簡介

「我忍不住要偷聽，可能是因為我就是在竊聽。」

—Anonymous

如果您能帶著這本書回到 1990s 年代初期，本書第 23 章所談到的 RSA 密碼實作內容是不能傳出美國的。因為關於 RSA 解密的訊息有可能被駭，釋出 RSA 解密軟體可是攸關國家安全，需要得到州政府許可才能這麼作。事實上，強大的密碼學與坦克、導彈和火焰噴射器是同一等級，需要受到監管。

在 1990 年，加州柏克萊大學的學生伯恩斯坦（Daniel J. Bernstein）想要發表一份關於 Snuffle 解密系統的學術論文，但美國政府要他先成為政府授權的武器供應商之後才能在網路上公開原始程式碼。政府還通知他，如果想要申請釋出和出口的許可，也會被拒絕，因為他的技術太過攸關國家安全。

電子前鋒基金會（Electronic Frontier Foundation），這個年輕的數位公民自由組織在伯恩斯坦訴合眾國案（Bernstein v. United States）中代表 Daniel J. Bernstein

進行司法訴訟挑戰美國密碼學出口限制。史上第一次由法院裁定軟體原始程式碼是受到第一修正案的言論自由所保護，政府阻止伯恩斯坦的發表是違反第一修正案。

現今強大的密碼學（Cryptography）已成為全球經濟的重要組成基石，每天都在保護數百萬網路購物者能安全地使用各種商業和電子商務的網站。美國情報體系推測加密軟體會成為嚴重的國家安全威脅是沒有根據的。

但在 1990 年代當時，自由傳播這些知識（如同本書所發表的），可能會讓您因武器販運而入獄坐牢。若想要了解更多關於密碼學為了自由而進行法律抗爭的詳細歷史，可閱讀 Steven Levy 的書：「Crypto: How the Code Rebels Beat the Government, Saving Privacy in the Digital Age (Penguin, 2001)」。

適合閱讀本書的讀者

坊間有很多書都在教導初學者如何使用密碼（cipher）來編寫機密訊息，也有一些書是指導初學者怎麼破解密碼。但卻沒有一本書在教導初學者如何編寫設計電腦程式來破解密碼。本書就是能滿足這種需要的書籍。

這本書適合那些對加密、破解駭入密碼或密碼學技術感興趣的讀者。本書所使用的密碼（除了第 23 和 24 章所用的公鑰密碼外）已有數百年的歷史，但任何一台筆電所具備的運算能力都能破解。現在的組織或個人都已經沒有在使用這些密碼了，但藉由學習這些舊有的密碼技術，您會學到現今駭客高手們所具備的密碼學基礎，以及他們怎麼破解比較弱的加密技術。

NOTE

在本書中所學習到的密碼用來玩玩是可以，但並不能提供真正的安全。別用書中的加密程式來對現實中的檔案進行加密保護，總而言之，別太相信您所製作的密碼。真實世界中所用的密碼在真的拿來使用之前，都是需要經過專業密碼學家多年的分析後才拿來運用的。

這本書也是針對那些沒寫過程式的讀者所設計的，本書會講解程式設計的基本概念，使用的是 Python 程式語言，這套程式語言對初學者來說是最好的選擇，Python 具有溫和的學習曲線，適合各種年齡層的新手，但它也是專業軟體開發人員所使用的強大程式語言。Python 可以在 Windows、Linux 和 Mac OS 系統

中執行，甚至在 Raspberry Pi 中執行也沒問題，它可以免費自由下載使用（請參考「下載與安裝 Python」小節的說明指示）。

在本書中，我會用到「駭客（hacker）」這個詞。這個詞具有兩種意義，駭客可以是個想要深究系統（像密碼或是某套軟體的規則），詳實了解其內涵，但不會被原有的規則所限制，並能修改系統，讓它能更有創意地發揮。駭客也可能是個犯罪者，駭入別人的電腦系統，侵害他人的隱私並造成破壞損失。本書所使用的「駭客（hacker）」這個詞是偏向前者的意義。駭客也可以是很酷的人，但犯罪份子就只是想要破壞別人的東西來突顯自己的聰明而已。

本書內容

最前面的幾章內容是介紹 Python 和密碼學的基礎概念，後續的章節大都在解釋建立密碼的程式，以及說明破解該密碼的程式。每一章都有練習題目，幫助讀者溫習回顧所學的東西。

Chapter 01：**製作紙上的密碼工具**。本章介紹幾個用紙張來處理的工具，說明在還沒有電腦前是怎麼處理加密的工作。

Chapter 02：**在互動式 Shell 中程式設計**。本章介紹怎麼使用 Python 的互動式 Shell 模式來逐行處理程式碼。

Chapter 03：**字串與編寫程式**。本章開始要編寫設計出完整的程式碼，並介紹在本書中所有程式都會使用到的字串資料型別。

Chapter 04：**反轉密碼**。介紹如何編寫出您的第一支簡易型的加密程式。

Chapter 05：**凱撒密碼**。介紹這個已有千年歷史的基礎簡易加密方法。

Chapter 06：**以暴力破解凱撒密碼**。解說暴力破解的技術，並說明在沒有密鑰下如何利用暴力破解來進行解密。

Chapter 07：**使用換位密碼進行加密**。本章介紹換位密碼的原理，以及編寫以這個加密方法來對訊息進行加密的程式。

Chapter 08：**使用換位密碼進行解密**。介紹換位密碼的那一半內容：使用密鑰來解密訊息。

Chapter 09：**編寫程式來測試您的程式**。介紹設計測試其他程式的相關技術。

Chapter 10：**加密和解密檔案**。說明如何編寫從硬碟讀取和寫入檔案的程式。

Chapter 11：**以程式來偵測英文**。介紹如何使用電腦來偵測英文句子。

Chapter 12：**破解換位密碼**。融合前幾章所學到的觀念來破解換位密碼。

Chapter 13：**仿射密碼所用的模算術模組**。介紹仿射密碼背後的數學概念。

Chapter 14：**仿射密碼程式設計**。介紹如何編寫以仿射密碼加密的程式。

Chapter 15：**破解仿射密碼**。解說如何編寫可以破解仿射密碼的程式。

Chapter 16：**簡易替換密碼程式設計**。介紹如何編寫使用簡易替換密碼加密的程式。

Chapter 17：**破解簡易替換密碼**。解說如何編寫破解簡易替換密碼的程式。

Chapter 18：**維吉尼亞密碼程式設計**。介紹如何編寫比替換密碼更複雜的維吉尼亞密碼加密程式。

Chapter 19：**頻率分析**。探索英文單字的結構，並利用這個技巧來破解維吉尼亞密碼。

Chapter 20：**破解維吉尼亞密碼**。解說如何編寫破解維吉尼亞密碼的程式。

Chapter 21：**一次性密碼本密碼**。介紹一次性密碼本密碼，以及說明為什麼以數學運算不能破解它。

Chapter 22：**尋找與產生質數**。解說如何編寫可以快速判別某個數字是否為質數的程式。

Chapter 23：**生成公鑰密碼的公鑰**。解說公鑰密碼學的知識，以及如何編寫生成公鑰和私鑰的程式。

Chapter 24：公鑰密碼程式設計。說明如何編寫不可能用電腦來破解的公鑰密碼程式。

附錄：偵錯 Python 程式碼。介紹如何使用 IDLE 的 Debugger 來找出和修改程式中的錯誤。

如何使用本書

本書與其他書籍不太一樣，因為本書的焦點放在完整程式中的原始程式碼解說。不像其他書只教您一些程式設計的概念，就把您丟在那，讓您自己想辦法寫出程式。本書會秀出完整的程式碼，並詳實解釋其運作的原理。

一般來說您最好依照書中章節順序來閱讀與學習，因為有些知識和觀念是建構在前面章節內容，以前面內容為基礎循序學習才更有效果。不管怎麼說，最前面幾章之後的 Python 程式碼很具有可讀性，您可以直接跳後面感興趣的章節，看看這些程式有什麼功用。如果您直接跳到後面章節閱讀學習時碰到瓶頸，可先跳回到前面章節學習。

輸入原始程式碼

在您閱讀本書時，*強烈建議您自己動手輸入書中講述的 Python 程式碼*，這樣會讓您更容易理解程式碼的運作原理。

當您輸入原始程式碼時，不要把書上最左側的行號也一併輸入，這些行號不是程式的內容，僅是在書上方便我們閱讀和指出某行程式來說明時使用的。除了行號之外，請照著書上所示的內容完整輸入，包括字母大小寫等都要一樣。

您可能看到書上的某些程式行並不是從最左側起始，而是有縮排，有空 4 格、8 格或更多空格。請一定要照著書上所示正確輸入縮排的空格數，這樣才不會有錯誤發生。

如果您真的不想自己動手輸入，您也可以連到下列網址來下載本書所有的程式碼檔案：https://www.nostarch.com/crackingcodes/。

第 1 章
製作紙上的密碼工具

「加密的精靈已經跑出瓶子，沒什麼能阻止它發展與前進了。」

—Jan Koum, WhatsApp founder

在我們編寫密碼程式之前，先來看看使用鉛筆和紙張來加密與解密的處理過程。這樣有助於讓您理解密碼運作的原理，以及在生成這些秘密訊息中所引用的數學觀念。在本章中，您將會學到密碼學的意義以及代碼（code）與密碼（cipher）有何不同。隨後會用筆和紙，並以簡易的凱撒密碼來對訊息進行加密和解密處理。

本章內容

- 什麼是密碼學（cryptography）？
- 代碼（code）與密碼（cipher）
- 凱撒密碼（Caesar cipher）
- 密碼轉盤（Cipher wheels）
- 以算術處理密碼學
- 雙重加密

什麼是密碼學？

從歷史上看，任何想與他人分享秘密的人，例如間諜、士兵、駭客、海盜、商人、獨裁者和政治人物，都仰賴密碼學來確保他們的秘密能守得住。

密碼學（cryptography）就是運用秘密代碼的科學。想要了解密碼大概長成什麼樣子，可看下列兩段文字：

```
nyr N.vNwz5uNz5Ns662ONzON3z2v
N yvNwz9vNz5N6!9Nyvr9
yOQNnvNwv tyNz
Nw964N6!9N5vzxys69O,N.vN2z5u-
3vNz Nr Ny64v,N.vNt644!5ztr vNz
N 6N6 yv9O,Nr5uNz Nsvt64vON
yvN7967v9 BN6wNr33Q N-m63 rz9v
```

```
!NN2 Nuwv,N9,vNN!vNrBN3zyN4vN
N6 QvvOz6nvN.7NOyv4N 4 zzvNN
vyN,NN99zOzz6wzOy3vv26 9
w296vyNNrrNyQst.56ON94Nu5y
rN5nz5vv5t6v63zNr5.
N75sz6966NNvw6 zuO wtNxs6t
49NrN3Ny9Nvzy!
```

左側的文字是經過加密（encrypted）的訊息文字，這段文字已變成秘密代碼。如果不知道怎麼解密（decrypt），或者說不知道怎麼把它變回普通的英文訊息，那就完全看不懂這是在說什麼。右側的訊息則是亂編的字碼，是沒有任何意義的內容。加密訊息能對他人保密，就算別人取得加密後的訊息也看不懂。加密後的訊息看起就像隨機亂編的字碼一樣。

密碼研究員（cryptographer）是運用和研究秘密代碼的人。當然囉，沒有秘密是能一直維持的，密碼分析（cryptanalyst，或稱破密學），又稱破密者（code breaker）或駭客（hacker），能破解秘密代碼，讀懂別人加密的訊息。本書會教

您怎麼利用不同的技術來加密和解密訊息。但請小心，希望您在本書所學到的破解技術不會讓您陷入違法的麻煩事中。

代碼與密碼

代碼（code）與密碼（cipher）並不同，代碼做出來的目的是可讓人能理解並能公開使用。任何人都能認出或找出代碼符號所隱含的意義，並轉譯成可理解的訊息。

在 19 世紀之初，有個大家都知道的代碼運用就是電報的發明，電報透過電線的傳輸可跨越遠距離的各大洲來進行即時的通訊。以電報來傳送訊息會比背著一袋信件騎馬派送是快得多了。不過電報不能直接傳送寫在紙上的訊息，只能傳送兩種電子脈衝：短脈衝叫作「點（dot）」，而長脈衝則叫作「破折線（dash）」。

若想要將英文字母轉換成點和破折線，則需要一個能將英語轉譯成電子脈衝的編碼系統。把英文轉換成點和破折線來用電報傳送的過程稱為編碼（encoding），在接收到訊息時把電子脈衝轉換成英文的過程稱為解碼（decoding）。用於電報（後來有用無線電）來傳送，對訊息進行編碼和解碼的代碼稱為摩斯電碼（Morse code），如表 1-1 所示。摩斯電碼是由 Samuel Morse 和 Alfred Vail 發明的。

表 1-1　國際摩斯電碼編碼對應表

字母	編碼	字母	編碼	數字	編碼
A	•–	N	–•	1	•————
B	–•••	O	———	2	••———
C	–•–•	P	•––•	3	•••––
D	–••	Q	––•–	4	••••–
E	•	R	•–•	5	•••••
F	••–•	S	•••	6	–••••
G	––•	T	–	7	––•••
H	••••	U	••–	8	–––••
I	••	V	•••–	9	––––•
J	•–––	W	•––	0	–––––
K	–•–	X	–••–		
L	•–••	Y	–•––		
M	––	Z	––••		

藉由一個電報按鍵敲出點和破折線，電報操作員可以在幾乎是即時的情況下把英文訊息傳送給世界另一端的某個人。（如果想要了解更多關於摩斯電碼的內容，請連到 https://nostarch.com/crackingcodes/ 網站內 Extra Stuff 的 Additional online resources 連結，或是直接連到 https://zh.wikipedia.org/wiki/摩斯電碼）

相較於代碼，密碼是一種特定類型的代碼，是用來讓訊息保密的。我們可以把能理解的英文文字（明文，plaintext）轉換成隱藏有訊息的亂碼（密文，ciphertext）。密碼（cipher）是指一組可轉換明文和密文的規則（rule）。這些規則通常是用密鑰（secret key）來加密和解密，而此密鑰只有傳播者自己知道。在本書中，您將會學到多種不同的密碼，並學會使用這些密碼規則編寫設計程式來對文字進行加密和解密的處理。不過，我們先學習使用簡易的紙筆工具，以手動方式對訊息進行加密處理。

凱撒密碼

第一個要學習的是凱撒密碼（Caesar cipher），是以 2000 年前使用這個密碼的 Julius Caesar 來命名。這個密碼的好處很簡單且容易學習，但壞處也是因為簡單而容易被分析破解。不論如何，凱撒密碼還是個很有用的教學練習。

凱撒密碼的工作原理是對訊息的英文字母依照字母表順序向前向後移動，以新的字母替換訊息的每個字母。舉例來說，Julius Caesar 決定對信件中的字母以字母表向後 3 個字母的順序來替換，將信件中原本的字母全都替換成向後移的字母。

舉例來說，訊息中每個 A 字母都會以 D 來替換，而 B 則以 E 來替換，以此類推。當凱撒向後移超出字母表範圍時，例如 Y，則轉回到字母表從開頭來算，以 B 來替換。在這一小節中，我們會試著以手動的方式運用凱撒密碼來對訊息進行加密。

密碼輪盤

為了要讓凱撒密碼把明文轉換成密文更容易些，我們會使用密碼輪盤（cipher wheel），或稱密碼碟（cipher disk）。密碼輪盤有兩個字母環，每個環分成 26 個槽（用來放 26 個字母）。外側的環代表明文用的字母，內側的環代表密文對應

的字母。內側環也會對字母編號，從 0 到 25 號。這些數字代表的是密鑰（encryption key），也就是從 A 轉換到內環上對應字母所用的數值。因為是圓形環狀位移的，若用大於 25 的密鑰數值來位移則會轉回環狀的開頭，所以位移 26 與位移 0 相同，位移 27 與位移 1 相同，以此類推。

您可以連到 https://nostarch.com/crackingcodes/ 網站，點選 Extra Stuff 的 Additional online resources 連結，其中有個 Online Cipher Wheel（線上的虛擬密碼輪盤）可使用。圖 1-1 為線上虛擬密碼輪盤的畫面，以滑鼠點按一下，就可移動滑鼠游標轉動外環字母，設定到您想要對應的內環字母數值上，再點按一下畫面就能停止轉動。

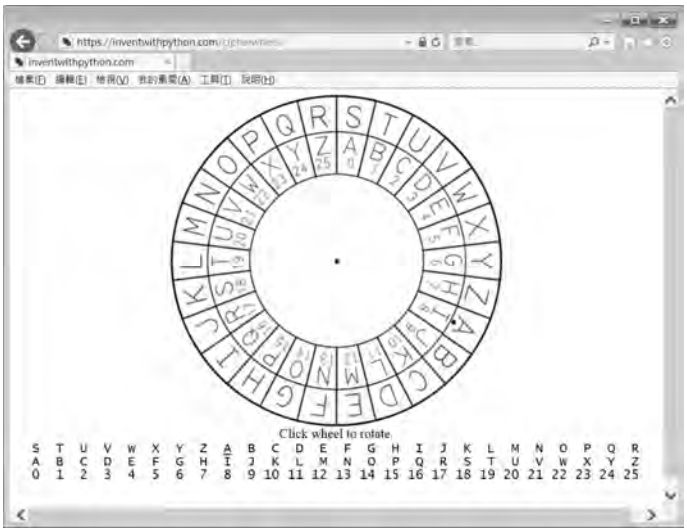

圖 1-1　線上密碼輪盤

也可將上面的密碼輪盤以印表機印出來，再分別將內環和外環用剪刀剪下，把小環放在大環中間，在兩個圓環中間插上一根大頭釘，這樣就可以手動旋轉了。

不管是使用線上密碼輪盤或是您剪下的紙質密碼輪盤，您都能自己手動把訊息進行加密。

以密碼輪盤進行加密

若要開始進行加密處理，要先把訊息以英文寫在紙張上。舉例來說，我們想要對「THE SECRET PASSWORD IS ROSEBUD」這行訊息進行加密。隨即轉動內環，直到其字母對應到外環的字母，請留意外環 A 字母下有個小黑點，再查看這個黑點所對應到的內環字母槽中的數值，此數值就是密鑰。

舉例來說，在圖 1-1 中外環 A 對應到內環的 I 槽，其數值為 8，這表示我們會用 8 這個密鑰來加密範例訊息，如圖 1-2 所示。

T	H	E		S	E	C	R	E	T		P	A	S	S	W	O	R	D		I	S		R	O	S	E	B	U	D
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
B	P	M		A	M	K	Z	M	B		X	I	A	A	E	W	Z	L		Q	A		Z	W	A	M	J	C	L

圖 1-2　使用凱撒密碼 8 為密鑰來進行加密

對訊息中的每個字母都先從外環找出字母，然後以對應到內環的字母來替換。在這個範例中，第一個字母是 T（「THE SECRET…」的第一個字母 T），我們在外環先找到 T，然後再找到內環對應的字母，這裡是指 B。所以秘密訊息中的 T 都會替換成 B（假如我們不是使用 8 來當作密鑰，則明文裡的 T 會替換成別的字母）。訊息內的第二個字母是 H，會轉換成 P。字母 E 則轉換成 M。每個在外環的字母都要在加密時對應到內環相同的字母。為節省時間，在您找過 THE SECRET…訊息的第一個字母 T，並替換成 B 之後，接著若在訊息中看到 T 時，就可直接替換成 B，所以您只要查一次就好。

在您把整段訊息都進行加密之後，原本訊息「THE SECRET PASSWORD IS ROSEBUD」會變成「BPM AMKZMB XIAAEWZL QA ZWAMJCL」。請留意訊息中非字母的部分，像空格是沒有改變的。

現在您可以把這條加密訊息傳送給別人（或是自己留下），除非您有告知密鑰為 8，不然別人是看不懂這條訊息的。請小心保存密鑰，任何人若知道此訊息是以密鑰為 8 來加密的，那麼密文就會被解讀。

以密碼輪盤進行解密

若要對密文進行解密，要從密碼輪盤的內環開始移向外環。舉例來說，您收到一條密文「IWT CTL EPHHLDGS XH HLDGSUXHW」，假如您沒有密鑰，那您就沒法子解密（除非是您厲害的駭客）。不過您朋友已把密鑰為 15 告訴您了，圖 1-3 所示為設好密鑰 15 的密碼輪盤。

圖 1-3　密碼輪盤設定密鑰為 15

現在可以把外環字母 A（下面有小黑點的那個）對準內環上數字為 15 的字母（也就是 P）。隨後開始從內環找密文訊息裡的第一個字母 I，再看其外環對應的字母，也就是 T。密文第二個字母 W 解密後是 H。接著把密文裡的字母都解密成明文，就可得到「THE NEW PASSWORD IS SWORDFISH」的訊息，如圖 1-4 所示。

I	W	T		C	T	L		E	P	H	H	L	D	G	S		X	H		H	L	D	G	S	U	X	H	W
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
T	H	E		N	E	W		P	A	S	S	W	O	R	D		I	S		S	W	O	R	D	F	I	S	H

圖 1-4　以凱撒密碼的密鑰為 15 來解密訊息

如果您使用了錯誤的密鑰，例如 16 來解密時，得到的結果是「SGD MDV OZRRVNQC HR RVNQCEHRG」的訊息，完全看不懂其意義。除非有正確的密鑰，否則解密的訊息也一樣讓人看不懂。

以數學算術來加密和解密

對於以凱撒密碼來進行加密和解密來說，密碼輪盤雖然是個很方便的工具，但我們還可以透過算術運算來進行加密和解密。其作法是寫下字母表 A 到 Z，並在每個字母下面編上數字編號，從 0 到 25。在 A 下面是 0，B 下面是 1，以此類推到 Z 下面是 25。圖 1-5 列出這樣的字母編號列表。

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
0	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

圖 1-5　字母編號列表

這樣我們就可以字母-數字的配對，來以數字代表字母了。這是個很厲害的觀念，因為這樣就能以數學來處理字母了。舉例來說，如果我們把 CAT 字母以數字 2、0 和 19 來代表，那麼分別加上 3 時，就可得到 5、3 和 22。這三個新數字以圖 1-5 來查找對應，其代表的就是 FDW 字母。我們只是對 CAT 字「加上」3，接著就要來學習使用程式，讓電腦來幫我們處理算術的運算。

若要以算術運算來處理凱撒密碼的加密工作，只要找出要加密字母下所代表的數字，再加上密鑰數值即可。舉例來說，我們要來把「HELLO. HOW ARE YOU?」訊息加密，使用 13 為密鑰（也可以用 1 到 25 之間任意整數為密鑰）。首先要找出 H 字母代表的數字 7，然後加上 13：7+13=20，由於得到的 20 由對應表中找到其代表字母為 U，所以 H 就加密為 U。

同樣地，對字母 E（4）加密，加上 13：4+13=17，數字 17 對應的是字母 R，所以 E 加為 R，以此類推。

這個處理過程直到 O 時出現問題，O 代表的數字是 14，再加上 13 時得到 27，但字母表編號只到 25，如果在加總字母代表數字和密鑰數值後的結果大於等於 26 的話，就要減掉 26。以前面的例子來看，27-26=1，數字 1 所對應的字母為 B，所以字母 O 以密鑰 13 加密後是 B，當您把訊息的字母全都加密後得到的密文為「URYYB. UBJ NER LBH?」。

若要對密文進行解密，則是要減掉密鑰值而不是加上。密文字母 B 代表的數字是 1，若減掉 13 得到-12，像在加密時「減掉 26」這些的規則，當在解密時碰到結果小於 0 的話，就「加上 26」。前面-12+26 的結果是 14，所以密文 B 解密回去就是 O。

NOTE

如果您還搞不太清楚怎麼處理負數的加和減，請連到 https://nostarch.com/crackingcodes/ 網站，點選 Extra Stuff 的 Additional online resources 連結，其中有個 Adding and Subtracting with Negative Numbers 的連結有更進一步的補充說明。

如您所見，使用凱撒密碼不一定要用到密碼輪盤，只要有紙、筆和簡單的數算運算也能搞定！

為什麼雙重加密不可行

您可能想到用兩個不同的密鑰對訊息進行雙重的加密處理，讓加密效果加倍。但這對凱撒密碼（大多數其他密碼）並不適用。實際上，雙重加密的結果與一次正常加密後的結果相同。讓我們試著對某條訊息進行雙重加密，看看其結果如何。

舉例來說，如果您對「KITTEN」這個字使用密鑰 3 來加密，對明文字母所代表的數字加上 3，然後取得密文結果為「NLWWHQ」。隨即再對「NLWWHQ」進行密鑰 4 的加密處理，加上 4 之後取得密文結果為「RPAALU」。這樣二次加密的結果與「KITTEN」這個字使用密鑰 7 來加密是一樣的。

對大多數的密碼來說，多次加密並不會增強加密的強度。事實上，如果我們用了兩個密鑰加起來等於 26 的數字來加密某些明文，所得到的結果是密文和明文是一樣的。

總結

幾個世紀以來，凱撒密碼和其他類似的密碼都一直用來對秘密資訊進行加密的處理。但如果想要對很長的訊息（例如一整本書）進行加密，若以人工方式來進行可能要花上數天或數週的時間才能完成。這種情況下程式設計能幫得上忙。用一台電腦來處理加密和解密的大量的文字可能只花數秒鐘就搞定了！

要使用電腦來進行加密，您要先學會編寫設計程式或指令，我們只要使用電腦看得懂的語言來指示，它就會完全照著指示來工作。幸運的是，學習像 Python 這樣的程式語言並不會像學日文或西班牙文這麼難。除了加、減和乘法之外，您並不需要知道太多其他的數學知識。您只需要有這本書和一台電腦就行了。

接著要繼續進入第 2 章，我們將要學習如何使用 Python 的互動式 Shell 模式來逐行處理指令程式碼。

練習題

練習題的解答可連到 https://www.nostarch.com/crackingcodes/ 取得。

1. 對 Ambrose Bierce 所著的《The Devil's Dictionary》一書中的某些文字進行加密處理：
   a. 使用密鑰 4 加密：「AMBIDEXTROUS: Able to pick with equal skill a righthand pocket or a left.」
   b. 使用密鑰 17 加密：「GUILLOTINE: A machine which makes a Frenchman shrug his shoulders with good reason.」
   c. 使用密鑰 21 加密：「IMPIETY: Your irreverence toward my deity.」
2. 以下列給定的密鑰來對密文進行解密處理：
   a. 使用密鑰 15 解密：「ZXAI: P RDHIJBT HDBTIXBTH LDGC QN HRDIR WBTC XC PBTGXRP PCS PBTGXRPCH XC HRDIAPCS」
   b. 使用密鑰 4 解密：「MQTSWXSV: E VMZEP EWTMVERX XS TYFPMG LSRSVW.」
3. 使用密鑰為 0 對這句子進行加密：「This is a silly example.」。

4. 下列為某些文字和其加密後的密文，請找出它們使用的密鑰是多少？

   a. ROSEBUD – LIMYVOX

   b. YAMAMOTO – PRDRDFKF

   c. ASTRONOMY – HZAYVUVTF

5. 「UMMSVMAA: Cvkwuuwv xibqmvkm qv xtivvqvo i zmdmvom bpib qa ewzbp epqtm.」這段文句是以密鑰 8 來加密的，若改用密鑰 9 加密時是什麼？