP VPN 等，連結到企業內部 Intranet 查詢公司內部資料或執行一些公司內部應用程式。

(一)、遠端登入 Radius 協定

Radius 是一個認證及授權的協定，MOTP 主機本身是一個 Radius 的認證主機，所有前端的安全閘道器系統，包括：SSL VPN, UTM, Firewall, Citrix Access Gateway 或者 Windows PPTP VPN 等系統，都可以透過 Radius 的協定，連結到 MOTP 主機進行 OTP 或雙因素認證。

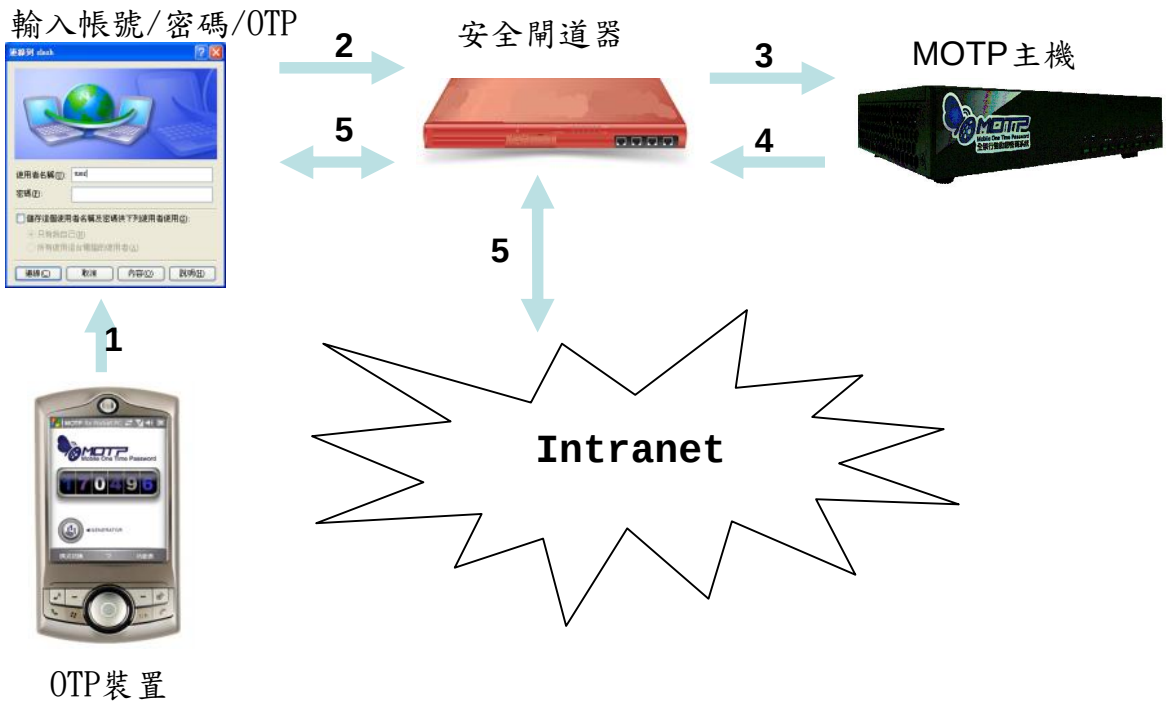


圖 1. Radius 應用架構圖

OTP 在遠端登入的應用流程：

1. 產生 OTP，將帳號/密碼/OTP 輸入登入畫面(連線或瀏覽器)
2. 安全閘道器主機收到「帳號/密碼/OTP」
3. 安全閘道器將「帳號/密碼+OTP」或者「帳號/OTP」送到 MOTP 主機驗證 OTP
4. 驗證結果
5. 建立安全通道，進入 Intranet

(二)、遠端登入 MOTP 主機設定

Radius 是一個認證及授權的協定，認證的要求端及回覆端都必須先設定雙方的 IP 及溝通加密密碼，以下列出 MOTP 主機及幾款 VPN 的設定方式，以提供練習之用。另外，因為 MOTP 特別針對 Radius 提供雙因素認證模式，也在此章節介紹。

1、MOTP 設定

登入管理網頁後，進入「Radius 設定」功能，於功能選單中點擊「系統管理」的「Radius 設定」，如下圖所示：

圖 2. 功能選單—Radius 設定

點擊「Radius 設定」後，會出現以下的畫面：

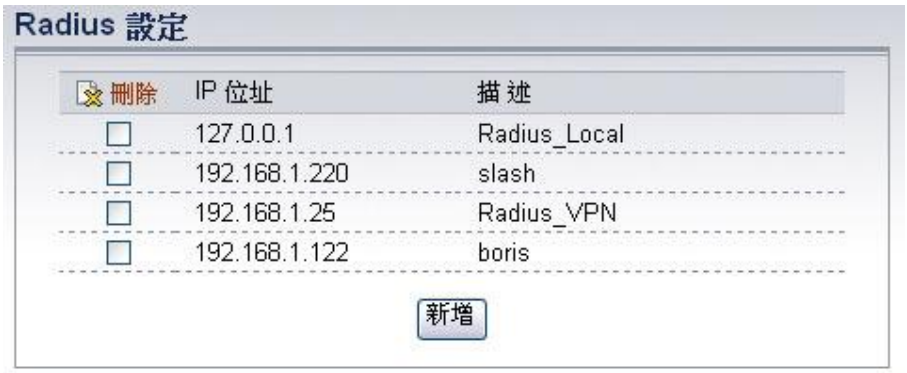


圖 3. Radius 設定

1. 頁面列出所有 Radius 設定，當游標滑入每一列資料上時，點選後能連結至該項設定的修改頁面，可瀏覽詳細的設定值，修改 Radius 設定畫面如下：

圖 4. 修改 Radius 設定

2. Radius 設定列表上，每個設定左側皆有核取方塊，核取後點選上方刪除移除設定，系統會跳出小視窗作多重確認，確認刪除後則無法復原。

3. 若點選新增可新增一組 Radius 設定，其畫面如下：

圖 5. 新增 Radius 設定

4. 將該 VPN 閘道器的 IP 及共享金鑰填入後，按下更新即可。

2、雙因數認證設定

有些高階的安全閘道器，如 Citrix，本身可以支援雙因數認證，在其介面可以個別設定兩個認證主機，此時 MOTP 只需單純驗證 OTP 即可。但大多數低階的 SSL VPN，一次只支援一個 Radius 認證主機，此時就必須透過 MOTP 的雙因數認證機制來達到此功能。若啟用 MOTP 的雙因數認證功能，只須在原先安全閘道器的密碼欄位，輸入密碼及 OTP 即可，例如：原先密碼為「pass」，而 Token 產生的 OTP 為「135790」，則在密碼欄位輸入「pass135790」即可。

1. 雙因數設定

設定[系統參數]->[MOTP 設定]->[Two Factor]，此值可為三種 0,1,2.

0：表示只驗 OTP，適合用在可自行處理雙因素的高階安全閘道器 。

1：表示使用 MOTP 的管理者密碼當成雙因數的第一個固定密碼。若使用此機制，所有 OTP 使用者都必須升級成管理者，但真正的管理者應自行新增一個無任何權限的角色，讓這些 OTP 使用者都歸屬於此角色。

2：表示使用 AD 密碼當成雙因數的第一個固定密碼。若使用此機制，管理者必須設定 AD 的連線資料，[OTP 使用者]->[同步使用者資料]->[AD/LDAP 同步]。

2. AD/LDAP 設定

當主機位置及登入帳號及密碼設定好後，可以按連線測試來測試 AD 網路是否正確。有些 AD 系統必須輸入網域，方可認證，可勾選 Append 後，輸入該單位的網域，認證時系統將自動帶入網域認證。例如：登入時帳號必須輸入/domain/account 時，就必須設定此欄位。

圖 6. AD/LDAP 同步設定

(三)、遠端登入應用端設定

1、SSL VPN

虛擬私有網路（Virtual Private Network, VPN）是在 Internet 上使用安全通道及加密方法建立一個私人且安全的網路。所使用的加密技術主要有兩種方式，一種是標準的 IPSec（IP Security）方式，適用於點對點的加密通道。另一種是 SSL VPN，SSL VPN 運用瀏覽器或 Windows 與 VPN 閘道器建立 SSL 連線，將資料加密，讓使用者可以透過此 SSL 通道存取一些受保護的資料。讓使用者從網際網路上任何地方，透過 SSL VPN 連線後，能夠像在公司內部使用 Intranet 時，存取相關的應用程式及內部資料。

本系統可加強 SSL VPN 閘道器的身份認證功能，讓所有透過 SSL VPN 的連線都需要經

過 OTP 的驗證後才能使用。

由於 VPN 閘道器的種類繁多，但設定方式大同小異，主要分成幾個步驟：

1. 將 SSL VPN 的登入認證方式，設定為 Radius 驗證。有些 VPN 只有一個 Radius 主機，有些則可以根據使用者群組的不同，使用多組 Radius 主機。
2. 將 Radius 驗證的主機 IP 設為 MOTP 主機。
3. 設定共享金鑰(KEY)與 MOTP 上相同，須兩兩配對。
4. 驗證方式設為 PAP。
5. 某些 SSL VPN(如：合勤)必須設定固定的「使用者群組」，方可成驗證，請將 Group 設成"FSMOTP"，即可。

以下以 O2 SSL VPN 的設定為例：

以 O2 Micro 的 SSL VPN 而言，首先新增一個 MOTP 的 Radius 驗證連線，在 Portal 中將它指定為認證選項，設定使用者的認證方式即可。在登入時，使用者利用瀏覽器登入 VPN 的認證網頁時，選擇 OTP 的認證選項，即可使用 OTP 認證服務了。如附圖：

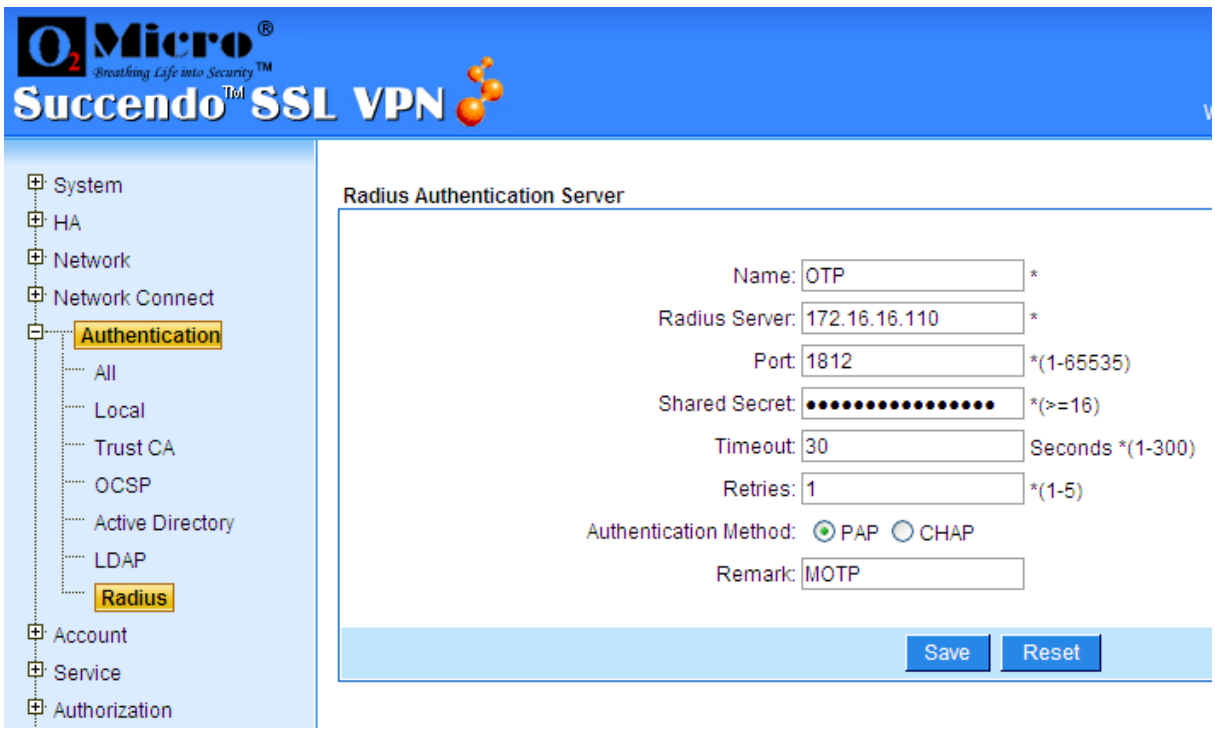


圖 7. Radius Setting

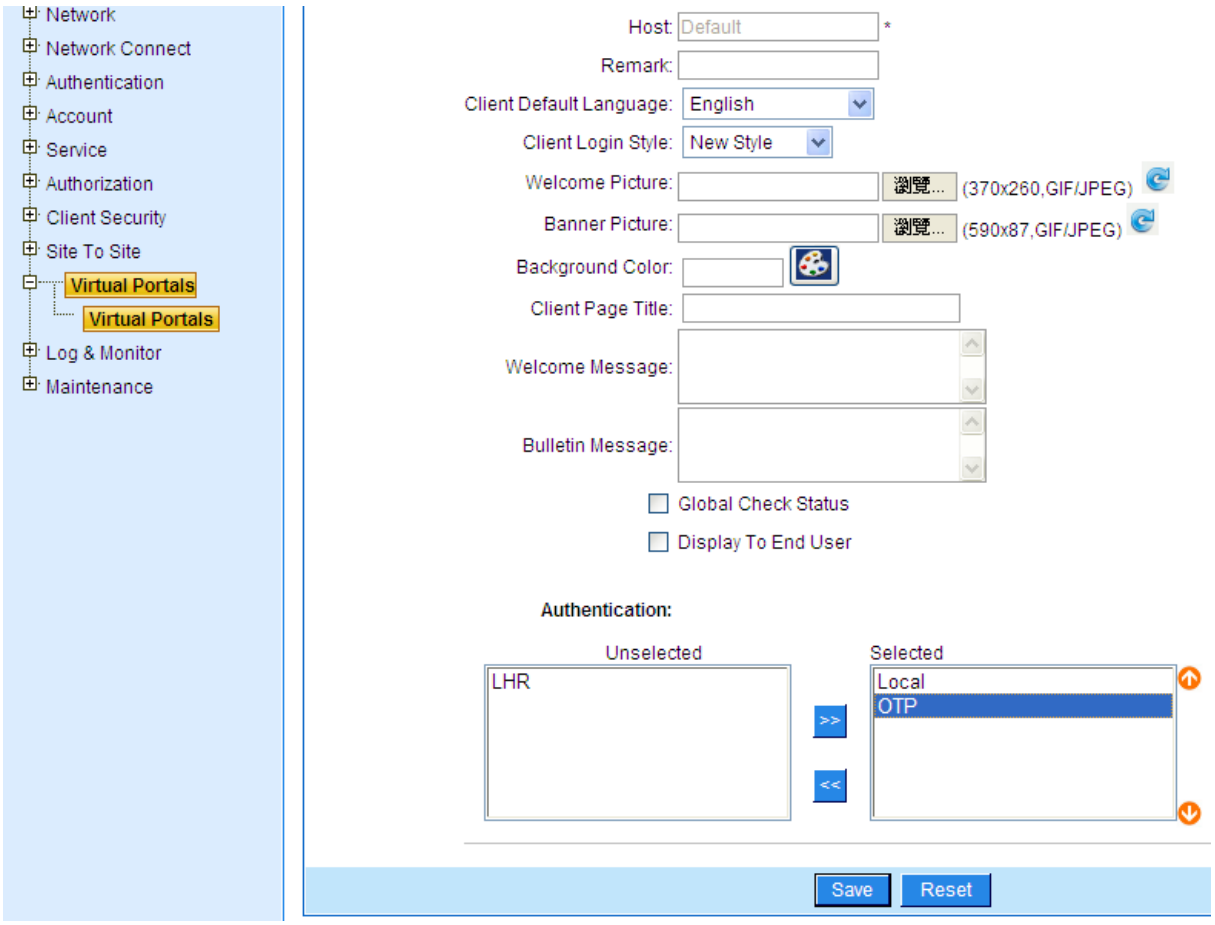


圖 8. Add OTP to be an option in Portal

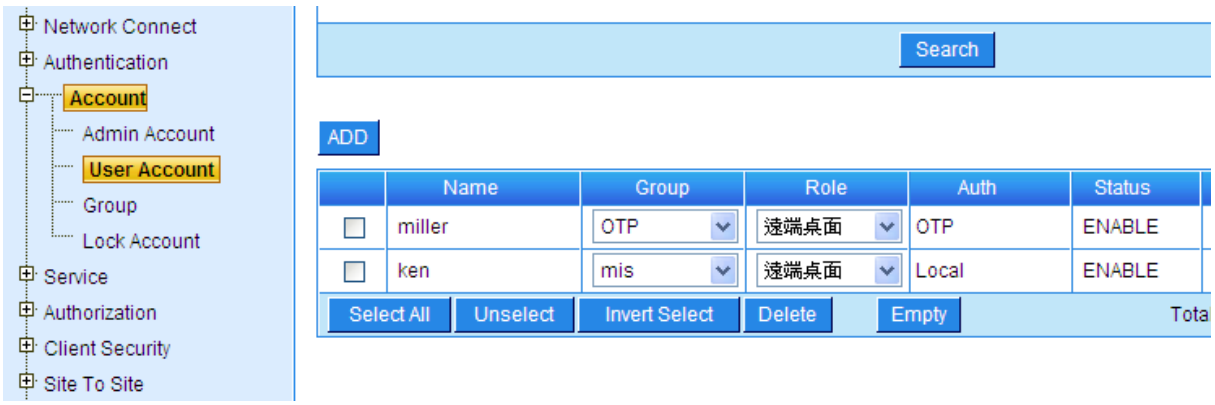


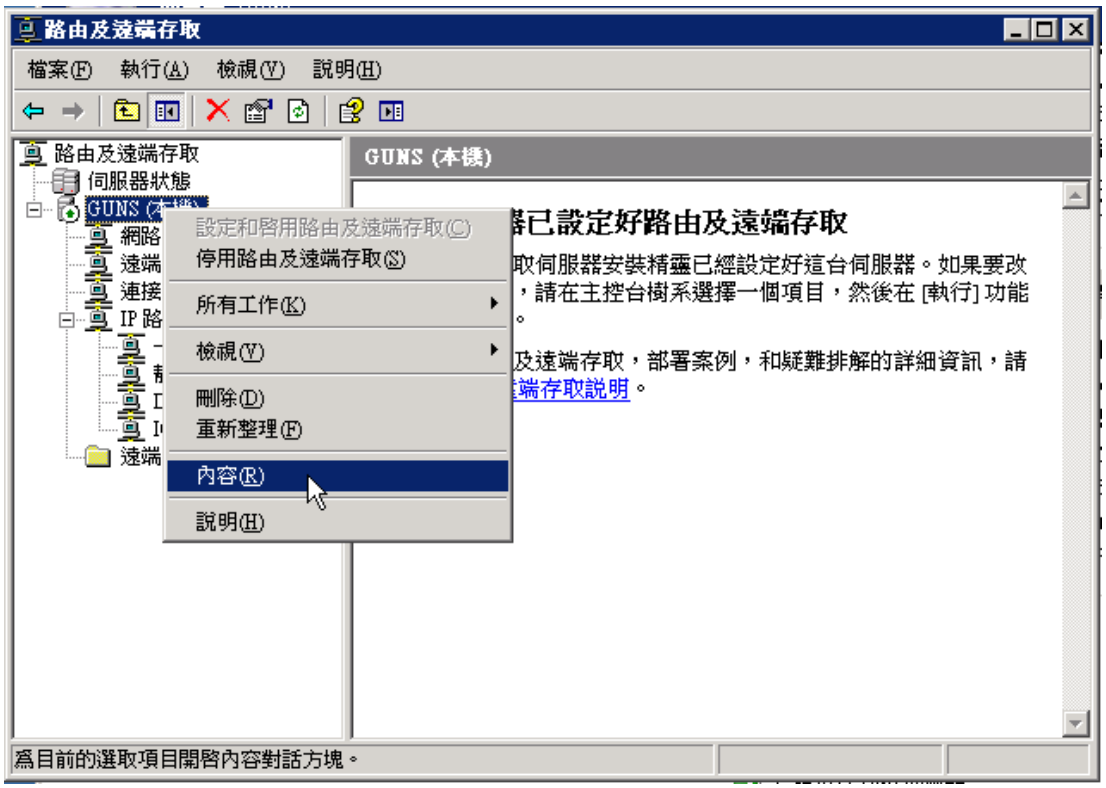
圖 9. Add an OTP User

圖 10. Portal Login Page

2、Windows PPTP VPN

除了一般的硬體 SSL VPN 機器外，也可以利用 Windows Server(2000 above)中內建的遠端存取或 VPN 伺服器功能，提供遠端存取的應用。設定方式如下：

在[管理您的伺服器]介面中，選擇[管理這台遠端存取或 VPN 伺服器]。右鍵選擇[本機]，[內容]。將這台電腦啟用為遠端存取伺服器。選擇[安全性], [驗證提供者]設定為[RADIUS 驗證]。設定 Radius，新增伺服器，設到 MOTP 主機。驗證方法選擇 PAP。設定完成。



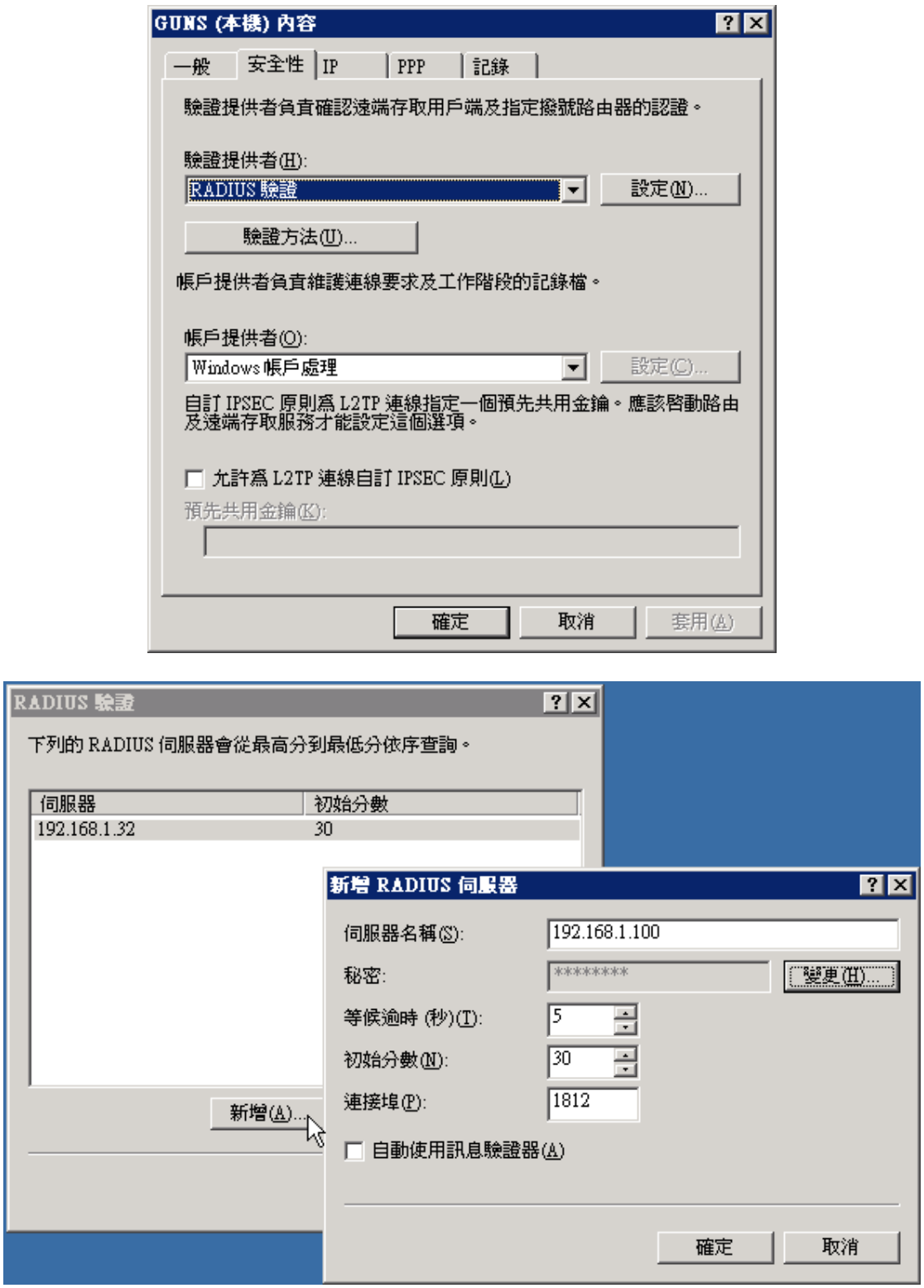


圖 11. Windows PPTP VPN Setting

在使用者登入端，[新增網路連線]，[連線到公司網路]，[虛擬私人連線網路]，設定連線名稱、IP，最後新增捷徑到桌面。點選該連線輸入使用者名稱及 OTP 即可完成登入驗證。

圖 12. Windows PPTP VPN Client setting and logon

二、 Windows Logon

當員工在公司內部登入 Windows 網域時，通常必須輸入帳號及密碼來認證使用者的身份。而近來由於資安事件頻傳，許多企業主都引進一些安全政策，如: BS7799(目前稱為 ISO17799)等安全稽核規範。在此規範中提到許多有關帳號安全的規定，若使用 OTP 加強身份認證的安全性，則可以很輕易的達到該安全規範的要求。本段將介紹如何使用 MOTP 系統來加強身份認證的安全性。在本應用中必須安裝兩個外掛程式 Windows Logon Plug-in 及 MOTP Plug-in for DC proxy。

Windows Logon Plug-in 須安裝在所有要使用 OTP 登入的一般使用者電腦上，此程式將置換掉原來 Windows 的登入畫面，當要登入網域時，跳出須輸入 OTP 的認證視窗。主要在 Client 端的作業系統(WinXP or Win2000) 上安裝一組動態密碼(One-Time Password, OTP)模組，每當使用者要登入網域或系統時，除了輸入原先的密碼外，額外輸入一組 OTP 動態密碼，因而達到雙因素認證。配合公司政策、也可以安裝只須輸入 OTP 的登入模組。

MOTP Plug-in for DC proxy 則是安裝在 Windows AD(Domain Controller)上的外掛程式，負責相關身份證證訊息的傳遞。

MOTP 的 Windows Logon 支援兩種登入模式，「雙因數」及「OTP」。

1、「WinLogon-雙因素」架構與流程說明

本端登入(Local login)輸入原系統密碼即可，而網域登入(Domain login)需輸入系統密碼和一組 OTP，架構如圖 1 所示。

1. 使用者登入時將送帳號和 OTP 到 AD 端的 DC proxy 程式。
2. 當 DC proxy 收到後將帳號和 OTP 轉送至 MOTP Server 主機驗證。
3. 回傳 MOTP 驗證結果，若成功就允許登入系統，若失敗則不允許登入系統並回傳錯誤訊息。

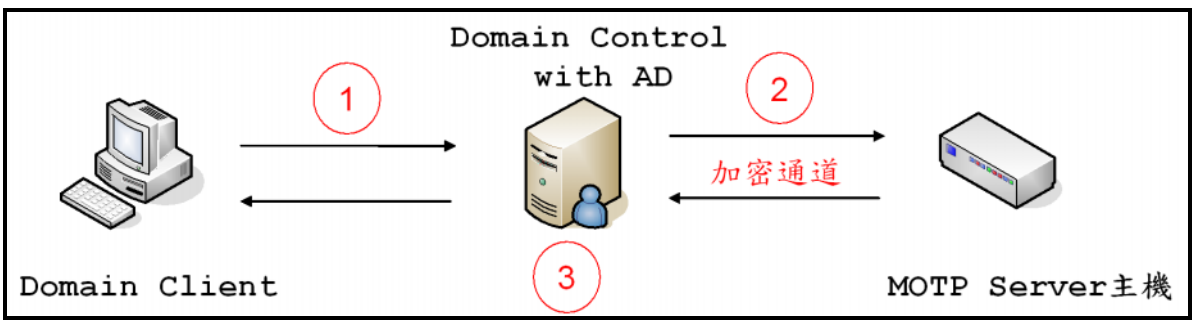


圖 13 WinLogon-雙因素登入架構圖

2、「WinLogon-OTP」架構與流程說明

本端登入輸入系統密碼即可，而網域登入只需輸入一組動態密碼，架構及流程如圖 2 所示。

1. 使用者登入時將送帳號、OTP 和網域名稱(Domain Name)到 AD 端的 DC proxy 程式。
2. 當 DC proxy 收到帳號、OTP 和網域名稱取出帳號、OTP 轉送到 MOTP Server 驗證 OTP 是否正確。
3. 回傳 MOTP Server 驗證的結果，若 OTP 正確將到 Step4 變更此帳號在 AD 上的密碼，若 OTP 不正確將到 Step5 回傳錯誤訊息。
4. 當 OTP 驗證成功將送帳號、網域名稱變更使用者的密碼，每次變更的密碼都不一樣，變更完成後到 Step5 將新的密碼傳給 Client 端。
5. 訊息回傳，若成功將收到此帳號新的密碼且加密儲存並登入系統。若失敗將回傳錯誤訊息並登入失敗。

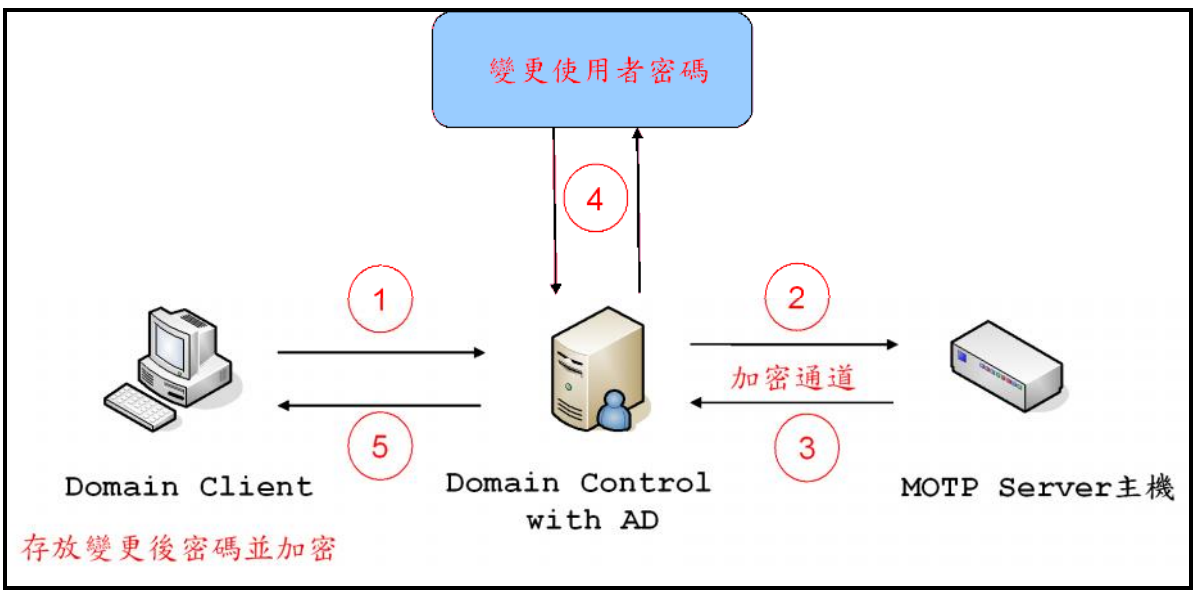


圖 14 WinLogon-OTP 登入架構圖

詳細的安裝說明請參閱「Plugin-DC Proxy 安裝手冊」以及「Plugin-WinLogon Client 安裝手冊」。

三、網站內容過濾器 Filter

在某些情況下，網站的管理者可能需要限制某些網頁或文件必須先通過 OTP 的認證方可允許存取；或者管理者想要保護整個網站卻不想修改網頁的登入機制，此時就可以使用 MOTP 的網站內容過濾器 Filter 來限制網站的存取。

一般的動態網頁，可透過授權的方式來限制使用者的存取，但對於靜態網頁或一般檔案卻無法有效保護，因此對於一般網頁或文件的保護亦可使用此 Filter 套件達到存取控制的效果。

本系統提供兩種 Web Server 的 Filter，IIS 以及 Tomcat。分述如下：

(一)、IIS Filter

當 IIS Server 安裝了「MOTP Plug-in for IIS」軟體後，開啟 IIS 預設的網站內容在 ISAPI 篩選器新增 Filter 程式，利用此 Filter 處理瀏覽網頁訊息並到傳到後端 MOTP Server 驗證。

使用者在瀏覽一般網頁時就像平時瀏覽網頁一般，但是當瀏覽到受保護的網頁時，系統會將網頁導到 OTP 的認證網頁，此時使用者必須輸入帳號及 OTP 進行驗證，若驗證成功則進入網頁，若失敗則無法進入網頁。

Filter 運作流程：

1. Client 端 User 登入受保護網頁時輸入一組帳號及 OTP 送至 IIS Server。
2. IIS Server 利用 Filter 將帳號及 OTP 送至 MOTP Server 驗證。
3. 若驗證成功則可進入網頁瀏覽，反之驗證失敗則無法進入網頁。

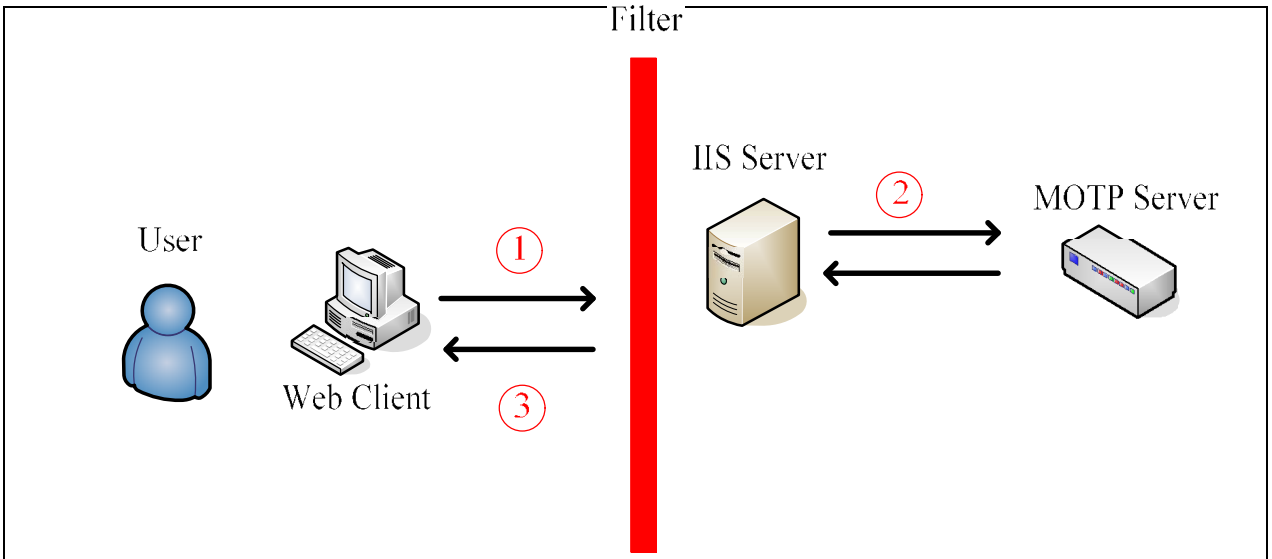


圖 15 IIS Plug-in 架構圖

有關 IIS Plug-in 的詳細安裝及設定，請參考「MOTP_IIS Filter 安裝及操作手冊」。

四、練習

1. 練習設定一個 Windows PPTP VPN 連線，並套用 OTP 身份認證方式。
2. 練習安裝一套使用 MOTP 的 Windows Logon 系統。
3. 練習安裝一套 OWA 系統。
4. 練習設定 IIS Filter 系統，並設定該主機的安全政策。

參、MOTP 網頁整合應用

上一章所談到的應用都已經有相關的協定及安裝套件，管理者只需安裝完畢即可套用，而本章節將介紹，如何在一般的認證網頁上套用 MOTP 系統，加強網站的安全認證機制。

以下介紹三種網站語言的 API 整合，JSP、ASP 及 PHP。

一、JSP 網頁

JSP 為 JAVA 的網頁應用程式語言，以下使用 Tomcat 為伺服器當範例。

在整合網頁前，必須先做好環境設定，操作步驟如下：

將 **MOTPFacade.jar** 和 **formosoft-util.jar** 兩個檔案放在

%Tomcat%/webapps/%APName%/WEB-INF/lib 目錄中。

JSP 網頁必須匯入所需元件，設定內容如下：

```
<%@ page import="com.formosoft.motp.stub.Constant"%>
<%@ page import="com.formosoft.motp.stub.MOTPFacad e"%>
```

(一)、OTP 驗證整合

由於一般的認證網頁都有原來的帳號密碼驗證機制，因此只需在原先密碼的下方新增一個 OTP 欄位，再呼叫 MOTP 的 API 即可。如下圖所示：

圖 16 原始登入畫面

圖 17 加入 OTP 驗證後的登入畫面

在此節介紹兩隻驗證的 API，FSMOTP_AuthOTP 只單純的驗證 OTP，而 FSMOTP_AuthOTPex 則在驗證成功後，會回傳一個驗證碼，網頁設計者可以將此驗證碼顯示在登入成功的頁面上，若使用者的載具是有驗證碼機制的 OTP Token，則會同時顯示該驗證碼，此時使用者可比對兩個驗證碼是否相同，若不同即表示該網站可能為釣魚網站，使用者必須小心注意。

FSMOTP_AuthOTP	
函式宣告	String[] FSMOTP_AuthOTP(String sUrl, String sAccount, String sOTP, String sLogmsg, int iFlags);
說明	驗證 OTP
參數	sUrl: MOTP 主機位置 sAccount: 使用者帳號 sOTP: 密碼 sLogmsg: 保留參數 iFlags: 保留參數
回傳值	第一個元素為錯誤代碼，0 為成功，其他值為失敗。 第二個元素為錯誤訊息。
參考	

FSMOTP_AuthOTPex	
函式宣告	String[] FSMOTP_AuthOTPEx(String sUrl, String sAccount, String sOTP, String sLogmsg, int iFlags);
說明	驗證 OTP，並回傳驗證碼，為雙向認證使用。
參數	sUrl: MOTP 主機位置 sAccount: 使用者帳號 sOTP: 密碼 sLogmsg: 保留參數 iFlags: 保留參數
回傳值	第一個元素為錯誤代碼，0 為成功，其他值為失敗。 第二個元素為錯誤訊息。 若為成功，第三個元素為驗證碼。
參考	

在驗證 OTP 的網頁上加入以下程式碼：

```
<%@ page import="com.formosoft.motp.stub.Constant"%>
<%@ page import="com.formosoft.motp.stub.MOTPFacade"%>
<%
String sUrl = "http://motp.xxx.com.tw/motp/MOTP";
String userName = request.getParameter("userName");  //取得 ID
String otpPass = request.getParameter("otpPass");  //取得 OTP 值
//呼叫 API 送到 MOTP 主機驗證
String result[] =  MOTPFacade.FSMOTP_AuthOTP(sUrl,userName,otpPass,"",0);
//String result[] =  MOTPFacade.FSMOTP_AuthOTPex(sUrl,userName,otpPass,"",0);
int code = Integer.parseInt(result[0]);
if(code==Constant.SERVER_RTN_SUCCESS){
    //成功，do something,  if(FSMOTP_AuthOTPex) show 驗證碼 = result[2]
}else{
    //失敗，do something
}
%>
```

(二)、新增 OTP 使用者

當一個使用者要開始用 OTP 來保護帳號時，在 MOTP 主機中必須先新增此帳號，再使用下一節「OTP 註冊」，讓使用者登記該 Token 的資訊。使用說明如下：

FSMOTP_CreateOTPUser	
函式宣告	String[] FSMOTP_CreateOTPUser(String sUrl, String opAcct, String opPwd, String sAccount, int deviceType, String brand, String model, int clientLang, String mobileNumber, String eMail1, String eMail2, String sDesc, String sLogmsg, int iFlags);
說明	新增一個 OTP 使用者
參數	sUrl: MOTP 主機位置 opAcct: 管理者帳號 opPwd: 管理者密碼 sAccount: 使用者帳號 deviceType: 載具類型 (30:Windows Phone,50:JAVA Phone,70:myPass,90:Hardware)

	brand: 載具設定檔，(standard) model: 載具類型字串 clientLang: 語言(0:English, 1:中文) mobileNumber: 手機號碼(非必要欄位，使用簡訊下載通知時使用) eMail1: 電子郵件帳號 eMail2: 保留參數 sDesc: 使用者描述 sLogmsg: 保留參數 iFlags: 特別參數選項，可使用相加 (1:強制新增, 2:線上註冊, 4:不寄 Email)
回傳值	第一個元素為錯誤代碼，0 為成功，其他值為失敗。 第二個元素為錯誤訊息。
參考	iFlags: 特別參數選項，可使用相加同時使用多種選項 1:強制新增, 不管帳號是否已存在。 2:線上註冊, 網頁上及 Email 中的註冊訊息將不顯示帳號及臨時密碼。 4:不寄 Email

範例如下：

```
<%@ page import="com.formosoft.motp.stub.Constant"%>
<%@ page import="com.formosoft.motp.stub.MOTPFacade"%>
<%
String sUrl = "http://motp.xxx.com.tw/motp/MOTP";
String user = request.getParameter("userName");
String pass = DigestUtils.shaHex(request.getParameter("userPass"));
String sAccount = request.getParameter("sAccount");
int deviceType=30;
String brand="standard";
String sEmail = request.getParameter("sEmail");
String fault[] =
MOTPFacade.FSMOTP_CreateOTPUser(sUrl,user,pass,sAccount,deviceType,brand,
"",1,"",sEmail,"","","",0);
if(result[0].equals("0")){
    //成功，do something
}else{
    //失敗，do something
}
%>
```

(三)、OTP 註冊整合

因為本系統可同時接受硬體 Token 及軟體 Token 兩種載具，由於載具型態不同，註冊的程序也有些許不同。硬體載具的使用者只需將載具背面的序號填入該開通帳號即可；而軟體載具使用者，則必須先下載安裝手機程式，並在使用 MOTP 服務前，先將手機上 OTP 程式所顯示的資訊註冊到系統上，做類似「開卡」的動作，系統才能正確的運作，以避免其他非法使用者的惡意登入。因此註冊的呼叫方式分為軟體載具 FSMOTP_RegisterOTPUser 和硬體載具 FSMOTP_RegisterHwTokenUser 兩種：

FSMOTP_RegisterOTPUser	
函式宣告	String[] FSMOTP_RegisterOTPUser(String sUrl, String opAcct, String opPwd, String sAccount, int iHotpId, String sSerial, String sDeviceID, String factMoving, String sLogmsg, int iFlags);
說明	登記 OTP 使用者的註冊資訊
參數	sUrl: MOTP 主機位置 opAcct: 管理者帳號 opPwd: 管理者密碼 sAccount: 使用者帳號 iHotpId: 載具代號(-1:自動尋找未註冊之載具) sSerial: 手機上顯示之 Token 序號 sDeviceID: 載具的機器碼 factMoving: OTP 計數器 sLogmsg: 保留參數 iFlags: 保留參數
回傳值	第一個元素為錯誤代碼，0 為成功，其他值為失敗。 第二個元素為錯誤訊息。
參考	

FSMOTP_RegisterHwTokenUser	
函式宣告	String[] FSMOTP_RegisterHwTokenUser(String sUrl, String opAcct, String opPwd, String sAccount, String sSerial, String timeCounter, String sLogmsg, int iFlags);
說明	登記硬體 Token 使用者的註冊資訊
參數	sUrl: MOTP 主機位置

	opAcct: 管理者帳號 opPwd: 管理者密碼 sAccount: 使用者帳號 sSerial: 載具的硬體序號 timeCounter: 保留參數 sLogmsg: 保留參數 iFlags: 保留參數
回傳值	第一個元素為錯誤代碼，0 為成功，其他值為失敗。 第二個元素為錯誤訊息。
參考	

程式碼的範例如下：

1、軟體載具

```
<%@ page import="com.formosoft.motp.stub.Constant"%>
<%@ page import="com.formosoft.motp.stub.MOTPFacade"%>
<%
String sUrl = "http://motp.xxx.com.tw/motp/MOTP";
String user = request.getParameter("userName");
String pass = DigestUtils.shaHex(request.getParameter("userPass"));
String sAccount = request.getParameter("sAccount");
String sSerial = request.getParameter("sSerial");
String sDeviceID = request.getParameter("sDeviceID");
long factMoving = Integer.parseInt(request.getParameter("factMoving"));
String fault[] = MOTPFacade.FSMOTP_RegisterOTPUser(sUrl,user,pass,sAccount,-1,
sSerial,sDeviceID,factMoving,"",0);
int code = Integer.parseInt(result[0]);
if(result[0].equals("0")){
    //成功，do something
}else{
    //失敗，do something
}
%>
```

2、硬體載具

```
<%@ page import="com.formosoft.motp.stub.Constant"%>
<%@ page import="com.formosoft.motp.stub.MOTPFacade"%>
<%
String sUrl = "http://motp.xxx.com.tw/motp/MOTP";
String user = request.getParameter("userName");
String pass = DigestUtils.shaHex(request.getParameter("userPass"));
String sAccount = request.getParameter("sAccount");
String sSerial = request.getParameter("sSerial");
String fault[] =
MOTPFacade.FSMOTP_RegisterHwTokenUser(sUrl,user,pass,sAccount,sSerial,0,"",0);
int code = Integer.parseInt(result[0]);
if(result[0].equals("0")){
    //成功，do something
}else{
    //失敗，do something
}
%>
```

(四)、OTP 同步整合

當使用者發現輸入 OTP 多次都驗證失敗時，很有可能使用者的行動裝置已經出現未同步的狀況，可透過此功能與系統作溝通同步。呼叫方式同樣分為軟體載具和硬體載具兩種：

FSMOTP_ReSync	
函式宣告	String[] FSMOTP_ReSync(String sUrl, String opAcct, String opPwd, String sAccount, String factMoving, String sTokenNum, String OTP1, String OTP2, String sLogmsg, int iFlags);
說明	同步 OTP 載具的設定
參數	sUrl: MOTP 主機位置 opAcct: 管理者帳號 opPwd: 管理者密碼 sAccount: 使用者帳號 factMoving: 軟體載具 counter(若為硬體載具, 此欄位填 0) sTokenNum: 硬體載具序號(若為軟體手機載具, 此欄位填””) OTP1: 第一個 OTP 碼 OTP2: 第二個 OTP 碼

	sLogmsg: 保留參數 iFlags: 0 表示為軟體載具，1 表示為硬體載具
回傳值	第一個元素為錯誤代碼，0 為成功，其他值為失敗。 第二個元素為錯誤訊息。
參考	

程式碼的呼叫範例如下：

1、軟體載具

```
<%@ page import="com.formosoft.motp.stub.Constant"%>
<%@ page import="com.formosoft.motp.stub.MOTPFacade"%>
<%
String sUrl = "http://motp.xxx.com.tw/motp/MOTP";
String user = request.getParameter("userName");
String pass = DigestUtils.shaHex(request.getParameter("userPass"));
String sAccount = request.getParameter("sAccount");
long factMoving = Integer.parseInt(request.getParameter("factMoving"));
String OTP1 = request.getParameter("OTP1");
String OTP2 = request.getParameter("OTP2");
String fault[] =
MOTPFacade.FSMOTP_ReSync(sUrl,user,pass,sAccount,factMoving,"",OTP1,OTP2,"",0);
int code = Integer.parseInt(result[0]);
if(result[0].equals("0")){
    //成功，do something
}else{
    //失敗，do something
}
%>
```

2、硬體載具

```
<%@ page import="com.formosoft.motp.stub.Constant"%>
<%@ page import="com.formosoft.motp.stub.MOTPFacade"%>
<%
String sUrl = "http://motp.xxx.com.tw/motp/MOTP";
String user = request.getParameter("userName");
```

```
String pass = DigestUtils.shaHex(request.getParameter("userPass"));
String sAccount = request.getParameter("sAccount");
String sSerial = request.getParameter("sSerial");
String OTP1 = request.getParameter("OTP1");
String OTP2 = request.getParameter( "OTP2");
String fault[] = MOTPFacade.FSMOTP_ReSync(sUrl,user,pass,sAccount,0,
sSerial,OTP1,OTP2,"",1);
int code = Integer.parseInt( result[0]);
if(result[0].equals("0")){
    //成功，do something
}else{
    //失敗，do something
}
%>
```

(五)、練習

請使用上述範例，寫出一個 OTP 軟體 Token 驗證、註冊及同步的 JSP 測試網頁。

二、ASP 網頁

在 ASP 網頁使用 ActiveX 元件，請先安裝 MOTPFacadeATL.dll。將 MOTPFacadeATL.dll 存放在硬碟中，如：c:\，在該位置執行 regsvr32 MOTPFacadeATL.dll 指令，如圖所示。

```
C:\> regsvr32 MOTPFacadeATL.dll
```

圖 18 註冊 ActiveX DLL 元件

並在 ASP 的網頁中呼叫下列程式碼，用來呼叫 MOTP 的函式：

```
Set obj = Server.CreateObject("MOTPFACADEATL.FSMOTPATL")
```

(一)、OTP 驗證整合

由於一般的認證網頁都有原來的帳號密碼驗證機制，因此只需在原先密碼的下方新增一個 OTP 欄位，再呼叫 MOTP 的 API 即可。

在此節介紹兩隻驗證的 API，FSMOTP_AuthOTP 只單純的驗證 OTP，而 FSMOTP_AuthOTPex 則在驗證成功後，會回傳一個驗證碼，網頁設計者可以將此驗證碼顯示在登入成功的頁面上，若使用者的載具是有驗證碼機制的 OTP Token，則會同時顯示該驗證碼，此時使用者可比對兩個驗證碼是否相同，若不同即表示該網站可能為釣魚網站，使用者必須小心注意。

FSMOTP_AuthOTP	
函式宣告	int FSMOTP_AuthOTP(BSTR url, BSTR acc, BSTR otp);
說明	驗證 OTP
參數	url: MOTP 主機位置 acc: 使用者帳號 otp: 密碼

回傳值	0 為成功。 其他值為失敗，請呼叫 FSMOTP_GetErrorMsg()取得錯誤資訊
參考	

FSMOTP_AuthOTPEx	
函式宣告	int FSMOTP_AuthOTPEx(BSTR url, BSTR acc, BSTR otp);
說明	驗證 OTP，並回傳驗證碼，為雙向認證使用 。
參數	url: MOTP 主機位置 acc: 使用者帳號 otp: 密碼
回傳值	0 為成功。呼叫 FSMOTP_GetOTPAuthCode()取得驗證碼。 其他值為失敗，請呼叫 FSMOTP_GetErrorMsg()取得錯誤資訊
參考	

```
<%
'使用 ActiveX 物件(MOTPFACADEATL.FSMOTPATL)。
Set obj = Server.CreateObject("MOTPFACADEATL.FSMOTPATL")
'設定 MOTP 主機位址。
URL = "http://motp.xxx.com.tw/motp/MOTP"
username = request("username")
OTP = request("OTP")
'驗證 Username 與 OTP
result = obj.FSMOTP_AuthOTP(URL,username,OTP)
'result = obj.FSMOTP_AuthOTPEx(URL,username,OTP)
if result = 0 then
    '成功，do something
'MAC=obj.FSMOTP_GetOTPAuthCode()
'Show MAC on webpage
else
    '失敗，do something
end if
%>
```

(二)、新增 OTP 使用者

當一個使用者要開始用 OTP 來保護帳號時，在 MOTP 主機中必須先新增此帳號，再使用下一節「OTP 註冊」，讓使用者登記該 Token 的資訊。使用說明如下：

FSMOTP_CreateOTPUser	
函式宣告	int FSMOTP_CreateOTPUser(BSTR url, BSTR opacc, BSTR oppwd, BSTR acc, int device, BSTR profile, BSTR model, int lang, BSTR phone, BSTR email, BSTR email2, BSTR desc, BSTR logmsg, int flag);
說明	新增一個 OTP 使用者
參數	url: MOTP 主機位置 opacc: 管理者帳號 oppwd: 管理者密碼 acc: 使用者帳號 device: 載具類型(30:WM,50:JAVA,70:myPass, 90:Hardware) profile: 載具設定檔，(standard) model: 載具類型字串 lang: 語言(0: English, 1: Chinese) phone: 手機號碼(非必要欄位，使用簡訊下載通知時使用) email: 電子郵件帳號 email2: 保留參數 desc: 使用者描述 logmsg: 保留參數 flag: 特別參數選項，可使用相加 (1:強制新增, 2:線上註冊, 4:不寄 Email)
回傳值	0 為成功。FSMOTP_GetRegMsg(int lang) 其他值為失敗，請呼叫 FSMOTP_GetErrorMsg()取得錯誤資訊
參考	flag: 特別參數選項，可使用相加同時使用多種選項 1:強制新增, 不管帳號是否已存在。 2:線上註冊, 網頁上及 Email 中的註冊訊息將不顯示帳號及臨時密碼。 4:不寄 Email

範例如下：

```
<%
'使用 ActiveX 物件(MOTPFACADEATL.FSMOTPATL)。
Set obj = Server.CreateObject("MOTPFACADEATL.FSMOTPATL")
'設定 MOTP 主機位址。
url = "http://motp.xxx.com.tw/motp/MOTP"
```

```
opacc ="admin"
oppwd ="D033E22AE348AEB5660FC2140AEC35850C4DA997"
acc = request("username")
deviceType=50
profile=""
sEmail=request("sEmail")
'註冊 User 帳號與軟體 OTP 程式上所顯示的註冊資訊
result = obj. FSMOTP_CreateOTPUser(sUrl,user,pass,sAccount,deviceType,profile,
"",1,"",sEmail,"","","",0)
if result = 0 then
    '成功，do something
else
    '失敗，do something
end if
%>
```

(三)、OTP 註冊整合

因為本系統可同時接受硬體 Token 及軟體 Token 兩種載具，由於載具型態不同，註冊的程序也有些許不同。硬體載具的使用者只需將載具背面的序號填入該開通帳號即可；而軟體載具使用者，則必須先下載安裝手機程式，並在使用 MOTP 服務前，先將手機上 OTP 程式所顯示的資訊註冊到系統上，做類似「開卡」的動作，系統才能正確的運作，以避免其他非法使用者的惡意登入。因此註冊的呼叫方式分為軟體載具 FSMOTP_RegisterOTPUser 和硬體載具 FSMOTP_RegisterHwTokenUser 兩種：

FSMOTP_RegisterOTPUser	
函式宣告	int FSMOTP_RegisterOTPUser(BSTR url, BSTR opacc, BSTR oppwd, BSTR acc, int OTPId, BSTR SN, BSTR deviceID, BSTR count, BSTR logmsg, int flag);
說明	登記 OTP 使用者的註冊資訊
參數	url: MOTP 主機位置 opacc: 管理者帳號 oppwd: 管理者密碼 acc: 使用者帳號 OTPid: 載具代號(-1:自動尋找未註冊之載具) SN: 手機上顯示之 Token 序號

	deviceID: 載具的機器碼 count: OTP 計數器 logmsg: 保留參數 flag: 保留參數
回傳值	0 為成功。 其他值為失敗，請呼叫 FSMOTP_GetErrorMsg()取得錯誤資訊
參考	

FSMOTP_RegisterHwTokenUser	
函式宣告	int FSMOTP_RegisterHwTokenUser(BSTR url, BSTR opacc, BSTR oppwd, BSTR acc, BSTR serial, BSTR count, BSTR logmsg, int flag);
說明	登記硬體 Token 使用者的註冊資訊
參數	url: MOTP 主機位置 opacc: 管理者帳號 oppwd: 管理者密碼 acc: 使用者帳號 serial: 載具的硬體序號 count: 保留參數 logmsg: 保留參數 flag: 保留參數
回傳值	0 為成功。 其他值為失敗，請呼叫 FSMOTP_GetErrorMsg()取得錯誤資訊
參考	

程式碼的範例如下：

1、軟體載具

```
<%
'使用 ActiveX 物件(MOTPFACADEATL.FSMOTPATL)。
Set obj = Server.CreateObject("MOTPFACADEATL.FSMOTPATL")
'設定 MOTP 主機位址。
url = "http://motp.xxx.com.tw/motp/MOTP"
opacc ="admin"
oppwd ="D033E22AE348AEB5660FC2140AEC35850C4DA997"
acc = request("username")
```

```
serial=request("SN")
deviceID=request("deviceID")
count=request("count")
'註冊 User 帳號與軟體 OTP 程式上所顯示的註冊資訊
result =
obj.FSMOTP_RegisterOTPUser(url,opacc,oppwd,acc,-1,serial,deviceID,count,"log",0)
if result = 0 then
    '成功，do something
else
    '失敗，do something
end if
%>
```

2、硬體載具

```
<%
'使用 ActiveX 物件(MOTPFACADEATL.FSMOTPATL)。
Set obj = Server.CreateObject("MOTPFACADEATL.FSMOTPATL")
'設定 MOTP 主機位址。
url= "http://motp.xxx.com.tw/motp/MOTP"
opacc="admin"
oppwd ="D033E22AE348AEB5660FC2140AEC35850C4DA997"
username = request("username")
serial=request("SN")
'註冊 User 帳號與硬體 OTP 載具背面所印之序號
result = obj.FSMOTP_RegisterHwTokenUser(url,opacc,oppwd,acc,serial,0,"log",0)
if result = 0 then
    '成功，do something
else
    '失敗，do something
end if
%>
```

(四)、OTP 同步整合

當使用者發現輸入 OTP 多次都驗證失敗時，很有可能使用者的行動裝置已經出現未同步

的狀況，可透過此功能與系統作溝通同步。呼叫方式同樣分為軟體載具和硬體載具兩種：

FSMOTP_ReSync	
函式宣告	int FSMOTP_ReSync (BSTR url, BSTR opacc, BSTR oppwd, BSTR acc, BSTR devnum, BSTR OTP1, BSTR OTP2, BSTR logmsg, int flag);
說明	同步 OTP 載具的設定
參數	url: MOTP 主機位置 opacc: 管理者帳號 oppwd: 管理者密碼 acc: 使用者帳號 devnum: 硬體載具序號或 counter OTP1: 第一個 OTP 碼 OTP2: 第二個 OTP 碼 logmsg: 保留參數 flag: 0 表示(軟體)devnum 為 counter，1 表示(硬體)devnum 為硬體序號
回傳值	0 為成功。 其他值為失敗，請呼叫 FSMOTP_GetErrorMsg()取得錯誤資訊
參考	

程式碼的呼叫範例如下：

1、軟體載具

```
<%
'使用 ActiveX 物件(MOTPFACADEATL.FSMOTPATL)。
Set obj = Server.CreateObject("MOTPFACADEATL.FSMOTPATL")
'設定 MOTP 主機位址。
url = "http://motp.xxx.com.tw/motp/MOTP"
opacc ="admin"
oppwd ="D033E22AE348AEB5660FC2140AEC35850C4DA997"
acc = request("username")
OTP1=request("OTP1")
OTP2=request("OTP2")
count=request("count")
'註冊 User 帳號與軟體 OTP 程式上所顯示的註冊資訊
result = obj.FSMOTP_ReSync(url,opacc,oppwd,acc,count,OTP1,OTP2,"log",0)
```

```
if result = 0 then
    '成功，do something
else
    '失敗，do something
end if
%>
```

2、硬體載具

```
<%
'使用 ActiveX 物件(MOTPFACADEATL.FSMOTPATL)。
Set obj = Server.CreateObject("MOTPFACADEATL.FSMOTPATL")
'設定 MOTP 主機位址。
url = "http://motp.xxx.com.tw/motp/MOTP"
opacc ="admin"
oppwd ="D033E22AE348AEB5660FC2140AEC35850C4DA997"
acc = request("username")
serial=request("SN")
OTP1=request("OTP1")
OTP2=request("OTP2")
'註冊 User 帳號與軟體 OTP 程式上所顯示的註冊資訊
result = obj.FSMOTP_ReSync(url,opacc,oppwd,acc,serial,OTP1,OTP2,"log",1)
if result = 0 then
    '成功，do something
else
    '失敗，do something
end if
%>
```

(五)、練習

請使用上述範例，寫出一個 OTP 硬體 Token 驗證、註冊及同步的 ASP 測試網頁。

三、PHP 網頁

PHP 為一種 CGI 的網頁應用程式語言，以下使用 Apache 為伺服器當範例。

使用者必須先安裝 Apache 網頁伺服器，再安裝 PHP 的安裝套件。

在整合網頁前，必須先做好環境設定，操作步驟如下：

將 **MOTPFacade.php** 和 **MOTPErrMsg.php** 兩個檔案放在呼叫網頁所在目錄的 includes 子目錄中。

PHP 網頁必須在網頁中匯入所需元件，設定內容如下：

```
<?php
    include_once("includes/MOTPFacade.php");
    include_once("includes/MOTPErrMsg.php");
?>
```

(一)、OTP 驗證整合

由於一般的認證網頁都有原來的帳號密碼驗證機制，因此只需在原先密碼的下方新增一個 OTP 欄位，再呼叫 MOTP 的 API 即可。

在此節介紹兩隻驗證的 API，FSMOTP_AuthOTP 只單純的驗證 OTP，而 FSMOTP_AuthOTPex 則在驗證成功後，會回傳一個驗證碼，網頁設計者可以將此驗證碼顯示在登入成功的頁面上，若使用者的載具是有驗證碼機制的 OTP Token，則會同時顯示該驗證碼，此時使用者可比對兩個驗證碼是否相同，若不同即表示該網站可能為釣魚網站，使用者必須小心注意。

FSMOTP_AuthOTP	
函式宣告	string[] FSMOTP_AuthOTP(string sUrl, string sAccount, string sOTP, string sLogmsg, int iFlags);
說明	驗證 OTP
參數	sUrl: MOTP 主機位置 sAccount: 使用者帳號 sOTP: 密碼 sLogmsg: 保留參數

	iFlags: 保留參數
回傳值	第一個元素為錯誤代碼，0 為成功，其他值為失敗。 第二個元素為錯誤訊息。
參考	

FSMOTP_AuthOTPex	
函式宣告	string [] FSMOTP_AuthOTPEx(string sUrl, string sAccount, string sOTP, string sLogmsg, int iFlags);
說明	驗證 OTP，並回傳驗證碼，為雙向認證使用。
參數	sUrl: MOTP 主機位置 sAccount: 使用者帳號 sOTP: 密碼 sLogmsg: 保留參數 iFlags: 保留參數
回傳值	第一個元素為錯誤代碼，0 為成功，其他值為失敗。 第二個元素為錯誤訊息。 若為成功，第三個元素為驗證碼。
參考	

在驗證 OTP 的網頁上加入以下程式碼：

```
<?php
$sUrl = "http://motp.xxx.com.tw/motp/MOTP";
$sAccount = $_POST['sAccount'];
$otp = $_POST['otp'];
include_once("includes/MOTPFacade.php");
include_once("includes/MOTPErrMsg.php");
$motp = new MOTPFacade();
$rtns = $motp->FSMOTP_AuthOTP($sUrl, $sAccount, $otp, "", 0);
//$rtns = $motp->FSMOTP_AuthOTPEx($sUrl, $sAccount, $otp, "", 0);
if (!is_array($rtns)) {
    //nothing happen
} else if ($rtns[0] == SERVER_RTN_SUCCESS) {
    //success
//$msg = $rtns[2]; //雙向驗證碼
```

```
    header("Location: success.html");
    return;
} else {
    //print error
    $msg = $rtns[1];
    $msg = "<br><font color=red size='3'>[{$rtns[0]}] {$msg}</font>";
}
?>
```

(二)、新增 OTP 使用者

當一個使用者要開始用 OTP 來保護帳號時，在 MOTP 主機中必須先新增此帳號，再使用下一節「OTP 註冊」，讓使用者登記該 Token 的資訊。使用說明如下：

FSMOTP_CreateOTPUser	
函式宣告	string[] FSMOTP_CreateOTPUser(string sUrl, string opAcct, string opPwd, string sAccount, int deviceType, string brand, string model, int clientLang, string mobileNumber, string eMail1, string eMail2, string sDesc, string sLogmsg, int iFlags);
說明	新增一個 OTP 使用者
參數	sUrl: MOTP 主機位置 opAcct: 管理者帳號 opPwd: 管理者密碼 sAccount: 使用者帳號 deviceType: 載具類型 (30:Windows Phone,50:JAVA Phone,70:myPass,90:Hardware) brand: 載具設定檔，(standard) model: 載具類型字串 clientLang: 語言(0:English, 1:中文) mobileNumber: 手機號碼(非必要欄位，使用簡訊下載通知時使用) eMail1: 電子郵件帳號 eMail2: 保留參數 sDesc: 使用者描述 sLogmsg: 保留參數 iFlags: 特別參數選項，可使用相加

	(1:強制新增, 2:線上註冊, 4:不寄 Email)
回傳值	第一個元素為錯誤代碼，0 為成功，其他值為失敗。 第二個元素為錯誤訊息。
參考	iFlags: 特別參數選項，可使用相加同時使用多種選項 1:強制新增, 不管帳號是否已存在。 2:線上註冊, 網頁上及 Email 中的註冊訊息將不顯示帳號及臨時密碼。 4:不寄 Email

範例如下：

```
<?php
define("sUrl", "http://motp.xxx.com.tw/motp/MOTP");
//Account
define("opAcct", "admin");
//Password
define("opPwd", "D033E22AE348AEB5660FC2140AEC35850C4DA997");
$sAccount = $_POST['sAccount'];
$eMail1 = $_POST['eMail1'];
$deviceType = 70;

include_once("includes/MOTPFacade.php");
include_once("includes/MOTPErrMsg.php");
$motp = new MOTPFacade();
$rtns = $motp->FSMOTP_CreateOTPUser(sUrl, opAcct, opPwd, $sAccount, $deviceType, "",
"", 0, "", $eMail1, "", "", "", 4);
if (!is_array($rtns)) {
    //pass do nothing
} else if ($rtns[0] == SERVER_RTN_SUCCESS) {
    //success
    header("Location: FSMOTP_AuthOTP.php?sAccount=".$sAccount);
    return;
} else {
    $msg = $rtns[1];
    $msg = "<br><font color=red size='3'>[{$rtns[0]}] {$msg}</font>";
}
?>
```

(三)、OTP 註冊整合

因為本系統可同時接受硬體 Token 及軟體 Token 兩種載具，由於載具型態不同，註冊的程序也有些許不同。硬體載具的使用者只需將載具背面的序號填入該開通帳號即可；而軟體載具使用者，則必須先下載安裝手機程式，並在使用 MOTP 服務前，先將手機上 OTP 程式所顯示的資訊註冊到系統上，做類似「開卡」的動作，系統才能正確的運作，以避免其他非法使用者的惡意登入。因此註冊的呼叫方式分為軟體載具 FSMOTP_RegisterOTPUser 和硬體載具 FSMOTP_RegisterHwTokenUser 兩種：

FSMOTP_RegisterOTPUser	
函式宣告	string[] FSMOTP_RegisterOTPUser(string sUrl, string opAcct, string opPwd, string sAccount, int iHotpId, string sSerial, string sDeviceID, string factMoving, string sLogmsg, int iFlags);
說明	登記 OTP 使用者的註冊資訊
參數	sUrl: MOTP 主機位置 opAcct: 管理者帳號 opPwd: 管理者密碼 sAccount: 使用者帳號 iHotpId: 載具代號(-1:自動尋找未註冊之載具) sSerial: 手機上顯示之 Token 序號 sDeviceID: 手機載具的機器碼 factMoving: OTP 計數器 sLogmsg: 保留參數 iFlags: 保留參數
回傳值	第一個元素為錯誤代碼，0 為成功，其他值為失敗。 第二個元素為錯誤訊息。
參考	

FSMOTP_RegisterHwTokenUser	
函式宣告	string [] FSMOTP_RegisterHwTokenUser(string sUrl, string opAcct, string opPwd, string sAccount, string sSerial, string timeCounter, string sLogmsg, int iFlags);
說明	登記硬體 Token 使用者的註冊資訊

參數	sUrl: MOTP 主機位置 opAcct: 管理者帳號 opPwd: 管理者密碼 sAccount: 使用者帳號 sSerial: 載具的 Token 序號 timeCounter: 保留參數 sLogmsg: 保留參數 iFlags: 保留參數
回傳值	第一個元素為錯誤代碼，0 為成功，其他值為失敗。 第二個元素為錯誤訊息。
參考	

程式碼的範例如下：

1、軟體載具

```
<?php
define("sUrl", "http://motp.xxx.com.tw/motp/MOTP");
//Account
define("opAcct", "admin");
//Password
define("opPwd", "D033E22AE348AEB5660FC2140AEC35850C4DA997");
$sAccount = $_POST['sAccount'];
$sSerial = $_POST['sSerial'];
$sDeviceID =$_POST['sDeviceID'];
$factMoving = $_POST['factMoving'];

include_once("includes/MOTPFacade.php");
include_once("includes/MOTPErrMsg.php");
$motp = new MOTPFacade();
$rtns = $motp->FSMOTP_RegisterOTPUser(sUrl, opAcct, opPwd, $sAccount, -1, $sSerial,
$sDeviceID, $factMoving, 0);
if (!is_array($rtns)) {
    //pass do nothing
} else if ($rtns[0] == SERVER_RTN_SUCCESS) {
    //success
    header("Location: FSMOTP_AuthOTP.php?sAccount=".$sAccount);
```

```
    return;
} else {
    $msg = $rtns[1];
    $msg = "<br><font color=red size='3'>[{$rtns[0]}] {$msg}</font>";
}
?>
```

2、硬體載具

```
<?php
define("sUrl", "http://motp.xxx.com.tw/motp/MOTP");
define("opAcct", "admin");//Account
define("opPwd", "D033E22AE348AEB5660FC2140AEC35850C4DA997"); //Password
$sAccount = $_POST['sAccount'];
$sSerial = $_POST['sSerial'];
$sDeviceID =$_POST['sDeviceID'];
$factMoving = $_POST['factMoving'];

include_once("includes/MOTPFacade.php");
include_once("includes/MOTPErrMsg.php");
$motp = new MOTPFacade();
$rtns = $motp->FSMOTP_RegisterHwTokenUser(sUrl, opAcct, opPwd, $sAccount, $sSerial,
0, 0);
if (!is_array($rtns)) {
    //pass do nothing
} else if ($rtns[0] == SERVER_RTN_SUCCESS) {
    //success
    header("Location: FSMOTP_AuthOTP.php?sAccount=".$sAccount);
    return;
} else {
    $msg = $rtns[1];
    $msg = "<br><font color=red size='3'>[{$rtns[0]}] {$msg}</font>";
}
?>
```

(四)、OTP 同步整合

當使用者發現輸入 OTP 多次都驗證失敗時，很有可能使用者的行動裝置已經出現未同步的狀況，可透過此功能與系統作溝通同步。呼叫方式同樣分為軟體載具和硬體載具兩種：

FSMOTP_ReSync	
函式宣告	string[] FSMOTP_ReSync(string sUrl, string opAcct, string opPwd, string sAccount, string sTokenNum, string OTP1, string OTP2, int iFlags);
說明	同步 OTP 載具的設定
參數	sUrl: MOTP 主機位置 opAcct: 管理者帳號 opPwd: 管理者密碼 sAccount: 使用者帳號 sTokenNum: 若 iFlags 為 1，此欄位為硬體載具序號；若 iFlags 為 0，則此欄位為軟體載具 counter OTP1: 第一個 OTP 碼 OTP2: 第二個 OTP 碼 iFlags: 0 表示為軟體載具，1 表示為硬體載具
回傳值	第一個元素為錯誤代碼，0 為成功，其他值為失敗。 第二個元素為錯誤訊息。
參考	

程式碼的呼叫範例如下：

```
<?php
define("sUrl", "http://motp.xxx.com.tw/motp/MOTP");
define("opAcct", "admin");//Account
define("opPwd", "D033E22AE348AEB5660FC2140AEC35850C4DA997"); //Password
$sAccount = $_POST['sAccount'];
$sOTP1 = $_POST['sOTP1'];
$sOTP2 =$_POST['sOTP2'];
$factMoving = $_POST['factMoving'];
//$sSerial=$_POST['sSerial']; //for 硬體載具同步

include_once("includes/MOTPFacade.php");
include_once("includes/MOTPErrMsg.php");
$motp = new MOTPFacade();
$rtns = $motp->FSMOTP_ReSync(sUrl, opAcct, opPwd, $sAccount, $factMoving, $sOTP1,
$sOTP2, 0);
```

```
//$rtns = $motp->FSMOTP_ReSync(sUrl, opAcct, opPwd, $sAccount, $sSerial, $sOTP1,
$sOTP2, 1); //for 硬體載具同步

if (!is_array($rtns)) {
    //pass do nothing
} else if ($rtns[0] == SERVER_RTN_SUCCESS) {
    //success
    header("Location: FSMOTP_AuthOTP.php?sAccount=".$sAccount);
    return;
} else {
    $msg = $rtns[1];
    $msg = "<br><font color=red size='3'>[{$rtns[0]}] {$msg}</font>";
}
?>
```

(五)、練習

1. 請使用上述範例，寫出一個 OTP 軟體 Token 驗證、註冊及同步的 PHP 測試網頁。

2. 請使用上述範例，寫出一個 OTP 硬體 Token 驗證、註冊及同步的 PHP 測試網頁。

肆、加值應用

一、網路銀行應用

在網路銀行的應用中，通常使用者必須親自到銀行臨櫃申請使用憑證或 OTP。開通帳號的介面通常為銀行行員所操作，因此，在登記使用者的流程通常在臨櫃申請時，即完成使用者註冊的程序。在本例中以實體 Token 為主要實作目標，而本應用須實作以下幾個網頁：

1. 註冊網頁：銀行行員代替使用者註冊其銀行帳號及 Token 序號，並發給使用者該 Token。
2. OTP 登入網頁：當使用者要登入網路銀行時，必須輸入使用者名稱、密碼及 OTP。
3. 同步網頁：當使用者發生 Token 無法登入時，使用者告知行員，該使用者的帳號、Token 序號以及連續的兩個 OTP 值，OTP1 及 OTP2，由銀行代為同步。
4. 網路銀行登入成功的模擬網頁。

(一)、行員代註冊網頁

由於此範例為註冊硬體載具，此行員代註冊網頁將一次做完新增使用者及註冊的功能。範例如下：

```
<%@ page import="com.formosoft.motp.stub.Constant"%>
<%@ page import="com.formosoft.motp.stub.MOTPFacade"%>
<%
String sUrl= "http://motp.XXX.com.tw/motp/MOTP";
String msg = "";
boolean regresult=false;
String opAccount=request.getParameter("opAccount");
String opPasswd=request.getParameter("opPasswd");
String sAccount = request.getParameter("sAccount");
String sSerial = request.getParameter("sSerial");
if(sAccount!=null && sSerial!=null){
    String fault[] =
```

```
MOTPFacade.FSMOTP_CreateOTPUser(sUrl,opAccount,opPasswd,sAccount,90,"standard","
",0,"",sAccount+"@abc.tw",null,"add user from api","",3);
    if(fault!=null){
        if(fault[0].equals("0")){
            fault = MOTPFacade.FSMOTP_RegisterHwTokenUser(sUrl,opAccount,
opPasswd,sAccount,sSerial,0,"",0);
            if("0".equals(fault[0])){
                regresult=true;
            }
        }
    }
    msg += "【回傳代碼】" + fault[0] + "<br>";
    msg += "【回傳訊息】" + fault[1] + "<br>";
}
if(regresult){%>
<html>
<head>
   <meta http-equiv="refresh" content="10;url=FS_BankLogin.jsp" />
</head>
<body>
    <img src="images/top_1.gif">
    使用者註冊成功，在幾秒後將導向使用者登入畫面，可在該畫面測試密碼的正確性。
    或直接點選<a href="FS_BankLogin.jsp">登入網頁</a>
</body>
</html>
<% }else{%>
<html>
<title>全景網路銀行 OTP 使用者註冊</title>
<body>
    <img src="images/top_1.gif">
    <table>
        <tr>
            <td>
                <img src="images/left1.jpg">
            </td>
            <td valign="top">
            <img src="images/register.gif">
```

```
            <form method="post">
            行員帳號 : <input type="text" name="opAccount"><br>
            行員密碼 : <input type="password" name="opPasswd"><br>
            客戶帳號 : <input type="text" name="sAccount"><br>
            載具序號 : <input type="text" name="sSer ial"><br>
            <input type="submit" name="Submit">
            </form>
            <%=msg%>
            </td>
        </tr>
    </table>
</body>
</html>
<%}%>
```

(二)、銀行 OTP 登入網頁

```
<%@ page import="com.formosoft.motp.stub.Constant"%>
<%@ page import="com.formosoft.motp.stub.MOTPFacade"%>
<%
String sUrl= "http://motp.XXX.com.tw/motp/MOTP";
boolean login = false;
String msg = "";
if(request.getParameter("Submit")!=null){
    String id = request.getParameter("sAccount");
    String myotp = request.getParameter("sOTP");
    String pass = request.getParameter("sPass");
    msg=request.getParameter("msg");
    if("admin".equals(id)&&"admin".equals(pass)){//進入管理者介面
        response.sendRedirect("FS_BankRegister.jsp");
    }else if("111111".equals(pass)){//使用者的密碼固定爲 111111, 真實的系統必須檢查 DB
        try{
            String fault[] = MOTPFacade.FSMOTP_AuthOTPex(sUrl, id, myotp, "", 0);
            if(fault!=null){
                if(fault[0].equals("0")){//驗證成功進入內頁
                    login = true;
                }else{
```

```
                    msg = fault[0] + " - "+
Constant.getErrMsg(Integer.parseInt(fault[0]),Constant.LANG_TRAD_CHINESE);
                }
            }
        }catch(Exception e){
            msg = "[Exception] " + e.toString();
        }
    }else{
        msg="帳號或密碼錯誤!!";
    }
}
%>
<html>
<title>全景網路銀行首頁</title>
<body>
<%if(!login){ %>
    <img src="images/top_1.gif">
    <table>
        <tr>
            <td>
                <img src="images/left1.jpg">
            </td>
            <td valign="top">
            <img src="images/login.gif">
            <form method="post">
            帳號 : <input type="text" name="sAccount"><br>
            密碼 : <input type="password" name="sPass"><br>
            OTP  : <input type="password" name="sOTP"><br>
            <input type="submit" name="Submit">
            </form>
            <%=msg%>
            </td>
        </tr>
    </table>
<%} else {
    response.sendRedirect("ATM.htm");
} %>
```

```
</body>
</html>
```

(三)、行員代同步網頁

```
<%@ page import="com.formosoft.motp.stub.Constant"%>
<%@ page import="com.formosoft.motp.stub.MOTPFacade"%>
<%
String sUrl= "http://motp.XXX.com.tw/motp/MOTP";
String msg = "";
boolean regresult=false;
String opAccount=request.getParameter("opAccount");
String opPasswd=request.getParameter("opPasswd");
String sAccount = request.getParameter("sAccount");
String sSerial = request.getParameter("sSerial");
String OTP1=request.getParameter("sOTP1");
String OTP2=request.getParameter("sOTP2");
if(sAccount!=null && sSerial!=null){
    String fault[] = MOTPFacade.FSMOTP_ReSync(sUrl,opAccount,opPasswd,sAccount,0,
sSerial,OTP1,OTP2,"",1);
    if(fault!=null)&& fault[0].equals("0")){
        regresult=true;
    }
    msg += "【回傳代碼】" + fault[0] + "<br>";
    msg += "【回傳訊息】" + fault[1] + "<br>";
}
if(regresult){%>
<html>
<head>
   <meta http-equiv="refresh" content="10;url=FS_BankLogin.jsp" />
</head>
<body>
    <img src="images/top_1.gif">
    使用者同步成功，在幾秒後將導向使用者登入網頁，可在該網頁測試 OTP 的正確性。
    或直接點選<a href="FS_BankLogin.jsp">登入網頁</a>
</body>
</html>
```

```
<% }else{%>
<html>
<title>全景網路銀行 OTP 使用者同步</title>
<body>
    <img src="images/top_1.gif">
    <table>
        <tr>
            <td>
                <img src="images/left1.jpg">
            </td>
            <td valign="top">
            <img src="images/resync.gif">
            <form method="post">
            行員帳號 : <input type="text" name="opAccount"><br>
            行員密碼 : <input type="password" name="opPasswd"><br>
            客戶帳號 : <input type="text" name="sAccount"><br>
            載具序號 : <input type="text" name="sSerial"><br>
            OTP1 : <input type="text" name="sOTP1"><br>
            OTP2 : <input type="text" name="sOTP2"><br>
            <input type="submit" name="Submit">
            </form>
            OTP1 及 OTP2 爲使用者在 OTP 載具連續按兩次所產生的 OTP 值
            <%=msg%>
            </td>
        </tr>
    </table>
</body>
</html>
<%}%>
```

(四)、練習

請將上述個別網頁串起來模擬一個真實的網路銀行認證流程 。

二、線上遊戲登入應用練習

由於線上遊戲的玩家眾多，一般的遊戲也不像銀行有實體店面，而遊戲的風險也比銀行來得低。因此，在此節應用的範例中，以軟體載具為開發目標，讓玩家可以下載手機的 OTP 程式，並且直接在遊戲的網站上開通 OTP 帳號，就連同步網頁也一併做在玩家的個人帳號下。一方面方便玩家操作，二方面可降低成本，對遊戲玩家來說，成本是很重要的考量因素。

此系統一共有 4 個主要網頁，玩家使用原帳號密碼登入，玩家下載 OTP 程式到手機中，玩家啟用 OTP 機制，玩家輸入 OTP 獲得 10 分鐘的遊戲登入權限。

(一)、初始登入

此網頁模擬一般遊戲網的登入模式。

```
<%@ page import="com.formosoft.motp.stub.Constant"%>
<%@ page import="com.formosoft.motp.stub.MOTPFacade"%>
<%
String sUrl= "http://motp.XXX.com.tw/motp/MOTP";
String message = "";
String errorName = "";
if(request.getParameter("doLogin")!=null){
    String name = request.getParameter("name");
    String pass = request.getParameter("pass");
    if("111111".equals(pass)){//使用者的密碼固定爲 111111, 真實的系統必須檢查 DB
        session.setAttribute("OTPUser", name);
        response.sendRedirect("FS_GameStart.jsp");
        return;
    }else{
        message = "密碼有誤，請重新輸入！";
    }
    errorName = name;
}else{
    session.removeAttribute("OTPUser");
    session.removeAttribute("OTPinitalKey");
}
%>

<HTML>
```

```
<HEAD><TITLE>全景遊戲網  - Formosoft International Inc. </TITLE></HEAD>
<BODY>
<TABLE>
  <TR><TD colSpan=2 background="images/index_bg.gif">
        <IMG height=95 alt="" src="images/index_fs.gif" useMap=#Map border=0>
    </TD></TR>
  <TR><TD valign=top bgColor=#ffffff>
    <FORM method=post>
     <IMG src="images/index_14.gif"><br>
     <IMG src="images/index_25.gif" border=0><INPUT type=text name=name
value="<%=errorName%>"><br>
     <IMG src="images/index_28.gif" border=0><INPUT type=password name=pass><br>
     <INPUT type=submit value=" 登 入 " border=0 name="doLogin"><br>
     <%= message %>
     <A href="FS_GameStart.jsp"><IMG height=77 src="images/set123_02.gif"
width=162 border=0></A><br>
     <A href="FS_GameRegister.jsp"><IMG height=71 src="images/set123_03.gif"
width=162 border=0></A><br>
     <A href="FS_GameVerify.jsp"><I MG height=61 src="images/set123_04.gif"
width=162 border=0></A><bR>
    </FROM>
   </TD>
   <TD>
     <TABLE cellSpacing=0 cellPadding=0 width=558 border=0>
       <TR><TD vAlign=top width=558 height=42>
         <IMG src="images/teach_03.gif">
       </TD></TR>
       <TR><TD>
         <b>下載安裝</b><br>
             依據行動裝置類型，選擇正確的安裝程式。下載執行安裝成功後，需至官方網站中進行啓用與
註冊動作。完成後，才可使用 OTP 來登入遊戲。<br><br>
         <b>免費註冊</b><br>
             執行手機中的 MOTP 程式,第一次使用時必須輸入初始金鑰,此金鑰會在帳號啓用時取得。<br>
             驗證成功後，程式才會進入註冊資訊頁面(如下圖)，將各欄位輸入到註冊網頁中。
             <br><img src="images/Client_Register.gif"><br><br>
         <b>遊戲 GO</b><br>
             進入遊戲後，在 OTP 欄位輸入所顯示之訊息進行驗證動作 ，如圖所示。<br>
```

```
            <img border="0" src="images/PPC1.jpg"><br><br>
        <b>手機說明</b><br>
            若手機爲觸控式螢幕時，可直接點擊或雙擊圖示。另外可使用手機的方向鍵及左右鍵或者數字鍵來操作系統，如下圖所示:
            <br><img src="images/mobile_tw.jpg"><br><br>
        </TD></TR>
        <TR><TD><a href="http://www.formosoft.com.tw/" target="_blank">全景軟體股份有限公司</a>
              300 新竹科學園區園區二路 50 號 2 樓  客服專線:03-5630200#8
            <br>Copyrightc 2008 Formosoft International Inc. All Rights Reserved.
      </TD></TR>
    </TABLE>
    </TD></TR>
</TABLE> 
</BODY></HTML>
```

(二)、下載 OTP 程式並啟用 MOTP 機制

此網頁讓玩家自行下載手機程式，分為 JAVA 及 WM 兩個版本，玩家下載程式並安裝完畢後，可按下啟用鈕，進行註冊 OTP 功能。

```
<%@ page import="com.formosoft.motp.stub.Constant"%>
<%@ page import="com.formosoft.motp.stub.MOTPFacade"%>
<%
String sUrl= "http://motp.XXX.com.tw/motp/MOTP";
String name = (String)session.getAttribute("OTPUser");
String fault[] = null;
String message = "";
String doType = request.getParameter("doType");
if(doType!=null){
    if(doType.equals("OTPStart")){
        fault =
MOTPFacade.FSMOTP_CreateOTPUserCode(sUrl,opAcct,opPwd,name,"user@formosoft.com",
null,"Game User","",Constant.FLAG_CREATE_FORCIBLY);
        if(!fault[0].equals("0")){
            message = "【錯誤】" + fault[0] + " - " + fault[1];
```

```
        }else{
            session.setAttribute("OTPinitalKey", fault[1]);
            response.sendRedirect("FS_GameRegister.jsp");
            return;
        }
    }
}
%>
<HTML>
<HEAD><TITLE>全景遊戲網  - Formosoft International Inc. </TITLE></HEAD>
<BODY>
<TABLE>
  <TR><TD colSpan=2 background="images/index_bg.gif">
    <IMG height=95 alt="" src="images/index_fs.gif" useMap=#Map border=0>
  <TD></TR>
  <TR>
    <TD valign=top bgColor=#ffffff>
    <IMG src="images/index_14.gif"><br>
    <% if(session.getAttribute("OTPUser")==null){
           response.sendRedirect("FS_GameLogin.jsp");
          return;
       } else { %>
          <br><b>Hi! <%=session.getAttribute("OTPUser")%></b>
          <br><input type="button" value=" 登 出 "
onclick="location.href='FS_GameLogin.jsp'"><br><br>
     <% } %>
     <A href="FS_GameStart.jsp"><IMG height=77 src="images/set123_02.gif"
width=162 border=0></A><br>
     <A href="FS_GameRegister.jsp"><IMG height=71 src="images/set123_03.gif"
width=162 border=0></A><br>
     <A href="FS_GameVerify.jsp"><IMG height=61 src="images/set123_04.gif"
width=162 border=0></A><bR>
   </TD>
   <TD>  <!-- OTPStart START -->
    <br>請先依據行動裝置類型，選擇正確的安裝程式，下載執行成功後，再進行啓用動作:
    <ul>
      <li><b>Java Phone</b><br>
```

```
      方式一: 手機上網 - 在手機上輸入以下連結，並下載安裝。<br>
      <a href="<%=webURL%>FSMOTPJM_C.htm"><%=webURL%>FSMOTPJM_C.htm</a><br>
      方式二: 傳輸線、藍牙、紅外線 - 下載以下檔案，傳輸到手機中，執行安裝程序。<br>
      <a href="<%=webURL%>FSMOTPJM_C.jar"><%=webURL%>FSMOTPJM_C.jar</a><br>
      <a href="<%=webURL%>FSMOTPJM_C.jad"><%=webURL%>FSMOTPJM_C.jad</a><br>
      </li>
      <li><b>Windows Phone</b><br>
      方式一: 網路安裝 - 在手機上輸入以下連結，並下載安裝。<br>
      <a href="<%=webURL%>FSMOTPWM_Cppc.htm"><%=webURL%>FSMOTPWM_Cppc.htm</a><br>
      方式二: 傳輸線 - 下載檔案後，透過手機和 PC 的連線，在 PC 上解壓縮並執行 MOTPInstall.exe。
<br>
      <a href="<%=webURL%>FSMOTPWM_Cppc.zip"><%=webURL%>FSMOTPWM_Cppc.zip</a><br>
    </li></ul>
    <form>
        <input name="doType" type="hidden" value="OTPStart">
        <input name="doSubmit" type="submit" value=" 啓 用 ">
    </form>
    <%=message%>
    <!-- OTPStart END -->
  </TD></TR>
</TABLE> 
</BODY></HTML>
```

(三)、玩家註冊手機

玩家將手機程式上的註冊訊息，填寫到註冊網頁中，即完成 MOTP 註冊步驟。可開始使用 MOTP 來開通遊戲了。

```
<%@ page import="com.formosoft.motp.stub.Constant"%>
<%@ page import="com.formosoft.motp.stub.MOTPFacade"%>
<%
String sUrl= "http://motp.XXX.com.tw/motp/MOTP";
if(session.getAttribute("OTPinitalKey")==null){
    response.sendRedirect("FS_GameStart.jsp");
    return;
}
String message = "";
```

```
if(request.getParameter("doSubmit")!=null) {
    String sSerial = request.getParameter("sSerial");
    String sDeviceID = request.getParameter("sFact1") +
request.getParameter("sFact2") + request.getParameter("sFact3") +
request.getParameter("sFact4");
    String fault[] =
MOTPFacade.FSMOTP_RegisterOTPUser(sUrl,opAcct,opPwd,(String)session.getAttribute
("OTPUser"),-1,sSerial,sDeviceID,0,"",0);
    if(fault[0].equals("0")){
        response.sendRedirect("FS_GameVerify.jsp");
        return;
    }else{
        message = "【錯誤】" + fault[0] + " - " + fault[1];
    }
}
%>
<HTML>
<HEAD><TITLE>全景遊戲網  - Formosoft International Inc. </TITLE></HEAD>
<BODY>
<TABLE>
  <TR><TD colSpan=2 background="images/index_bg.gif">
    <IMG height=95 alt="" src="images/index_fs.gif" useMap=#Map border=0>
  <TD></TR>
  <TR>
   <TD valign=top bgColor=#ffffff>
    <IMG src="images/index_14.gif"><br>
    <% if(session.getAttribute("OTPUser")==null){
          response.sendRedirect("FS_GameLogin.jsp");
          return;
       } else { %>
          <br><b>Hi! <%=session.getAttribute("OTPUser")%></b>
          <br><input type="button" value=" 登 出 "
onclick="location.href='FS_GameLogin.jsp'"> <br><br>
     <% } %>
     <A href="FS_GameStart.jsp"><IMG height=77 src="images/set123_02. gif"
width=162 border=0></A><br>
     <A href="FS_GameRegister.jsp"><IMG height=71 src="images/set123_03.gif"
```

```
width=162 border=0></A><br>
     <A href="FS_GameVerify.jsp"><IMG height=61 src="images/set123_04.gif"
width=162 border=0></A><bR>
    </TD>
    <TD> <!-- OTPRegister START -->
         請輸入「註冊資訊」畫面所顯示之訊息進行註冊動作。<br>
         若手機爲觸控式螢幕時，可直接點擊或雙擊圖示。<br>
         另外可使用手機的方向鍵及左右鍵或者數字鍵來操作系統:<br>
         1及5:執行功能, 3:換另一頁, 4:左移選項, 6:右移選項, 9:離開程式<br>
         <form>
             <b>初始金鑰 :</b><%= session.getAttribute("OTPinitalKey") %><br>
             註冊碼 :<input type="text" name="sSerial"><br>
             硬體序號  :<input type="text" name="sFact1" size="5">   -
                 <input type="text" name="sFact2" maxlength="5" size="5">  -
                 <input type="text" name="sFact3" maxlength="5" size="5">  -
                 <input type="text" name="sFact4" maxlength="5" size="5"><br>
             <input name="doSubmit" type="submit" value=" 註 冊 "><br><br>
         </form>
         <%= message %><br>
        <!-- OTPRegister END -->
    </TD></TR>
</TABLE> 
</BODY></HTML>
```

(四)、驗證 OTP 開通遊戲

此網頁主要功能為使用者輸入要登入遊戲的 OTP，然後將 OTP 送到主機驗證。當玩家輸入正確的 OTP 後，系統將玩家的遊戲帳號開放 10 分鐘的登入時間，玩家

```
<%@ page import="com.formosoft.motp.stub.Constant"%>
<%@ page import="com.formosoft.motp.stub.MOTPFacade"%>
<%
String sUrl= "http://motp.XXX.com.tw/motp/MOTP";
String message = "";
if(request.getParameter("sOTP")!=null) {
    String sOTP = request.getParameter("sOTP");
    String fault[] =
```

```
MOTPFacade.FSMOTP_AuthOTPex(sUrl,(String)session.getAttribute("OTPUser"),sOTP,""
,0);
    if(fault[0].equals("0")){
        message = "OTP 驗證成功，您可在十分鐘之內登入遊戲，超過 10 分鐘後，遊戲帳號將再度進入
鎖定狀態!";
        if(fault.length>2) message += " ( 驗證碼: " + fault[2] + " )";
    }else{
        message = "驗證 OTP 失敗，請確認您的資料是否正確!";
    }
}
%>
<HTML>
<HEAD><TITLE>全景遊戲網  - Formosoft International Inc. </TITLE></HEAD>
<BODY>
<TABLE>
  <TR><TD colSpan=2 background="images/index_bg.gif">
    <IMG height=95 alt="" src="images/index_fs.gif" useMap=#Map border=0>
  </TD></TR>
  <TR>
    <TD valign=top bgColor=#ffffff>
    <% if(session.getAttribute("OTPUser")==null){
           response.sendRedirect("FS_GameLogin.jsp");
          return;
       } else { %>
          <br>
          <IMG src="images/index_14.gif"><br>
           <b>Hi! <%=session.getAttribute("OTPUser")%></b>
          <br><input type="button" value=" 登 出 "
onclick="location.href='FS_GameLogin.jsp'"><br ><br>
     <% } %>
     <A href="FS_GameStart.jsp"><IMG height=77 src="images/set123_02.gif"
width=162 border=0></A><br>
     <A href="FS_GameRegister.jsp"><IMG height=71 src="images/set123_03.gif"
width=162 border=0></A><br>
     <A href="FS_GameVerify.jsp"><IMG height=61 src="images/set123_04.gif"
width=162 border=0></A><bR>
   </TD>
```

```
    <TD>
        MOTP 已啓用，請輸入 OTP，如圖所示。<br>
        <img border="0" src="images/PPC1.jpg"><br>
        若手機爲觸控式螢幕時，可直接點擊或雙擊圖示。<br>
        另外可使用手機的方向鍵及左右鍵或者數字鍵來操作系統 :<br>
        1 及 5:執行功能, 3:換另一頁, 4:左移選項, 6:右移選項, 9:離開程式<br>
        <form>
            OTP :<input type="text" name="sOTP"> 
            <input name="doSubmit" type="submit" value=" 驗證 "><br>
        </form>
        <%= message %> <br>
        </TD>
     </TR>
</TABLE> 
</BODY></HTML>
```

(五)、練習

請將上述個別網頁串起來模擬一個真實的遊戲認證流程 。

伍、參考書籍及網站

[1] OATH，http://www.openauthentication.org/

[2] 近代密碼學及其應用，賴溪松、韓亮、張真誠，松崗文魁

[3] JAVA 密碼學，Jonathan Knudsen，O'Reilly

[4] 資訊安全理論與實務，陳彥學，文魁資訊

[5] OpenSSL，http://www.openssl.org/

[6] BouncyCastle，http://www.bouncycastle.org/

[7] Tomcat，http://tomcat.apache.org/

[8] JAVA，http://java.sun.com/

[9] Apache httpd 網頁伺服器，http://httpd.apache.org/

[10] PHP，網頁 CGI，http://www.php.net/