密碼學與模算術

鄭振牟／國立臺灣大學電機工程學系教授

密碼學

密碼學是資訊安全的基石，在資訊網路技術日漸普及的現代社會中，密碼學在維持社會順利運作上，扮演著極為重要的角色。沒有密碼學，我們就無法在網路上安全地進行交易，銀行金融系統也無法正常運作。

密碼學中一個重要的問題是如何達成祕密通信，這也是古典密碼學中最重要的課題。在這個問題中，密碼學家喜歡用兩位同學作例子，通常是 Alice 跟 Bob，他們想要利用公開的通信管道，交換一些私密的信息。密碼學家雖然不見得都憤世嫉俗，但他們往往傾向為最壞的情況作打算，比如說在這個例子裡面，密碼學家就會假設壞人，通常叫做 Eve，一定能夠偷看到所有 Alice 和 Bob 之間交換的信息。在這樣的狀況下，Alice 和 Bob 必需使用一些「**特殊的編碼方式**」，讓 Eve 就算看到中間交換的信息，也無法推斷出 Alice 和 Bob 究竟想要告訴對方什麼消息。用密碼學的行話來說，Alice 和 Bob 真正想要傳達給對方的消息叫做明文（plaintext），而經過特殊編碼之後，在公開通信管道上流通的那個信息就叫做密文（ciphertext）；這個特殊編碼的過程叫做加密（encryption），之後還原明文的過程就叫做解密（decryption），整套特殊編碼的方法則叫做加密法（cipher）。

大家可以想像，這樣的技術在傳遞軍事情報上，有著非常大的價值，而古典密碼學的發展，的確至少可以追溯到古羅馬偉大的將領凱撒（Julius Caesar）所使用的凱撒加密法（Caesar cipher）；在二次大戰時，盟軍也藉由破解了德軍所使用的 Enigma 加密機，以及日軍所使用的乙式暗號機，攔截到許多重要的軍事情報，例如因此擊落了日本海軍聯合艦隊司令官山本五十六的座機。值得一提的是，當時參與 Enigma 密碼機破解工作的主要人員，其中有後來被尊稱為電腦科學之父的圖靈（Alan Turing），後來人們為了紀念他對電腦科學的貢獻，遂將電腦科學界最重要的獎項命名為圖靈獎（Turing award）。

臺灣大學電機工程學系

10617 台北市 大安區 羅斯福路四段 一號

Email: dept@cc.ee.ntu.edu.tw

http://www.ee.ntu.edu.tw/

在二次大戰前所用的加密法或密碼機，其運作原理往往被視為最高機密，絕對不能公諸於世。但是二次大戰中密碼破解的經驗告訴我們，這些運作原理很難保持機密，例如圖靈等人藉由逆向工程，研究被盟軍俘獲的 Enigma 密碼機，很快就充分掌握它所使用的加密法運作原理。因此，現代密碼學傾向於將加密法運作原理公諸於世，廣邀全世界密碼破解高手來攻擊，希望集結這些高手的經驗和智慧，找出可能的漏洞並加以修補，最後產生出來的加密法，通常會比沒有經過這個過程的加密法，要來的難以破解；正所謂三個臭皮匠，勝過一個諸葛亮。因此，在運用現代加密法的時候，Alice 和 Bob 必須要有一些共同的祕密是 Eve 所不知道的，這個共同的祕密就被稱作金鑰（[cryptographic] key）。換言之，現代密碼學假設加密法的運作原理是公開的，加密法的使用者只需要保證金鑰的安全即可。

公開金鑰密碼系統

密碼學在二次大戰後有了突破性的發展，其中最重要的發明，首推公開金鑰密碼系統（public-key cryptosystem）。在古典密碼學中，Alice 和 Bob 的通信方式可以用這樣的類比來描述。Alice 將她的明文鎖在一個保險箱裡，然後將這個保險箱寄給 Bob。假設這個保險箱是非常堅固的，所以沒有鑰匙的 Eve 是沒有辦法打開它的。Alice 事先已經和 Bob 各打了一只鑰匙，所以當 Bob 收到保險箱時，他可以用這把共同的鑰匙，打開保險箱並取出明文。很簡單，不是嗎？

可惜這樣的方法，有一些不太方便的地方。首先，Alice 必須和 Bob 先打好這把共同的鑰匙。當 Alice 和 Bob 已經分隔兩地之後，這件工作將變得十分困難。其次，如果有 N 個人想要用這個方法，兩兩進行祕密通信，那我們將會需要 $\binom{N}{2} \approx N^2$ 這麼多把不同的金鑰；例如說，假設 N 是台灣的人口，那我們將會需要約 265 兆把不同的金鑰，顯然不太經濟！

因此，為了改進這些不方便的地方，Diffie 和 Hellman 兩位密碼學家，在 1976 年提出一種新的想法，稱之為公開金鑰密碼系統 [1]。在這樣的密碼系統裡面，每個人都有一把屬於他自己的公開金鑰（public key）和對應的私鑰（private key）；公開金鑰是每個人都知道的，而對應的私鑰只有金鑰的主人才知道。公開金鑰密碼系統的運作原理保證，利用某一把公開金鑰加密過後的密文，只有對應的私鑰才能解密而得到明文。如此一來，N 個人只需要 N 對不同的金鑰對，就能達到兩兩祕密通信的目標。利用前面所提到的類比，就好像 Bob 先把很多扣鎖分給大家，這些扣鎖不需要鑰匙就可以鎖上；當 Alice 想要送信息給 Bob 的時候，她可以把明文放進保險箱裡，然後用 Bob 之前給她的扣鎖把保險箱鎖起來後寄給 Bob。假設這個保險箱和扣鎖是非常堅固的，那沒有鑰匙的 Eve 當然沒辦法打開它，因此達到了祕密通信的目的。

讀到這裡也許你會抗議，那 Bob 怎麼把扣鎖分給大家？難道他不用偷偷地和 Alice 先見面嗎？如果我們假設 Eve 只會偷看公開通信管道上流通的信息，那 Bob 大可把扣鎖寄給大家，反正 Eve 看了扣鎖也沒辦法造出對應的鑰匙。但是如果我們假設這個 Eve 神通廣大，她可以假冒 Bob，寄一個她有對應鑰匙的扣鎖給 Alice，並且讓 Alice 相信這個扣鎖就是 Bob 的扣鎖。在這種情況下，問題會變得比較複雜，通常我們需要用某種公開金鑰密碼基礎建設（PKI: public-key infrastructure）來解決，限於篇幅我們在這裡沒辦法詳細解釋，有興趣的同學可以來台大電機旁聽密碼學的課程，在裡面我們會詳加介紹。

RSA 公開金鑰密碼系統

公開金鑰密碼系統聽起來很不錯，但是它們真的存在嗎？這個問題很快就被三位密碼學家 Rivest、Shamir、以及 Adleman 解決了，他們在 1978 年提出一個後來被稱為 RSA 的公開金鑰密碼系統 [2]。這裡我們先舉一個簡單的例子來說明 RSA 的運作原理。

假設某天晚上 Alice 和 Bob 一起去聽演唱會，Bob 在鬧烘烘的會場裡趁亂向 Alice 告白。這兩位同學恰好都在台大電機系聽過密碼學這門課，所以他們約定好，Alice 將在第二天，把她的答案用 Bob 的 RSA 公開金鑰加密後，寫在教室的黑板上，以避免讓不相干的 Eve 同學得到這個八卦消息。

讀到這裡，也許你又會笑這兩位同學：Alice 的答案不是 1（yes）就是 0（no），而 Bob 的公開金鑰又是全世界都知道的，那 Eve 只要把這兩種答案都代進去對答案，不就知道 Alice 的回答了嗎？這的確是個重要的問題，在密碼學上，最早由 Goldwasser 和 Micali 這兩位密碼學家，在 1982 年的一篇文章裡進行了深入的探討，並提出「語意安全（semantic security）」這個概念 [3]。在這裡我們（再度）限於篇幅而無法詳細解釋，但是我們可以提出一個簡單的方法來解決這個問題，以達成語意安全：Alice 和 Bob 可以先約定好，把 Alice 的答案加上一個相當大的隨機偶數，這樣當 Bob 把密文解開後，奇數的明文就代表 1（yes），而偶數的明文就代表 0（no）。如此一來，因為 Eve 不知道 Alice 選的隨機亂數，因此她就沒辦法輕易地用代答案的方式，猜出 Alice 的回答。

在仔細思考了一個晚上之後，很不幸的，Alice 決定要發給 Bob 好人卡。在這個例子裡面，我們用一個很小的 RSA 密碼系統，讓大家可以很容易算出答案。Bob 的 RSA 公開金鑰是兩個數字，比如說，讓我們把它們叫做 $N = 33$ 和 $e = 7$。假設 Alice 選的隨機亂數是 4，那麼明文也會是 $m = 4$（因為她決定要發卡給 Bob），而 Alice 可以用公式 (1)，將她的回答加密成密文 c：

$$c = m^e \bmod N = 4^7 \bmod 33 = 16 \tag{1}$$

另一方面，Bob 對應的私鑰是一個數字 $d = 3$，而解密的過程則如公式 (2) 所示：

$$m = c^d \bmod N = 16^3 \bmod 33 = 4 \tag{2}$$

藉此，可憐的 Bob 同學可以很快地由公開的通信管道，收到了 Alice 所發出來的好人卡。值得欣慰的是，由於 RSA 密碼系統的幫忙，在旁邊看熱鬧的同學，並不能從 Alice 寫在黑板上的密文，得到任何八卦消息。（當然，這並不能阻止他們從 Bob 臉上的表情得到些蛛絲馬跡。）

同餘關係和費馬小定理

為什麼我們可以確定，在經過 RSA 解密的過程後，我們會得到原本的明文？要回答這個問題，我們必須先了解整數的同餘關係（congruence relation）。如果兩個整數 a 和 b 的差能夠被 N 整除，那我們說 a 和 b 在模 N 下是同餘的，記做 $a = b \bmod N$。

同餘關係是一種等價關係（equivalence relation）。什麼是等價關係呢？如果一個關係 R 滿足反身性（reflexivity）、對稱性（symmetry）、和遞移性（transitivity），那我們就說 R 是一個等價關係。一個集合上的任一個等價關係，會把這個集合劃分成一些等價類（equivalence classes）；任何一個元素只會屬於一個等價類，而每一個等價類裡面所有的元素，彼此之間都有等價關係。所以，如果 $a = b \bmod N$，那 a 和 b 就屬於同一個等價類，我們說 a 和 b 是這個等價類兩個不同的代表；而整數在模 N 的同餘關係下，被分成了 N 個等價類，一般用 0 到 N–1 這 N 個整數作為各自最簡單明瞭的代表。

除了把整數分成 N 個井水不犯河水的等價類以外，同餘關係也和普通的整數加法和乘法相容，意思是說，對任何整數 a_1、a_2、b_1、和 b_2，如果 $a_1 = a_2 \bmod N$，而且 $b_1 = b_2 \bmod N$，則 $a_1 + b_1 = a_2 + b_2 \bmod N$，且 $a_1 b_1 = a_2 b_2 \bmod N$。我們也可以把這些模運算（modular arithmetic），想成是這些等價類之間的運算，而這個運算的結果，跟每個等價類為了進行運算所挑出的代表無關。換言之，在上面的這個例子中，如果 A 是含有 a_1 的等價類，而 B 是含有 b_1 的等價類，那麼我們真正想要得到的是，A 和 B 的「**和等價類**」$A + B$ 以及「**乘積等價類**」AB，使得：一、這些等價類之間的「**加法**」和「**乘法**」的計算，是由挑選任意的代表進行整數運算所得到的；二、這些問題的答案，不會因為我們在計算中，選擇不同的等價類代表而改變。因此，在以下的討論中，我們將會把一個整數，和它所在的等價類，互相混淆而不特別加以區分。

讓我們考慮所有和 N 互質的等價類 x 和 y，亦即 $\gcd(x, N) = 1$，以及 $\gcd(y, N) = 1$。首先，xy 這個等價類一定會和 N 互質，亦即 $\gcd(xy, N)$。其次，對任何一個和 N 互質的 x，我們可以利用推廣的歐幾里得演算法

（extended euclidean algorithm，其實就是一種輾轉相除法），得到一組整數 a 和 b，使得 $ax + bN = 1$，亦即 $ax = 1 \bmod N$。這告訴我們，a 所在的等價類，和 x 所在的等價類，乘起來就是 1 所在的等價類；如果有另外一組 a' 和 b' 使得 $a'x + b'N = 1$，那我們很容易可以得到，a 和 a' 一定會在同一個等價類裡面。因此，我們可以定義含有 x 的等價類，它的乘法反元素就是含有 a 的等價類，亦即 $x^{-1} = a \bmod N$；同時，對任何和 N 互質的 x 和 y，$f_x(y) = xy \bmod N$ 會是一個映成函數（bijection）。

有了同餘關係的基礎後，我們可以證明**引理 1**。

> **引理 1** 假設 N 是一個大於 1 的正整數，而 x 是一個和 N 互質的正整數，則
>
> $$x^{\phi(N)} = 1 \bmod N,$$
>
> 其中 $\phi(N)$ 是歐拉函數（*Euler's totient function*），$\phi(N)$ 等於與 N 互質而又小於 N 的正整數的個數。

引理 1 在文獻中，一般被稱作是費馬小定理的一個推廣，而在原本的費馬小定理中，我們僅考慮 N 是質數的情況，亦即 $x^{p-1} = 1 \bmod p$。**引理 1** 有許多不同的證明，在這裡我們考慮一個較簡單的方式。對任何一個和 N 互質的正整數 x，我們首先觀察所有和 N 互質的等價類所形成的集合 Y，然後考慮一個 Y 到 Y 的函數 f_x，它對任一個 Y 的元素 y 的作用，就是把它乘上 x。我們剛剛在前一個段落裡已經證明了，這個函數 f_x 是一個映成函數，而它的定義域是有限的（$|Y| = \phi(N)$），所以這表示 $Y = \{y \mid 1 \le y < N, \gcd(y, N) = 1\}$ 和 $xY = f_x(Y) = \{xy \bmod N \mid 1 \le y < N, \gcd(y, N) = 1\}$ 這兩個集合其實是同一個集合，只不過我們在表示時候，也許用了不同的順序，這在集合的定義上，是不影響兩個集合是否相等的關係。因此，當我們把兩個集合裡面所有的元素分別乘起來，我們將會得到等式 (3)：

$$\prod_{y \in Y} y = \prod_{y \in Y} (yx) = \left(\prod_{y \in Y} y \right) x^{\phi(N)} \pmod N \tag{3}$$

很明顯地，等號左邊這個數會和 N 互質，所以我們可以在等號兩邊，同時乘上它的乘法反元素，完成**引理 1** 的證明。

RSA 模數與質因數分解

利用**引理 1**，我們現在可以來解釋，在經過 RSA 解密的過程後，為什麼我們會得到原本的明文。由公式 (1) 和 (2) 可知，如果我們能夠讓 $ed = 1 \bmod \phi(N)$，那麼我們不難得到，對所有和 N 互質的 m，我們都有 $m^{ed} = m^{1+n\phi(N)} = m(m^n)^{\phi(N)} = m \pmod N$。所以，在決定了模數 N 之後，我們只要挑個跟它互質的 e，然後再次利用推廣的歐幾里得演算法，就可以得到對應的私鑰 d，使得 $de + k\phi(N) = 1$，亦即 $de = 1 \bmod \phi(N)$。

那麼，我們怎麼決定要用什麼樣的模數？從以上的討論，我們不難發現，如果我們可以得到模數 N 的質因數分解，那麼我們就可以由公式 (4) 計算出 $\phi(N)$：

$$\phi(N) = N \prod_{p \mid N} \left(1 - \frac{1}{p} \right) \tag{4}$$

進而利用推廣的歐幾里得演算法，從公開金鑰 e，得到對應的私鑰 d。因此，RSA 密碼系統的安全性，被認為和質因數分解的困難度，有著高度的相關。很明顯地，如果質因數分解是件簡單的工作，那麼破解 RSA 密碼系統就變得簡單；不過到目前為止，我們仍然不知道這是不是最好的方法，所以即使質因數分解是件困難的工作，所以我們仍然不知道，是否 RSA 就可以高枕無憂。

要找出模數 N 所包含較小的質因數，是件相當容易的工作。因此，RSA 密碼系統使用一種特殊的模數，稱為「**RSA 數**」，是由兩個差不多大的質數相乘起來所得到的。比如說，當各位看到「**2048-bit RSA**」，它的意思是說，模數 $N \approx 2^{2048}$，而這個 N 是由兩個大小相若的質數 p 和 q 乘起來所得到，所以 $p \approx q \approx 2^{1024}$。目前分解 RSA 數的紀錄，是由一個（主要在歐洲）的密碼學家團隊，在 2009 年底創下的，他們花了超過兩年的時間，利用數百台個人電腦，成功地分解了一個 232 位的 RSA 數（768 bits）[5]。

模算術與蒙哥馬利模餘法

除了讓 Alice 同學可以運用公開的通信管道，祕密地發好人卡給 Bob 同學以外，RSA 密碼系統其實還有一些其他的用途，尤其在銀行金融系統以及電子商務等應用上面，RSA 是目前最為廣泛使用的公開金鑰密碼系統。因此，如何在像金融提款卡這樣的裝置上面，能夠很快地進行 RSA 運算，就變成一個非常重要的課題。如同公式 (1) 和 (2) 所示，RSA 的核心運算是模指數運算（modular exponentiation）。小學時學習長除法的經驗告訴我們，與乘法相較之下，整數除法的複雜度要高的多，而包括模指數運算在內的模算術，似乎都需要整數除法作為基礎。

令人驚訝的是，如果是在同一個模數下進行多次模算術，則我們只需要在開始的時候做一些長除法，而不需要在每一步都進行長除法。這個著名的方法，是由密碼學家 Peter Montgomery 在 1985 年提出的，現在廣泛地被使用在各式各樣需要模算術的地方，其中關鍵的一步，一般稱之為蒙哥馬利模餘法（Montgomery reduction）。

在模算術中，我們真正關心的，是兩個在模 N 下的等價類 a 和 b，它們相加跟相乘的結果，是哪一個等價類。在我們前面的論述中，我們總是選擇某個被包含在該等價類中的元素作為代表；因為如此一來，同餘關係和整數加法和乘法就會相容。這種選擇主要的問題，在於利用整數加法或乘法計算出結果之後，我們需要將結果除以 N 求餘數，使得我們能夠回到 0 到 $N-1$ 這些最簡單明瞭的代表。但是，如果我們用不在該等價類中的元素作為代表，會不會有其它的好處？

蒙哥馬利模餘法的主要概念是這樣的。首先，我們先選擇一個大於 N 且和 N 互質的 R，使得除以 R 和模 R 的運算是非常容易的。在電腦上，我們一般會選 2 的某一個次方。其次，我們用 $Rx \bmod N$ 來代表在模 N 下包含 x 的這個等價類。由於 R 和 N 互質，乘以 R 這個運算是可逆的，因此不同的等價類，就會有不同的代表。對任何兩個等價類 a 和 b 來說，$Ra + Rb = R(a + b) \pmod N$。也就是說，在模加法中我們可以直接用 Ra 來代替 a。可是，對於模乘法來說，$(Ra)(Rb) = R^2(ab) \neq R(ab) = \pmod N$。因此，在這裡我們需要把結果除以 R，這樣才能得回正確的代表。

怎麼除以 R 呢？記得這些數字其實都是模 N 下某個等價類的代表，所以我們可以把它加上任何一個 N 的倍數，都不會改變所在的等價類。所以，在算出 $T = (Ra \bmod N)(Rb \bmod N)$ 之後，我們的目標就是要找一個 x，使得 $T + xN$ 能夠被 R 整除。怎麼找出這個 x 呢？我們可以先算好 $0 < R^{-1} < N$，以及 $0 < N' < R$，使得 $R^{-1}R - N'N = 1$，亦即 $R^{-1}R = 1 \bmod N$，以及 $N' = -N^{-1} \bmod R$。現在，我們要解 $T + xN = 0 \bmod R$ 這個方程式，也就是說，$x = -N^{-1}T = N'T \pmod R$。由此，我們可以得到：

$$R(ab) = \frac{T + (N'T \bmod R)N}{R} \pmod N \tag{5}$$

因為 $N < R$，所以 $T < N^2 < R^2$，而我們可以檢查，在公式 (5) 裡面，等號右邊這個數的範圍，一定會落在 0 到 $2N$ 之間，所以模 N 會變得很簡單，如果發現它比 N 大，那就把 N 減去，就會得到最簡單明瞭的代表。

最後，我們用 $R = 100$ 作例子，利用蒙哥馬利模餘法，來進行 Bob 的解密運算，體會一下接到好人卡時，Bob 所需要做的計算。

1. Bob 收到 $c = 16$，他知道 $N = 33$，$d = 3$
2. $N' = -N^{-1} \bmod R = 3$
3. $Rc \bmod N = 1600 \bmod 33 = 16$
4. $T_1 = (Rc \bmod N)^2 = 256$
5. $x_1 = N'T_1 \bmod R = (T_1 \bmod R)N' \bmod R = 56 \cdot 3 \bmod 100 = 68$
6. $Rc^2 \bmod N = R^{-1}T_1 \bmod N = (T_1 + x_1N)/R = (256 + 68 \cdot 33)/100 = 25$
7. $T_2 = (Rc^2 \bmod N)(Rc \bmod N) = 400$
8. $x_2 = N'T_2 \bmod R = 0$

9. $Rc^3 \bmod N = R^{-1}T_2 \bmod N = (T_2 + x_2N)/R = 4$

10. $x = N'(Rc^3 \bmod N) \bmod R = 12$

11. $m = c^3 \bmod N = R^{-1}(Rc^3 \bmod N) \bmod N = ((Rc^3 \bmod N) + xN)/R = 4$

12. 4 mod 2 = 0：由此 Bob 同學終於瞭解到，他又被發了一張好人卡

References

1. W. Diffie and M. E. Hellman, “New directions in cryptography,” IEEE Transactions on Information Theory, Vol. IT-22, 1976, pp. 644–654.
2. R. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryp-tosystems,” Communications of the ACM, Vol. 21 (2), 1978, pp. 120–126.
3. S. Goldwasser and S. Micali, “Probabilistic encryption and how to play mental poker keeping secret all partial information,” In Annual ACM Symposium on Theory of Computing (STOC), 1982.
4. P. L. Montgomery, “Modular multiplication without trial division,” Mathematical Computation, Vol. 44, 1985, pp. 519–521.
5. T. Kleinjung, K. Aoki, J. Franke, A. Lenstra, E. Thomé, J. Bos, P. Gaudry, A. Kruppa, P. L. Montgomery, D. A. Osvik, H. te Riele, A. Timofeev, and P. Zimmermann, “Factorization of a 768-bit RSA modulus,” Cryptology ePrint Archive: Report 2010/006, http://eprint.iacr.org/2010/006.