質數測試於現代密碼學中的應用

作者/周兆龍 上尉・婁德權 博士

提要

早在古希臘時代，人類就已經知道在戰場上用簡單的書信來做秘密的訊息交換。二次大戰期間，密碼技術更開始被廣泛應用在戰場上，例如美國因爲有效破解日軍的通訊密碼，而在太平洋戰爭上能以劣勢的兵力逐漸取得優勢；而英國也不斷破解德軍在戰場所使用的密碼機，進而有效破解截獲的重要情報，取得戰略上的先機。直到60年代，電腦的出現，正式將密碼學(cryptography)帶入了一個新的紀元。

今日電腦強大的運算能力，使得以往傳統的密碼技術顯得有些脆弱不堪，於是近代的密碼學研究重新開始結合一些數學上的特殊性質或定理，希望能找到符合現代需求而又不易被破解的密碼系統。質數測試(Primality Testing)便是廣泛應用在現代密碼學上的一種數學方法，可以增進許多現代密碼演算法的效率與安全性。

前言

密碼術(cryptology)，這個字源自於希臘文的"Kryptos"及"logos"，指的是"隱藏"的"文字"。一般所謂的密碼術包含了兩大分支，一是密碼學(cryptography)，另一是破密學(cryptanalysis)。密碼學泛指與密碼加解密相關的技術或科學，而破密學主要是關於如何破解密碼中所隱含意義的技術或科學。本文的討論重點將以密碼學的範疇爲主。簡單來說密碼就是將訊息本身的「意義」隱藏起來，轉變成無法理解的文字或符號，讓他人即使攔截到密碼訊息也無法得知這些訊息原本的眞正含意。

傳統密碼的基本作法可分爲「代換」(substitution)與「移位」(transposition)。代換指的是將原始的訊息用不同的訊息取代，舉例來說，若有下列一段訊息要傳送：

「這是一個重要的秘密」

若用0100代表「這」，0101代表「是」，0110代表「一」……等等，則上面那句話就可以表示爲：

「010001010110...」。

至於移位，就是將訊息用不同的方式將原順序打散重新排列，舉個例子，若有下

列一段訊息要傳送：

「THIS IS A SECRET MESSAGE」。

假如將每個奇數位字母放在第一排，偶數位字母放在第二排，如下列所示：

T I I A E R T E S G

H S S S C E M S A E

則原本那句話就可以表示爲：

「TIIAERTESGHSSSCEMSAE」。

這類傳統密碼的傳送與接收雙方必須同時知道密碼的運作方式，才能順利的傳遞訊息，而不知道密碼運作方式的人，看到上面這些數字或符號時，便無法得知眞正的原意。若想破解這種密碼，需要靠「暴力法」不斷去嘗試各種可能的排列或替代的方式，若用人力計算十分費時，但若靠電腦強大的運算能力，則這種密碼方式就十分容易遭到破解。

密碼在軍事上的用途，早在古希臘時代就有文獻記載，直到二次世界大戰期間，美、日與英、德更於戰場上進行激烈的競爭。自從60年代電腦開始流行，軟硬體能力不斷的發展突破，許多傳統的密碼規則在這些電腦面前逐漸無法維持安全性，紛紛遭到破解，因此各種新的密碼概念相繼提出，利用了許多如數論(Number Theory)的數學理論，爲的就是要加強密碼以抵擋電腦分析與運算破解的能力。

近代密碼學開始受到各界重視是從1977年美國訂定國家資料加密處理標準DES(Data Encryption Standard)開始，DES由美國IBM電腦公司研究開發，爲第一個供美國政府各機關所使用的密碼標準，於1998年則出現了AES(Advance Encryption Standard)這個更安全的區塊加解密系統。而近代密碼學最具代表性的突破是1976年由Diffie與Hellman兩位學者所提出的公開金鑰(public-key)概念；接著1978年美國麻省理工學院Rivest、Shamir及Adleman三位學者利用了這種概念，完成了第一個眞正可以應用的公開金鑰密碼系統RSA。因爲公開金鑰概念的提出與RSA系統的成功，陸續有許多各式的密碼演算法被發展出來，密碼學也正式邁入一個新的里程碑。

本文將介紹現代密碼學的演進與基本概念、公開金鑰密碼系統相關技術、質數測試方法以及未來的發展方向。

密碼系統

在密碼學中，將訊息轉變的過程稱作加密(encryption)，而將轉變後的訊息還原的過程稱作解密(decryption)，一般所稱的密碼系統(cryptosystem)必定同時包含加解密的程序，常見的密碼系統如圖一所示：

圖一中X表示欲傳送的原始文件，稱做明文(plaintext)；Y表示經過加密處理後的加密文件，稱做密文(ciphertext)；$E_e(X)$表示加密的演算法；$D_d(Y)$表示解密的演算法。整個系統運作的流程是：

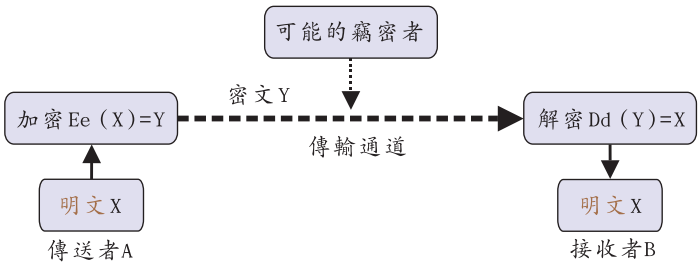
一、e與d分別代表加密金鑰與解密金鑰。

二、由傳送者A將明文X用加密演算法$E_e(X)$運算後成爲密文Y。

三、密文Y經過一個傳輸通道(channel)傳給接收者B。

四、接收者B收到密文Y之後用解密演算法$D_d(Y)$運算後還原爲明文X。

其中傳輸通道泛指常見的傳輸資訊的方式，如衛星、微波、電話、網路等等。當訊息在一個不安全的傳輸通道中傳遞時，很



▲圖一　一般密碼系統架構圖(作者自繪)

容易遭受別人的竊取、監聽、假冒或竄改，因此密碼系統通常要求須達到「私密性」(privacy)與「鑑別性」(authentication)兩項安全需求。「私密性」指的是訊息即使在傳遞的過程中遭到非法竊取，也能確保不被他人破解出眞正的訊息；而「鑑別性」指的是能判斷收到的訊息是否經過竄改及其來源是否正確。

至於加密與解密，就是利用演算法(algorithm)來做資料的轉換，這類轉換通常是利用一些特殊的數學公式配合金鑰(key)來做計算。爲何會需要使用金鑰呢？基本上是爲了使用的方便性與安全性，如果一個密碼系統的運作方法，不愼遭破解，若要重新改用新密碼系統，實在不符成本效益。最直接方便的方法，就是沿用原來的演算法，只將金鑰的數值更改，別人就必須重新嘗試破解的方法；此外，在安全需求較高的環境下，只要常常自行更換金鑰，讓他人無法有足夠的時間嘗試破解，同樣能使整個系統保持一定的安全性。因此，金鑰在密碼系統中扮演了一個很重要的角色。

一般我們將加解密使用同一把金鑰的系統稱爲秘密金鑰密碼系統(Private-key Cryptosystem) 或對稱式密碼系統(Symmetric Cryptosystem)，如DES與AES等密碼系統；而如果加解密分別使用彼此配對的公鑰(public-key)及私鑰(private-key)，就稱爲公開金鑰密碼系統(Public-key Cryptosystem)或非對稱式密碼系統(Asymmetric Cryptosystem)，如RSA密碼系統。表一列出秘密金鑰與公開金鑰密碼系統間一些明顯的差異。

簡單來說秘密金鑰密碼系統一般指的就是傳統密碼系統。通常這類方法不需要大量的計算，金鑰長度較短，產生金鑰的效率也較高，但若他人也知道其加解密的方法，則只要在網路上能擷取到訊息，就能輕易知道其訊息內容，因此安全性稍差，所以會需要一個安全的傳輸通道(Secure Channel)來增強其安全性。而公開金鑰密碼系統，使用大量複雜的計算來確保他人無法在短時間內破解，通常這類系統的金鑰長度較長（如RSA要求至少1024位元以上），產生金鑰也較耗時，而且加密與解密使用不同的金鑰，在網路上所傳遞的訊息是由接收者的公鑰加密而

秘密金鑰密碼系統	公開金鑰密碼系統
計算簡單快速	計算複雜耗時
加解密使用同一把金鑰	加解密分別使用配對的兩把金鑰
金鑰需保密	僅私鑰需保密，公鑰可公開
金鑰管理成本較高	金鑰管理成本較低
安全性靠演算法及金鑰之保密	安全性靠配對金鑰之分解難度
對傳輸通道的安全要求較高	對傳輸通道的安全要求較低

▲表一　秘密金鑰密碼系統與公開金鑰密碼系統間的比較(作者整理)

成，即使訊息遭人擷取，因無法輕易算出其配對的私鑰爲何？因此安全性較高，也不需過度仰賴安全的傳輸通道，在網際網路盛行的現今，是比較適合的一項選擇。

除此之外，這兩類密碼系統所需要的金鑰數目也有很大的不同。以秘密金鑰密碼系統而言，每對使用者在互傳訊息時要用同一把鑰匙，當有n位使用者要互傳資料時，就需要n(n-1)/2把鑰匙；而公開金鑰密碼系統，每位使用者僅需保留一對鑰匙中自己應保留的一把鑰匙，因此當有n位使用者要互傳資料時，只需要2n把鑰匙。當使用者數量很多時，金鑰管理的成本與應用的效能就會產生明顯的差異。在電腦計算能力愈來愈強大的今日，若考慮金鑰管理的實用性與安全性，仍是以公開金鑰密碼系統較爲合適。因此現代密碼系統的研究重心幾乎都是圍繞在公開金鑰密碼系統上的相關應用。

著名公開金鑰密碼系統

在介紹幾個著名的公開金鑰密碼系統之前，先看看最早在1976年美國史丹福大學的Diffie與Hellman兩位學者所提出公開金鑰的概念，他們提出一種單向暗門函數(Trapdoor One-way Function)的概念。基本上所謂的單向函數(One-way Function)就是指單向的計算很容易，但反方相的計算卻很困難的一種函數。例如有一單向函數$y=f(x)$，其反函數即爲$x=f^{-1}(y)$，要能符合下列條件：

一知道x，要很容易求出y，

二知道y，要很難求出x。

而單向暗門函數本身即是單向函數的一種，差別在於隱含著某種暗門(trapdoor)，只要知道這個暗門，就能讓反方向的計算變得很容易。舉例來說，有一個函數$y=a\times b\times c$，其中a, b, c均爲質數，若已知a, b, c要求出y是輕而易舉，若知道y要求出a, b, c則相對困難得多，但是知道暗門則又變得容易得多，例如已知y=13868153，要求a, b, c就不太容易，但若掌握了暗門，例如已知其中一數爲911則求解a, b, c就化爲分解15223(13868153/911=15223)的問題，顯然非常容易得到$13868153=13\times911\times1171$。這種單向暗門函數的正運算與反運算應用在密碼系統中的加密與解密程序，就產生所謂的公鑰(public-key)與私鑰(private-key)的概念。公鑰與私鑰互爲某一函數的正反運算，公鑰就好比是經過函數運算結果，可以在網路上公開；而私鑰就好比是那個「暗門」，需要仔細保管不讓人知道。

當時Diffie與Hellman提出此想法，受到許多人的注目，不過他們並沒有找到一個眞正符合條件的單向暗門函數。直到1978年美國麻省理工學院的Rivest、Shamir及Adleman三位學者利用了這種概念，首次提出符合這種條件的演算法，就是現今鼎鼎有名的RSA密碼系統。RSA密碼系統是目前網際網路電子商務(e-commerce)上被廣泛使用的一種密碼系統，也是最早應用公開金鑰概念的密碼系統。他們利用因式分解(Integer Factorization)的數學方法來設計這套密碼系統，整個系統當中利用了許多數學的理論，如質數性質、尤拉定理、費瑪小定理、同餘定理、歐基里得演算法、模指數運算等。RSA密碼系統的運作流程說明如下：

一找兩個大質數p、q。

二假設$n=p\times q$，算出n的尤拉ψ函數$\psi(n)=$

(p-1)×(q-1)。

三找一個與$\psi(n)$互質的數e，亦即GCD(e, ψ(n))=1。

四再找一個d，條件需滿足ed≡1mod $\psi(n)$。

假設我們以X代表明文，Y代表密文，在RSA中加解密的動作可以分別表示爲：

$Y=(X)^e \bmod n$，

$X=(Y)^d \bmod n$。

舉個例子，假設A要傳資料給B，A只需知道B的公鑰與n，整個過程大致如下：

一B要先找出兩個大質數，假設B任選質數p=47，q=71，可以得出n=3337且$\psi(n)$=46×70=3220。

二接著B任選與$\psi(n)$=3220互質的數e=79當作公鑰，代入79×d≡1 mod 3220，即可得到d=1019當作私鑰。B可以將公鑰e=79與n=3337公開給其他人，但須保留自己的私鑰d。

三假設A要傳送訊息X=688，只需將B的n與公鑰拿來計算，得出Y=(688)79=1570 mod 3337。因此A傳送給B時是傳送Y = 1570的值。

四當B收到加密訊息Y=1570時，便利用自己的私鑰d=1019與n=3337代入$X=(1570)^{1019}=668 \bmod 3337$，而得出正確的原始訊息X=668。

就理論上看來，RSA的加解密動作並不複雜，只是利用了指數運算、模式運算及因式分解等方法，那RSA到底是如何維持系統的安全性呢？關鍵就是因式分解，因爲因式分解一直就是數學上非常困難的問題，隨著數字愈來愈大所花的時間會成等比增加，更何況實際在電腦上的應用並不是像上面例子這樣簡單的數字，而是至少大於10^{65}（約128位元）的大整數，即使計算能力強大的電腦，也需花上數日甚至數月才能因式分解成正確的數值。

RSA當中的n是由兩個大質數p和q相乘所得出的值，之後的公鑰e與私鑰d也是經過p和q計算求得的，因此要想破解RSA密碼就必須想辦法將n的p和q因式分解出來。假設p和q長度爲256位元，則n就會是一個長度爲512位元的數字。當p和q大到無法被因式分解出來，公鑰與私鑰就不會被人猜出，也就代表整個系統是安全的。現今有許多學者專門針對RSA密碼系統在研究到底需要多少長度的數字才能保障系統的安全，因此有許多專門嘗試因式分解大數字的方法相繼提出，例如Quadratic Sieve、Number Field Sieve等。表二列出近年來已被成功分解的RSA數字長度。

最近被分解出的數字爲RSA-160(530位元)，是由德國BSI研究機構所完成，大約使用了一百多部工作站等級的電腦運算，花費近六個月才完成，可見因式分解的難度。現

數字長度(digit)	位元長度(bit)	使用方法	破解日期
RSA-100	330	Quadratic Sieve	1991 四月
RSA-110	364	Quadratic Sieve	1992 四月
RSA-120	397	Quadratic Sieve	1993 六月
RSA-129	425	Quadratic Sieve	1994 四月
RSA-130	430	Number Field Sieve	1996 四月
RSA-140	463	Number Field Sieve	1999 二月
RSA-155	512	Number Field Sieve	1999 八月
RSA-160	530	Number Field Sieve	2003 四月

▲表二　已被破解的RSA數字(作者整理)

今一般使用RSA都已經建議p和q至少要1024位元以上較安全。

其他著名的公開金鑰密碼系統還有1985年ElGamal密碼系統，利用離散對數(Discrete Logarithm)的方法在有限場(Finite Field)中進行加解密運算；同樣在1985年提出的橢圓曲線密碼系統(Elliptic Curve Cryptosystem)，則利用橢圓曲線方程式特性與多項式在有限場中的運算，來作爲加密與解密運算的演算法。

質數測試

一為什麼需要質數測試

在現代公鑰式密碼系統中，應用了如模式運算(Modular Arithmetic)、尤拉公式、費瑪定理、中國餘式定理(Chinese Reminder Theorem)、質數理論、離散對數、因式分解、抽象代數(Abstract Algebra)等數學基礎。而質數(Prime Number)更與前述理論密切相關最常被拿來研究與應用的數字，因爲質數本身具有許多的特性，很適合應用在一些單向暗門函數的運算。例如RSA當中必須使用兩個大的質數來做加解密運算；ElGamal密碼系統利用質數來建立一個有限場Zp以產生其公鑰與私鑰；橢圓曲線密碼系統利用質數在橢圓曲線方程式上可找到整數解的特性，來產生其公鑰與私鑰。質數在現代密碼系統中扮演了十分重要的角色，沒有質數，這些豐富的密碼演算法就將無用武之地，因此如何選擇合適的質數就成了現代密碼學中一個重要的課題。質數測試(Primality Testing)可有效地找出質數，以便讓這些公開金鑰系統來使用。

二什麼是質數測試

簡單來講質數測試就是將一個整數拿來測試是否爲質數(prime)，只要不是質數的統統稱之爲合成數(composite)。

古希臘的一位數學家Eratosthenes發現了一種找質數的簡單方法，他利用了算術基本定理，假如我們想找出1~50之間有哪些數字是質數，我們可以畫一個表格，如圖二所示。

先從最小的質數2開始，消去含有質因數爲2的數字(亦即消去2的所有倍數)，如4, 6, 8, 10..., 50（用星號*標示），再消去掉含有質因數爲3的數字，如9, 15, 21,..., 45（用刪除線－標示），依序再針對質因數爲5及7的數字（分別使用底線_及雙底線＝標示），因爲數字最大爲50，最多只要算至小於$\sqrt{50}$的最大質因數7即可，最後剩下未被刪除的數字爲2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47，即爲1~50之間的所有質數。

這種方法的名稱叫做Sieve of Eratosthenes或稱爲試除法(Trial Divisions)，這方法可以算是最早的質數測試法，觀念簡單，方法也不難，而且在數字範圍不大時，非常快速，但是當我們想找的質數數字範圍很大時，這個方法就十分沒有效率了。舉例

1	2	3	4*	5	6*	7	8*	~~9~~	10*
11	12*	13	14*	~~15~~	16*	17	18*	19	20*
~~21~~	22*	23	24*	25	26*	~~27~~	28*	29	30*
31	32*	~~33~~	34*	35	36*	37	38*	~~39~~	40*
41	42*	43	44*	~~45~~	46*	47	48*	49	50*

▲圖二　Eratosthenes尋找質數方法(作者整理)

來說，如果想找1~10^{10}（約20位元）間的質數，概略將近有4億5千個質數，如果以電腦每秒找一個質數的效率來看，就需要約14年(60×60×24×365×14=441504000)的時間才能算完，實在不符效益。因此我們需要找到其他更有效率或者變通的方法。

數學家費瑪(Fermat)在1640年發現了一個定理，稱爲費瑪小定理(Fermat Little Theorem)。這個定理是說：『假設n是一個質數，a爲與n互質的任意整數，則存在a^n=a mod n的關係』。例如若n=5，a=4，則4^5=1024=4(mod 5)。這裡有一個重點要注意的，就是當n已確定是質數時，a^n=a mod n的條件式會成立；但若有任意a、n也能滿足該條件式時，並不能保證n一定就是質數。利用這種性質，我們可以用來設計一個簡單的質數測試，稱做『費瑪質數測試法』(Fermat's Primality Test)。方法如下：

㈠待測數當作n，再任選一個與n互質的數a。

㈡將n、a分別代入a^n=a mod n式子做檢驗。

㈢若是$a^n \neq$ a mod n則確定n一定不是質數；反之則判定n是質數。

這種方法，嚴格來說不是眞正的質數測試法，而算作是合成數測試法，因爲質數測試法基本上結果只有兩種：「是質數」或「不是質數」，而費瑪測試法結果如果「不是質數」，那這個數字保證一定不是質數；但若結果「是質數」時，卻不能保證這個數字一定就是質數，可能有一些仍是合成數。爲什麼會有這種情形呢？以日常生活上的例子來看，例如我們想從裝滿稻米與黃豆的籃子中挑出稻米，可以直接從籃子裡一個一個挑，也可以靠篩子將稻米與黃豆分開。用篩的方法因爲稻米顆粒較小所以都能順利被過濾出來，而黃豆可能因爲部分的顆粒較小或篩子洞口稍大，仍可能有一小部分會跟著被過濾出來，這種情形雖有稍微的誤差，仍比在籃子中一個個去挑出稻米有效率得多。

費瑪質數測試法有點類似這種情況，要從無限多的整數中找出質數，若按照大小順序來找，耗時費力；反之如果利用一些特殊公式，能讓質數很容易被找出，雖然有一小部分的非質數混在其中，但因爲效率較高，也能適合一般的應用需求。像Sieve of Eratosthenes這種直接將質數挑出的作法，一般稱爲確定式質數測試法(Deterministic Primality Test)。而像費瑪質數測試法這類的方法，就稱爲機率式質數測試法(Probabilistic Primality Test)。確定式質數測試法，保證測試出來的結果一定是質數，因此耗費的時間極長，比較適合應用在機密性較高的軍事、商業等方面；至於機率式質數測試法，雖然結果不保證一定是質數，但也都會要求在極低的錯誤率以下，速度相對比較快，在網際網路或機密性不高的個人應用方面，即可滿足需求。所以針對不同的應用，可以選擇適合的質數測試法。

三、Miller-Rabin質數測試法

通常我們將通過機率式質數測試法判定爲質數的數字稱爲「假質數」(pseudo-primes)，表示不保證這些數字全部是質數。費瑪質數測試法所產生的假質數有個比較明顯的缺點，就是發現所謂的Carmichael number這種形式的整數，能通過費瑪質數測試法，但實際上卻是合成數的機率很大。有另一種方法可以改進這個問題，稱做

Miller-Rabin質數測試法。

Miller-Rabin質數測試法是由1974年Miller最早提出的方法，而後由Rabin加以改進，也是目前最常使用的一種機率式質數測試法。這個方法可以有效解決費瑪質數測試法無法判別Carmichael number的問題，同時也提供了更有效率、錯誤率更低的測試方法。這個方法利用了費瑪小定理延伸出下列兩個條件：

㈠$a^r=1 \pmod n$，$1 \leqq a \leqq n-1$。

㈡$a^{2^j r}=-1 \pmod n$，$0 \leqq j \leqq s-1$。

這裡將質數n表示爲$n=2^s r+1$，其中r是奇數。Miller-Rabin的質數測試方法就是將數字n代入上面兩個條件式去做驗證，假設不滿足其中任何一項，則n一定不是質數，反之如果n能同時滿足上面兩個條件式，則n即爲假質數。其中a這個基底(base)數字的選擇會影響Miller-Rabin質數測試方法的正確性。

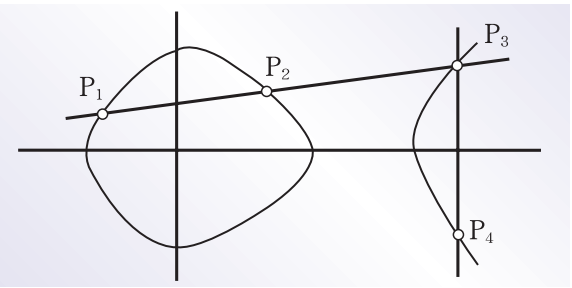
舉個例子，假設有一個數字n=91(7×13)，利用Miller-Rabin的方法，先將$91=2^s r+1$，可得s=1，r=45，繼續將r代入條件式中$a^{45}=1 \pmod{91}$，任選一個介於0~90之間的a代入驗證，發現除了當a爲{1、9、10、12、16、17、22、29、38、53、62、69、74、75、79、81、82、90}這18個數字時，Miller-Rabin的方法會誤判91爲質數；當a爲其他數字時，都能順利將91判斷爲合成數。換句話說，當a有90個基底範圍時，約有18/90≒1/5的誤判率。實際上經過許多實驗的結果，Miller-Rabin質數測試法的誤判率每次最多不會超過1/4，一般爲了降低誤判率會重複執行k次測試，每次選取不同的基底做運算，則錯誤率就可以有效降至約$(1/4)^k$。例如重複執行Miller-Rabin質數測試法10次($(1/4)^{10}$~2^{-20})，錯誤率約百萬分之一。

四橢圓曲線質數測試法(Elliptic Curve Primality Proving, ECPP)

在公開金鑰密碼系統中另一個著名的密碼系統爲橢圓曲線密碼系統(Elliptic Curve Cryptosystem)，應用了橢圓曲線的數學理論來作爲其加解密的演算法。就安全性而言，RSA系統通常金鑰長度需要1024位元以上，而橢圓曲線密碼系統相對只要約160位元長度的數字(金鑰)就能達到相同的安全性，對於金鑰的儲存管理有較好的改善。

橢圓曲線應用在質數測試中，最常見的是像圖三所表示的這種曲線。

一般密碼學中所謂的橢圓曲線是指$y^2=x^3+ax+b$這個橢圓方程式，事實上橢圓曲線並不是眞的橢圓形，而是會依據橢圓方程式中各項係數的不同而改變形狀。圖三中P_1與P_2爲橢圓曲線中的點，P_1與P_2連成的直線能與另一段弧線的交點P_3相交，且P_3以x軸映射後即找出P_4這點。橢圓曲線測試法主要就是利用質數可以在$y^2=x^3+ax+b$這個橢圓方程式中找出P_1、P_2、P_3與P_4這些整數解，而合成數不具有這樣的特性，用來當做質數測試的方法。不過橢圓曲線測試法仍有許多數學條件的限制，而且仍有許多學者提出不同形



▲圖三　橢圓曲線示意圖(作者自繪)

式的測試方式，在此就不詳細贅述。

橢圓曲線測試法是確定式質數測試法，能正確的判斷出質數，但是計算效率仍沒有一般機率式質數測試法來得好，因此橢圓曲線測試法在實用上最常被拿來當作機率式質數測試法的檢驗工具，用來確保其他機率式測試法結果的正確性。

五AKS質數測試法

電腦演算法(algorithm)中一般可以分爲非多項式時間NP(non-polynomial)與多項式時間P(polynomial)兩大類。屬於非多項式時間的演算法通常表示一個很難解決的難題，無法估算輸入的資料需耗費多少執行時間才能求出答案；而屬於多項式時間的演算法通常表示一個較簡單的問題，可以估算需耗費多少的時間求出答案。

以往的質數測試法都是屬於非多項式時間的演算法，因爲一直沒有一個質數測試法能眞正單純用演算法本身分析出時間複雜度，而大都需要依靠許多實驗值來估算大約的時間複雜度。直到2002年有了一個重大的突破，就是由印度的三位學者Agrawal、Kayal及Saxena所發表的AKS演算法，這個演算法不僅是確定式的質數測試法，也是屬於多項式時間的演算法。這個方法是說：『假設a與n互質，若n是質數，則一定滿足$(x-a)^n=(x^n-a) \pmod{n, x^r-1}$』。

這個方法利用上面這個簡單的數學恆等式，在有限場(Finite Field)中做多項式模式運算，嘗試去代入所有不同的a與r，檢查n是否符合上列式子的條件。只要是質數一定能滿足上列恆等式，而只要不是質數一定無法滿足上列恆等式。AKS的特色在於利用了數論中基本的恆等式，可以將質數正確的判斷出來，而且演算法可以利用數學分析的方式，直接算出執行的時間複雜度。雖然AKS因爲演算法中需要很多$(x-a)^n$的指數運算，而需要耗費許多執行時間，但是讓質數測試演算法能成爲多項式時間的象徵意義，就如同宣告質數測試在理論上是屬於簡單的問題，也因此受到許多人的重視。在經過許多人證明之後，確定這個演算法是正確無誤的，時間複雜度大約在$O((\log n)^{12})$左右。

六實用性分析

許多公開金鑰密碼系統中，產生質數的動作通常要在進行加解密運算之前就決定好，而後系統在使用時就直接拿這些數字來做運算。這種前置處理的好處是可以利用其他的電腦專門負責質數產生的工作，增進系統實際執行加解密運算時的效率；並且也利用其他的電腦檢驗所產生的質數是否正確，因此也有較高的可靠度。

一般程序是先依據使用者需求產生適當長度的隨機(random)數字，接著利用質數測試法來檢驗這些數字是否是質數，如果是質數，便可以將之儲存至資料庫內備用。這裡就應用到之前所介紹的各種質數測試法，而這些方法到底該如何選擇應用呢？我們就拿Miller-Rabin測試法、橢圓曲線ECPP測試法與AKS測試法來做個執行效率的比較，如表三所示。

這裡針對一個330 位元長度的數字來執行質數測試，其中Miller-Rabin測試法是屬於機率式質數測試法，ECPP與AKS都是屬於確定式質數測試法，因此Miller-Rabin測試法在執行速度上比ECPP或AKS都快得多，但正確性就不如ECPP與AKS。不過Miller-Rabin測試法仍可以利用反覆執行的

方法名稱	執行次數	耗費時間
Miller-Rabin	1	0.01秒
Miller-Rabin	10	0.10秒
Miller-Rabin	100	1.00秒
ECPP	1	6.68秒
AKS	1	2.36年

▲表三　質數測試法時間比較圖(作者整理)

方式，降低其錯誤的機率，例如RSA建議只要質數測試方法錯誤率低於2^{-100}，就算合格，若以Miller-Rabin測試法來執行，只要反覆執行50次($(1/4)^{50}$~2^{-100})就可以達到合格需求了。

一般經過實驗後估算每次執行Miller-Rabin測試法的電腦演算複雜度約$O((\log n)^2)$，ECPP約$O((\log n)^6)$，而AKS約$O((\log n)^{12})$。從表三中來看，Miller-Rabin測試法執行100次只需要1秒鐘的時間，而執行1次的ECPP需要6.68秒，AKS則需要將近2.36年的時間，明顯的可以看出AKS所花的時間過長，在實用上並不適合採用。因爲密碼系統本身應具有一定的執行效率與安全性，但有時候兩者很難同時兼顧，如果爲了要安全性，使用像AKS這種方法，恐怕會嚴重影響時效性，因此有時寧可選擇使用假質數，以爭取時效。例如RSA或ElGamal密碼系統本身的難度在於因式分解或離散對數的計算上，只要數字夠大就足以讓人難以破解，質數有時只是增加破解的難度而已。再加上如果每次數字使用後又再更換成不同的數字，在實用上仍能保持良好的安全性。

目前在許多應用上常見的是先使用Miller-Rabin質數測試法，針對一個數字反覆選取不同的基底測試數次後，如果答案仍判定爲質數，就再利用橢圓曲線質數測試法加以確認。這種方式利用Miller-Rabin質數測試法節省執行時間，又利用橢圓曲線質數測試法確保安全性，很適合在各種領域上的應用。當然質數測試方法可以依據需求來使用，並沒有限制，只要能符合使用者安全及效率上的考量，各種質數測試方法都可以直接採用或搭配使用。

結論

質數測試法在現代密碼系統中已經應用十分廣泛，可以用來產生大質數以供密碼系統使用，也能用來評估公開金鑰密碼系統中公鑰與私鑰是否具有足夠的安全性。面對密碼系統中所使用的質數長度愈來愈長的趨勢，執行速度快的機率式質數測試法可說是目前在實務上使用的優先考量。而AKS演算法的提出，象徵了質數測試法的一個新里程碑，未來勢必會朝向增進執行速度與同時具備正確性的方向來發展。數學在這方面所扮演的角色也將愈來愈重要，許多數學上的簡單定理，可能有一天會被應用成爲更安全快速的質數測試演算法。

現今網際網路的影響無遠弗屆，電子化政府、電子商務、電子郵件等等應用，也讓電腦與人的關係愈來愈密切。密碼學勢必也將突破軍事或商業上應用的侷限，逐漸朝向個人應用來發展。如今密碼已受到全世界普遍的重視，但同時也有許多國家刻意把與密碼相關的發展訊息封鎖，甚至不允許相關技術繼續被公開討論，可見世界各國對密碼研究的謹愼態度，以及對密碼這種又愛又恨的複雜情緒。量子電腦、DNA演算法、生物電腦等新的技術概念不斷被提出，未來密碼的型態也許會有令人意想不到的改變。因

此除了期待未來密碼學的新風貌之外，持續堅定在這個領域的研究發展，才是保有競爭優勢的實際作法。

參考資料

一、S. Y. Yan, Number Theory for Computing, Springer-Verlag, 2nd Edition, 2002.
二、W. Stalling, Cryptography and Network Security - Principles and Practices, Prentice-Hall, 3rd Edition, 2002.
三、A. Menezes, P. van Oorschot, and S. Vanstone, Handbook of Applied Cryptography, CRC Press, 3rd Edition, 1996.
四、B. Schneier, Applied Cryptography, John Wiley & Sons, 2nd Edition, 1996.
五、R. M. Young, Cryptography and Secure Communications, McGraw-Hill, 1994.
六、K. H. Rosen, Elementary Number Theory and Its Application, Addison-Wesley, 4th Edition, 2000.
七、J. A. Buchmann, Introduction to Cryptographyecry, Springer-Verlag, 2000.
八、D. E. Knuth, The Art of Computer Programming - Fundamental Algorithms, Addison-Wesley, 3rd Edition, 1997.
九、R. P. Brent, "Primality testing," seminar presented at Clemson University, South Carolina, 4 April 2003.
十、M. Agrawal, N. Kayal, and N. Saxena, "PRIMES in P," August 2002.
十一、R. Bhattacharjee, and P. Pandey, "Primality testing," IIT Kanpur, 2001.
十二、A. O. L. Atkin, and F. Morain, "Elliptic curves and primality proving," Math. Comp., Vol.61, no.203, July 1993 ,pp. 29-68.
十三、T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," IEEE Trans. Inform. Theory, Vol.31, 1985, pp.469-472
十四、R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Comm. ACM, Vol.21, 1978, pp.120-126.
十五、http://www.rsasecurity.com/rsalabs/challenges/factoring/rsa160.html.
十六、http://members.tripod.com/irish_ronan/rsa/primality.html.
十七、http://mathworld.wolfram.com/PrimalityTest.html.
十八、賴溪松，韓亮，張眞誠，近代密碼學及其應用，（台北：松崗電腦圖書資訊有限公司，1999)。
十九、楊吳泉，現代密碼學入門與程式設計，(台北：全華科技圖書有限公司，1995)。
二十、劉燕芬譯，碼書-編碼與解碼的戰爭，（台北：台灣商務印書館，1995)。

作者簡介

周兆龍上尉，中正理工學院資訊系58期。曾任電腦硬體工程官。現於國防大學中正理工學院電子工程研究所攻讀碩士。

婁德權博士，中正理工學院電機系47期、國立中山大學電機研究所碩士、國立中正大學資訊工程所博士。曾任排長、裝載官、助教、講師、副教授。現任國防大學中正理工學院電機系教授兼電子組組長。