一個高效率通訊安全協定
A High Efficiency Protocol for Session Key on Internet

孫光天
K. T. Sun
國立台南師範學院
資訊教育研究所
ktsun@ipx.ntntc.edu.tw

胡育銘
Y. M. Hu
國立台南師範學院
資訊教育研究所

摘 要

使用者與伺服器間欲建立起安全的連線，並協商出此次連線所需的連線金鑰(session key)，大多數的做法是利用使用者的密碼並結合指數運算來完成，如 EKE、SPEKE 等方法。這些金鑰協定不但可讓使用者與伺服器互相做認證，以安全的協商出一把連線金鑰，並且可以避免離線攻擊法（off-line guessing attack）破解，因此，在安全性上是相當高的。在本研究中，我們以 EKE 和 SPEKE 方法為基礎概念，簡化伺服器公開金鑰的傳送次數，使我們方法的安全性和其他類似方法是一樣的，但在效能方面，我們所提的方法，比起現有其他方法，不但指數運算次數減少，而且建立連線金鑰所需的溝通次數更是降低，整個連線金鑰協定的效能大大提昇，而此通訊協定已建立在本實驗室研發之分散式測驗系統中，初步驗證本方法具有高安全性與效能。

關鍵詞：連線金鑰、通訊協定、指數運算

一、簡介

為了讓使用者與伺服器間建立起安全的通訊，傳統的方式是利用使用者自訂的密碼來對傳送的資料加密，而伺服器端在事先知道此密碼的情況下，便可利用相同的密碼來對收到的資料解密。不過為了記憶方便，一般人所選用的密碼都是有意義的字串或數字，因此容易遭受離線攻擊法（off-line guessing attack）破解，而使整個通訊變的不安全。

在 1976 年時，Diffie 和 Hellman [4] 利用指數運算的方式，讓通訊的雙方建立起一把連線所需的連線金鑰(session key)，並使連線金鑰的安全性落在解離散對數的困難性，而使得此連線金鑰不容易遭受到攻擊而破解，但是由於通訊的雙方在建立連線金鑰的過程中，無法確認對方的身分，因此容易遭受到中間人攻擊法的攻擊。所以，在 1992 年時，Bellovin 和 Merritt 提出將使用者密碼與 Diffie-Hellman 交換金鑰方式結合，不但可使雙方互相確認對方身分，避免中間人攻擊法，並可使產生的連線金鑰很安全，離線攻擊法也很難破解，這種通訊雙方共同享有使用者密碼，並結合指數運算方式產生連線金鑰，例如：EKE [2, 3]、SPEKE [6]、SRP [11]、Kwon 和 Song 的方法 [10]及 Guo 和 Hong（郭強毅、洪國寶，2000）等；另一種方式則是通訊的雙方透過彼此互相信任的伺服器，並利用伺服器的公開金鑰（public key）建立連線金鑰，此方式有 Gong [5]、Keung and Siu [7]、Kwon and Song [8, 9]等，為近年來研究之通訊安全協定。

在本篇論文中，我們將提出以 EKE 為基礎概念，並加入伺服器公開金鑰的想法在裡面，應用於使用者與伺服器建立安全連線，我們整個協定的安全性和現有的幾種方法一樣，但是在建立連線金鑰所需的指數運算次數及回合數卻更少。

二、相關研究

我們的方法由於是基於使用者密碼與指數運算建立連線金鑰，以下將介紹幾種也是利用此種方式的協定，如 EKE、SPEKE、Known 和 Sun 及 Guo 和 Hong。

2.1 EKE 協定

EKE（Bellovin & Merritt [2]）是利用 Diffie-Hellman 產生金鑰的方式並結合使用者的密碼所制定出的一種通訊協定，利用密碼來辨別對方身分，可防止攻擊者偽裝成通訊的另一方，並利用這次產生的金鑰作為這次通訊雙方資料所需加密的金鑰，整個流程如下：

首先有兩個公開的數 p 及 g，g 為 p 的原根，p 為一個大質數（1024 bits），使用者 A 和 B 分享著 A 的密碼（PWD），此時，要協商出一把這次連線用的金鑰，步驟如下。

（一） A 產生一個亂數 Xa，Xa<p，計算出 Ya 並利用自己的密碼 PWD 來將 Ya 加密得到 $E_{PWD}(Ya)$，將 $E_{PWD}(Ya)$傳給 B。

$Ya = g^{Xa} \bmod p$ (1)

A to B：$E_{PWD}(Ya)$

（二） B 也產生一個亂數 Xb，Xb<p，計算出 Yb 並用已知 A 的密碼 PWD 來對 Yb 加密得到 $E_{PWD}(Yb)$。並將收到的 $E_{PWD}(Ya)$ 利用 PWD 解開得到 Ya，接著用 Ya 和自己產生的 Xb 計算出這次的連線金鑰 K，再接著產生一個亂數 Cb 並用 K 將 Cb 加密得到 $E_K(Cb)$，將 $E_K(Cb)$連同 $E_{PWD}(Yb)$ 一起傳給 A。

$Yb = g^{Xb} \bmod p$ (2)

B to A：$E_K(Cb)$、$E_{PWD}(Yb)$

（三）使用者 A 先將收到的 $E_{PWD}(Yb)$利用 PWD 來解開得到 Yb，並再用自己的 Xa 計算出 K，接著用 K 將 $E_K(Cb)$解開得到 Cb，並另外產生一個亂數 Ca，將 Ca 和 Cb 用 K 加密得 $E_K(Ca、Cb)$傳給 B。

A to B：$E_K(Ca、Cb)$

（四）B 收到 $E_K(Ca、Cb)$後，用 K 將它解開，並驗證解開的 Cb 和自己產生的 Cb 有無一樣，如果一樣，將 Ca 用 K 加密得到 $E_K(Ca)$再回傳給 A。

B to A：$E_K(Ca)$

（五）A 收到 $E_K(Ca)$後，用 K 將它解開，並且驗證 Ca 和自己產生的 Ca 有無一樣，如果相同，代表兩方建立相同的金鑰。

由 EKE 流程我們可以知道，在 EKE 建立連線金鑰的過程，總共需要 4 次模指數運算及 4 個回合數，如圖一所示。

2.2 SPEKE 協定

SPEKE（Jablon [6]）也是一種讓通訊的雙方建立起這次連線金鑰的方法，SPEKE 和 EKE 都是利用 Diffie-Hellman 作交換金鑰的動作，不過最大的差別在於 EKE 中，$g^x \bmod p$ 裡 g 為 p 的原根，而在 SPEKE 中，g 為使用者的密碼 PWD 經過一個函數 f 運算之後的值。

如同 EKE 一樣，使用者 A 和 B 分享著 A 的密碼 PWD，h()為單向雜湊函數，整個 SPEKE 過程如下。

（一） 使用者 A 產生一個亂數 Xa，Xa<p，計算出 Ya，將 Ya 送給 B。

$Ya = f(PWD)^{Xa} \bmod p$ (3)

A to B：Ya

（二） 使用者 B 產生一個亂數 Xb，Xb<p，計算出 Yb，並利用收到的 Ya 和自己的 Xb 計算出金鑰 K。

$Yb = f(PWD)^{Xb} \bmod p$ (4)

$K=Ya^{Xb} \bmod p$ (5)

計算 h(h(K))，將 Yb 與 h(h(K)) 送給 B。

B to A：Yb、h(h(K))

（三）使用者 A 利用收到的 Yb 和自己產生的 Xa，計算出金鑰 K。

$K=Yb^{Xa} \bmod p$ (6)

並計算 h(h(K))，驗證此值和收到由 B 傳來的 h(h(K))有無一樣，如果一樣，代表 K 正確，並回傳 h(K)給 B 做確認。

A to B：h(K)

在 SPEKE 中，驗證 K 有無正確除了可用單向雜湊函數外，也可以和 EKE 一樣，利用金鑰 K 來加密一個亂數作驗證，如果利用單向雜湊函數作驗證的話，整個金鑰建立的回合數比用加密一個亂數還少一次。

2.3 Kwon and Song 協定

Kwon and Song（Kwon and Song [10]）也是利用 Diffie-Hellman 建立連線金鑰，並結合使用者密碼作為互相認證的一種通訊協定。

首先有兩個公開的數 p 及 g，g 為 p 的原根，p 為一個大質數，h()為單向雜湊函數，通訊雙方開始連線前，伺服器(簡稱 Server)的資料庫中，存放著使用者(簡稱 Client)的名稱 ID、$R \oplus PWD$ 和 $g^u \bmod p$，R 為伺服器為每個使用者產生不同的隨機亂數，PWD 為使用者密碼，⊕代表 XOR 邏輯運算，+為加法運算，u=f(R+PWD)，f()為一個自訂的函數，整個連線步驟如下。

（一） Client 輸入自己的 ID 和密碼 PWD，並送出自己的 ID 給 Server 接著 Client 產生一個亂數 Xc，Xc < p，並計算 Ya1。

$Ya1 = g^{Xc} \bmod p$ (7)

Client to Server：ID

（二） Server 根據收到的 ID，到資料庫中找尋該使用者的 $g^u \bmod p$，並產生一個亂數 Xs，Xs< p，Server 計算 g^{Xs}，並和 $g^u \bmod p$ 相加得到 Yb1，再連同該使用者的 $R \oplus PWD$ 一起送回給該使用者，並計算出 Yb2。

$Yb1 = g^{Xs} + (g^u \bmod p)$ (8)

$Yb2 = (g^u)^{Xs} \bmod p$ (9)

Server to Client：Yb1、$R \oplus PWD$

（三） Client 收到 Yb1、$R \oplus PWD$ 後，利用自己的 PWD 和 $R \oplus PWD$ 做 XOR 得到 R，計算出 u=f(PWD+R)及 $g^u \bmod p$，利用

g^u mod p 得到 g^{Xs} 並計算 Ya2 和 $(g^{Xs})^u$ mod p，最後計算 Ya3 和 h(Yb1，Ya2)送給 Server。

$$Ya2 = (g^{Xs})^{Xc} \bmod p \quad (10)$$

$$Ya3 = (g^{Xs})^u \bmod p + g^{Xc} \bmod p \quad (11)$$

Client to Server：Ya3、h(Yb1，Ya2)

（四）Server 收到後，利用已計算出的 $(g^u)^{Xs}$ mod p 得到 g^{Xc} mod p，並計算$(g^{Xc})^{Xs}$ mod p 和 $h(g^{Xs} + g^u \bmod p)$，$(g^{Xs})^{Xc}$ mod p)，Server 並驗證和 Client 的 h(Yb1，Ya2)是否一樣，如果一樣，Server 送給 Client h(Ya3、Ya2)。

Server to Client：h(Ya3、Ya2)

（五）Client 計算 $h((g^u)^{Xs} \bmod p + g^{Xc} \bmod p$、$(g^{Xc})^{Xs} \bmod p)$並驗證和收到的 h(Ya3、Ya2)有無一樣，如果相同，雙方以 $h((g^{Xc})^{Xs} \bmod p)$ 作為這次連線資料加密用的金鑰 K。

$$K = h((g^{Xc})^{Xs} \bmod p) \quad (12)$$

2.4 Guo and Hong 協定

Guo and Hong's protocol（郭強毅、洪國寶[1]）也是利用 Diffie-Hellman 建立連線金鑰並結合使用者密碼作為互相認證的一種通訊協定。

g 為 p 的原根，p 為一個大質數，h()為單向雜湊函數，伺服器(簡稱 Server)的資料庫中，存放著使用者(簡稱 Client)的名稱 ID 和以 h(PWD)加密的 g^{Rc} mod p （以 $E(g^{Rc} \bmod p)$ 表示 ），其中 Rc 為伺服器為每個使用者預先產生不同的隨機亂數，PWD 為使用者密碼，X 為 Server 的私密金鑰，f()函數為帶入 IP 產生 IP+1 。

三、Sun and Hu 協定

我們的通訊協定（Sun-Hu protocol），主要是整合前面所提及的 EKE 架構，並且加入了 Server 公開金鑰的想法在裡面，將我們的協定應用在使用者（簡稱 Client）與伺服器（簡稱 Server）間建立安全連線金鑰，而此公開金鑰在連線第一次建立時才加密傳送，可增加系統安全性與方便姓，隨後連線時，則不再傳送此公開金鑰，增加連線建立之效率。

首先，在連線建立通訊之前，假設 Server 的資料庫中已儲存使用者的名稱（簡稱 User）及經過一次單向雜湊函數的密碼值 h(PWD)，h(PWD)由 h(Xs)加密過才存放在資料庫，Xs 為 Server 的私密金鑰，Ys 為 Server 公開金鑰，Xs 和 Ys 的關係為 $Ys=g^{Xs} \bmod p$，Xs 為一個 1 到 p-1 間的亂數，g 為 p 的原根，p 為一個大質數(1024 bits)。由於使用者端(簡稱 Client)為第一次與此 Server 連線，所以電腦裡則事先無存放任何資料（包括公開金鑰），整個協定流程如下：

一、 Client 輸入 User 及 PWD，然後 Client 產生一個亂數 Xa，Xa<p，計算 Ya 及 h(PWD)，將 Ya 和 h(PWD)做 XOR 後得到 Ya'，將 User 及 Ya' 送給 Server。

$$Ya = g^{Xa} \bmod p \quad (13)$$

$$Ya' = Ya \oplus h(PWD) \quad (14)$$

二、 Server 程式收到後，根據 User 找尋相對應的 h(PWD)，由於 h(PWD)被 h(Xs)加密過，因此先將其解開，並將 Ya'和 h(PWD)做 XOR 後得到 Ya 值，利用 Ya 和 Xs 計算出此次的連線金鑰 K_C，並計算出 $h(K_C)$和 $h(h(K_C))$。然後將 Ys 和 h(PWD)做 XOR 後得到 Ys'， 送回給 Client 為 Ys'及 $h(K_C)$。

$$K_C = Ya^{Xs} \bmod p \quad (15)$$

$$Ys = g^{Xs} \bmod p \quad (16)$$

$$Ys' = Ys \oplus h(PWD) \quad (17)$$

三、Client 收到資料後，用 Ys'和 h(PWD)做 XOR 得到 Ys，利用自己產生的亂數 Xa 及 Server 公開金鑰 Ys 計算 K_C，並計算出 $h(K_C)$和 $h(h(K_C))$，並和收到 $h(K_C)$做驗證，假如正確，代表雙方建立起此次相同的連線金鑰 K_C，回傳 $h(h(K_C))$給 Server。

四、Server 收到後，驗證 $h(h(K_C))$有無一樣，如果一樣，代表雙方建立相同的連線金鑰。

整個連線流程如下圖二所示。

使用者經過第一次連線後，Client 程式會將 Server 公開金鑰 Ys 儲存在 Client 自己的電腦裡，當 Client 欲進行第二次連線時，整個流程如下：

一、Client 輸入 User 及 PWD，然後產生一個亂數 Xa 值，Xa < p，計算 $Ya=g^{Xa} \bmod p$ 及 h(PWD)，利用 Ys 及 Xa 來產生連線金鑰 K_C ($K_C=Ys^{Xa} \bmod p$)，計算 $h(K_C)$和 $h(h(K_C))$，然後將 Ya 和 h(PWD)做 XOR 後為 Ya'，將 User、Ya'及 $h(K_C)$透過網路傳送給 Server 程式。

二、Server 程式收到後，根據 User 找尋相對應的 h(PWD)，由於 h(PWD)被 h(Xs)加密過，因此先將其解開，並將 Ya'和 h(PWD)做 XOR 後得到 Ya，利用 Ya 和 Xs 計算出 K_C，$K_C=Ya^{Xs} \bmod p$，然後計算出 $h(K_C)$

和 $h(h(K_C))$，將 $h(K_C)$ 和收到的 $h(K_C)$ 做比較，如果一樣代表為合法使用者，且此次產生連線用的金鑰 K_C 正確，並傳回 $h(h(K_C))$ 給 Client。

三、Client 收到後，驗證 $h(h(K_C))$，假如正確，代表雙方建立起此次相同的連線金鑰 K_C。

整個連線流程如下圖三所示。

Client 的 K_C 是利用 Server 的公開金鑰 Ys 和自己產生的亂數 Xa 計算出來的，而 Server 的 K_C 是利用亂數 Ya 和自己的私密金鑰 Xs 計算出來的，由以下推論可知，Client 與 Server 經由我們的協定後，雙方所產生的連線金鑰 K_C 都是一樣的，推論過程如下：

Client 產生的金鑰 K_C

$$
\begin{aligned}
&= Ys^{Xa} \bmod p \\
&= (g^{Xs} \bmod p)^{Xa} \bmod p \\
&= (g^{XsXa} \bmod p) \bmod p \qquad (18) \\
&= (g^{Xa} \bmod p)^{Xs} \bmod p \\
&= Ya^{Xs} \bmod p \\
&= \text{Server } K_C
\end{aligned}
$$

在經由第一次連線後，使用者端的電腦裡存放著伺服器的公開金鑰，由於是公開金鑰，所以使用者不需另外去保護此金鑰，也不需記憶此金鑰值，使用者只需在第一次使用時，透過網路取得此公開金鑰後，並將此金鑰存放在自己電腦裡，之後使用者與伺服器每一次做安全連線時，仍只需輸入自己所記憶的密碼即可，並不會額外增加使用者負擔，由於每一 Client 端的私密金鑰 Xa 不同，所以，連線金鑰 Kc 也不同。

四、Sun-Hu protocol 安全性分析

本研究所提之通訊協定中，使用單向雜湊函數，基本上其安全性是很高，攻擊者是無法從 $h(h(K_C))$ 或 $h(K_C)$ 反推回去得到 K_C，此外，解離散對數也是很困難的，攻擊者是無法從伺服器的公開金鑰 Ys 解得私密金鑰 Xs，所以，本身很難直接破解，至於其他攻擊法是否能承受，分析如下。

4.1 離線攻擊法（off-line guessing attack）

攻擊者欲以離線的方式，反覆不斷的猜測密碼來攻擊 K_C 是很難的，假使攻擊者藉由猜測的密碼，從 Ya’得到猜測的 Ya，攻擊者欲知道此猜測值有無正確，還必須知道 Xs，進而計算出 K_C，並和 $h(h(K_C))$ 或 $h(K_C)$ 驗證此 K_C 有無正確，但我們知道攻擊者要知道 Xs 則必須克服解離散對數的問題，目前困難性非常高，所以，我們的方法是可以抵擋離線攻擊的。

4.2 中間人攻擊法（Man in the middle attack）

假使一攻擊者 C 利用線上的方式，在伺服器 A 與使用者 B 之間，偽裝成使用者 B，要和 A 建立一把連線金鑰，由於攻擊者 C 沒有 B 真正的密碼，因此無法和伺服器 A 建立起一把連線金鑰。另外，假使攻擊者 C 欲偽裝成伺服器 A，和 B 建立一把連線金鑰，由於攻擊者不知道伺服器真正的私密金鑰，因此仍無法和 B 建立連線金鑰，所以，我們的方法是可以抵擋中間人攻擊法。

4.3 重複攻擊法（Replay attack）

攻擊者將所竊取的資訊 Ya’及 $h(h(K_C))$，重複的傳送給伺服器，也只能得到伺服器傳回來的 $h(K_C)$，而此對 K_C 的建立是沒有幫助的，所以，我們的方法是可以抵擋重複攻擊法。

4.4 Denning Sacoo 攻擊法

Denning Sacoo 是假設使用者密碼被竊，如何求取連線金鑰 K_C。由於本協定 K_C 建立的過程中，皆和使用者的密碼沒有關係，假如使用者密碼被竊取，攻擊者也無法由使用者的密碼得知 K_C，所以我們的協定是可以抵擋 Denning Sacoo attack。

五、效能分析

通訊雙方建立連線金鑰的協定中，在安全性足夠（已非常高）的情況下，如果使用到的指數運算次數及建立金鑰所需的回合數越少，則通訊雙方建立金鑰的時間就會較短，因此我們可以說這種協定在效能上是較好的。所以我們將幾種利用使用者密碼與指數運算來建立金鑰的協定，如 EKE（Bellovin & Merritt [2]）、SPEKE（Jablon [6]）、Kwon and Song（Kwon & Song [10]）以及 Guo and Hong（Guo and Hong [1]）與我們的協定作比較，在安全性都足夠的情況下，根據這幾種協定所需溝通的回合數與計算指數運算所需的次數作一分析比較，如表一所示。

由表一可知，本研究所提的方法，指數運算只需三次，而溝通次數，第一次連線建立需三次外，其餘連線均只需兩次，即可達到連線建立與金鑰確定工作，有最佳的溝通效率。

表一、不同金鑰協定效能分析比較

	指數運算次數	建立金鑰溝通回合數
EKE (1992)	4	4
SPEKE (1996)	4	3
Known and Song (1999)	5	4
Guo and Hong (2000)	3	4
Sun-Hu's protocol (the first time)	3	3
Sun-Hu's protocol (after the first time)	3	2

六、結論

在本論文中, 我們的方法主要架構在 EKE 原理上，並加入伺服器公開金鑰理念在裡面，在不會額外增加使用者負擔的情況下，我們的方法不但安全性高, 而且整個協定過程非常簡單，在建立連線金鑰的過程中，如果為使用者第一次連線, 需要 3 次指數運算, 3 個回合數，此後，使用者建立連線金鑰的回合數，降到只需要 2 個回合，達到最低溝通需求數，使溝通次數達到最佳(最少)化。

誌　謝

本研究計劃，經費由國科會科教處贊助，研究計劃編號：NSC 91-2520-S-024-014。

參考文獻

[1] 郭強毅、洪國寶（2000）。*Creating Secure Session Key Using Weak Passwords*。第十屆全國資訊安全會議論文集，367-373。

[2] Bellovin, S. and Merritt, M. (1992). Encrypted key exchange password-based protocols secure against dictionary attack. *IEEE Symposium on Research in Security and Privacy*, 72-84.

[3] Bellovin, S. and Merritt, M. (1993) Augmented Encrypted Key Exchange：a Password-Based Protocol Secure Against Dictionary Attacks and Password File Compromise. *At&t bell Laboratories.*

[4] Diffie, W. and Hellman, M. E. (1976). New Directions in Cryptography. *IEEE Transactions on Information Theory*, 644-654.

[5] Gong, L. (1995). Optimal Authentication Protocol Resistant to Password Guessing Attacks. *Proceedings of the 8th IEEE Computer Security Foundations Workshop*, 24-29.

[6] Jablon, D. (1996). Strong Password-Only Authentication Key Exchange. *Computer Communication Review*, 26(5), 5-26.

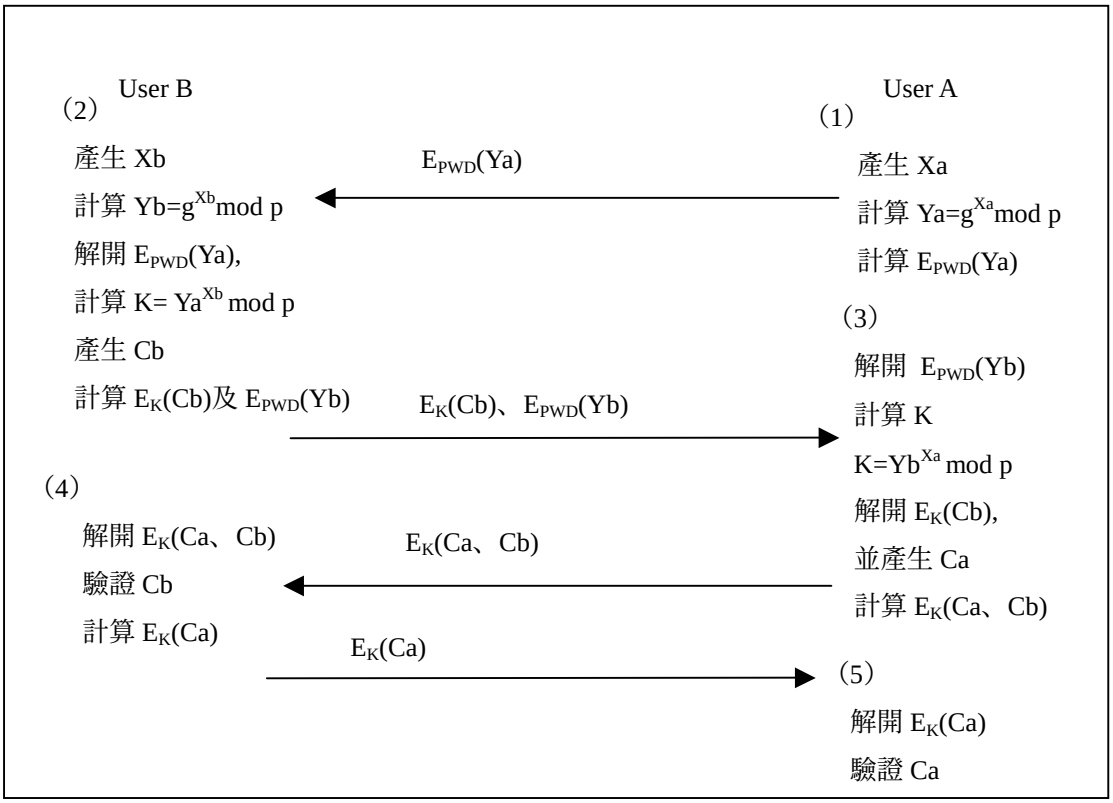
[7] Keung, S. and Siu, K. (1995). Efficient protocols secure against guessing and replay attacks. Computer Communications and Networks, Proceedings., Fourth International Conference ,105-112.

[8] Kwon, T. and Song, J. (1997). Security and efficiency in authentication protocols resistant to password guessing attacks. Local Computer Networks. Proceeding., 22nd Annual Conference, 245-252.

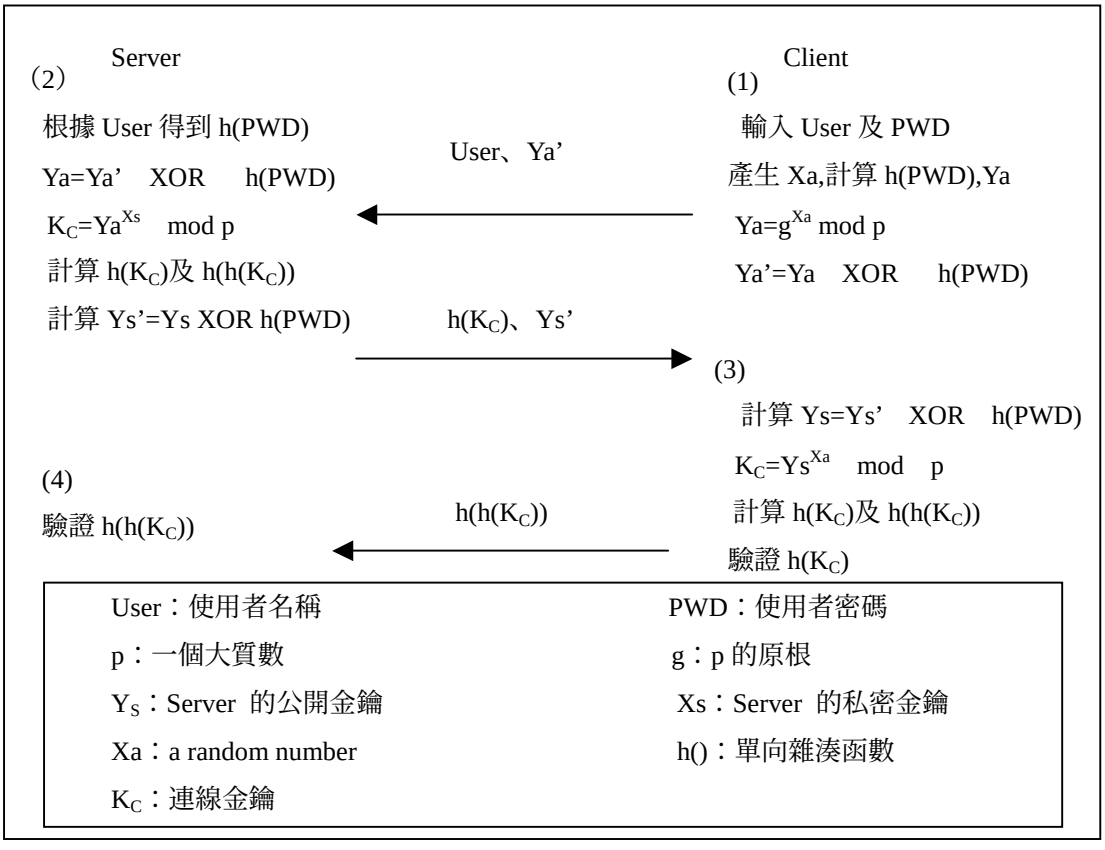
[9] Kwon, T. and Song, J. (1998). Authenticated key exchange protocols resistant to password guessing attacks. Communications, IEE Proceedings, 145, 304-308.

[10] Kwon, T. and Song, J. (1999). Secure Agreement Scheme for g^{xy} Via Password Authentication. *Electronics Letters*, 35(11), 892-893.

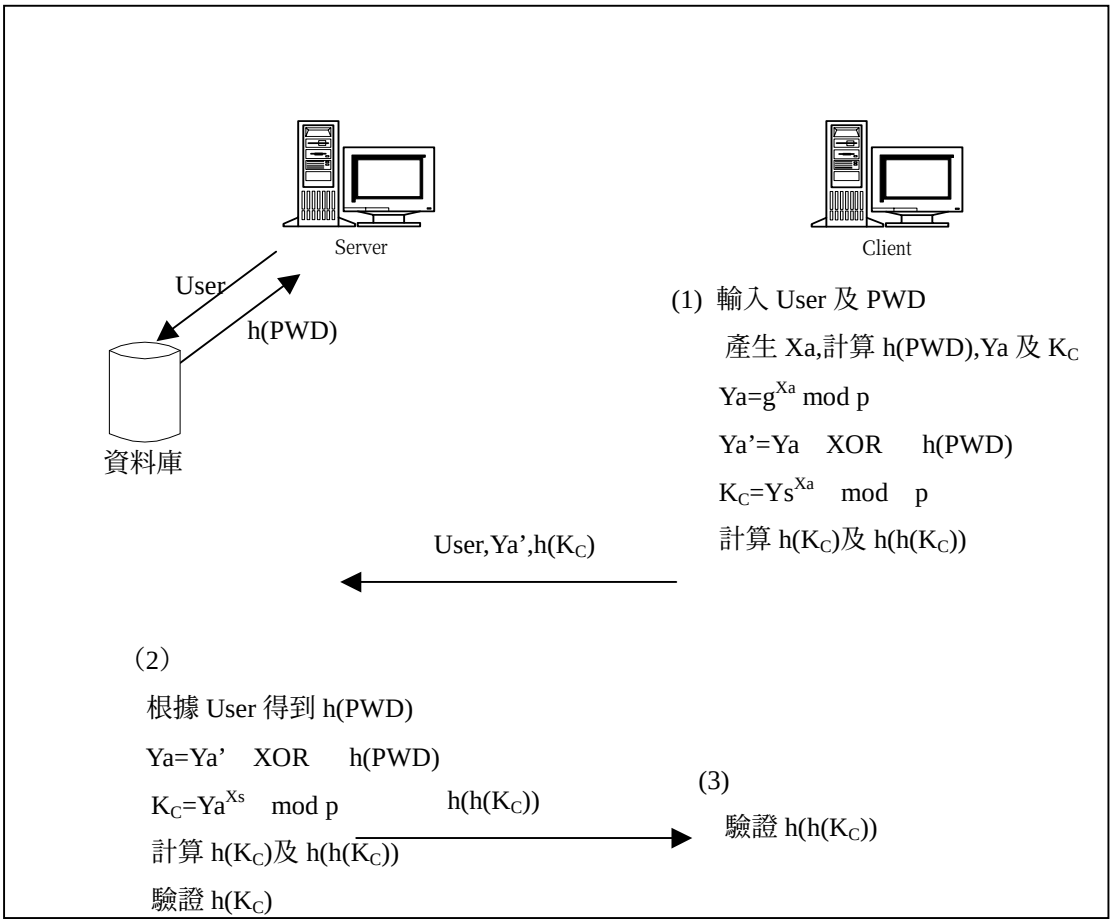
[11] Wu, T. (1998). Secure remote password protocol. *Internet Society Symposium on Network and Distributed System Security.*



圖一、EKE protocol



圖二、Sun and Hu protocol（第一次連線建立）



圖三、Sun and Hu protocol（第二次及隨後之連線建立）