Beginning Cryptography with Java™

David Hook
Beginning Cryptography with Java™
Beginning Cryptography with Java™

David Hook

Wiley Publishing, Inc.
About the Author

David Hook

David Hook has been writing software in a variety of domains and languages for the last 20 years. He has worked with Java since 1995, originally doing medical imaging before moving into cryptography and security a year or so later. In April 2000, he co-founded the open source Bouncy Castle cryptography project and has played an active role in it ever since. He currently works as a freelance consultant, mainly in Java, doing the odd bit of lecturing and writing on the side. When he is not using his spare time to work on Bouncy Castle, he spends it pursuing his other interest in computer graphics. He lives in Melbourne, Australia, with his most patient wife Janine and a cat named Hamlet, who really seems to think he’s a little cryptographer in a fur coat. David can be reached at dgh@bund.com.au.
Credits

Acquisitions Editor
Carol Long

Development Editor
Kezia Endsley

Production Editor
Angela Smith

Copy Editor
Joanne Slike

Editorial Manager
Mary Beth Wakefield

Vice President & Executive Group Publisher
Richard Swadley

Vice President and Publisher
Joseph B. Wikert

Project Coordinator
Erin Smith

Graphics and Production Specialists
April Farling
Denny Hager
Jennifer Heleine
Julie Trippetti

Quality Control Technician
Carl Pierce
Brian H. Walls

Proofreading and Indexing
TECHBOOKS Production Services
To FB and HC.
Acknowledgments

First of all, I’d like to thank Peter Grant for reviewing the chapters and exercises during the development of this book, Jon Eaves who also provided additional feedback, and Bernard Leach, who, with Peter and Jon, helped bring the Bouncy Castle project to life.

I would also like to thank Simon McMahon for additional feedback and comments, Jan Leuhe and Sharon Liu for answering my questions on early JCE history for this book, not to mention many of the other questions I’ve had over the years. Thanks must also go to all the people making up the Bouncy Castle user community — where the project has been successful; it is as much due to your feedback, comments, contributions, and patience, as to any other efforts.

To the Wrox crew, especially Carol Long and Carol Griffith for helping me get started and keeping me on track, and to my development editor Kezia Endsley, who never commented on the fact, that as a first time author, I clearly had no idea what I was doing. If this book provides you with all you expected, it is as much due to Kezia’s patient editing and direction as it is with any knowledge I have of the subject.

Finally, I would like to thank the members of my family. To my parents, Geoff and Pauline, brothers, Brendan, Martin, and Warwick, and my sister, Sarah, you have all been a constant source of inspiration and support. To my parents-in-law, Ron and Maureen, who quietly went about helping my wife Janine repaint our house while I was hiding in the office typing furiously, and to my wife Janine who found time to support me through the book as well, what can I possibly say? Thank you.
Contents

Acknowledgments ix
Introduction xxvii

Chapter 1: The JCA and the JCE 1

Basic Architecture 1
Provider Signing 4
Jurisdiction Policy Files 4
Installing the Unrestricted Policy Files 4
Troubleshooting Other Issues 7
How Do You Know the Policy Files Really Behave as Sun Says They Do? 7
Installing the Bouncy Castle Provider 7
Installing by Configuring the Java Runtime 8
Install the JAR File Containing the Provider 8
Enable the Provider by Adding It to the java.security File 8
Installing During Execution 10
How Provider Precedence Works 10
Examining the Capabilities of a Provider 12
Summary 13
Exercises 14

Chapter 2: Symmetric Key Cryptography 15

A First Example 15
A Basic Utility Class 16
The SecretKeySpec Class 19
The Cipher Class 19
Cipher.getInstance() 19
Cipher.init() 20
Cipher.update() 20
Cipher.doFinal() 20
Symmetric Block Cipher Padding 21
PKCS #5/PKCS #7 Padding 21
Other Padding Mechanisms 24
# Contents

<table>
<thead>
<tr>
<th>Symmetric Block Cipher Modes</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECB Mode</td>
<td>25</td>
</tr>
<tr>
<td>CBC Mode</td>
<td>26</td>
</tr>
<tr>
<td>Inline IVs</td>
<td>28</td>
</tr>
<tr>
<td>Creating an IV</td>
<td>30</td>
</tr>
<tr>
<td>Random IVs</td>
<td>31</td>
</tr>
<tr>
<td>Creating a SecureRandom Object</td>
<td>31</td>
</tr>
<tr>
<td>Pseudorandom IVs</td>
<td>32</td>
</tr>
<tr>
<td>A Look at Cipher Parameter Objects</td>
<td>34</td>
</tr>
<tr>
<td>The AlgorithmParameters Class</td>
<td>34</td>
</tr>
<tr>
<td>CTS Mode: A Special Case of CBC</td>
<td>34</td>
</tr>
<tr>
<td>Streaming Symmetric Block Cipher Modes</td>
<td>35</td>
</tr>
<tr>
<td>CTR Mode</td>
<td>35</td>
</tr>
<tr>
<td>OFB Mode</td>
<td>37</td>
</tr>
<tr>
<td>CFB Mode</td>
<td>38</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Symmetric Stream Ciphers</th>
<th>39</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generating Random Keys</td>
<td>40</td>
</tr>
<tr>
<td>The Key Interface</td>
<td>42</td>
</tr>
<tr>
<td>Key.getAlgorithm()</td>
<td>42</td>
</tr>
<tr>
<td>Key.getEncoded()</td>
<td>42</td>
</tr>
<tr>
<td>Key.getFormat()</td>
<td>42</td>
</tr>
<tr>
<td>The KeyGenerator Class</td>
<td>42</td>
</tr>
<tr>
<td>KeyGenerator.getInstance()</td>
<td>43</td>
</tr>
<tr>
<td>KeyGenerator.init()</td>
<td>43</td>
</tr>
<tr>
<td>KeyGenerator.generateKey()</td>
<td>43</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Password-Based Encryption</th>
<th>43</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic PBE</td>
<td>44</td>
</tr>
<tr>
<td>The Password</td>
<td>45</td>
</tr>
<tr>
<td>The Salt</td>
<td>45</td>
</tr>
<tr>
<td>The Iteration Count</td>
<td>45</td>
</tr>
<tr>
<td>PBE in the JCE</td>
<td>45</td>
</tr>
<tr>
<td>The PBEParameterSpec Class</td>
<td>48</td>
</tr>
<tr>
<td>The PBEKeySpec Class</td>
<td>48</td>
</tr>
<tr>
<td>The SecretKeyFactory Class</td>
<td>48</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Key Wrapping</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Doing Cipher-Based I/O</td>
<td>52</td>
</tr>
<tr>
<td>Summary</td>
<td>55</td>
</tr>
<tr>
<td>Exercises</td>
<td>55</td>
</tr>
</tbody>
</table>
Contents

Chapter 3: MessageDigests, MACs, and HMACs 57
  Getting Started 57
  The Problem of Tampering 60
  MessageDigests 62
    The MessageDigest Class 64
    MessageDigest.update() 65
    MessageDigest.digest() 65
    MessageDigest.isEqual() 65
  Tampering with the Digest 66
  MACs Based on Digests — the HMAC 68
    The Mac Class 71
      Mac.init() 71
      Mac.update() 71
      Mac.doFinal() 71
  MACs Based on Symmetric Ciphers 72
  Digests in Pseudorandom Functions 73
    PBE Key Generation 74
    Mask Generation 77
  Doing Digest-Based I/O 79
  Summary 81
  Exercises 82

Chapter 4: Asymmetric Key Cryptography 83
  Getting Started 84
  The PublicKey and PrivateKey Interfaces 85
  The RSA Algorithm 85
    The KeyFactory Class 88
    RSAPublicKeySpec and RSAPublicKey 88
    RSAPrivateKeySpec and RSAPrivateKey 89
    Creating Random RSA Keys 89
      The KeyPair Class 90
      The KeyPairGenerator Class 91
      The RSAKeyGenParameterSpec Class 91
    Improving RSA Performance 91
      Chinese Remainder Theorem 92
      RSAPrivateCrtKeySpec and RSAPrivateCrtKey 92
      Multi Prime Chinese Remainder Theorem 93