

使用通行碼的數位簽章協定

學生：李尚宸

指導教授：曾文貴 博士

國立交通大學資訊科學研究所



我們知道使用通行碼的協定必須能抵抗字典攻擊法，因此我們提出一個必須結合服務者的秘密資訊、客戶所知道的秘密資訊及通行碼才能產生合法的數位簽章的協定。因為協定的需求是正確的客戶及服務者才能產生合法的數位簽章，所以我們必須要確認客戶以及服務者的身份。完成了身份認證的要求後所產生的數位簽章可以利用相對應的公開金鑰來驗證其合法性。在客戶的秘密資訊洩漏，或是服務者的秘密資訊洩漏的情況下，攻擊者皆無法利用字典攻擊法得知通行碼或是偽造出一個合法的數位簽章。另外我們也對我們的協定給予一個安全性的證明。

關鍵詞：通行碼驗證、數位簽章、抵擋字典攻擊法

Password-based Signature Scheme


Student : Shang-Chen Lee

Advisor : Dr. Wen-Guey Tzeng

Institute of Computer and Information Science

National Chio Tung University

Abstract

The logo of National Chio Tung University is a circular emblem with a gear-like border. Inside the circle, there is a stylized building and a banner with the year '1896'.

We know that password-based schemes must be able to against dictionary attacks. We proposed a digital signature scheme that we have to know the password, client and server's secret to sign out a valid signature. Because we require that the valid signature could only be produced if both client and server are correct, client and server need to authenticate each other. We can verify the signature which is produced after authentication by the corresponding public keys. If the secret either server or client holds has been leaked out, the attacker can not neither find out the correct password nor forge a valid signature by using dictionary attacks. And there is also a security proof for our signature scheme.

Key Words : Password-based authentication 、 Signature scheme 、 Against dictionary attack

誌謝

在此感謝我的指導老師曾文貴教授，在我碩士班兩年的學習過程中，不只讓我在學業上受益良多，更在生活上以及言行上給我許多教導。此外，我要感謝口試委員，交大資工系蔡錫鈞教授和清大資工系孫宏民教授，在論文上給予我許多良好的建議和指導，讓我的論文更加完善。除此之外我要感謝實驗室同學，兆儀、坤杉、振魁和佩琳的幫忙，實驗室學長成康、惠龍、學姊季穎的指導，以及最可愛的女朋友和實驗室學弟妹們在精神方面的鼓勵。

最後，我要感謝我的家人，不論在精神或物質上都給予我極大的支持，讓我在無後顧之憂的情況下可以順利完成學業。在此，謹以此文獻給我所有我想要感謝的人。



目錄

摘要	i
Abstract	ii
誌謝	iii
目錄	iv
第一章 引言	- 1 -
第一節 研究動機.....	- 2 -
第二節 研究目標與成果.....	- 4 -
第三節 各章節介紹.....	- 6 -
第二章 數位簽章、身份認證協定與互動式證明系統	- 7 -
第一節 相關研究.....	- 7 -
第二節 各種攻擊.....	- 8 -
第一項 竊聽攻擊法.....	- 9 -
第二項 重送攻擊法.....	- 9 -
第三項 字典攻擊法.....	- 10 -
第四項 Denny-Sacco攻擊法.....	- 12 -
第五項 中間人攻擊法.....	- 14 -
第六項 小型子群限制.....	- 15 -
第三節 零知識互動式證明系統及知識簽章.....	- 16 -
第三章 使用通行碼的數位簽章協定	- 21 -
第一節 數學符號與假設.....	- 21 -

第二節 使用通行碼的數位簽章協定.....	- 22 -
第四章 安全性分析	- 27 -
第一節 安全性定義與證明模式.....	- 27 -
第一項 理想領域的介紹.....	- 27 -
第二項 真實領域的介紹.....	- 29 -
第三項 安全性定義.....	- 31 -
第二節 針對一般攻擊法的安全性討論.....	- 32 -
第三節 安全性證明.....	- 35 -
第一項 理想世界與真實世界的不可分辨.....	- 35 -
第二項 ElGamal加密系統的安全性	- 38 -
第三項 數位簽章的安全性.....	- 40 -
第五章 結論與未來工作方向	- 47 -
第六章 參考文獻	- 48 -



第一章 引言

由於現今科技的蓬勃發展，造就了網際網路的發達與行動裝置的普遍，使得人們在許多地方都能藉由行動裝置連上網際網路取得許多的服務，然而在網路的環境中，一切傳遞的訊息都是公開的，因此這些網路服務必須結合安全的機制進行保護，而數位簽章正是其中一種，因為數位簽章具有簽章的性質並且數位化，因此在電子商務的發展中數位簽章扮演著相當重要的角色。

傳統的文件在發表聲明時，常常會利用簽名讓收文者辨識這份文件的正確性，而在現今這個科技發達的時代，我們可以利用數位簽章演算法 (Digital Signature Algorithm) 來達到將簽章數位化的目的。數位簽章演算法是建立在公開金鑰架構 (Public Key Infrastructure) 上，在這樣的架構中，簽章者擁有一組私密金鑰 (Private Key) 與公開金鑰 (Public Key)，私密金鑰顧名思義必須是只有簽章者才知道的金鑰。當簽章者對某份文件簽署數位簽章時，他根據私密金鑰透過簽章演算法 (Signature Algorithm) 產生出該文件的數位簽章，而公開金鑰與驗證演算法則是公布在網路上，讓驗證者 (Verifier) 取得以確認數位簽章的正確性，確認的方式是透過驗證演算法以及公開金鑰，對我們收到的數位簽章進行驗證的工作，如果驗證的結果是正確的，那麼表示該數位簽章的確是對應此公開金鑰的私密金鑰擁有人所簽署。

第一節 研究動機

在行動裝置與網際網路的發達下，結合兩者產生的服務將會更多樣化，我們可以想見的，利用行動裝置的便利性和運算能力，我們可以在行動裝置上針對特定的文件作簽名的動作，產生數位簽章，然而，光是這樣是不夠的，我們所希望的是只有真正的客戶才能產生出這樣的數位簽章，於是乎，要滿足這樣的要求就必須加入身份認證的功能。一般來說使用者的身份認證的方法基本上可以分為以下三類：

- 一、 使用者知道什麼：這個方式是目前最普遍使用的方式，利用使用者所知道的資訊來做身份認證，也就是一般我們所知道的利用通行碼 (Password) 來確認身份，但是由於人的記憶力有限，所以通行碼通常只能是一小段由數字及文字所組合而成的字串，因此通行碼長度都不會很長。然而這樣的特性卻會產生一些問題，除了通行碼在網路上傳遞時容易遭到竊聽外，通行碼很容易遭受字典攻擊法 (Dictionary Attack) 的攻擊。
- 二、 使用者的生理特徵：每一個人都有自己獨特的生理特徵，例如指紋，聲音，或是對視網膜做掃瞄，每個人所具有的生理特徵都不一樣，因此可以利用生理特徵作為身份認證的方法，這樣可以將證明的物品被竊取而遭到冒用的可能性減到最低，但是目前辨別這些生理特徵的機器並不普遍，因此這樣的身份認證方式一樣有問題存在。
- 三、 使用者擁有什麼：使用者使用一些實際的物品來證明自己的身份，例如 IC 卡裡面所儲存的資訊，而在要求證明身份的時候，使用者將 IC 卡插入讀卡機，讀出這類物品所擁有的資訊，來證明使用者

的身份，但是這樣的方式有一個明顯的問題存在，當證明使用者身份的物品遺失了，就很容易遭到惡意攻擊者的冒用。

正如我們前面所說的三種身份認證的問題，如果對特定文件簽名的動作，只仰賴行動裝置的秘密資訊來計算產生，那麼當行動裝置遺失了，惡意的攻擊者就可以冒充合法的使用者，偽造出一個合法的數位簽章，為了解決這個問題，於是乎就有人想利用通行碼來計算產生數位簽章，如此一來就可以解決行動裝置遺失，攻擊者便可以偽造數位簽章的問題了，但是這樣的方法依然存在著問題，那就是惡意的攻擊者只要利用字典攻擊法來攻擊，猜測通行碼並且計算出數位簽章，之後將這個數位簽章與合法的數位簽章做比對，不斷的重複這樣的動作，就可以利用這樣的方式，攻擊者可以找出正確的通行碼，有了通行碼之後就可以偽造出合法的數位簽章，然而這並不是我們所希望的，因此這樣的方法也不可行。如果我們結合上述兩者來產生數位簽章，那麼或許會是一個不錯的方式，但是單純的結合行動裝置的秘密資訊與通行碼來計算產生數位簽章並不可行，這兩種身份認證方式所會遭遇的問題特性依然存在，也就是說，如果行動裝置遺失了，惡意的攻擊者可以利用字典攻擊法產生數位簽章，去比對出正確的通行碼，之後便可偽造出合法的數位簽章，那麼怎麼去解決這個這樣的問題，是一個有趣的課題。

我們在這篇論文中提出了結合服務者與客戶的秘密資訊，再加上使用者的通行碼產生的數位簽章，在這樣的情況下，藉由服務者秘密資訊的保護，如果行動裝置遺失了，惡意的攻擊者沒有辦法利用字典攻擊法找出通行碼，當然也沒有辦法產生合法的數位簽章，而萬一服務者被攻擊者攻陷了，也就是說，攻擊者知道了服務者所擁有的秘密資訊，在這樣的情形下攻擊者一樣沒有辦法找出通行碼，自然也不能產生合法的數位簽章。

第二節 研究目標與成果

在我們的協定中的數位簽章是利用知識簽章 (Signature of Knowledge) 的方式來產生，所謂的知識簽章是利用零知識互動式知識證明 (Zero knowledge interactive Proof of knowledge) 轉換成零知識非互動式證明系統 (Zero knowledge non-interactive Proof System) 來完成，而零知識非互動式證明系統就是我們所稱的知識簽章，一般來說對於數位簽章而言必須滿足下面三個性質：

- 一、 正確性 (Correctness)：只要擁有正確秘密資訊的客戶與服務者使用正確的通行碼便可以產生合法的數位簽章。
- 二、 私密性(Privacy)：對於攻擊者而言，在協定交換訊息的過程中，觀察公開資訊是無法得到秘密資訊及通行碼的資訊。
- 三、 不可偽造性(Unforgability)：攻擊者在不知道秘密資訊的情況下，無法偽造出合法的數位簽章。

其中針對私密性而言，客戶與服務者在互相確認對方身份的過程中所傳遞的訊息同樣要滿足私密性，也就是說對於攻擊者而言，身份認證過程中所傳遞的訊息是沒有辦法幫助攻擊者得知秘密資訊的。然而在早期絕大多數的身份認證協定都沒有正規化的安全性分析，這些身份認證的協定都只是單純的針對現有的攻擊方式來防禦，但是只對現有的攻擊法做防禦是不夠的，因為攻擊者攻擊的方式與技巧一直在更新，因此正規化的分析對於身份認證的協定而言是不可或缺的。

在 2000 年的 PAK[4]中提出了一個在隨機聖賢模式 (Random oracle model) 下針對「以通行碼為基礎的身份認證與金鑰交換協定」的正規化模式 (Formal model) 定義。在[4]中定義了兩個系統來證明 PAK 的安全性，

分別是理想領域 (Idealworld) 和真實領域 (Realworld)。我們說 PAK 中定義的理想領域中是安全的，因為身份認證與金鑰交換的動作都是透過可信任的第三者 (Trust Third Party) 來完成，然而在真實領域中，並沒有可信任的第三者存在，所以身份認證與金鑰交換是使用「以通行碼為基礎的身份認證與金鑰交換協定」來完成。在這樣的模式下，利用我們造出的模擬器，使得理想領域與真實領域是相同的，藉此證明其安全性。我們的目標是在產生數位簽章的過程中加入身份認證的機制，雖然我們的目的地與 PAK 不同，但是對身份認證的需求是一致的，因此我們利用 PAK 所提出的證明方式來證明「使用通行碼的數位簽章協定」在做身份認證時的安全性，那麼，根據 PAK 中的定義，我們對於身份認證所需要證明的安全性必須滿足以下兩個特性：

- 一、 正確性 (Completeness)：對於任意真實世界的使用者，只要正確的執行協定，則一定能完整並成功的認證。
- 二、 可擬性 (Simulatability)：「使用通行碼的數位簽章協定」在身份認證時傳遞的訊息，其真實領域執行的概觀 (View) 與在理想領域執行的概觀是等價的。

由以上我們可以根據數位簽章，以及身份認證所需滿足的特性，來設計我們的協定，因此我們的研究目標與主要成果有下面幾點：

- 一、 抵擋一般的攻擊方法：針對目前現有的攻擊方式作探討，確保我們的協定不會遭受到攻擊。
- 二、 對於身份認證正規化的安全性證明：利用 PAK 提出的模式，對我們協定的私密性做了一個安全性證明。證明的方式是假設攻擊者能夠破解身份認證協定，那麼我們就可以利用攻擊者來解 DDH 問題。
- 三、 對於數位簽章的安全性證明：我們證明我們協定所產生的數位簽章，是知識簽章，也就是零知識非互動式證明系統來滿足對於數位

簽章的安全性要求。

第三節 各章節介紹

在接下來的章節中，第二章會著重於在數位簽章、身份認證協定與零知識互動式證明系統的介紹，包括了身份認證協定的相關研究，針對通行碼驗證的攻擊方式，以及零知識互動式證明系統和知識簽章。第三章則會介紹我們提出的「使用通行碼的數位簽章協定」。在第四章將會對我們提出的協定做安全性的定義及分析。其中第一節將會介紹在 PAK 中所定義的正規化模式與我們所需要的安全性定義，在第二節中討論為什麼我們的協定可以抵擋現有的攻擊法，第三節針對我們的協定提出正規化的證明，這當中包括了身份認證過程的安全性，ElGamal 加密系統的安全性，以及我們所產生的數位簽章是知識簽章的證明。最後我們在第五章做了一個總結，以及未來可以繼續努力的方向。




第二章 數位簽章、身份認證協定與互動式

證明系統

在我們的協定中的數位簽章是建立在互動式證明系統的基礎之上，而產生合法的數位簽章前必須要對客戶的身份作確認，因此除了數位簽章的相關研究、數位簽章的建立基礎這兩部分的介紹外，我們還會介紹身份認證的相關研究，以及在身份認證協定中可能的攻擊方式。

第一節 相關研究



在西元 1978 年 R. Rivest, A. Shamir, 以及 L. Adleman 在[21]提出了 RSA 的加解密系統及數位簽章後，數位簽章的發展便如雨後春筍般蓬勃發展，幾個比較著名的數位簽章有在西元 1984 年 T. ElGamal 的[22]，由 NIST 在西元 1994 年所訂立的 DSA (Digital Signature Algorithm)，以及在西元 1988 年 C. Guillou 以及 J. Quisquater 提出的[23]等等，之後數位簽章協定便根據不同的情況與需求而設計，例如門檻式數位簽章系統 (Threshold Signature Scheme) [26][27]，代理簽章系統 (Proxy Signature Scheme) [24][25]，群體簽章系統 (Group Signature Scheme) [28][29] 等等。而我們提出的協定在產生數位簽章之前，必須要確認服務者與客戶雙方的身份。目前絕大多數身份認證的協定在驗證身份後，都會交換一把交談金鑰 (Session key)，並且利用這把金鑰加密之後傳遞的訊息，所以接下來我們介紹“身份認證金鑰交換協定”的相關研究。

最早利用通行碼為基礎來進行認證並且交換金鑰的機制是在西元 1989

由 M. Lomas 等四人所提出的[7]，這篇的做法是建立在公開金鑰架構上，到了西元 1992 年，S. Bellare 以及 M. Merritt 提出 EKE[1]，EKE 是第一個不需要在公開金鑰架構下的「以通行碼進行身份認證的金鑰交換協定」，EKE 的觀念在於他將金鑰交換協定過程中的傳遞訊息利用通行碼進行對稱式加密，而 DH-EKE[1]與 EKE 不同之處在於 DH-EKE 用通行碼加密的是 Diffie-Hellman 金鑰交換過程中的傳遞訊息。

西元 1993 年提出 EKE 的兩人，S. Bellare 與 M. Merritt 再度提出 A-EKE[2]，目的在於解決 EKE 與 DH-EKE 共同的問題，在 EKE 與 DH-EKE 中，在服務者端所存放的驗證資訊是通行碼，因此當服務者被惡意攻擊者入侵，那麼攻擊者便輕易的得到通行碼，而 A-EKE 在服務者端存放的驗證資料是非對稱的形式 (Asymmetric Model)，這樣的方式解決了之前的問題。之後陸陸續續有許多的協定被提出，例如由 EKE 衍生的 M-EKE[10]、SPEKE[5]、建立在 RSA 假設上的 OKE[8]、解決服務者資訊洩漏的門檻式身份認證協定 T-PAKE[12]、AMP[6]等。在這些協定中有些協定被破解 [3][11][16]有些則是仍然停留在安全性分析的階段，原因在於他們沒有一個正規化的證明。然而在 S. Lucks 所提出的 OKE 中有了一個證明的方向。

最早達到安全性證明要求的是 O. Goldreich 與 Y. Lindell 提出的[14]，接下來的 SNAPI[9]、PAK、KOY[13]、FPAKE[15]也都證明了他們的協定的安全性，而我們協定對於身份認證的證明模式則是利用 PAK 中所提出的方式。

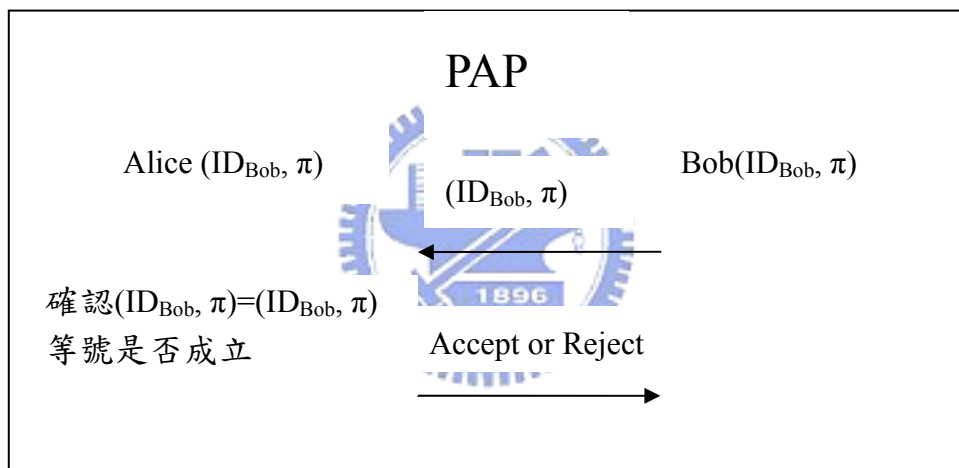
第二節 各種攻擊

在網路這樣一個開放的環境中，認證身份雙方所傳遞的訊息都是公開的，於是乎，惡意的攻擊者便可以利用這些公開的訊息來進行攻擊，因此

在我們設計協定時，必須要考慮到是否會遭受到這些攻擊方式的攻擊。所以我們必須瞭解這些攻擊方式，以及為何這樣的攻擊方式可以成功攻擊協定。

第一項 竊聽攻擊法

竊聽攻擊法顧名思義，就是利用在網路上傳遞的訊息是公開的，藉由監聽網路的封包取得有用的資訊，我們以 PAP 這個協定來做說明，竊聽攻擊法如何取得有用的資訊，PAP 協定的運作方式如下：



由於在 PAP 中，由於帳號與通行碼是直接網路上傳送給服務者來驗證，所以惡意的攻擊者只要監聽網路上傳遞的封包，就可以得到一組帳號與相對應的通行碼，因此通行碼不能不經過任何處理就在網路上傳送。

第二項 重送攻擊法

重送攻擊法 (Replay Attack) 的攻擊方式，是利用監聽網路環境，紀錄雙方溝通的訊息，之後偽裝客戶，重新傳遞這些記錄下來的訊息，以達到

偽裝合法客戶的目的，我們以前面的PAP來做說明。惡意的攻擊者只需要監聽網路環境，並且記錄下Bob傳送給Alice的 (ID_{Bob}, π) 值，之後就可以偽裝成Bob，將這組值重新傳送給Alice，如此一來就可以取得服務。

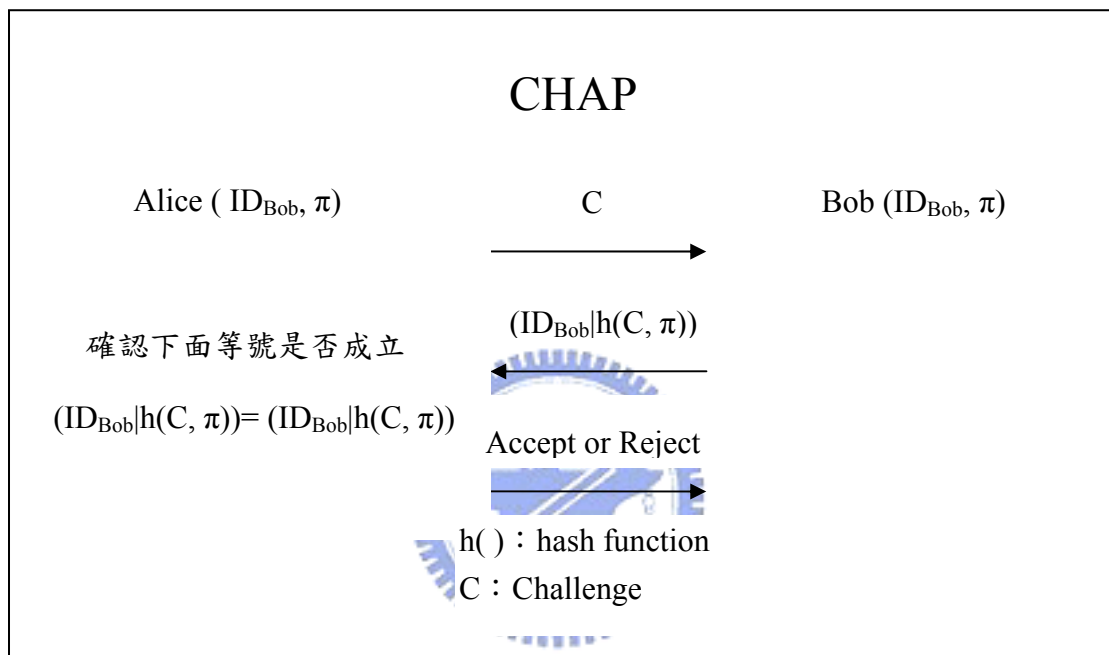
重送攻擊法之所以能夠成功的原因在於，客戶與服務者每次做身份認證的時候，傳遞的訊息都相同，因此如果要避免重送攻擊法的攻擊，在協定中每次認證傳遞的訊息必須不相同。

第三項 字典攻擊法

由於通行碼是方便人腦記憶的一小段特殊的文字，通常在八到十個位元，相較於我們使用的安全金鑰是短很多的，因此惡意的攻擊者可以針對這個特性，以這八到十個位元所有可能的通行碼進行窮舉（exhaustive search）的測試方式，來猜測出通行碼，將這些可能的通行碼經由協定運作的計算方式，去比對運算後的值，藉此可以找出通行碼，而字典攻擊法又可以分為以下兩種：

- 一、即時字典攻擊法（On-line Dictionary attack）：即時的字典攻擊法是指攻擊者直接偽裝使用者的角色與另一個使用者執行協定，攻擊者重複猜測使用者的通行碼，直到與另一個使用者執行協定的結果為成功為止，那麼成功的偽裝合法使用者所使用的通行碼，就是正確的通行碼。對於這樣的攻擊方式我們可以利用限制執行協定時錯誤的次數來減低攻擊者猜出通行碼的可能性，我們以機率的角度來看，假設攻擊者嘗試猜測通行碼的次數為 t ，而通行碼的字典空間為 d ，那麼攻擊者成功猜測到通行碼的機率就是 t/d ，也就是說我們只要將 t 的值限制在一個很小的值，就可以讓攻擊者成功猜測到通行碼的機率變的很低。

二、離線字典攻擊法 (Off-line Dictionary attack)：離線的字典攻擊法是指攻擊者觀察並且記錄之前身份認證過程中的訊息，並且窮舉所有可能的通行碼進行計算並且與溝通的訊息比對，藉此猜測通行碼，我們以 CHAP (Challenge handshake authentication protocol) 來做說明，其運作方式如下圖所示：

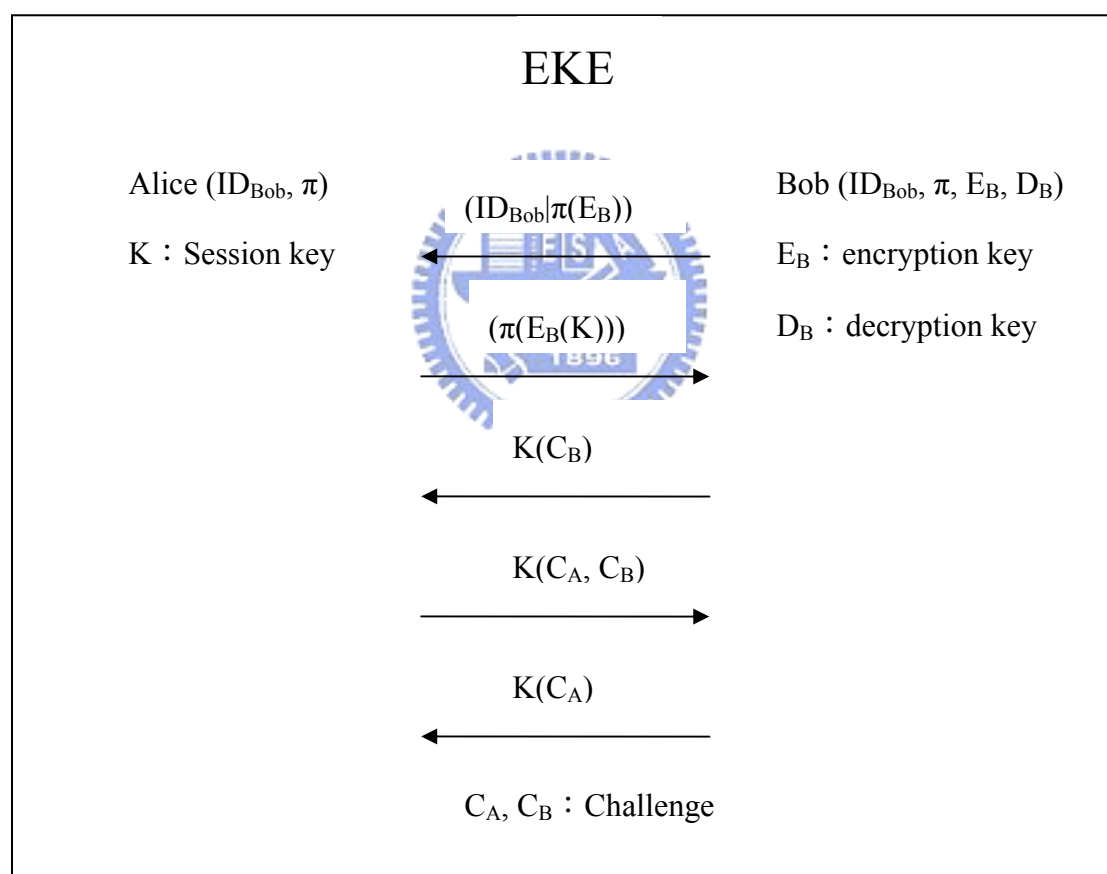


在CHAP這個協定中，每次Bob需要服務時，Alice都會給他一個挑戰值C，之後Bob這個挑戰值C與通行碼當作雜湊函數h()的輸入，最後將雜湊函數的輸出與ID_{Bob}一同傳給Alice，讓Alice去比對他自己計算出的(ID_{Bob}||h(C, pi))與Bob傳送過來的(ID_{Bob}||h(C, pi))是否相同。而離線的字典攻擊法如何攻擊CHAP這個協定呢？首先，惡意的攻擊者會將Alice與Bob溝通的訊息記錄起來，接著攻擊者猜測可能的通行碼pi'，並且依照協定的方式，將挑戰值C與通行碼pi'作為雜湊函數h()的輸入，然後與記錄下來的溝通訊息h(C, pi)比對，如果相等，表示攻擊者猜測到正確的通行碼，如果不等，則繼續猜測下一個可能的通行碼。由以上的例子我們可以知道單純的將通行碼至入

雜湊函數作為保護是不夠的，我們需要設計更精密更複雜的方式來保護通行碼的安全。

第四項 Denny-Sacco 攻擊法

我們將會利用一個可以抵擋字典攻擊法的通行碼認證協定 EKE (Encrypted Key Exchange) 來介紹 Denny-Sacco 攻擊法[19]的攻擊方式，我們先來看 EKE 的運作方式，如下圖所示：



EKE 的認證過程步驟如下：

1. 首先Bob產生加密金鑰 E_B 以及解密金鑰 D_B ，並且利用通行碼 π 加密 E_B 之後，傳送 $(ID_{Bob}, \pi(E_B))$ 給Alice。

2. Alice收到 $(ID_{Bob}, \pi(E_B))$ 後利用通行碼 π 解密 $\pi(E_B)$ 可以得到 E_B ，接著 Alice產生一把交談金鑰 K ，並且利用 π 將 E_A 加密過的 K 再加密一次，之後將 $\pi(E_B(K))$ 傳送給Bob。
3. Bob先後利用 π 以及 D_B 解密出交談金鑰 K ，接著隨機產生一個挑戰值 C_B ，並且將 C_B 用 K 加密後，回傳給Alice。
4. Alice收到 $K(C_B)$ 後，利用 K 解密出 C_B ，然後隨機產生一個挑戰值 C_A ，並且利用 K 將 C_A 與 C_B 加密傳給Bob。
5. Bob將 $K(C_A, C_B)$ 解密得到 C_A 與 C_B ，接著比對這個 C_B 與之前傳送給Alice的挑戰值 C_B 是否相等，如果相等，則用 K 將 C_A 加密後回傳給Bob。

如果步驟 1.到步驟 5.都能正確無誤的執行完畢，那麼 Alice 與 Bob 雙方便能確認對方的身份，同時也產生了一把交談金鑰。

Denny-Sacco 攻擊法是假設攻擊者取得一把之前認證過程中所產生的交談金鑰，攻擊者便可以利用這把交談金鑰設法找出通行碼。接下來我們假設惡意的攻擊者已經記錄了在 EKE 認證過程中所有的通訊內容，並且得到了一把該認證內容產生的交談金鑰 K ，攻擊的方式如下：

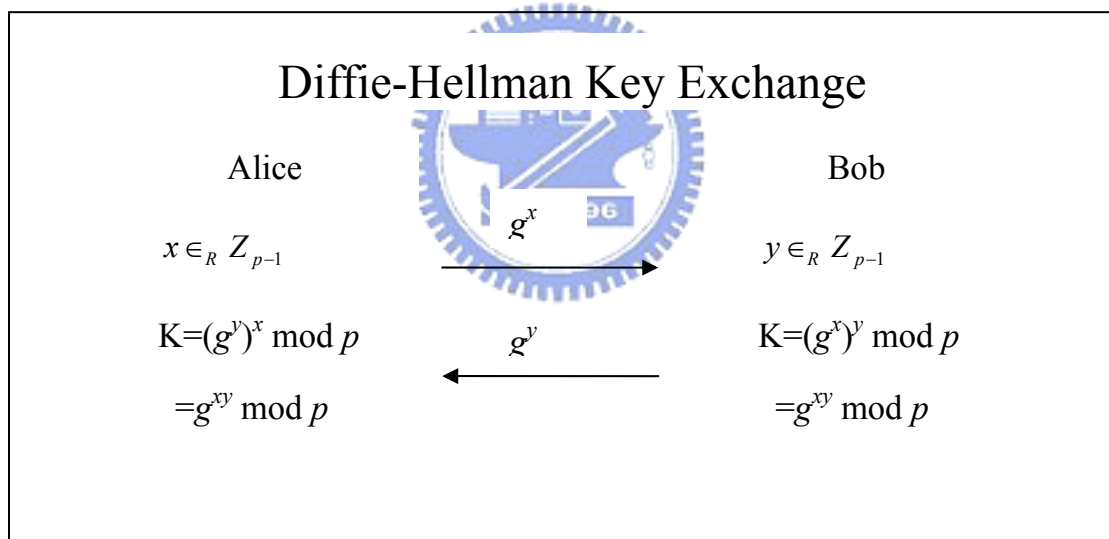
1. 攻擊者猜測一個客戶的通行碼 π' 。
2. 以 π' 將之前竊聽到的 $\pi(E_B)$ 進行解密的動作後可以得到 E_B' ，也就是說 $E_B' = \pi'^{-1}(\pi(E_B))$ 。
3. 利用 E_B' 加密之前得到的交談金鑰 K ，得到 $E_B'(K)$ 。
4. 根據 π' 我們可以加密 $E_B'(K)$ 得到 $\pi'(E_B'(K))$ ，接著與之前的 $\pi(E_B(K))$ 做比較，如果不相同則回到步驟 1.，如果相同則代表攻擊者猜測的 π' 就是客戶的通行碼。

因為通行碼的長度通常很小，所以對於 Denny-Sacco 攻擊法中的攻擊者而言，重複地去猜測使用者的通行碼造成的影響並不大，這樣的攻擊方式之所以會成功的原因在於，交談金鑰與認證過程中的通訊內容有著直接的

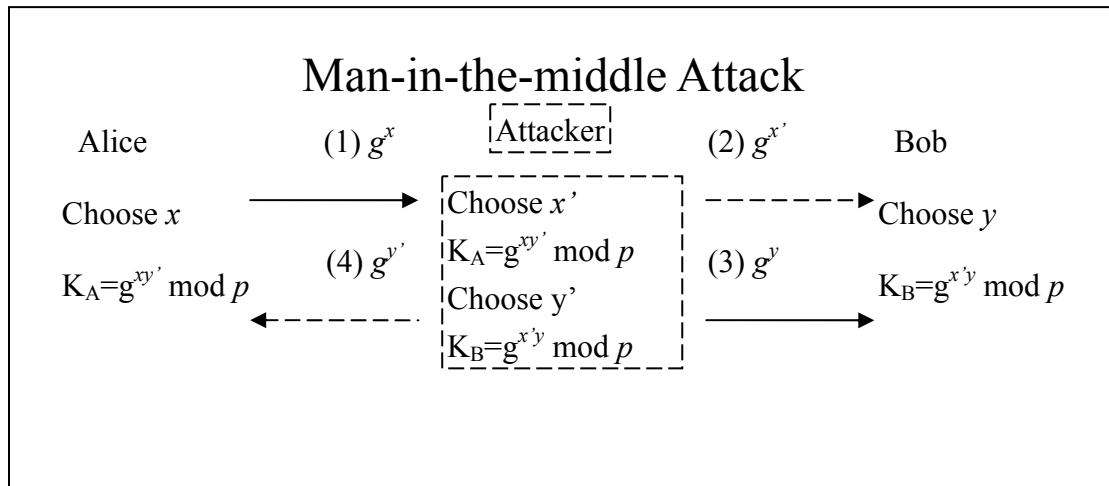
關係。

第五項 中間人攻擊法

中間人攻擊法 (Man-in-the-middle attack) 的攻擊方式是指攻擊者位在客戶與服務者之間，並且同時扮演客戶與服務者的角色，也就是說，對客戶而言，攻擊者扮演著服務者的角色，而對服務者來說，攻擊者扮演客戶的角色，使得攻擊者與客戶擁有一把交談金鑰，而攻擊者與服務者則擁有另外一把交談金鑰。我們以 Diffie-Hellman 金鑰交換協定[20]為例子來做說明，運作的方式如下：



由上圖我們可以很清楚地知道，在執行Diffie-Hellman金鑰交換協定時，Alice會從 Z_{p-1} 中隨機選取一個 x ，接著計算出 g^x 並且送給Bob，而Bob也是隨機從 Z_{p-1} 中選取 y ，計算出交談金鑰 $K = g^{xy} \bmod p$ ，之後將 g^y 傳送給Alice，而Alice也可以計算出交談金鑰 K 。而中間人攻擊法的攻擊方式我們利用下圖來表示：

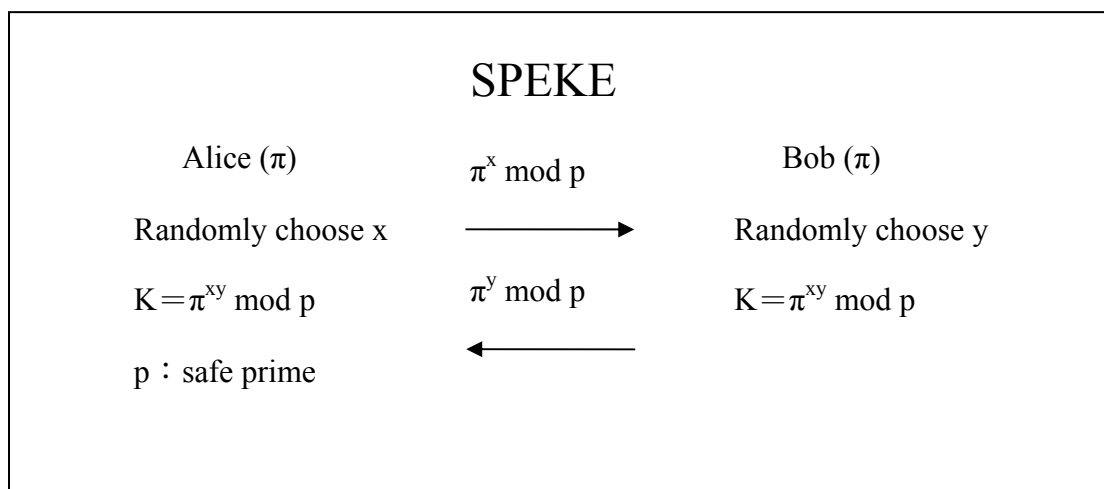


很明顯的，由上圖我們可以看出攻擊者在完成四個步驟的攻擊後攻擊者會與Alice擁有一把交談金鑰 $K_A = g^{xy'} \pmod p$ ，也與Bob擁有一把交談金鑰 $K_B = g^{x'y} \pmod p$ ，如此一來攻擊者可以利用這兩把金鑰來解開Alice與Bob經過加密的溝通訊息。中間人攻擊法關鍵在於在做金鑰交換的雙方，並沒有做身份認證的動作。

第六項 小型子群限制

在一個公開金鑰系統 (Public Key System) 中會使用到許多的數值，而這些數值的運算都是封閉 (Close) 在一個有限群 (Finite Group) 中，這樣的有限群都會有一些基數 (Order) 較小的子群 (Subgroup)，而這些基數較小的子群一樣具有封閉性 (Closure)，所以在子群內的元素互相做運算後的值，仍然會是這個子群的元素，也就是說假設我們有一個質數 p ，而 q 是 $p-1$ 的因數，那麼則會存在一個 Z_p^* 的子群稱之為 G_q ，在 G_q 中的成員互相做運算後，仍然會落在 G_q 這個子群中。小型子群限制 (Small subgroup confinement) 就是利用這一點來攻擊公開金鑰系統。接下來我們以SPEKE

來說明如何進行這樣的攻擊。

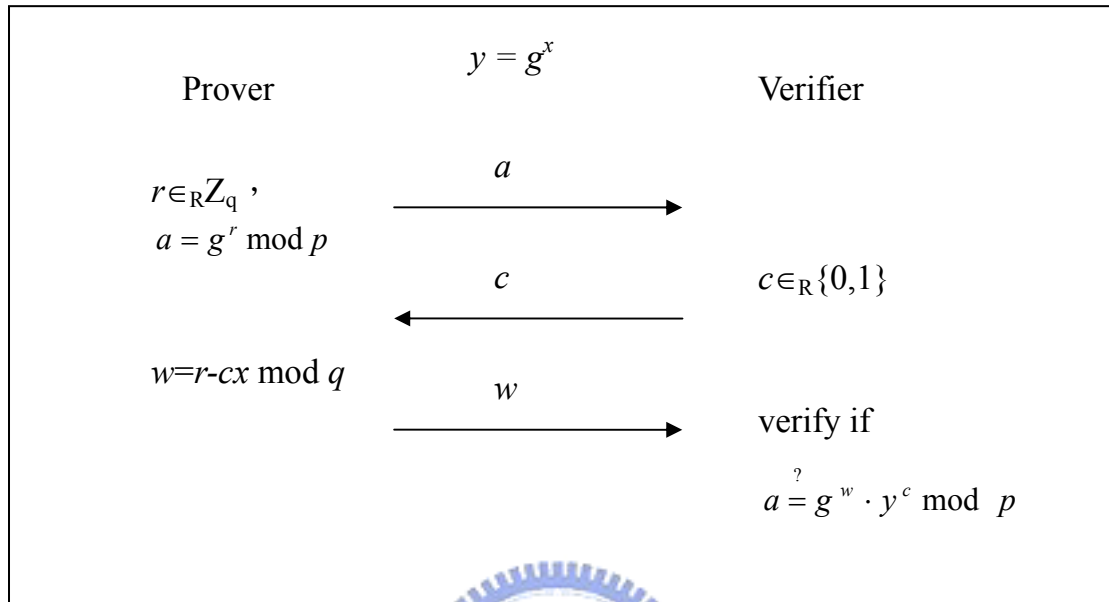


攻擊者攻擊的方式是中間人攻擊法的一種變化，攻擊者位於Alice與Bob之間，之後攔截Alice要傳送給Bob的 $\pi^x \bmod p$ ，並且修改為 $(\pi^x)^q \bmod p$ 傳送給Bob，接著同樣地將Bob要傳送給Alice的 $\pi^y \bmod p$ 修改為 $(\pi^y)^q \bmod p$ ，其中 $q=(p-1)/2$ 也是一個值數，這樣攻擊者就可以知道Alice與Bob溝通後所產生的交談金鑰為 1。根據尤拉定理 (Euler's Theorem)，對基數是 $|G|$ 的有限群 G 中的任意成員 h ， $h^{|G|}$ 等於 G 這個群中的單位元素。因此在SPEKE中，任何數值的 q 次方再同餘 (module) p 的值一定等於單位元素 1。如果要避免這種攻擊方式，我們得注意在協定中計算產生的數值是否為單位元素，如果是單位元素的話，就必須重新運作一次協定。

第三節 零知識互動式證明系統及知識簽章

在我們的協定中，數位簽章的安全性是建立在一個很重要的基礎上，那就是互動式證明系統 (Interactive Proof System)。如下圖所示，在互動式證明系統的架構中，有一個證明者 (Prover) 與一個驗證者 (Verifier)，證

明者要向驗證者證明或說服 (Convince) 他知道某個秘密值，而且證明者不希望洩漏任何跟這個秘密值有關的資訊。



互動式證明系統

一個互動式證明系統基本上又可以分為兩類，一種是證明者希望向驗證者證明他知道一個語言的關係 (Proof of membership of language)，而另一種術語知識的證明 (Proof of Knowledge of predicates) 則是證明者要向驗證者證明知道秘密值的實際值如上圖所示，而我們所定義的零知識互動式證明系統是屬於後者。接下來我們引用[17]的定義，其中：

- PPTM 表示機率的多項式時間杜林機器 (Probabilistic polynomial-time Turing machine)
- Q 是一個二元的術語，使得任何擁有正確型式的 x 值，都有一個秘密值 ρ ，使得 $Q(x, \rho) = 1$ 。

定義1、 $\langle P(\rho), V \rangle(x)$ 為 P 與 V 的一個互動式證明系統，使得 x 是一般公開的輸入值，而 ρ 是 P 私有的輸入值。

定義2、讓 P, V 皆為 PPTM，我們說對於 Q 的 $\langle P, V \rangle$ 這一個零知識互動式證明系統，必須滿足下面三個條件：

1. 完整性 (Completeness)：對於任何符合 $Q(x, \rho) = 1$ 的 x 和 ρ ， $\Pr[\langle P(\rho), V \rangle(x) = 1] = 1$ 。

2. 完美性 (Soundness)：存在一個機率多項式時間的知識擷取者 (Probabilistic polynomial-time knowledge extractor) E ，使得對於任何的 x 屬於 $\text{Dom}(Q)$ (Q 的定義域) 以及任何的 P^* ，滿足下面的式子：

$$\Pr[\langle P^*, V \rangle(x) = 1] \geq \frac{1}{p(|x|)} \Rightarrow \Pr[E(P^*, v, x) = \rho^*, Q(x, \rho^*) = 1] = 1 - \varepsilon(|x|)$$

，其中 $p(\cdot)$ 是一個多項式而 $\varepsilon(\cdot)$ 是可忽略的 (Negligible)。

3. 零知識 (Zero-knowledge)：對於每一個驗證者 V^* ，都存在一個模擬器 (Simulator) M_{V^*} ，使得下面兩個分佈 (Distribution) 在多項式時間內不可分辨 (Polynomial-time indistinguishable)：

- $\{\langle P(\rho), V^* \rangle(x)\}_{x \in \text{Dom}(Q), Q(x, \rho) = 1}$
- $\{M_{V^*}(x)\}_{x \in \text{Dom}(Q)}$

在完整性的條件中， $\Pr[\langle P(\rho), V \rangle(x) = 1] = 1$ 所代表的含意是說，只要 x 和 ρ 符合 $Q(x, \rho) = 1$ ，那麼 P 說服 V 說他擁有秘密值 ρ 的機率是 1。而在完美性

方面， $\Pr[\langle P^*, V \rangle(x) = 1] \geq \frac{1}{p(|x|)}$ 的意思是說，某個證明者 P^* 說服證明者 V

他知道秘密值的機會是一個不可以忽略 (Non-negligible) 的值，而

$\Pr[E(P^*, v, x) = \rho^*, Q(x, \rho^*) = 1] = 1 - \varepsilon(|x|)$ 則是表示知識擷取者 E 有一個很高的

的機率可以擷取出正確的 secret 資訊 ρ^* ，所以完美性的條件就是說如果對於

任意的證明者 P^* 能說服 V 的機率是一個不可忽略的值的話，那麼必定存在

一個機率多項式時間知識擷取者 E 有很高的機率找出符合 $Q(x, \rho) = 1$ 的 secret

值 ρ^* ，而找出 ρ^* 的方法是把 P^* 當成子程式 (Sub-routine) 來呼叫，配合 V

以及公開值 x 這兩個輸入值做一些運算。另外，在零知識方面，我們希望對

於任何一個驗證者 V^* ，都能造出一個輸入為 x 的模擬器 M_V^* ，使得 M_V^* 的分佈與原本的互動式證明系統的分佈一樣，而這兩個分佈一樣代表著一件事，那就是攻擊者無法分辨這些傳遞的訊息是來自於真正協定的分佈或者是來自於模擬器 M_V^* 所造出來的分佈，那也就代表當實際執行互動式證明系統的協定時，從溝通訊息以及公開值中，是無法得知 ρ 這個秘密值的任何資訊。

知識簽章也就是滿足零知識條件的非互動式證明系統 (Non-interactive proof system)，而在非互動式證明系統中，證明者在不與驗證者互動的情況下，產生一個字串來取代互動式證明系統所該擁有的性質，此外這個字串還得代替互動式證明系統在驗證者端所做的挑戰 (Challenges)，在非互動式的證明系統中取代挑戰的方式有兩種，分別是隨機信號模式 (Random Beacon model) 與安全雜湊函數模式 (Secure Hash function model)。知識簽章是利用安全雜湊函數模式來達到取代挑戰的目的。下面是關於知識簽章的定義，引用自[17]：

定義1、 抗碰撞雜湊函數 (Collision-resistant hash function) : $H = \{h_n : \{0,1\}^* \rightarrow \{0,1\}^n\}$ ，則對於所有的PPTM M ，所有的多項式 $p(\cdot)$ ，以及所有足夠大的 n ， $\Pr[M(1^n) = (x_1, x_2), h_n(x_1) = h_n(x_2)] < \frac{1}{p(n)}$ ，左式的機率是根據 M 的隨機位元來決定。

定義2、 $G = \langle g \rangle$ 是一個 g 為原生根的循環群， $y \in G$ 為相對於基底 (Base) g 的離散對數值 $x \in Z$ ，它們之間的關係式為 $y = g^x$ ，另外 $H : \{0,1\}^* \rightarrow \{0,1\}^k$ 是一個將任意長度的二元字串對應到長度為 k 的二元字串的抗碰撞雜湊函數。那麼我們說配對 (c, s) 是一個 $y = g^x$ 相對於基底 g 在訊息 $m \in \{0,1\}^*$ 上離散對數值的知識簽章。驗證者可以用 $c = H(y \parallel g \parallel g^s y^c \parallel m)$ 等式來驗證。

由上面的定義我們可以知道簽名者 (Signer) 可以證明他知道 y 相對於

基底 g 的離散對數值 x ，因為只有知道 x 的人，才能用下面的方式產生離散對數的知識簽章，因此我們說知識簽章具有不可偽造性。

1. 亂數選取 $t \in Z$
2. 計算 $c = H(y \parallel g \parallel g' \parallel m)$
3. 計算 $s = t - cx \pmod{q}$
4. 輸出配對 (c, s) 為知識簽章



第三章 使用通行碼的數位簽章協定

我們將提出的「使用通行碼的數位簽章協定」(Password-based Signature Scheme) 簡稱之為 PS，在介紹我們提出的 PS 協定之前，必須先定義一些數學符號的參數設定和密碼學上的一些假設，接著才能介紹協定運作的方式包括簽章演算法，以及驗證演算法，並且對兩個部分來做說明。

第一節 數學符號與假設

在執行這個協定時，有兩個主要的角色，分別是服務者(Server)與客戶(Client)，而 π 代表的是客戶的通行碼，另外 α, β 分別是客戶與服務者所擁有的秘密資訊，而驗證用的公開金鑰則是 z 。

k 和 l 是我們協定的安全參數，其中 k 是雜湊值的長度(例如：128 或是 160 個位元)，而 l 是公開金鑰的長度，例如：1024 或 2048 個位元，在這邊的公開金鑰是針對基於離散對數(Discrete log based)的公開金鑰系統而言，例如 ElGamal 加密系統。 $\{0,1\}^*$ 表示所有有限長度的字串集合， $\{0,1\}^n$ 表示長度為 n 的字串集合。

p 為一個長度為 l 的安全質數(Safe Prime)，使得 $p=2q+1$ ，其中 q 也是一個長度為 l 的質數。在這樣的設定下，我們可以得到一個乘法群(Multiplicative group) Z_p^* 和他的一個基數為 q 的子群 (Subgroup)，稱為 $G_{p,q}$ ($G_{p,q} = \{x \mid x^q \equiv 1 \pmod p, x \in Z_p^*\}$)， $G_{p,q}$ 是 Z_p^* 下的二次剩餘 (Quadratic residuosity) 所形成的集合，並且 g_1, g_2, g_3 皆為 $G_{p,q}$ 的原生根。我們需注意的是所有協定中的值都是在 $G_{p,q}$ 這個群中運作。 H_1, H_2 和 h 為雜湊函數， $H_1 : Z_q \rightarrow Z_q$ ， $H_2 : G_{p,q} \rightarrow G_{p,q}$ ， $h : \{0,1\}^* \rightarrow \{0,1\}^n$ 。

接下來我們介紹密碼學上的一些問題定義和假設：

- 離散對數問題(Discrete logarithm problem，簡稱為DLP)是指給定一串數值 (y, p, g) ， p 是一個安全質數，使得 $p=2q+1$ ，其中 q 亦為一個質數。 g 是一個在 Z_p^* 下的原生根， $y \in_R G_{p,q}$ ，而要如何計算出 x ， $x \equiv \log_g y \pmod{p}$ 。
- 離散對數假設(Discrete logarithm assumption)是假設離散對數問題(DLP)是一個在計算上很難解的問題。
- Diffie-Hellman判決問題(Decisional Diffie-Hellman Problem)，在這個問題中我們隨機選取兩個值 $X=g^x \pmod{p}$ ， $Y=g^y \pmod{p}$ ，其中 p 是一個安全質數，使得 $p=2q+1$ ， q 也是一個質數，而 g 是在 $G_{p,q}$ 這個群的原生根。這個問題是要去判別在 $G_{p,q}$ 這個群的一串數值 (X, Y, Z) 之中， Z 是一個隨機的值還是 X 與 Y 的Diffie-Hellman值 $DH(X, Y)=g^{xy}$ 。
- DDH 假設(DDH Assumption)是假設Diffie-Hellman判決問題是一個在計算上很難解的問題。

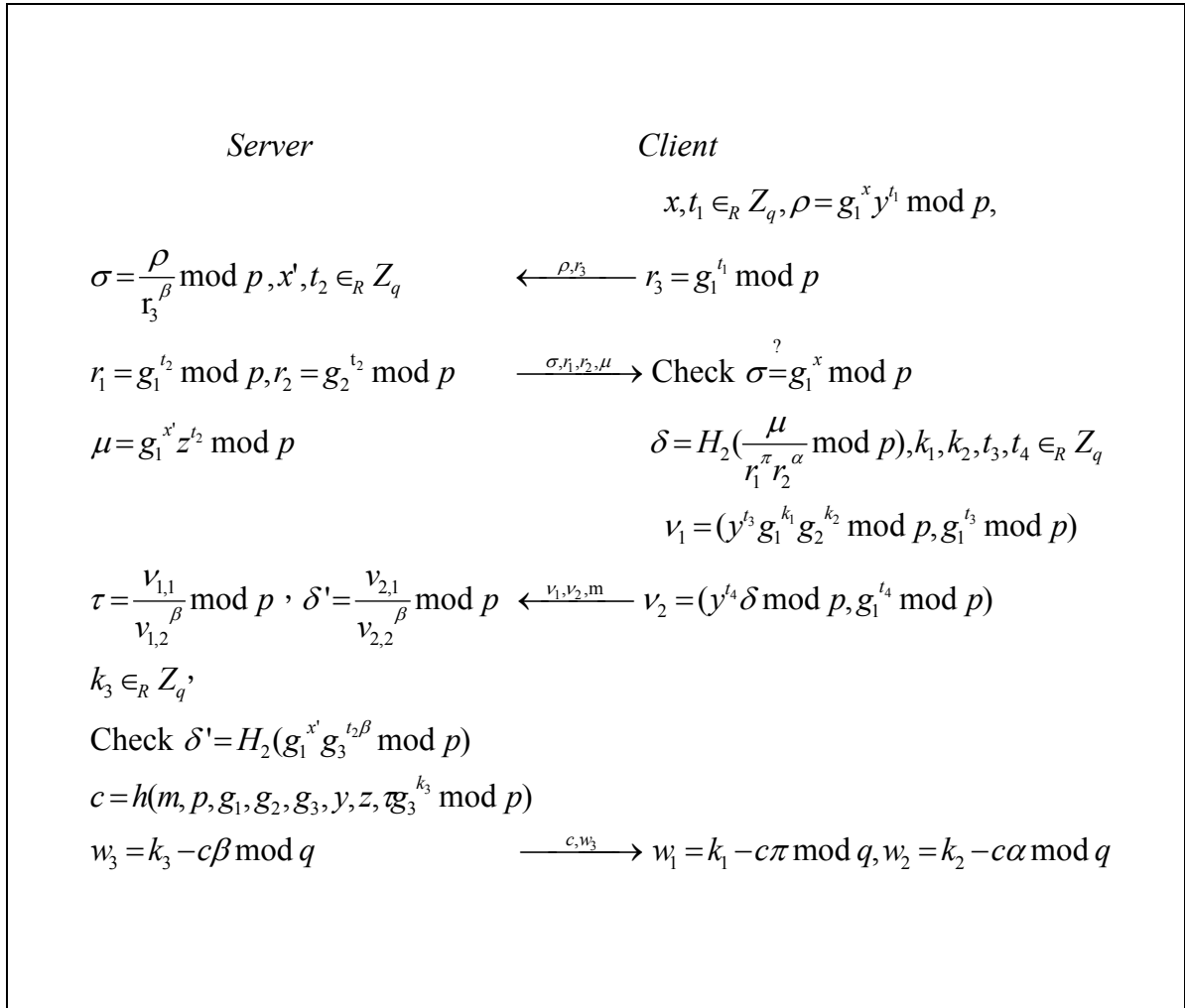
第二節 使用通行碼的數位簽章協定

在介紹我們所提出的“使用通行碼的數位簽章協定”PS之前，我們需要注意一件事：服務者所擁有的秘密資訊是 β ，而 β 是經過雜湊函數 H_1 雜湊後產生的值。我們提出的PS協定中包含了三個部分的演算法，分別是金鑰產生演算法，簽署數位簽章演算法，以及驗證數位簽章演算法，演算法的內容分別說明如下：

- KGen (金鑰產生演算法)：在這個演算法中，輸入是安全參數以及通行碼 (k, l, π) ，而輸出則是公開的值 p, q, g_1, g_2, g_3 以及公開金鑰

與私密金鑰值組 α, β, y, z ，其中 $p=2q+1$ ， $y = g_1^\beta \bmod p$ ，
 $z = g_1^\pi g_2^\alpha g_3^\beta \bmod p$ ，而 $\beta = H_1(\beta')$ ， g_1, g_2, g_3 則是 Z_p^* 中的原生根。

- Sign (簽署數位簽章演算法)：結合服務者秘密資訊、客戶秘密資訊、以及通行碼並且進行身份認證的簽署數位簽章演算法如下。



- 1、首先客戶端隨機從 Z_q 選取 x 及 t_1 ，並計算出
 $\rho = g_1^x y^{t_1} \bmod p$ ， $r_3 = g_1^{t_1} \bmod p$ 然後將 ρ 以及 r_3 傳遞給服務者要求進行身份驗證。

2、服務者收到客戶身份驗證的要求，利用服務者所擁有的秘密資訊 β 計算出 $\sigma = \frac{\rho}{r_3^\beta} \bmod p$ ，並且隨機由 Z_q 中隨機選取 x' 及 t_2 ，並且計算出 $\mu = g_1^{x'} z^{t_2} \bmod p$ ， $r_1 = g_1^{t_2} \bmod p$ ， $r_2 = g_2^{t_2} \bmod p$ ，然後將數值組 (σ, r_1, r_2, μ) 傳送給客戶。

3、客戶收到 (σ, r_1, r_2, μ) 之後，先驗證 $\sigma = g_1^x$ ，如果這兩個值不相等，則終止與服務者的身份驗證過程，若這兩個值相同，則計算 $\delta = H_2\left(\frac{\mu}{r_1^\alpha r_2^\alpha} \bmod p\right)$ ，接著再由 Z_q 中隨機選取 k_1, k_2, t_3, t_4 接著計算出 $v_1 = (y^{t_3} g_1^{k_1} g_2^{k_2} \bmod p, g_1^{t_3} \bmod p)$ 與 $v_2 = (y^{t_4} \delta \bmod p, g_1^{t_4} \bmod p)$ 的值，之後再將 v_1, v_2 以即要簽署的文件 m 傳送給服務者。

4、服務者收到 v_1, v_2, m 時，首先計算出 $\delta' = \frac{v_{2,1}}{v_{2,2}^\beta} \bmod p$ ，接著比較 $\delta' = H_2(g_1^{x'} g_3^{t_2 \beta} \bmod p)$ 如果等號不成立，則終止與客戶的身份驗證過程，如果等號成立，則計算 $\tau = \frac{v_{1,1}}{v_{1,2}^\beta} \bmod p$ 的值以及隨機由 Z_q 中選取 k_3 ，接著產生我們經過身份認證協定後的部分數位簽章 $c = h(m, p, g_1, g_2, g_3, y, z, \tau g_3^{k_3} \bmod p)$ 與 $w_3 = k_3 - c\beta \bmod q$ ，最後將數值組 (c, w_3) 傳送給客戶。

5、客戶收到 (c, w_3) 後，計算出 $w_1 = k_1 - c\pi \bmod q$ ， $w_2 = k_2 - c\alpha \bmod q$ 的值。

當上述的步驟都順利完成後，在客戶端就產生了一個結合服務者秘密

資訊，客戶秘密資訊，以及通行碼的數位簽章 (c, w_1, w_2, w_3) 。

- Verify (驗證數位簽章演算法)：這個驗證數位簽章的演算法的輸入是數位簽章 (c, w_1, w_2, w_3) ，以及公開金鑰 z ，驗證的方法是計算出 $c' = h(m, p, g_1, g_2, g_3, y, z, g_1^{w_1} g_2^{w_2} g_3^{w_3} z^c \text{ mod } p)$ 之後比對 $c' \stackrel{?}{=} c$ ，如果等號成立則輸出”有效”，否則輸出”無效”。

我們假設客戶端的行動裝置擁有一個秘密資訊 α 而使用者知道可記憶的通行碼 π ，而客戶擁有的秘密資訊是 β' (我們前面提過，對應於公開金鑰 y 以及 z 的私密金鑰 $\beta = H_1(\beta')$)，由於我們的目的是正確的客戶端與服務者端秘密資訊配合正確的通行碼才能產生合法的數位簽章，因此在產生數位簽章之前，客戶與服務者都必須確認對方的身份，在確認身份成功之後，才會產生出我們的數位簽章，身份認證的方法有許多種，大多數的通行碼身份驗證的協定是使用通行碼進行身份認證的動作，而我們的協定則是利用對應秘密資訊以及通行碼的公開金鑰來做認證的動作。

在簽署數位簽章的演算法中，我們可以分為幾個階段來看，首先是客戶要求確認服務者的身份，第二步則是服務者要求確認客戶的身份，第三步在前面兩個步驟都執行正確無誤，也就代表客戶與服務者的身份都正確，才會產生數位簽章。

在第一個步驟中，客戶如何確認服務者的身份呢？我們知道服務者所擁有的秘密資訊是 β' ，也就是說，服務者擁有 ElGamal 加密系統中公開金鑰 y 的對應私密金鑰，因此我們在客戶端隨機選一個訊息並且利用公開金鑰加密隨機選的訊息之後，交給服務者，如果服務者能解出正確的訊息，那麼表示他是正確的服務者，如果不能，那麼服務者便無法證明他的身份。在第二個步驟中服務者要求確認客戶的身份時，也是利用相同的概念，不過不同的地方在於，正確的客戶端所擁有的秘密資訊與通行碼相對

應的公開金鑰 z 還包含了服務者端的秘密資訊，所以我們在做這一個步驟的身份確認時，所驗證的不是單純的隨機訊息，而是經過部分解密後的訊息，也就是說，在服務者端隨機選了一個訊息並且利用 z 這個公開金鑰加密後傳送給客戶，而客戶在收到這個密文後，利用他所知到的秘密資訊及通行碼，可以將密文部分解密到剩下訊息以及服務者端秘密資訊的密文，而服務者端在收到這個部分解密過後的密文，再與同一個訊息卻只使用服務者端秘密資訊加密的結果作比較，如果相同則表示客戶端確實部分解密成功，也就是說客戶端的確擁有正確的秘密資訊以及通行碼，原因可以由這個式子來說明：

$$\frac{\mu}{r_1^\pi r_2^\alpha} \bmod p = \frac{g_1^{x'} \cdot z^{t_2}}{g_1^{t_2\pi} g_2^{t_2\alpha}} \bmod p = \frac{g_1^{x'} \cdot g_1^{t_2\pi} g_2^{t_2\alpha} g_3^{t_2\beta}}{g_1^{t_2\pi} g_2^{t_2\alpha}} \bmod p = g_1^{x'} \cdot g_3^{t_2\beta} \bmod p$$

由我們最後產生的數值組 (c, w_1, w_2, w_3) ，及驗證的方式可以知道這是知識簽章，也就是說這是一個零知識的非互動式證明系統，在這個零知識非互動式證明系統中，我們必須知道，證明者的身份是由客戶與服務者共同扮演，而驗證者則是所有需要驗證這個知識簽章的人，這樣的情況與一般的證明系統不一樣的地方在於，共同扮演證明者的客戶與服務者之間會有溝通的訊息，而這個訊息可能會造成無法滿足私密性的條件，因此我們使用了 ElGamal 的加密系統將產生知識簽章的過程中唯一需要溝通的訊息利用 y 加密後傳遞給服務者，因為服務者知道的值，所以可以解回正確的明文，而服務者再利用這個訊息來產生我們的知識簽章。

第四章 安全性分析

我們將安全性分成幾個部分來討論，第一個部分是針對在第二章中介紹過的攻擊方式來討論，為什麼這樣的攻擊方式無法攻擊我們的協定，第二個部分則是要證明我們提出的協定滿足了私密性與不可偽造性，要達到私密性的要求必須要求身份認證過程傳遞的訊息，產生數位簽章時傳遞的訊息，以及產生的數位簽章這三個部分都不會讓攻擊者取得秘密資訊或是通行碼的資訊。身份認證中傳遞的訊息我們利用理想領域與真實領域模式來證明我們提出的協定的 PS 在身份認證的過程中是安全的，證明 ElGamal 加密系統是安全的可以保證我們在產生數位簽章所需傳遞的訊息是安全的。最後則是證明 PS 產生的數位簽章是知識簽章，也就是零知識非互動式證明系統，這樣便滿足了私密性的要求，而不可偽造性也是利用數位簽章是知識簽章的證明來達到。



第一節 安全性定義與證明模式

在這一節中，我們介紹在 PAK 中所提出的證明模式，也就是 IdealWorld 以及 RealWorld 兩個系統，並且證明在 RealWorld 中，攻擊者所能進行的攻擊，在 IdealWorld 都可以藉由模擬器模擬出不可分辨的概觀，第三節會針對我們的協定定義證明需求。

第一項 理想領域的介紹

在理想領域中客戶及服務者的通行碼驗證與數位簽章都是透過可信任的第三者 TTP (Trust Third Party) 來完成，也就是說若客戶將通行碼 π 與

秘密資訊 α 透過絕對安全的通道傳送給 TTP，而服務者則是也將秘密資訊 β 透過絕對安全的通道傳送給 TTP，TTP 將這些秘密資訊與通行碼 π 和公開金鑰 z 做比較，確認無誤後，則產生一個數位簽章 (c, w_1, w_2, w_3) 給客戶，因此在理想領域下的通行碼驗證數位簽章協定是安全的。

我們假設在理想領域中有一個使用者的集合（包括了客戶與服務者），我們用 user i 來編號， $i=1,2,3,\dots$ ，因為每個客戶都只會有一個請求連線（Instance），所以 i 也代表了請求連線，因此我們將 (i) 稱之為使用者請求連線（user instance），一個使用者請求連線隱含了使用者在利用通行碼驗證的數位簽章協定中所扮演的角色（role），不是客戶，就是服務者。

接下來我們介紹攻擊者在理想領域所能做的幾種運作，這些運作都由 TTP 來處理產生一定的隨機值（Random value）給攻擊者，並確保這些隨機值的一致性，以避免被攻擊者識破，並假設 TTP 可以存取一個隨機二元字串（Random bit string）上面的值，此隨機二元字串稱為 R ，而成功的客戶要求連線所產生的數位簽章 (c, w_1, w_2, w_3) ，TTP 也都會將它記錄起來。另外攻擊者某些運作的結果都將收集起來，稱為謄本記錄（Transcript），而對於一個理想領域的攻擊者 A^* ， $\text{IdealWorld}(A^*)$ 是一個隨機變數（Random Variable），代表所有的謄本紀錄。

一、啟用使用者請求連線（Initialize user instance）—謄本記錄：“啟用使用者請求連線”， $i, \text{role}(i)$ ）：

攻擊者指定一個使用者請求連線 (i) ，並且指定其角色為 $\text{role}(i) = \{\text{客戶或服務者}\}$ ，例如：攻擊者可以啟用客戶 Alice（假設 Alice 的使用者編號是 2）的使用者要求連線 (2) ，且指定 (2) 的角色是客戶。

二、終止使用者請求連線（Terminate user instance）—謄本記錄：“終

止使用者 請求連線”， i)：

攻擊者要求終止之前的一個使用者請求連線。

三、猜測請求連線的通行碼 (Test instance password)：

攻擊者猜測一個使用者請求連線 (i) 的通行碼為 π ，如果猜對則回傳“正確”給攻擊者，猜測錯誤則回傳“不正確”，這個運作對於一個使用者請求連線只能執行一次，而且當下面的「啟動會議」執行之後也不能執行，主要是為了要模擬攻擊者用即時字典攻擊法來猜測通行碼。

四、啟用會議 (Start session) — 謄本記錄：(“啟用會議”， i)：

所謂的會議是指 TTP 去幫使用者之間的通行碼驗證與產生數位簽章的過程，當攻擊者對一個使用者請求連線 (i) 做這個運作時，攻擊者會先猜測該會議的“連線指定值 (connection assignment)”，連線指定值有以下兩種可能

1. 接受客戶的請求連線：這需要 (i) 這個使用者請求連線已經被啟用，之後 TTP 會產生一個數位簽章 (c, w_1, w_2, w_3) 給客戶。
2. 要求服務者給予連線：這需要 (i) 這個使用者請求連線已經被啟用，和接受客戶請求連線相同，TTP 會產生一個數位簽章 (c, w_1, w_2, w_3) 給客戶。

五、履行 (Implementation) — 謄本記錄：(“履行”，註釋)：

攻擊者可以在謄本記錄中放一些“註釋”，這是為了證明的需要而且不會影響到理想領域中的運作。

第二項 真實領域的介紹

在真實領域中 TTP 不協助通行碼驗證與簽章的進行，所以要進行通行

碼驗證與簽章時，都要靠客戶與服務者之間傳遞訊息來完成，與理想領域不同的是，每一個「使用者請求連線」都是一個狀態機(State machine)，可以接收一個訊息的輸入，並根據輸入的訊息來改變其狀態，狀態有下列三種

- 繼續：「使用者請求連線」(i) 準備接收下一個訊息。
- 接受：「使用者請求連線」(i) 結束身份認證過程，並且產生一組數位簽章 (c, w_1, w_2, w_3) 。
- 拒絕：「使用者請求連線」(i) 結束身份認證的過程，但是沒有產生數位簽章。

接下來我們介紹在真實領域中攻擊者的運作，對於一個真實領域的攻擊者 A， $RealWorld(A)$ 也是一個隨機變數，代表所有的謄本紀錄。

一、啟用使用者請求連線—謄本記錄：(“啟用使用者請求連線”，i，role (i))：

攻擊者指定一個使用者請求連線 (i)，並且指定其角色為 $role(i) = \{ \text{客戶或服務者} \}$ 。

二、傳送訊息—謄本記錄 (“履行”，“傳送訊息”，i，輸入訊息，回傳訊息，狀態)：

攻擊者傳送一個“輸入訊息給”「使用者請求連線」，使用者請求連線根據這個訊息改變自己的狀態，並輸出“回傳訊息”和目前的“狀態”給攻擊者。另外根據“狀態”來增加謄本記錄，如果狀態為「接受」，則紀錄 (“啟用會議”，i)，代表已經完成通行碼驗證與產生數位簽章的動作，如果狀態為「拒絕」，則記錄 (“終止使用者請求連線”，i)，代表認證失敗而且沒有產生數位簽章。

三、隨機聖賢(Random Oracle)—謄本記錄 (“履行”，“隨機聖賢”，x， $H_i(x)$)：

攻擊者可以詢問隨機聖賢 x 經過雜湊函數 H_i 的雜湊值 $H_i(x)$ ，

$i \in \{1,2\}$ 。值得我們注意的是，隨機聖賢模式是假設隨機聖賢所輸出的查詢值是隨機的，另外隨機聖賢會記錄下查詢過的輸入數值組與相對輸出值。

第三項 安全性定義

對於一個數位簽章系統而言，我們知道必須要滿足下面的三個特性：

- 一、正確性：只要擁有正確秘密資訊的客戶使用正確的通行碼便可以與正確的服務者產生合法的數位簽章。
- 二、私密性：對於攻擊者而言，在協定交換訊息的過程中，觀察公開資訊是無法得到秘密資訊及通行碼的資訊。
- 三、不可偽造性：攻擊者在不知道秘密資訊的情況下，無法偽造出一個合法的數位簽章。

關於私密性的部分，根據前面第一項的理想領域與第二項的真實領域的定義，一個安全的通行碼身份認證協定在身份認證的部分必須要滿足下面兩個條件：

1. 正確性：對於任意真實世界的使用者，只要正確的執行協定，則一定能完整並成功的認證。
2. 可擬性：對於任意的“以通行碼為基礎的身份認證的數位簽章協定”在真實世界中的攻擊者A，都存在著一個有效率的理想領域攻擊者 A^* ，使得 $\text{RealWorld}(A)$ 與 $\text{IdealWorld}(A^*)$ 這兩個隨機變數是計算上的不可分辨(Computationally indistinguishable)。

於是我們利用一個模擬器 (Simulator) 使得真實領域的狀態會等價於理想領域的狀態，我們在理想領域中造出這個模擬器，用來增加謄本記錄，只要在真實領域中存在的謄本記錄，我們都利用這個模擬器來產生相

同的謄本記錄，使得真實領域和理想領域的概觀是等價的，也就是 $\text{RealWorld}(A) \stackrel{c}{=} \text{IdealWorld}(A^*)$ 。

另外只要我們所產生的數位簽章是零知識非互動式證明系統，那麼表示我們產生的簽章不會洩漏任何有關秘密資訊或是通行碼的資訊，如此一來，代表我們的協定在溝通的過程中留下的訊息，以及溝通完後產生的數位簽章，都能達到私密性的要求，而不可偽造性也可以利用零知識非互動式證明系統來達成。

第二節 針對一般攻擊法的安全性討論

接下來我們將會針對在第二章所介紹過的攻擊方式來討論，為什麼這樣的攻擊方式對我們的 PS 協定無法產生效果，最後並加上兩個特殊情況的討論，也就是當客戶的秘密資訊洩漏，以及當服務者的秘密資訊洩漏時所可能產生的問題。

- 一、竊聽攻擊法：單純的竊聽攻擊法所得到的訊息有 $\rho, \sigma, \mu, r_1, r_2, r_3, v_1, v_2, c, w_3$ ，其中 $\rho, \sigma, v_1, v_2, r_1, r_2, r_3, w_3$ ，不包含通行碼 π 的資訊，而要從 μ 得到通行碼 π 的資訊等於由公開金鑰 z 去計算通行碼 π ，也就是說要破解 ElGamal 加密機制，也就是說，如果我們能解離散對數，那麼就可以破解 ElGamal 加密系統，但是根據 DH 假設，要解離散對數問題在計算上來講是很困難的。而如果要從 w_1 得到通行碼 π 的資訊，則必須先計算出 k_1 ，若要由 v_2 得到 k_1 的資訊，同樣的要能破解 ElGamal 加密機制，而若要由 c 得到 k_1 的資訊，則必須對 $g_1^{k_1}, g_2^{k_2}, g_3^{k_3}$ 解一個離散對數的問題，同樣地根據離散對數假設，要解一個離散對數的問題在計算上來講是很困難的，因此我們說可以抵擋竊聽攻擊法。

- 二、重送攻擊法：由於每次Client在做身份認證時，都會選擇不同的 t_i 值，如果直接使用 ρ ，及 r_3 來重送確認Server的身份時，在Server回送 σ, r_1, r_2 ，要確認Client的身份時，由於攻擊者不知道正確的 α 與 π 的值，所以無法回送利用ElGamal加密的正確 δ ，因此在認證Client身份時就會失敗。
- 三、字典攻擊法：對於即時的字典攻擊法，在Server確認Client端身份時，Client必須計算出 ρ ，但是要計算出 ρ 必須同時有 π 和 α 的資訊，所以即時的字典攻擊法無法成功，同樣地，對於離線的字典攻擊法也無法成功，攻擊者必須能解離散對數問題才能得到通行碼 π 的資訊。
- 四、Denny-Sacco 攻擊法：在這個攻擊法中，我們假設攻擊者取得了之前曾經產生過的數位簽章，並利用這個簽章來猜測出通行碼，然而這個攻擊法在我們的協定中是不可能的，因為我們產生的數位簽章是知識簽章，也就是零知識非互動式的證明系統，因此想藉由我們的數位簽章來猜測出通行碼是不可能的。
- 五、中間人攻擊法：如果攻擊者位於客戶與服務者之間，企圖同時扮演兩個角色是不可能的，因為我們的PS認證過程是利用公開金鑰來做認證，攻擊者沒有相對的私密金鑰，因此無法同時偽裝客戶與服務者。
- 六、小型子群限制：在我們的協定中所選取的 p 是一個安全的質數，也就是說除了基數為2和基數為 q 的兩個子群外，不會有其他的子群，因此只要檢查協定中的值避免落在較小的子群就可以抵擋小子群限制的攻擊方式。
- 七、行動裝置所擁有的秘密資訊 α 洩漏：若 α 洩漏，則可以利用公開的值計算出 $g_1^\pi \cdot g_3^\beta \bmod p$ ， $k_2 \bmod q$ ， $g_1^{k_1}, g_3^{k_3} \bmod p$ 的值，接著我們

能利用 $(g_1^\pi \cdot g_3^\beta)^c \bmod p$ 乘以 $g_3^{w_3} \bmod p$ 算出 $g_1^{c\pi} \cdot g_3^{k_3} \bmod p$ 如果要從前述的值計算出 π ，則需要找出 $g_3^{k_3} \bmod p$ 的值，才能利用字典攻擊法猜出 π 的值，但是要計算 $g_3^{k_3} \bmod p$ 的值，則一樣要對 $g_1^{k_1} \cdot g_2^{k_2} \cdot g_3^{k_3}$ 在 $G_{p,q}$ 下解離散對數的問題，所以如果行動裝置所擁有的秘密資訊 α 洩漏，一樣不能利用字典攻擊法找出通行碼 π 的值，另外在已知 α 的情況下依然無法偽造出合法的數位簽章，原因是我們的數位簽章是零知識非互動式證明系統，攻擊者對於服務者的秘密資訊 β 以及通行碼 π 仍然沒有任何的資訊。

八、服務者所擁有的秘密資訊 β' 洩漏：如果 β' 洩漏，首先攻擊者查詢 H_1 這個雜湊函數找出 β ，則所有利用 y 當作公開金鑰加密的資訊，都可以解密得到，所以 $g_1^\pi \cdot g_2^\alpha \bmod p$ ， $k_3 \bmod q$ ， $g_1^{k_1} \cdot g_2^{k_2} \bmod p$ ， ρ 及 $r_1^\pi \cdot r_2^\alpha \bmod p$ 都可以計算，若由七中相同的方式要猜出通行碼 π ，則必須要計算出 $g_2^{k_2} \bmod p$ 的值，相同的，要計算 $g_2^{k_2} \bmod p$ 的值，必須在 $G_{p,q}$ 解 $g_1^{k_1} \cdot g_2^{k_2} \cdot g_3^{k_3}$ 離散對數的問題。另外，如果要利用 $r_1^\pi \cdot r_2^\alpha \bmod p$ 來找出通行碼 π ，那麼必須要知道 α 才能利用字典攻擊法找出通行碼 π ，也就是說要知道 α 必須要能夠由公開金鑰 z 中計算出 α 的值，根據離散對數假設我們可以知道要找出 α 是計算上很難的問題，因此即使秘密資訊 β' 洩漏，我們的系統依然是安全的，而知道秘密資訊 β' 也無法偽造出合法的數位簽章，原因與七無法偽造數位簽章的原因相同。

第三節 安全性證明

在上一節我們討論了各種攻擊法來分析為什麼無法成功攻擊PS的原因，但是攻擊的方式不斷的在更新與改變，因此單純的討論是無法滿足我們所需求的安全性的，所以本節將對我們的PS分三個步驟，做一個正規化的證明，第一個步驟是要證明 $\text{RealWorld}(A)$ 與 $\text{IdealWorld}(A^*)$ 是等價的，第二個步驟則是要證明在傳遞訊息中利用ElGamal加密系統是安全的，第三步則是證明我們的數位簽章是知識簽章。

第一項 理想世界與真實世界的不可分辨

正如第一節中所定義的，我們的協定對於簽章的私密性而言必須滿足兩個性質，首先我們先來看正確性，我們直接檢視協定的運作，便可以發現他滿足了正確性的要求。而在可擬性方面，我們需要做的就是理想領域中造出一個模擬器，使得 $\text{RealWorld}(A) \stackrel{c}{=} \text{IdealWorld}(A^*)$ ，也就是說針對攻擊者A在真實領域中所有可能得到的謄本記錄，我們的模擬器必須對理想領域中的攻擊者 A^* 造出等價的謄本記錄，並且維持一致性。

讓我們回想之前對真實領域中與理想領域中攻擊者運作的定義，我們可以發現在真實領域中的「傳遞訊息」以及「隨機聖賢」是在理想領域中不存在的，因此我們必須要利用理想領域中的“履行”來造出相同的謄本記錄，至於真實領域其他的運作都可以由理想領域的運作造出相同的謄本記錄。在接下來的過程中，我們會造出模擬器，一旦無法造出真實領域的謄本記錄，我們可以證明這個謄本記錄不會發生，否則我們可以造出一個演算法來解決 DDH 問題。

我們定義攻擊者在真實領域所的傳遞訊息運作

- 一、 A0：啟用一個使用者要求連線 (i)，且送出一個回傳訊息(ρ, r_3)
- 二、 B1(ρ, r_3)：服務者收到傳送訊息 A0 且送出回傳訊息(r_1, r_2, σ, μ)
- 三、 A2(r_1, r_2, σ, μ)：客戶收到傳送訊息(r_1, r_2, σ, μ)，且送出回傳訊息(v_1, v_2)

模擬器運作方式如下

1. A0

從 Z_q 中隨機選取 d_1, d_2 ，並送出 $r_3 = g_1^{d_1} \bmod p, \rho = g_1^{d_2} y^{d_1} \bmod p$ 。

2. B1(ρ, r_3)

隨機產生 $e_1, e_2 \in Z_q, r_1 = g_1^{e_1} \bmod p, r_2 = g_2^{e_1} \bmod p, \mu = g_1^{e_2} z^{e_1} \bmod p$ ，

並且送出 r_1, r_2, σ, μ ，接著我們根據(ρ, r_3)來討論以下兩種情況

狀況 1. 如果 A0 是由客戶所傳送

使 $\sigma = \rho / y^{d_1} \bmod p$ 。

狀況 2. 如果 A0 不是由客戶所傳送

由於在協定中 x, t_1 的值每次都不一樣，因此每次 B1 的輸出 r_1, r_2, σ, μ 都不相同，所以不必擔心一致性的問題。

3. A2(r_1, r_2, σ, μ)

隨機產生 $u_1, u_2, k_1, k_2, f_1, f_2 \in Z_q, \delta = \delta' \in_R G_{p,q}$ 並且計算

$v_1 = (g_1^{f_1} g_2^{f_2} \cdot y^{u_1} \bmod p, g_1^{u_1} \bmod p), v_2 = (\delta \cdot y^{u_2} \bmod p, g_2^{u_2} \bmod p)$ ，接著我

們根據 r_1, r_2, σ, μ 來討論以下兩種狀況

• 狀況 1. 如果 r_1, r_2, σ, μ 是由服務者傳送

設定 $\delta' = H(R(\frac{\mu}{r_1^\pi r_2^\alpha}))$ ，也就是我們隨機選一個值，作為雜湊函數 H

的輸入，並且設定其輸出為 δ' 。

- 狀況 2. 如果 r_1, r_2, σ, μ 不是由服務者傳送
在協定中 $x, k_1, k_2, t_2, t_3, t_4$ 每次的值都不一樣，因此每次 A2 的輸出 v_1, v_2 都不相同，所以不需要擔心一致性的問題。

接著我們介紹模擬器如何處理隨機聖賢的運作：

1. $H_1(\beta')$

- 狀況 1. 如果 β' 曾經被查詢過
根據隨機聖賢的紀錄輸出記錄中的 β 。
- 狀況 2. 如果 β' 沒有被查詢過
產生 $\beta \in_R Z_q$ 並且回傳。

2. $H_2\left(R\left(\frac{\mu}{r_1^\pi r_2^\alpha}\right)\right)$

- 狀況 1. 如果 $R\left(\frac{\mu}{r_1^\pi r_2^\alpha}\right)$ 曾經被查詢過
回傳當時回傳的 δ' 。
- 狀況 2. 如果 $R\left(\frac{\mu}{r_1^\pi r_2^\alpha}\right)$ 沒有被查詢過
產生 $\delta' \in_R G_{p,q}$ 並回傳。



接下來我們要證明當攻擊者執行 $A2(r_1, r_2, \sigma, \mu)$ 時，攻擊者不會詢問隨機聖賢 $H_2\left(R\left(\frac{\mu}{r_1^\pi r_2^\alpha}\right)\right)$ 的值，否則我們可以造出一個分辨器 D 來解決 DDH 問題，分辨器的造法如下：

對於一數值組 (X, Y, Z) ，我們要判別 Z 是否等於 $DH(X, Y)$ 。

1. 我們將 PS 系統選取的 g_1, g_2, g_3 關係設為

$$g_2 = g_1^m \bmod p, g_3 = g_1^n \bmod p, \text{ 因此我們的公開金鑰}$$

$$z = g_1^{\pi+m\alpha+n\beta} \bmod p.$$

2. 我們將公開金鑰的值設為 Z ，在 $B1(\rho, r_3)$ 這個動作中，將 r_1 設為 Y 而 μ 則設為 $g_1^{x'}C$ 並且回傳。
3. 如果攻擊者詢問 $H_2\left(R\left(\frac{\mu}{r_1^\pi r_2^\alpha}\right)\right)$ 的值，那麼，根據隨機聖賢在攻擊者查詢時的紀錄，我們可以找到攻擊者查詢 $R\left(\frac{\mu}{r_1^\pi r_2^\alpha}\right)$ 的值。
4. 比對 $R\left(\frac{\mu}{r_1^\pi r_2^\alpha}\right)$ 是否等於 $g_1^{e_2} g_1^{ne_1\beta} \bmod p$ ，如果等號成立，則輸出“True” 否則輸出 “False”。

我們假設攻擊者能夠查詢 $R\left(\frac{\mu}{r_1^\pi r_2^\alpha}\right)$ 的機率是不可以忽略的值 ε ，那麼我們所造出的辨別器 D 能夠成功分辨DDH數值組的機率為 $(1/2 + \varepsilon)$ ，因此我們知道攻擊者是不會去詢問 $R\left(\frac{\mu}{r_1^\pi r_2^\alpha}\right)$ 的雜湊值，在完成了模擬器與這個辨別器之後，已經使得 $\text{IdealWorld}(A) = \text{RealWorld}(A^*)$ ，也就是說如果真實領域的攻擊者 A 可以得到通行碼，那麼理想領域的攻擊者 A^* 也可以得到通行碼，這與我們理想領域是絕對安全的定義相違背，所以我們可以知道我們的PS的身份認證在真實領域中是安全的。

第二項 ElGamal 加密系統的安全性

接下來我們將會證明 ElGamal 加密系統的安全性，證明的方式是假設攻擊者可以將利用 ElGamal 加密的密文解密回明文，那麼就可以利用這個攻擊者來解 Diffie-Hellman 判決問題。

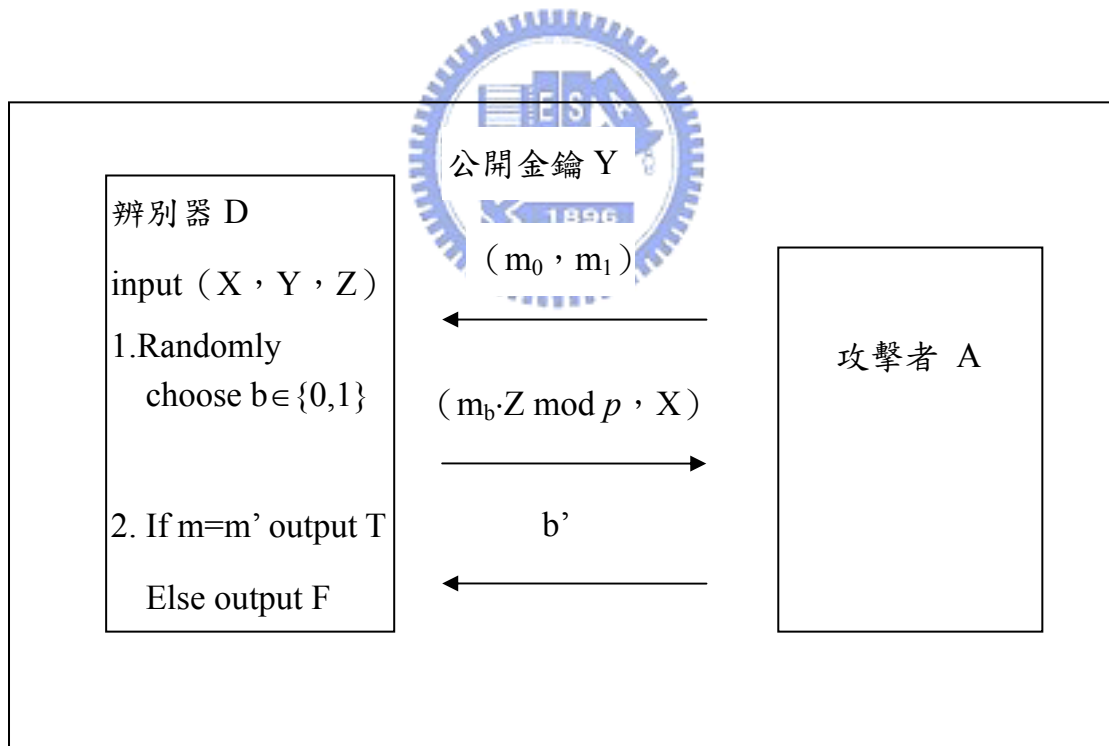
ElGamal 加密系統包含了三個部分的演算法，分別是 KGen、Enc、Dec 分別介紹如下：

- $\text{KGen}(1^n) = ((p, g, y), (p, g, x))$ ：這個演算法的輸入是安全參數 n ，在這個演算法中，首先我們隨機選取一個長度為 n 個位

元的安全質數 p ，接著計算出在 Z_p^* 中的原生根 g ，然後隨機由 Z_{p-1} 中選取私密金鑰 x ，並且計算 $y=g^x \bmod p$ 。最後輸出金鑰配對 $((p, g, y), (p, g, x))$ 。

- $\text{Enc}((p, g, y), m) = (g^k \bmod p, m \cdot y^k \bmod p)$ ：其中訊息 m 是 Z_p^* 這個群的元素，而 k 則是隨機由 Z_{p-1} 所選出。
- $\text{Dec}((p, g, x), (r, s)) = s / r^x \bmod p$ ：解密的方式是將 s 除以 r 的 x 次方，因為 $s / r^x \bmod p = m \cdot y^k / g^{xy} \bmod p = m \bmod p$ ，這樣我們就可以計算出我們的明文 m 。

EaGamal 加密系統的安全性是建立在 DDH 假設上，我們說對於一個攻擊者 A 在不知道私密金鑰的情況下， A 是無法辨別加密過後的明文的，否則我們可以造出一個辨別器 D 來解 DDH 問題，辨別器 D 如下圖所示：

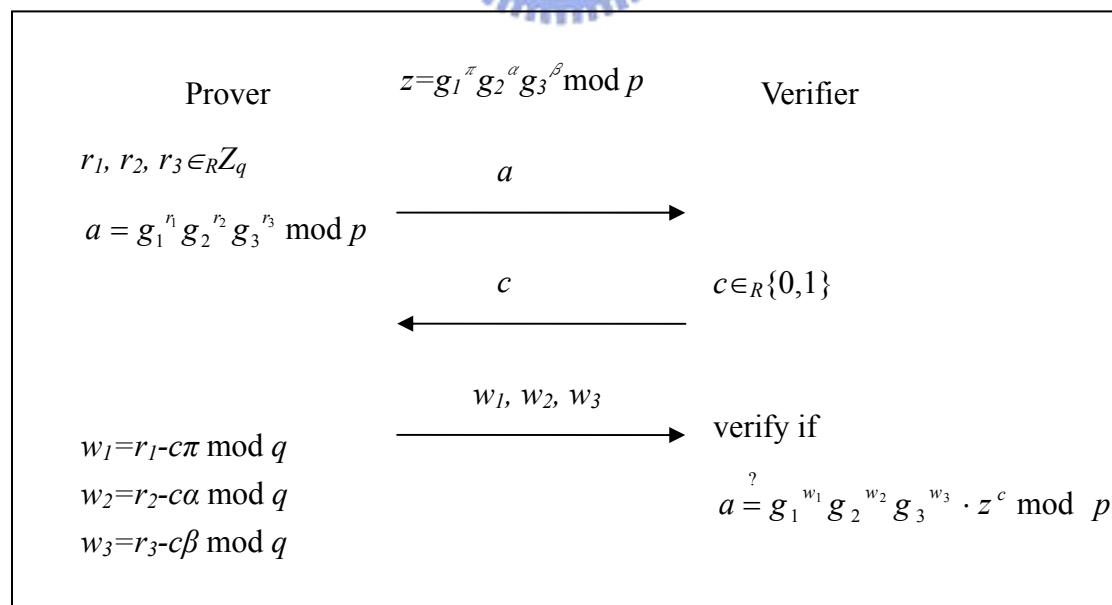


我們要判別的DH值是 (X, Y, Z) ，在這裡我們將把攻擊者 A 當作一個子程式來呼叫。首先我們將ElGamal加密系統中的公開金鑰設為 Y ，攻擊者選取兩個明文 m_0 與 m_1 ，接著我們由 $\{0,1\}$ 中隨機選取一個值 b ，並且將

$(m_b \cdot Z \bmod p, X)$ 當作密文傳送給攻擊者，此時攻擊者會輸出 b' ，我們比較 b 與 b' 的值，如果兩者相等，那麼表示 (X, Y, Z) 這數值組中的 Z 等於 $\text{DH}(X, Y)$ ，否則表示 Z 是一個隨機的值。假設攻擊者 A 能夠成功辨別加密過後的訊息的機率是一個不可忽略的值 ϵ ，那麼我們所造出的 D 正確地解 DDH 問題的機率就是 $(1/2 + \epsilon)$ 。

第三項 數位簽章的安全性

如果我們能證明我們的數位簽章是知識簽章也就是零知識非互動式證明系統，那麼便可以達到私密性以及不可偽造性的要求。在這裡我們先將我們協定所產生的數位簽章轉換為零知識互動式證明系統並且證明之，接著再利用 U. Feige、A. Fiat 與 A. Shamir 在 [18] 中所提出的方式，將這個互動式證明系統轉換為利用雜湊函數取代挑戰的零知識非互動式證明系統，也就是經由我們協定所產生的數位簽章。



$$\langle P, V \rangle (p, g_1, g_2, g_3, z)$$

我們假設證明者 P 擁有 π 、 α 、 β 這三個秘密資訊，而驗證者 V 要利用 $z = g_1^\pi g_2^\alpha g_3^\beta \bmod p$ 來驗證 P 確實知道 π 、 α 、 β 這三個秘密資訊，我們將下圖的零知識互動式證明系統稱之為 $\langle P, V \rangle(p, g_1, g_2, g_3, z)$ 。在這個零知識互動式證明系統 $\langle P, V \rangle(p, g_1, g_2, g_3, z)$ 中，證明者要說服驗證者他的確知道 π 、 α 、 β 這三個秘密資訊，而要驗證者被說服，那麼 $\langle P, V \rangle(p, g_1, g_2, g_3, z)$ 必須執行 $q(n)$ 次，而且每次的結果都是輸出接受才表示驗證者被證明者說服他的確知道秘密資訊的值。

$\langle P, V \rangle(p, g_1, g_2, g_3, z)$ 運作方式說明如下：

1. 首先證明者隨機由 Z_q 中選取，計算出 $a = g_1^{r_1} g_2^{r_2} g_3^{r_3} \bmod p$ ，並且將 a 傳送給驗證者。
2. 驗證者隨機設定 $c \in_R \{0, 1\}$ 並且回傳給證明者。
3. 證明者計算出 $w_1 = r_1 - c\pi \bmod q$ ， $w_2 = r_2 - c\alpha \bmod q$ ， $w_3 = r_3 - c\beta \bmod q$ 並且回傳給驗證者。
4. 驗證者確認 $a = g_1^{w_1} g_2^{w_2} g_3^{w_3} \cdot z^c \bmod p$ 等號是否成立，如果等號成立那麼輸出接受，否則輸出 \perp 。

在證明 $\langle P, V \rangle(p, g_1, g_2, g_3, z)$ 是一個零知識互動式證明系統之前，有幾點是我們需要注意的，首先，對於驗證者而言，我們可以回溯 (Rewind) 到驗證者驗證的某一步驟，回溯可以利用紀錄所有步驟的交換訊息來達成，另外我們說零知識的模擬是黑盒子模擬 (Black Box Simulation)，也就是說模擬器將驗證者視為黑盒子，我們不能將驗證者拆解而去瞭解其內部運作的方式。

根據第二章所介紹的零知識互動式系統我們可以知道，要證明

$\langle P, V \rangle (p, g_1, g_2, g_3, z)$ 是一個零知識的互動式證明系統必須要滿足三個條件，分別是完整性，完美性，以及零知識，完整性的部分很明顯的可以看出如果 P 確實知道這三個秘密資訊，那麼他執行這個互動式證明系統，就一定可以說服驗證者 V 他知道這三個資訊。

而完美性方面我們需要造出一個知識擷取者 E 來計算秘密值 π, α, β ，假設 $\langle P^*, V \rangle (p, g_1, g_2, g_3, z)$ 的概觀是：

$$((A_1, C_1, W_{1,1}, W_{1,2}, W_{1,3}), (A_2, C_2, W_{2,1}, W_{2,2}, W_{2,3}), \dots, (A_{q(n)}, C_{q(n)}, W_{q(n),1}, W_{q(n),2}, W_{q(n),3}))$$

其中， $(A_i, C_i, W_{i,1}, W_{i,2}, W_{i,3})$ 是第 i 次執行時的概觀，那麼我們的知識擷取者 E 如下：

1. 執行 $\langle P^*, V \rangle (p, g_1, g_2, g_3, z)$ 得到一個執行的概觀實體

$$((a_1, c_1, w_{1,1}, w_{1,2}, w_{1,3}), (a_2, c_2, w_{2,1}, w_{2,2}, w_{2,3}), \dots, (a_{q(n)}, c_{q(n)}, w_{q(n),1}, w_{q(n),2}, w_{q(n),3}))$$

2. 隨機選取一個 j 值， $1 \leq j \leq q(n)$ ，接著回溯到第 j 次執行的時間點，也就是在 $(a_1, c_1, w_{1,1}, w_{1,2}, w_{1,3}), (a_2, c_2, w_{2,1}, w_{2,2}, w_{2,3}), \dots, a_j$ 被產生的時候。這時候我們將 c_j 設為其補數 c_j' ，並且由這個時間點繼續執行 $\langle P^*, V \rangle (p, g_1, g_2, g_3, z)$ ，而得到之後執行的實體：

$$(a'_j, c'_j, w'_{j,1}, w'_{j,2}, w'_{j,3}), \dots, (a'_{q(n)}, c'_{q(n)}, w'_{q(n),1}, w'_{q(n),2}, w'_{q(n),3})$$

在第二步中我們選取的 j 有一個要值得注意的地方，就是 j 必須選在計算樹 T 的重 (heavy) 階層 (level)，否則 $\langle P^*, V \rangle (p, g_1, g_2, g_3, z)$ 會輸出接受的機率是可以忽略的值，如果以 N_i 來表示 T 在第 i 個階層的點 (node) 個數，

那麼重階層就是指 $N_{i+1} \geq 0.75N_i$ 。我們的知識擷取者 E 完成第一步的機率是 $1/p(n)$ ，而在第二步中，我們選的 l 確實落在重階層的機率是 $1/q(n)$ ，在設定 c_l' 之後，能成功走到驗證者接受的結果的機率至少是 $1/p(n)$ ，所以我們得到下面兩個執行實體

$$(a_1, c_1, w_{1,1}, w_{1,2}, w_{1,3}), \dots, (a_j, 0, w_{j,1}, w_{j,2}, w_{j,3}), \dots,$$

$$(a_{q(n)}, c_{q(n)}, w_{q(n),1}, w_{q(n),2}, w_{q(n),3})$$

以及

$$(a_1, c_1, w_{1,1}, w_{1,2}, w_{1,3}), \dots, (a_j, 1, w_{j,1}, w_{j,2}, w_{j,3}), \dots,$$

$$(a'_{q(n)}, c'_{q(n)}, w'_{q(n),1}, w'_{q(n),2}, w'_{q(n),3})$$

的機率是 $1/(q(n)p(n)^2)$ 。因此我們有 $1/(q(n)p(n)^2)$ 的機率可以計算出 $\pi = w_{j,1} - w'_{j,1} \pmod q$ ， $\alpha = w_{j,2} - w'_{j,2} \pmod q$ ，以及 $\beta = w_{j,3} - w'_{j,3} \pmod q$ ，接著利用重複試驗的方式，便可以將成功的機率提升到 $1-2^{-n}$ 。

最後我們需要滿足零知識的要求，也就是說我們必須造出一個模擬器來模擬這個互動式證明系統的概觀，使得模擬的概觀與執行 $\langle P^*, V \rangle(p, g_1, g_2, g_3, z)$ 的概觀是不可分辨的，假設 r 是長度為 $q(n)$ 的隨機位元字串 (bit string)，而對於所有的驗證者 V^* 與 P 互動的概觀是 (a, c, w_1, w_2, w_3) ，根據觀察，我們的可以知道 c 的值是根據 z 、 r 、 a 來決定，也就是說 $V^*(p, g_1, g_2, g_3, z, r, a) = c$ ，因此對於固定的 $(r_0, a_0, c_0, w_{0,1}, w_{0,2}, w_{0,3})$ 其中 $(a_0, c_0, w_{0,1}, w_{0,2}, w_{0,3})$ 滿足 $a_0 = g_1^{w_{0,1}} g_2^{w_{0,2}} g_3^{w_{0,3}} \cdot z^{c_0} \pmod p$ ，我們可以知道

$$\begin{aligned} \Pr[(R, A, C, W_1, W_2, W_3) = (r_0, a_0, c_0, w_{0,1}, w_{0,2}, w_{0,3})] \\ = \frac{\|\tilde{V}^*(p, g_1, g_2, g_3, z, r_0, a_0) = c_0\|}{2^{q(n)} q} \end{aligned}$$

而我們的模擬器 M^* 運作方式如下：

1. 隨機選取一個位元字串 r_0 。
2. 隨機由 $G_{p,q}$ 中選取 (w_1', w_2', w_3') ，並且隨機選取一個位元 $c' \in \{0,1\}$ 接著計算出 $a' = g_1^{w_1'} g_2^{w_2'} g_3^{w_3'} \cdot z^{c'} \bmod p$ 。
3. 模擬 $V^*(p, g_1, g_2, g_3, z, r, a) = c''$ ，如果 $c'' = c'$ ，那麼就輸出 $(r', a', c'', w_1', w_2', w_3')$ ，否則輸出 \perp 。

對於在第二步中的 r_0, a_0 ，而言， V^* 並不知道 M^* 選擇的 c' 也就是說

$$\begin{aligned}
 & \Pr_{w_1, w_2, w_3, C''} [C'' = \tilde{V}^*(p, g_1, g_2, g_3, r', z, g_1^{w_1'} g_2^{w_2'} g_3^{w_3'} \cdot z^{C''} \bmod p)] \\
 &= \frac{1}{2} \cdot \frac{1}{q} \cdot \frac{1}{q} \sum_{w_1', w_2', w_3'} (\| \tilde{V}(p, g_1, g_2, g_3, r', z, g_1^{w_1'} g_2^{w_2'} g_3^{w_3'} \cdot z^0 \bmod p) = 0 \| \\
 & \quad + \| \tilde{V}(p, g_1, g_2, g_3, r', z, g_1^{w_1'} g_2^{w_2'} g_3^{w_3'} \cdot z^1 \bmod p) = 1 \|) \\
 &= \frac{1}{2} \cdot \frac{1}{q} \cdot \frac{1}{q} \cdot q \sum_{a'} (\| \tilde{V}(p, g_1, g_2, g_3, r', z, a') = 0 \| \\
 & \quad + \| \tilde{V}(p, g_1, g_2, g_3, r', z, a') = 1 \|) \\
 &= \frac{1}{2} \cdot \frac{1}{q} \cdot q = \frac{1}{2}
 \end{aligned}$$

因此對於固定的 $(r_0, a_0, c_0, w_{0,1}, w_{0,2}, w_{0,3})$ ，我們利用執行模擬器的步驟模擬出來相同概觀的機率為

$$\begin{aligned}
 & \Pr[(R', A', C', W_1', W_2', W_3') = (r_0, a_0, c_0, w_{0,1}, w_{0,2}, w_{0,3}) \mid \text{succed}] \\
 &= 2 \cdot \sum_{c''} (\Pr[R' = r_0, W_1' = w_{0,1}, W_2' = w_{0,2}, W_3' = w_{0,3}, C'' = c'']) \cdot \\
 & \quad \| a_0 = g_1^{w_{0,1}} g_2^{w_{0,2}} g_3^{w_{0,3}} \cdot z^{c''} \bmod p, \tilde{V}^*(p, g_1, g_2, g_3, z, r_0, a_0) = c_0, c'' = c_0 \|) \\
 &= \frac{1}{q(n)} \cdot \frac{1}{q} \| a_0 = g_1^{w_{0,1}} g_2^{w_{0,2}} g_3^{w_{0,3}} \cdot z^{c_0} \bmod p, \tilde{V}^*(p, g_1, g_2, g_3, z, r_0, a_0) = c_0 \| \\
 &= \frac{1}{q(n)} \cdot \frac{1}{q} \| \tilde{V}^*(p, g_1, g_2, g_3, z, r_0, a_0) = c_0 \|
 \end{aligned}$$

而且很明顯的， M^* 的執行時間 (Runtime) 是被限制在多項式時間

(Polynomial time bounded) 內的。所以我們的 $\langle P, V \rangle (p, g_1, g_2, g_3, z)$ 是一個零知識互動式證明系統，接下來我們對 $\langle P, V \rangle (p, g_1, g_2, g_3, z)$ 做一些修改成為下圖的形式，接著再根據 U. Feige、A. Fiat 與 A. Shamir 在 [18] 中所提出的方式，我們便可以將轉換成零知識非互動式證明系統：

Input (p, g_1, g_2, g_3, z)

1. $h : \{0,1\}^* \rightarrow \{0,1\}^n$ 是一個抗碰撞雜湊函數
2. P 計算 $a_i = g_1^{k_1} g_2^{k_2} g_3^{k_3} \bmod p, k_i \in_R \mathbb{Z}_q, 1 \leq i \leq n$
3. P 計算 $c_1 c_2 \dots c_n = h(p, g, z, a_1, a_2, \dots, a_n)$ 而且
 $w_{i,1} = k_{i,1} - c_i \pi \bmod q, w_{i,2} = k_{i,2} - c_i \alpha \bmod q, w_{i,3} = k_{i,3} - c_i \beta \bmod q,$
 $1 \leq i \leq n$
4. P 輸出 $(a_1, a_2, \dots, a_n, w_{1,1}, w_{1,2}, w_{1,3}, w_{2,1}, w_{2,2}, w_{2,3}, \dots, w_{n,1}, w_{n,2}, w_{n,3})$ 作為 $z = g_1^\pi, g_2^\alpha, g_3^\beta \bmod p$ 的非互動式證明系統

非互動式證明系統

根據 U. Feige、A. Fiat 與 A. Shamir [18]，我們可以知道經由 PS 協定產生的數位簽章是知識簽章，也就是零知識的非互動性證明系統。

由上述的證明，我們可以確定我們的數位簽章是知識簽章，所以不會透露出任何關於 π 、 α 、 β 的資訊，即使知道了其中一個，仍然無法取得另外兩個的資訊，因此在第一節所討論的客戶行動裝置秘密資訊洩漏，或是服務者遭攻陷使得秘密資訊 β 洩漏，在這兩種情況下，攻擊者依然無法偽造出合法的數位簽章，因為除了洩漏的資訊外，我們的簽章對於另外兩個秘密資訊而言，仍然是零知識非互動式證明系統，也因此滿足私密性與不

可偽造性這兩個性質。

我們將協定的安全性分成三個步驟討論，經由第一項第二項以及第三項的證明，我們可以確定我們的 PS 協定是安全的無論是在做身份確認的動作、產生數位簽章過程或是最後產生出的數位簽章，都可以滿足我們所定義的私密性以及不可偽造性的要求，因此我們說我們的協定 PS 是安全無虞的。



第五章 結論與未來工作方向

在本篇論文中，我們根據行動裝置以及網路的發達，研究了如何使用通行碼在行動裝置上簽署數位簽章，而且當行動裝置遭惡意攻擊者竊取時，攻擊者無法找出通行碼也無法偽造出合法的數位簽章，為了要達到這樣的目的，我們利用網路發達的優勢，將行動裝置的秘密資訊，以及客戶的可記憶通行碼與服務者的秘密資訊結合在一起來產生一個數位簽章。由於我們希望正確的使用者擁有其行動裝置才能產生數位簽章，因此在產生數位簽章之前必須要對客戶與服務者的身份進行驗證，驗證通過才產生出數位簽章，而在身份驗證的過程中可能會遭遇到許多的問題，例如：利用字典攻擊法來猜測出通行碼。此外我們還對我們的數位簽章給予了一個正規化的證明，包括了利用 PAK 所提出的證明模式對我們協定中身份認證的部分做了一個正規化的證明、傳遞訊息時所需要的 ElGamal 加密系統的安全性，以及我們的 PS 協定所產生的數位簽章的安全性證明。

在未來的工作方向上，希望能對簽章的性質做一些變化，例如前進式安全 (Forward Secure) 或者是將我們簽章用的秘密資訊以及通行碼改造成金鑰獨立式 (Key Isolated) 的模式，以更符合實際上安全性的要求，並且能對這樣的協定提出一個更完整的證明模式來證明協定的安全性。

第六章 參考文獻

- [1] S. Bellovin and M. Merritt, “Encrypted key exchange : password-based authenticated key exchange,” *Proceedings of the IEEE Symposium on Security and Privacy*, pp.72-84 1992.
- [2] S. Bellovin and M. Merritt, “Augmented encrypted key exchange : a password-based protocol secure against dictionary attacks and password-file compromise,” *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pp.244-250, 1993.
- [3] M. Boyarsky, “Public-key cryptography and password protocols : the multi-user case,” *Proceedings of the 6th ACM Conference on Computer and Communications Security*, pp63-72 1999.
- [4] V. Boyko, P. MacKenzie and S. Patel, “Provable secure password authenticated key exchange using Diffie-Hellman,” *EUROCRYPT'00*, LCNS1807, pp.156-171, 2000.
- [5] D. Jablon, “Strong password-only authenticated key exchange,” *ACM SIGCOMM Computer Communication Review*, vol.26, no.5, pp.5-26, 1996.
- [6] T. Kwon and J. Song, “Authentication and key agreement via memorable passwords,” *ISOC Network and Distributed System Security Symposium*, 2001.
- [7] M. Lomas, L. Gong, J. Saltzer and R. Needham, “Reducing risks from poorly chosen keys,” *Proceedings of the twelfth ACM symposium on Operating systems principles*, pp.14-18, 1989.

- [8] S. Lucks, "Open key exchange : how to defeat dictionary attacks without encrypting public keys," *Proceedings of the 5th International Workshop on Security Protocols*, pp.79-90, 1997.
- [9] P. MacKenzie and R. Swaminathan, "Secure network authentication with password identification," *manuscript*.
- [10] M. Steiner, G. Tsukil and M. Waidner, "Refinement and extension of encrypted key exchange" *ACM SIGOPS Operating System Review*, vol.29, no.3, pp.22-30, 1995.
- [11] Y. Ding and P. Hoster, "Undetectable on-line password guessing attacks," *ACM SIGOPS Operating Systems Review*, vol.29, no.4, pp.77-86,1995.
- [12] P. MacKenzie, T. Shrimpton, and M. Jakobsson, "Threshold Password-Authenticated Key Exchange," *Proceedings of CRYPTO'02*, LNCS2442, pp.385-400, 2002.
- [13] J. Kats, R. Ostrovsky and M. Yung "Efficient password-authenticated key exchange using human-memorable passwords," *EUROCRYPT'01*, LNCS2045, pp.475, 2001.
- [14] Goldreich and Y. Lindell, "Session key generation using human password only," *CRYPTO'01*, LNCS2139, pp.408-432, 2001.
- [15] R. Gennaro and Y. Lindell, "A framework for password-based authenticated key exchange" *EUROCRYPT'03*, LNCS2656, pp.524-543, 2003.
- [16] S. Patel, "Number theoretic attacks on secure password schemes," *Proceedings of the 1997 IEEE Symposium on Security and Privacy*, pp.236-248, 1997.

- [17] O. Goldreich, "Foundations of cryptography : basic tools," Cambridge University Press, 2001.
- [18] U. Feige, A. Fiat and A. Shamir, "Zero knowledge proofs of identity" *Journal of Cryptology*, vol.1,no.2, pp.77-94, 1988.
- [19] D. Denning and G. Sacco, "Timestamps in key distribution protocols," *Communications of the ACM*, vol.24, no.8, pp.533-536, 1981.
- [20] W. Diffie and M. Hellman, "New directions in cryptograpy" *IEEE Transactions on Information Theory*, vol.22, no.6, pp.644-654, 1976.
- [21] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signature and public key cryptosystems" *Communications of the ACM*, vol.21, no.2, pp.120-126, 1978.
- [22] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms" *Crypto '84*, LNCS196, pp.10-18, 1984.
- [23] C. Guillou and J. Quisquater, "A paradoxical identity-based signature scheme resulting from zero-knowledge" *Advances in Cryptology - CRYPTO'88: Proceedings*, LNCS 403, pp.216-231, 1988.
- [24] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures for delegating signing operation," *Proceedings of the 3rd ACM conference on Computer and communications security*, pp.48-57, 1996.
- [25] S. Kim, S. Park and D. Won, "Proxy signatures, revisited," *Proceedings of International Conference on Information and Communication Security*, LNCS1334, pp.223-232, 1997.
- [26] A. De Santis, Y. Desmedt, Y. Frankel and M. Yung, "Who to share a function securely," *Proceeding of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, pp.522-533, 1994.

- [27] V. Shoup, “Practical threshold signature,” *Proceedings of Advances in Cryptology-EuroCrypt’00*, LNCS1807, pp.207-220, 2000.
- [28] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik, “A practical and provably secure coalition-resistant group signature scheme,” *Proceedings of Advances in Cryptology-CRYPTO’00*, LNCS1880, pp.255-270, 2000.
- [29] J. Camenisch, “Efficient and generalized group signatures,” *Proceedings of Advances in Cryptology-EUROCRYPT’97*, LNCS233, pp.465-479, 1997.

