數學傳播 35卷4期, pp. 34-48

公鑰密碼之旅

沈淵源

我們依舊緊跟著傳統密碼之旅[17, 18]中的三個主要人物: 張三毛、李四郎和王五爺, 一起來暢遊密碼之旅的下半部; 此乃公鑰密碼之旅, 亦即近代密碼之旅也。

出發之前, 我們先瀏覽一篇數學遊戲專欄中的有趣報導; 就算是公鑰密碼之旅的暖身操, 或說是品嚐近代密碼「世紀大餐」的開胃菜。

1. 數百萬年才解得開的新式密碼

科學的美國人(Scientific American) 雜誌於 1977 年 8月在數學遊戲專欄中出現了一篇文章題爲「數百萬年才解得開的新式密碼[8](A new kind of cipher that would take millions of years to break)」, 作者就是該雜誌的數學遊戲專欄作家馬丁卡德那(Martin Gardner)。文中他介紹了瑞沙葉密碼系統 (RSA Cryptosystem)。在解釋完加密方式如何運作後, 卡德那代替 M.I.T. 的作者群向讀者提出一項挑戰 (簡稱 RSA-129)。他刊出一則密碼文 c, 以及其加密鑰匙 (n, e) 如下:[1]

```
In[1]:= n = 114381625757888867669235779976146612010218296\
            721242362562561842935706935245733897830597123\
            5639587050589890751475992900268795435 41;
          e = 9007;
          c = 968696137546220614771409222543558829057599 9\
            112457431987469512093081629822514570835693 1\
            476622883989628013391990551829945157815154;
```

[1]數學套裝軟體MATHEMATICA的簡介請參考數論輕鬆遊[16]第一節。

這項挑戰就是分解整數 n, 然後再將密碼文 c 解密。這是已知唯一找到明文的方法。在 1977 年那個年代, 據估計用當時最新的分解法, 需要 4×10^{16} 年的時間才能完成。所以作者認爲挺安全的提供了 $100 美元獎金給任何在 1982 年 4 月 1 日之前破解的人。爲了證明這的確來自 M.I.T. 的作者群, 還特別附上簽名如下:

```
In[4]:= s = 1671786115038084424601527138916839824543690\
            1032358311217835038446929062655448792237114\
            4905095786086556624965779748400040570203 73;
```

然而, 由於因數分解技巧的不斷精進與改良, 不用數百萬年, 此項挑戰終於在 17 年後就被破解[1]。1994 年 4 月 26 日由 Atkins, Graff, Lestra 與 Leyland 成功地分解了上面的整數 n。

600 位自告奮勇人士, 連同總數達 1600 部的電腦, 利用閒暇時間加入工作的行列, 一起尋找所需要的關係式。透過電子郵件回報至中央機器統一整合, 去掉重複的, 並將資料儲存在一個更大的矩陣當中。經過七個月的努力, 他們得到一矩陣有 569466 列與 524338 行。很幸運地, 這個矩陣相當的稀疏, 亦即當中有許多元素爲 0, 因此可以很有效率的儲存。用高斯消去法將此矩陣縮簡至一非稀疏的矩陣有 188614 列與 188160 行。所花的時間不超過 12 個小時。再花另外 45 個小時的計算, 他們找到了 205 個關係式。前三個什麼也沒得到, 但第四個就得到了 n 的分解式爲:

```
In[5]:= p = 3490529510847650949147849619903898133417764\
            638493387843990820577;
        q = 3276913299326670954996198819083446141317764\
            2967992942539798288533;
```

算出 $9007^{-1} \pmod{(p-1)(q-1)}$, 得到解密次冪爲:

```
In[7]:= d = PowerMod[e, -1, (p - 1)*(q - 1)]
Out[7]=1066986143685780244428687713289201547807099066 3\
       3937862801226224496631063125911774470873340168 5\
       9746230655396854451327710905360609 5
```

計算 $c^d \pmod{n}$, 得到明文信息爲:

```
In[8]:= m = PowerMod[c, d, n]
Out[8]= 200805001301070903002315180419000118050019172\
        105011309190800151919090618010705
```

用 01=A, 02=B,. . ., 26=Z, 且 00=空白, 代換回去文字得到

```
In[9]:= abc = " abcdefghijklmnopqrstuvwxyz";
no="000102030405060708091011121314151617181920212223242526";
        digitalize=Table[StringTake[abc,{i}]->
           StringTake[no, {2*i-1, 2*i}], {i, 1, 27}];
        alphabetize=Table[StringTake[no,{2*i-1,2*i}]->
                    StringTake[abc,{i}],{i, 1, 27}];
        A[digit_]:=StringReplace[digit, alphabetize];

In[14]:= A[ToString[m]]
Out[14]= the magic words are squeamish ossifrage
```

(a squeamish ossifrage 是超敏感的鷹, 信息如此選擇是爲了防止猜測)。詳情見論文[2], 或查看由 Atkins, Graff, Lestra, 與 Leyland 透過電子郵件所發布的消息: http://www.mit.edu:8001/people/warlord/RSA129-announce.txt 或拜訪 Derek Atkins 的網頁[2] 。

至於簽名檔說了些什麼呢? 這只需加密次冪 e 即可看出:

```
In[15]:= PowerMod[s, e, n]
Out[15]= 609181920001915122205180023091419001\
         514050008211404180504000415121201 1819

In[16]:= A["609181920001915122205180023091419001\
            514050008211404180504000415121201 1819"]
Out[16]= first solver wins one hundred dollars
```

[2]參考網頁 http://www.mit.edu:8001/people/warlord/home_page.html

2. 公鑰密碼的孕育

現在回到密碼之旅的靈魂人物: 三毛與四郎。三毛要傳送信息給四郎, 他們沒有事前的接觸, 也不希望花時間交信差遞送鑰匙。因此, 所有三毛送給四郎的信息都有可能被第三者五爺給攔截。在此種情況之下, 三毛是否有可能秘密傳送信息給四郎呢? 也就是說, 所傳送的信息只有四郎能閱讀但五爺卻不行。

在所有傳統方法中, 這是不可能的。因爲三毛必需透過公開的頻道將鑰匙遞送給四郎, 所以五爺可將此鑰匙攔截, 如此一來他就能閱讀所有未來三毛所傳送的信息。這說明了傳統密碼術的安全性完全仰賴於鑰匙的秘密性。如何打破這個僵局與困境呢?「解鈴還須繫鈴人」, 當然得從鑰匙這邊來動腦筋、想點子。傳統密碼術最大的致命傷在於其加密鑰匙和解密鑰匙是對稱的; 也就是說, 解密鑰匙很容易就可以從加密鑰匙推導出來, 甚至有時候更是單純到解密鑰匙就是加密鑰匙。如此看來, 突破之點就在於

打破加密鑰匙和解密鑰匙之間的對稱性;

也就是說, 即使給你加密的鑰匙, 你也沒有任何的法子可以計算出或得到解密的鑰匙。所以很自然的, 我們將過去的那些傳統密碼系統歸類爲對稱密碼系統(Symmetric Cryptosystems), 而尚未誕生的那些密碼系統就稱之爲非對稱密碼系統 (Asymmetric Cryptosystems)。

在思考這個問題時, 很難避免的我們會從鑰匙聯想到門。有許多公共建築物的大門, 當你從門內到門外只要將門一推即可, 毫無困難; 但反過來則否, 必須有鑰匙才能從門外回到建築物內。從門內將門一推, 表面上好像是不需鑰匙; 實際上那推的動作因爲每個人都知道, 可以看成是公開的鑰匙。門裡門外是全然不同的兩個世界。所以如何打破加密鑰匙與解密鑰匙之間的對稱性呢? 乍看之下似乎是不可能, 然而門的比喻帶給了我們些許的啓發與暗示。

出去簡單、容易、快速, 進來複雜、困難、緩慢;

亦即一個方向是簡、易、快而另一個方向則雜、難、慢。這樣子的東西到底是怎麼樣的一個東西呢? 是一個運算嗎? 是一個函數嗎? 是一個演算法嗎? 這整個探索過程的歷史是相當耐人尋味的; 我們現在就順著歷史的軌跡來訪古尋幽一番。

這其中最關鍵的人物是

費特費德·迪費(Whitfield Diffie),

迪費 1944 年 6 月 5 日生在華府 (Washington D.C.), 長在紐約市區。打從十歲起就已斷斷續續對密碼術產生了興趣, 這要歸功於小學老師瑪麗·柯林斯 (Mary Collins) 跟他們上了一天簡易密碼系統的課程後讓他興奮莫名, 回家後並要父親借來紐約城市學院圖書館所有跟密碼有關的兒童圖書; 甚至於連海倫·甘尼斯 (Helen Gaines) 的書「密碼分析[7] 」也想讀, 只可惜沒能讀懂。他在童年就迷上數學, 從化學橡膠公司數學手册到哈地的純數課程[9] , 幾乎只要跟數學有關的書他都讀。長大後, 先在柏克萊後進入麻省理工(M.I.T.) 主修數學。

迪費在 1965 年畢業之後隨即進入密特 (Mitre) 公司同卡爾·英格曼 (Carl Engelman) 工作, 研發那被稱爲 Mathlab 的符號數學演算系統, 後來在 M.I.T. 發展成數學套裝軟體 Macsyma。老闆告訴過他當時的密碼系統是如何產生鑰匙串並加密之過程, 這激發他開始思考密碼術裡面的東西。當時工作的環境是在 M.I.T. 的人工智慧實驗室, 必須藉著彼此的互信來保護資料; 所以他想密碼術應該跳出這個框框。當時他只懂數學而電腦卻一竅不通, 但工作需要逼他得進入系統程式並開始接觸編譯器, 從而得知程式正確性之證明的概念, 同時也意識到那可能是現代工程學裡頭最重要的問題, 因此他就開始考慮那問題。在當時約翰麥卡錫[3] (John McCarthy) 是唯一了解那問題之重要性的人, 剛好到了波士頓。迪費跟他談過之後就受聘轉到史丹佛大學的人工智慧實驗室同麥卡錫還有他的幾個研究生一起工作, 那是 1969 年 6 月下旬。年底麥卡錫寫完關於家庭終端機的論文, 其中並構想人們透過家庭終端機買賣交易之事, 因而促使迪費開始思考到我們現在稱爲數位簽名的問題。

工作的機緣加上老闆的要求, 一步步帶領著迪費在密碼術上作更多的思考並願意閱讀密碼術的古典文獻。1972 年秋, 他開始研讀歷史學家大衛·坎恩 (David Kahn) 的「破碼者[10]」。這是第一本詳細探討密碼發展史的書, 對初入門的密碼研究者而言, 是最佳的入門讀本。迪費閱讀的速度很慢, 一直到隔年春天, 除了搞密碼之外什麼事也沒做。他看麥卡錫的臉色不對, 於是就留職停薪離開史丹佛大學開始雲遊四方並思考一些密碼問題, 其中最主要的有兩個:

- 第一個是鑰匙的發送問題。若有兩個未曾謀面的人想藉著慣用的加密方式秘密互通信息的話, 他們必須事先同意所要使用的鑰匙, 這鑰匙只有他們兩個知道而別無他人。

- 第二個是簽名的問題, 顯然跟第一個風馬牛不相及。能否設計一方法提供給純數位信息的接收者透過某種管道讓他足以向別人證明所收到的信息確實來自某人, 如同此人親自在信息上簽上大名一般?

[3]目前網址爲 http://www-formal.stanford.edu/jmc/

這兩個問題看起來就是不可能達成的任務(Mission Impossible)。在第一種情況, 若未曾謀面的兩個人既然有辦法互通其秘密鑰匙, 那爲什麼他們沒有辦法秘密地彼此互通信息呢? 而第二種情況也不會更好。爲求有效, 簽名必須是難以複製。然而如何在一個可複製得跟原版完全一樣的數位信息當中簽上尊名呢?

在當時這兩個問題困擾迪費都好幾年了, 一個有八年、九年之久, 另一個也有四年、五年之久。1974 年夏季他大部份時間待在麻州劍橋並參加吉姆·芮茲[4] (Jim Reeds) 所主持的密碼術研討會。那時他們已發展出所謂的單向函數 (one-way function) 的觀念, 他們都在思考著並認爲大概很難建造出來; 也不確定, 果眞建造出來了, 又何用? 之間, 迪費受邀到紐約州約克城高地 (Yorktown Heights) 的 IBM 實驗室, 那兒擁有美國當年非官方最大的密碼研究群。設計資訊加密標準 DES(Data Encryption Standard) 的那批人馬就在這兒。他跟 IBM 資深密碼專家亞倫孔翰 (Allen Konheim) 聊, 而孔翰顯得神秘兮兮的什麼都不講。唯獨說了一件事, 之後他又但願他什麼都沒說。他說: 我的老朋友馬丁·黑爾曼 (Martin Hellman), 幾個月前在這裡講過鑰匙的發送問題, 他在史丹佛大學, 而兩個人一起研究問題比一個更好, 所以當你回到史丹佛, 你應該去看他。

所以就如此這般的在那一年秋高氣爽又雲淡風輕之際, 迪費再度雲遊回到史丹佛並打電話跟黑爾曼聯絡上。黑爾曼從未聽過他的名字, 勉爲其難同意在那天下午跟他談半個小時, 沒想到由辦公室到家裡直聊到半夜才走。黑爾曼驚覺到迪費是他所遇過見聞最廣博的人, 可惜沒有足夠的研究經費來聘任如此志同道合的朋友當研究員。只好退而求其次, 迪費申請成爲研究生並開始一個密碼術的研討會。隔年春天, 迪費北上去拜訪人在柏克萊的倫斯·霍夫曼[5] (Lance Hoffman), 因他博士論文搞的是電腦安全方面的東西, 他卻向迪費推薦他有個學生叫彼得·布雷特曼 (Peter Blatman) 的對密碼術有興趣。於是迪費找到布雷特曼, 談過後他也願意參與研討會。布雷特曼跟迪費提及瑞夫·默克 (Ralph Merkle) 說: 默克已在「如何與未見過面的人有安全的通訊」之問題上工作一陣子了。而這正是迪費久思未解如上所說的第一個問題, 而他也自我論證過此問題之不可解, 因此不僅要說服自己也要說服布雷特曼同意這樣的觀點。但他還是回頭再去思考這個問題, 默克在此扮演著一個極關鍵的角色。其實默克當時已經解出, 但他切入問題的方式與迪費迥然不同。默克那時選了霍夫曼一門課, 這門課要求每個人必須找一個學期報告的題目, 且學期初就得將大綱擬妥。默克提出鑰匙交換系統, 但霍夫曼看不懂, 發回叫他重寫。默克重寫之後, 霍夫曼還是沒看懂。如此兩個回合後, 結局是默克退選了那門課, 而霍夫曼則丟失了在一個偉大發現上掛上尊名的良機。否則我們看到的將是默克-霍夫曼鑰匙交換系

[4]當時在哈佛大學統計系攻讀博士, 目前網址 http://www.dtc.umn.edu/~reedsj/。

[5]目前在華府喬治華盛頓大學, 其網址爲 http://www.seas.gwu.edu/~lanceh/。

統, 而不是迪費-黑爾曼鑰匙交換 (Diffie-Hellman key exchange) 系統。雖然如此, 默克還是繼續思考那個問題, 最後終於解決了, 但完成後一直到論文登出卻歷時五年之久, 可謂好事多磨也[6] 。論文題之爲: 不安全頻道上的安全通訊[11]。

3. 公鑰密碼的誕生

那時迪費除了照顧、看守麥卡錫的房子外同時也繼續思考上面所提的兩個問題。一方面把他認爲很難的問題丟到他的密碼術問題聚寶盆裡等以後再想, 另一方面他想到利用一個稱爲 IFF 的古典密碼術問題來製造一個安全的密碼系統。IFF 就是 Identification Friend or Foe 的簡寫, 現代密碼學巨人霍斯特·費斯妥[7] (Horst Feistel) 用來衡量一個密碼系統強度的標準。其實這就是傳統密碼之旅 (上)[17]所提到的, 用選擇明文攻擊法來鑑定一部飛機是友是敵的實例。單向函數的觀念也湧現在他心頭, 忽然間好像數位簽名有眉目了。當你判斷是友是敵的時候, 你所比較的是加密過的信息, 你不用將密文破解。這似乎有一點認證的味道。所以, 你若擁有一個單向函數, 你要在一份文件上簽名; 你就用不公開的 (解密) 鑰匙拿來完成簽名的動作。任何其他的人, 可透過你公開的(加密) 鑰匙將其還原成先前的文件, 就知道這就是你簽名過的文件。

簽過名, 但不留一絲痕跡。

難以複製的部分彰顯在單向函數的逆方向, 而不是字面上複製的意思。反過來卻是一套天生的密碼系統; 因爲加密鑰匙是公開的, 所以沒有鑰匙發送的問題, 而且順理成章就稱之爲公鑰密碼系統。令人難以置信的是, 看似豪無關聯的兩個問題卻俯首稱臣於公鑰密碼系統之下, 可謂一箭雙鵰也。而這也解決了在迪費腦後搔癢了九年、十年的問題。這發生在 1975 年 5 月天的一個清晨。王荊公有云:

「看似平常最奇絕, 成如容易卻艱難。」

應可稍稍描繪出迪費當時的心情。那天迪費在家直等到太太工作回來才第一個告訴她, 然後走下坡去告訴黑爾曼且花了 45 分鐘的時間解釋並說服他相信這是眞的; 但由於沒有實例, 迪費只能試圖說服他應可以做出一些東西。剛好黑爾曼有 IEEE 的資訊理論期刊 Transactions 主編吉姆·麥錫 (Jim Massey) 的寫稿邀請函, 於是乎兩人就聯手在這主題上開始工作[4]。

接下來他們經過好幾個月孜孜矻矻的努力, 在年底12月整理出他們的結果寫好一篇論文, 最後在 1976 年的全國電腦會議上發表。他們廣泛寄出論文的預訂本 (preprint), 並給了布雷

[6]此論文之歷史, 見 http://www.itas.fzk.de/mahp/weber/merkle.htm#contents。

[7]見賽門辛的碼書[13, 15]pp. 248-250的介紹。

特曼一份, 再傳至默克手上。默克看了之後才恍然大悟, 原來伯樂在此, 馬上打電話給在柏克萊的迪費並寄了份他的論文給黑爾曼。黑爾曼再度發揮他高明偵察員的本色, 馬上嗅出默克是一個非常聰明的人。於是默克從柏克萊轉到史丹佛, 加入這個團隊並繼續攻讀博士學位。後來又有研究生史提夫·波立格 (Steve Pohlig) 也加入密碼術的工作行列。如此一來, 這個團隊搖身一變, 變成史丹佛活躍的四人小組, 繼續奔向前面的路程。那時期的結果綜括在論文密碼術的新方向[5]中, 請參考賽門·辛(Simon Singh) 的碼書[13]及 1992 年福蘭寇·富葛 (Franco Furger) 就公鑰密碼術的發展在 Palo Alto 訪談迪費的實錄[6]。

然而他們並沒有設計出一套實際可運作的公鑰密碼系統。之後幾年, 陸陸續續有好幾個可執行的方法被提了出來。其中最成功也最有名, 根基於分解因數的困難性, 由麻省理工學院(M.I.T.)的三位學者雷諾·瑞維斯特 (Ronald Rivest)、葉迪·沙密爾 (Adi Shamir) 及李歐那德·葉德曼 (Leonard Adleman) 於 1977 年 5 月所提出來的, 因而就稱爲瑞沙葉演算法 (RSA Algorithm)[12]。當迪費與黑爾曼撰寫密碼術的新方向時, 對這三個人是完全陌生的。迪費在史丹佛大學的人工智慧實驗室工作時, 瑞維斯特是那兒電腦系的研究生。迪費有一個同事叫駱哈·馬納(Zohar Manna) 不久後回以色列 Weitzman 學院指導沙密爾的博士研究。葉德曼則是道道地地的舊金山佬, 大學及研究所都在柏克萊。儘管有著如此近距離的連接網, 他們彼此之間的脚蹤卻從未交會過, 彼此的名字也不熟悉[4]。

4. 瑞沙葉演算法(RSA Algorithm)

瑞維斯特及沙密爾花了一年的時間, 不斷的提出新點子; 而葉德曼則花了一年的時間, 將這些新點子一一否決掉。這個三人小組是有點洩氣了。然而他們卻仍沒有意識到, 此種持續失敗的過程, 其實是研究工作必經歷的一部份。最後, 終於在 1977 年 4 月天的某一個夜晚, 因爲睡不著覺, 瑞維斯特躺在長沙發上翻閱一本數學教科書作水平思考。忽然間, 妙發靈機腦海中; 當下提筆抓住那瞬間寫下他的構想, 在天亮之前完成了一篇論文。這項突破誕生在瑞維斯特的腦中, 但孕育自他跟沙密爾及葉德曼這一年的合作, 缺少任何一個人, 不會有這項突破。因此寫好論文後, 按字母順序列上作者名字: 葉德曼、瑞維斯特、沙密爾 (Adleman, Rivest, Shamir)。

隔天一早, 瑞維斯特將完成的論文交給葉德曼, 而葉德曼也預備如往常一樣把它撕掉。但這次卻找不出一絲的破綻, 唯一可挑出的骨頭是作者欄上他的名字不應該擺上去, 因爲不是他想出的。但瑞維斯特沒答應, 他們討論了一陣子後, 同意葉德曼回家考慮一個晚上再說。第二天回來後, 葉德曼建議將他的名字擺在最後。所以變成 RSA(瑞沙葉), 而非原先的 ARS(葉瑞沙)。

瑞沙葉演算法 (RSA Algorithm) 的步驟[12]如下:

1. 四郎選取二相異大質數 p 和 q, 並將此二數相乘得 $n = pq$。
2. 然後選取加密次冪 e 使得 $\gcd(e, (p-1)(q-1)) = 1$, 將 (n, e) 經由公開的頻道告知三毛, 但 p 和 q 則保密。
3. 三毛將欲傳送的信息轉換成一個數 m, 如果 $m > n$ 三毛必需將 m 分割成區塊 $\{m_1, m_2, m_3, \ldots, m_k\}$, 使得所有的區塊 m_i 都小於 n。
4. 計算密文 $\{c_1, c_2, \ldots, c_k\}$ 如下: $c_i \equiv m_i^e\,,\ i = 1, 2, 3, \ldots, k \pmod{n}$, 然後將密文 $\{c_1, c_2, \ldots, c_k\}$ 傳送給四郎。
5. 因爲只有四郎知道 p 和 q, 所以他可以算出 $\varphi(n) = (p-1)(q-1)$。
6. 再透過延伸輾轉相除法求得解密鑰匙 d, 滿足 $de \equiv 1 \pmod{\varphi(n)}$。
7. 最後四郎將密文 c_i 取 d 次冪, 如此即可還原成明文並讀取此信息:

$$m_i \equiv c_i^d\,, \quad i = 1, 2, 3, \ldots, k \pmod{n}。$$

例題01: 四郎選取 $p = 336998653,\ q = 951871033$, 並將此兩數相乘得

$$n = pq = 320779255950718549。$$

```
In[17]:= p=336998653; q=951871033; n=p*q
Out[17]= 320779255950718549
```

再選取 $e = 83089$, 並將 (n, e) 傳送給三毛。假設三毛要傳送的信息就是 wonderful-weekend。首先我們將 a 用 01 取代, b 用 02取代, $\cdots$, z 用 26 取代。這與前面的習慣不一致, 主要是要避免信息以 a 起頭時的困境。如此我們可以得到明文 m 爲 231514040518062112230505110514O4, 再將 m 分割成兩個區塊 $\{m_1, m_2\}$。

```
In[18]:= e=83089;
In[19]:= abc=" abcdefghijklmnopqrstuvwxyz";
 no="000102030405060708091011121314151617181920212223242526";
         digitalize=Table[StringTake[abc,{i}]->StringTake[no,
                                        {2*i-1,2*i}],{i,1,27}];
               alphabetize=Table[StringTake[no,{2*i-1,2*i}]->
                              StringTake[abc,{i}],{i,1,27}];
```

```
Q[plaintext_]:=StringReplace[plaintext,digitalize];
Qd[plaintext_]:=ToExpression[Q[plaintext]];
A[digit_]:=StringReplace[digit, alphabetize];

In[26]:= won="wonderfulweekend";s=StringLength[won]; w=Q[won]
Out[26]= "23151404051806211223050511051404"

In[27]:= m=ToExpression[{StringTake[w, s], StringTake[w,-s]}]
Out[27]= {2315140405180621, 1223050511051404}
```

並在模 n 之下計算密文 $c = \{c_1, c_2\}$,

```
In[28]:= c=PowerMod[m,e,n]
Out[28]= {96135210630221818,78216213811502237}
```

然後將 c 傳送給四郎。因爲四郎知道 $\phi(n) = (p-1)(q-1)$, 所以利用延伸輾轉相除法求得解密鑰匙 $d = 138212005402570609$。

```
In[29]:= d=PowerMod[e,-1,EulerPhi[n]]
Out[29]= 138212005402570609
```

最後解密, 我們計算 $c^d \pmod{n}$ 得到

```
In[30]:= PowerMod[c,d,n]
Out[30]= {2315140405180621,1223050511051404}
```

這是否就是原先的信息呢? 請看:

```
In[31]:= PowerMod[c,d,n]== m
Out[31]= True
```

5. 公鑰密碼系統的另一章

長久以來有人聲稱政府部門的密碼局早先幾年就發現瑞沙葉演算法 (RSA Algorithm), 但保密的法規阻止他們透漏任何的證據。終於在 1997 年由英國密碼局 CESG[14]所發布的文件得知, 早在 1970 年時詹姆士·艾利斯[8] (James Ellis) 就已發現公鑰密碼術且在 1973 年克

[8]在 1982 年第一次遇見艾利斯時, 迪費認爲是眞實的, 但後來變得比較不相信。迪費說: 我曾跟他談過好幾個小時的話, 我不瞭解他的論文, 因無法說服我什麼。他說他有一個公鑰密碼系統的構想與我有同樣的形式, 但我從未發現任何具體的證據, 他或他的同事瞭解其意義。詳情可參考 Andy Coote 於 2004 年 7 月 8 日在 SC 雜誌 Features 專欄訪問迪費之報導: 密碼人[3](Crypto man)。

里佛德·寇克斯(Clifford Cocks) 寫過一份內部的文件描述瑞沙葉演算法 (RSA Algorithm) 的一個版本, 其中的加密次冪 e 與模 n 相同。

艾利斯是一位求知欲很強卻有一點反常的怪人。主要的興趣是科學, 到了帝國學院 (Imperial College) 攻讀物理。1965年 4月 1日加入 GCHQ 新成立的通訊電子安全組 CESG, 一個專門負責英國通訊安全的特別部門。他最顯著的特質之一是吸取知識的能力。任何他拿到的科學期刊, 他都會讀過一遍, 而且從不丟棄。在同事之間, 他擁有密碼宗師 (cryptoguru) 的名號。1969 年初, 軍方要求艾利斯尋找發送鑰匙的方法。1969年底, 艾利斯在理論上找到了可行的證據。但他還需要一個特別的單向函數來落實其構想。1970年開始接著三年, GCHQ 的頂尖頭腦都在努力的尋找可以滿足艾利斯要求的單向函數, 卻沒有任何斬獲。1973 年 9 月, 寇克斯加入這個小組, 他剛從劍橋大學畢業, 主攻數論, 是最純的數學領域之一。他加入 GCHQ 時, 對密碼學及軍事與外交通訊的世界沒有多少概念。因此他們指派尼克·貝德森 (Nick Patterson) 輔導他, 在他進入 GCHQ 的頭幾個星期給他一些指引。大約六星期後, 貝德森跟他講起那個「實在很古怪的點子」。他簡介了艾利斯的構想, 並解釋說還沒有人找出完全符合條件的數學函數。那天寇克斯坐下來工作時, 心想沒什麼特別的事, 不如來思索一下這個點子。他研究的領域一直是數論, 自然會想到單向函數就是只能演算出來不能逆算回去的東西。質數和分解因數是理所當然的對象, 於是就成了他的起點。寇克斯回憶說: 從開始到完成, 只花了我半小時的時間。我想, 嗯!不錯哦。人家給我一個問題, 我把它解決了。

6. 瑞沙葉密碼系統 (RSA Cryptosystem)

現在回到瑞沙葉密碼系統, 有幾個方面需要解釋的。

- 最重要的可能是

$$\boxed{\text{爲何 } m_i \equiv c_i^d \pmod{n} \text{ 呢?}}$$

記得嗎? 歐拉定理說: 若 a 與 n 互質, 則 $a^{\varphi(n)} \equiv 1 \pmod{n}$。在目前的例子中, 我們有 $\varphi(n) = \varphi(p\,q) = (p-1)(q-1)$。假設 $\gcd(m_i, n) = 1$。這種情形可能性非常高, 因爲 p 跟 q 都很大, m_i 就可能不會有 p 或 q 爲其因子。由於 $de \equiv 1 \pmod{\varphi(n)}$, 必存在一整數 k 使得 $de = 1 + k\varphi(n)$, 故

$$c_i^d \equiv (m_i^e)^d \equiv m_i^{1+k\varphi(n)} \equiv m_i \cdot (m_i^{\varphi(n)})^k \equiv m_i \cdot 1^k \equiv m_i \pmod{n}。$$

因此我們已證明四郎能將密文還原回明文。假設 $\gcd(m_i, n) \neq 1$, 四郎依舊能將密文還原回明文[9] 。

- 五爺能做些什麼呢? 他能從公開的頻道上攔截到 c, n, e, 但卻不知道 p, q, d。假設五爺沒有辦法分解 n。而如果五爺想將密文還原成明文的話則必須求 d, 因此五爺必須知道 $\varphi(n)$。底下我們會說明尋找 $\varphi(n)$ 和尋找 p, q 其實是等價的。是否有其他的方法可以算出 d 呢? 下面我們會證明如果五爺能找到 d, 他就有可能可以將 n 分解。因此之故, 五爺能找到 d 的機會是微乎其微的。

- 五爺知道 $c \equiv m^e \pmod{n}$, 爲什麼我們不就取 c 的 e 次方根呢? 倘若我們不是在模 n 之下, 那就毫無困難, 但在我們的情況下卻是艱難無比。例如我們知道 $m^3 \equiv 3 \pmod{85}$, 但你無法從你的計算器算出 3 的立方根 1.2599..., 然後再化簡至模 85 之下。當然你可以一個一個去找, 最後終於找到 $m = 7$, 但對大 n 來說, 此法是行不通的。

- 四郎如何選取兩個相異的大質數 p和 q呢? 當然要隨機的選, 而且彼此不相干。至於多大, 就得看你所要求的安全等級的高低而定, 但似乎是至少要 100位數才夠。這兩個數的大小也是要維持一定的距離, 理由我們下面會討論到。當我們討論到質數判別法之時, 你會發現尋找此種質數相當的快。但也需用到一些其他的判別法來確定所選的 p或 q不是壞的質數。例如, 若 $p-1$只有小的質因子的話, 則用所謂的 $p-1$因數分解的方法來分解 n是容易的, 所以在此種情況之下應該選取另外的質數來代替這個質數。

- 爲何四郎所選取的 e 必須滿足 $\gcd(e, (p-1)(q-1)) = 1$ 呢? 這是因爲

$$de \equiv 1 \pmod{(p-1)(q-1)} \text{ 有一解 } d \Longleftrightarrow \gcd(e, (p-1)(q-1)) = 1。$$

因此, 這個條件對 d 的存在是必要的。用延伸輾轉相除法來計算 d 值相當的快。因爲 $p-1$ 是偶數, e 不能選 2, 你可能很容易會去選 $e = 3$。然而, 選取小的 e值有它的危險在, 所以選大一些的 e 值是一般性的要求。例如, 你可選 e爲中等大小的質數。如此應該毫無攔阻的可確定 $\gcd(e, (p-1)(q-1)) = 1$。

- 在加密過程中, 三毛在計算 $m^e \pmod{n}$時可利用如連續平方法, 這不需要用到太大的記憶體即可完成。這的確是模算術的一大優點: 倘若三毛試著先計算 m^e, 然後再簡化至模

[9]如果 $\gcd(m_i, n) \neq 1$, 則 $\gcd(m_i, n) = p$ 或 $q \Leftrightarrow p \mid m_i$ 或 $q \mid m_i$。因 $m_i < n$, 所以只有其中之一會發生, 說是前者 $p \mid m_i$ 。則在模 q 之下, 我們有

$$m_i^{q-1} \equiv 1 \Rightarrow m_i^{\varphi(n)} \equiv (m_i^{q-1})^{p-1} \equiv 1 \Rightarrow m_i^{j\varphi(n)} \equiv 1 \Rightarrow m_i^{ed} \equiv m_i \pmod{q},$$

但 $p \mid m_i$, 在模 p 之下, 也有 $m_i^{ed} \equiv m_i \pmod{p}$。因此, 我們有 $m_i^{ed} \equiv m_i \pmod{n}$。

n之下; 如此一來, 有可能在紀錄 m^e時, 她電腦的記憶體就溢位了。同樣地, 解密過程中計算 $c^d \pmod n$可很有效率的完成。因此, 所有加密、解密需要的運算都可以快速地完成(即在 $\log n$的冪次時間之內)。其安全性完全建立在 n不可能被分解的假設上。

7. 兩個有待證明的命題

現在回到上面提過的兩個有待證明的命題。記得嗎? 這兩個命題最主要的觀點簡單說就是

> 尋找 $\varphi(n)$ 或尋找解密次冪 d 本質上與分解 n 是一樣困難的。

- 先說明尋找 $\varphi(n)$ 和尋找 p, q 其實是等價的。假設 $n = pq$ 爲兩個相異質數的乘積。如果我們知道 $n, \varphi(n)$ 則

$$n - \varphi(n) + 1 = pq - (p-1)(q-1) + 1 = p + q,$$

因此我們就能知道 pq 和 $p + q$ 。接著再用一元二次方程式根與係數的關係, 我們可以得到

$$X^2 - (n - \varphi(n) + 1)X + n = X^2 - (p+q)X + pq,$$

因此我們有

$$p,\ q = \frac{(n - \varphi(n) + 1) \pm \sqrt{(n - \varphi(n) + 1)^2 - 4n}}{2}。$$

例題02: 令 $n = 851$。假設我們已經知道 $\varphi(n) = 792$, 試分解 n。

解: 考慮二次方程式 $x^2 - (n - \varphi(n) + 1)x + n = X^2 - 60X + 851$。
解出得到 $n = 23 \times 37$, 因爲其根爲 $\dfrac{60 \pm \sqrt{60^2 - 4 \cdot 851}}{2} = 23,\ 37$ 或是

```
In[32]:= n=851;phin=792;Roots[x^2-(n-phin+1)x+n==0,x]
Out[32]= x==23 || x==37
```

例題03: 令 $n = 11313771275590312567$。假設我們已經知道 $\varphi(n)$ 之值爲 11313771187608744400, 試分解 n。

解: 考慮二次方程式 $x^2 - (n - \varphi(n) + 1)x + n$。解出其根爲

```
In[33]:= n=11313771275590312567;phin=11313771187608744400;
         Roots[x^2-(n-phin+1)x+n==0,x]
Out[34]= x==128781017||x==87852787151
```

所以 $n = 128781017 \times 87852787151$。

- 其次我們證明: 若已知 d與 e, 則我們有可能將 n分解。這裡我們要用到的分解因數法就是所謂的統一冪次分解法, 茲敍述如下 (證明省略, 因離主題甚遠) : 對所有與 n互質的 a, 若我們有統一冪次 b使得 $a^b \equiv 1 \pmod{n}$, 則我們有可能將 n分解。因爲 $de-1$ 爲 $\varphi(n)$ 的倍數, 所以 $de - 1 = k\varphi(n)$ 且對所有與 n 互質的 a, 我們有

$$a^{de-1} \equiv (a^{\varphi(n)})^k \equiv 1 \pmod{n}。$$

因此統一冪次分解法就可派用上場了。

何時用瑞沙葉演算法 (RSA Algorithm)?

當有好幾家銀行要彼此傳送財務資料時, 可試著來使用瑞沙葉演算法 (RSA Algorithm)。倘若有好幾千家的話, 而每兩家就需要有一把鑰匙來作爲彼此秘密通訊之用, 那就太不實際了。更好的一個方式爲: 每家銀行選取一個整數對 (n, e)如上。然後將這些資料出版成爲公共的書籍如電話簿一樣或者印在銀行的名片及廣告上。假設甲銀行要送資料給乙銀行。那麼甲銀行就查出乙銀行的公鑰 (n, e) 並用此來傳送資料。實際上, 在傳送大量資料時, 瑞沙葉演算法 (RSA Algorithm) 的加密速度是不夠快的。因此, 瑞沙葉演算法 (RSA Algorithm) 經常拿來傳遞更快速加密法如 DES 的鑰匙之用。

參考文獻

1. Atkins D., et al.: The Magic Words are Squeamish Ossifrage, *American Scientist*, Vol. 82, No. 4, July-August, 1994, pp. 312–316. Postscript Version of Paper-rsa129.ps.gz
2. Atkins, D./Graff, M./Lenstra, A./Leyland, P. : The Magic Words are Squeamish Ossifrage, *Lecture Notes in Computer Science*, 917, Springer, 1995, pp. 263–277. (Advances in Cryptology–ASIACRYPT '94)
3. Coote, Andy: Crypto man, *SC Magazine*, Thu, Jul 8, 2004. http://www.scmagazine.com/features/index.cfm?fuseaction=FeatureDetails&newsUID=a4b8fe9a-34b9-4a2b-bde4-9a2042176851&newsType=Latest+Issue
4. Diffie, Whitfield: The First Ten Years of Public-Key Cryptography, Proceedings of the IEEE, vol. 76, no. 5, May 1988, pp: 560-577. http://cr.yp.to/bib/1988/diffie.pdf
5. Diffie, Whitfield/Hellman, Martin E.: New Directions in Cryptography, *IEEE Trans. Information Theory*, Vol. IT-22, (Nov. 1976), pp. 644–654.

6. Furger, Franco: *Interview with Whitfield Diffie on the Development of Public Key Cryptography*, Conducted by Franco Furger in Palo Alto, 1992.
http://www.itas.fzk.de/mahp/weber/diffie.htm
7. Gaines, Helen Fouche: *Cryptanalysis*, Dover, New York, 1956.
8. Gardner, M.: Mathematical Games, A new kind of cipher that would take millions of years to break, *Scientific American*, (Aug. 1977), pp. 120-124.
http://www.fortunecity.com/emachines/e11/86/cipher1.html
9. Hardy, G.H.: *A Course of Pure Mathematics*, Cambridge Mathematical Library, 1993 (First published in 1908).
10. Kahn, David: The Codebreakers, The Story of Secret Writing, Scribner, Revised and Updated, 1996.
11. Merkle, Ralph C.: Secure communications over insecure channels, CACM April 1978, 294-299. Submitted in 1975.
12. Rivest, R.L., Shamir A., and Adleman L.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, *Communications of the ACM*, **21**(1978), 120–126.
13. Singh, Simon: The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography, Anchor Books, New York, 2000. 中文譯本見[15]
14. Wayner, Peter: British Document Outlines Early Encryption Discovery, New York Times, December 24, 1997.
`http://www.nytimes.com/library/cyber/week/122497encrypt.html#1`
15. 劉燕芬譯, 碼書: 編碼與解碼的戰爭 (The Code Book), 台灣商務出版社, 89年9月。原文見[13]
16. 沈淵源, 數論輕鬆遊, 數學傳播第二十九卷第四期(116), 94年12月, 第45-71頁。全文見網頁
`http://www.math.sinica.edu.tw/math_media/d294/29408.pdf`
17. 沈淵源, 傳統密碼之旅 (上), 數學傳播第三十卷第一期 (117), 95年3月, 第61-80頁。全文見網頁
`http://www.math.sinica.edu.tw/math_media/d301/30108.pdf`
18. 沈淵源, 傳統密碼之旅 (下), 數學傳播第三十卷第二期 (118), 95年6月, 第55-76頁。全文見網頁
`http://www.math.sinica.edu.tw/math_media/d302/30206.pdf`

—本文作者任教私立東海大學數學系—