國立臺灣師範大學應用電子科技研究所

碩士論文

指導教授：莊謙本博士

王宏鈴博士

模糊式加解密之研究

A Study of Fuzzy-based Encryption and Decryption

研究生：林連源　撰

中 華 民 國 九 十 六 年 一 月

模糊式加解密之研究

學生：林連源　　指導教授：莊謙本博士

王宏鈴博士

國立臺灣師範大學應用電子科技研究所碩士班

摘　　要

現今有許多密碼系統，這些密碼系統廣泛的被使用在各個領域，如銀行的提款機、金融業的財金系統、電子交易貨幣中，各個密碼系統都有其特定的特色、優點與缺點，本論文提出的方法是將 Fuzzy 演算法移植至資訊安全的領域中，使用了 Fuzzy 集合概念、Fuzzy 關係概念、Fuzzy 矩陣概念發展了一套密碼系統，並與一些近代的密碼系統作些分析比較，由實驗分析的結果知道其複雜度明顯的增加，相對地也較具安全性，並且密碼系統也有不錯的效能。

關鍵字：密碼系統、模糊加解密

A Study of Fuzzy-based Encryption and Decryption

Student: Lian-Yuan Lin

Advisors: Chien-Pen Chuang

Kenvi Wang

Institute of Applied Electronic Technology
National Taiwan Normal University

ABSTRACT

Many methods of cryptography are available nowadays. They are used in several situations such as financial processing system, electronic commerce and electronic data interchange. Each has its own specific purpose and implementation algorithm. But no one is without any disadvantage even the best policy is carried on. The thesis proposed a method based on Fuzzy Theory to improve the information security by increasing the complexity of decryption for fuzzy-based encryption source data. The simulation results of security quality are better than former cryptographies as in this paper. Thus, the fuzzy-based encryption algorithm can be used in cryptosystem if needed.

Keywords: cryptosystem, fuzzy-based encryption/decryption

謝　　誌

在台灣師範大學讀研究所的這些日子，內心要感謝的人時在太多了，因篇幅有限，倘若漏失之處，敬請見諒，首先要感謝莊謙本教授、王宏鈴教授這些日子以來的辛勤指導，若無指導教授們的專業素養傾囊相受，學生在研究上可得比一般人更加辛苦鑽研，在撰寫論文其間，也時常勞煩教授們不捨晝夜的批評指正，才得以使研究與論文能有所成果登上大雅之堂，學生再次感謝教授們的辛勞。

感謝應用電子科技研究所上各位師長們與助教們的辛勤指導與大力幫忙，讓學生在求學之路上獲益良多且時常感受到溫情，課堂上教授們的學識風采、教學熱誠使學生如沐春風。接著感謝口試委員們的建議與指正，使學生的研究更加縝密、論文更加完整。還有各位親愛的同學、學長學弟們、各校朋友、外籍友人，感謝你們在日常生活的互相扶持、熱心幫忙；學業上的鼓勵互助，過往的淚水與歡笑依然歷歷在目，心中細想真是受益匪淺，你們的往日情懷長存我心啊！

最後，要感謝我家人親友們的支持與鼓勵，使我在求學中牽掛的心能有所減緩，沒有你們的體諒與關懷，我將無法在此勤奮不懈的邁向高峰，對家人的感謝之心，實是溢於言表，本人因言辭拙劣，再請見諒。

林連源　謹誌

于台灣師範大學通信電子實驗室

200 年 01 月 28 日

目　錄

圖 目 錄

表 目 錄

第一章　緒論

1.1　研究背景

隨著科技的進步，網際網路已相當廣泛地被使用，二十一世紀是資訊爆炸的時代，多數企業與政府皆已邁向電子化，然而因網際網路的四通八達，各個組織機構皆須對其資料與資訊做好安全防護也漸漸地被重視。

在網路通訊系統中，當一個系統被發展出來，首先的設計結果考量是系統可否確實執行，再來是系統執行的效能，隨後才考量系統的安全性。就如當初微軟公司剛發行WINDOWS系統時，先以功能為主，然後是系統的效率，直到最近幾年的作業系統才重視到安全性。

隨著通訊網路的發達，駭客的攻擊也是時有所聞，之所以如此，因為最初的網際網路是以開放式架構為設計基礎，且包含了所有以電腦構成的網路與擴充性。然而，當初的工程師和發展人員並未對安全性給予相當的著墨，故使用者的個人資料有相當的危險性。

電腦系統中TCP/IP並沒有保護的功能，網際網路使用者在互相傳輸資料時，並不事先查驗身份。所以，若有不法份子假冒或者任意利用網路散播惡意程式與病毒，將會使正當使用者遭受惡意攻擊與破壞。因此，電腦系統必須配備一些安全防護功能，才得以有效確保使用網路時避免遭受攻擊。

在當今的電腦系統，如Microsoft、Linux、Sun在安全防護上已有不錯的成果，若有加裝安全防護功能的電腦使用者，其在傳輸資料之前，電腦會對這些資料做一些加密的動作，此方法除了具有防護功能外，還可確認身份與增進傳送效率等等功能。

1.2 研究動機與目的

在現今的網路通訊中，於安全防護上，約略須做到可將私人網域與廣域網路區隔出來，還有分辨並阻擋不法攻擊。為何於安全防護上，須有能力將不同性質的網域分隔開，因為網路俱有相連性的性質。當立即有危險性時，系統可立刻切斷連線斷絕網路，但是此法屬下下策，如此一來就因噎廢食，枉費了使用網際網路的目的。

目前的網路設備和資源有路由器、交換機、各式伺服器等等裝置，皆有可能成為攻擊目標而被癱瘓，然而要如何降低駭客的攻擊，其方法有許多種，因此，安全機制的建立是不可缺少的。

當用戶端使用網際網路來傳送電子郵件、電子文件或者進行一些電子交易行為等等，使用網路的確改善傳統的方式增進效率帶來許多的便利，這些是較正面的；然而，在進行這些行為時也須以另一角度思考，這些資料以封包的形式在網路上傳輸時，只要有心人士使用某些程式或設備來抓取這些封包，或者是假冒當事人、傳送偽造訊息，這是非常嚴重的問題，小則危害個人，大則以致於危害國家安全，所以必須以一些安全機制來保護資訊，以保護資訊的安全，然而，若要做到“絕對的”安全有相當的困難，可說是天方夜譚。

譬如，當Bob與Alice使用電子郵件互相聯絡時，若Bob傳送一封信給Alice，Alice收到信後首先可能對此封信件產生一些疑問，比如：這封信在傳送的過程中是否有被有心人士調換或修改；這封信真的是由Bob所寄送出來的還是假冒Bob寄出來的。這些問題是值得去思考的，所以必須有些安全機制來防護雙方的通訊。

現今各企業已完成電子化，並且形成全球共通趨勢，然而從安全性的觀點考量，一些密碼系統的演算法複雜度不夠深度，且見fuzzy理論別於一般理論，因而本論文提出的安全機制目的，為探討fuzzy在資訊安全加解密技術上的應用，並將其發展出來與一般相關方法作效能比較，以作為強化加解密技術之參考。

1.3 論文架構

本論文的架構分為四章。

第一章：

緒論，其包含了研究背景、研究動機、研究目的、論文架構。

第二章：

相關知識與文獻，介紹了與本研究相關的理論、知識，包含了密碼學架構、密碼系統功能、密碼分析概述、數論、Fuzzy 理論等等。

第三章：

研究理論與方法，本章說明了發展的密碼系統架構，並使用了理論上的哪些地方與演算流程，還有相關軟體的使用。

第四章：

密碼系統分析，對所發展的系統作一些性質上的分析。

第五章：

結論與未來研究方向，對整個研究作總結，並提出未來研究方向的建議。

第二章　相關知識與研究

2.1　密碼系統簡介

在現今的資訊系統中，加解密模式主要是由明文、密文、加密、解密、金鑰這五大元件所組成，此架構在企業對資訊安全需求的達成上扮演相當重要的角色，而其中的密碼系統安全機制更是最重要的一環。

在此介紹幾個密碼系統，並將密碼系統流程大略說明

2.1.1 密碼學架構

密碼學（Cryptology/Cryptography）就是指訊息加密、訊息隱藏的計算機科學，也可說是一切與密碼學有關的科學。密碼系統一般可分為加解密系統與數位簽章系統。加解密系統元件包括原文/明文（plaintext / cleartext）、密文（ciphertext）、加密（encryption / encipherment）、解密（decryption / decipherment），與加密/解密金鑰（encryption / decryption key）。其中前面所說的原文（明文）乃是指初始型態的資料；密文則是指呈現無意義或亂數（無規律性）型態的資料；加密轉換就是把明文轉變成密文的過程；解密轉換就是把密文還原成明文的過程；加密/解密金鑰的意思為驅使加密或解密轉換過程的一種方法。密碼系統的加密/解密轉換一般使用自數論上的排列(permutation)、組合(combination)、取代（replace）等等技巧。

因此，在密碼系統中，會有傳送端、接收端、破密端這些角色，傳送端要傳送資料給接收端時，會先經過某複雜無規律性的加密動作，再把訊息傳送至接收端，接收端收到訊息後，再作解密動作就可得到明文。而破密端則會在資料傳送中將訊息截取，然後試圖使用各種分析工具、破解方法來得知明文，或者，破密端偽裝成傳送端送出假訊息給接收端，使接收端真的以為來自所認識的傳送端。

密碼系統的組成大致分為下列：

明文P（plaintext）：未經加密的原始資料。

密文C（ciphertext）：將明文加密之後的訊息。

金鑰K（key）：指加密或解密的方法。

E_k（）：加密函數，E_k（P）＝C。

D_k（）：解密函數，D_k（C）＝P。

在一般的密碼系統中，傳送端將明文送至加密裝置E，再使用加密金鑰後K，明文就變成密文$C_k = E_k$（P），然後把密文C傳送至接收端，接收端收到密文C後，就使用解密裝置D和解密金鑰K，就可把密文還原回明文$C = D_k(C) = D_k(E_k(P))$，如下圖2-1所示。

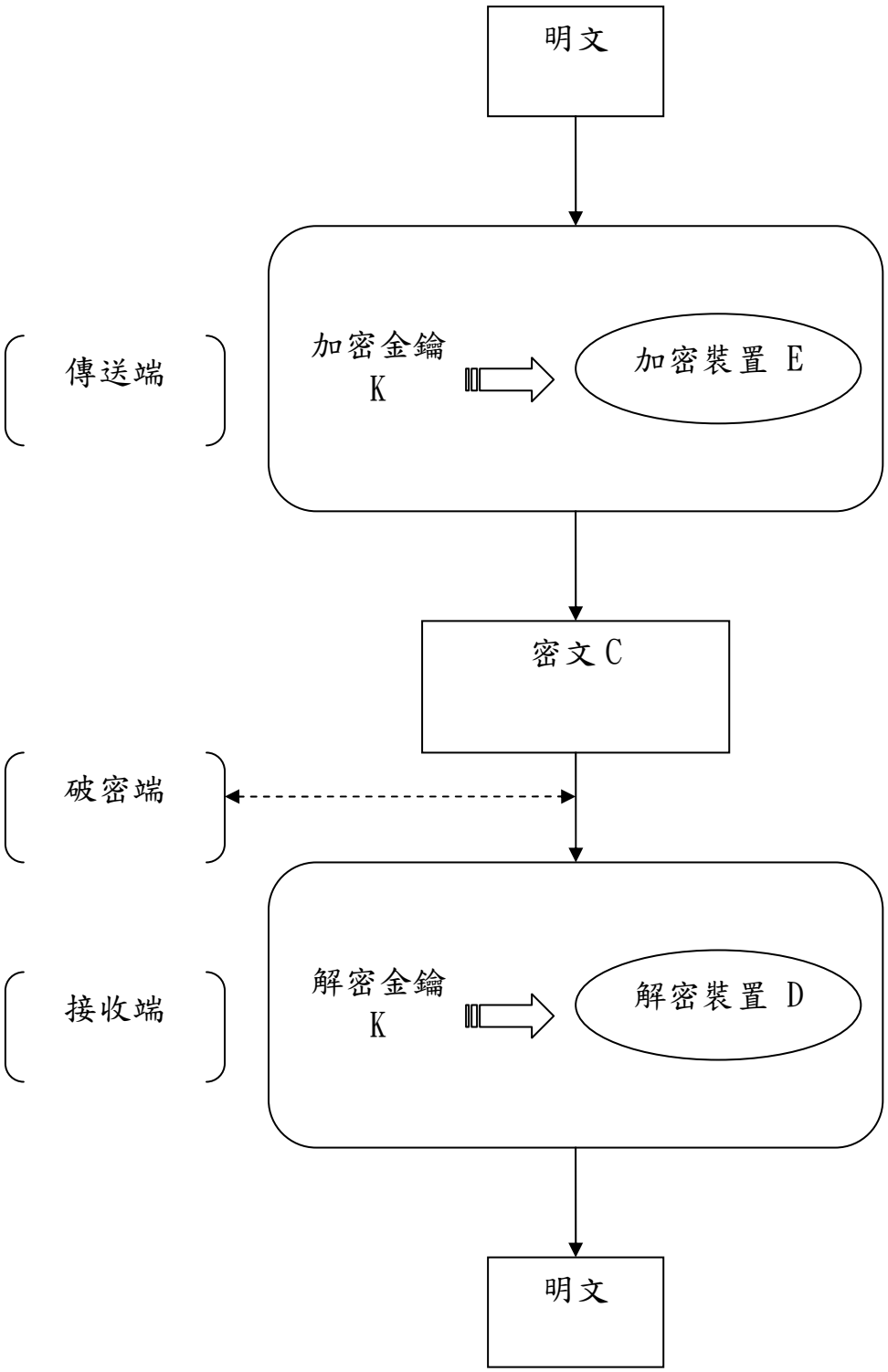

圖 2 - 1 密碼系統結構圖

2.1.2 密碼系統功能

一個完備的密碼系統會具備下列的功能：

1. 機密性（secrecy）：

防止非法使用者得到傳送的資料訊息，只允許合法的使用者取得訊息。保護的方式為在傳送資料的某個時間區段中，以最大範圍的方式保護所有被傳送的資料。例如，當兩用戶間的傳輸通訊協定連線之後，此種較廣大範圍的保護方式會確保之間的傳送資料不會洩漏。當然，也有其他形式的保護方式，例如：只保護資料的某些欄位，但是如此的保護方式會較無安全性。

2. 完整性（integrity）：

表示傳送的資料訊息未被非法的取代、變換順序或者是修改。保護的方式與機密性相當類似，可以最大範圍的方式保護所有被傳送的資料，當兩系統間的傳輸通訊協定連線之後，此種廣大範圍的保護方式會確保之間的傳送資料不會洩漏。當然也可只保護資料的某些欄位。

3. 確認性（authenticity）：

確認資訊的來源確確實實來自傳送端，並不是來自非法用戶。例如所收到的通知、警告訊息都是屬於此項。還有若兩台電腦要連線時，也得互相確認對方的身份，並不是由其他所偽裝來互相連線通訊進行，非法的傳送、接收資料。

4. 不可否認性（non-repudiation）：

傳送端所送出去的資料訊息一定是來自傳送端，無法否認其傳送的資料。

2.1.3 密碼分析概述

在密碼學領域當中，密碼分析也是個挺重要的問題，在建構密碼系統的演算法機制時，建構者必須考慮個基本的重要因素，就是想像一個具有強力破解密碼系統能力的分析家，為何要想像有此分析家呢？如此是為了讓密碼系系統建構者知道所建構的密碼系統，無時無刻都在面對著厲害的攻擊者與駭客等等的挑戰。密碼分析家們來自四面八方的個個層面，有來自本身具有先天的數理天份加上後天卓越的電腦科學技術能力的駭客還有來自國家機構、部門的人員，這些都是潛在的密碼分析強者，而且，這些人員可使用些技術、方法或設備的資源來達到他們分析密碼的目的。因此，密碼系統建構者對密碼分析家絕對不能掉以輕心，只要一輕忽他們，密碼系統可能遭受到不可估計的損害。

一般來說，密碼的分析大略有以下數種：

1. 只知明文的攻擊
2. 只知密文的攻擊
3. 已知明文、密文的攻擊
4. 攻擊明文

在第一種情形下，當密碼分析家得到明文資料的某部份，然後試著破解整個密碼系統。第二種情形是密碼分析家已經知道密文訊息，以此來攻擊密碼系統，希望得到所對應的明文訊息或金鑰，但是，因為密碼分析家只知道密文訊息，要以已知來破解密碼系統是頗為困難的。接下來的第三種情形，為密碼分析家已經拿到明文資料與密文資料，因此，密碼分析家會將攻擊主軸設在金鑰的身上，要是金鑰被破解出來，那麼整個密碼系統宣告瓦解。最後一種情形為密碼分析家利用統計技巧，將明文資料的某些特性抓出，進而對密碼系統測試，然後再從密文資料中得到密碼系統的特性，慢慢的邁向破解密碼系統。

在這一些密碼分析中，一般是常利用統計的方法來得到明文資料或密文資料的特性。經過系統的統計特徵分析，明文資料和密文資料間的某些特性或某些規

律的應對關係會顯現出來並加以使用，然後就攻擊到密碼系統的安全性了。

利用統計特性方法的密碼分析具有一定的殺傷力，在現今的電腦科技之下，已有發展專門用來密碼分析的統計特性晶片組，其破解密碼系統的時間花費已可達到微秒的水準，但是，隨著大型積體電路的摩爾定律與半導體業的技術日益進步，專門用來密碼分析的晶片設計技術也跟著提升了。

記得有本書籍—擁抱未來（The Road Ahead）、作者：Bill Gates，作者曾說：「由於系統的保密性及電子貨幣的安全性都依賴於加密技術，所以任何在數學或計算機科學方面的一個足以破解密碼體系的進展，都將是一場災難。」由這句話看來，未來的密碼系統的堅固性必須要有更高的水準。

2.1.4 私密金鑰密碼系統與公開金鑰密碼系統

在一般的密碼系統中，倘若加密的金鑰只有發送方知道，那麼這一種密碼系統就稱之為「私密金鑰密碼系統」，如之前圖2-1所示。一般來說，私密金鑰密碼系統的加密金鑰與解密金鑰是相同的，所以只要知道了加密金鑰，也就知道了解密金鑰；相反地，要是知道了解密金鑰，也就知道了加密金鑰。所以，私密金鑰密碼系統又稱之為「對稱金鑰系統」

私密金鑰密碼系統一般有以下的功能：

1. 確保資料機密性：

   當資料被加密過後，一定要擁有解密金鑰才可以解密得到原來的資料。

2. 確認傳送端的身份：

   接收端將一隨機產生的數送至傳送端，傳送端接收到後將之加密送回至接收端，若接收端將之解密後與原來的的數一樣，即可確認傳送端的身份無誤，並不是非法的傳送端。

3. 保護資料的完整性：

一般用於非重要性資料且確保不會被更改內容的情況，傳送端可將資料經過加密再附於明文之後傳送給接收端，接收端接收到之後，將密文解密並與明文做比對是否一樣。若一樣就表示明文正確，反之，即明文非完整性。

公開金鑰密碼系統又可稱為非對稱密碼系統，傳送端與接收端使用不同的金鑰，這兩個不同金鑰，一個為私密金鑰(private key)由擁有人自行保存、另一個為公開金鑰(public key)公諸大眾，兩個金鑰彼此配對使用，稱為「金鑰對」(Key Pair)。所以，每個人都可以拿公開金鑰來加密，要解密的話，就唯獨擁有私密金鑰的人才可以解密。

公開金鑰密碼系統的功能如下：

1. 確保資料機密性：

每個人都可以使用公開金鑰來加密，但是只有擁有私密金鑰的人才可以解密。

2. 金鑰管理方法：

任何一個人都可以產生公開金鑰與私密金鑰，因此除了高安全性之外，對金鑰的管理也相當簡便。

3. 不可否認性：

假設在網路交易的過程中，若買家否認之前對產品的採購需求，並拒絕付款，或者是賣家謊稱沒收到對方的付款，這些是令人頭大的問題。不可否認性的功能就是為了要解決這類抵賴的行徑。因此可以透過數位簽章功能來解決不可否認性的問題，因為數位簽章是經過獨一無二的私鑰配合雜湊函數運算加密而成的，所以其他人絕對無法仿製出一模一樣的數位簽章。

2.2 集合概述

2.2.1 明確集合

集合是用來表示性質、概念、運算，也可以表示推理與判斷，因此集合在各個領域的系統和語言都有被使用。集合是由一堆有共同特性的事物所組成的，它是用來表示一堆共同特徵事物的工具，例如：「某班級的男同學」、「公園的植物」、「所有的運動節目」等皆可被稱為集合。集合的特性如下：

1. 元素組合而成集合整體，元素可被區別出。
2. 處於同一集合中的所有元素具備某一共通的性質。

我們以圖來說明集合的概念，圖中為一個大橢圓包住一個小橢圓，其中大橢圓代表論域，而小橢圓則代表集合。何謂「論域」(universal of discourse)呢？就是當我們探討個具體的問題時，把所要考慮的目標侷限在一個範圍中，稱之為「論域」；何謂「集合」(set)呢？就是論域中的各個概念。舉個例子，探討『奇數』這個概念，就可將自然數設為論域，故論域的表示為

$$U = \{自然數論域\} = \{1,2,3,4\ldots\}$$

概念則是由論域中劃分出，因而所有的奇數可形成集合A，將其表示如下

$$A = \{奇數\} = \{1,3,5,7\ldots\}$$

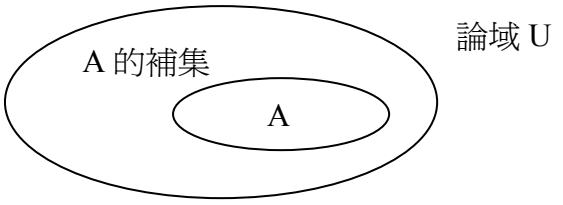

圖 2 - 2 集合示意圖

然而所有的偶數所形成的集合為A的補集(complement set)，並以$\overline{A}$示之，與A集合是完全相反的，或者說是論域U中，除了A集合之外的集合。

在集合中的元素數量並無限制，所以可以有限的或無限的，我們再由下面的定義知道如有限集合、無限集合、空集合的特質：

有限集合(finite set)：

集合中含有可以數得出來元素。舉個例子，書架中某層有幾本書，因為它是有限的元素。

無限集合(infinite set)：

集合中含有可以數也數不盡的元素。舉個例子，整數中有多少個偶數，因為它是有限的元素。

空集合(empty set)：

在集合中不含任何元素，並以∅表示。

2.2.2 集合的表現形式

1. 逐一列出法

若集合中有許多元素，且是有限個數的話，我們可以此方法表示。舉個例子，若欲表示一組1至20的偶數，可以一集合A = {2,4,6,8,10,12,14,16,18,20}表示之。

2. 敘述法

若集合中有許多元素，但無法一個一個地表示出來時，就可以採取此方法來表示這些元素的共同性質。舉個例子，若一集合為A = {2,4,6,8,…,100}的話，也可將之敘述為：

A = {x ｜ x為偶數, x ≦ 100}……..…（12）

2.2.3 特徵函數

一般我們探討論域時，論域中的每一個目標都稱為「元素」，就某個概念來說，元素存在著屬於與不屬於的關係，舉例來說，一數字10為自然數論域中的一個元素，但10不是奇數。所以，若在一論域X中設一變數 u代表元素，當要檢視u是不是某個概念時，一般用A(u)表示之，且讀做「u是A」，這就是述語（predicate）例如：u是奇數。若元素u屬於集合A時，就以 u∈A表示；反之，若元素 u 並不

屬於集合A時，就以 u∉A表示之。[12]

若 u 不變時，述語也可稱為命題（proposition），舉個例子，3是奇數。命題就是用來判斷一句話的真假，若命題為真，就可以用 T 來表示真（truth）；若命題為假，就可以用 F 來表示假（false）。因此，再由圖2-1來看，可衍生出下面式子的關係式：

$$T(A(u)) = \begin{cases} 1 & \text{若且爲若 } u \in A \\ 0 & \text{若且爲若 } u \notin A \end{cases} \quad \text{.....(13)}$$

可利用特徵函數來表示元素與集合的關係，由圖2.1知集合A的特徵函數為

$$\chi_A(u) = \begin{cases} 1 & \forall u \in A \\ 0 & \forall u \notin A \end{cases} \quad \text{.....(14)}$$

我們將上式集合的特徵函數以圖形表示，如下：

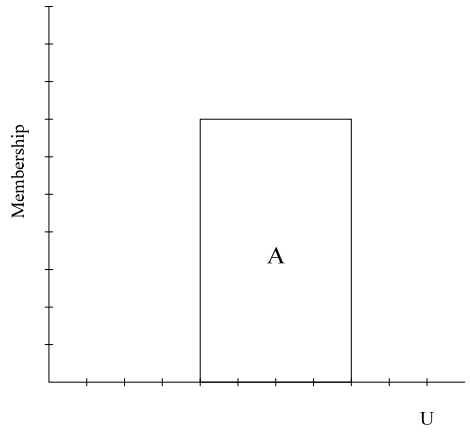

圖 2 - 3 明確集合特徵函數

假設 u 為A的元素，那麼特徵函數 $\chi_A(u)$ 是 1 ；若 u 為A的元素，那麼特徵函數 $\chi_A(u)$ 是 0 。並且以 $(u, \chi_A(u))$ 如此的式子來表達集合中元素與特徵值間的關係，欲表示集合論中「非0即1」的特性，可使用特徵函數來表示之，特徵函數的值域可寫為

$$\chi_A(u) \rightarrow \{0,1\} \quad \text{.....(15)}$$

2.3 Fuzzy理論概述

此理論的發展已有40年之久，其為將世間存在的模糊性發展成一門學問，創始人是美國的一位學者Lotfi.A.Zadeh所率先提出的，把模糊性轉化為「量」的形式且工具化，因此不明確的模糊性可被表達出來，其把人們語言中的模糊話語表示為量化有不錯的效果。

2.3.1 Fuzzy集合相關定義

由Fuzzy集合來看，論域上的元素 u ，對概念 p 得思索下列三點命題：

1. u 含有概念 p 的性質（即 u 是 p）。
2. u 未含有概念 p 的性質（即 u 非 p）。
3. u 使前兩點命提在某些程度上可成立。

在傳統的集合論上只思索前兩點命題，是屬於「非0即1」的觀點。而Fuzzy集合多加考慮第三點命題，如此一來就轉變為「亦此亦彼」。換句話說，就將傳統明確集合「一定屬於」的概念轉變為「相對屬於」的概念。除此之外，Fuzzy集合含有在論域中不同元素對於相同集合有不一樣的歸屬程度（grade of membership），等於就將「屬於」的概念量化，進而敘述了元素與集合的關係。倘若以X表示論域，以 u 表示元素，將一些Fuzzy的定義敘述於下：

定義　模糊集合（Fuzzy Sets）

於論域X中的某個 Fuzzy 集合 $\underset{\sim}{A}$ ，並設 $u_{\underset{\sim}{A}}$ 表示歸屬函數來描述集合的特性，並以[1,0]的區間來取值。也就是說，歸屬函數可以下式來表示閉區間的對應。

$$u_{\underset{\sim}{A}} : X \rightarrow [0,1] \quad \text{.....(16)}$$

除此之外，Fuzzy集合 $\underset{\sim}{A}$ 可以一組有序數對（ordered pairs）來表示，如下式所示：

$$\underset{\sim}{A} = \{(u, u_{\underset{\sim}{A}}(u)) \,|\, u \in X\} \quad \text{.....(17)}$$

所以，對於定義域X上的一個Fuzzy集合 $\underset{\sim}{A}$ 與一個元素 u ，不再是輕易地說 u 屬不屬於Fuzzy集合 $\underset{\sim}{A}$ ，而是說 u 在某個程度上屬於Fuzzy集合 $\underset{\sim}{A}$ ，並且 u 屬

於Fuzzy集合 $\underset{\sim}{A}$ 的程度指標記做 $u_{\underset{\sim}{A}}$，稱為歸屬度。由歸屬度 $u_{\underset{\sim}{A}}$ 來表示一個Fuzzy集合 $\underset{\sim}{A}$，並將歸屬函數值定義在[0,1]內，且歸屬度值可為任一實數值。要反應某個元素 u 在Fuzzy集合 $\underset{\sim}{A}$ 的程度，可由歸屬函數 $u_{\underset{\sim}{A}}(u)$ 的值大小得知。很清楚的可以明瞭，明確集合的特徵函數不是0就是1的表示方式；Fuzzy集合的特徵函數為0到1之間的任一值。舉個例子如下：

例：

設人類年齡0歲至150歲的論域為 X = [0,150]，說到「青年」兩個字，每個人對青年的年齡認定並不一致且相當模糊的。假設許多人的認定是20歲至30歲是「青年」，將其明確集合轉化為特徵函數並以圖表示，那麼，將會在20歲至30歲以值1來表示之，其他年齡則為值0來表示。如此的年齡區劃，相信大多數人並不苟同，會有以下幾點的問題：

1. 年齡20歲與30歲相差了10歲，怎能同樣以值1來表示？
2. 若再過數日才滿20歲，為何就不算是青年？
3. 若年齡剛過30歲，為何就不算是青年？

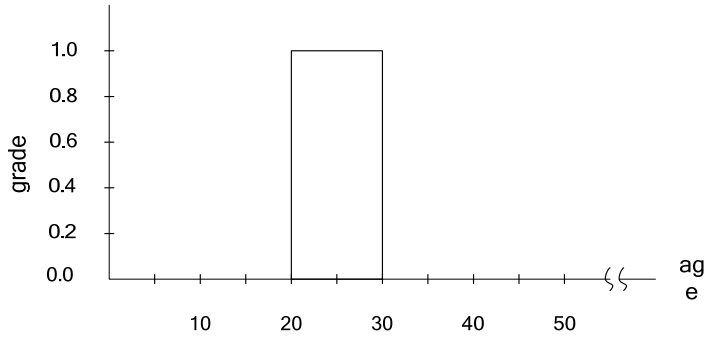

圖 2 - 4 特徵函數表示青年的明確集合

接下來要拿歸屬函數表示出每個年齡對於「青年」的程度，就以 $\underset{\sim}{F}$ 來表示「青年」的Fuzzy集合，並將圖2-3的特徵函數圖轉變為以特徵函數以特徵函數曲線表示「青年」的Fuzzy集合。

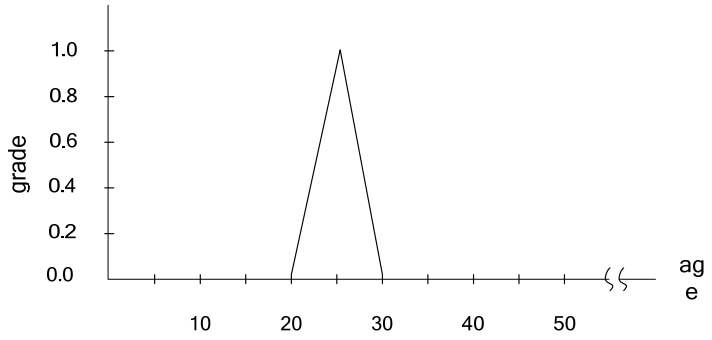

圖 2 - 5 特徵函數表示青年的 Fuzzy 集合

對於Fuzzy集合的歸屬函數敘述至此，可知其在Fuzzy理論上所佔的重要地位，因此Fuzzy歸屬函數和集合的特性，最後將其統整概述如下：

1. 論域中的所有元素 u 皆相當清楚的屬於X，因此X算是明確集合，只有論域的子集合 $\underset{\sim}{A}$ 才為Fuzzy集合。
2. 論域X中將一堆具有共同某種性質的元素劃分在一個沒有明顯分界的群體，此群體即為論域X的Fuzzy集合，以 $\underset{\sim}{A}$ 記號表示之。在論域中的各個元素可依某種性質給予個介於0至1之間的數值表示其 u 屬於 $\underset{\sim}{A}$ 的程度，這數值就是元素 u 在Fuzzy集合的歸屬度（grade of membership）。
3. $\underset{\sim}{A}$ (u)的歸屬度愈低，表示 u 愈不屬於 $\underset{\sim}{A}$ ， 若 $\underset{\sim}{A}$ 值等於0的話就表示完全不屬於 $\underset{\sim}{A}$ ；換句話說，若 $\underset{\sim}{A}$ 的值越接近1，就表示 u 越屬於 $\underset{\sim}{A}$ ，當 $\underset{\sim}{A}$ 完全等於 1 時，就表示 u 完全不屬於 $\underset{\sim}{A}$ 。倘若 $\underset{\sim}{A}$ ＝0.5來說，恰好是在「完全屬於」與「完全不屬於」中間，即表示 u 相當的模糊。

2.3.2 有限Fuzzy集合的表示法

Fuzzy的創始人Zadeh教授提出了一種數學式來表示有限Fuzzy集合，對於一有限的元素數量，其數量若為可數的由1至n，可以下式來表示出來，此種表示方式與2.2.2節集合的表現形式中的逐一列舉法相當類似。

$$\underset{\sim}{A}=\frac{\mu_{\underset{\sim}{A}}(u_1)}{u_1}+\frac{\mu_{\underset{\sim}{A}}(u_2)}{u_2}+.....+\frac{\mu_{\underset{\sim}{A}}(u_n)}{u_n}=\sum_{i=1}^{n}\frac{\mu_{\underset{\sim}{A}}(u_i)}{u_i},\quad \forall_i \in X \quad \text{.....（18）}$$

在這所談的 $\frac{\mu_{\underset{\sim}{A}}(u_i)}{u_i}$ 並不是一般數理中表示的分數表示式，它只是被用來表

示元素u與相對 $\tilde{A}$ 歸屬度的對應關係，因此不得被看做分數般來執行約分的動作。除此之外，式子的Σ並不是「總和」的運算符號，符號「+」也並不能看做加法運算，這幾個符號在Fuzzy理論中，只是用來連結各個元素和歸屬函數對應關係的符號。

例： 將Fuzzy集合以有限Fuzzy集合 $\tilde{F}$ 表示法展現

$$\tilde{F} = \frac{0}{22}+\frac{0.1}{23}+\frac{0.2}{24}+\frac{0.3}{25}+\frac{0.5}{26}+\frac{0.6}{27}+\frac{0.7}{28}+\frac{0.8}{29}+\frac{1.0}{30}+\frac{0.8}{31}+\frac{0.7}{32}+\frac{0.6}{33}+\frac{0.5}{34}+\frac{0.3}{35}+\frac{0.2}{36}+\frac{0.1}{37}+\frac{0}{38}$$

2.3.3 Fuzzy集合相關運算

明確集合與Fuzzy集合兩集合有些許相似之處，那就是論域X的元素是確定的，Fuzzy集合以歸屬函數來敘述集合和元素之間的關係，就好像明確集合是用特徵函數來敘述集合和元素之間的關係雷同。因此Fuzzy集合也得探討兩集合間的關係。

定義　Fuzzy集合（Fuzzy Sets）

若 $\wp(x)$ 下有Fuzzy集合 $\tilde{A}$ 和Fuzzy集合 $\tilde{B}$ ，其中 $\tilde{A}$ 中各個元素的歸屬度都小於等於 $\tilde{B}$ 的歸屬度，就叫做 $\tilde{A}$ 為 $\tilde{B}$ 的Fuzzy子集合，以符號 $\tilde{B} \supseteq \tilde{A}$ ，其意思是 $\tilde{B}$ 包含 $\tilde{A}$ ；或記做 $\tilde{A} \subseteq \tilde{B}$ ，其意思是 $\tilde{A}$ 包含於 $\tilde{B}$ 。倘若以歸屬函數說明，則表示為

$$\tilde{A} \subseteq \tilde{B} = \tilde{B} \supseteq \tilde{A} \equiv \mu_{\tilde{A}}(u) \le \mu_{\tilde{B}}(u),\ \forall u \in X$$

當 $\tilde{A}$ 的所有元素皆為 $\tilde{B}$ 的元素，且 $\tilde{B}$ 中的幾個元素不是 $\tilde{A}$ 中的元素，則稱做 $\tilde{A}$ 為 $\tilde{B}$ 的完全子集合(proper subset)，或者是稱做 $\tilde{A}$ 整個完全被包含於 $\tilde{B}$ ，以 $\tilde{A} \subseteq \tilde{B}$ 表示。

定義　Fuzzy相等集合（Fuzzy equal set）

若 $\wp(x)$ 下有Fuzzy集合 $\tilde{A}$ 和Fuzzy集合 $\tilde{B}$ ，假如 $\tilde{A} \subseteq \tilde{B}$ 且 $\tilde{B} \subseteq \tilde{A}$ 成立時，那麼稱 $\tilde{B}$ 、 $\tilde{A}$ 是相等集合，並記做 $\tilde{A} = \tilde{B}$ 。若將其以歸屬函數表示，如下式所示：

$$\tilde{A} = \tilde{B} \equiv \mu_{\tilde{A}}(u) = \mu_{\tilde{B}}(u) \quad , \forall u \in X \quad \cdots\cdots (19)$$

定義　Fuzzzy聯集（Fuzzy union set）

假設 $\wp(x)$ 下有Fuzzy集合 $\tilde{A}$ 和Fuzzy集合 $\tilde{B}$ ，假如Fuzzy集合 $\tilde{C}$ 為 $\tilde{A}$ 和 $\tilde{B}$ 的聯集，表示方式為 $\tilde{C} = \tilde{A} \cup \tilde{B}$ 。若要以歸屬函數來說明時，其表示方式如下，運算時為取兩者間的最大值，其中的符號 $\vee$ 為Zadeh所定的交集運算符號。

$$\begin{aligned} \mu_{\tilde{A} \cup \tilde{B}}(u) &= \max\{\mu_{\tilde{A}}(u), \mu_{\tilde{B}}(u)\} \\ &= \mu_{\tilde{A}}(u) \vee \mu_{\tilde{B}}(u) \quad , \quad \forall u \in X \end{aligned} \quad \cdots\cdots (20)$$

除此之外，因上式為兩個Fuzzy集合的運算，若往廣義方向思考，可將其擴展為多個Fuzzy集合聯集運算，假設若有n個Fuzzy集合，則其n個集合的交集運算如下式所示：

$$(\bigcup_{i=0}^{n} \tilde{A}_i(u)) = \max_{i=0}^{n} \mu_{\tilde{A}_i}(u) = \bigvee_{i=0}^{n} \mu_{\tilde{A}}(u), \quad \forall u \in X \quad \cdots\cdots (21)$$

定義　Fuzzzy交集（Fuzzy intersection set）

假設 $\wp(x)$ 下有Fuzzy集合 $\tilde{A}$ 和Fuzzy集合 $\tilde{B}$ ，假如Fuzzy集合 $\tilde{C}$ 為 $\tilde{A}$ 和 $\tilde{B}$ 的交集，表示方式為 $\tilde{C} = \tilde{A} \cup \tilde{B}$ 。若要以歸屬函數來說明時，其表示方式如下，運算時為取兩者間的最小值，其中的符號 $\wedge$ 為Zadeh所定的交集運算符號。

$$\begin{aligned} \mu_{\tilde{A} \cap \tilde{B}}(u) &= \min\{\mu_{\tilde{A}}(u), \mu_{\tilde{B}}(u)\} \\ &= \mu_{\tilde{A}}(u) \wedge \mu_{\tilde{B}}(u) \quad , \quad \forall u \in X \end{aligned} \quad \cdots\cdots (22)$$

除此之外，因上式為兩個Fuzzy集合的運算，若往廣義方向思考，可將其擴展為多個Fuzzy集交集運算，假設若有n個Fuzzy集合，則其n個集合的交集運算如下式所示：

$$(\bigcap_{i=0}^{n} \underset{\sim}{A}_i(u)) = \min_{i=0}^{n} \mu_{\underset{\sim}{A}_i}(u) = \bigwedge_{i=0}^{n} \mu_{\underset{\sim}{A}}(u), \quad \forall u \in X \quad \text{.....(23)}$$

若Fuzzy集合 $\underset{\sim}{A}$ 和Fuzzy集合 $\underset{\sim}{B}$ 的交集運算為空集合時，以 $\underset{\sim}{A} \cap \underset{\sim}{B} = \varnothing$ ，就稱此集合為無連接（disjoint）。[13]

例：

假設有5個同學組成集合論域X = {x_1,x_2,x_3,x_4,x_5}，而Fuzzy集合 $\underset{\sim}{A}$ 表示「喜歡運動」、Fuzzy集合 $\underset{\sim}{B}$ 表示「喜歡閱讀」，將其表示於下

$$\underset{\sim}{A} = \frac{0.2}{x_1} + \frac{0.4}{x_2} + \frac{0.8}{x_3} + \frac{1.0}{x_4} + \frac{0.3}{x_5}$$

$$\underset{\sim}{B} = \frac{1.0}{x_1} + \frac{0.7}{x_2} + \frac{0.6}{x_3} + \frac{0.4}{x_4} + \frac{0.1}{x_5}$$

以下將做Fuzzy集合 $\underset{\sim}{A}$ 、 $\underset{\sim}{B}$ 交集運算與聯集運算：

交集運算：表示喜歡運動也喜歡閱讀

$$\begin{aligned}\underset{\sim}{C} &= \underset{\sim}{A} \cap \underset{\sim}{B} \\ &= \frac{\min\{0.2,1.0\}}{x_1} + \frac{\min\{0.4,0.7\}}{x_2} \\ &+ \frac{\min\{0.8,0.6\}}{x_3} + \frac{\min\{1.0,0.4\}}{x_4} + \frac{\min\{0.3,0.1\}}{x_5} \\ &= \frac{0.2}{x_1} + \frac{0.4}{x_2} + \frac{0.6}{x_3} + \frac{0.4}{x_4} + \frac{0.1}{x_5}\end{aligned}$$

由Fuzzy集合 $\underset{\sim}{C}$ 可得知元素x_3為比較貼近此要求的同學

聯集運算：表示喜歡運動或喜歡閱讀

$$\begin{aligned}\underset{\sim}{D} &= \underset{\sim}{A} \cup \underset{\sim}{B} \\ &= \frac{\max\{0.2,1.0\}}{x_1} + \frac{\max\{0.4,0.7\}}{x_2} \\ &+ \frac{\max\{0.8,0.6\}}{x_3} + \frac{\max\{1.0,0.4\}}{x_4} + \frac{\max\{0.3,0.1\}}{x_5} \\ &= \frac{1.0}{x_1} + \frac{0.7}{x_2} + \frac{0.8}{x_3} + \frac{1.0}{x_4} + \frac{0.3}{x_5}\end{aligned}$$

由Fuzzy集合 $\underset{\sim}{D}$ 可得知元素x_1與x_4為比較貼近此要求的同學。

2.4 明確關係與Fuzzy關係

2.4.1明確關係和笛卡兒乘積

何謂「笛卡兒乘積」呢？首先，假設兩個集合為U和V且分別代表著某種概念，當要同時考量這兩種概念時，就得將在U中的每個元素 u 和V中的每個元素 v 做某種關係的組合，然後形成（u,v）元素對，如此可表示出所有的情況，此即為「笛卡兒乘積」，亦可稱為「直積」(direct product)。

定義　笛卡兒乘積（Descartes product）

設U、V兩個集合，此兩個集合的笛卡兒乘積為元素的組合，一般以符號 $U \times V$ 來表示，以數學表示式如下：

$U \times V = \{(u,v) \mid u \in U, v \in V\}$……（24）

若U、V兩個集合是不相等的話，$U \times V$ 和 $V \times U$ 兩直積所代表的意義是不同的，也就是說，元素對會有序對（ordered pair）的情形，元素對中前者為 U 的成員（member）；後者為 V 的成員。為了能較明瞭，例子如下：

例：

若有X、Y、Z三位同學的學科成績，會有哪幾種情形？

假設 U 表示同學的集合， V 表示學科的集合，且 U 和 V 分別表示如下

$U = \{X, Y, Z\}$和 $V = \{$數學, 英文, 國文$\}$

現在使用笛卡兒乘積來探討同學與學科的關係，這將會把所有的情形列出。

$U \times V =$｛（X , 數學） , （X ,英文）, （X ,國文）

（Y, 數學） , （Y ,英文）, （Y ,國文）

（Z , 數學） , （Z ,英文）, （Z ,國文）｝

由 $U \times V$ 的結果知曉有九種序對組合情形，也可說是直積集合有九個成員。

所以我們知道 U × V 元素之間是無受限的組合，若施加了某種限制，將可以把 U 和 V 之間的某種關係表現而出，因此，也可將 U 和 V 元素間的某種關係視為笛卡兒乘積U × V的子集合。

例：

在一份學生身高體重資料中，欲找出較符合「標準體重」的同學，假設標準體重為身高 － 體重＝100，並以 U 表示身高集合； V 表示體重集合。

U = { 150,160,170,180,190 }　　V = {50,60,70,80,90}

將會出現的有序對組合為：

$U \times V =\{$ （150 ,50）, （150 ,60）, （150 ,70）, （150 ,80）, （150 ,90）
, （160 ,50）, （160 ,60）, （160 ,70）, （160 ,80）, （160 ,90）
, （170 ,50）, （170 ,60）, （170 ,70）, （170 ,80）, （170 ,90）
, （180 ,50）, （180 ,60）, （180 ,70）, （180 ,80）, （180 ,90）
, （190 ,50）, （190 ,60）, （190 ,70）, （190 ,80）, （190 ,90）}

現在要找出較符合「標準體重」的同學，必須依據「身高 － 體重＝100」的公式，由以上的有序對可挑出符合「標準體重」關係的受限制有序對，將其以數學式表示於下，其中的符號$\Re$ 用來表示二元關係，由數學式可知道$\Re$ 為 U × V 的笛卡兒乘積U × V的子集合。

$\Re (U ,V)$ = { （150 ,50）, （160 ,60）, （170 ,70）, （180 ,80）, （190 ,90）}

以上所說的皆是兩集合間的二元關係（binary relation），倘若所探討的集合個數超過兩個以上，則稱之為多元關係（n-ary relation）。要表示關係我們一般以$\Re$ 來表示二元關係，拿 U 與 V 兩集合中的 u 、 v 元素來說，其會有兩種關係：

1. u 和 v 有關係，以$(u,v)\in\Re$ 表示。
2. u 和 v 無關係，以$(u,v)\notin\Re$ 表示。

以上兩個情形一定只有一個且不同時成立，所以成立與否的劃分相當明確。我們之前曾經使用特徵函數來敘述了明確集合，現在也得再次使用特徵函數來敘述二元關係，同樣地，含有二元關係的元素，其特徵值為為1；反之，未含有二元關係的元素，其特徵值為0，現將定義表示於下：

定義　二元關係（binary relation）

兩集合 U 與 V的直積 $U \times V$ 的子集合$\Re$ ，稱之 U 至 V 的二元關係。對於各個直積元素$(u,v) \in U \times V$ ，一般以下式的特徵函數來表示此關係是否存在。

$$\chi_{\Re}(u,v) = \begin{cases} 1 \ , \ \forall (u,v) \in \Re \\ 0 \ , \ \forall (u,v) \notin \Re \end{cases} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (25)$$

線性代數中的矩陣也可用來表示有限集合的二元關係的工具，使用矩陣來運算元素間的關係將十分方便，但論域為無限的話將無法使用矩陣運算來表示。

定義　二元關係矩陣（binary relation matrix）

假設有 U 和 V 兩個集合，$U = \{u_1, u_2, u_3, \ldots, u_n\}$與 $V = \{v_1, v_2, v_3, \ldots, v_n\}$ ，$\Re$ 為U、V間的二元關係，且$\Re$ 對應至個矩陣 $M_{\Re}$ 。

$$M_{\Re} = (\chi_{r_{ij}})_{m \times n} \ , \ i = 1,2,\ldots,m \ ; \ j = 1,2,\ldots,n \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (26)$$

上式中的$M_{\Re}$為關係矩陣、$\chi_{r_{ij}}$為矩陣的元素，更明確的說為集合 U 中第i個元素u_i與集合V中第j個元素v_j，組合成有序數對（u_i, v_j），而且$\chi_{r_{ij}}$還有兩情形發生。

$$\chi_{r_{ij}} = \begin{cases} 1 \ , \ if \ \ (u_i, v_j) \in \Re \\ 0 \ , \ if \ \ (u_i, v_j) \notin \Re \end{cases} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (27)$$

例：

將之前的標準體重例子改為以關係矩陣來表示

$$\begin{array}{c} \\ 150 \\ 160 \\ 170 \\ 180 \\ 190 \end{array} \begin{array}{c} \begin{array}{ccccc} 50 & 60 & 70 & 80 & 90 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

由此矩陣可知符合「標準體重」的元素對以數字1表示，反之，若不符合「標準體重」的以0表示。

2.4.2 Fuzzy 關係（fuzzy relation）

在現實中明確關係是不太適用的，它只能表示事物兩種狀況，這世界是存在著不確定、模糊的情況。比方說親子之間的長相相似度，總很難說長的一樣或長的不一樣這麼的絕對，一般大多以相似的程度來表示，因此就得將明確關係的觀念擴展至Fuzzy關係。之前是以歸屬函數來敘述Fuzzy集合，現在也可以用在Fuzzy關係上。

定義　Fuzzy 二元關係（fuzzy binary relation）

U 至 V 的Fuzzy二元關係也就是直積 $U \times V$ 的子集合 $\tilde{\Re}$ ，其歸屬函數以符號 $\mu_{\tilde{\Re}}$ 表示，且 $\mu_{\tilde{\Re}}$ 的值域為0至1的區間值，歸屬函數的區間對應表示如下。

$$\mu_{\tilde{\Re}} : U \times V \rightarrow [0,1] \quad \text{.....（28）}$$

對於 $\tilde{\Re}$ 的表示如下所示，式中的 $\mu_{\tilde{\Re}}(u,v)$ 是 $\tilde{\Re}$ 的歸屬函數，也就是（u,v）屬於 $\tilde{\Re}$ 的程度，且（u,v）為笛卡兒乘積 $U \times V$ 的元素。若式中的 $\mu_{\tilde{\Re}}$ 只可為0或1時，其將轉變為一般的明確關係。

$$\tilde{\Re} = \{((u,v), \mu_{\tilde{\Re}}(u,v) \,|\, (u,v) \in U \times V)\} \quad \text{.....（29）}$$

除了以上式的有序對表示有限集合的Fuzzy關係以外，也可以使用Fuzzy關係矩陣（fuzzy relation matrix）來表示。

定義　Fuzzy關係矩陣（fuzzy relation matrix）

假設有 U 和 V 兩個集合，$U = \{ u_1,u_2,u_3,\ldots,u_n \}$與 $V = \{ v_1,v_2,v_3,\ldots,v_n \}$ ，$\underset{\sim}{\Re}$ 為U、V間的Fuzzy關係，即直積 $U \times V$ 的Fuzzy關係，而且 U 和 V 為有限集合，那麼Fuzzy關係可以下面的矩陣來表示。

$$\underset{\sim}{\Re} = [r_{ij}]_{m \times n} = \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mn} \end{pmatrix} \quad \text{(30)}$$

矩陣中的r_{ij}係代表 U 中第 i 個元素 u_i 與集合 V 中第 j 個元素v_j，矩陣中的有序數對（u_i, v_j）若合於$\underset{\sim}{\Re}$ 的程度那麼下式將會成立。

$$\mu_{\underset{\sim}{\Re}}(u_i, v_j) = r_{ij} \quad , \quad r_{ij} \in [0,1]$$

接著再以先前的「標準體重」例子來闡述明確關係和Fuzzy關係的差異。以前面所談的明確關係來表示「標準體重」，只會有兩種情況，百分之百符合標準體重計算公式者才算是標準體重，未百分之百符合者，不管與標準體重差百分之幾皆為不符合。由此看來，這種分類方式頗為嚴苛，並不合乎現實生活中的使用，一般人們對於胖瘦的判斷並不是絕對的，而是與標準體重的差距程度來判斷，以下將以Fuzzy關係的矩陣來表示「標準體重」，由標準體重公式算出的值越趨近100的元素對，就給予越趨近1的歸屬度，如此的表示方式就較貼近於人性需求。

例：

再次拿之前「標準體重」的例子以Fuzzy關係的矩陣來表示，由下面的矩陣可以清楚的知道對於標準體重的描述，會比以明確關係的表示方式來的人性化。

$$\begin{array}{c} \\ 150 \\ 160 \\ 170 \\ 180 \\ 190 \end{array} \begin{array}{c} \begin{array}{ccccc} 50 & 60 & 70 & 80 & 90 \end{array} \\ \begin{bmatrix} 1.0 & 0.9 & 0.8 & 0.7 & 0.6 \\ 0.9 & 1.0 & 0.9 & 0.8 & 0.7 \\ 0.8 & 0.9 & 1.0 & 0.9 & 0.8 \\ 0.7 & 0.8 & 0.9 & 1.0 & 0.9 \\ 0.6 & 0.7 & 0.8 & 0.9 & 1.0 \end{bmatrix} \end{array}$$

在之前是假設$\mu_{\tilde{R}}$為直積 $U \times V$ 對應至[0 , 1]，也就是說每一個有序元素對（u , v）都會對應到一個在0至1間的值（歸屬度），然而有些時候必須將Fuzzy關係列入考慮，為一論域中的Fuzzy集合對應至[0 , 1]閉區間。

定義　廣域 Fuzzy 二元關係

設兩論域 X、Y 與Fuzzy集合$\tilde{A}$ 、Fuzzy集合$\tilde{B}$

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \,|\, x \in X\} \quad \cdots\cdots(31)$$

$$\tilde{B} = \{(y, \mu_{\tilde{B}}(y)) \,|\, y \in Y\} \quad \cdots\cdots(32)$$

若下面的性質存在時，

$$\mu_{\tilde{R}}(x, y) \le \mu_{\tilde{A}}(x) \ , \ \forall (x, y) \in X \times Y\} \quad \cdots\cdots(33)$$

且 $\mu_{\tilde{R}}(x, y) \le \mu_{\tilde{B}}(y) \ , \ \forall (x, y) \in X \times Y\} \quad \cdots\cdots(34)$

則Fuzzy集合$\tilde{A}$、$\tilde{B}$的Fuzzy關係為$\tilde{\Re} = \{[(x, y), \mu_{\tilde{\Re}}(x, y)] \,|\, (x, y) \in X \times Y)\}$

$\tilde{\Re}$ 中每一元素的歸屬度為Fuzzy集合$\tilde{A}$ 、$\tilde{B}$相對元素的最小值。

$$\mu_{\tilde{\Re}}(x, y) = \min\{\mu_{\tilde{A}}(x) \ , \ \mu_{\tilde{B}}(y)\} \quad \cdots\cdots(35)$$

例：

論域$X = \{x_1, x_2, x_3\}$及其Fuzzy集合$\tilde{A}$還有論域$Y = \{y_1, y_2\}$及其Fuzzy集合$\tilde{B}$，表示於下

$$\tilde{A} = \frac{1.0}{x_1} + \frac{0.6}{x_2} + \frac{0.2}{x_3} \quad , \quad \tilde{B} = \frac{0.4}{y_1} + \frac{0.9}{y_2}$$

接下來使用關係矩陣運算來求Fuzzy集合$\tilde{A}$與Fuzzy集合$\tilde{B}$的關係$\tilde{\Re}$

$$\tilde{\mathfrak{R}} = \tilde{A} \times \tilde{B} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \wedge \begin{bmatrix} y_1 & y_2 \end{bmatrix}$$

$$= \begin{bmatrix} 1.0 \\ 0.6 \\ 0.2 \end{bmatrix} \wedge \begin{bmatrix} 0.4 & 0.9 \end{bmatrix}$$

$$= \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array} \begin{array}{c} \begin{array}{cc} y_1 & y_2 \end{array} \\ \begin{bmatrix} 0.4 & 0.9 \\ 0.4 & 0.6 \\ 0.2 & 0.2 \end{bmatrix} \end{array}$$

此節的Fuzzy關係與數學上的兩數關係相當的類似，可說是兩數關係的演化，以兩項Fuzzy關係來說，是以「大略相等」、「似乎小於」等模糊的表示方式來描述兩者間的關係，顯示某種程度的關係。因為Fuzzy集合$\tilde{A}$與Fuzzy集合$\tilde{B}$的兩Fuzzy關係，意思為直積$\tilde{A} \times \tilde{B}$上所定義的Fuzzy集合，也就是說Fuzzy關係的運算是從Fuzzy集合的演算產生。

第三章　網際資訊加解密模式分析

現今各企業透過網際網路為通道發展電子化企業，已成為全球共通趨勢，對於安全性的考量，本論文所提出的安全機制目的為探討fuzzy在資訊安全上的應用，並且從應用觀點，撰寫不同模式之加密、解密程式，比較其效能。本文所提出之方法包括: 凱薩移位演算法、仿射演算法、維吉尼亞演算法、置換演算法。

3.1　系統架構

3.1.1 凱薩移位演算法分析

凱薩移位碼（Caesar Shift Cipher）是種簡易的古典密碼系統，其為將原明文字母做他種排列的方法，就是將明文(Plain)做某 bit 數的位移動作後即成為密文(Cipher)，例如將明文起始為 d、e、f…的順序轉換為密文 a、b、c…的起始順序。以凱薩的名言 anything one man can imagine, other men can make real.為例，加密之後的密文為 dqbwefkd lkb jxk zxk fjxdfkb, lqebo jxk jxhb obxi.表示。若要將其解密，只要逆推，即將密文字母順序往前移 3 個 bit，就為明文順序，例如將密文 a、b、c…的起始順序換轉為 d、e、f…的起始順序即為明文。

凱薩移位碼函數表示：

設 p：明文　c：密文　k：金鑰（移位的 bit 數）

加密函數 $E(p) \equiv p + k$ ……（1）

解密函數 $D(c) \equiv c - k$ ……（2）

為了容易瞭解，將字母與數字做一代碼表，如此也容易計算。凱薩移位碼並不一定只移 3 個 bit，也可往前或往後移 26 個以內的 bit 數，下面將用例子說明。

表 2 - 1　凱薩移位碼的字母與數字代碼表

字母	a	b	c	d	e	f	g	h	i	j	k	l	m
代碼	0	1	2	3	4	5	6	7	8	9	10	11	12
字母	n	o	p	q	r	s	t	u	v	w	x	y	z
代碼	12	14	15	16	17	18	19	20	21	22	23	24	25

例：

Bob 想將明文「we are gonna go to the theater tomorrow night.」加密為密文傳送給 Alice。

1. 產生金鑰：

   Alice 和 Bob 共同商量出一個編碼方式，假設為明文字母往後移 3 個 Bit，設密鑰 K=3，解密鑰也一樣。

2. 加密：

   Bob 將明文「we are gonna go to the theater tomorrow night.」轉變為代碼。$E(p) \equiv p + k = m + 3$

   （22,4,0,17,4,6,14,19,14,19,7,4,19,7,4,0,19,4,17,19,14,12,14,17,17,14,22）

   密文即為 te xob dlkkx dl ql qeb qebxqbo qljloolt kfdeq，送出密文。

3. 解密：

   Alice 收到密文 te xob dlkkx dl ql qeb qebxqbo qljloolt kfdeq

   轉成

   （22,4,0,17,4,6,14,19,14,19,7,4,19,7,4,0,19,4,17,19,14,12,14,17,17,14,22）

   經過解密函數 $D(c) \equiv c - k = c - 3$。得 we are gonna go to the theater tomorrow night.即為明文。

3.1.2 仿射演算法分析

仿射密碼系統(Affine)與凱薩密碼系統相當類似，也同樣為一替代法轉換。以26個英文字母為例，同樣使用凱薩密碼系統的字母代碼表，a = 0、b = 1、…、z = 25，其

加密函數　$E(p) \equiv \alpha x+\beta \pmod{26}$　……（3）

解密函數　$D(c) \equiv \alpha^{-1}(c-\beta) \pmod{26}$　……（4）

函數中的α、β皆為整數，α與26必須互質。其中α與26互質的原因如下：

假設 $\gcd(\alpha,26) = g > 1$，則產生E( )不為一對一的型式，即不同的x值(x_1、x_2)將會對應至相同的值；就 $\gcd(\alpha,26) = g > 1$ 論之，g值或許有兩種值 $g = 2$、$g = 13$，當 $g = 2$ 時，且 $x_1 \equiv x_2 \pmod{13}$，將發生不為一對一的型式；若 $g = 13$ 時，且 $x_1 \equiv x_2 \pmod{2}$，將發生不為一對一的型式。

以數學式表示如下：

$\alpha x_1+\beta \equiv \alpha x_2+\beta \pmod{26}$　……（5）

若加密函數不為一對一的型式，這將造成解密為明文時將會有錯誤結果，所以應避免此狀況發生。就一集合 $Z/26 = \{0, 1, 2, \ldots, 25\}$來看，此集合可做許多模運算，如模加法、模減法、模乘法、模除法，若做模除法運算上，將會有某幾個元素無法執行運算，這就得看 $x \in Z/26$，怎樣判斷x有無「乘法反元素」？即為有無 $y \in Z/26$ 存在，使得 $xy \equiv 1 \pmod{26}$，接著是怎樣得到y？要知道x是否有「乘法反元素」，只要看x與模數n是否互質，而接著使用輾轉相除法求解y，此為相當普遍的求解方式。

為了方便計算，對集合 $Z/26 = \{0, 1, 2, \ldots, 25\}$中之元素產生其「乘法反元素」並製成一表格，若 $\gcd(\alpha,26) = g > 1$ 時，則元素不存在「乘法反元素」將不置於表中：

表 2 - 2 Z/ 26 乘法反元素對照表

x	1	3	5	7	9	11	15	17	19	21	23	25
x^{-1}	1	9	21	15	3	19	7	23	11	5	17	25

經由乘法反元素對照表搭配加密函數 $E(p) \equiv \alpha\chi+\beta \pmod{26}$，求解解密函數 $D(c) \equiv \alpha^{-1}(c-\beta) \pmod{26}$ 將十分容易。

例：

Bob 想將明文「I love you.」以仿射密碼系統加密後傳送給 Alice。

1. 產生金鑰：

   Alice 和 Bob 共同商量出一個編碼方式，假設金鑰為 k = (3,8)且 gcd (3,26)=1。

2. 加密：

   Bob 利用加密函數 $E(p) \equiv 3x+8 \pmod{26}$產生密文

   m = I love you
   = (8,11,14,21,4,24,14,20)

   E( ) = (6,15,24,19,20,2,24,14)
   = g pytu cyo

3. 解密：

   Alice 接收到密文 g pytu cyo，利用解密函數 $D(c) \equiv 3^{-1}(c-8) \pmod{26}$ 反求明文

   c = g pytu cyo
   = (6,15,24,19,20,2,24,14)

   D( ) = (8,11,14,21,4,24,14,20)
   = I love you

3.1.3 維吉尼爾演算法分析

維吉尼爾（Vigenere）密碼系統，其與前述的密碼系統稍異，為一種多套字

母替代法，兩系統最大的相異之處，為加密時一個系統為只對單一字母做加密；另一個系統為一次對數個字母加密，維吉尼爾的加密方法如下所示：

令文字區塊長度為 d，將訊息以代碼表示為

$p = (p1, p\,2,\ldots, pd) \in (Z/26)$ ……..…（6）

金鑰為

$k = (k1, k2,\ldots, kd) \in (Z/26)$ ……（7）

知加密函數表示為

$E(p) = p + k = (p1 + k1, p2 + k2,\ldots, pd + kd) \pmod{26}$ ……（8）

則解密函數為

$D(c) = c - k = (c1 - k1, c2 - k2,\ldots, cd - kd) \pmod{26}$ ……（9）

也從中得知加密函數與解密函數可互逆的

$E(D(c)) = c$ ……（10）

$D(E(p)) = p$……（11）

所以，由此看來，可得知維吉尼爾密碼系統也是一種移位密碼系統，與之前的凱薩密碼系統有異的是，一種為整個字母區塊位移；另一種為單個字母的移位。[5][6][7]

例：

Bob 想傳訊「I miss you」給 Alice 並協商使用維吉尼爾密碼系統

1. 產生金鑰：

$$K = (4,7,15,18,21,24,25,1)$$

2. 加密：

利用加密函數 $E(p) = p + k$

加密程序如下為：

表 2 - 3 例子中的加密程序

訊息 p	I	m	i	s	s	y	o	u
訊息代碼	8	12	8	18	18	24	14	20
金鑰 k	4	7	15	18	21	24	25	1
模數加法	12	19	23	10	13	22	13	21
密文 c	d	t	x	k	n	w	n	v

3. 解密：

Alice 接收到密文 d txkn wnv，並利用解密函數 D(c) = c – k。

解密程序如下：

表 2 - 4 例子中的解密程序

密文 c	d	t	x	k	n	w	n	v
訊息代碼	12	19	23	10	13	22	13	21
金鑰 k	4	7	15	18	21	24	25	1
模數減法	8	12	8	18	18	24	14	20
訊息 p	I	m	i	s	s	y	o	u

3.1.4 置換演算法分析

置換（Permutation cipher）是一種數學變換，每次置換時皆可用個整數序列來表示，例如：原來的明文為P=（1,2,3,4,5,6），做個加密的動作，將位置1和位置2對調、位置3和位置4對調，也將位置5和位置6對調，產生密文C=（2,1,4,3,6,5）如此即表示了一個置換。每個置換都會有個與之對應的逆置換，明文序列經過置換和其逆置換之後，將保持不變。有時置換與其逆置換可能方法上是相同的，例如，就像上述明文P的逆置換也是M=（2,1,4,3,6,5）的情況。

置換密碼系統的單門為一個只有發送方和接收方知道的祕密置換（加密）和

其逆置換（解密）機制。明文在加密時是用加密置換去對明文進行置換。例如，若明文P=「資訊安全」，則將P加密之後就得到密文C=「訊資全安」。將密文解密是以逆置換來對密文訊息進行置換。例如，密文為C=「訊資全安」，則解密之後可得到明文P=「資訊安全」。

置換密碼系統中主要特點，由明文和密文做一比較可知兩者所含的元素是相同的，只是元素位置不一樣而已。置換密碼系統機制滿簡易的，頗易被破解而且安全性不高，但是這置換模式已在許多近代密碼機制中或多或少地使用了。

下面例子利用縱橫換位技巧產生了置換密碼。首先將明文各字逐一以固定的字數橫向寫於紙上，然而產生密文則以垂直的方向逐一字元讀出；若要解密則是將密文與原本長度一樣地直式書寫於紙上，然後橫向地讀出解得明文。

例：

Bob欲將訊息I could spend my life in this sweet surrender,Alice.經由置換密碼系統傳送給Alice.，其中金鑰為由橫式書寫轉換至直式讀取時的優先順序。

1. 產生金鑰：

$$K = (7,6,4)$$ 註：K=(行數, 列數, 起始行)

2. 加密：

表 2 - 5 例子中的加密程序

金鑰	5	6	7	1	2	3	4
明文	i	c	o	u	l	d	s
	p	e	n	d	m	y	l
	i	f	e	i	n	t	h
	i	s	s	w	e	e	t
	s	u	r	r	e	n	d
	e	r	a	l	i	c	e
密文	udiwrl lmneei dytenc slhtde ipiise cefsur onesra						

3. 解密：

Alice 接收到密文 udiwrl lmneei dytenc slhtde ipiise cefsur onesra，並將其配合金鑰順序與固定寬度以直式逐一寫下，接著橫式逐一讀取而出。

解密程序如下表：

表 2 - 6 例子中的解密程序

金鑰	5	6	7	1	2	3	4
密文	i	c	o	u	l	d	s
	p	e	n	d	m	y	l
	i	f	e	i	n	t	h
	i	s	s	w	e	e	t
	s	u	r	r	e	n	d
	e	r	a	l	i	c	e
明文	I could spend my life in this sweet surrender,Alice.						

由此密文看來，有著與原來明文相當高的相似度，容易由密文字母的頻率特徵進而嘗試解出明文。所以，可再執行多次的置換加密動作，將可有效降低密文破解率。

再次將前例做兩次加密動作：

密文一：udiwrl lmneei dytenc slhtde ipiise cefsur onesra

表 2 - 7 例子中的二次加密程序

金鑰	5	6	7	1	2	3	4
密文一	u	d	i	w	r	l	l
	m	n	e	e	i	d	y
	t	e	n	c	s	l	h
	t	d	e	i	p	i	i
	s	e	c	e	f	s	u
	r	o	n	e	s	r	a
密文二	Weciee rispfs ldlisr lyhiua umttsr dnedeo ienecn						

由上表可知，經過第二次的加密之後的訊息，其排列方式已變為沒什麼規律性了，若有心人士要攻擊此置換密碼系統就相當困難多了。

3.2 符號定義

Y(x)：明文資料的差異量。

U：為所有明文資料的論域，包含了所有要討論的資料。

U_i：將論域U平均分割成多個等間距區間（U_i, i=1,2,3,…,n），並將其以 $U = \{u_1,u_2,u_3,\ldots,u_n\}$ 表示，其中的 $u_i(x)$，定義為Y(x)的Fuzzy集合。

F(x)：為Y(x)的Fuzzy形式

$\mu_{\tilde{A}}$：定義為Fuzzy集合 $\tilde{A}$ 歸屬函數（membership function），$\mu_{\tilde{A}}$ 為一封閉區間的值域，其介於0至1之間。

$\tilde{A}$ ：為論域U的Fuzzy集合（fuzzy set），其數學表示式如下

$$\tilde{A} = \frac{\mu_{\tilde{A}}(u_1)}{u_1} + \frac{\mu_{\tilde{A}}(u_2)}{u_2} + \ldots\ldots + \frac{\mu_{\tilde{A}}(u_n)}{u_n} = \sum_{i=1}^{n} \frac{\mu_{\tilde{A}}(u_i)}{u_i}, \quad \forall_i \in U \quad \ldots\ldots\ldots\ldots\ldots\ldots (12)$$

式中的 $\mu_A(u_i)$為u_i在Fuzzy集合 $\tilde{A}$ 中的歸屬度，其所對應的關係為一封閉區間的值域，介於0至1之間。

$\tilde{R}^W$ （x,x-1）：為Fuzzy集合的Fuzzy關係運算子（fuzzy operator），表示第(x)筆與第(x-1)筆Fuzzy集合之間的Fuzzy關係。

3.3 系統模型演算法則

此密碼系統的模型建立，主要是將密碼系統的明文資料運用Fuzzy理論來轉換，使之帶有Fuzzy集合、Fuzzy關係，再找出Fuzzy集合間的歸屬度，經過矩陣運算而產生密文。

其流程大概如下：

步驟1.

假設明文資料為I(x)、差異量為Y(x)，並將各個差異量求出，也就是兩兩相鄰的資料做一皆差分運算，即得Y(x)。

$Y(x) = I(x) - I(x-1)$ ……（13）

步驟2.

求得所有的差異量之後，挑出最大的差異量D_{max}和最小的差異量D_{min}。然後將討論區域U分隔成數個等距的區間$u_1,u_2,u_3,\ldots,u_n$，集合可表示為

$U = \{u_1,u_2,u_3,\ldots,u_n\}$。

步驟3.

把所有的差異量Y(x)置入Fuzzy理論中。設一Fuzzy集合 $\tilde{A}$ 在u_i的歸屬函數為 $\mu_{\tilde{A}}(u_i)$，Fuzzy集合 $\tilde{A}$ 的表示為

$$\tilde{A} = \frac{\mu_{\tilde{A}}(u_1)}{u_1} + \frac{\mu_{\tilde{A}}(u_2)}{u_2} + \ldots\ldots + \frac{\mu_{\tilde{A}}(u_n)}{u_n} = \sum_{i=1}^{n} \frac{\mu_{\tilde{A}}(u_i)}{u_i}, \quad \forall_i \in U \quad \text{……（14）}$$

或

$$\tilde{A} = \{\mu_{\tilde{A}}(u_1)\ ,\ \mu_{\tilde{A}}(u_2)\ ,\ \ldots\ldots\ ,\ \mu_{\tilde{A}}(u_n)\} \quad \text{……（15）}$$

步驟4.

把經過差分運算的差異量Y(x)，配合使用歸屬函數轉換為Fuzzy集合 $\tilde{A}$ ，即將差異量Y(x)對應至所坐落的u_i區間，每一u_i區間會映射到每一Fuzzy集合 $\tilde{A}$ 的歸屬函數，將所要取的一區塊歸屬函數組合成Fuzzy集合 $\tilde{F}(x)$ ，所以Fuzzy集合 $\tilde{F}(x)$ 的子集合為Fuzzy集合 $\tilde{A}$ ，得Fuzzy集合 $\tilde{F}(x)$ 後，將其表示如下

$$\tilde{F}(x) = \frac{\mu_{\tilde{A}}(u_1(x))}{u_1} + \frac{\mu_{\tilde{A}}(u_2(x))}{u_2} + \ldots\ldots + \frac{\mu_{\tilde{A}}(u_n(x))}{u_n} \quad \text{……（16）}$$

或

$$\tilde{F}(x) = \{\mu_{\tilde{A}}(u_1(x))\ ,\ \mu_{\tilde{A}}(u_2(x))\ ,\ \ldots\ldots\ ,\ \mu_{\tilde{A}}(u_n(x))\} \quad \text{……（17）}$$

步驟5.

將前數筆的明文資料（x-1）、（x-2）....拿來做計算Fuzzy集合的基準矩陣 $\tilde{S}^{W}(x)$；此時不再以形式（x-w）（x-w+1）...（x-2）筆的明文資料組合成矩陣，而隨機取數筆 $\tilde{A}$ 中的元素來組合成矩陣 $\tilde{Q}(x)$。接著，使用矩陣運算把基準矩陣 $\tilde{S}^{W}(x)$ 和運算矩陣 $\tilde{Q}(x)$ 形成Fuzzy集合的關係矩陣 $\tilde{R}^{W}(x)$。

$$\tilde{S}^{W}(x) = F(x-w) = [S_{w,1}\ S_{w,2}\ S_{w,3}\ \cdots\ S_{w,m}] \quad \cdots\cdots (18)$$

$$\tilde{Q}(x) \neq \begin{bmatrix} F(x-w) \\ F(x-w+1) \\ F(x-w+2) \\ \vdots \\ F(x-2) \end{bmatrix}$$

$$= \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1m} \\ S_{21} & S_{22} & \cdots & S_{2m} \\ S_{31} & S_{32} & \cdots & S_{3m} \\ \vdots & \vdots & \ddots & \vdots \\ S_{v-1,1} & S_{v-1,2} & \cdots & S_{v-1,m} \end{bmatrix} \quad \cdots\cdots (19)$$

$$\tilde{R}^{W}(x) = \tilde{Q}(x) \times \tilde{S}^{W}(x)$$

$$= \begin{bmatrix} S_{11} \times S_1 & S_{12} \times S_2 & \cdots & S_{1m} \times S_m \\ S_{21} \times S_1 & S_{22} \times S_2 & \cdots & S_{2m} \times S_m \\ S_{31} \times S_1 & S_{32} \times S_2 & \cdots & S_{3m} \times S_m \\ \vdots & \vdots & \ddots & \vdots \\ S_{v-1,1} \times S_1 & S_{v-1,2} \times S_2 & \cdots & S_{v-1,m} \times S_m \end{bmatrix}$$

$$= \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1m} \\ R_{21} & R_{22} & \cdots & R_{2m} \\ R_{31} & R_{32} & \cdots & R_{3m} \\ \vdots & \vdots & \ddots & \vdots \\ R_{v-1,1} & R_{v-1,2} & \cdots & R_{v-1,m} \end{bmatrix} \quad \cdots\cdots (20)$$

步驟6.

將關係矩陣 $\underset{\sim}{R}^{W}(x)$ 內的各個元素R_{11}、R_{12} …依某序列排列組合成密文。

最後將所有步驟化繁為簡的概述，把整個模糊式加解密的流程表示於下：

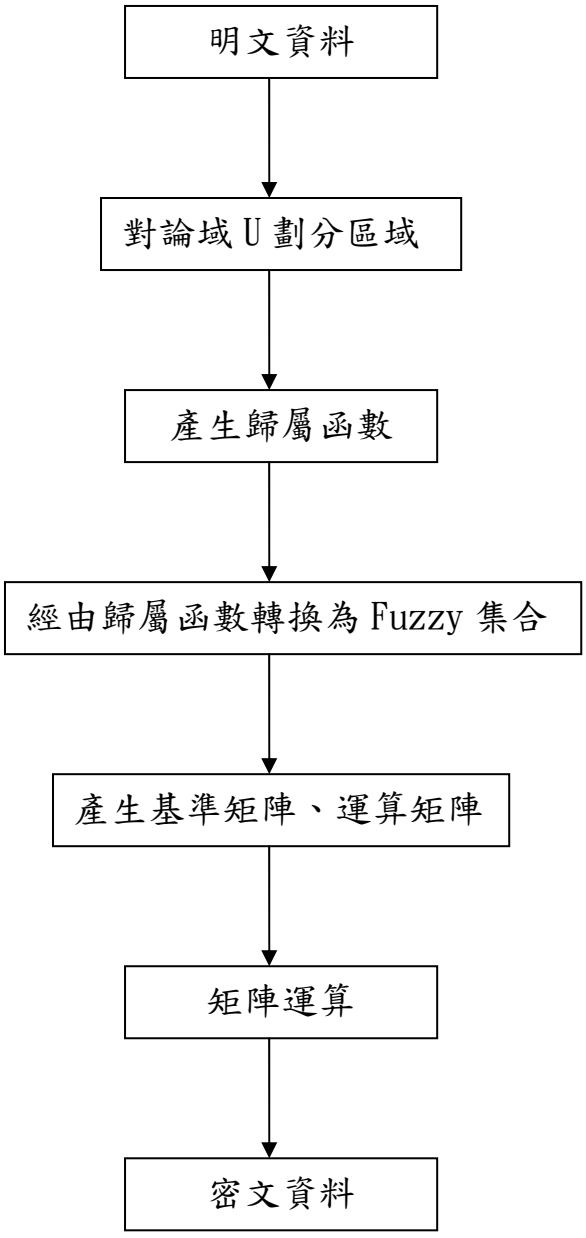

圖 3 - 1 密碼系統加密流程

前面所敘述的為明文經過密碼系統產生密文的過程，現要將經模糊式加密模式所得的密文解密回明文，只要把模糊式加密模式作逆運算即可得到明文，其流程大概如下：

步驟1.

將密文資料由頭至尾依序取出排列回原矩陣型式，即將密文各字元置入矩陣，從矩陣 $\underset{\sim}{R}^{W}(x)$ 的的元素位置R_{11}、R_{12}、R_{13}...依序置入，且矩陣 $\underset{\sim}{R}^{W}(x)$ 階數仍為原(w-1)行m列型式，此時所得矩陣即為原矩陣 $\underset{\sim}{R}^{W}(x)$ 。

$$\underset{\sim}{R}^{W}(x)=\begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1m} \\ R_{21} & R_{22} & \cdots & R_{2m} \\ R_{31} & R_{32} & \cdots & R_{3m} \\ \vdots & \vdots & \ddots & \vdots \\ R_{w-1,1} & R_{w-1,2} & \cdots & R_{w-1,m} \end{bmatrix}$$

步驟2.

將暫存在記憶體位址的資料取出組回原基準矩陣 $\underset{\sim}{Q}^{W}(x)$ 。

步驟3.

把取回的 $\underset{\sim}{R}^{W}(x)$ 與 $\underset{\sim}{Q}^{W}(x)$ 做矩陣運算來求得 $\underset{\sim}{S}^{W}(x)$ ，即

$$\underset{\sim}{S}^{W}(x)=\underset{\sim}{R}^{W}(x)\times(\underset{\sim}{Q}^{W}(x))^{-1}$$

$$=\begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1m} \\ R_{21} & R_{22} & \cdots & R_{2m} \\ R_{31} & R_{32} & \cdots & R_{3m} \\ \vdots & \vdots & \ddots & \vdots \\ R_{w-1,1} & R_{w-1,2} & \cdots & R_{w-1,m} \end{bmatrix}\begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1m} \\ S_{21} & S_{22} & \cdots & S_{2m} \\ S_{31} & S_{32} & \cdots & S_{3m} \\ \vdots & \vdots & \ddots & \vdots \\ S_{v-1,1} & S_{v-1,2} & \cdots & S_{v-1,m} \end{bmatrix}^{-1}$$

$$\underset{\sim}{S}^{W}(x)=[S_{w,1}\ S_{w,2}\ S_{w,3}\ \cdots\ S_{w,m}]$$

$$=F(x-w)$$

步驟4.

將運算矩陣 $\underset{\sim}{S}^{W}(x)$ 中的元素以列的型式取出比對Fuzzy集合 $\underset{\sim}{A}$ 的資料，可得知Fuzzy集合 $\underset{\sim}{A}$ 的各個項位。

$\underset{\sim}{A} = \{\mu_{\underset{\sim}{A}}(u_1)\ ,\ \mu_{\underset{\sim}{A}}(u_2)\ ,\ .....\ ,\ \mu_{\underset{\sim}{A}}(u_n)\}$

步驟5.

此時將所有比對出的Fuzzy集合 $\underset{\sim}{A}$ 的各個項位依序由小至大排列出Fuzzy集合 $\underset{\sim}{F}(x)$ 。

$\underset{\sim}{F}(x) = \{\mu_{\underset{\sim}{A}}(u_1(x))\ ,\ \mu_{\underset{\sim}{A}}(u_2(x))\ ,\ .....\ ,\ \mu_{\underset{\sim}{A}}(u_n(x))\}$

步驟6.

此步驟將Fuzzy集合 $\underset{\sim}{F}(x)$ 作歸屬函數逆轉換會得到Y(x)。即把F(x)中的各個歸屬函數映射至所對應的Ui，暫存區的論域U是被分隔成數個等距的區間 $u_1,u_2,u_3,\ldots,u_n$，每一區間皆有其所對應的變動量源Y(x)。

步驟7.

最後利用式子I(x) = Y(x) + I(x-1) 的方法將Y(x)中的每一筆資料運算回I(x)，此時排列的I(x)即為原明文資料。

由上面模糊式解密所說的解密步驟看來，的確只要將經模糊式加密所得到的密文作逆向處理即可解密得到明文，最後，將密碼系統的解密所有步驟化繁為簡的概述，把整個模糊式加解密的流程表示於下：

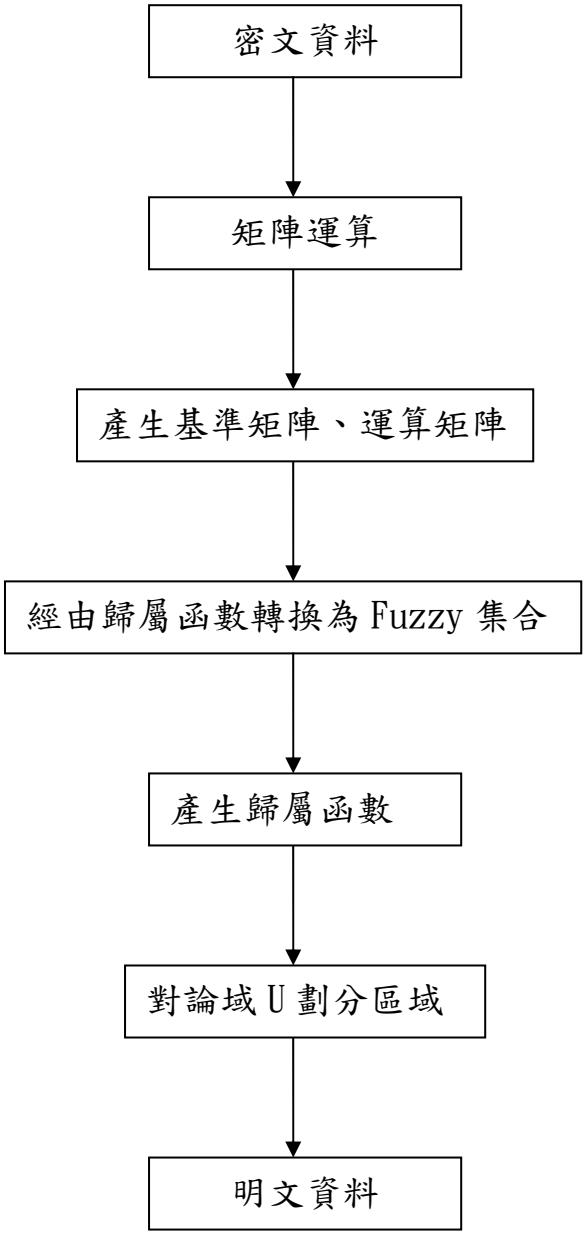

圖 3-2 密碼系統解密流程

3.4 密碼系統開發環境

本密碼系統的相關軟硬體為使用Intel Celeron 2.66 GHz中央處理器為主的個人電腦，配有1 GHz的動態記憶體，作業系統為Microsoft windows XP SD1。軟體開發工具部份為Microsoft Visual C++ Studio 6.0，因為密碼系統在做加密與解密時，會有相當大的計算量，所以採用高階程式語言Visual C++來製作。

表 3－1 系統規格表

電腦配備	規格
CPU	Intel Celeron 2.66GHz
RAM	512 GHz
HD	120GB / 7200rpm / 8MB
OS	XP SP1
Software	Microsoft Visual C++ Studio 6.0

第四章　實證分析

4.1　各演算法的執行

將各演算法執行流程結果表示於下列各小節，下圖為4種演算法的選單。第一點為凱薩移位演算法（Caesar Shift Cipher）；第二點為仿射演算法（Affine）；第三點為維吉尼爾演算法（Vigenere）；最後一個第四點為置換演算法（permutation）

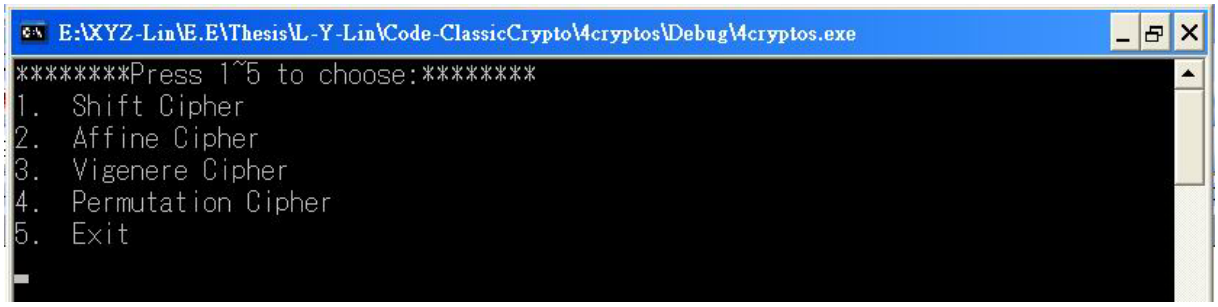

圖 4 - 1 選擇要執行的演算法

4.1.1 移位演算法流程

移位演算法的執行畫面：

1.由圖4-1選單選擇演算法，輸入1（凱薩移位演算法）

2.請輸入明文

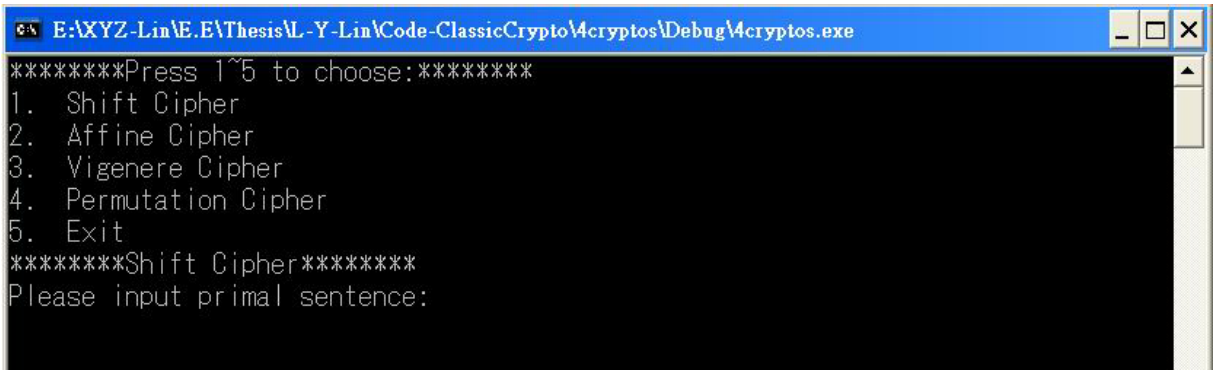

圖 4 - 2 移位演算法的執行畫面 a

3.由鍵盤輸入明文：todayisgoodday

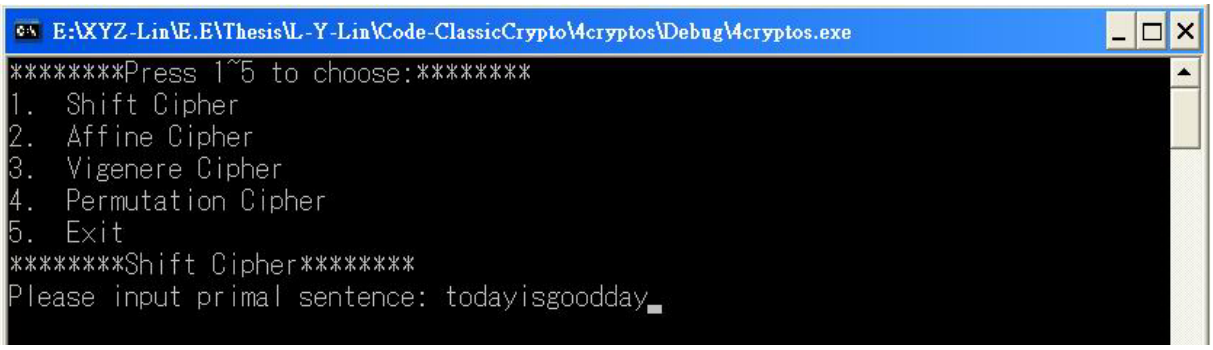

圖 4 - 3 移位演算法的執行畫面 b

4.請輸入金鑰：12

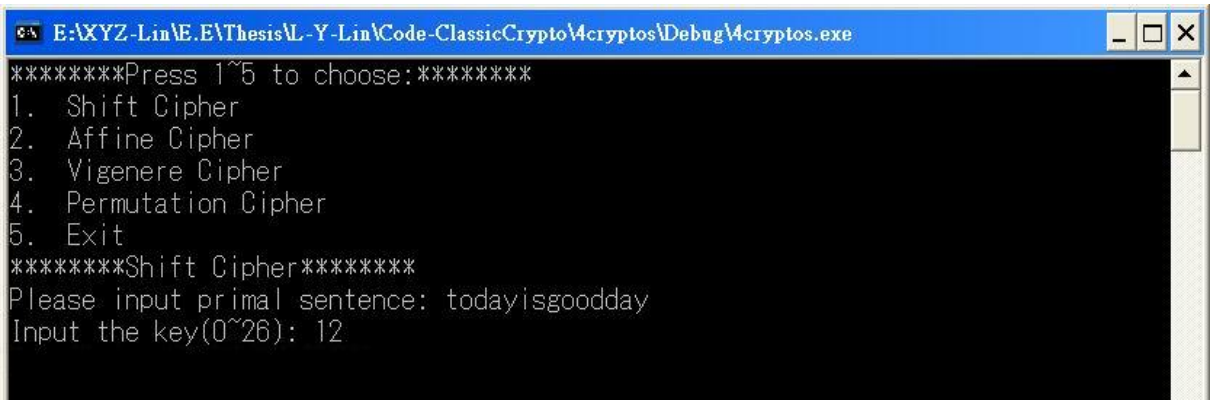

圖 4 - 4 移位演算法的執行畫面 c

5.產生密文：FAPMKUESAAPPMK，並把密文還原回來。

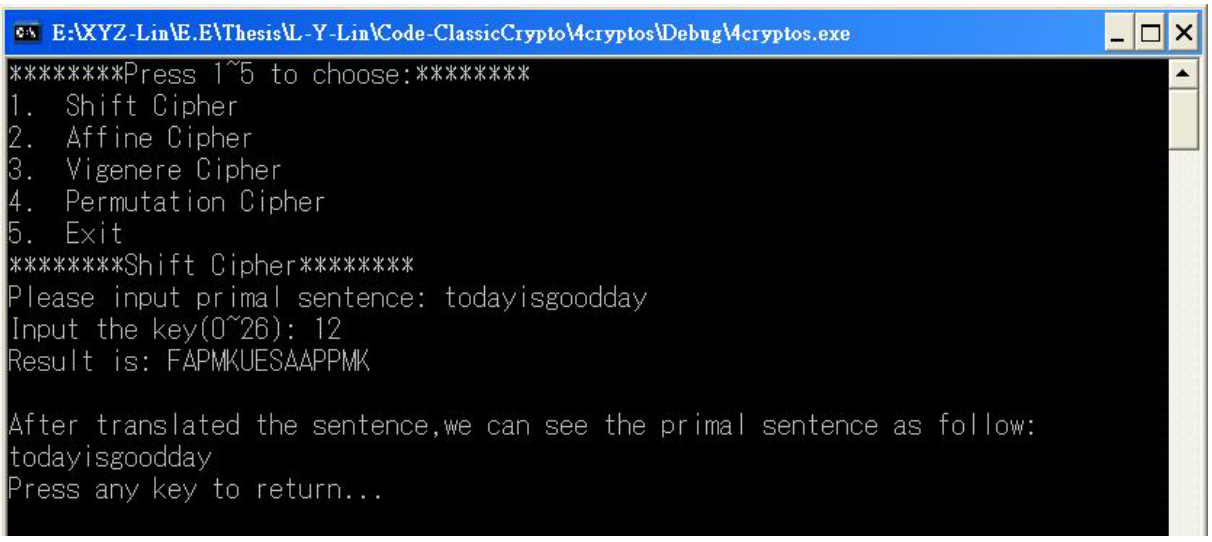

圖 4 - 5 移位演算法的執行畫面 d

移位演算法的程式運算過程：

1. 輸入明文。
2. 將明文放至記憶體空間內。
3. 求出明文長度。
4. 螢幕顯示請輸入金鑰。
5. 輸入兩數值當金鑰（α、β）。
6. 將 α 與 26 做最大公因數運算，若結果為 1，則繼續進行；若結果不為 1，則顯示錯誤，請再重新輸入。
7. 由記憶體逐一取出明文，並搭配金鑰做 $C = \alpha * P + \beta \pmod{26}$ 運算，運算結果即為密文，將運算完後的密文值依序回存至記憶體。
8. 由記憶體抓出密文顯示於螢幕上。
9. 由記憶體逐一取出密文，並搭配金鑰做 $P = (C - \beta) * \alpha^{-1} \pmod{26}$ 運算，運算結果即為明文。
10. 將運算完後的明文值依序顯示於螢幕上。

4.1.2 仿射演算法流程

仿射演算法的執行畫面：

1.由圖4-1選單選擇演算法，輸入1（凱薩移位演算法）

2.請輸入明文

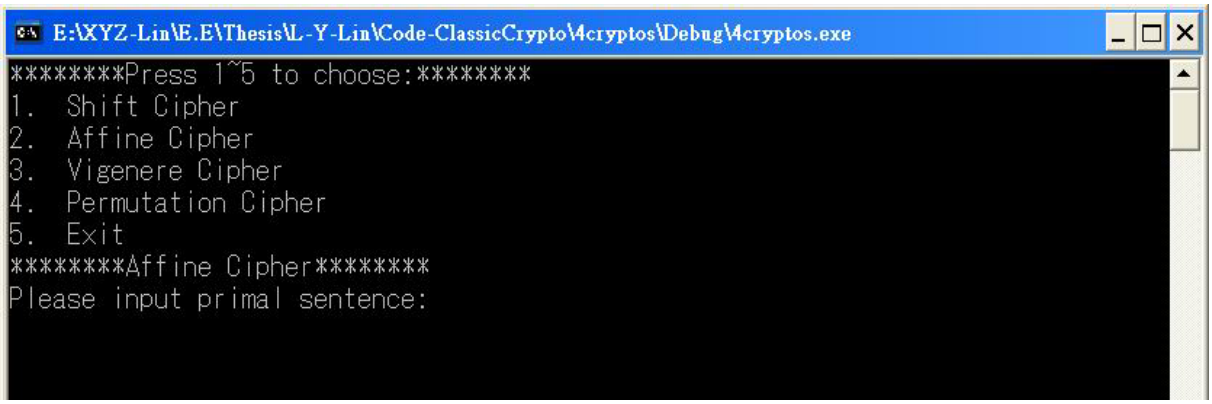

圖 4 - 6 仿射演算法的執行畫面 a

3.由鍵盤輸入明文：ilovethisgame

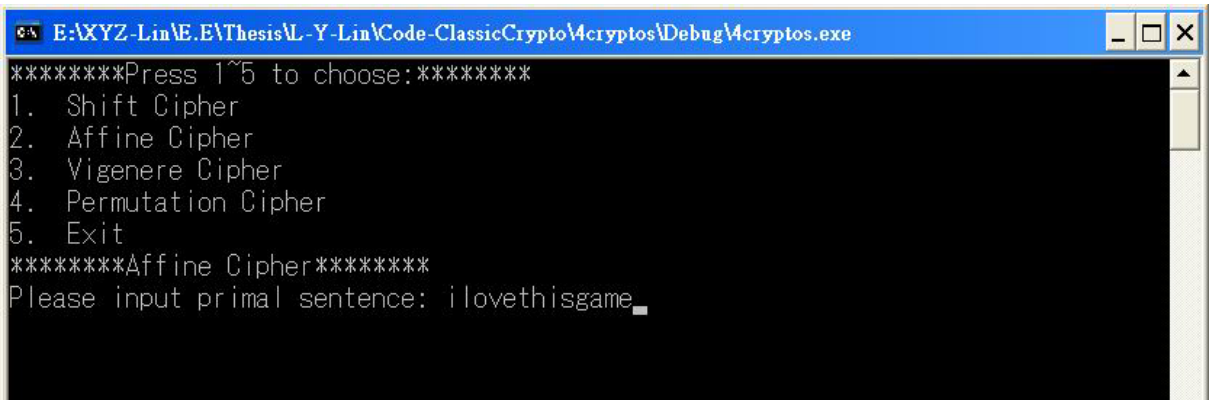

圖 4 - 7 仿射演算法的執行畫面 b

4.請輸入金鑰

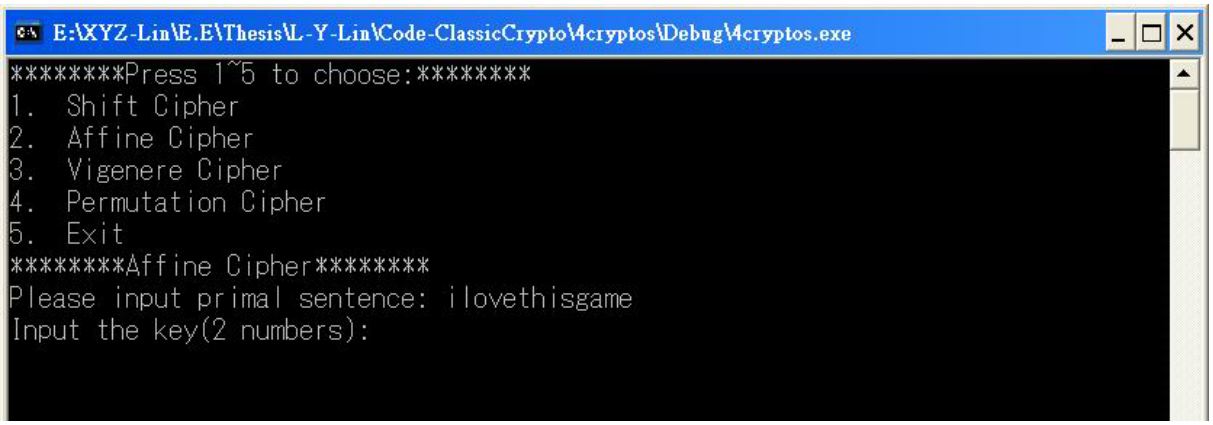

圖 4 - 8 仿射演算法的執行畫面 c

5.輸入金鑰：3 , 10

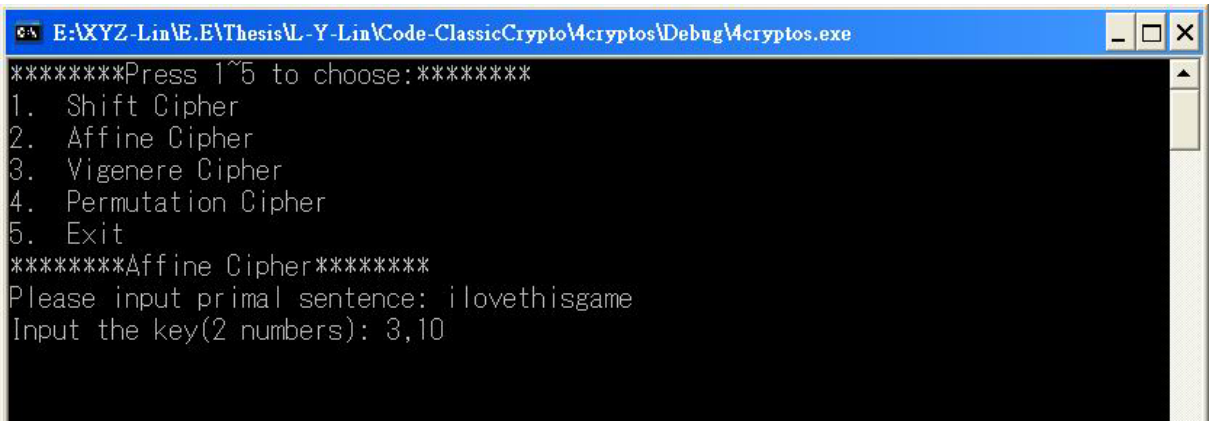

圖 4 - 9 仿射演算法的執行畫面 d

6.產生密文：YHOLMFVYCSAKM，並把密文還原回來。

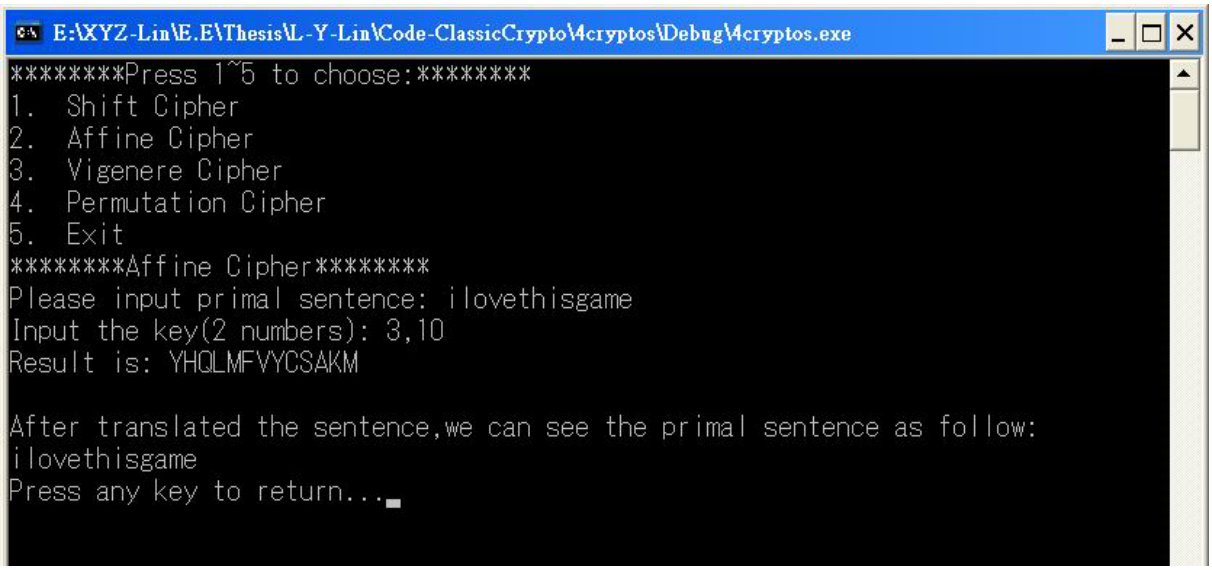

圖 4 - 10 仿射演算法的執行畫面 e

仿射的程式運算過程：

1. 輸入明文。
2. 將明文放至記憶體空間內。
3. 求出明文長度。
4. 螢幕顯示請輸入金鑰。
5. 輸入兩數值當金鑰（α、β）。
6. 將 α 與 26 做最大公因數運算，若結果為 1，則繼續進行；若結果不為 1，則顯示錯誤，請再重新輸入。
7. 由記憶體逐一取出明文，並搭配金鑰做 $C = \alpha * P + \beta \pmod{26}$ 運算，運算結果即為密文，將運算完後的密文值依序回存至記憶體。
8. 由記憶體抓出密文顯示於螢幕上。
9. 由記憶體逐一取出密文，並搭配金鑰做 $P = (C - \beta) * \alpha^{-1} \pmod{26}$ 運算，運算結果即為明文。
10. 將運算完後的明文值依序顯示於螢幕上。

4.1.3 維吉尼爾演算法流程

維吉尼爾演算法的執行畫面：

1.由圖4-1選單選擇演算法，輸入1（凱薩移位演算法）

2.請輸入明文

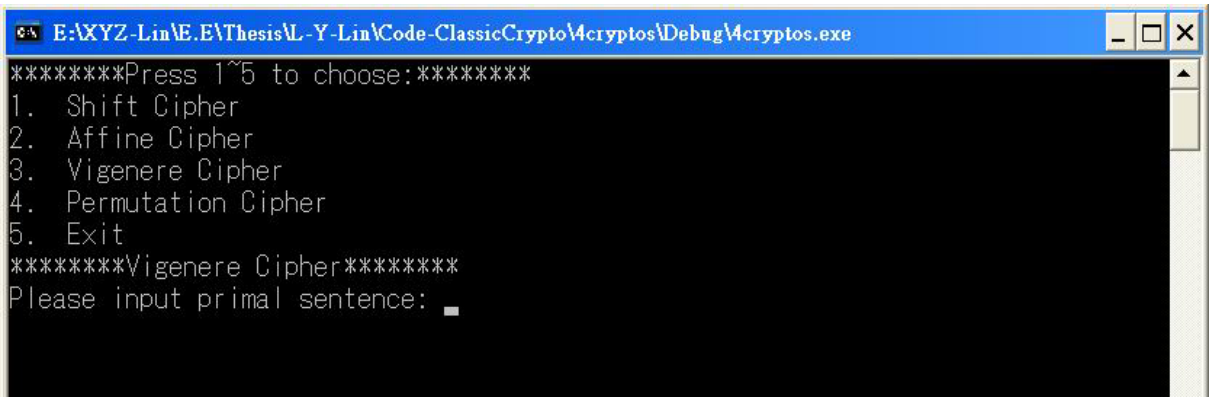

圖 4 - 11 維吉尼爾演算法的執行畫面 a

3.由鍵盤輸入明文：givemeahand

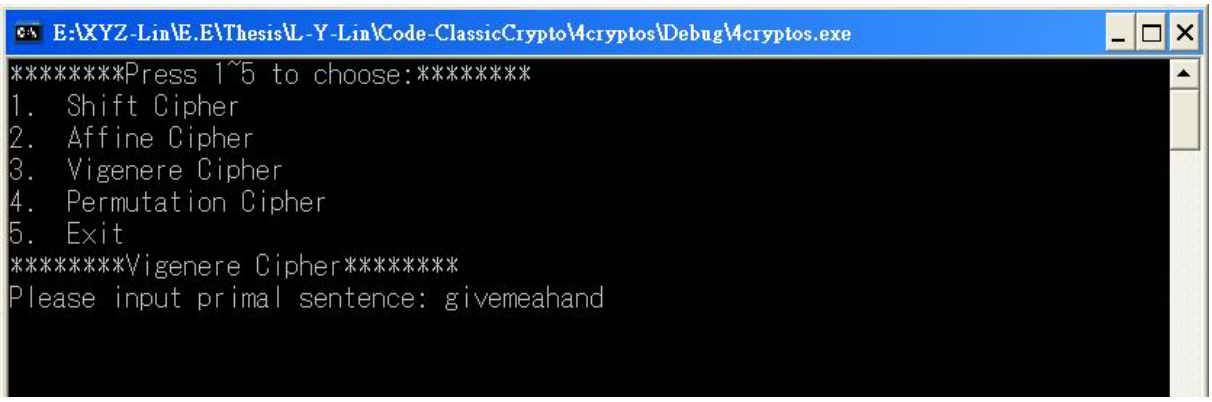

圖 4 - 12 維吉尼爾演算法的執行畫面 b

4.請輸入金鑰：11223344556

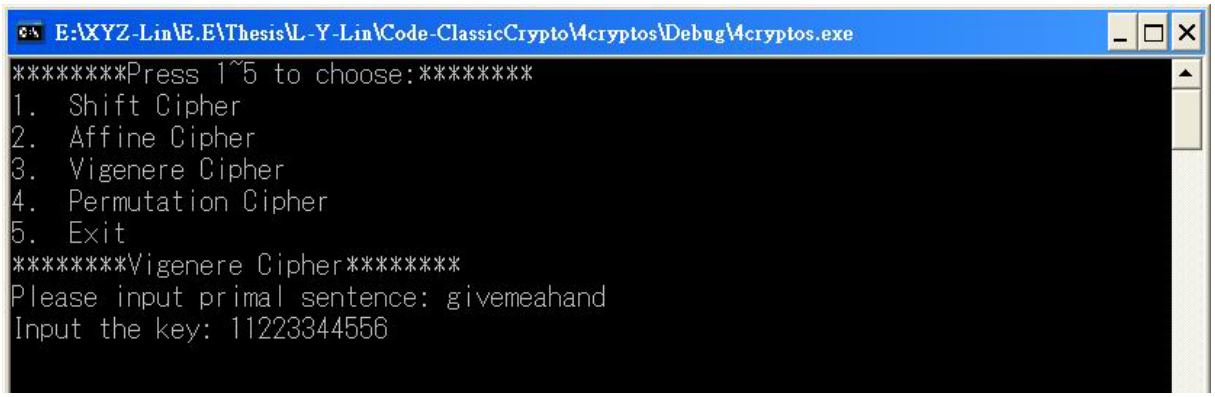

圖 4 - 13 維吉尼爾演算法的執行畫面 c

5.產生密文：79G6?74;5B9，並把密文還原回來。

圖 4 - 14 維吉尼爾演算法的執行畫面 d

維吉尼亞的程式運算過程：

1. 螢幕顯示請輸入明文。
2. 輸入明文。
3. 將明文放至記憶體空間內。
4. 求出明文長度。
5. 螢幕顯示請輸入金鑰。
6. 輸入金鑰。

7. 將金鑰放至記憶體空間內。
8. 計算金鑰長度。
9. 由兩記憶體逐一取出明文且逐一取出金鑰，並互相一對一搭配做 $C_i = P_i + K_i$（mod 26）運算，運算結果即為密文 C_i，將運算完後的密文 C_i 值依序回存至記憶體。
10. 由記憶體抓出密文顯示於螢幕上。
11. 由記憶體逐一取出密文，並搭配金鑰做 $P_i = C_i - K_i$（mod 26）運算，運算結果即為明文。
12. 將運算完後的明文值依序顯示於螢幕上。

4.1.4 置換演算法流程

置換演算法的執行畫面：

1.由圖4-1選單選擇演算法，輸入1（凱薩移位演算法）

2.請輸入明文

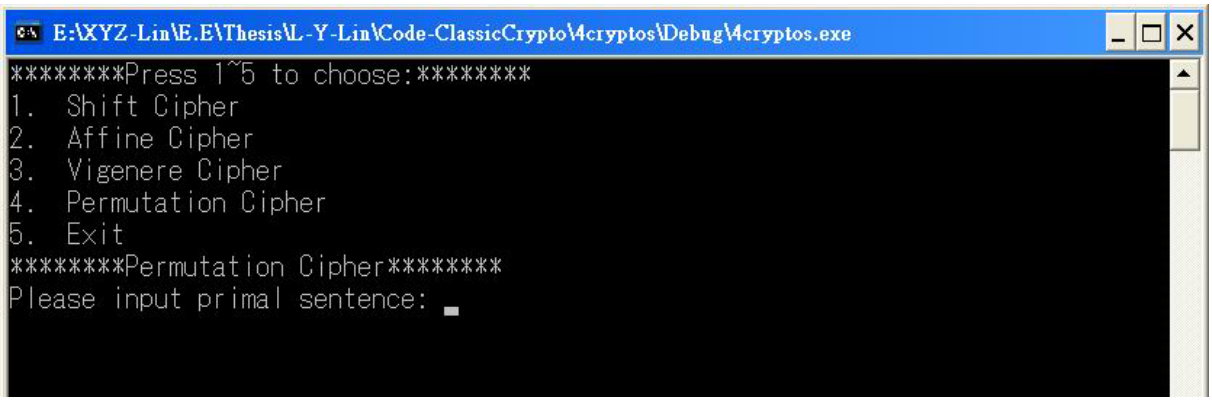

圖 4 - 15 置換演算法的執行畫面 a

3.由鍵盤輸入明文：lovepeace

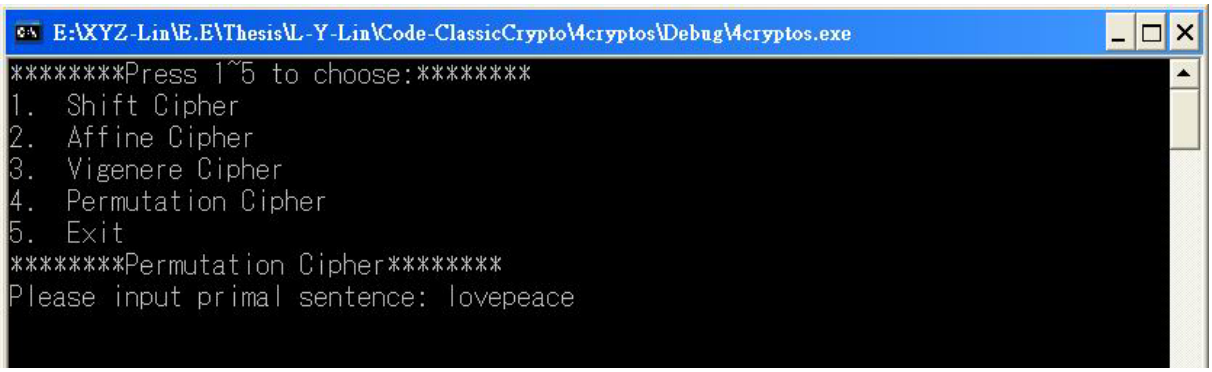

圖 4 - 16 置換演算法的執行畫面 b

4.請輸入金鑰：

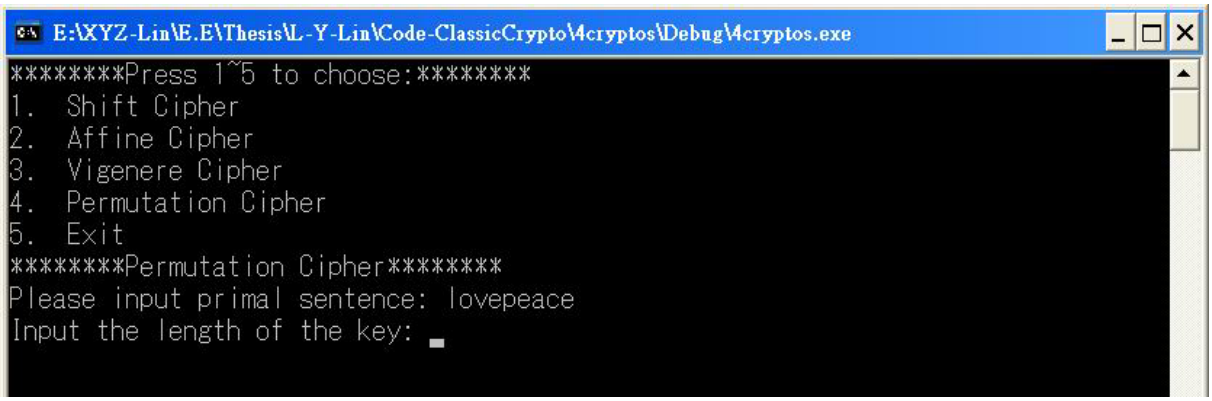

圖 4 - 17 置換演算法的執行畫面 c

5.輸入金鑰：4 bit長度（4 , 3 , 2 , 1）

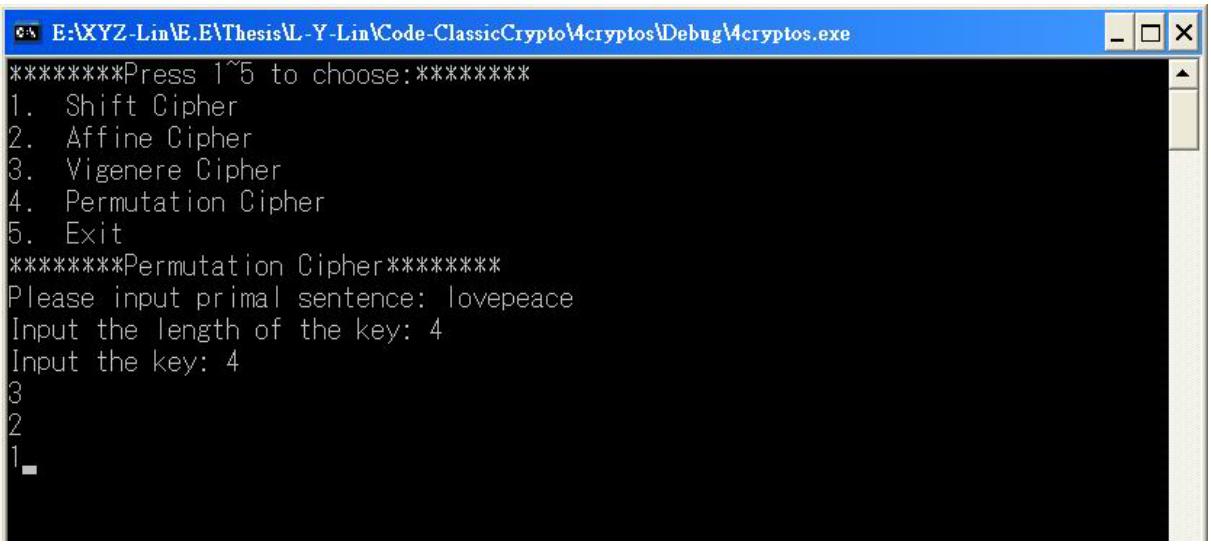

圖 4 - 18 置換演算法的執行畫面 d

6.產生密文：EVOLCAEP?，並把密文還原回來。

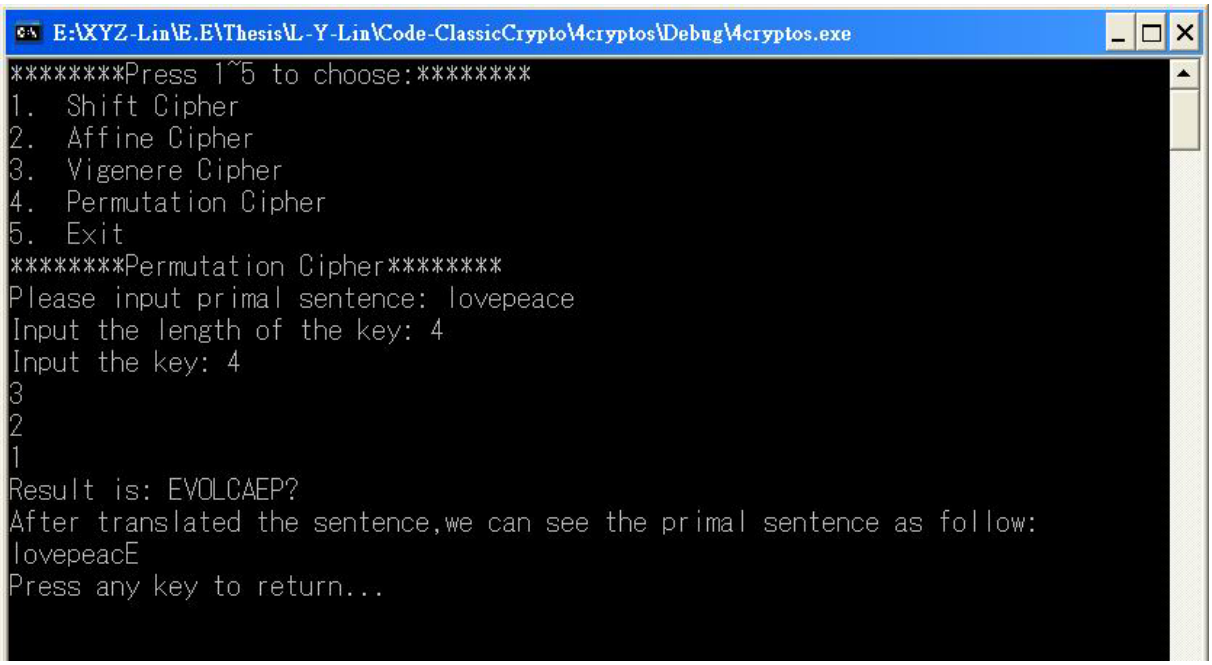

圖 4 - 19 置換演算法的執行畫面 e

置換的程式運算過程：

1. 螢幕顯示請輸入明文。
2. 由鍵盤輸入明文。
3. 將明文放至記憶體空間內。
4. 求出明文長度。
5. 螢幕顯示請輸入金鑰。
6. 輸入金鑰。
7. 將金鑰放至記憶體空間內。
8. 計算金鑰長度。
9. 由兩記憶體逐一取出明文且逐一取出金鑰，並互相一對一搭配做 $C_i = P_i + K_i$（mod 26）運算，運算結果即為密文 C_i，將運算完後的密文 C_i 值依序回存至記憶體。
10. 由記憶體抓出密文顯示於螢幕上。
11. 由記憶體逐一取出密文，並搭配金鑰做 $P_i = C_i - K_i$（mod 26）運算，運算結

果即為明文。

12. 將運算完後的明文值依序顯示於螢幕上。

4.1.5 本文演算法流程

本文演算法的執行畫面：

1.請選擇做加密解密還是退出。

圖 4 - 20 本文演算法的執行畫 a

2.選擇1後，執行加解密動作並顯示密文明文。

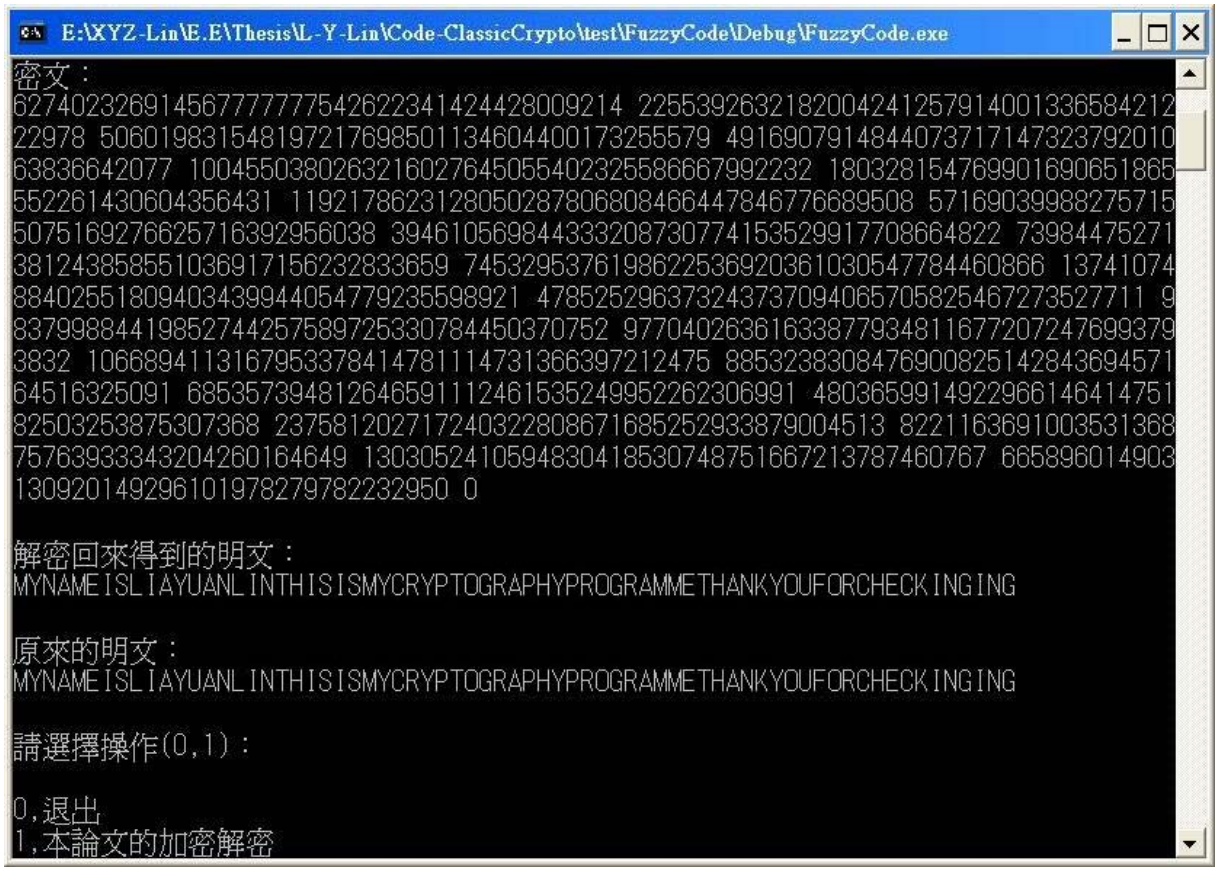

圖 4 - 21 本文演算法的執行畫 b

4.2 複雜度分析

演算法複雜度的分析間接影響了演算法的分類，大略分為兩種，屬於多項式時間者與非多項式時間者。一般來說，計算演算法的複雜度主要是用來衡量此法在一問題上的效率，即需多少個運算次數或需花費多少時間，若問題是相當困難的話，則解決問題上的計算量與時間將也相對的增加；反之，若問題是相當簡易的話，則解決問題上的計算量與時間將也將隨之減少。

在複雜度分析上，下面的各個資料輸入表格的形成，是分別以三種不同資料量輸入，經過其各自的加解密演算法之後，將其所花費的計算時間記錄下來。由各表看來，各演算法中做加密解密的動作時的反應時間，由表看來各演算法各有其複雜度。表4-1顯示出這幾個演算法花費時間相當的短，這是因為其演算法相當的簡易所造成；當增加資料量輸入時，表4-2顯示各演算法的花費時間比上個分析有些許的增加；當再增加資料量時，各演算法的花費時間也顯著的遞增。因此，就整個分析看來，本文的模糊式加解密模式比這些演算法有較高的複雜度。

表 4 - 1 256 bits 的資料輸入

演算法	Shift			Affine			Vigenere			Permutation		
筆數	1	2	3	1	2	3	1	2	3	1	2	3
時間（s）	.001	.001	.001	.001	.001	.001	.001	.001	.001	.001	0.001	.001

表 4 - 2 512 bits 的資料輸入

演算法	Shift			Affine			Vigenere			Permutation		
筆數	1	2	3	1	2	3	1	2	3	1	2	3
時間（s）	0.001	.001	0.001	0.011	0.010	0.014	.001	.001	.001	.001	.001	.001

表 4 - 3 1024 bits 的資料輸入

演算法	Shift			Affine			Vigenere			Permutation		
筆數	1	2	3	1	2	3	1	2	3	1	2	3
時間（s）	0.002	0.002	0.002	0.116	0.121	0.105	0.002	0.002	0.002	.001	.002	.001

表 4 - 4 本文演算法的花費時間

輸入資料量 (bits)	256			512			1024		
筆數	1	2	3	1	2	3	1	2	3
加密時間（s）	3.242	3.238	3.247	4.648	4.657	4.641	7.392	7.398	7.399
解密時間（s）	22.139	22.558	33.785	32.384	32.447	32.337	51.498	51.527	51.526

4.3 映射性分析

此分析乃是由使用者輸入明文資料，再經過電腦執行加密的程序後，對明文與密文作分析。在此分析中，所輸入的資料將被轉變為數位的型式輸出，對於所輸入的資料型式限制，一般為ASCII字元即可，為了易於分析起見，本分析以英文字母資料型式輸入。

以下所列出的實驗分析，為以a,b,c…z的英文字母做明文輸入字串，其由a開始至z，接著由aa開始至az以此類推，由電腦加密過後得到一長串數字即為密文。仔細觀察比對每一筆字串，可發現並沒有重複的字串出現，因此明文與密文的映射關係可說是固定的一對一關係，也就是說不同的明文會對應至不同的密文，且明文間不重複時，經過加密之後所得的數字字串也不會重複。

Plaintext:
a000000a
Ciphertext:
979611846331538221461844967313353302211552533816173311293166352
Plaintext:
a000000b
Ciphertext:
513144686134429205216405546644313692418742084994506530585411712 9
Plaintext:
a000000c
Ciphertext:
256855497451911816768730650528155221590143541543356628245420306 9
Plaintext:
a000000d
Ciphertext:
717021919795420061141636623153554145150759244631051026113106262 4
Plaintext:
a000000e
Ciphertext:
165539873225521201965655184544568663562568660745512444303258 3
Plaintext:
a000000f
Ciphertext:
216514121543848334721591703592886947085411371463742923252508321 0
Plaintext:
a000000g
Ciphertext:
932524864545537233648445442262661572032606276544261617425374 62
Plaintext:
a000000h
Ciphertext:
241129218736364572354515050865462454167858114136037462535812159 9
Plaintext:
a000000i

Ciphertext:
6863715461036016247114757744714936775646431124349826161048113 01
Plaintext:
a000000j
Ciphertext:
95607721246412621156326134940432222399317185362351466334615377 4
Plaintext:
a000000k
Ciphertext:
71328038331426216645222305861934876295683455203832815912813654 9
Plaintext:
a000000l
Ciphertext:
55615554924815154216818642523917935026230619103664183663563554 57
Plaintext:
a000000m
Ciphertext:
34139186221422418229140553681340364212951110222665536826143504 3
Plaintext:
a000000o
Ciphertext:
41598552568449005423644057254309911761048224051210576129356362 82
Plaintext:
a000000p
Ciphertext:
46122776244415106456591053533299357219516572220510458651118819 57
Plaintext:
a000000q
Ciphertext:
44724214357540151467411393307297807622125054246757204205644645 12
Plaintext:
a000000r
Ciphertext:
23351204166404124129614442350192032516160688302241460721335695 23

Plaintext:
a000000s
Ciphertext:
171561452921163246017183445483375834241718477442562028716107175 2
Plaintext:
a000000t
Ciphertext:
4665542514826575145257221184597457254191515762626747355763483 81
Plaintext:
a000000u
Ciphertext:
3336446685530178520601905496141531502158263601249385224630334292
Plaintext:
a000000v
Ciphertext:
1385419870700614275442616519592451196686882883847385631522581919
Plaintext:
a000000w
Ciphertext:
3274226532785583523262445421313125633684469104894376613322190126
Plaintext:
a000000x
Ciphertext:
9756768185132690613910553384250512109635773356625142071546966 30
Plaintext:
a000000y
Ciphertext:
2855857659423552322238970767672656696866162358066324611363840660
Plaintext:
a000000z
Ciphertext:
7082550646363670561127272896747514131656613312335571386751172930
Plaintext:
a00000aa

Ciphertext:

695611115061633334300451604425562465743295461399192147655726342 6

Plaintext:

a00000ab

Ciphertext:

13785135850464161646638438587166448212521352548268786727727348 21

Plaintext:

a00000ac

Ciphertext:

56766245642021525352255526145565641398330495026664355221210393 87

Plaintext:

a00000ad

Ciphertext:

35765259525234794611814555256408461652089522326869021115292536 53

Plaintext:

a00000ae

Ciphertext:

19543536194803722475712526164457703751796515033616384866284703 57

Plaintext:

a00000af

Ciphertext:

56657854363429275842138449243644536579362418131714654061227160 12

Plaintext:

a00000ag

Ciphertext:

59147873871464141215626838823287389161538894861500554686634612 82

Plaintext:

a00000ah

Ciphertext:

44267768152227511177366069521666652425636120311232955454545562 65

a00000ai

Ciphertext:

43660514461613555667330783165453964226219874189730196543023326 74

4.4 頻率分析

有名的密碼學專家大衛.卡恩（David Kahn）所寫的相關文章與著作皆曾提過 Deductive solutions are those based on frequency analysis, they are the general solutions for any cipher system，意思是說破解任何一般體制的密碼系統，各演算破解法都是以頻率分析為基礎。因此，本分析為了瞭解加解密模式的安全性，而對明文作頻率分析，當然也得對密文作頻率分析，以此來探討兩者間某些關連性的存在。此分析資料來源為三個15K以下的文字當明文，這些資料大小已相當足夠來作分析。於密碼理論中，加密的意義是轉變明文中的某些規則變化，使產生的密文不帶有原明文所含的特性或某些規則特徵。原本的明文是ASCII編碼，這種方式可看到帶有某種規則特性，這對於懂計算機概論的人來說都可發現此特性，於是一般會將之經由加密後可產生一大串毫無規則的碼，以使他人無法觀察出特性。因此使用頻率分析來測試這一大串毫無規則的碼是否有某些特徵出現。

下面的三個分析圖是分析結果。先由資料1的分析圖來看，即圖4-25所示，可知曉資料1的密文字串分佈非常的均勻，只有少數的3、4個字串較突出於平均範圍，這種加密結果可算是有相當優異的表現。

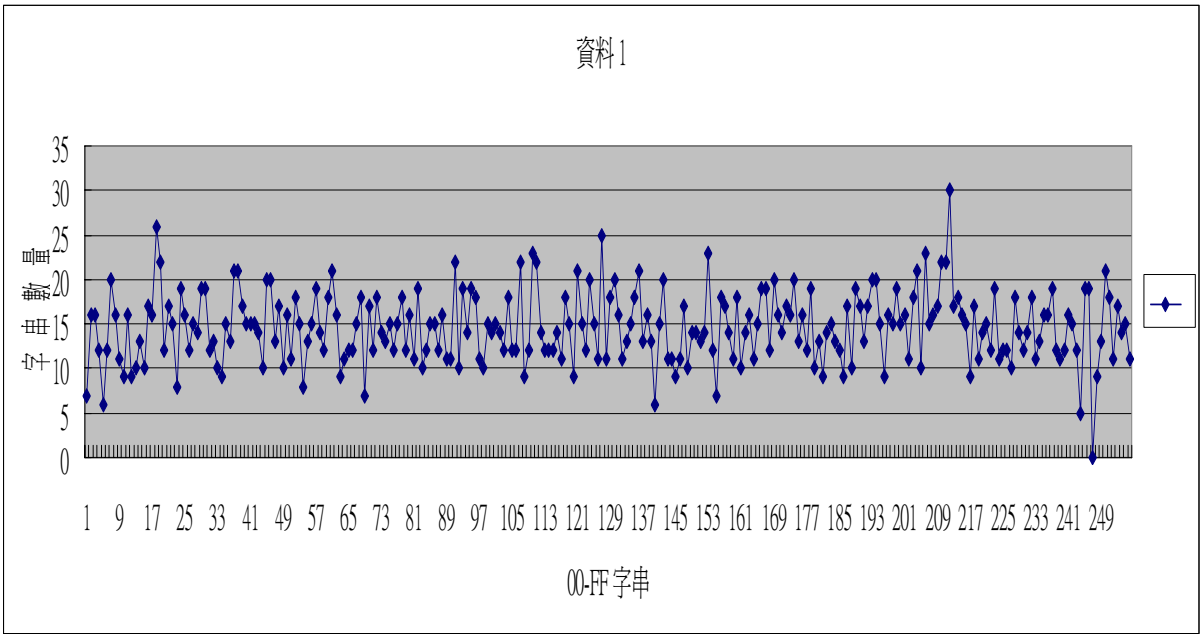

圖 4-25 資料 1 頻率分析圖

接著看圖4-26資料2的分析結果，這張資料2的分析圖明顯的比資料1的分析圖更佳，因為資料2的結果可說分佈的相當均勻，由圖看來，從密文字串00開始的分佈一直到106或107才出現特高值，然後密文字串分佈又回到一定的分佈範圍直到最後一個密文字串，整個資料密文字串範圍都固定在某一範圍，只有一個較其他字串相異，可說是相當不錯的加密表現。

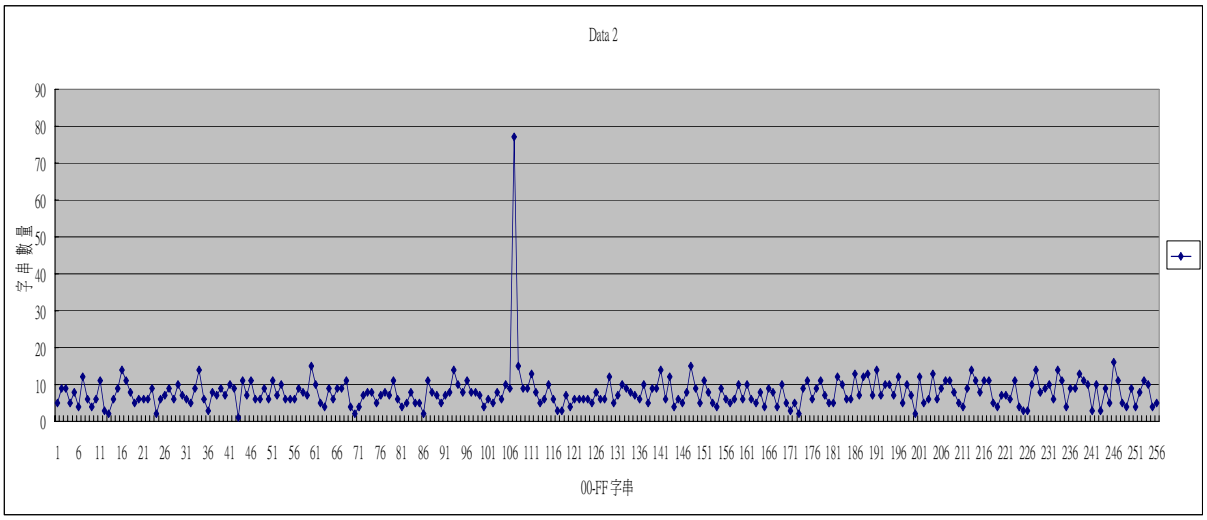

圖 4-26 資料 2 頻率分析圖

最後看圖4-27資料3的分析結果，這分析圖顯示出資料3的結果分佈的也相當均勻，各個字串都散落在某一區間，只是區間幅度較之前稍大一點點，差異性較大的字串約出現在19左右、115左右、151左右、217左右，總共也才也約4、5個左右，其整體的結果也算是相當不錯。

這圖比資料1的分析圖更佳，因為資料2的結果可說分佈的相當均勻，由圖看來，從密文字串00開始的分佈一直到106或107才出現特高值，然後密文字串分佈又回到一定的分佈範圍直到最後一個密文字串，整個資料密文字串範圍都固定在某一範圍，只有一個較其他字串相異，可說是相當不錯的加密表現。

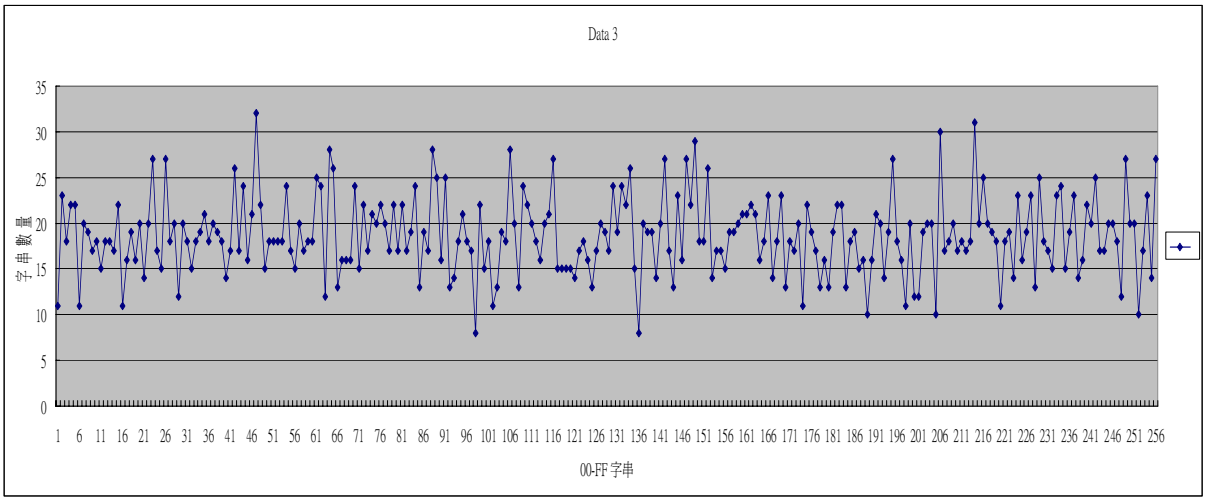

圖 4-27 資料 3 頻率分析圖

第五章　結論與未來發展

5.1　結論

在現今的世界中有許多密碼系統，密碼系統的安全性與效能兩者之間的取捨是個相當大的問題。本論文的加解密模式在第四章複雜度的分析上，為考量計算量與所花費的時間兩項因素，由分析結果看來時間花費上有所增加，也就是說計算量上略有增加，複雜度算是有某程度上的成效，因此本論文的加解密模式為了減少被破解的機率，而在複雜度上略予增加，相對地，也確實在安全性上而有了提高。

在作映射性分析上，是為了知道此加解密模式是否有一對一的映射，若發生一對多的映射或多對多的映射，那麼這密碼系統就失敗了。在映射分析的章節只條列了一部份來做印證，因篇幅有限，其餘的印證部分則置於附錄以加強印證；若要有較完整的映射性分析，則可以執行更大量的資料來做比對，如此的明文集合與密文集合的映射，就可越能顯現出等價的意思。

接著的頻率分析是以另一個方向思考而作的分析，此分析的角度是假設有個意圖破解此加解密模式的駭客，想單單只靠密文來試圖破解，首先此非法使用者必須分析密文特性，於是經由頻率分析圖來看，可看出此三筆資料的頻率分析圖顯示了密文字串分佈的相當均勻，相當不容易破解。若想要破解的話，必須再試試其他方法分析有無某種特徵或其他的規律性出現，否則非法使用者將花費大量的時間破解。

5.2 未來發展

現今電腦科學日益精進，不久的將來光處理器的問世與量子電腦的出現，現今的密碼系統將遭受重大的衝擊，故已有許多學者與研究人員為了大量資訊在網際網路傳遞時的安全性，已把研發的焦點放在次世紀密碼系統的發展上。

本研究由於軟、硬體資源投入有限，所以仍有些地方待改進。比方說，在NP問題（Non-deterministic Polynomial Complete Theory）上的研究，此為相當困難且複雜的數理問題，即求解問題時的時間複雜度與空間複雜度研究，要如何取得最佳化呢？在運算量的問題上，因計算量相當沈重，進而對即時性的影響會如何？在所用數值的取決上，要如何做才能避免統計分析的攻擊？本密碼系統能否往較渾沌的方向設計？重複加密是否能達到此境界？這些問題仍需要投入一些人員與資源來幫助探討。

參 考 文 獻

[1] Feng Bao, Introducing decryption authority into PKI, Computer Security Applications, 2000. ACSAC '00. 16th Annual Conference, pp. 288 – 296, Dec 2000

[2] G. Ateniese,"Efficient protocols for verifiable encryption of digital signatures", Proceedings of the 6th ACM Conference on Computer and Communications Security, 1999.

[3] Ho Won Kim, Sunggu Lee, "Design and implementation of a private and public key crypto processor and its application to a security system" Consumer Electronics, IEEE Trans. Vol. 50 ,Issue 1, pp.214 – 224, Feb 2004.

[4] Kofahi, N.A., Turki Al-Somani, Khalid Al-Zamil, "Performance evaluation of three encryption/decryption algorithms", Circuits and Systems, 2003. MWSCAS '03. Proceedings of the 46th IEEE International Midwest Symposium, Vol. 2, pp.790 - 793, Dec 2003.

[5] Jun Yang, Lan Gao, Youtao Zhang, "Improving memory encryption performance in secure processors" Computers, IEEE Trans. Vol. 54, Issue 5,pp. 630 - 640, May 2005.

[6] Kenvi Wang, Chien-Pen Chuang, Lian-Yuan Lin, "Integrated Security Analysis of IP VPN Based Information System," 2006 International Conference on Information Security conference,pp. 57 – 61, Taiwan, May 2006.

[7] Kenvi Wang, Chien-Pen Chuang, Lian-Yuan Lin, "A Simulated Traffic Engineering Model of IP/MPLS Network Enhancing Multi-Service Performance,"convergence of telecommunications, networking and broadcasting , UK, June 2006

[8] 蔡政安，"高速網路動態化訊務管理策略之研究"，台灣師範大學，碩士，Jun 2001

[9] 賴溪松、張真誠、韓亮，近代密碼學及其應用，松岡電腦圖書資料有限公司，1999

[10] William Stallings，密碼學與網路安全：原理與實務（Cryptography and Network Security：Principles and Practice），曾志光、巫坤品，碁峯資訊股份有限公司，2004

[11] 劉尊全、劉興鋼，破譯 RSA-Cracking RSA，全華科技圖書股份有限公司，2003

[12] Douglas R. Stinson，密碼學原理與實踐，馮登國，電子工業出版社，2003

[13] 張真誠，電腦密碼學與資訊安全（Computer cryptography and information security），松岡電腦圖書資料有限公司，1998

[14] 余泰興，資通安全概論，科資中心，2002

[15] 劉尊全，數位時代密碼技術的現狀與未來，松岡電腦圖書資料有限公司，2001

[16] 霍安琪，密碼學，九章出版社出版，2003

[17] 王旭正、柯宏叡、ICCL 資訊密碼暨建構實驗室，密碼學與網路安全：理論、實務與應用，博碩文化有限公司，2004

[18] 鄧安文，密碼學：加密演算法，全華科技圖書股份有限公司，2004

[19] 孫宗瀛、楊英魁，Fuzzy 控制：理論實作與應用，全華科技圖書股份有限公司，2005

[20] 王金標，模糊理論及其應用，文笙書局，1993

[21] 王文俊，認識 Fuzzy，全華科技圖書股份有限公司，2005

[22] 陳正凱、陳錦輝，C++函式庫精華錄，金禾，2001

[23] 韓丹，擁抱 Visual C++：新世代視窗程式設計，儒林圖書有限公司，2003

[24] Ivor Horton，C++教學範本，蔡明志，碁峯資訊股份有限公司，2001

[25] 侯捷、孟岩，C++ 標準程式庫，碁峯資訊股份有限公司，2002

自　　傳

 我是林連源，父母親都是從小苦過來的，所以我也在「苦」的教育下成長，父母親對我們小孩是言教身教並重，老爸時常講述做人處事的道理且耳提面命「腳踏實地」的重要性，母親則時常讓我們小孩子感受到幸福與甜蜜。從國小高年級至大學除了上學補習課餘，做過許多打工，比如有家庭代工、發傳單、送報紙等等打工，打工的經驗中尤以機械工廠最累，使的我寒暑假每天都腰酸背痛，這些日子讓我比同期孩子較早知道什麼是血汗錢，使我對日後要如何謀生有些許的影響，也學到人與人相處的人際關係哲學、經營管理的策略等等的領悟。

小學高年級起喜歡看百科全書並有時會靜坐養性；國中看各學家書籍與其他國學經典著作；專科則看歷史、哲學、科幻等等的領域書籍；大學時因主修理工，閒時喜愛翻閱投資理財方面的書籍，之後轉向工商管理及其他我不懂的各種領域書刊，不看勵志方面的書籍，會去翻都是好奇為何這上排行榜。大二時認為自學理財有一定的程度瞭解了，於是瞞著家人開始實際進場廝殺。

在大學時期，心裡就不把自己認定只是學生的定位而已，創業的想法已時常在心中響起，創業的念頭在大三、四時才轉為強烈的慾望，可能來自從小至大累積的閱讀知識與生活經驗吧！記得在郭台銘總裁發表某言論說許多人只想作小生意前，我就曾想過要賺國際財，所以在讀研究所時的休息空檔偶爾會和同好中人共同研習，以更邁向夢想一步，不再只是與同好中人談論時自身感動的顫抖；渴望把過去的創新實現，不再只是寫在筆記本上的「預言」。

最後，期望將來自己能與大家同為台灣貢獻心力，使台灣成為轉動世界的主流。這一句話或許大家會覺得很好笑，但它並非只有字眼上的意思，其實有相當深沈的含意，因為我已悄悄的在您心中撥動了一根弦，其效果似如蝴蝶效應。

學 術 成 就

論文發表：

1. Kenvi Wang, Chien-Pen Chuang, **Lian-Yuan Lin**, "Integrated Security Analysis of IP VPN Based Information System," accepted for 2006 International Conference on Information Security conference, Taiwan, May 2006.
2. Kenvi Wang, Chien-Pen Chuang, **Lian-Yuan Lin**, "A Simulated Traffic Engineering Model of IP/MPLS Network Enhancing Multi-Service Performance," accepted for convergence of telecommunications, networking and broadcasting , UK, 26th-27th, June 2006