**解析傳輸協定與運作流程**

**網站SSL加密原理簡介**

老薯條

SSL是Secure Socket Layer（安全套接層協議）的縮寫，可以在Internet上提供秘密性傳輸。

**Secure Socket Layer說明**   
  
SSL是Secure Socket Layer（安全套接層協議）的縮寫，可以在Internet上提供秘密性傳輸。最早是Netscape公司所提出，SSL的目標在於保證兩個應用間通訊的機密性和完整性以及可驗證伺服器身分    
  
SSL目前已廣泛的應用在HTTP連線上，當以「https://」方式連上網站，如果瀏覽器的右下角有一個鑰匙，即代表該網站有支援SSL。SSL主要是運作在應用層與傳輸層間，如下圖示。

|  |
| --- |
| http://www.netadmin.com.tw/images/news/NP110614000811061414250101.png |
| ▲SSL主要是運作在應用層與傳輸層間。 |

主要分為兩層，上層為SSL Handshake、SSL Change Cipher spec及SSL Alert通訊協定，其主要作用如下所述：    
  
**SSL Handshake**   
  
這是SSL在傳輸之前事先用來溝通雙方（用戶端與伺服器端）所使用的加密演算法或密鑰交換演算法，或是在伺服器和用戶端之間安全地交換密鑰及雙方的身分認證等相關規則，讓雙方有所遵循。在身分認證方面，SSLHandshake可用來認證伺服器的身分。SSL Handshake的運作流程如下所述：

|  |
| --- |
| C:\Users\tommy\AppData\Local\Temp\enhtmlclip\Image(9).png |
| ▲SSL Handshake的運作流程。 |

(1) SSL用戶端利用Client Hello訊息將本身支援的SSL版本、加密演算法、演算法等資訊發送給SSL伺服器。    
  
(2) SSL伺服器收到Client Hello訊息並確定本次通訊採用的SSL版本和加密套件後，利用Server Hello訊息回覆給SSL用戶端。    
  
(3) SSL伺服器將利用Certificate訊息將本身公鑰的數位憑證傳給SSL用戶端。    
  
(4) SSL伺服器發送Server Hello Done訊息，通知SSL用戶端版本和加密套件協商結束，並開始進行密鑰交換。    
  
(5) 當SSL用戶端驗證SSL伺服器的證書合法後，利用伺服器的證書中之公鑰加密SSL用戶端隨機生成的Premaster Secret（這是一個用在對稱加密密鑰產生中的46位元組的亂數字），並透過Client Key Exchange消息發送給SSL伺服器。    
  
(6) SSL用戶端發送Change Cipher Spec消息，通知SSL伺服器後續報文將採用協商好的密鑰和加密套件進行加密。    
  
(7) SSL用戶端計算已交互的握手消息的Hash值，利用協商好的密鑰和加密演算法處理Hash值，並透過Finished消息發送給SSL伺服器。SSL伺服器利用同樣的方法計算已交互的握手消息的Hash值，並與Finished消息的解密結果比較，如果兩者相同，則證明密鑰和加密套件協商成功。    
  
(8) SSL伺服器發送Change Cipher Spec訊息，通知SSL用戶端後續傳輸將採用協商好的密鑰和加密套件進行加密。    
  
(9) SSL伺服器計算已交互的握手消息的Hash值，利用協商好的密鑰和加密套件處理Hash值，並透過Finished消息發送給SSL用戶端。SSL用戶端利用同樣的方法計算已交互的握手消息的Hash值，並與Finished消息的解密結果比較，如果兩者相同，且MAC值驗證成功，則證明密鑰和加密套件協商成功。在SSL用戶端接收到SSL伺服器發送的Finished消息後，如果解密成功，則可以判斷SSL伺服器是數位證書的擁有者，即SSL伺服器身分驗證成功。這是因為只有擁有私鑰的SSL伺服器才能從Client Key Exchange消息中解密得到Premaster Secret，從而間接地實現了SSL用戶端對SSL伺服器的身分驗證。    
  
**SSL Change Cipher spec**   
  
用來變更雙方傳輸加解密的演算法與訊息驗證的規格，傳輸雙方可利用此協定進行溝通並設定本次傳輸所使用的協定。    
  
**SSL Alert**   
  
當傳送雙方發生錯誤時，用來傳遞通訊雙方所發生錯誤的訊息。訊息包含告警的嚴重級別和描述。    
  
**SSL Record Protocol**   
  
SSL Record Protocol主要提供訊息的完整性及機密性要求。就機密性而言，將會利用在SSL Handshake階段所得到的Key（加密金鑰）針對往來的HTTP通訊加密；至於完整性，SSL將使用MAC（Message Authentication Code）的方式來驗證訊息是否有被更動。MAC通常使用MD5演算法來確保往來的訊息並沒有被篡改。    
  
以下簡單說明MD5演算法。MD5是使用「單向函數」來驗證訊息的完整性，單向Hash函數指的是給與一個鍵值（Key），即可透過鍵值藉由Hash函數取得唯一的雜湊值（Vale），例如F(KEY)=VALUE，如果存在兩個以上的不同鍵值可得到相同的雜湊值，即是指此Hash函數產生碰撞（Collision）。一個正常的單向函數是不能有碰撞情況的產生。    
  
單向Hash（雜湊）定義如下所述，假設：   
明文為M 雜湊函數是H 雜湊值為h = H(M)    
  
單向（One-way）Hash函數需符合下列的功能：   
● 對任意長度的明文輸入，須能產生固定長度的雜湊值輸出。   
● 對於任何的明文一定可產生相對應的Hash值，且可利用硬體或軟體來產生。   
● 須可從明文產生Hash值，但不能由Hash值反推而得到明文，例如h=H(“Hello!! World”) ;。假如得到Hash值E23d21341DEFA789；在任何情況下都不能由E23d21341DEFA789反推而得到「Hello!! World」的明文，這也就是單向Hash函數的特性。   
● 對於明文 M1，在計算上是無法找出另一個明文 M2 ≠ M1，使得 H(M1) = H(M2)。 也就是一個明文產生一個Hash碼；而不能兩個不同的明文產生相同的Hash碼。   
● 若H(M1) = H(M2)，則M1 = M2，若H(M1)≠H(M2)，則M1≠M2。亦即同一個明文產生同一個Hash碼，不同個明文產生不同的Hash碼。    
  
接著以Linux系統為例，說明md5sum相關指令：

|  |
| --- |
| C:\Users\tommy\AppData\Local\Temp\enhtmlclip\Image(10).png |

(1) 產生內容為hello world的「/tmp/test」檔案。   
(2) 利用「/tmp/test」檔案內容產生md5檢驗碼，稱為known。   
(3) 修改「/tmp/test」檔（僅差別一個空白字元）。   
(4) 利用修改後的「/tmp/test」檔案內容產生md5檢驗碼。    
  
由上得知，即使檔案僅更動一個字元，利用md5演算法所產生的檢驗碼也會完全不同。藉此來驗證訊息是否有被更動。簡而言之，SSL連線約可劃分為下列四個步驟：

|  |
| --- |
| C:\Users\tommy\AppData\Local\Temp\enhtmlclip\Image(11).png |
| ▲SSL連線四個步驟。 |

1.SSL Client端發出ClientHello給SSL伺服器端。告知伺服器端本身可實現的算法列表和其他一些需要的資訊。    
  
2.SSL的服務器端在接收ClientHello後會回應一個ServerHello，裡面確定了這次通訊所需要的演算法，並送出伺服器本身的憑證（資訊內包含身分及公鑰）。    
  
3.SSL Client會新增一個秘密金鑰，並利用伺服器傳來的公鑰來加密，而且會回傳加密後的秘密金鑰密文給伺服器。    
  
4.伺服器使用自己的私鑰解開秘密金鑰密文，取得秘密金鑰後，即利用此秘密金鑰來相互通訊。    
  
上述流程中所謂的數位憑證，採用的格式為x509，x509的架構即如下圖中金字塔般的架構：

|  |
| --- |
| C:\Users\tommy\AppData\Local\Temp\enhtmlclip\Image(12).png |
| ▲x509憑證架構。 |

其中，Root CA（在微軟系統的IE8稱為「根憑證授權單位」，讀者可看到IE8已經預設許多根憑證授權單位）授權給CA（在微軟系統的IE8稱為「中繼憑證授權單位」），然後使用者再至認證中心註冊來取得可證明身分的憑證，x509憑證主要的欄位內容如下：

|  |
| --- |
| C:\Users\tommy\AppData\Local\Temp\enhtmlclip\Image(13).png |

在IE8中，可利用點選功能表【工具】→【網際網路選項】，然後按下〔內容〕活頁標籤內的〔憑證〕按鈕來查看憑證內容，如下圖所示。    
  
以電子商務應用而言，SSL可輕易達成「機密性」、「完整性」及可幫助使用者驗證伺服器的身分。因為一個完善的網站會去申請相關憑證，而在申請憑證的過程中即會驗證該伺服器的身分。

|  |
| --- |
| C:\Users\tommy\AppData\Local\Temp\enhtmlclip\Image(14).png |
| ▲查看憑證內容。 |

這對一個支援SSL的網站而言，使用者相信資料在傳輸的過程中不會外洩及不會被更改，並且會將資料傳送到正確的網站，但對於網站經營者來說，SSL機制由於未對使用者強制認證，所以無法分辦使用者的身分。也因此出現了SET（Secure Electronic Transaction）的機制。

SET是由Master和Visa兩大信用卡組織所提出的標準，主要用來提供一個安全的線上購物環境，SET與SSL最大的差異在於使用者必須取得認證，方可在符合SET規格的商務網站上進行交易，藉此確認使用者的身分。