近代密碼學序曲

沈淵源

一、憶兒時

當我們小的時候, 或多或少會玩過一些密語的遊戲; 就是以一種很頑皮的方式傳達信息給對方, 懂的人就知道你在講什麼, 不懂的人就鴨子聽雷似的根本不曉得你在胡說些什麼。比如我們要說:「寄錢來」, 但講的時候將每個字注音的最後一個音不發出來, 如果只有一個音就還是發那個音。所以講出來就變成

「ㄐ　　ㄑㄧ　　ㄌ」,

請問你抓得到是什麼意思嗎? 這個例子當然還不夠格稱之爲密碼, 僅僅是一個有趣的開場白而已。現在讓我們看看眞正密碼的例子, 由這些例子中我們也能體會到數論在密碼術中的份量。難怪 Bruce Schneier[1] 說:「These days almost all cryptologists are also theoretical mathematicians-they have to be」。

二、一個例子

現在我們用英文的二十六個字母來傳達這個信息

「SEND MONEY」。

首先我們用 NUMBER THEORY 當成所謂的 (加密) 鑰匙, 將重複的字母去掉剩下 NUMBERTHOY, 然後把這些字母放在依序排列的二十六個字母下面, 再將其餘字母依序排列如下:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
N U M B E R T H O Y A C D F G I J K L P Q S V W X Z

很顯然的這是個一一對應, 我們把上一行的字母用下一行相對應的字母頂替, 那麼所要傳達的信息就變成爲

「LEFB DGFEX」。

換句話說 LEFB DGFEX 是 SEND MONEY 的密文。

---

[1] Bruce Schneier (布魯斯·施奈爾) 是國際聞名的一位資訊安全專家及作家, 他是 Counterpane Internet Security, Inc.(專精於密碼術與電腦安全的一個顧問公司) 的創始者同時也是這個公司的 CTO (Chief Technical Officer)。他所寫的應用密碼術 (Applied Cryptography) 一書被譽爲密碼術的聖經, 是最好的入門介紹書之一。其著作翻譯成中文者有「秘密與謊言」, 由商周出版社於 2001 年 9 月 16 日初版發行。

三、安全性可慮?

代換法雖然對長一點的信息來說並不是一個妥當保持信息安全的好方法, 但誰能識破其眞相呢? 值得懷疑! 因爲從觀察 LEFB DGFEX 中, 重複的字母提供了唯一的線索。由此線索推論出來與原信息同一形式的信息可能是

LIST MUSIC, MINE TUNIS, 或 DRAW CHART

當然你如果知道那加密的鑰匙以及整個方法, 你就能如法製作上表並將整個過程逆推回去得到原先的信息。這意味著解密的鑰匙很容易就可以從加密的鑰匙得到, 此乃傳統密碼術的特性, 通常稱之爲對稱式密碼系統。所以這整個信息的安全性, 完全掌控在那 (加密的) 鑰匙, 只要能保住此鑰匙的秘密性則安全性大抵上沒問題。此處尙有多重的困難需要考慮:

1. 此鑰匙是傳遞信息的雙方都要知道的, 只要有一方沒保住, 安全性就被破壞了。
2. 何況有時候傳遞信息的對象是多方的, 而人多嘴雜, 安全性更是可慮。
3. 從另一個角度來看, 如果鑰匙很複雜, 複雜到必須「寫下來」; 那麼被發現的機會更是有增無減。
4. 而且利用此法互通信息時, 也必須事先安排好如何讓要秘密通信息的雙方或多方同意或知道此鑰匙。這整個的「事先安排」也要妥善極機密的進行才可以。

由於這種種的原因, 刺激人們去探討「更高桿的」或「更深入的」密碼術。所以我們希望「信息的安全性」不會受制於「鑰匙的秘密性」, 亦即我們希望「信息的安全性」能獨立於「鑰匙的秘密性」。換句話說, 我們希望即使鑰匙是公開的, 信息仍然能維持其安全性。更具體的說, 我們想要達到的目標是:

「給你鑰匙 , 你只能將原信息變成密碼文, 卻無法將密碼文破解回歸其廬山眞面目。」

這對傳統的密碼系統來說是不可能的, 因爲解密鑰匙與加密鑰匙是對稱的, 公開加密鑰匙就等於是公開解密鑰匙。所以我們必須將加密鑰匙與解密鑰匙之間的對稱性打破, 唯有如此才能突破僵局。很自然的, 說到鑰匙就會聯想到門。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。

四、來自數論的靈感

如何能達到上面的目標呢? 乍看之下似乎不可能, 但門的比喻給我們一些啓發或暗示。在數論中有類似的例子。譬如, 找出兩個很大的質數 p 與 q (說是一百位數好了) , 然後將這兩個

質數相乘得到其乘積爲 $n = pq$ 。這整個過程比起其逆過程 (給你 n, 找出這兩個質數) 來得簡單, 而且簡單得許多。

下面的例子中爲了方便說明起見, 我們取較小的兩個質數 p 與 q; 如 p 至少25, 那你就查驗26,27,28都不是, 29是也; q 至少40, 下一個數41就是了。這個步驟很快, 然後將其相乘得到 $n = pq = 1189$, 這個步驟也很快。反之, 給你1189, 那你就得試 2,3,5,7,11,13,17,19,23 等共九個除法之後的29才得到整除:

$$1189 = 29 \times 41。$$

這裡因爲1189很小, 感覺不出找29會有困難, 但當 n 爲一200位數的整數時, 其困難度可想而知! 我們還會用到整數 e, 此數與 $\phi(n) = (p-1)(q-1)$ 互質。在我們的例子中可取 $e = 3$, 此處

$$n = 1189, \qquad e = 3$$

是公開的鑰匙, 而 p 與 q 則保持秘密。這次我們將26個英文字母及空白用整數來表示如下:

空白	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

再一次的我們要傳遞 SEND MONEY 這一信息。先將其化爲對應的數字, 如下:

$$19 \quad 05 \quad 14 \quad 04 \quad 00 \quad 13 \quad 15 \quad 14 \quad 05 \quad 25$$

因爲1189是四位數, 所以我們將信息分爲三位數一組, 如下:

$$190 \quad 514 \quad 040 \quad 013 \quad 151 \quad 405 \quad 250$$

(在尾巴加了一個0, 因爲最後一組數25不是三位數)。我們用 $P_1, P_2, P_3, \ldots$ 來表示這些三位數, 然後按照下面的法則轉換成密碼:

$$C_i \equiv P_i^e \pmod{n}, \quad i = 1, 2, 3, \ldots,$$

此處 C_i 爲介於0與 $n-1$ 之間的整數, 亦即被 n 除的餘數。

$$\begin{aligned} C_1 &\equiv 190^3 = 6859000 \equiv 848 \pmod{1189}, \\ C_2 &\equiv 514^3 = 135796744 \equiv 1054 \pmod{1189}, \\ C_3 &\equiv 040^3 = 64000 \equiv 983 \pmod{1189}, \end{aligned}$$

$$C_4 \equiv 013^3 = 2197 \equiv 1008 \pmod{1189},$$
$$C_5 \equiv 151^3 = 3442951 \equiv 796 \pmod{1189},$$
$$C_6 \equiv 405^3 = 66430125 \equiv 695 \pmod{1189},$$
$$C_7 \equiv 250^3 = 15625000 \equiv 351 \pmod{1189}。$$

所以收到的密文爲:

$$848 \quad 1054 \quad 983 \quad 1008 \quad 796 \quad 695 \quad 351。$$

如何將這些密文 $\{C_i\}$ 破解回歸其廬山眞面目 $\{P_i\}$ 呢? 令 $b = [p-1, q-1]$ 爲 $p-1$ 與 $q-1$ 的最小公倍數 (注意: 若不知 p, q 則 b 就無從知道), 且令 d 爲同餘方程式 $ex \equiv 1 \pmod{b}$ 中的最小正整數解。解出得 $b = 280, d = 187$, 然後計算 $C_i^d \pmod{1189}$, $i = 1, 2, 3, \ldots$ 即可得 P_i, $i = 1, 2, 3, \ldots$。

爲何如此呢? 且將此問題暫時擺在一旁, 我們先來算 C_i^d。第一個碰到的是 848^{187}, 即將848自乘187次, 這是一個很大的數。表面上你得執行186次的乘法, 這相當花時間。如果只是平方, 速度就非常快。重複此一運算, 我們就得到4次方、8次方、16次方、32次方、64次方、128次方 ..., 這提供了一個解決 848^{187} 的計算問題。先將187寫成 (即 187之二進位表示法爲10111011)

$$2^7 + 2^5 + 2^4 + 2^3 + 2^1 + 2^0 = 128 + 32 + 16 + 8 + 2 + 1,$$

再依次計算848的 2^n 次方如下:(這兒我們引進負數, 爲的是將每個數的大小調成比 $n/2$ 小; 如此可稍稍減少所要執行的計算量, 特別是當你用手算時這是一大幫助。)

$$\begin{aligned}
848^1 &\equiv -341 \pmod{1189},\\
848^2 &\equiv (-341)^2 \equiv 948 \equiv -241 \pmod{1189},\\
848^4 &\equiv (-241)^2 \equiv -180 \pmod{1189},\\
848^8 &\equiv (-180)^2 \equiv 297 \pmod{1189},\\
848^{16} &\equiv 297^2 \equiv 223 \pmod{1189},\\
848^{32} &\equiv (223)^2 \equiv -209 \pmod{1189},\\
848^{64} &\equiv (-209)^2 \equiv -312 \pmod{1189},\\
848^{128} &\equiv (-312)^2 \equiv -154 \pmod{1189}。
\end{aligned}$$

所以我們得到

$$848^{187} \equiv (-154)(-209)(223)(297)(-241)(-341) \equiv 190 \pmod{1189}。$$

值得注意的是, 原來需執行186次乘法的計算, 用上面的演算法只需12次。同法可算出其他的如下:

$$\begin{aligned}
1054^{187} &\equiv (-135)(390)(141)(-332)(-353)(-187) \equiv 514 \pmod{1189},\\
983^{187} &\equiv (-206)(-368)(-573)(165)(-122)(165) \equiv 40 \pmod{1189},\\
1008^{187} &\equiv (-181)(-531)(-312)(-154)(-64)(426) \equiv 13 \pmod{1189},\\
796^{187} &\equiv (-393)(-121)(16)(256)(141)(-353) \equiv 151 \pmod{1189},\\
695^{187} &\equiv (-494)(291)(-318)(59)(-86)(-318) \equiv 405 \pmod{1189},\\
351^{187} &\equiv (351)(-455)(297)(223)(-209)(-154) \equiv 250 \pmod{1189}。
\end{aligned}$$

所以我們就得到原來的 $P_1, P_2, P_3, \ldots$。爲何這個方法行得通呢? 從 P 到 C 其運算爲 $C \equiv P^e \pmod{n}$, 而解密 C 則計算 $P' \equiv C^d \pmod{n}$, 我們要證明 $P' = P$。實際上我們假設 P, P' 都是小於 n 的正整數, 所以只要證明 $P' \equiv P \pmod{n}$ 即可! 由定義可知

$$P' \equiv C^d \equiv (P^e)^d \equiv P^{de} \pmod{n},$$

而 d 爲同餘方程式 $ex \equiv 1 \pmod{b}$ 中的最小正整數解, 此處 $b = [p-1, q-1]$ 爲 $p-1$ 與 $q-1$ 的最小公倍數 (因我們取 e 與 $p-1$ 及 $q-1$ 互質, 因此 e 與 b 互質, 所以此同餘式有解)。令 $de = 1 + kb$。若 $(P, p) = 1$, 則費馬小定理告訴我們 $P^{p-1} \equiv 1 \pmod{p}$。因 $p-1 \mid b$, 所以 $P^{kb} \equiv 1 \pmod{p}$。因此

$$P^{de} \equiv P^{kb+1} \equiv P \pmod{p}。$$

若 $p \mid P$, 則上面的同餘式顯然成立。同理 $P^{de} \equiv P \pmod{q}$。因此

$$P^{de} \equiv P \pmod{n}。$$

五、計算之複雜度的分析

經過上面的一番計算之後, 讓人深深的覺得這些計算最好是由電腦來代勞。我們同時也注意到分解 $n = 1189$ 的困難度是整個方法的安全性之關鍵, 但在上例中有些計算卻比分解 1189 來得複雜得多。當然如果你有辦法算出848自乘187次, 則分解1189應該是簡單之至。這好像有點似是而非, 如何解釋這種似是而非的現象呢? 這就牽涉到整個計算當中數字的大小以及計算的複雜度的問題。記得嗎? 在實際的應用當中, 質數 p與 q 都是100位數之大, 相乘之後就變成200位數。所以在加密或解密的計算很大次方的過程當中, 雖然我們不可嗤之以鼻的輕看其計算量, 但比起要分解一個200位數 (就目前已知的演算法) 的計算量是來的少, 而且少很多。

要分析這其中的奧秘, 我們就得將所涉及到的運算分解成一些最基本的運算, 如加法、減法、乘法、除法、以及相互比較 (Comparison)。當然, 數字的大小會影響到電腦去執行這些運算的時間。所以爲了簡單起見, 我們假設所有這些運算所需的時間都是一樣的, 比如說一秒一百萬次。

在解密的過程中, 最困難的地方似乎是將一個數自乘 d 次, 此處 d 爲同餘方程式 $ex \equiv 1 \pmod{b}$ 中的最小正整數解, 而

$$b = [p-1, q-1] = (p-1)(q-1)/(p-1, q-1) < pq/2 < pq/2 = n/2,$$

所以我們可假設 $d < n/2$。首先我們把 d 連續除以2, 將之化爲二進位數。這需要幾次的除法呢? 很明顯的, 這個數目就是 d 的二進位表示法的位數, 說是 k 吧! 則

$$2^{k-1} \le d \Rightarrow k - 1 \le \log_2 d \le \log_2(n/2) = \log_2 n - 1,$$

所以我們得知 k 的一個上界爲 $\log_2 n$。此數隨著 n 的增加而增加, 但成長的非常緩慢; 如 n是200位數, 而 $\log_2 n$ 比700還小。

我們在上面用了平方法來處理次冪的計算, 先做了 k 次的平方然後除以 n 得其餘數。這需要少於700次的乘法及少於700次的除法, 所以合起來不會超過2100次的基本運算。若要我們用手去算, 門都沒有! 但對電腦來說這可是輕而易舉的事情, 不吭半聲所有的計算就結束了。接著將這些所得到的平方數兩兩相乘再除以 n 得其餘數, 這不超過 k 次的乘法也不超過 k 次的除法, 所以所需要的基本運算合起來不會超過3500次。

再來我們得估計一下解同餘方程式

$$ex \equiv 1 \pmod{b}$$

所需的基本運算的次數。這需要將 b 與 e 輾轉相除, 不難證明其次數會少於 $2\log_2 n$, 即1400次的除法。逆推回去, 將1 寫成 b 與 e 的線性組合, 所需要的基本運算跟前面的輾轉相除法合起來不會超過 2800次。

所以加密與解密合起來所需要的基本運算最多不會超過10000次, 而以一秒一百萬次的速度來執行這些基本運算的電腦, 在一秒鐘裡面就可以完成十件這種計算的工作。也許一秒作一百萬次基本運算的電腦是快過現有的任何一部電腦, 但這樣的假設只不過是提供我們作爲「比較」的用途。退一萬步來想, 我們假設那試圖破解密文的人有如此快速的電腦可使用也是應該的, 因爲做最壞的打算才是上上之策!

六、誰還管生生世世夜夜朝朝?

話說回來, 那試圖破解密文的人必先分解那巨大無比的200位數 n 成爲 $p \times q$, 由此來解出 b 然後再執行上述的計算來解密。如果此人所用來分解 n 的方法是除以從1開始一直到 $\sqrt{n}$ 的所有整數的話, 那麼光這項工作就得執行 10^{100} 次的除法。如果我們用一秒一百萬次快速的的電腦也需要 10^{94} 秒鐘, 亦即 3×10^{86} 年左右才能算完。當然實際上只需除以其中的質數, 但問題是要去判別在100位數之內何數爲質數, 這是相當頭痛的事情。即使用現今最快速的電腦及最有效率的演算法要分解一個200位數, 也要40億年才能完成。到那時候, 誰還管何人會來看我們的信息呢?

—本文作者任教於東海大學數學系—