
前言

誰該看這本書？

本書是為稍具 Java 開發經驗，且對密碼學有興趣的程式設計者所寫的。本書描繪 Java 密碼學程式開發的大概輪廓。如果你對密碼學絲毫沒有概念，不用擔心，本書第二章整章都在介紹密碼學入門的基本觀念，夠你閱讀接下來的整本書了。本書的主要目的是詳盡介紹 Java 程式語言中支援密碼學的類別和技術，讓你可以在你的應用程式設計中，加入密碼學技術，走在科技尖端。

這本書時時刻刻緊貼主題：Java 密碼學程式開發。如果你對密碼學的數學本身或是對其政治層面有興趣，請參考 Bruce Schneier 的應用 Applied Cryptography。雖然在第九章中，我會實作 ElGamal 編碼 (cipher) 及簽章 (signature) 演算法，但是我要示範的是 Java 程式設計，而不是密碼學的數學本身。同時，雖然我在第三章中，解釋了美國出口法裡對密碼學套件的分類管理方式，但我並不對其作任何評論，也不對一些細節做描述。關於密碼學的數學部份，有一本質量均霽的書：Alfred J. Menezes 的 Hand book of Applied Cryptography (CRC Press 出版)，供大家參考。最近一本政治議題性的密碼學專書則是 Whitfield Diffie 和 Susan Landau 的隱私線上：Privacy on the Line: The Politics of Wiretapping and Encryption (MIT Press 出版)。

如果你想快速上手 Java 程式設計，我建議你幾本 O'Reilly 的書：

- David Flanagan 的 *Java in a Nutshell* 給有經驗程式設計者的快速介紹。
- Pat Niemeyer 和 Joshua Peck 的 *Exploring Java* 一本給新手程式設計師循序漸進學習的入門指南。
- 要縱覽整個 Java 安全性應用程式界面，請參考 Scott Oaks 的 *Java Security*。

本書內容

本書的結構採用三明治教學法。第一章，第二章和最後一章是本書其它章節的理論基礎與前後文的鋪陳。中間的三到十二章則偏重於 Java 密碼學程式發展的方法上與實用上的描述，當中並有許多有用的範例程式。

- 第一章：介紹。描述密碼學在安全系統發展中的角色，並介紹幾個密碼學程式設計的簡短範例。
- 第二章：基本概念。介紹密碼學的基礎理論：ciphers、信息摘要、簽章和亂數。
- 第三章：架構。以鷹般的銳利觀點觀察 Java 密碼學軟體套件，並介紹 Java 安全性應用程式界面下層 Provider 的架構。
- 第四章：亂數。描述 Java 密碼學技術中所用的亂數。
- 第五章：Key 的管理。描述 JDK 中可以用來做 Key 管理的類別。
- 第六章：驗證。說明如何用信息摘要（Message Digest），簽章（Signature）和證明（Certificate）等的方法達成身份驗證的任務。
- 第七章：加密。含蓋了對稱性與非對稱性的 cipher、各種 cipher 模式及混雜系統等課題。
- 第八章：經過簽署的 Applet。描述如何建立簽署的 Applet。

- 第九章：撰寫 Provider 程式。描述如何撰寫提供安全性 API 的 Provider 程式。囊括了實作 ElGamal cipher 及簽章演算法的類別。
- 第十章：SafeTalk 程式。展現一個全功能的應用程式、一個支援密碼學技術的網路聊天程式。
- 第十一章：CipherMail 程式。另一個完整的程式、支援密碼學技術的客戶端電子郵件軟體。
- 第十二章：密碼學外一章。討論和密碼學無關的安全性議題，這些都是你應該要知道的。
- 附錄 A：BigInteger。討論 BigInteger 這個類別。它在實作密碼學演算法的運算上，十分有用。
- 附錄 B：Base64。說明不同基底的數字系統轉換的類別。
- 附錄 C：JAR。描述 JAR 工具程式。這個工具可以把組成 applet 或應用程式的所有檔案集成一個檔案，方便下載，並有檔案壓縮功能。
- 附錄 D：Jvakey。包含了 JDK 1.1 jvakey 工具。Jvakey 工具是拿來管理 Key 及證明資料庫的。
- 附錄 E：快查索引。包含本書討論範圍內的所有密碼學類別的索引。

本書未包含 . . .

- Class Loaders 類別
- 位元組碼 (bytecode) 解譯器
- SecurityManagers 類別
- 存取控制與許可權

以上的主題請參考 O'Reilly 出品的 Java Security 一書。

關於本書範例程式

版本

本書中的範例程式必需要用 JDK 1.2 (Java 開發工具, Java Developer's Kit) 及 JCE 1.2 (Java 加密編碼延伸函式庫, Java Cryptography Extension) 來編譯及執行。書中所列的這些範例程式都已經在 JDK 1.2beta3 及 JCE 1.2ea2 上測試過。書中某些主題亦可在 JDK 1.1 上實作, 特別是第五章中所提到的「以 Identity 為基礎的 Key 管理」及第六章中的信息摘要與簽章類別。不過, 由於所有和加密有關的東西都需要 JCE, 而 JCE 目前唯一的支援版本是只能搭配 JDK1.2 執行的 JCE 1.2。(雖然 JCE 曾有 1.1 的版本, 但是此版本的功能尚被局限在早期處理存取控制的階段, Sun 並不支援它, 而且現在你也不可能從任何一個網站下載 JCE 1.1)。

第八章中經過簽署的 applets, 必需在 HotJava 1.1, Netscape Navigator 及 IE 4.0 上執行。

檔案命名

這本書假定你已經熟悉 Java 程式的寫作, 並且對套件與 CLASSPATH 等也都頗有概念。本書範例程式的原始碼儲存時是以類別名稱為基準。請看下面的原始碼所舉的例子:

```
import java.applet.*;
import java.awt.*;

public class PrivilegedRenegade extends Applet {
    ...
}
```

上面的原始碼描述的是 PrivilegedRenegade 類別, 你應該將其儲存到名為 PrivilegedRenegade.java 的檔案中。

其它有一些類別會隸屬於某特定套件。舉個例子來說，以下是第九章裡一個範例程式的開頭：

```
import java.math.BigInteger;
import java.security.*;

public class ElGamalKeyPairGenerator
    extends KeyPairGenerator {
    ...
}
```

上面的程式碼應該被儲存到：

oreilly/jonathan/security/ElGamalKeyPairGenerator.java。

本書從頭到尾的範例程式，我都把它定義到 oreilly.jonathan.* 套件中。其中的一些範例程式會被其它的範例程式用到。為了讓這些程式能正確無誤的執行，你必須確認你的 CLASSPATH 設定中包含了包含了 oreilly 目錄所在的目錄。舉例來說，在我的電腦上，oreilly 這個目錄存在 c:\Jonathan\classes 目錄下。所以我的 CLASSPATH 設定便包含了 c:\Jonathan\classes 這個目錄；這將讓所有的 Java 應用程式都可以使用 oreilly.jonathan.* 階層裡的套件。

CLASSPATH

本書中有一些範例程式是由散布在各個檔案的類別所組成。在這種情況下，我並不直截了當的匯入該範例程式所需要的檔案。要讓這些檔案正確的編譯，你必需將你目前所在的目錄當作你 CLASSPATH 的一部份。舉例來說，我的 CLASSPATH 就包含了目前所在的目錄及 JCE。在我的 Windows 95 系統上，我將 autoexec.bat 檔裡的 CLASSPATH 設定如下：

```
set classpath=.
set classpath=%classpath%;c:\jdk1.2beta3\jce12-ea2-dom\jce12-ea2-dom.jar
```

變數命名

本書中的範例程式是我個人的程式風格所呈現的，是各個平台因循慣例的混合物。

我遵循標準 Java 程式寫作所用的大寫表示法。此外，類別的每一個成員變數都會用一個小寫的 m 開頭，如下：

```
protected int mPlainBlockSize;
```

上面的例子讓分辨成員變數與區域變數這件事變得很簡單。靜態變數則會以一個小寫的 s 開頭，如下：

```
protected static SecureRandom sRandom = null;
```

至於最終靜態成員變數則是以一個小寫的 k 開頭（它代表的是常數 constant，信不信由你）：

```
protected static final String kBanner = "SafeTalk v1.0";
```

陣列永遠都是在陣列型態的後面加一個中括號做為表示。這讓變數的型態資訊能夠集中在同一個地方：

```
byte[] ciphertext;
```

下載

本書中絕大部份的範例程式都可以從 <ftp://ftp.oreilly.com/pub/examples/java/crypto/> 目錄下載。另一些範例程式則不能合法地在網路上公開。因為美國政府將某些形式的加密軟體視為武器，嚴格管制這些軟體及程式原始碼的出口。我們的網頁伺服器上所放的所有東西都可以被世界各地任何一台連上網路的電腦下載。所以說，我們不能將某些範例程式的程式碼上網供大家下載。至於這本書本身由於受到第一次美國憲法修正條文的保護，得以自由出口。

感謝

我的妻子 Kristen 是目前我所認識的人裡，密碼學知識最豐富的。我要感謝她對這個計畫從頭到尾的鼓勵與熱忱，以及不遺餘力的校對。我也要感謝 Mike Loukides，它是第一個向我建議這本書的人，在本書創作的過程中也一直很耐心地給予我指導。我將永遠感謝 Mike 及 Frank Wilson，在我告訴他們我能寫作，並且希望在家工作時，他們能夠賦予我完全的信任。我也要感謝 Tim O'Reilly，他建立了一個有質感又正派的成功企業。

