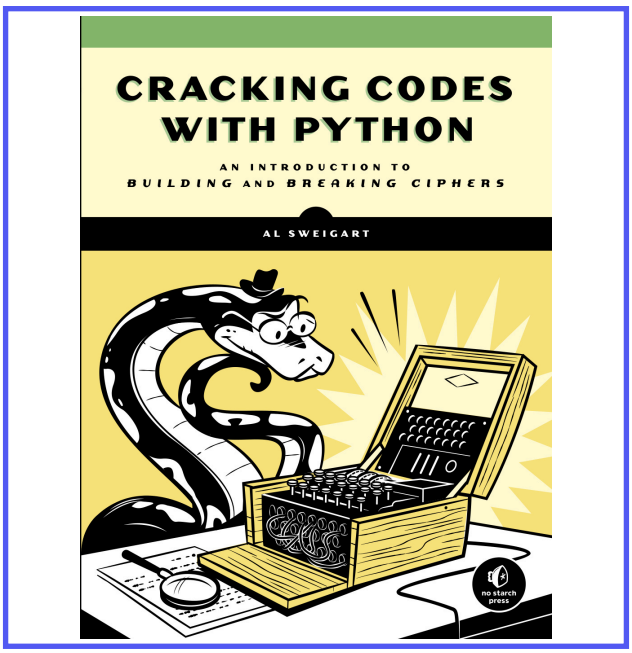


ebookbig.com

or scan the QR code to access the digital study portal.

Dedicated to Aaron Swartz, 1986–2013

"Aaron was part of an army of citizens that believes democracy only works when the citizenry are informed, when we know about our rights—and our obligations. An army that believes we must make justice and knowledge available to all—not just the well born or those that have grabbed the reins of power—so that we may govern ourselves more wisely. When I see our army, I see Aaron Swartz and my heart is broken. We have truly lost one of our better angels."

—Carl Malamud

About the Author

Al Sweigart is a software developer and tech book author living in San Francisco. Python is his favorite programming language, and he is the developer of several open source modules for it. His other books are freely available under a Creative Commons license on his website *https://inventwithpython.com/*. His cat weighs 12 pounds.

About the Technical Reviewers

Ari Lacenski creates mobile apps and Python software. She lives in Seattle.

Jean-Philippe Aumasson (Chapters 22–24) is Principal Research Engineer at Kudelski Security, Switzerland. He speaks regularly at information security conferences such as Black Hat, DEF CON, Troopers, and Infiltrate. He is the author of *Serious Cryptography* (No Starch Press, 2017).

BRIEF CONTENTS

Acknowledgments xix
Introduction xxi
Chapter 1: Making Paper Cryptography Tools. 1
Chapter 2: Programming in the Interactive Shell. 11
Chapter 3: Strings and Writing Programs 21
Chapter 4: The Reverse Cipher 39
Chapter 5: The Caesar Cipher. 53
Chapter 6: Hacking the Caesar Cipher with Brute-Force 69
Chapter 7: Encrypting with the Transposition Cipher 77
Chapter 8: Decrypting with the Transposition Cipher 99
Chapter 9: Programming a Program to Test Your Program 113
Chapter 10: Encrypting and Decrypting Files. 127
Chapter 11: Detecting English Programmatically 141
Chapter 12: Hacking the Transposition Cipher 161
Chapter 13: A Modular Arithmetic Module for the Affine Cipher 171
Chapter 14: Programming the Affine Cipher 185
Chapter 15: Hacking the Affine Cipher 197
Chapter 16: Programming the Simple Substitution Cipher 207
Chapter 17: Hacking the Simple Substitution Cipher 221
Chapter 18: Programming the Vigenère Cipher 247
Chapter 19: Frequency Analysis 259
Chapter 20: Hacking the Vigenère Cipher 279

Chapter 21: The One-Time Pad Cipher 315
Chapter 22: Finding and Generating Prime Numbers 321
Chapter 23: Generating Keys for the Public Key Cipher 335
Chapter 24: Programming the Public Key Cipher 349
Appendix: Debugging Python Code 375
Index 381

CONTENTS IN DETAIL

ACKNOWLEDGMENTS **xix**

INTRODUCTION **xxi**

Who Should Read This Book? . . . xxii
What's in This Book? . . . xxiii
How to Use This Book . . . xxiv
Typing Source Code . . . xxiv
Checking for Typos . . . xxv
Coding Conventions in This Book . . . xxv
Online Resources . . . xxv
Downloading and Installing Python . . . xxv
Windows Instructions . . . xxvi
macOS Instructions . . . xxvi
Ubuntu Instructions . . . xxvi
Downloading pyperclip.py . . . xxvi
Starting IDLE . . . xxvii
Summary . . . xxviii

1
MAKING PAPER CRYPTOGRAPHY TOOLS **1**

What Is Cryptography? . . . 2
Codes vs. Ciphers . . . 3
The Caesar Cipher . . . 4
The Cipher Wheel . . . 4
Encrypting with the Cipher Wheel . . . 5
Decrypting with the Cipher Wheel . . . 6
Encrypting and Decrypting with Arithmetic . . . 7
Why Double Encryption Doesn't Work . . . 8
Summary . . . 8
Practice Questions . . . 9

2
PROGRAMMING IN THE INTERACTIVE SHELL **11**

Some Simple Math Expressions . . . 12
Integers and Floating-Point Values . . . 13
Expressions . . . 13
Order of Operations . . . 14
Evaluating Expressions . . . 14
Storing Values with Variables . . . 15
Overwriting Variables . . . 17
Variable Names . . . 18
Summary . . . 18
Practice Questions . . . 19

STRINGS AND WRITING PROGRAMS **21**

Working with Text Using String Values 22
String Concatenation with the + Operator 23
String Replication with the * Operator 24
Getting Characters from Strings Using Indexes 24
Printing Values with the print() Function 27
Printing Escape Characters 28
Quotes and Double Quotes 29
Writing Programs in IDLE's File Editor 30
Source Code for the "Hello, World!" Program 31
Checking Your Source Code with the Online Diff Tool 31
Using IDLE to Access Your Program Later 32
Saving Your Program 32
Running Your Program 33
Opening the Programs You've Saved 34
How the "Hello, World!" Program Works 34
Comments 34
Printing Directions to the User 34
Taking a User's Input 35
Ending the Program 35
Summary 36
Practice Questions 37

4
THE REVERSE CIPHER **39**

Source Code for the Reverse Cipher Program 40
Sample Run of the Reverse Cipher Program 40
Setting Up Comments and Variables 41
Finding the Length of a String 41
Introducing the while Loop 42
The Boolean Data Type 43
Comparison Operators 43
Blocks 45
The while Loop Statement 46
"Growing" a String 47
Improving the Program with an input() Prompt 50
Summary 50
Practice Questions 51

5
THE CAESAR CIPHER **53**

Source Code for the Caesar Cipher Program 54
Sample Run of the Caesar Cipher Program 55
Importing Modules and Setting Up Variables 56
Constants and Variables 57
The for Loop Statement 58
An Example for Loop 58
A while Loop Equivalent of a for Loop 59

An Example if Statement 60
The else Statement. 60
The elif Statement 61
The in and not in Operators 61
The find() String Method 62
Encrypting and Decrypting Symbols 63
Handling Wraparound 64
Handling Symbols Outside of the Symbol Set 65
Displaying and Copying the Translated String 65
Encrypting Other Symbols 66
Summary 66
Practice Questions 67

6
HACKING THE CAESAR CIPHER WITH BRUTE-FORCE 69

Source Code for the Caesar Cipher Hacker Program 70
Sample Run of the Caesar Cipher Hacker Program 71
Setting Up Variables. 72
Looping with the range() Function. 72
Decrypting the Message 73
Using String Formatting to Display the Key and Decrypted Messages 75
Summary 76
Practice Question 76

7
ENCRYPTING WITH THE TRANSPOSITION CIPHER 77

How the Transposition Cipher Works 78
Encrypting a Message by Hand 79
Creating the Encryption Program. 80
Source Code for the Transposition Cipher Encryption Program. 81
Sample Run of the Transposition Cipher Encryption Program 82
Creating Your Own Functions with def Statements 82
Defining a Function that Takes Arguments with Parameters 83
Changes to Parameters Exist Only Inside the Function 84
Defining the main() Function 85
Passing the Key and Message As Arguments 86
The List Data Type 86
Reassigning the Items in Lists. 87
Lists of Lists 88
Using len() and the in Operator with Lists 89
List Concatenation and Replication with the + and * Operators 89
The Transposition Encryption Algorithm. 90
Augmented Assignment Operators 91
Moving currentIndex Through the Message 92
The join() String Method 93
Return Values and return Statements 94
A return Statement Example 94
Returning the Encrypted Ciphertext 95
The __name__ Variable 95
Summary 96
Practice Questions 97

DECRYPTING WITH THE TRANSPOSITION CIPHER 99

How to Decrypt with the Transposition Cipher on Paper . . . 100
Source Code for the Transposition Cipher Decryption Program . . . 101
Sample Run of the Transposition Cipher Decryption Program . . . 102
Importing Modules and Setting Up the main() Function . . . 102
Decrypting the Message with the Key . . . 103
The round(), math.ceil(), and math.floor() Functions . . . 103
The decryptMessage() Function . . . 104
Boolean Operators . . . 106
Adjusting the column and row Variables . . . 109
Calling the main() Function . . . 110
Summary . . . 110
Practice Questions . . . 111

9
PROGRAMMING A PROGRAM TO TEST YOUR PROGRAM 113

Source Code for the Transposition Cipher Tester Program . . . 114
Sample Run of the Transposition Cipher Tester Program . . . 115
Importing the Modules . . . 116
Creating Pseudorandom Numbers . . . 116
Creating a Random String . . . 118
Duplicating a String a Random Number of Times . . . 118
List Variables Use References . . . 119
Passing References . . . 121
Using copy.deepcopy() to Duplicate a List . . . 122
The random.shuffle() Function . . . 122
Randomly Scrambling a String . . . 123
Testing Each Message . . . 123
Checking Whether the Cipher Worked and Ending the Program . . . 124
Calling the main() Function . . . 124
Testing the Test Program . . . 125
Summary . . . 125
Practice Questions . . . 126

10
ENCRYPTING AND DECRYPTING FILES 127

Plain Text Files . . . 128
Source Code for the Transposition File Cipher Program . . . 128
Sample Run of the Transposition File Cipher Program . . . 130
Working with Files . . . 130
Opening Files . . . 131
Writing to and Closing Files . . . 131
Reading from a File . . . 132
Setting Up the main() Function . . . 132
Checking Whether a File Exists . . . 133
The os.path.exists() Function . . . 133
Checking Whether the Input File Exists with the os.path.exists() Function . . . 134
Using String Methods to Make User Input More Flexible . . . 134
The upper(), lower(), and title() String Methods . . . 134
The startswith() and endswith() String Methods . . . 135
Using These String Methods in the Program . . . 135

Measuring the Time It Took to Encrypt or Decrypt . . . 136
The time Module and time.time() Function . . . 136
Using the time.time() Function in the Program . . . 137
Writing the Output File . . . 137
Calling the main() Function . . . 138
Summary . . . 138
Practice Questions . . . 139

11
DETECTING ENGLISH PROGRAMMATICALLY 141

How Can a Computer Understand English? . . . 142
Source Code for the Detect English Module . . . 143
Sample Run of the Detect English Module . . . 145
Instructions and Setting Up Constants . . . 145
The Dictionary Data Type . . . 146
The Difference Between Dictionaries and Lists . . . 147
Adding or Changing Items in a Dictionary . . . 147
Using the len() Function with Dictionaries . . . 148
Using the in Operator with Dictionaries . . . 148
Finding Items Is Faster with Dictionaries than with Lists . . . 149
Using for Loops with Dictionaries . . . 149
Implementing the Dictionary File . . . 150
The split() Method . . . 150
Splitting the Dictionary File into Individual Words . . . 151
Returning the Dictionary Data . . . 151
Counting the Number of English Words in message . . . 152
Divide-by-Zero Errors . . . 152
Counting the English Word Matches . . . 153
The float(), int(), and str() Functions and Integer Division . . . 154
Finding the Ratio of English Words in the Message . . . 154
Removing Non-Letter Characters . . . 155
The append() List Method . . . 155
Creating a String of Letters . . . 156
Detecting English Words . . . 156
Using Default Arguments . . . 157
Calculating Percentages . . . 157
Summary . . . 159
Practice Questions . . . 160

12
HACKING THE TRANSPOSITION CIPHER 161

Source Code of the Transposition Cipher Hacker Program . . . 162
Sample Run of the Transposition Cipher Hacker Program . . . 163
Importing the Modules . . . 164
Multiline Strings with Triple Quotes . . . 164
Displaying the Results of Hacking the Message . . . 165
Getting the Hacked Message . . . 166
The strip() String Method . . . 167
Applying the strip() String Method . . . 168
Failing to Hack the Message . . . 168