國立交通大學

資訊科學與工程研究所

碩 士 論 文

關 於 密 碼 系 統 抵 抗 側 錄 攻 擊 之 研 究

A study on password system with shoulder surfing resistance

研 究 生：李侑昇

指導教授：蔡文能　教授

中 華 民 國　九 十 九 年 六 月

關 於 密 碼 系 統 抵 抗 側 錄 攻 擊 之 研 究

A study on password system with shoulder surfing resistance

研 究 生：李侑昇　　Student：You-Shung Lee

指導教授：蔡文能　　Advisor：Wen-Nung Tsai

國 立 交 通 大 學

資 訊 科 學 與 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年六月

關於密碼系統抵抗側錄攻擊之研究

學生：李侑昇　　指導教授：蔡文能

國立交通大學資訊科學與工程研究所碩士班

摘要

在使用者認證的機制方面，傳統上是利用使用者自行設定的帳號與密碼來進行身分認證，用來保障使用者帳戶的安全。然而實際上帳戶盜用的情形並不少見，除了一些易於被猜到的密碼或人為疏失造成的密碼外流之外，側錄工具也是造成文字形式的密碼被他人知道的一大主因，尤其部分側錄工具的功能完整且難以被發覺，在側錄成功的情況下，就等同於登入過程被他人監視，有如直接觀察他人舉動的肩窺攻擊一般。

雖然新式的認證型態較不易受到側錄攻擊影響，但多半需要靠額外的硬體裝置才能進行認證，因此便有一些研究是探討如何只利用基本輸入裝置來達到相對的安全性。

本研究提出一套認證方法，也是在不依靠額外硬體的情況下，利用隨機產生的網格資料搭配使用者自定的處理規則，達到動態密碼的功能，可以抵抗螢幕與鍵盤側錄和肩窺攻擊。一般傳統文字密碼只要被側錄登入過程，側錄者就取得使用者的帳號與密碼。但是本論文所提出的認證方法，即使登入過程被側錄，側錄者仍無法從側錄資料分析出登入規則。

A study on password system with shoulder surfing resistance

Student：You-Shung Lee Advisor：Wen-Nung Tsai

Department of Computer Science and Information Engineering
National Chiao-Tung University

ABSTRACT

The most common way to protect the user accounts is to authenticate users through their textual account/password. However, account stealing is still a serious problem. Besides the use of weak/simple password or accidently letting out the password by themselves, data logging tools like keystroke logger (keylogger) are often used to steal account/password. This behavior is called “shoulder surfing” attack because that it is very similar to the case that someone watching you while you are typing your password.

Although there are new types of authentication method, which data logging has less effect on, but those methods usually need extra hardware during the login procedure. Some researchers had been trying to find better authenticating methods without extra hardware.

In this thesis, we proposed a method with shoulder surfing resistance to authenticate user without special hardware by using an on screen grid structure with user defined rules. Applying user-defined rules to random grid layout on the screen, a dynamic password is required during the login procedure. And thus, it is hard to analyze the logging data when the authenticating rules are unknown.

誌謝

目前的感想是，論文終於要完成了！沒想到在修改與格式部分也不輕鬆，真是不到最後不能鬆懈。對於從小對作文有恐懼的我來說，回想起這段寫論文的日子，從一開始的收集資料及研究，慢慢推進，但是到後來必須把腦內龐大思緒擠成論文的期間，還真一項大工程，所幸有指導教授—蔡文能老師的指引及監督，非常感謝老師在這段期間所提供的協助及建議，讓我了解自己沒注意到的細節和問題，以及實驗室同仁在我報告時的聆聽及意見，使我能一步步邁向論文的完成階段，也感謝父母在這段期間的幫忙，在進入最忙碌的階段時協助完成一些當時無暇兼顧的事情。

總之，本人目前為止內容最多的一篇文就在碩士期間完成了，最後謝謝這項議題的其他研究者，這些學者的研究讓我能夠看得更遠，期望不久後能出現無需用戶端額外硬體且能完美抵抗側錄攻擊的認證方式。

目錄

表目錄

圖目錄

一、序論

使用者認證機制通常是爲了確認使用者身分而存在，因爲有些資訊或是設備，是使用者本人專用而不希望讓他人存取，這時就會透過這項機制來保護個人資料的安全，除非通過認證機制確認使用者身分，不然無法存取這些私人資料。而在目前的使用者認證機制中，最普遍的便是檢查文字帳號及文字密碼其對應關係是否與認證資料庫中的相同，這就是傳統的文字型密碼驗證。

而傳統文字密碼出現長久以來，面臨到許多安全上的考驗，由於整個認證過程確認身分的重點就在那組密碼的內容，只要帳號密碼的內容外流，那麼就可以存取該帳號使用者所想保護的資料，因此一些相對應的竊取密碼手段也被一一發掘出來，如暴力破解、字典檔攻擊之類，尤其在電腦運算能力日益強大的今日，這類破解手段的花費時間也越來越少，加上使用者設定密碼的習慣不一，過於簡單的密碼也會大大增加被破解的機率，而使用較複雜的密碼會有不易記憶的問題，加上一些側錄密碼輸入的手段出現，即使再複雜的密碼都有可能被一五一十的記錄下來，因此此種傳統認證方式的安全性是值得注意的。

在發現傳統密碼上一些潛在問題後，便有其他人開始思考有無其它替代方案，於是便出現了一些新式的身分認證機制，例如手機鎖、生物特徵辨識、手持密碼產生器之類，這些新式認證方式，避免了一些傳統認證方式的安全問題，但這些方式都是認證過程中需要使用額外的硬體，且使用者需要額外負擔這些硬體的成本，不像一般認證只需要基本的鍵盤或滑鼠之類。

那麼在沒有額外硬體的輔助下，是否也能達到相對的安全登入方式？爲了實現這種想法，一些非文字密碼或是文字密碼的變化型方法被陸續提出，其中的創意和巧思各不相同，也都有不同的安全特性，例如非文字密碼中以圖形記憶爲基礎的圖形式密碼，或是結合圖形與文字聯想的動態密碼。但是在攻擊技術持續進步的情況下，某些進階的側錄技巧仍然可以取得上述認證方式在登入時使用者所輸入的密碼，而且對於具有與肩窺攻擊（Shoulder Surfing）對等效果的側錄工具來說，許多圖形化密碼都將功虧一簣，因此在目前新式側錄工具的影響下，不須額外硬體的安全登入方式是

非常有挑戰性的一項研究。

因此本文的目標便是著重在不依靠額外硬體的情況下，確保登入過程中密碼不被他人取得，例如側錄攻擊之類的手段，而在這邊有一項大前提要先說明，就是這些認證系統，通常都不考慮在註冊階段或修改密碼時的資料外流問題，因爲註冊階段就等於是在設定認證方與被認證方之間共享秘密的階段，這個秘密的設定方法及內容是整套認證機制如何辨認使用者身分的核心所在，因此註冊時使用者必定要自行確保周圍以及電腦內部是在夠安全的環境下，而存放這些重要共享秘密的認證方資料庫，我們也先以不考慮資料庫被內部人員或駭客竊取的情況下，進行研究與討論。

接下來的內容將從第二章開始，將介紹一些傳統文字密碼以及一些常見的密碼竊取手法，接著是一些新式密碼或非文字類的密碼，再來提到所謂肩窺攻擊以及它所產生的影響。在第三章我們將介紹一些能夠抵抗不同側錄手段的登入機制之相關研究。第四章則提出我們的方法，以網格架構爲基礎的自訂認證規則登入方式，第五章則是對這套方法的一些使用流程介紹以及相關安全性分析。最後以第六章的結論以及未來研究工作部分來做結尾。

二、背景知識

此章將介紹傳統密碼的內容以及其可能的安全問題，接下來提到常見但不易被使用者發覺的側錄攻擊，以及它的一些相關介紹，之後是一些可能不受傳統側錄攻擊影響的登入機制，包括需要和不需要額外硬體的方法，最後帶出肩窺問題和新式側錄攻擊存在以及他們的相關介紹。

2.1 傳統密碼與鍵盤側錄

我們的生活中常利用密碼來保護重要的資訊，一旦帳號密碼外流，可能造成這些重要資訊的損失，於是一般人都會盡可能的保護自己的帳密資訊，以免被其他人知道。在傳統文字形式密碼的情況下，系統往往也會鼓勵使用者使用複雜的密碼，例如英文數字加上大小寫組合，而在輸入密碼時，螢幕上會使用「*」來避免密碼被直接從螢幕看到。

2.1.1 密碼遭竊的可能情況

儘管使用者不會刻意去洩漏自己的密碼，但盜用帳戶的情況仍然在發生，尤其是近年熱門的線上遊戲環境更是層出不窮，從大部分被盜帳號的玩家的人為因素來看，可大略分為以下幾種情況：

- 密碼遭暴力攻擊破解

  暴力攻擊顧名思義就是不用特別的技術，直接進行對目標密碼的猜測，利用窮舉法或是包含常見密碼型態的字典檔，不斷嘗試不同密碼來登入目標帳號，或是以比對密碼 hash 值的方式，期望能猜到正確的密碼。要是使用者的密碼過於簡單，加上現今電腦運算速度不斷提升，很有可能在短時間內就被猜出使用者的密碼。

- 帳號密碼與他人共用

  個人的帳密保護著的是私人重要資訊，但是在某些情況下會需要與他人共用一組帳號密碼，這時除了自己以外的人知道這些資訊，就多了一份風險，你無法保證對方是否也能好好保護這帳密資訊不外流，有時也因利益問題而發生認識的人反而是盜取帳號的元凶。

- 使用與個人資料關聯性高的帳號密碼

  有些人習慣將帳號密碼設為自己的生日或是電話號碼之類，這樣是可以幫助對密碼的記憶，可是也有攻擊者利用這點，專門蒐集一些目標的個

人資料，再以這些資料的內容去猜測目標的可能密碼爲何，當攻擊者運氣好碰到這些用個資當密碼依據的帳號時，在非常短的時間就能試出對應密碼。

- 帳密沒有保管妥當（例如寫在紙上貼在電腦旁）

  上面提到的是密碼過於簡單的情況，於是也有人將密碼設定得很複雜來避免這種情形，例如長度很長或是文字組成沒有特別意義的密碼，但是在這種情況下，常發生因爲密碼複雜難記，乾脆寫下來記在某個地方之類，更甚者例如直接寫在便條紙上後就貼在螢幕旁，假如被有意人士看到，可能就會造成相關的損失。

- 遊戲帳號的買賣

  帳號交易往往伴隨者個人資料的流動，因爲帳號中往往存著某些個人相關資訊，例如身分證號碼，而這些資訊在帳號轉讓的當下可能還不會被立即修改，在原使用者沒注意到的情況下，甚至會發生密碼沒有先改掉的情況，此時這些資訊就很有可能隨著帳號的轉讓而外流。

- 被釣魚網站或 e-mail 所騙

  有時會出現冒牌網站，或是所謂的釣魚網站，或是利用 e-mail 來假裝是客戶通知信之類，來誘騙使用者進行平常在真正網站的帳密輸入行爲，或是回覆相關個人資料，這些方式都會盡可能把信息內容或假網站外觀做到與原本正牌非常相似的地步，要是使用者不注意看根本看不出來，有時就因爲這樣，等於親自的把帳號密碼或是其他個人資訊交給第三者。

- 帳密網路封包未加密且被攔截

  這個通常是在網路容易被監控的情況下，像是在公共區網內例如網路咖啡廳的環境較有可能，如果透過網路認證時，傳送密碼用的資訊未被加密而被攔截下來，也就是截取到未加密的網路封包，這些封包經過多筆資料的組成還原後，很可能就是當初從使用者電腦傳出的密碼資訊，不過現今的網路認證通常會注意到這點，因而採用加密過後的封包來傳送認證信息。

- 遊戲公司帳密資料庫內容外流

  這種情況的可能性較低，而且在現今企業重視資訊安全的情況下，由外部因駭客之類的攻擊行爲造成資料外流較難發生，通常是內部員工或離

職員工所爲，事後也通常會有相關新聞報導，表示這些事件並不會經常發生，不過通常文字密碼可以先經過加密處理後才存入資料庫，因此竊取到這些資訊的人也不一定能立刻將這些資料還原成原本的密碼。

最後還有一種猜測是，直接從本機電腦記憶體擷取你的帳號密碼，不過這似乎是一般人的誤解，而且目前也沒發現可以有效達到這目的的工具，雖然是有記憶體資料 dumping 的工具沒錯，但是在作業系統的保護下，你很難去知道你想要的資料被放在記憶體何處，而且一般帳密輸入時間短暫，資料比對又是在伺服器端那，也就是說你的帳密資訊不太需要保存在本機電腦，可以說是不會留在記憶體中的（可能只暫存在緩衝區幾秒鐘），但是對於帳密比對位於本地端的情況來說（例如登入本機作業系統），可能就有風險，但此時也能靠系統本身的加密機制來防範。

不過有人是以上狀況都沒發生，但帳密仍然外流，那就有可能是自己電腦的問題，很多人這時就會聯想到電腦中毒，但是惡意程式通常都是破壞或竊取電腦內資料，對於帳密沒保存在電腦裡的人來說，惡意程式是如何取得帳號密碼的？側錄攻擊便是其中一種方式。

2.1.2 側錄攻擊手法的介紹

所謂側錄，就是在一旁記錄，鍵盤側錄就是記錄下使用者敲打在鍵盤上的每個按鍵及順序，而且記錄程式本身都是設計成能隱藏在作業系統裡如同不存在一般，難以被使用者察覺，所謂的 key logger 軟體就是專門做這種工作的，尤其是對於單純由鍵盤輸入的文字型密碼，當你在登入時輸入的帳號密碼，會被一並照著順序記錄下來。

基本上 key logger 可分爲硬體型與軟體型，硬體型的通常比較單純，簡單來說就是將特殊的硬體裝置接在鍵盤與電腦主機之間，如下圖的裝置，以記錄鍵盤所送出的信號。

圖 1：硬體型側錄裝置

資料來源：維基百科

不過這不是目前鍵盤側錄的大宗，因爲這裝置必須人工裝上，而且裝置本身體積有一定大小，不難被使用者發覺，但是在使用者沒注意到的情況下，硬體型對於作業系統或防毒軟體來說是完全無法察覺的。

至於軟體型的變化就比較多了，實作方式也很多種，例如運用作業系統本身的 API、利用作業系統本身的運作機制(ex. windows 的 system hooks)、攔截 kernel 層的信息、device driver 攔截，或許還有新的方式在未來出現，不過他們所造成的結果都是相同的，就是紀錄下你鍵入的資料，再把記錄檔透過網路傳到他人手中，而整個過程在電腦能力強大今日，只會對系統效能產生些微影響，使用者恐怕難以察覺。

相較於硬體型，防毒軟體這時是可以派上用場，但它們也很難偵測到最新版本的側錄軟體，在防毒資料庫尚未更新前，你的帳密可能早就被傳出去了。不過軟體型側錄工具通常需要先透過安裝才能正常運作，但是看看病毒四竄的今日，就知道這不是很大的難題，尤其是線上遊戲玩家使用來路不明的外掛程式時，可能間接的將側錄程式給裝進了電腦。

由上面說明可以發現，傳統文字形密碼在側錄攻擊成功的情況下，是會將帳密資訊完全洩漏的，而在[1]中有提到一些小技巧來讓使用者避免這些攻擊，例如手動打亂密碼的輸入順序，或是使用作業系統中的剪貼方式和螢幕小鍵盤，但儘管如此，對側錄涉及範圍較大的工具可能就無效。在現今較完整的認證機制是可以避免這種情況，通常都是利用其他裝置來爲認證過程增添一道確認手續，以下將簡單介紹幾個例子：

- 手機鎖

圖 2：由手機螢幕顯示通行碼
資料來源：維基百科

跟傳統的認證程序差不多，只是多了一道要用使用者當初註冊所設定的手機撥通特定號碼，或是在使用較先進的手機時，輸入手機螢幕上出現的通行碼，才能在特定的時間，例如手機認證成功的 30 秒內，去做傳統的帳密登入程序，也就是即使知道他人帳密，也會因沒有對方的手機而無法登入。

- 晶片卡

圖 3：晶片卡及其讀卡機
資料來源：維基百科

將密碼資訊儲存在卡片中，並利用加密機制來保護其中的資訊，在登入時透過讀卡機將卡片中的密碼資訊讀出後進行認證。在沒有他人卡片的情況下，可以說是完全沒辦法登入他人的帳號。

- 生物特徵

圖 4：人類指紋
資料來源：維基百科

利用指紋或是聲紋這類個人特徵，來進行使用者身分的辨識，由於這些都是每個人獨有的一套特徵，要找到第二位有相同特徵的人之機率可以說是微乎其微，尤其假冒指紋或聲紋的技術不算普及，因此除了使用者本人外，其他人皆無法假冒身分來登入。

- 手持式動態密碼

圖 5：可產生動態密碼的手持裝置
資料來源：維基百科

這種認證機制它有個特點就是，使用的密碼是每次都會變化的，也就是說這次登入所用的密碼下次就無法使用了，因此簡稱 OTP（One Time Password）。它能讓使用者在登入時所輸入的密碼每次都不相同，而使用者輸入何種密碼的依據是靠一個小型手持裝置，它能顯示此次登入所需要的亂數密碼，也就是即使登入時的密碼被側錄下來，攻擊者也無法將這筆密碼用在下次的登入，必須要靠使用者自己的手持裝置才能得知下次登入需

要的密碼爲何。

以上這些方法相對於單純的文字帳號密碼來說，顯得格外安全，但都必須要借助額外的工具才能達成，不過因爲我們的目標是想探討是否能靠基本的鍵盤滑鼠即可，且希望能盡量降低使用者使用系統的成本，所以後續不再往此方向討論。既然純文字密碼有它的缺點，那麼是否有只需要基本硬體即可達成認證的非文字類密碼？接下來將介紹一些可能達成這目的非傳統密碼。

2.2 非傳統密碼

相較於傳統文字密碼，非傳統密碼不完全以文字形式的帳號密碼來作認證，此類密碼也可能由文字和其他多種能呈現在螢幕上的效果搭配進行認證，而其中一大宗便是圖形密碼。

圖形密碼簡單來說就是利用人類對於圖形的認知來代替或是輔助文字認知上的功能。起初這類型的密碼被提出不完全是因爲側錄方面的安全問題，而是對於人類直觀上來說，一般人不擅於記住一長串的密碼，尤其是當密碼本身毫無意義時，但是有意義的密碼所受字典攻擊成功的機率就會更大，而當去記憶一張圖片時，儘管圖形的資訊量可說比一串文字密碼來得要多，但人類就是較容易記下圖片的模樣[2]，而且相較於文字密碼能精確的用人類語言描述，圖片無法用人類語言精確的表達出，這也降低了意外洩漏密碼內容的可能。

既然圖形相較於文字有許多適合當作密碼的特性，就有人開始探是否有類似圖形形態或其他非傳統密碼來進行身分認證方式，以下將舉幾個這方面的例子，包含運用圖片、使用者產生的事件和記憶聯想。

2.2.1 以圖形爲基礎的密碼

在發現前述一些文字密碼的缺點後，便有人開始尋找有無其他適合代替文字作爲密碼的，由於文字是人類經由視覺來獲取資訊，因此後來便有人提出使用同樣能由視覺獲得資訊的圖片來代替文字，而且對於記憶文字來說，一般人對於記憶圖片是較直觀的，如同在[3]中所使用的圖片，即使顯示的是抽象形態，使用者大多還是能夠記下，因此利用記憶圖片出現順序來代替文字順序，就是圖形密碼的一種例子。

除了單純的記憶圖片順序外，也有其他例如記憶一張圖片中的特定位置、圖片的相關性之類，總之圖形密碼不僅僅只有代替文字順序的作用，由於圖片所包含的資訊遠比字元來的多，因此也可以產生別於圖片排序的其他密碼形式，在第三章會介紹一些這方面的實例。圖形密碼還有一個特性是相較於文字密碼，假如是一張構圖不是太單純的圖，是不容易藉由語言或是用筆寫下的，這也減少的密碼不經意告訴了別人或是寫在某處被看到的風險。

2.2.2 以事件爲基礎的密碼

傳統密碼在登入時是需要將註冊時所輸入的密碼，於認證時再輸入一次，那麼將這種概念擴大範圍就可以想成，在登入時去重複一遍設定密碼時所作的動作，那麼這個「動作」是否能直接用使用者產生的行爲來當作密碼？這就是事件導向型的密碼系統所要達到的功能。

這邊的事件可以泛指使用者在登入時對認證系統能做的一切動作，因此便有人將認證系統設計成使用者可以用類似玩遊戲的方式跟認證系統互動，可能是做出一個環境讓使用者用輸入裝置在裡面移動，或是將一些圖形置於畫面上讓使用者去跟它互動，而這一連串使用者產生的事件，就可用來當作密碼。

由於這些系統所創造的互動環境都會盡量讓使用者有多種不同事件可產生，也許會發生記憶密碼上的困難，不過由於是使用者在跟系統環境互動產生的事件，因此環境本身就有著類似密碼提示功能的存在，而使用者親自操作所產生的事件，在密碼回想上也有加強記憶的作用，總之，就是藉由特殊的設計，來讓使用者對系統所作的特定互動來當作登入的條件，也就是將傳統密碼的文字輸入行爲擴大成任何該系統所能識別的使用者動作，這在第三章同樣會有一個實例說明。

2.2.3 運用記憶及聯想產生的密碼

通常這類型的密碼在登入時並不會輸入跟設定密碼時一樣的資訊，因爲這類方法用的是將當初註冊時的設定，經過聯想或是記憶規則的方式，將這些設定轉成一組密碼，這邊的聯想可以是例如圖片與文字間的聯想，總之都會先透過另一層的轉換才會產生真正的密碼。

這種方式的好處就是在登入時透露的密碼資訊較不直接，因此因側錄

問題而密碼外流的機率較低，因爲登入用的是轉換過的密碼，未必能用在下次登入，所以攻擊者需要做的是把轉換後的密碼，還原成原始的狀態，而在這類登入方式中如果不知道使用者註冊時設定的細節的話，就無法輕易的把轉換後的密碼還原，這部分也會在第三章再仔細說明。

2.3 肩窺攻擊

Shoulder surfing 原本指的是站在別人背後，偷看螢幕或鍵盤上資訊的行爲，如果認證過程中，與密碼內容有高度相關的資訊，直接透過鍵盤滑鼠輸入，或是直接出現在螢幕上的話，透過肩窺就可以把這些資訊一一記錄下來，進而得知密碼的內容，尤其在上一節提到的一些特殊非傳統密碼，在這種情況下可能就無法發揮它原本想保護密碼避免被側錄分析的作用。

一般肩窺的情況是被他人偷看到自己的登入過程，例如在圖形密碼的情況下被從螢幕看到選用的圖片之類，那麼使用者在登入前如果注意一下周圍的環境是否就能避免這類情況發生？確實，一般人在進行登入程序時，只要注意一下周圍就可以知道有沒有人在偷看，但是有沒有可能即使沒人在身旁，你的登入過程還是被監視？從先前提到的鍵盤側錄來看是很有可能的，但這也只局限於鍵盤的部分而已，不過科技在進步，側錄方面的技術也在慢慢改進，從基本的鍵盤側錄，到後來有針對滑鼠輸入密碼方法的滑鼠側錄，甚至螢幕畫面的截圖或是錄影都在慢慢增加中，因爲在以往電腦運算能力不強時，在到達滑鼠或螢幕側錄這種側錄等級時，往往會拖慢使用者的系統效能，因此容易被使用者察覺問題，但是現在電腦運算能力已進步許多，造成這些進階側錄攻擊可以在不影響系統效能的情況下於系統中運作，假如防毒軟體沒有適時發現，那麼使用者在作登入時的帳密流出機會相當大，圖 6 便是這樣的一個例子。

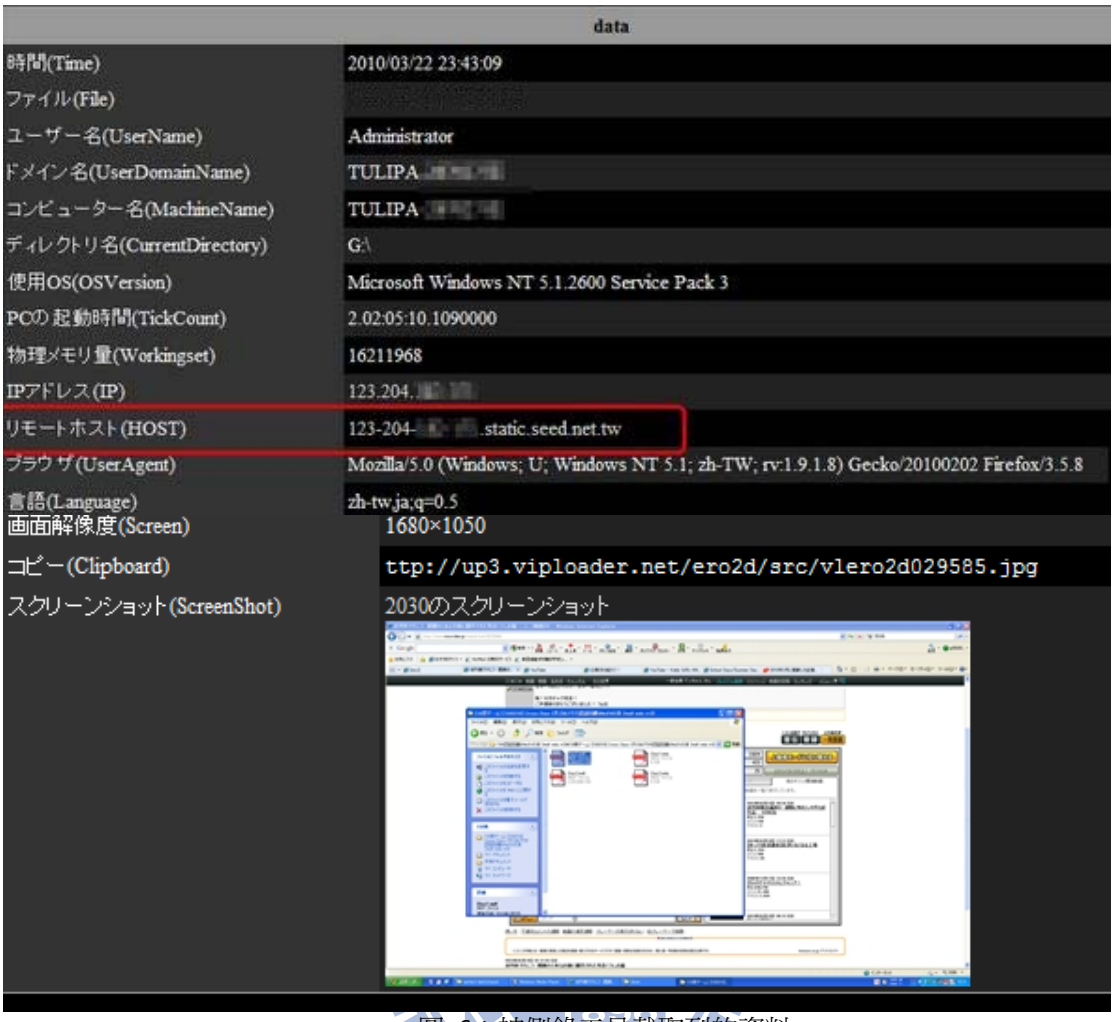

圖 6：被側錄工具截取到的資料

資料來源：http://pcuser.pixnet.net/blog/post/26531073

所以當這些側錄工具合併使用的後果，就跟有一位真人站在你背後偷看沒甚麼兩樣。

因此 shoulder surfing resistance password 的要求就是在只有傳統鍵盤滑鼠輸入工具的情況下，想出一種非傳統密碼輸入方式，讓側錄工具無法直接獲得該密碼。

要達到這種條件，由於假設的是側錄工具能完全錄下使用者傳達出的認證訊息，因此不能讓認證訊息清楚且直接的透露密碼資訊，而在密碼資訊不精確的情況下，認證過程就要經過不只一次成功才能順利確認該筆認證，因此登入所需時間通常會比傳統密碼的只需單次成功來得長。

而且通常爲了混淆側錄工具的視聽，有一種方法是會在一個畫面上出現上百個物件來引導使用者去注意螢幕上某個位置，這也會讓使用者爲了尋找物件而加長登入所需時間，以及物件過多時造成人眼搜尋上的不便。

因此雖然目的在於防止密碼資訊外流，但結果可能讓使用者不方便，是否能在維持安全性的情況下，縮短登入時間以及增加使用上的直覺及合理性，將是這類 shoulder surfing resistance password 的重要目標。

三、相關研究

本章將介紹一些非傳統密碼的相關研究，首先會介紹早期的圖形密碼和一些不同型態的非文字密碼，接著介紹的是一些考量到側錄和肩窺問題的密碼系統，最後是對於以上系統的一些測試以及便利性評估方面的相關研究。

3.1 不同樣貌的非文字密碼

首先列出一些較早期所提出的非文字密碼系統，其中大部分以圖行密碼爲主，希望能盡量發揮圖形相較於文字易於記憶以及無法被鍵盤側錄記下密碼的優點，不過因爲此時研究大多尚未考量到登入時被他人窺視的相關問題，所以對於新式側錄工具以及肩窺攻擊較無抵抗能力。

3.1.1 最早期的圖形化密碼

這是早期由 Greg Blonder（1996）[4]提出的一種方法，它的方式主要是使用系統內定的圖片，讓使用者對圖片中特定的幾個區域做點選的動作，而這些被使用者點選區域的位置和順序就被當作密碼來使用，當使用者在登入時，便依照之前的位置和順序點擊圖片的相關位置，如圖 7 所示，如果跟當初註冊時的設定相符，那麼就可以成功登入。

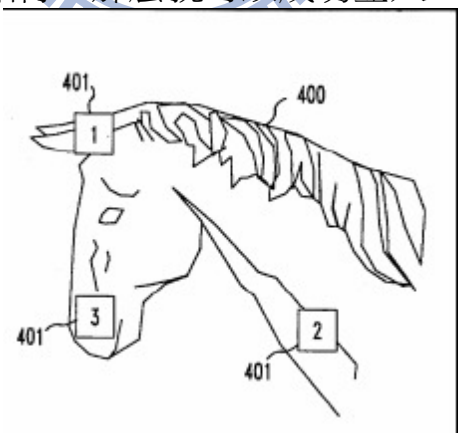

圖 7：最早期的圖形密碼

此方法雖然將密碼以別於文字的方式來表達，避免的鍵盤側錄造成的安全問題，不過由於提供給使用者所用的圖片和點選位置還無法自訂，因此在密碼設定上的自由度較不那麼高。而且由於點擊位置只有內定的幾處，使用側錄滑鼠座標的方式就可以簡單的分析出，使用者在登入時所點選的圖片中之位置，而這些位置在每次登入時都是固定不變的，所以攻擊者便

可以把側錄到的位置直接用在下一次的登入。

3.1.2 Draw-A-Secret (DAS)

這是由 Jermyn 等人（1999）[5]所提出的一套方法，A.F.Syukri 也提出過一套類似方式[6]，主要就是利用使用者所畫出來的東西來當作密碼。在進行畫出密碼的動作時，是在以一個 5X5 網格爲背景的畫面上進行，使用者所畫出的東西會以某種順序經過裡面的方格，而就由這個順序來表示密碼的組成，當使用者要登入時，就重複一次當初註冊時所畫出的東西，如圖 8 所示，只要繪出的軌跡誤差不要太大，理論上是會產生同樣的網格經過順序，假如順序相同，就可以成功登入。

雖然直接記憶網格經過順序對使用者不容易，但由於這是經由使用者畫出的結果所產生的順序，因此只要記下畫出的圖形的大略模樣即可，而且自己繪出的東西在記憶上也比較容易。由於網格的經過順序可由任意的 5x5 共二十五個格子來制定，經過他們的評估發現一個由長度十二格所產生的順序當作密碼，其理論上的密碼空間相當於長度爲八的傳統文字密碼。

不過實際上的密碼空間方面，他們發現到，由於使用者一般都是畫出一個連續線段來表示密碼，在這情況下，二十五個方格所會經過的順序可能就不是那麼的隨機，例如前進到某格，要畫出一個 90 度角的折線，這時便只有 25 個方格裡的 2 格可能會被用到（往左右或上下），要達到類似隨機的效果，以畫出多個點來表示可能會有較好的效果，不過相較於點，使用者還是比較傾向於使用較容易記憶的線段圖案來組成密碼圖形，因此實際上的密碼空間就會比理論上小得多。

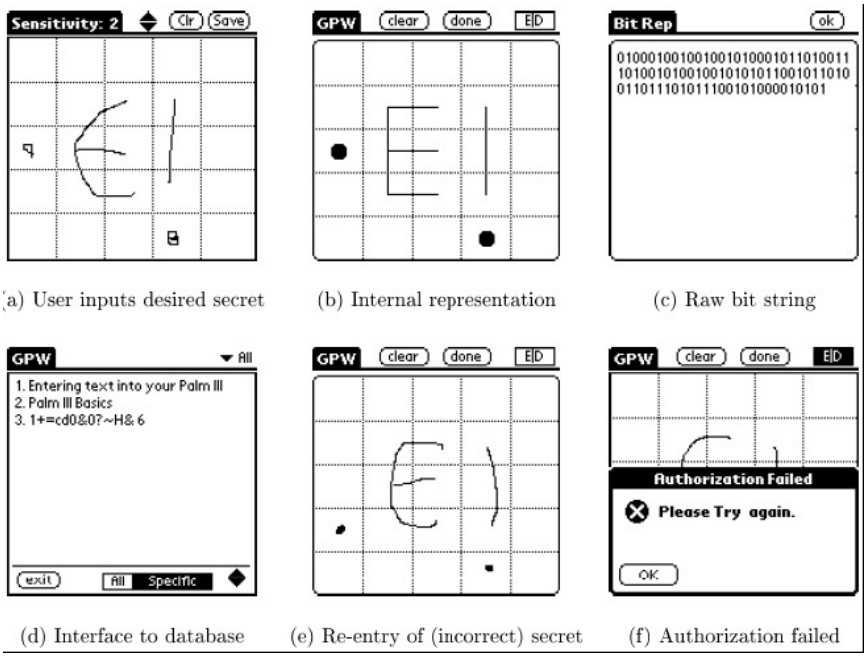

圖 8：以手動繪出物爲密碼

不過這套方法雖然也是以非文字來代替密碼，但是假如用來繪圖的裝置，例如滑鼠被側錄的話，所畫出的軌跡還是會被記錄下來，更甚者是用一張螢幕截圖就可能拍下密碼的全貌，而由於每次登入所要畫出的東西都是一樣的，因此攻擊者獲得的資料可以馬上用在下次的登入上。

3.1.3 PassFaces™

PassFaces™是由 Real User Corporation（2001）這家公司所提出的一套有被商業化的系統[7]，運用的是圖形密碼的方式，以記憶一連串圖形的順序來代替傳統文字密碼的文字順序，而他們認爲人對於人臉相較於其他普通圖片擁有更高的辨識度以及記憶力，因此他們的方法使用的是人臉的圖片，如圖 9 所示，這也是此系統的名稱由來。

圖 9：記憶人臉代替記憶文字密碼

使用者在設定密碼時是以挑選幾個人臉圖形來當作密碼，在登入時，會一次在畫面中出現九張人臉圖，使用者需要從裡面找出其中一個當初所選擇的人臉，接下來會換上另一張由九張人臉所組成的圖，使用者一樣要從裡面找出當初所選的圖，由於一個畫面只有九種可能選擇，爲了避免被他人意外猜對而登入，設定上是要重複五次這些步驟，並且每次都要選對才能完全成功登入。

在使用方面的研究上，他們確實發現相較於純文字密碼，這種記憶人臉的方式更容易被記憶，而且理論上的密碼空間可藉由人臉資料庫內圖片的數量上升而增加。不過此種方法，在攻擊者成功利用適當時機的螢幕截圖和滑鼠側錄的話，還是能藉由比照螢幕資料和滑鼠座標資料來找出使用者在登入時所點選的圖片，由於註冊時所選的圖片就是當初記憶的那幾張，因此在側錄多次登入後，就可以找出使用者當初全部的註冊臉形，這樣就等於密碼完全被攻擊者知道。

3.1.4 PassPoints

這是由 S.Wiedenbeck et al.（2003）[8]所提出的一套方法，跟本章開頭

所介紹的早期圖形密碼相似，也是以圖片中的點選位置及順序當作密碼，不同的地方是，PassPoints 所採用的密碼設定方式較爲自由，除了系統內定的圖片外，它還可以讓使用者使用自己準備的適當格式的圖片來代替，而且可以允許滑鼠點擊的位置並沒有設限，只要是在圖片的範圍內皆可，如圖 10 所示。由於可供點選的位置並非特定位置，因此圖中的每個像素都有可能是被點選的地方，但是人在使用滑鼠點擊時，要做到精確到像素大小的地步並不是很容易，因此這套方法並不要求在登入時一定要精確的點擊在當初選擇的位置上，而是可以容忍一定範圍內的誤差，例如使用者註冊時選擇的點位於圖片中的一個小物件上，那麼在登入時，他所點選的位置只要大致有落在這個小物件的範圍內，都算是正確的密碼輸入。由於選用的圖片可由使用者自行提供，因此便可挑選自己在設定密碼點擊區時較方便記憶的圖，例如把圖中一些對使用者有特殊意義的地方當作密碼點來使用，這樣在密碼記憶上會有更大的幫助。不過在後來使用方面的研究上發現，在密碼輸入的部分，此種方法所花的時間會比傳統用鍵盤鍵入文字密碼來的久一些，因爲在點擊密碼點時，使用者會因想要精確的去點選某個位置而放慢速度，或是在滑鼠操作技巧上不是很穩定之類。

圖 10：以點擊位置及順序作爲密碼

雖然比起這節一開始介紹的方法，這邊多了任意替換的圖片以及任意的點擊區，可是在螢幕截取和滑鼠側錄的雙重影響下，經過分析對照還是可以歸納出使用者在登入時所點擊的各個位置，而這些位置也不會因爲下次登入而改變，所以攻擊者還是可以用這些觀察到的點擊位置來登入原本

使用者的帳戶。

3.1.5 V-Go

V-Go 是由 Passlogix Inc.（2004）[9]提出的一套有經過商業化的系統，他所用的方式是以讓使用者對一個環境產生一些事件，而那些事件就被當作密碼來使用。他們所提供使用者操作的環境一個佈景主題，可能是廚房或臥房之類，這些佈景主題中散布著許多物件，如圖 11 所示，使用者可以跟這些物件作互動，像是打開某個箱子，把時鐘撥到三點，將衣服放進衣櫃之類。所以所謂的將以上這些使用者對佈景所產生的事件當作密碼，就是指在登入時，要將當初對這個佈景主題作過的一切動作，按照當初註冊時設定的順序，完整的重現一次，如果中途步驟都跟註冊時設定的相符，那麼就可以正確登入。由於這佈景中可以操作的地方不少，順序也都由使用者自行決定，因此密碼設定上的自由度不低，能產生的密碼組合也非常多樣，而且對於親自去對佈景裡的物件作出行為的使用者來說，在記憶上也是不容易忘記才對，因此對密碼的記憶性也不低。

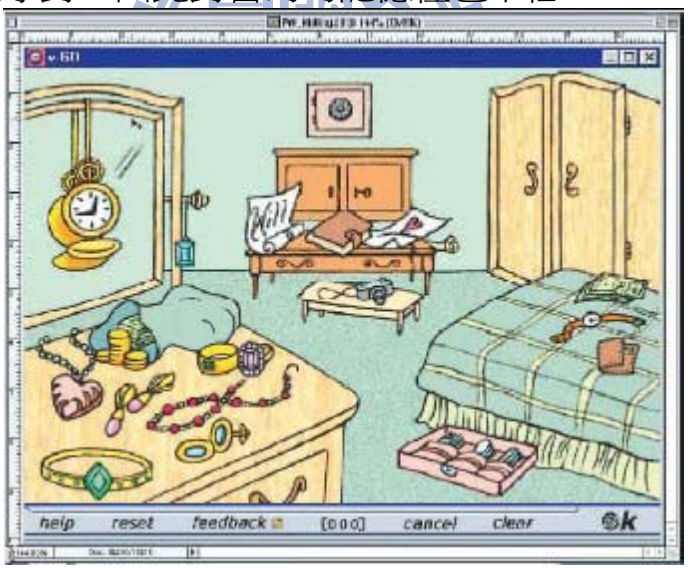

圖 11：以對虛擬環境的互動作為密碼

至於在有側錄攻擊的情況下，因為佈景主題以及物件各有不同操作方式的特性，鍵盤以及滑鼠側錄均無法明確了解使用者正在對哪種物件作出甚麼樣的動作，就算搭配上螢幕截圖，由於整個登入過程是由一連串的連續動作所組成，因此單張截圖也未必能了解整個登入情況是如何。但要是能對螢幕的連續影像進行側錄或錄影，或是在肩窺攻擊的情況下直接觀測

對方螢幕，那麼登入過程跟物件的互動就能被完整記錄下來，攻擊者只要在相同的佈景主題出現時，重複一次當初記錄下的動作，就有辦法登入原使用者的帳戶。

3.2 具備側錄抵抗及肩窺抵禦的登入系統

接下來將要介紹含有防止進階側錄攻擊概念的系統，在早期的非文字密碼及圖形密碼開始被實際應用後，由於登入過程所需呈現的圖案有可能被螢幕旁的人窺視，而圖形往往是登入這類非文字密碼的重要關鍵，因此所謂肩窺攻擊的防範便開始被注意，加上近代側錄技術已進步到不光是傳統的鍵盤側錄，而是連同滑鼠跟螢幕都能一起捕捉，即使是身後沒有其他人，還是會受到同等於肩窺攻擊的影響，因此以下密碼系統便是著重在抵抗這類攻擊。

3.2.1 以辨認 convex hull 及圖形交點的方法

以下將介紹三種由 Birget et al[10]所提出的方法，基本上都是用到一些交點與幾何上的性質，主要特點在於能藉由認證方和被認證方所共享的一項秘密，來讓認證方能經由被認證方提供的資訊，去判斷被認證方是否確實知道這項共有秘密，但是在認證過程中又不會直接透露關於這項秘密的信息。

首先第一種是利用多個點能圍成一個範圍的特性，認證方與被認證方之間共享的就是要用哪幾個點來做參考。這種方式就是在註冊時，使用者挑選一些系統內定的小圖案，在登入時，畫面上會顯示出隨機排列的許多小圖案，而在這其中會出現至少三個當初使用者選擇的圖案，由這三個以上的圖案可以形成一個範圍，使用者便利用滑鼠去點擊畫面，如果點擊的位置位於這個範圍內，就完成了這回合的認證動作，但此時還不一定是完成登入，因爲爲了避免被他人意外認證成功的問題，所以通常需要連續數回合的成功認證才能完成登入，範圍的示意圖如圖 12 所示。

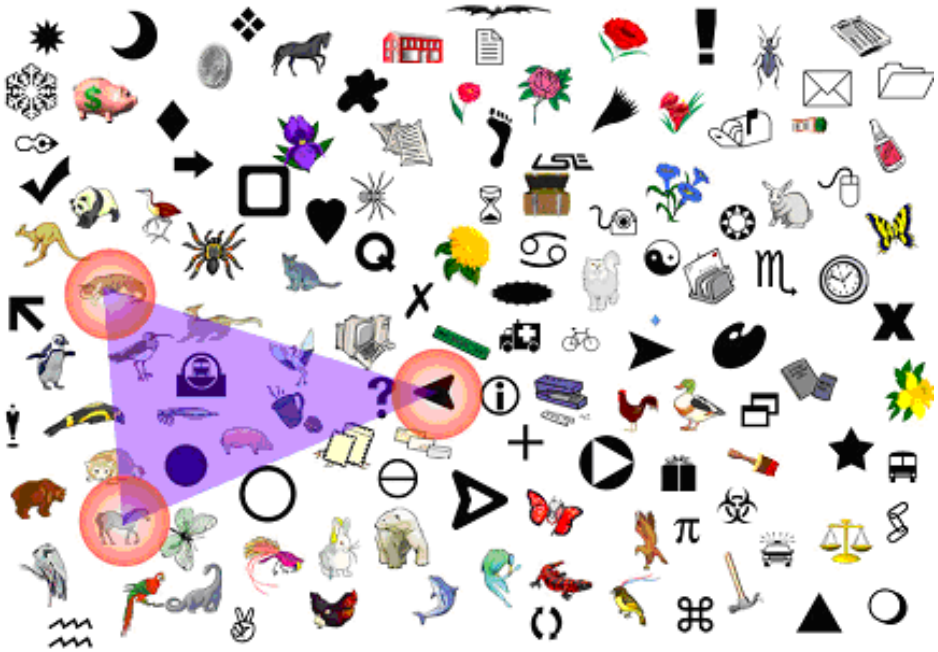

圖 12：以特定圖案圍成範圍爲通關區

第二種方法是利用一個特殊的結構，由外環和中間部分所組成，中間部分一樣散布著許多圖案，而其中有兩個是使用者當初所選擇的圖案，至於外環部分，則是會由數個圖案排成一圈，其中會有一個是當初註冊時選擇的圖案。要正確的通過一回認證的話，需要利用可轉動的外框，將上述提到的三個圖案連成一條線即可，如圖 13 的情形。

圖 13：以特定圖案連線位置形成通關位置

第三種方法一樣是去判斷圖案點所形成的幾合意義，在這邊則是利用

四個點，形成兩條線，然後使用者需要去點擊這兩條線所形成的交叉點才能完成一回的認證，如圖 14 的情況。

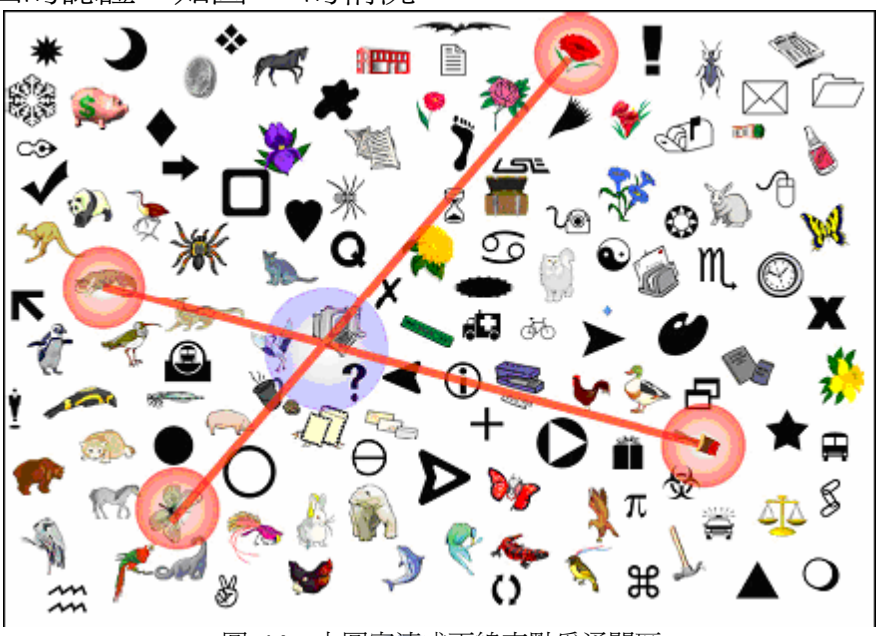

圖 14：由圖案連成兩線交點爲通關區

這幾種方法都能在不直接透露秘密資訊爲何的情況下，讓認證方了解被認證方是知道這些秘密資訊的，整個過程使用者也都沒有直接去點選關鍵圖案之類的動作，因此不論是鍵盤側錄、滑鼠側錄、或是螢幕側錄，得到的都不會是關於這項秘密的直接資料。

雖然側錄所得到的資料並不是直接性的資料，但是可能經由分析多筆側錄資料後，推測出可能的關鍵資料爲何，於是爲了增加被分析的困難度，畫面上出現的物件數量不能太少，這也是爲什麼這套方法會在螢幕上顯示數以百計的物件的原因，以期望能適當干擾側錄後的分析。不過這就造成了一個使用上的問題，由於物件數量龐大，又是隨機分佈的狀態，因此使用者在搜尋自己註冊時所選的圖案的困難度會上升，登入所花時間也會增加。

還有一些狀況會造成側錄後易於被分析的危險，就是當關鍵圖案都出現在邊緣或角落時，如果都是在同一邊的話，那麼關鍵圖案所圍成的範圍就會非常的小且非常靠近畫面邊緣，也就是當攻擊者側錄到的點擊點是位於邊緣的話，那麼就有很大的機率是關鍵圖案就在點擊點旁而已；至於圖案都出現在邊緣但不是同一邊的話，則圖案所形成的範圍就會非常大，這時不知道關鍵圖案位置的人，往畫面中間點選就有可能會通過這一回的認

證，有一項研究便是統計關鍵點所產生的範圍的各涵蓋區域的機率[11]，他們發現有很大的比例，圍成範圍會出現在畫面中央地帶，提高了他人意外成功認證的機率。

3.2.2 以關聯性聯想爲基礎的 Challenge Response 認証方式

接下來介紹的是 S.Man et al.（2003）[12]所提出的方式，裡面主要的機制在於讓使用者去記憶或聯想一個圖案與一個字串間的對應關係，使用者在註冊階段時換先從系統內定的圖案中挑選出幾個，並記下這些圖案由系統所指定對應的字串。當要登入時，畫面上會出現列出一系列圖案，爲了避免被側錄後分析，螢幕上的物件數量也是到近百個的程度，使用者此時便在畫面中尋找當初註冊所選擇的幾個圖案，並依照出現順序將該圖案對應的字串鍵入來當作這次的密碼。這樣的過程等於讓密碼能夠在每次登入都會產生不同變化，不過由於對應字串是由系統指定，對於使用者來說不一定好記憶，而且系統指定的字串可能有可以被預測的風險在。

因此後來 Hong et al.（2004）[13]改進了上述的方法，將字串指定交由使用者來決定，並利用一些圖案特性的說明或是相似圖案的變化來協助使用者設定較容易與圖案產生關聯的字串，如圖 15 所示

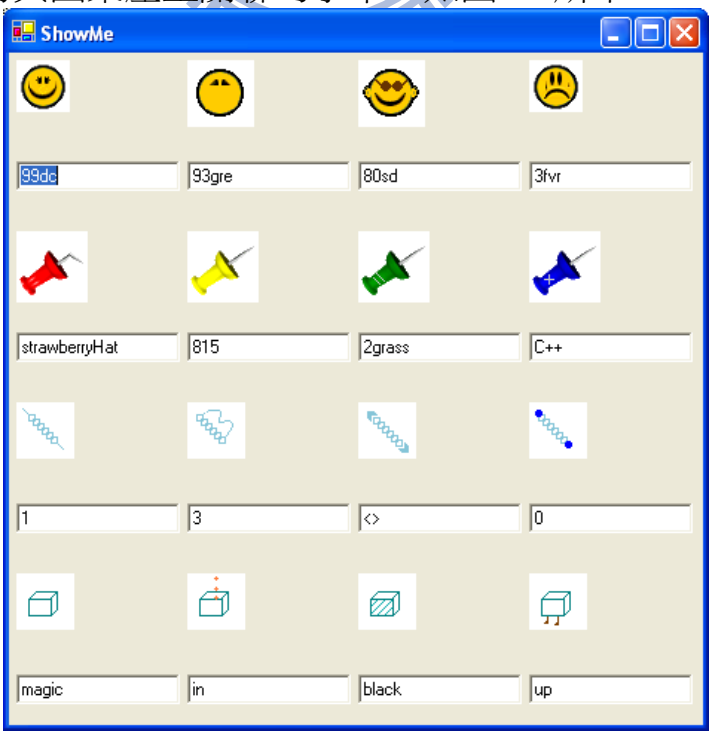

圖 15：圖與字串的對應設定

不過這樣的設定會拉長註冊所需要的時間，假如預設是選擇四種圖案以及四種變化圖案的話，那麼就有十六組字串需要進行對它的聯想以及設定。在登入方面，一樣是搜尋畫面中當初註冊圖案，並依照在畫面中的出現順序打出對應的字串來當這次的密碼，如圖 16 所示，如果跟註冊時的設定相符就可以成功登入。

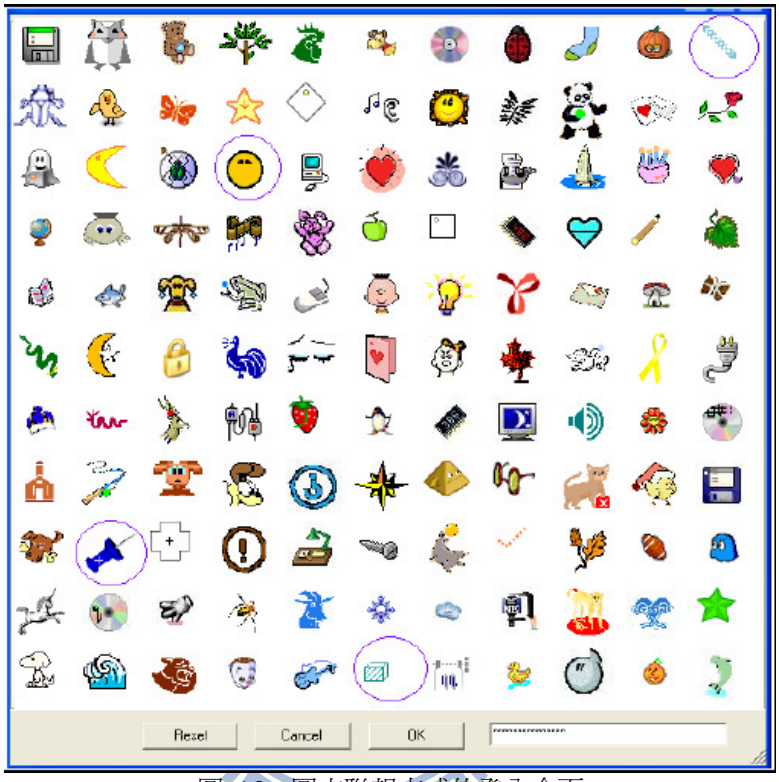

圖 16：圖文聯想方式的登入介面

雖然這樣的作法提高了設定註冊時的自由度以及被分析密碼的難度，但是這種字串聯想的設定方式對某些人來說可能仍然不易記憶，而利用圖片特性說明以及提示來協助字串聯想的方法，是否有幫助可能也因人而異，況且圖案對應的字串雖然可由使用者自行指定，但是指定完後就固定住不會變化，而在登入要輸入密碼時，這些資訊是由鍵盤直接輸入，也就是說，多次的鍵盤側錄結果可以歸納出使用者所設定的對應字串有哪機種，在知道了所有的對應字串後，可能就能配合螢幕側錄後的圖案觀察，搭配排列組合的方式去將使用者的圖案與字串間的對應關係找出來，進而破解使用者的密碼。

3.2.3 使用者可自行定義的通行機制

再來是由 Yung（2008）[14]所提出的一套方法，它的重點在於認證過

程的細節可由使用者來自訂，他提出了兩種作法，首先是類似密碼學中的替換（substitution），由於語言本身具有的特性，用來加密有語意且具有語言結構的資訊時有遭受分析歸納的可能，他提出了將之應用在登入時的密碼加密，一種利用密碼學中的替換實現肩窺抵禦的機制。

至於替換機制設定方式要從註冊階段說起，建立帳號與讓使用者設定密碼的介面如圖 17。首先使用者必須選擇構成驗證碼的座標點與字母替換的規則。Pass Position 欄位讓使用者去指定由畫面上矩陣中的哪幾個位置來產生驗證碼；Substitution 則是讓使用者去定義驗證碼裡出現的字母要如何替換成登入密碼，如圖 18 所示。

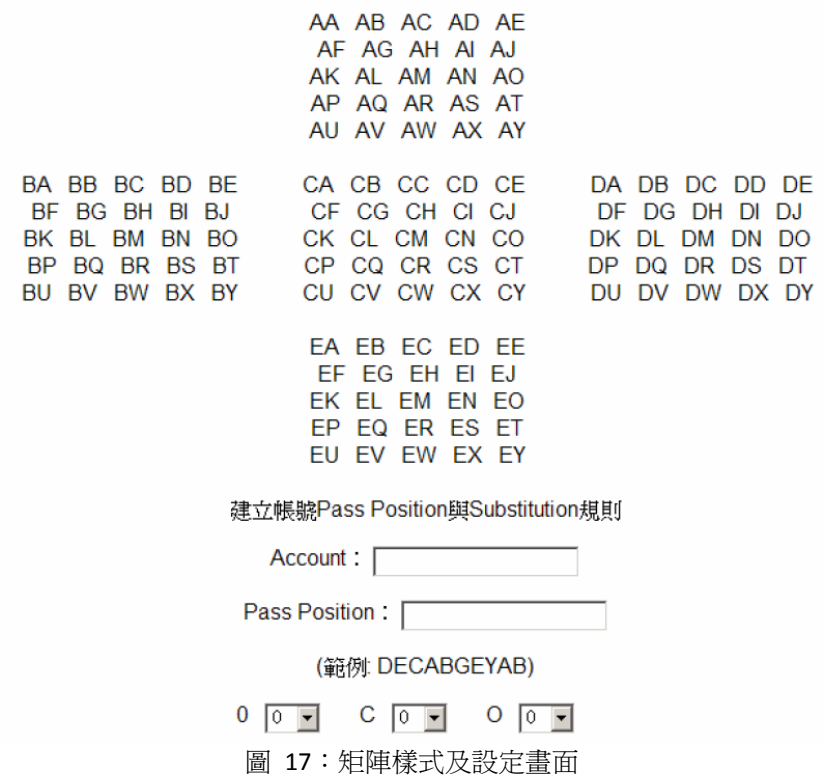

圖 17：矩陣樣式及設定畫面

使用登入系統時，先進行帳號的確認，輸入帳號後畫面上會出現五個 5 乘 5 的矩陣，矩陣內每個元素都是隨機的字元集合，包含數字 0-9 與 A-Z 共 36 個字元所構成的集合中的隨機結果。而在這五個矩陣中，使用者依之前在 Pass Position 的設定，查看自己所選定的矩陣中的位置上的字元爲何，將這些字元組成驗證碼，並套入當初自己設定的字母替換規則將驗證碼轉換完後，打入底下的 Password 欄位當作此次登入使用的密碼。

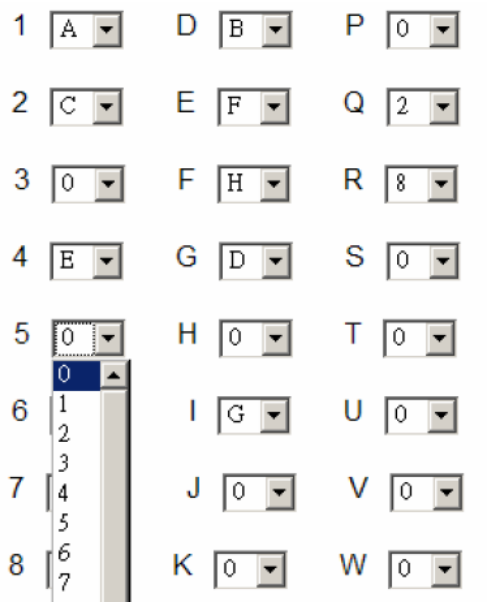

圖 18：字元替換設定介面

簡單來說，就是將隨機產生的矩陣內容所形成的驗證碼，透過自訂的替換規則產生真正密碼，雖然安全，不過替換的規則包含數字加英文字最高可達 36 條，一般人恐怕不易記憶，可能會將規則抄在小紙條上而增加規則曝光的風險。

此作者提出的第二種作法就是自訂式的肩窺抵禦登入方法：FMNS。FMNS 是 Five Matrices Numeric System 的縮寫，登入與認證介面與上述相似，不過將五個矩陣的個別元素改爲二位數的隨機數字，如圖 19。

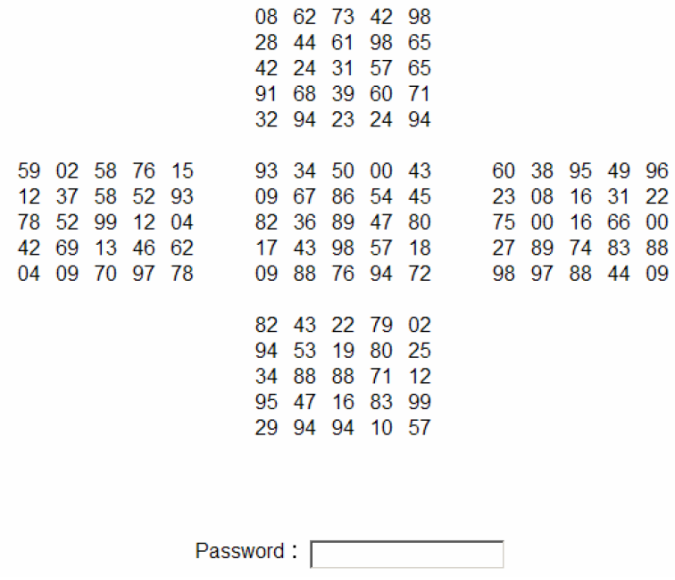

圖 19：數字矩陣內容

FMNS 發揮了使用者定義的這種精神，允許使用者去定義自己的密碼規則，爲了達到肩窺抵禦的效果，使用者必須利用隨機產生的五個矩陣作

爲規則套用時的運算資料來源。

建立帳號及定義規則的過程主要是使用類似簡化過的 C 語言的方式來寫出使用者的密碼規則，例如圖 20 中，第一行的 Pass 部分，代表使用者選定了隨機內容的矩陣中的 CA、CG、CM、CS、CY 等五個位置，由這些位置對應到的五個元素以自訂方式來構成通行碼，並在剩下的規則中對這串通行碼進行一些例如替換或是運算的動作，最後把經過規則運算處理的通行碼當作此次登入所用的密碼。另外也提供類似 C 語言中存取矩陣時 A[i] 的寫法，其 index 值限制在 1~125，代表的意義是對應到文字座標表示的 AA、AB、…、EX、EY 這 125 個，隨機矩陣中該位置的字元。

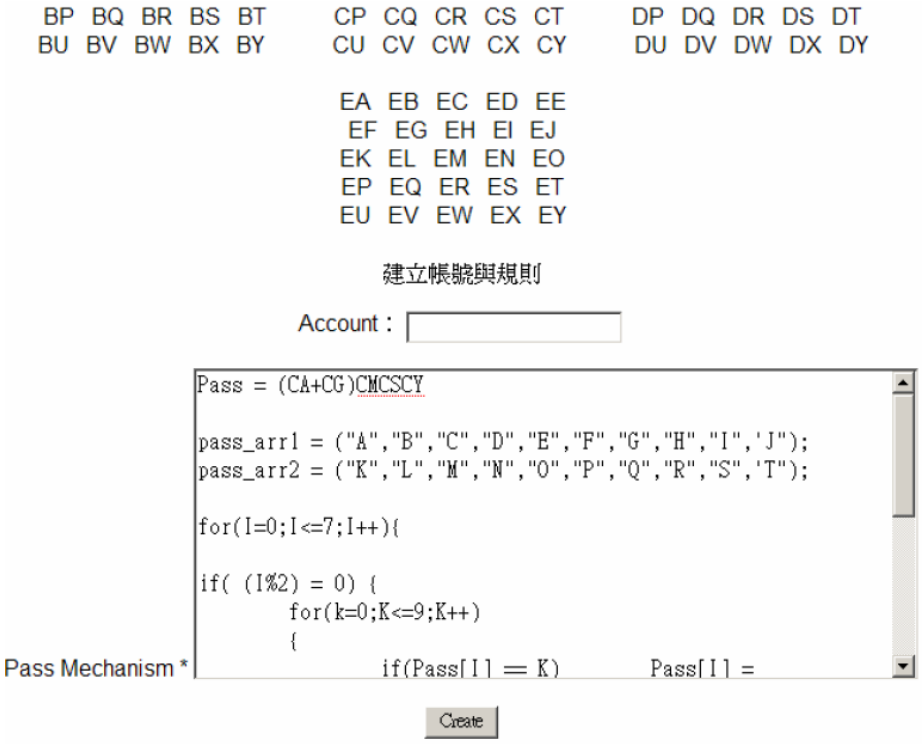

圖 20：密碼規則編輯畫面

此方法同樣是藉由將驗證碼經過規則轉換成實際密碼的方式，不過作者也有提到這套方法雖然比單純的文字替換法更安全，但可能僅適合有資訊背景的人使用，對於一般沒有接觸過程式設計的人來說，手動設計一套以程式語言爲基礎的規則，或是記憶這些簡化 C 語言產生的規則，可能稍嫌困難。

3.3 對於系統便利及可用性上的相關研究

有些認證系統在研發時最主要的考量是安全性上的問題，但有時夠安

全的系統在其他方面或許會存在問題，例如對使用者的便利與否，或是人類對於圖片選擇上的習慣性造成的安全問題，因此接下來將介紹一些對這類因素進行討論的相關研究。

3.3.1 分析 PassFaces™中使用者的選擇偏好

對於 PassFaces™系統，有人對它作了一些關於使用者行爲的統計[15]，發現使用者在選擇人臉圖作爲密碼時，有一些可預測的習慣性，例如習慣選擇膚色與自身人種相同的，或是在男性方面，偏向選擇女模特兒類的臉部圖形，他們統計出來的結果如圖 21。

Pop.	Female Model	Male Model	Typical Female	Typical Male
Female	40.0%	20.0%	28.8%	11.3%
Male	63.2%	10.0%	12.7%	14.0%

Pop.	Asian	Black	White
Asian Female	52.1%	16.7%	31.3%
Asian Male	34.4%	21.9%	43.8%
Black Male	8.3%	91.7%	0.0%
White Female	18.8%	31.3%	50.0%
White Male	17.6%	20.4%	62.0%

圖 21：各種使用者對臉部選取的偏好

由上面可以發現，男性在選擇做爲密碼的臉部圖片時，的確有超過一半的受測者傾向挑選女模的照片，這將影響設定密碼時的隨機性，而變得可預估，而對於人種方面所做的測試也呈現了明顯的結果，大部分受測者都傾向選擇跟自身人種膚色有關的圖片，而受測者中黑人女性方面更是高達九成都有這種傾向，這也顯示了這套認證方式最好要考量到使用者的性別與人種，來提供適合的人臉圖片集合，而不是只要隨機從資料庫挑出即可，否則使用者可能只會去選擇特定的圖片來當作密碼使用。

最後他們得到的結論是，使用者在使用這套認證方法時，所設定的密碼跟使用者的人種或是性別有一些明顯可預測的關連性，因此它的密碼空間可能不如預期來的高，這也是一個非傳統密碼較傳統密碼不容易分析實際密碼空間的例子。

3.3.2 歸納 PassPoints 中使用者對於認證點的選擇

後來也有人對 PassPoints 作類似的研究[16]，看看有是否能從一張圖片來歸納出使用者最可能點選的地方，進而計算某張圖所代表的密碼空間。他們首先推測，一般人會比較去注意圖片中的物件，例如一朵雲，而較不會去注意它的背景，例如天空，再來它們也推測一般人對圖片中顏色對比度較高的部分會較關注，於是他們就用以上兩種特性，去模擬使用者會選擇的圖片中之認證點，再與實際的結果作比較。最後他們選用了兩種比較具有代表性的結果，如下圖所示：

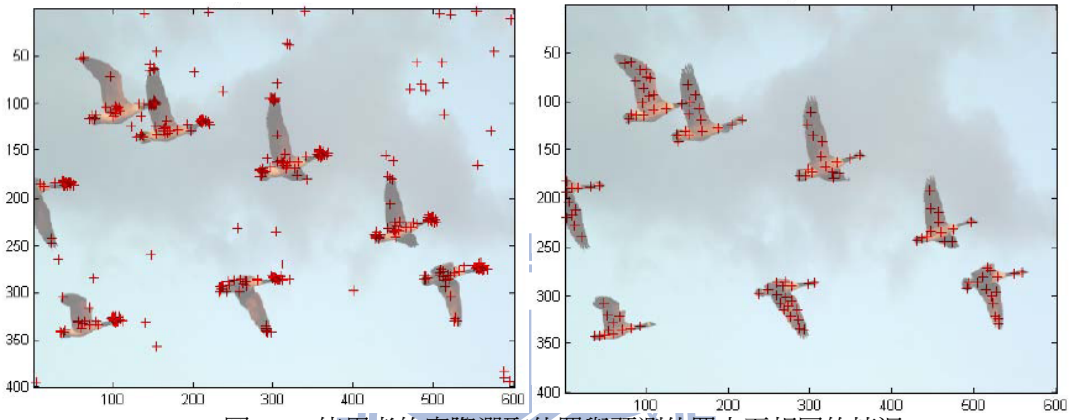

圖 22：使用者的實際選取位置與預測位置大至相同的情況

例如在圖 22 的情況，他們的預測跟使用者實際的選取情況非常吻合，這是由於這張圖片的物件明顯（鳥類），而且與背景天空的對比度高。因此這張圖片的密碼空間就不如理論上來的大，在作這類型的認證機制時就該避免使用這種圖片。至於較不會影響理論密碼空間的圖就如圖 23 所示，此圖是一張群眾的圖，由於裡面的前後景與物件較爲混雜，預測出來的點選點和實際的點選點就不會非常吻合，像這類的圖用於認證程序，密碼空間就會比前一張來得大，因此較適合選爲系統使用。這又是一個非傳統密碼需要仔細的分析後才能判斷實際密碼空間的例子。

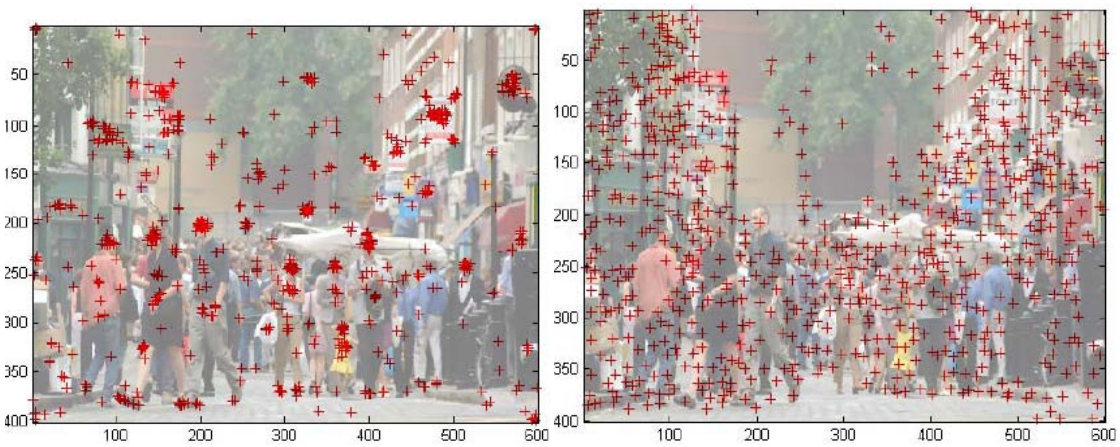

圖 23：使用者的實際選取位置與預測位置出現差異

3.3.3 使用者登入所需時間的評測

由於在 Convex hull 方法中，相較於傳統文字密碼的精確輸入，認證時輸入的資訊是以不精確的方式表達，在這種情況下要登入通常就必須經過連續數回合成功的認證，才能完整確定使用者的身分，而爲了使螢幕側錄後可能產生的分析行爲變得更加困難，畫面上會有許多用來掩護關鍵物件的其它圖形，也就是說使用者需要經過多回在上百個物件裡作搜尋的動作，才能完成登入程序，這似乎會大大影響使用者登入所需要的時間，於是他們便測試了使用者成功登入所花的時間[17]，結果如圖 24 所示：

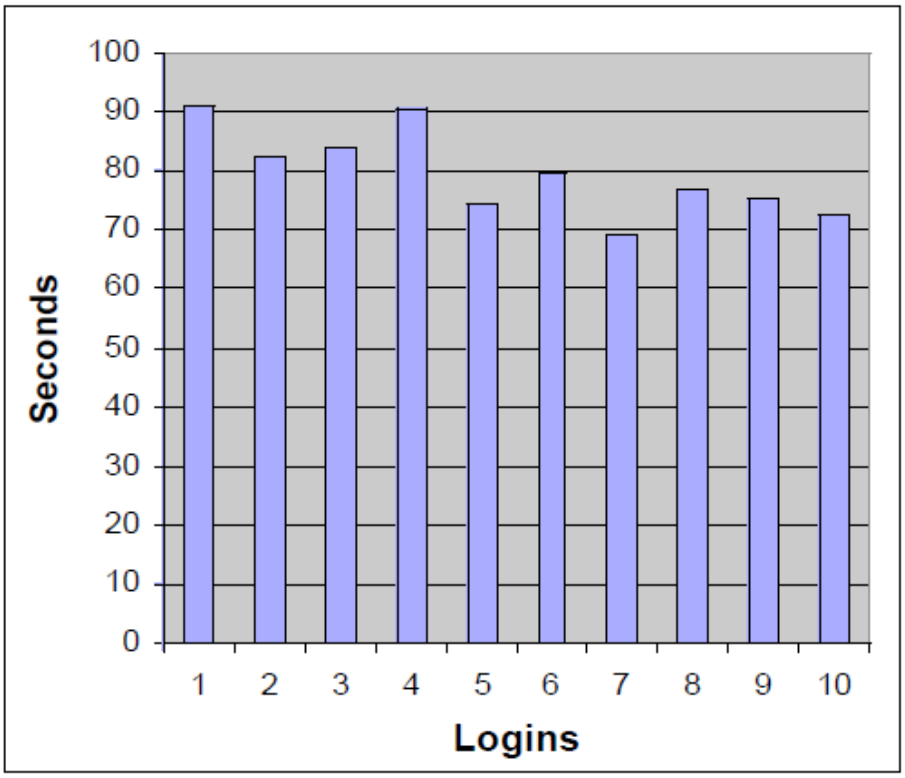

圖 24：使用者各次登入成功所花平均時間

最終的結果顯示，在登入上平均要花上七十四秒左右的時間，在物件不易搜尋的情況下，更有受測者出現長達一百秒以上的情況，可以說是爲了增強安全性而犧牲了登入過程的便利性，對於使用者來說，不需要如此高安全性的人可能就無法接受這種登入所費時間。

四、防肩窺密碼的分析及設計

此章主要會介紹我們所提出的抵抗肩窺的密碼，首先會回顧一下第三章的一些密碼系統，藉此來分析有利或不利於肩窺的相關系統要素，接下來便會介紹針對以上要素分析後，發展出來的一套方法，也就是我們所提出的以網格爲基礎的自訂規則認證方法，以及這個方法的詳細內容。

4.1 肩窺問題的剖析

在第三章我們介紹了一系列具有抵抗不同程度側錄攻擊的密碼系統，有些能使特定側錄方式無效，有些則是會被特定側錄方式攻擊，而我們可以看到儘管許多系統對於滑鼠及鍵盤側錄都能避免密碼資訊直接洩漏，但是對於肩窺攻擊都無法抵擋，因爲肩窺是全方位的觀測行爲，也是最棘手的一種。但是第三章後半還是有出現幾個可以抵抗肩窺攻擊的例子，究竟哪些因素會影響肩窺攻擊的成功與否，接下來便要探討這些特性。

4.1.1 利於肩窺的要素

由第三章的例子我們可以歸納出以下幾點容易讓肩窺或是側錄攻擊成功的要素：

- 密碼輸入沒有變化性

也就是使用者每次登入所需要輸入或要做的動作是固定的，這樣的話只要知道使用者某次登入所做的動作，別人也可以照著這些動作來登入原使用者的帳戶。而在現今資料側錄功能完備的情況下，使用者登入所做的動作有可能被非常精確的側錄下來，造成密碼外流的情況，這可以說是對肩窺攻擊最有利的一種特徵。

- 使用者的輸入資訊明確

這是代表使用者在登入時所回應的訊息，例如密碼方面的資訊，過於明確，以至於即使密碼可以隨著每次登入而有不同變化，還是有可能經由多次的側錄結果來分析密碼的變化趨勢。由於在分析時需要精確的資料來做推算，假如使用者輸出的信息越明確，那麼由側錄收集到的資料用於密碼分析就越有可能成功。

- 認證過程或模式變化不大

在使用者的輸入可能會被側錄的情況下，我們也得盡量讓認證的過程能有

變化，因爲太固定的認證過程也許會產生一些利於攻擊者去歸納的情況，例如第三章提到的幾何式認證所發生的點擊區過於集中於中央的問題，這時即使使用者輸出的資料並不明確，還是可能因固定認證模式所產生的一些特性，造成使用者密碼易於被分析的情況。

4.1.2 不利於肩窺的要素

爲了要避免肩窺攻擊的影響，一套認證方法應該要避免出現上節所提到的一些特性，因此要讓肩窺攻擊難以成功的話，以下幾點也是需要具備的：

- 密碼需要有變化

在使用者的輸入有可能被側錄的情況下，密碼每次登入時的動態變化是必需的，而且它的變化機制要不容易被推敲，但是又要能讓使用者知道密碼會如何變化。

- 使用者的回應含有假的資料或不精確的資料

在密碼會變動的情況下，攻擊者還是可能藉由側錄到的資料來分析密碼，並預測密碼的下一次變化，如果收集到的資料並不全是能用來做正確分析的，像是其中有幾筆是亂數資料之類，那麼就可能誤導整個分析過程，增加密碼被破解的困難度，例如使用者跟認證端達成某些協議後，允許在密碼中的特定位置加入使用者任意亂填的資料之類。不過由於使用者回傳的資訊不明確，往往需要經過多回認證才能確保對方真的知道通行密碼。

- 認證方式不固定

一般系統的認證程序皆是每位使用者都使用相同的機制去登入，唯有密碼的部分有差異，假如連整個認證程序的都能做到因使用者不同而異的話，那就更降低了被破解的危險性，因爲此時每位使用者的認證程序可能都不太相同，對某個使用者的認證機制的破解不代表也能破解其他人的。通常能做到這樣的系統都需要用到類似使用者自訂認證方法的機制。

- 較大的可視範圍

我們在先前提到了螢幕側錄的存在，即使某些系統利用螢幕上出現的物件來搭配進行認證，藉此降低鍵盤與滑鼠側錄的危害，但螢幕側錄在這種情況下卻是一大問題，例如以圖片記憶代替文字記憶密碼的系統。因此有些系統便會採用在螢幕一次放上許多物件來混淆側錄的方式，雖然螢幕側錄

仍可以完整的抓取整個畫面裡的物件，不過由於數量龐大，加上系統設計成使用者做出的選擇不會明確表達出來的話，會大幅增加分析密碼的時間，因爲攻擊者必須去從許多物件中猜測哪幾個物件是使用者真正有用到的。

4.2 系統概念

在看了幾項與肩窺抵禦有關的要素後，接下來就是要介紹一套能達到以上幾點條件的做法，基本上是屬於使用者自訂認證過程的一種方法，利用使用者自訂的認證規則，搭配上螢幕上出現的內容，去產生出一組數字，由這組數字來當作類似動態密碼讓使用者輸入，以下將開始介紹這種方法的實際內容。

4.2.1 以網格爲基礎之認證畫面

首先從螢幕上會出現的畫面說起，這種認證方式是屬於需要搭配螢幕上出現的物件來進行認證的，基本上是一個正方型網格的架構，由 36 個基本網格組成，具有以下幾種特性，範圍由小到大列出：

- 6x6 大小的基本網格，內含 0~9~A~Z 共三十六個字元隨機分佈在其中，如圖 25。

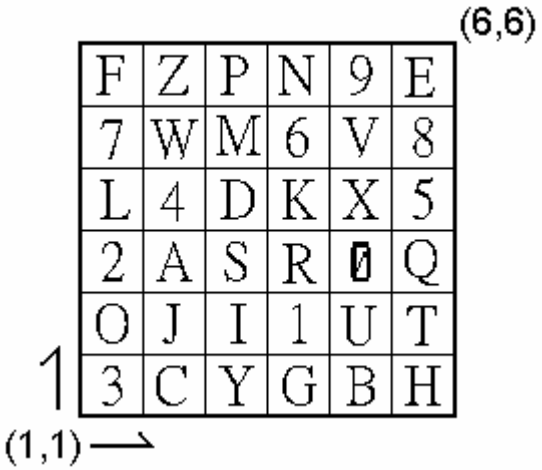

圖 25：基本網格

- 再由九個依上面規範隨機產生的基本網格形成一個 18x18 的區域，九個基本網格分別編號爲網格 1~9，如圖 26。

Grid 7	Grid 8	Grid 9
Grid 4	Grid 5	Grid 6
Grid 1	Grid 2	Grid 3

圖 26：由基本網格形成一個區域

- 最後由四個上面提到的區域形成最終的 36x36 的整體網格，4 個區域分別編號爲區域 1~4，如圖 27。

Section 3	Section 4
Section 1	Section 2

圖 27：由四個區域組成整體畫面

所以使用者最終在螢幕上看到的是由 36 個 6x6 的基本網格所組成的 36x36 整體網格，而這 36 個基本網格內如上述各含有隨機產生且分布方式不同的 0-Z 字元。之所以將網格以此種架構排列，不單只是爲了形成密切的結構，可以看到這種架構與某個數字有很大關連，也就是 36，例如出現的字元有三十六種、一個基本網格內有三十六小格、一共有三十六個基本網格以及四個區域組合成的是 36x36 的範圍，因爲這些性質的數量相同，有助於在規則設計時的統一性與範圍命名對應到字元上的方便。

我們提出的方法便是讓使用者去對這些網格裡的資料，做一連串的處理，因此才會選用方便與座標系統結合的網格架構，以及在網格內加入能用文字對應的資料，至於要如何處理這些資料，則是藉由使用者所自訂的規則來決定。

4.2.2 可自訂的認證規則

我們的認證方式要是讓使用者對網格內的資料做一些處理或運算，例如把某格內的字元和另一格的字元相加，雖然對使用者來說 A-Z 的部分並不是數字型態，不過可以把它當成 11-36 這樣的數值來看待，或是說去看某字元爲在網格內的座標爲何，並把這些座標數據做簡單運算處理。換句話說，就是讓使用者從網格內抓取他想要的數據，並將它做一些運算處理後當作密碼來輸出。

至於使用者要怎麼指定如何抓取或處理網格內的資訊，就是透過認證規則的設定，所以我們會訂出一系列模擬使用者對網格做操縱的一些基本指令，使用者再透過這幾個基本指令去完成他想要達成的動作，所以這些基本指令需要含蓋大部分對網格會有的幾種基本操縱行爲，並且要設計成能容易在網格架構上去代入並使用。

4.2.3 其他可增加認證過程變化的要素

在我們設計的基本指令中，除了一些較直覺的抓取資料或運算資料的指令外，還有一些爲了增加規則變化性才加入的指令，但基本上這些指令還是以能與網格架構配合爲重點。

舉例來說，去判斷某兩個物件在網格中的相對位置，可能不如上節所提到的抓取和運算那麼直覺，但它可藉由相對位置的不同回傳不同的資料供運算，來達到增加規則變化度的功能。或是利用網格內 Section 的特性來達到一個所謂間接選取的技巧，所謂的間接選取在這邊是指利用其他區域的物件來選擇目前區域實際要選擇的物件，在下一節的說明中會有更詳細的介紹。

4.3 認證規則的內容

由於我們的方法是讓使用者設計一套規則，並利用這套規則去對網格內容做處理，並把產生的結果當作密碼。而在上一節有提到，這個規則是由系統內定的基本指令所一條條組成，因此這些基本指令都必須跟網格操作有一定的關連，接下來便會介紹這些基本指令的內容以及用途。

4.3.1 與不利於肩窺要素的對應

在說明基本指令的內容之前，我們要先從它的由來講起。由於我們的方法是希望能達到肩窺抵禦的效果，所以設計出的基本指令除了要能適合套用在網格上的動作，例如尋找網格中的字或注視某座標外外，還必須要能達成先前提到的一些不利於肩窺的要素，除了藉由設計認證規則的內容來達到認證過程的變化外，像是代表使用者輸入的基本指令，就會在裡面包含可以設定輸入格式參數，以達到可以輸出不精確信息或是假資料的目的，還有爲了對應較大的可視範圍，會有一套各種大小範圍的命名及指定方式，並且在指定範圍後，盡量讓剩餘的規則套用動作維持在 6x6 的基本網格範圍上，以兼顧大可視範圍和減少認證時使用者需要同時注意的物件量。

4.3.2 認證規則的組成及運行方式

認證規則在執行時是依序由裡面第一個基本指令執行到最後一個基本指令，中間除非系統本身預設的行爲，否則不會出現某個指令造成跳耀或是迴圈的行爲，必定是循序一條一條去執行。

那麼在何種情況下系統會產生類似迴圈的行爲？這邊就要提到一般完整認證規則會包含四大部分：

- User Data

  這個部分嚴格來說並不算規則，但是在套用規則時會用到裡面的一些設定，例如規則欲套用的回合數，也就是完成一次登入所需要的認證回數，因爲只套用一次所產生的結果使用者可能感到不夠安全，那麼它就可以增加套用回數以增長最後輸出的密碼長度。使用者如果有自訂一些規則會用到的常數，也是會放在這部分。

- Starting Grid

  由於整個畫面上的基本 Grid 多達三十六個，而一般在套用認證規則時爲了避免使用者需要同時在多個距離不一的 Grid 裡作規則套用的思考，因此都是盡量讓規則套用在目前或是鄰近的 Grid 上，而 Starting Grid 這部分的規則就是讓使用者設定如何去選擇從哪個 Gird 開始套用認證規則。

- Rule

這個部分便是實際要套用在 Grid 上做處理和運算的規則，算是認證規則的主體所在，一般會由最多指令所構成的部分。

- Next Grid

  當利用目前或鄰近的 Grid 內容套用完一次 Rule 的部分後，依照使用者設定的回合數，可能還得進行數回的 Rule 套用以產生更長的密碼輸入，Next Grid 的部份即是設定如何挑選下一回合的 Grid 來給 Rule 部分進行下一回合的認證。

由上面可以知道，所謂規則套用時的迴圈行爲就是發生在 Next Grid 執行完畢時，也就是說整個規則的套用順序，在使用者有設定一次登入需要多回合認證時，是以 Starting Grid -> Rule -> Next Grid -> Rule -> Next Grid ->Rule…這種順序來執行的，直到完成使用者指定的回合數才停止，這也就是整個認證規則在套用時唯一會有順序變化的部分。說明完認證規則的架構，接下來就將介紹基本指令的內容。

4.3.3 基本指令的特性及格式

認證規則是由一連串的基本指令所構成的，這些基本指令就代表了使用者能對網格所進行的操作或運算動作，包括 GET、CALCULATE、DIRECTION、INDIRECT、OUTPUT，爲了避免使用者對於指令種類過多而感到不方便，因此基本指令就只有這五個，而一些有些微差異但是目的差不多的行爲，就合併到相關基本指令中的參數部分去設定。

規則的運行方面，在套用認證規則時，就是把規則內一連串的基本指令依序拿來執行，而每一條基本指令執行完時，它都會產生（回傳）一組運算結果，可以讓後面的其他基本指令把它拿來再做運算，例如假設規則（Rule）中的第三條是利用 GET 抓取 grid 中的一筆資料，那麼規則中剩餘的指令就可以設定成要把第三條所產生結果拿來當作參數使用，也就是說，規則中的每一條指令都可以當作一個暫存器來使用，裡面存放的便是該指令運行後所產生的結果。

爲了讓取得每行的資料更方便，我們將回傳資料的結果統一化，也就是設計成基本指令所回傳的結果皆有三筆資料，設成三筆的原因是正好可以代表 Grid 中某格的內容（文字）、它的 X 座標、它的 Y 座標。而由於回

傳的資料有三筆，因此只接使用某指令回傳的資料時，預設是去用第一筆資料，如果想要用第二或第三筆資料的話，就需要用到 GET 指令的某參數來指定是要拿 X 座標或 Y 座標部分的資料，不過對於有些不會產生座標資料的指令來說，仍然可以去抓取裡面的座標資訊，不過抓到的會是預設值(1,1)之類的無意義資料。

在開始基本指令的解釋前，要在這裡先說明一下稍後會用到的一些表達格式，由於每個基本指令都是由指令名稱，加上接在後面的對應參數來組成，這些參數都有一定格式規範，簡單分爲以下三類

- 數值

就是這個參數所接收的是數值資料，例如座標數值，網格裡的某格資料之類，而網格內的資料會出現 A-Z 的字母，不過這是爲了方便使用者觀看所以以字元型態表示，是實上這些字母資料實際被用在規則裡時是由系統當作數值的，也就是當作 11-36 的數值來看待，因此更精確來說，從系統本身的角度來看，網格內的資料是由 1-36 這三十六個數字組成，只是在畫面上呈現的方式是 0-9，A-Z 這般。

除了可從網格內抓取的資料外，如先前所提到每個基本指令執行的結果都會回傳三筆資料，也是當作數值資料來看，在一組規則裡第 N 條指令回傳的資料以 LineN 表示，Line 因爲內含三筆資料，因此不算是單純的數值參數，而是下段會提到的範圍參數的一種，通常要先經過其他指令（例如 GET）將 Line 內的某筆資料取出，不過如果是要用第一筆資料的（也通常會是用第一筆），可以直接當成參數使用，只要直接標明它的 Line 編號即可，例如使用者把基本指令中的某個參數設成 Line2，就表示要使用規則中第二條指令執行後所回傳的預設（第一筆）資料當作參數。

因此，數值類型的參數會有 0-9, A-Z, Line N 這幾種合法寫法。不過除了這數值參數外，還有一些內定的系統變數也可以當作數值來用，會隨著目前狀況而有不同的值，例如表示目前 Grid 的 grid_num，以下於表 1 列出目前的全部系統參數，只有四個：

表 1：系統內定變數

grid_num	目前規則所選定的 grid 編號，可能值 1~9
section_num	目前所在 section，可能值 1~4

full_grid_num	目前規則所選定的 grid 編號(全域)，可能值 1~36
round	目前爲第幾回合的認證，可能值 1~5

這些系統參數被放在規則當中時，就會依當時的情況而代換成對應數值來使用，其中 round 由於會對應到目前的認證回合數值，有類似 C 語言中 for 迴圈的 i 值(index)的特性可利用，例如搭配稍後會介紹的使用者自訂 String 資料，可以達到每回合的認證規則有些許變化的效果。

- 範圍

有些指令的參數會用到範圍指定，例如抓取某「範圍」的資料，或是挑選某「範圍」，而且由於我們的四個 Section 合起來的範圍不小，因此必須要有指定某個小範圍的命名方法，因此就有用於指定特定範圍的「範圍」參數。

在前面的介紹有提到我們劃分整個網格架構的方式，由九個編號 1~9 的基本 Grid 形成一個 Section，由四個編號 1~4 的 Section 形成整個畫面，以及在整個畫面下能直接指定 1~Z 共 36 個 Grid 的 Full_Grid 型態，這邊的 Grid、Section 和 Full_Grid 就可以當作「範圍」的參數，至於範圍編號就由後面緊接著的「數值」參數來指定。

除了 Grid 和 Section 這類基本的範圍外，還有先前提到 Line，以及其他範圍參數例如 Direction，可以用 9 宮格式的方位來指定周圍的區域，以後面接著的 1~9 的數值參數表示方位。另外一個就是 String，由於我們的網格內容是設計成隨機產生的，假如使用者想要取用的是固定的資料的話，它可以將這些資料寫在 User Data 中提供使用者自訂 String 的地方，再用取網格資料的方式去取 String 內的資料，所以 String 跟 Grid 這類都是算一種範圍。

因此，範圍類型的參數會有 Grid、Section、Full_Grid、Line、Direction、String 這幾種合法寫法。

- 其他

數值和範圍以外的資料，例如做四則運算的加減乘除之類的符號參數。

以上就是三種參數類型的介紹，接下來就要舉出各個基本指令的實際

格式，一共有五個指令，GET、CALCULATE、DIRECTION、INDIRECT、OUTPUT。

- **GET**

這個指令的用途就是去找某個範圍內的某個資料，也就是類似模擬使用者去看某個地方資料的功能，例如 Grid 裡的資料，它的格式如下，而表 2 是參數內容：

GET data_info1, data_info2 **IN** area_type, area_number

表 2：GET 指令參數內容

data_info1	data_info2	area_type	area_number	描述
數值：1~36 爲座標時：1~6	(數值：1~6)	範圍：grid	數值：0~9 (grid 0 代表目前所在 grid)	取得目前 Section 中某 grid 的內容
數值：1~36 爲座標時：1~6	(數值：1~6)	範圍：full_grid	數值：1~36	取得整個畫面上某 grid 內容
數值：1~36 爲座標時：1~18	(數值：1~18)	範圍：section	數值：0~4	取得整個畫面上某 Section 對應到目前 grid 位置的內容
數值：1~36 爲座標時：1~6	(數值：1~6)	範圍：direction	數值：1~9	取得目前 grid 旁某方位 grid 的內容
數值：1~36 爲座標時：1~string 長度	(數值：1)	範圍：string	數值：1~3 (使用者可自訂 3 組字串)	取得 string constant 的內容
其他：x	(無)	範圍：Line(N)	(無)	取得某 Line 回傳的第二筆資料

其他：y	(無)	範圍：Line(N)	(無)	取得某 Line 回傳的第三筆資料

如先前所提到，雖然網格上的資料是以字元方式顯示在螢幕上，不過以系統觀點來看都是 1-36 的數字而已，所以對使用者來說，data_info1 的允許數值爲 1-Z 跟系統觀點的 1-36 是不互相衝突的。而 data_info 由於在表示座標以外的資料時不一定會兩個參數都用到，因此 data_info2 的部分以括號註明，而有些參數會有本身的合適範圍，例如普通 grid 編號只有 0-9，使用者設定參數或是規則運算中當作此參數的數值超出這個範圍的話，會將數值自動對應回此範圍內，利用取循環回來的方式，像是 10 超過了最大 grid 編號 9，就當作超過 9 之後又跳回 0，因此 grid 10 就當作 grid 0 來處理。

在表中還出現的一個特殊的範圍指定法，利用 direction 來指定，它的用意是去指定目前 grid（grid 0）的周圍的其他 grid 的用法，我們在前面的設定部分介紹到，一個 section 有九個 grid，就類似九宮格一般，而我們給這九個 grid 的編號就跟鍵盤上 numpad 的排列方式一樣，現在就要依照這種編號方式來說明 direction 用於指定範圍的情況，假設目前 grid（grid 0）是位在 grid 9 這個位置，當我們用 direction 2（下）來指定位置時，就等於是指定目前 grid 的下方的 grid 這個範圍，也就是 grid 6，注意，目前 grid（grid 0）是作爲那個回合認證時的基準點，在一回合的認證未完成之前都不會變動，因爲目前設定是只能靠 Next Grid 的運行才會去選擇其他 gird 來當次一回合的基準點（grid 0），而 Next Grid 是在 Rule 執行完一次後才會跟著執行的，也就是說我們在同一回合的接下來使用 direction 6（右）來指定位置的話，一樣是以 grid 9 爲基準，只是這次改爲指定它的右方的 grid，由於 grid 9 的右方不存在（同 section 的）grid，此時會採用一個繞回的方式，因此就變成指定 grid 7，同理當使用 direction 8（上）時，會往上繞回而去指定 grid 3。

還有，關於 string 範圍的部分，由於 string 是使用者自訂的字串，可看作是一個一維陣列型態的資料，就等於是 grid 資料中的「一列」一般，因此它的 data_info2，也就是一般情況下的 y 座標，只會用到 1 這個數值。

這邊舉一個 GET 的實際套用的情形，假設指令爲

GET 3,4 **IN** grid, 0

而目前網格如圖 28：

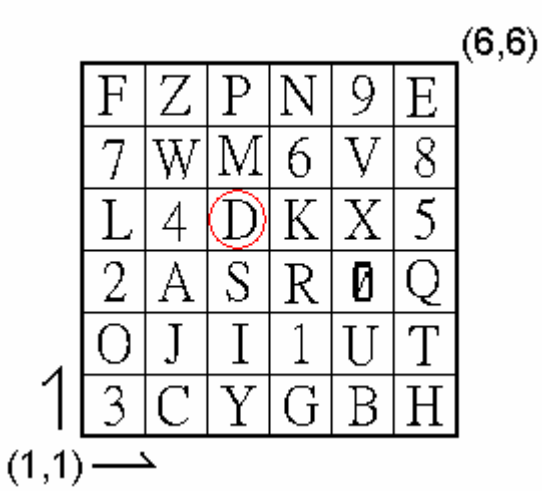

圖 28：GET 指令範例所用之基本網格

由於前面兩數值參數均有指定，因此是當作座標處理，而後半部參數指定範圍的部分，則是指定 grid 0，也就是本回合的目前所在位置的 grid，這個指令就等於去抓目前 grid 中座標(3,4)的地方的資料，如同先前介紹的統一回傳格式，等於回傳了三筆資料，也就是 D，3 和 4，代表得到的內容爲 D，而其座標在(3,4)這個位置，附帶一提，假如今天的指令爲

GET D,"" **IN** grid, 0

且同樣是套用在上圖的 grid 的話，那麼他回傳的資料一樣是 D，3 和 4。在這邊要再詳細一點說明關於字元跟數值的關係，由於 GET 回傳的資料中有字元類的資料，前面有提到說字元資料是對於使用者的觀點而言，實際上在系統內是把字元當成一般數值，1-Z 對應到 1-36 的詳細對應方式爲：

1 2 3 4 5 6 7 8 9 0 對應到 1 2 3 4 5 6 7 8 9 10

A B C D…Y Z 對應到 11 12 13 14…35 36

其中字元 0 不直接對應到 0 的原因是，0 這個資料在網格中被抓出來運算時，如果當作零的話，用在後續計算上意義不是很大，例如不論何數乘以 0 結果都是 0，這樣就缺少了由計算來產生多種結果的效果。

- **CALCULATE**

GET 主要是取得某資料，因此 CALCULATE 就是對資料作運算了，這裡的運算只是單純的兩數值參數作數學運算，格式如下，表 3 爲參數內容：

CALCULATE data1, data2, operation

表 3：CALCULATE 指令參數內容

data1	data2	operation	描述

數值	數值	其他：+	加法後結果
數值	數值	其他：-	減法後結果
數值	數值	其他：*	乘法後結果
數值	數值	其他：/	除法後整數商數
數值	數值	其他：%	同餘後結果

這個指令通常用來把其它指令回傳的資料拿來作運算，例如在上面的 GET 範例中，取得的三筆資料中第一筆爲 D，因此把這個指令拿來計算的話，在系統內就等於把 D 當成數字 14 來運算，如果想要抓取第二或第三筆資料來運算，那就要先用另一個 GET 把第二或第三筆資料先抓出來後再去用。先前有提到字元與數字的相互對應，因爲字元對應到的數字只有 1-36，而 CALCULATE 很容易就超過這個區間，超過區間的數當作密碼輸入時沒有問題，因爲密碼是用數字表示，不過當使用者要把區間外的數值當作字元去 grid 中尋找的話，就需要用跟之前一樣的字元循環方式對應回字元，例如數值 36 對應到字元 Z，而數值 37 則對應回字元 1，數值 38 對應字元 2，以此類推，而減法運算時可能產生負數，一樣以循環方式來看，數值 1 對應到字元 1，數值 0 對應到字元 Z，數值-1 對應到字元 Y，也就是當負數要轉成字元時跟 36 一次一次相加，直到大於 0 再轉換即可。在其他有數值區間限制的參數也是依這種循環方式來決定超過限制的數值要如何處理。
這個指令回傳的結果爲了統一格式一樣有三筆，第一筆當然是運算的結果，至於第二筆跟第三筆則是保留作爲未來其他可能功能用，目前的話是設定成存有第一筆資料的字元型態在目前 grid（grid 0）中的座標，所以一般規則制定雖然不禁止使用者去用 CALCULATE 回傳的第二或第三筆資料，不過這種用法的實用性有待討論。

- **DIRECTION**

這個指令是去看兩個物件在某範圍裡的相對位置來決定結果，他會回傳一個 1~9 的數字，用來表示由第一個物件到第二個物件的方位，方位的表式法跟鍵盤上 num pad 的排列一致，例如 3 是代表右下。這個指令的格式如下，參數內容於表 4：

DIRECTION　data1, data2, grid, num

表 4：DIRECTION 指令參數內容

data1	data2	grid	num	描述
數值：1~36	數值：1~36	範圍：grid	數值：0~9	使用目前 Section 中的某 grid
數值：1~36	數值：1~36	範圍：full_grid	數值：1~36	使用整個畫面上的某 grid
數值：1~36	數值：1~36	範圍：direction	數值：1~9	目前 grid 旁某方位的 grid

以下舉一個例子說明，假設指令是

DIRECTION W, E, full_grid, D

由於他所用的範圍是用全域 grid 型式，因此我們要去看整個畫面上三十六個 grid 中第 D（數值 14）個 grid 的內容，假設全域編號 14 的 grid 內容如圖 29：

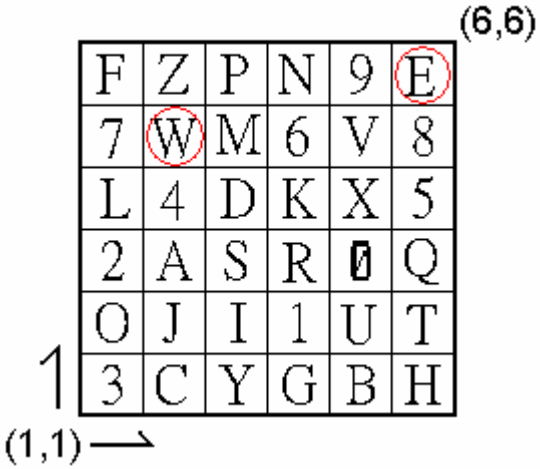

圖 29：DIRECTION 指令範例所用之基本網格

那麼這個指令所回傳的第一筆資料就是 9，因爲由 W 爲原點的話，E 在 W 的右上角，也就是 numpad 中 9 的方位，而這個回傳的數字 9 就可以例如拿去給計算指令用，至於第二筆跟第三筆資料目前則是預留位置而不使用，假如使用者誤用而去取這第二三筆資料，可以同 CALCULATE 指令的情況處理。

- **INDIRECT**

這個指令是讓使用者去使用間接選取的功能，在前幾節的地方舉過一個例子，就是用鄰近兩個 section 中的兩個物件來選取目前 section 中某 grid 裡的資料位置，他所回傳的是兩個物件的位置資訊組合成的座標，以及該座標對應到此 section 中的字元內容。它的格式如下，參數內容於表 5：

INDIRECT selector1, selector2, target

表 5：INDIRECT 指令參數內容

selector1	selector2	target	描述
數值：1~36	數值：1~36	數值：1~36	target 使用的是目前 grid 中的資料

這個指令要配合範例才能清楚了解，由於這個指令牽涉到的是 Section 級的範圍（18x18），爲了示意上的簡單，我們用 6X6 的範圍來說明，但套用在 Section 上也是一樣的意思。假如指令如下

INDIRECT S, A, R

表示使用者的目標是目前網格中的 R 字元，然後是用 A 和 S 來進行間接選取，而我們知道以 Section 的排列方式，周圍只會有兩個 Section 跟自己鄰近，假設目前網格是在左上的位置，那麼左下和右上的 Section 就是與自己鄰近的 Section，如圖 30：

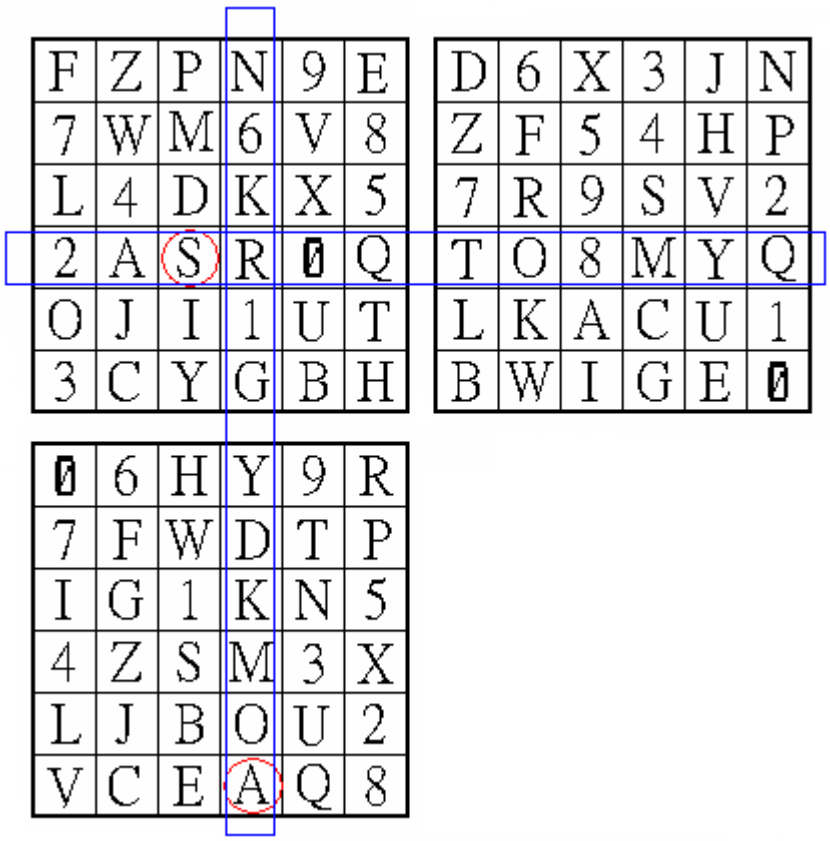

圖 30：INDIRECT 作用示意圖

從圖中可以發現，我們先找到 R 的位置，並橫向搜尋 S 所在的位置，再來也是由 R 出發，縱向尋找 A 的位置，這種作法由圖中即可了解，會類似在用 S 和 A 定位出 R 的位置，因爲他們交錯的點正好是 R。

而這項動作一樣會回傳三筆資料，第一筆是由第二和第三筆資料所形成的全域座標所含的資料，在這個例子是 E，這是因爲第二筆跟第三筆是定位資訊，也就是 S 的全域 X 座標和 A 的全域 Y 座標，在這個以 6x6 範圍代替 section 的例子來說，全域座標的範圍變成 1-12，S 位於全域 X 座標 3，A 位於全域 Y 座標 1，也就是此指令回傳的資料爲 E、3、1，也因此第一筆回傳的資料是全域座標(3,1)的資料：E。以上是範圍大小簡化過的例子，所以在實際套用在 Section 上時，第二筆和第三筆資料的值的範圍是可以有 1-36 的。

而在這邊會有一個問題，要是搜尋時出現不只一個定位字元會如何，例如在這個例子中橫向搜尋結果發現有兩個 S 時，這時只要取第一個看到的 S 來當參考即可（也就是取全域 X 座標較小的）。那麼還有一個可能是搜尋時定位字元沒出現，雖然在整個畫面的範圍下一整個橫排或縱列均有 36 個

字母，但不會正好三十六個字母都只出現一次，在找不到定位字元的情況下，就會以這個定位字元的數值當作座標值回傳，例如在例子中如果縱向蒐尋找不到 A，那麼第三筆回傳資料就會變成 11（A 的數值表示法）。
這個指令算是目前運行時需要使用者橫跨範圍最大的指令，雖然有點不符合希望盡量侷限在 6x6 基本網格的情形，不過這個指令所涵蓋到的範圍主要只到三個 section 的大小，而且出發點（target 參數）仍然是在目前 grid，再加上使用者搜尋 selector 時只要找與出發點同一列或同一排的資料即可，因此也不算真的需要搜尋大範圍的物件，但是這指令可達到產生不易推算的資料的效果。

- **OUTPUT**

這個指令就是代表使用者在中途或做完一系列動作後所要輸入的結果，也就是當作密碼輸入的部分，因此認證規則裡一定要有一個這個基本指令，但不限定只能出現一次，而是跟其他指令一樣可以重複使用，例如舉一個簡單的情況，假如把認證規則設成純粹只由三個 OUTPUT 指令構成，而指令內容分別是輸入 7、1 和 8，那麼這套認證規則代表的意義就跟使用傳統文字密碼系統並將密碼設成 718 是一樣的意思。
輸入的密碼用的是 0-9 的數字資料，但是規則的套用可能出現需要使用者輸入文字字元當密碼的情況，例如輸入 grid 中某座標的字元當密碼，此時使用者應自行將該字元轉成對應的數字後再輸入（例：B 轉成 12 來輸入）。因爲同一組認證規則對同一個內容的 grid 所產生的結果會是一樣的，所以認證方就可以藉由這個 OUTPUT 的結果來確認被認證方是否真的知道認證規則的內容。
雖然有了其他規則的協助達到混淆輸出資料的明確意義的結果，但我們還是希望使用者的輸入不要太過明確，因此在 OUTPUT 中也加入了可以讓使用者輸入假資料的功能，指令的格式如下，表 6 爲參數內容：

OUTPUT　data, Pattern A, Pattern B

表 6：OUTPUT 指令參數內容

data	patternA	patternB	描述
數值	數值：1~9	數值：1~9	依 pattern 形式輸入 data

其中 pattrenA 是指使用者這次 output 要輸入幾筆資料，patternB 則是要在第幾筆資料時輸入 data，因此在 patternA 筆資料中，只有第 patternB 位置的資料是實際資料，其它都是假資料，在認證時也只會對照 patternB 所指定的位置，不過假資料雖然是由使用者任意打入的數字，但是認證還是會要求這些假資料的數量要和 OUTPUT 指令的設定相符才行，例如使用者如果設定如下：

OUTPUT　Line4, 3, 2

表示使用者要輸入 Line4 所回傳的（第一筆）資料當作密碼，至於格式是以在輸入到第二個數字時，才使用 Line 4 的資料，之後再打入第三個隨機數字，以前段提到的字元 B 爲例，假設 Line 4 回傳的是 B，便是輸入「?12?」作爲此回合認證所需要的密碼，其中?的部分可隨意輸入 0-9 的數字。

所以在這樣的情況下，patternB 應當是要小於等於 patternA 才合理，如果使用者設定或是變數運算結果使然，造成 patternB 大於 patternA，那麼 patternA 的數值會自動和 patternB 互換，以符合 A 需要大於 B 的情形，例如假使是設定用「2,4」的格式輸入，是會被系統當成總共輸入四筆資料而在第二筆輸入真實 data 的情況。

4.4 規則設定介面

我們讓使用者自訂規則的方式是使用簡化過後的基本指令所組成，雖然規則的格式不像傳統的程式語言那麼多樣，但對於完全沒有程式寫作經驗的一般使用者可能還是不容易上手，爲了讓剛接觸這套方法的使用者能更快速的適應，可以設計一種 GUI 介面的小工具來讓使用者更方便的制定規則。

以下將提供一個介面設計的可能例子，首先由於基本指令只有五個，因此可以將這五個指令以按鈕的方式呈現在畫面上，如同圖 31 中的左方所示。至於圖中間的部分，則是當這些按鈕被按下去時，會在已制訂的規則列表中出現對應的基本指令，當點選中間規則列表中的該指令時，則會在右方出現對應該基本指令的相關參數，如果該參數是屬於較自由的數值或爲文字輸入的話，就採用使用者可直接打入的方式，如果是比較需要依照格式的參數，例如範圍或四則運算符號，則參數的設定可以採用下拉式功能表的方法來避免使用者輸入不合理的參數值。

圖中下方有著可以列出文字的空白部分，由於中間的制定規則列表是用比較易於使用者看懂的方式表達，但是系統在認證並實際讀取這些規則時，以適合該認證系統的語言來描述會更恰當，因此下方這個部份就是在規則制定好後，會列出轉換後適合認證系統讀取的規則格式，至於格式可以依照認證系統的特性作調整，這也是將規則介面設計界面獨立出來的好處。

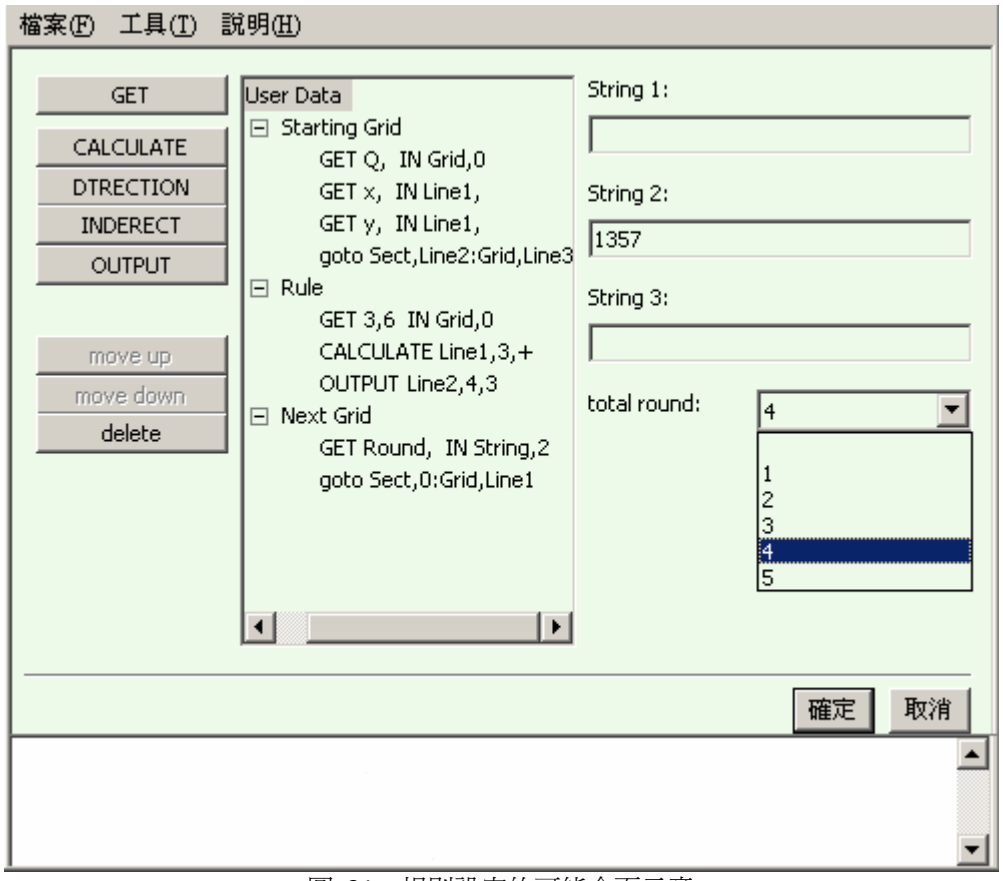

圖 31：規則設定的可能介面示意

此圖的右方資料是顯示出關於 User Data 的部分，雖然它不是一個指令，不過同樣會在右方出現可設定的參數，可以用來設定前文提到的使用者自訂 string 以及成功登入所需要的認證回數。較自由的 string 部分就以直接輸入的方式指定，而認證回數則以下拉式功能表來設定。

在圖中間的規則制定區可發現跟選擇目前 grid 有關的兩個項目，也就是 Starting Grid 和 Next Grid 中，都以一個沒看過的指令 goto 爲結尾，goto 是指令沒錯，不過他不算在基本指令內，它是預設一定會存在於這兩個項目之中，用於執行最後的 grid 選擇，由於它會實際改變到「目前 grid」（grid 0）所代表的位置，因此沒有列在能讓使用者新增的指令中，不過使用者可

以去設定它的參數。它有四個參數，前兩個是指定 section，後兩個是指定 grid，一樣是以”範圍參數,數值參數”的方式指定。

因此，使用者在利用這種介面編寫規則時，大部分只需要使用滑鼠點擊指令，不需要記憶太多語法相關的格式，因爲全部的指令都會列在左邊按鈕上，而該指令的格式基本型態也會自動在中間生成，使用者只需調整指令的參數即可，且各參數的作用即使在使用者並不太了解的情況下，也可藉由右邊出現的參數名稱和下拉式功能表中出現的資料類型來獲得一些提示。

五、運作流程與相關問題探討

此章主要是說明使用過程及一些範例，後半部則是探討安全方面的問題，例如不同側錄攻擊對本方法的影響程度，最後則是安全度與便利上的考量，以及使用者制定規則所需時間的預測及分析。

5.1 使用流程

我們的使用方法跟一般傳統密碼差不多，只是差在我們以規則設定來代替密碼設定，因此一開始跟一般密碼系統一樣要先註冊基本使用者資料，以便在忘記密碼時可跟相關人員確認使用者身分。

接下來是密碼設定，由於我們的方法沒有所謂的固定密碼，因此改由設定認證規則，使用者可以選擇使用 GUI 工具或是直接使用手動方式打出規則設定。

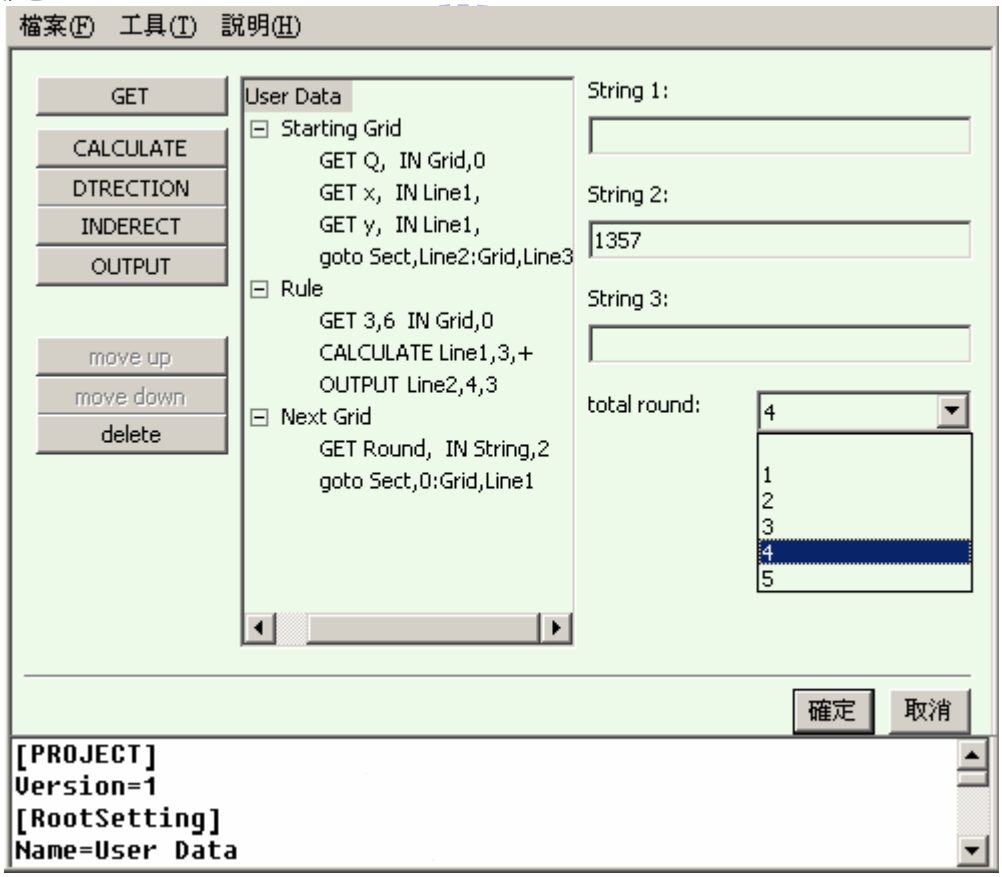

圖 32：由規則設計界面下方列出轉化後的規則格式

當規則設定完成後，在下方會產生具有特殊格式的規則編碼，如圖 32，跟一般便於使用者閱讀的形式不同，這邊的特定格式可隨著認證伺服器的

特性而選用較適合的格式來處理。規則轉換完成後便可以進行上傳規則的動作，使用 GUI 介面的使用者可以存檔後或直接將 GUI 視窗下方的格式規則如圖 33 複製貼上到上傳頁面。

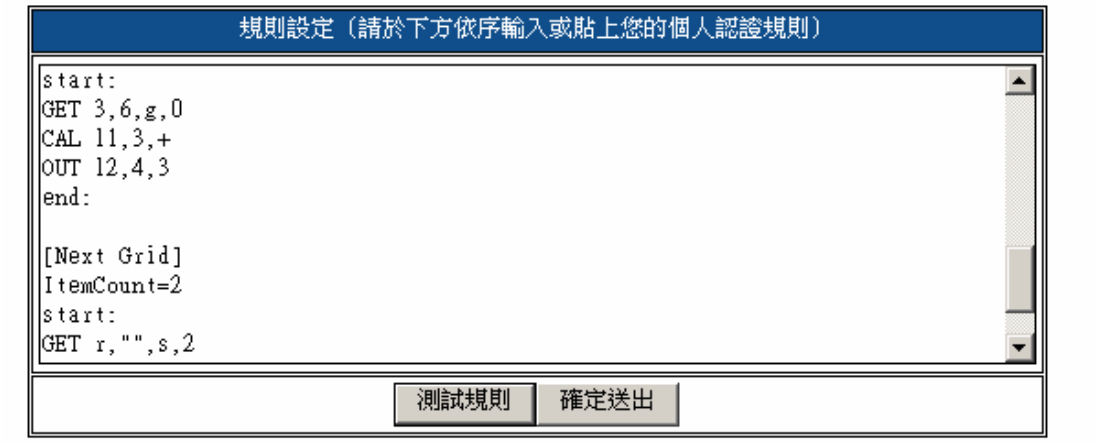

圖 33：規則格式後的上傳

在確認要使用這套規則前，可以先測試這套規則產生的是不是使用者預期所想的結果。在測試時，會在畫面上顯示出這套規則套用在範例網格上後所產生的密碼字串，如圖 34 所示，使用者可以藉由這字串比對和心中所想的預期結果有無相符，如果相符的話就表示使用者設定的規則跟期望中的一樣，此時就能確定套用這項規則。

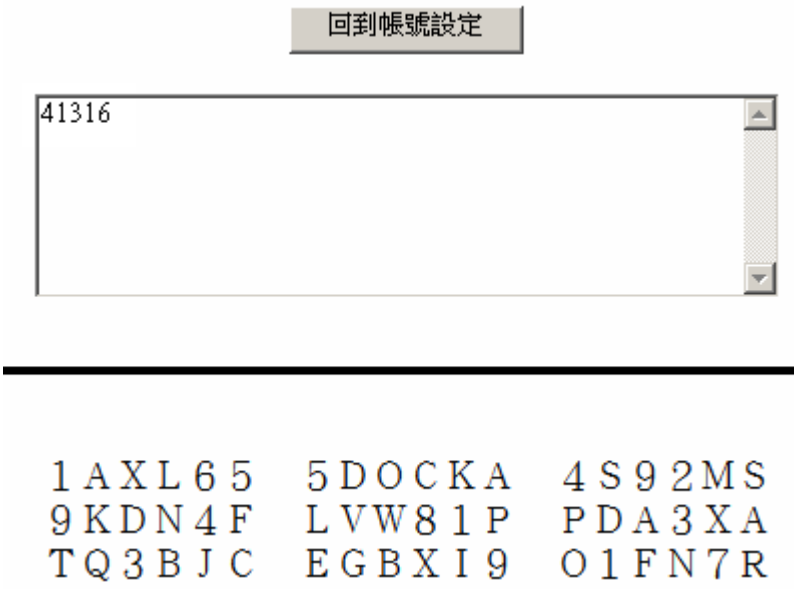

圖 34：儲存規則設定前的試用測試

5.2 登入範例說明

在經過規則設定以及上傳確認無誤之後，往後登入時使用者只要想著當初設定的規則做了哪些事，再將它套用在登入畫面中隨機產生的網格內容，並依序輸入各認證回合的 output 來把整串數字當作密碼使用，如果使用者輸入的跟規則的設定相符，就能成功登入，以下將舉一個部分的例子，假設依規則所選到目前網格內容如圖 35：

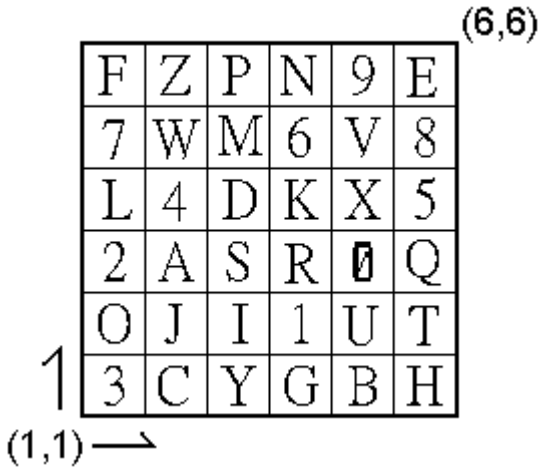

圖 35：登入範例第一回合所選中之基本網格的內容

然後把它套用在以下的規則上：

GET 4,2 IN grid,0
CALCULATE Line1,4,*
GET D,”” IN grid,0
DIRECTION Line2,Line3,grid,0
CALCULATE Line2,10,%
GET y,”” IN Line3,””
OUTPUT Line5,1,1
OUTPUT Line4,Line2,Line6

詳細來說就是去看目前網格中(4,2)所含的字元，將這個字元數值化再乘以 4 得到資料甲，再去看這資料甲跟目前網格中 D 的相對位置，並把這相對位置所代表的數字作爲密碼，資料甲 mod 10 當作另一個密碼，而將 GET D 所回傳的第三筆資料（D 的 y 座標）當作資料乙，接下來直接輸入剛才 mod

10 時得到的密碼，再來輸入格式爲輸入「資料甲」筆資料，在第「資料乙」筆才輸入另一個真實密碼，這套規則代入網格運算之後，將 Line 當作數值參數時所代表的結果替換後會變成：

(Line1) GET 4,2 IN grid,0　　（回傳 1, 4, 2）
(Line2) CALCULATE 1,4,*　　（1*4 = 4　回傳 4）
(Line3) GET D,"" IN grid,0　　（回傳 D, 3, 4）
(Line4) DIRECTION 4,D,grid,0　　（4 的左方是 D　回傳 6）
(Line5) CALCULATE 4,10,%　　（4%10 = 4　回傳 4）
(Line6) GET y,"" IN Line3,""　　（Line3 回傳的第三筆資料是 4 回傳 4）
(Line7) OUTPUT 4,1,1　　（輸入一筆數字　在第一筆輸入 Line5 回傳的 4）
(Line8) OUTPUT 6,4,4　　（輸入四筆數字　在第四筆輸入 Line4 回傳的 6）

所以我們這種認證方式，等於是用一連串事件來構成密碼，而驗證這「事件密碼」的方式就是比對它對此次隨機生成的網格資料所產生的通行碼，在此例就是把位於(4,2)表示的資料乘上 4 後去看它和 D 的相對位置，以及它 mod 10 的結果來當作通行碼，而輸入的格式則是依上述乘上 4 後的那筆資料和 D 的 y 座標來決定，也就是這一回合的認證需要輸入 4???6 當作密碼，問號部分可以是 0-9 任何數字。我們這邊是假設登入要多回認證，因此接下來是下個回合的認證，假設被挑到的網格內容爲圖 36：

(6,6)

0	6	H	Y	9	R
7	F	W	D	T	P
I	G	1	K	N	5
4	Z	S	M	3	X
L	J	B	O	U	2
V	C	E	A	Q	8

(1,1)

圖 36：登入範例第二回合所選中之基本網格的內容

這時套用相同規則後，把 O 看成數值 25 乘上 4 後得到 100，這個 100 mod 10 會得到第一個密碼 0，而 100 對應回字元爲減掉兩次 36 等於 28，28 看成字元爲 R，我們可以得到需要輸入 R 跟 D 的相對位置，也就是 1 來當作另一個密碼，格式爲 100,5，不過由於 OUTPUT 參數限制數值爲 1-9，所以把 100 依循環方式化簡後，格式變成 1,5，因爲 5 比 1 大因此是總共輸入 5 筆資料，在第 1 筆時輸入密碼，兩個 OUTPUT 合起來就是 01????。

在經過第二回合的認證後，此次的登入密碼和第一回串起來已累積成爲 4???601????，最後完整密碼便是經過使用者設定的回合數後所生成的數字串，如果這例子用的是設定兩回認證，那麼這串 4???601????就是此次登入所要用的密碼。在這個例子當中，認證過程會碰到一些不是很直覺的數學運算，雖然數值範圍較大的運算可以增加密碼的變化性，不過對於不擅長運算的使用者可以避免去用例子中的乘以 4 之類的設定，以免數值在運算中變得過於龐大，相反的，善於數學計算的使用者就可以利用這種方式，讓密碼更有變化。

5.3 對側錄攻擊上安全性的探討

在討論相關側錄攻擊對此方法安全性的影響前，要注意的一點是，在此考慮的，是在使用者登入時被攻擊的安全問題，而不會去考慮註冊或是修改密碼時被資料側錄的問題，因爲一般的系統，例如第三章所介紹的那些，都是藉由在註冊時設定一個認證方和被認證方都知道的一個秘密，在傳統文字密碼的情況下就是所謂的密碼設定，而這個秘密一旦洩漏就會造成帳號的安全問題，在沒有額外硬體輔助的情況下，想在註冊階段交換這個秘密又要抵抗側錄是非常困難的，但是在登入階段時就可能在不洩漏這些秘密的情況下確認雙方都知道這個秘密，這也是不靠額外硬體的認證方式在安全性上所想達成的目標，所以接下來將不會討論到註冊時或是修改密碼時的側錄問題。

由前幾章的介紹可知，目前的側錄攻擊多半朝三種目標裝置下手，也就是鍵盤、滑鼠和螢幕，一般在分析側錄攻擊對使用者登入時的安全性影響，也是從這三方面來分析，因此接下來便將一一探討這幾種裝置被側錄時，對於我們這套登入方式所產生的影響。

5.3.1 登入過程被鍵盤側錄

由於我們想讓使用者利用基本硬體就能進行登入，因此使用者必定會用到鍵盤或是滑鼠來進行認證步驟。在一般的密碼輸入情況下，透過鍵盤的輸入如果被側錄下來，就等於密碼也一併的被錄下來，不過這是在使用鍵盤輸入固定密碼的情形，因此後來就有了動態密碼或是改爲不經由鍵盤輸入密碼的方式，來避免鍵盤側錄可能造成的問題。

而在我們的登入方式中，雖然認證用的規則是固定的，但是密碼是會隨著網格資料的不同而變化，也就是類似上一段提到的藉由動態密碼來避免鍵盤側錄的問題，而且在沒有記錄到網格內容的情況下，攻擊者也無法對這個動態密碼的結果進行分析工作。

5.3.2 登入過程被鍵盤及滑鼠側錄

某些登入方法藉由以滑鼠來輸入密碼來代替鍵盤輸入的資料，讓鍵盤側錄無法得知密碼內容，但是後來也相對的出現了滑鼠的側錄工具，因此一些靠滑鼠輸入跟密碼關連性非常大的資料時，就容易被側錄後分析出登入時所需要用滑鼠做出的動作。

當然有些登入方式能做到即使用滑鼠登入，但是被側錄時亦不會透露太多訊息，例如滑鼠對某範圍的點選。而在目前我們的方法中，並沒有用到所謂的滑鼠輸入方法，所以滑鼠側錄並不是個問題，但是我們的登入方法或許可以套用這項機制，以加強密碼的變化度，例如加入一個形成範圍的基本指令,而輸出這個範圍當作密碼的方式就是用滑鼠去點擊這個範圍，不過由於在網格裡自訂範圍的問題需要較多考量，因此可能在往後未來的研究才會探討到。

5.3.3 登入過程被完整側錄

此時的情況就代表著類似肩窺攻擊的情況，由於鍵盤、滑鼠、螢幕都被側錄，而使用者又只能用滑鼠鍵盤這類的基本硬體，等於是除了使用者腦中所想的事情外，其他一切登入過程所輸入或螢幕出現的資料都會被記錄下來。

在這種情況下，登入時所輸入的固定密碼或是螢幕上出現的固定資料都將直接影響到登入時的安全性，因爲資料固定不會變化，被側錄下來後

可直接用於下次登入。但是我們的登入方式是採用隨機生成的網格資料，進而影響到此次登入時所需要的密碼，因此螢幕上的資訊和使用者的密碼輸入都是每次登入有不同變化的，所以直接側錄到的資料跟只有鍵盤側錄時的情況差不多，都無法用於下次登入。

不過這次攻擊者多了螢幕上的網格內容來當作分析依據，如果使用者將認證規則設計成直接用GET和OUTPUT指令來輸出特定位置的資料的話，那這些特定位置在比對網格資料跟密碼後是有可能推測出認證規則的內容，不過通常會需要準備多次的網格畫面跟對應密碼的組合，因爲在第四章有提到認證規則中的 OUTPUT 指令可以設定讓使用者輸入假資料，因此要分析的話還需要將假資料給排除才行，況且由於密碼是由數字構成，因此還得去猜測真實資料的部分是一位數或是兩位數組成之類，這也是我們選用數字密碼的原因，像是輸入 C 跟輸入成 13，前者就知道一定是表示 C，但後者可以有 1 真 3 假、1 假 3 真、13 皆真三種可能。除此之外，認證時除了檢查真實資料的部分外，第四章有提到假資料的資料數量也在認證時的檢查範圍內，OUTPUT 指令設定得當的話，雖然可能會增加認證時使用者的負擔，但是能產生每次登入時假資料數量本身的變化，進而影響到整體密碼的長度，更是爲攻擊者增加一個需要分析的變因。

那麼如果是密碼被亂猜的情況，由於網格加上規則產生的密碼是會隨者網格資料在改變的，因此依序嘗試不同密碼組合的暴力破解法並不有效，因爲密碼非固定，無法藉由窮舉錯誤密碼來減少需要檢查的密碼組合，這種情況比較像是每次猜測都要從所有可能性中去挑一個密碼組合來猜，變成了類似只能靠運氣的情況，加上大部分認證系統都有錯誤登入的次數限制，要在有限次數內猜中更是難上加難。但是這邊的動態密碼也稱不上完全隨機，因爲使用者設計出的認證規則可能無法完美的產生隨機密碼，即使網格資料是隨機產生，認證規則套用下去後所產生的密碼可能還是會有某種可預測的特性存在，如何檢查出這些特性並警告使用者將是可能的未來研究工作之一。

5.4 系統使用上便利性的探討

到目前爲止看到的這些無須額外硬體的密碼系統，都是處於一個安全性和便利度如何拿捏的情況下，通常方便使用的方法，帶來的可能是安全

性上的不足；而高安全度的系統，在便利度上例如登入時間卻又無法兼顧，因此安全性與便利度似乎無法兩全。

而我們的方法也不例外，雖然安全性方面是可行，但依照之前的經驗，應該是會犧牲便利度的部分，也就是我們還是會讓使用者多花點時間，不過我們的方法不是讓使用者多花時間在登入上，而是花時間在註冊時的思考及設定規則上，又因爲在使用的過程中，登入的次數一定比的註冊來得多，因此我們認爲讓註冊費時比登入費時來得划算。

我們的方法在制定規則時，如果利用 GUI 介面輔助，通常由滑鼠點擊輔以一些鍵盤鍵入即可完成一個基本指令的設定，正常操作下應該是十秒內即可完成一條基本指令，而一組基本的由三部分所組成的規則，應該也是由十六條基本指令即可完成，因此設定規則的時間大約落在二分鐘至三分鐘間。

爲了驗證這項時間，我們找了四十五位大學部學生來做測試，首先用十五分鐘的時間說明一些相關須知，接著給他們五分鐘的時間去思考要對網格架構做出哪些行爲，接下來便開始計時他們將這些對網格的行爲轉成規則的時間，且要求裡面三部份的規則都必須要設定。而 User Data 的部分則看規則有無使用到而定，因此這部分不一定要去設定。最後的結果如圖 37。

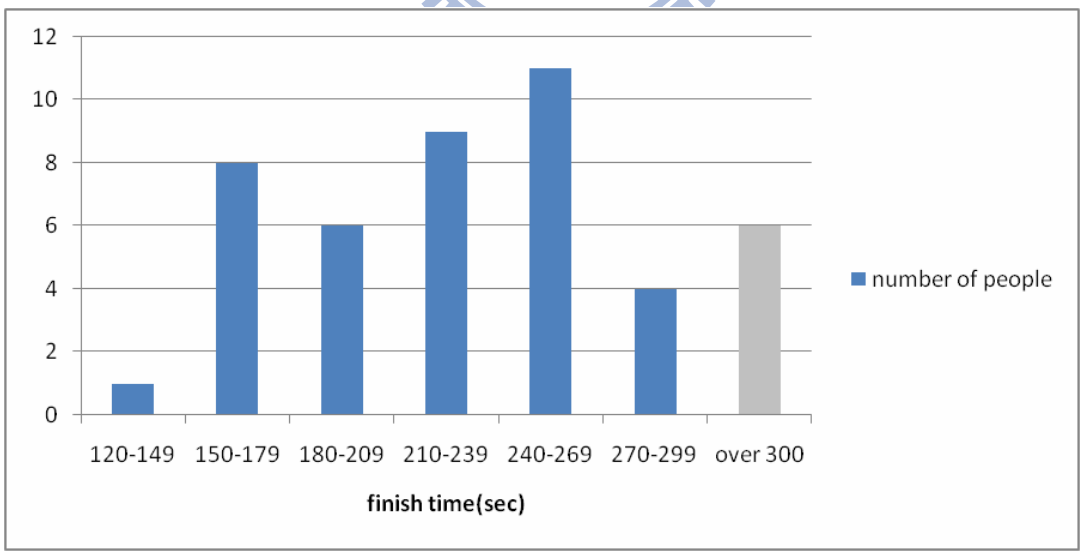

圖 37：完成網格認證規則所花時間（不包含構思內容的時間）

其中有六個人超過五分鐘，我們將他們當作可能沒聽清楚說明所以無法成功完成設定，不列入最後統計結果當中，因此最後的結果爲，最快完

成是 137 秒，最慢的爲 292 秒， 平均爲 218.7 秒，標準差爲 40.5。

我們從統計圖發現，在完成時間上有兩個高峰群，而不是趨近中間，我們推測應該是一般使用的心態所導致，因爲一群是習慣較高安全性的人，可能會寫出變化較大的規則，因此會多花點時間，而另一群講求便利即可，不需要太高安全性的人，便會制定一些較爲簡單的規則，制定上所花的時間也較少，這也是這種自訂認證規則系統的優點之一，可以讓使用者在便利度和安全度上做調整。

至於認證規則本身在記憶上可能不如一般傳統密碼來得習慣，不過如果不將認證規則當成是一條條規則，而是把這幾條規則結合起來當作是一體來看的話，就會較方便記憶，例如用文字 A 位於目前網格中的 y 座標數值，去減掉目前網格中(3,y)座標內的值，雖然寫成規則需要幾條基本動作，但是它所代表的行爲用兩句話就可以表達。

如果要追求跟傳統密碼相似的記憶方便性的話，甚至可以在本文提到的網格架構與規則基礎下稍對系統稍做改變即可達成，例如維持基本的網格架構，也就是一樣是 6 乘 6，內含三十六格的的基本網格，但是將畫面上原本網格內會出現的字母，再加上一條隨機產生的基本指令。而使用者所要設定的東西只有密碼，但是這組密碼當然不會直接拿來作登入時的輸入，而是利用密碼裡的文字，依序對應到網格裡三十六格（同樣是因爲數字加文字是共三十六種字元）中的其中幾格，然後使用者照著對應到的這幾格裡所寫的規則，進行類似本文前面提到的規則套用，這樣一來，等於是用一組固定密碼，在每次登入時產生不同的認證規則，再由這套認證規則產生此次登入所用的動態密碼，而使用者只要記憶當初註冊設定的固定密碼即可，但是以一組固定的文字密碼來延伸出認證規則會不會提高字典檔攻擊或被分析的風險，是值得思考的地方。不過由於本文意旨在介紹及探討網格規則的基礎架構，後續的架構延伸或變化便不在此文中仔細探討，或許可以成爲未來研究的方向之一。

在經過一連串對本文提出方法的介紹與分析後，第五章最後這邊將列出跟其他無需額外硬體而有肩窺抵禦功能的方法的簡單比較，包含第三章提到的圖形聯想式和幾何式方法，以及使用者自訂文字替換或是資料處理規則，表 7 爲比較表，各項評價使用大、中、小三層評比方式：

表 7：含肩窺抵禦的認證方式之簡單比較及說明

	圖文聯想[13]	**圖片圍成範圍**[10]	**自訂字元對應**[14]	**自訂認證程序**[14]	**網格規則(本文方法)**
註冊所需時間	長。 需要思考圖形與自串的對應方式	短。 挑選圖片即可	短。 設定字元對應即可	長。 需要設計並手動打入規則	中。 規則樣式可半自動生成
登入所需時間	中。 只需輸入圖片所對應字串	長。 使用者需要在大範圍搜尋物件	短。 輸入文字替換後的通行碼即可	中。 須將通行碼利用規則轉換	中。 對網格進行處理，搜尋範圍大部分侷限在 6x6 網格上
畫面上的物件量	中。 約近百個圖形整齊排列在畫面上。	大。 約上百個圖形隨機散佈在畫面上	中。 125 個元素排列成五個矩陣，由隨機 0-9、A-Z 的字元組成	中。 125 個元素排列成五個矩陣，由隨機二位數組成	大。 整體 36x36 的範圍，特定 6x6 範圍各含 0-9、A-Z 的隨機排列字元
密碼的記憶性	中。 由圖片提示但可能稍嫌不足	易。 回想註冊物件即可	難。 字元對應無特殊關連且繁多	中。 回想程式運算內容，難度隨內容設計的複雜度而增加	中。 回想對網格的行動，難度隨行動的豐富度而增加
由多次	易。 因字串本身的內容與圖	中。 圖形圍成範圍可分析但	中。 只得知轉換後的通行	難。 程式設計有非常多種可	難。 同樣由處理結果不易推

側錄分析密碼	片的對應關係爲固定式	爲數眾多的干擾物件會增加難度	碼，需要大量參考比對才能歸納出字元對應法	能性，靠處理結果很難推算出過程內容	算出過程行動，連處理結果本身都有多種解讀方法，

六、結論與未來研究方向

在最後的一章，將回顧一些前面部分所提出的內容，並且對於本文提出抵抗側錄攻擊的方式，總結一些相關的結論。在最後則是提出一些可能的未來研究方向，期望能讓此種認證方式的特性被完整發掘或歸納出來，以便於繼續改進方法，達到更方便或更安全的效果。

6.1 結論

我們使用了讓使用者自訂認證規則的方式，搭配上隨機產生的網格資料，就可以在不使用基本硬體以外的裝置下，達到類似產生動態密碼的效果。而這組動態密碼，是以註冊時使用者自行設計的一套規則所產生，假設註冊時所設定的東西能有效保密，那麼在登入時，即使這組動態密碼被記錄下來，因爲每次所輸入的密碼會隨著網格內容的變化而不同，攻擊者還是得從使用者登入時側錄到的資料，來想辦法還原出規則的內容，假使使用者用的規則不是非常單純，要還原出原本的規則由於有非常多種可能性，因此不是一件容易的事，加上假資料以及密碼內容變動的特性，更增加了分析的困難度。

而在規則制定上的分析中，我們發現在 GUI 介面的輔助下，假如使用者有先了解使用方式以及先想好要對網格做哪些動作，那麼將那些動作寫成規則通常只需要約三到四分鐘的時間，雖然比起傳統密碼想出一組密碼再輸入註冊的時間來的長許多，但是由於註冊的次數與登入的次數比起來相對少很多，因此我們認爲多花點時間在註冊階段以增加安全性是值得的。

6.2 未來研究方向

雖然我們已經爲此套方法定了一些用來訂規則的基本內建指令，但由於這是初次進行這方面的思考，恐怕無法涵蓋到所有情況，因此可能還有其它適合套用在網格架構裡的指令，例如加入類似區域定義的指令，讓使用者自行在網格中定義出一個範圍，然後可以將兩個範圍做交集的動作，或是輸出該範圍相關的特性，例如某邊長度、面積、周長或是範圍中的某一個字元之類的。不過指令種類越多是否會影響使用者在設定上的困難度也是我們值得注意的。

至於我們提出的這種認證方式它的安全度跟使用者所訂出的規則特性息息相關，如果能有一套機制能去分析使用者所訂出的規則的安全度，那麼便可以在當下提醒使用者將規則訂得更安全些，避免使用者直接套用容易被分析的規則而不自知，以維持登入時的安全性。

參考文獻

[1] C. Herley and D. Florˆencio. “How to login from an Internet caf´e without worrying about keyloggers”, In: Symposium on Usable Privacy and Security, 2006.

[2] Thorpe J. and van Oorschot P. C. , “Graphical dictionaries and the memorable space of graphical passwords”, Proceedings of the 13th USENIX Security Symposium, pp. 9-13, San Deigo, USA, 2004

[3] Rachna Dhamija, Adrian Perrig, “Deja Vu: A User Study Using Images for Authentication”, Proceedings of the 9th USENIX Security Symposium, 2000.

[4] Blonder, G., “Graphical passwords”, United States Patent 5559961, Lucent Technologies, Inc., Murray Hill, NJ, 1996

[5] Jermyn, I., Mayer, A., Monrose, F., Reiter, M. K. and Rubin, A. D. “The design and analysis of graphical passwords”, Proceedings of the 8th USENIX Security Symposium, 1999

[6] Agus Fanar Syukri , Eiji Okamoto , Masahiro Mambo, “A User Identification System Using Signature Written with Mouse”, Proceedings of the Third Australasian Conference on Information Security and Privacy, p.403-414, July 01, 1998

[7] Real User Corporation, “The Science Behind Passfaces” [Online], Available:http://www.realuser.com/published/ScienceBehindPassfaces.pdf, 2001

[8] Wiedenbeck, S., Waters, J., Birget, J., Brodskiy, A. and Memon, N. ”PassPoints: Design and longitudinal evaluation of a graphical password system”, International Journal of Human-Computer Studies, vol. 63, issues 1-2, 2005

[9] Passlogix Inc, www.passlogix.com, 2004

[10]L. Sobrado and J.-C. Birget. “Graphical passwords”. The Rutgers Scholar: An Electronic Bulletin for Undergraduate Research, 2002.

[11]Hoanca, B., Mock, K. ”Screen Oriented Technique for Reducing the Incidence of Shoulder Surfing”. In: International Conference on Security and Management (SAM) (2005).

[12]S.Man, D. Hong, and M. Mathews. “A shoulder surfing resistant graphical password scheme”. In Proceedings of International conference on security and management, Las Vegas, NV, 2003.

[13]Hong, D., Man, S., Hawes, B. and Mathews, M. “A password scheme strongly resistant to spy ware”, Proceedings of International conference on security and management, Las Vergas, NV, 2004

[14]楊文鋒，「使用者可定義之肩窺抵禦通行機制」，國立臺灣大學，碩士論文，民國 97 年。

[15]Darren Davis, Fabian Monrose, Michael Reiter, “On User Choice in Graphical Password Schemes”. Proceeding of the 13th Usenix Security Symposium, 2004.
[16]Ahmet Emir Dirik , Nasir Memon , Jean-Camille Birget, “Modeling user choice in the PassPoints graphical password scheme”, Proceedings of the 3rd symposium on Usable privacy and security, Pittsburgh, Pennsylvania, July 18-20, 2007
[17]Susan Wiedenbeck , Jim Waters , Leonardo Sobrado , Jean-Camille Birget, “Design and evaluation of a shoulder-surfing resistant graphical password scheme”, Proceedings of the working conference on Advanced visual interfaces, Venezia, Italy, May 23-26, 2006