

TABLE OF CONTENTS

About This Book	6
Table of Contents	7
Chapter 1 - Making Paper Cryptography Tools.....	1
What is Cryptography?	2
Codes vs. Ciphers	3
Making a Paper Cipher Wheel.....	4
A Virtual Cipher Wheel	7
How to Encrypt with the Cipher Wheel	8
How to Decrypt with the Cipher Wheel	9
A Different Cipher Tool: The St. Cyr Slide	10
Practice Exercises, Chapter 1, Set A	11
Doing Cryptography without Paper Tools	11
Practice Exercises, Chapter 1, Set B	13
Double-Strength Encryption?.....	13
Programming a Computer to do Encryption	14
Chapter 2 - Installing Python	16
Downloading and Installing Python.....	17
Downloading pyperclip.py.....	18
Starting IDLE.....	18
The Featured Programs	19
Line Numbers and Spaces.....	20
Text Wrapping in This Book	20
Tracing the Program Online.....	21
Checking Your Typed Code with the Online Diff Tool.....	21
Copying and Pasting Text	21
More Info Links	22
Programming and Cryptography.....	22
Chapter 3 - The Interactive Shell	26
Some Simple Math Stuff.....	26
Integers and Floating Point Values	27

Expressions	27
Order of Operations	28
Evaluating Expressions	29
Errors are Okay!	29
Practice Exercises, Chapter 3, Set A	30
Every Value has a Data Type	30
Storing Values in Variables with Assignment Statements	30
Overwriting Variables	32
Using More Than One Variable	33
Variable Names	34
Practice Exercises, Chapter 3, Set B	35
Summary - But When Are We Going to Start Hacking?	35
Chapter 4 - Strings and Writing Programs	36
Strings	36
String Concatenation with the + Operator	38
String Replication with the * Operator	39
Printing Values with the print () Function	39
Escape Characters	40
Quotes and Double Quotes	41
Practice Exercises, Chapter 4, Set A	42
Indexing	42
Negative Indexes	43
Slicing	44
Blank Slice Indexes	45
Practice Exercises, Chapter 4, Set B	46
Writing Programs in IDLE's File Editor	46
Hello World!	47
Source Code of Hello World	47
Saving Your Program	47
Running Your Program	48
Opening The Programs You've Saved	49
How the "Hello World" Program Works	50
Comments	50
Functions	50

The <code>print()</code> function	51
The <code>input()</code> function	51
Ending the Program	52
Practice Exercises, Chapter 4, Set C	52
Summary	52
Chapter 5 - The Reverse Cipher.....	53
The Reverse Cipher.....	53
Source Code of the Reverse Cipher Program.....	54
Sample Run of the Reverse Cipher Program.....	54
Checking Your Source Code with the Online Diff Tool	55
How the Program Works.....	55
The <code>len()</code> Function	56
Introducing the <code>while</code> Loop.....	57
The Boolean Data Type	58
Comparison Operators	58
Conditions.....	61
Blocks	61
The <code>while</code> Loop Statement	62
“Growing” a String	62
Tracing Through the Program, Step by Step	65
Using <code>input()</code> In Our Programs.....	67
Practice Exercises, Chapter 5, Section A	67
Summary	68
Chapter 6 - The Caesar Cipher.....	69
Implementing a Program.....	69
Source Code of the Caesar Cipher Program.....	70
Sample Run of the Caesar Cipher Program.....	71
Checking Your Source Code with the Online Diff Tool	72
Practice Exercises, Chapter 6, Set A	72
How the Program Works.....	72
Importing Modules with the <code>import</code> Statement.....	72
Constants.....	73
The <code>upper()</code> and <code>lower()</code> String Methods	74
The <code>for</code> Loop Statement.....	75

A while Loop Equivalent of a for Loop.....	76
Practice Exercises, Chapter 6, Set B	77
The if Statement	77
The else Statement.....	78
The elif Statement.....	78
The in and not in Operators.....	79
The find() String Method.....	80
Practice Exercises, Chapter 6, Set C	81
Back to the Code.....	81
Displaying and Copying the Encrypted/Decrypted String	83
Encrypt Non-Letter Characters	84
Summary.....	85
Chapter 7 - Hacking the Caesar Cipher with the Brute-Force Technique.....	87
Hacking Ciphers	87
The Brute-Force Attack	88
Source Code of the Caesar Cipher Hacker Program	88
Sample Run of the Caesar Cipher Hacker Program	89
How the Program Works.....	90
The range() Function	90
Back to the Code.....	92
String Formatting.....	93
Practice Exercises, Chapter 7, Set A	94
Summary.....	94
Chapter 8 - Encrypting with the Transposition Cipher	95
Encrypting with the Transposition Cipher	95
Practice Exercises, Chapter 8, Set A	97
A Transposition Cipher Encryption Program.....	97
Source Code of the Transposition Cipher Encryption Program	97
Sample Run of the Transposition Cipher Encryption Program	98
How the Program Works.....	99
Creating Your Own Functions with def Statements.....	99
The Program's main() Function	100
Parameters.....	101
Variables in the Global and Local Scope	103

The global Statement.....	103
Practice Exercises, Chapter 8, Set B	105
The List Data Type	105
Using the list () Function to Convert Range Objects to Lists.....	107
Reassigning the Items in Lists.....	108
Reassigning Characters in Strings.....	108
Lists of Lists	108
Practice Exercises, Chapter 8, Set C	109
Using len () and the in Operator with Lists	109
List Concatenation and Replication with the + and * Operators.....	110
Practice Exercises, Chapter 8, Set D	111
The Transposition Encryption Algorithm	111
Augmented Assignment Operators	113
Back to the Code.....	114
The join () String Method.....	116
Return Values and return Statements	117
Practice Exercises, Chapter 8, Set E	117
Back to the Code.....	118
The Special __name__ Variable.....	118
Key Size and Message Length	119
Summary	119
Chapter 9 - Decrypting with the Transposition Cipher	121
Decrypting with the Transposition Cipher on Paper	122
Practice Exercises, Chapter 9, Set A	123
A Transposition Cipher Decryption Program.....	124
Source Code of the Transposition Cipher Decryption Program	124
How the Program Works.....	125
The math.ceil (), math.floor () and round () Functions.....	126
The and and or Boolean Operators.....	130
Practice Exercises, Chapter 9, Set B	131
Truth Tables.....	131
The and and or Operators are Shortcuts	132
Order of Operations for Boolean Operators	133
Back to the Code.....	133

Practice Exercises, Chapter 9, Set C	135
Summary	135
Chapter 10 - Programming a Program to Test Our Program	136
Source Code of the Transposition Cipher Tester Program	137
Sample Run of the Transposition Cipher Tester Program	138
How the Program Works	139
Pseudorandom Numbers and the <code>random.seed()</code> Function	139
The <code>random.randint()</code> Function	141
References	141
The <code>copy.deepcopy()</code> Functions	145
Practice Exercises, Chapter 10, Set A	146
The <code>random.shuffle()</code> Function	146
Randomly Scrambling a String	147
Back to the Code	147
The <code>sys.exit()</code> Function	148
Testing Our Test Program	149
Summary	150
Chapter 11 - Encrypting and Decrypting Files	151
Plain Text Files	152
Source Code of the Transposition File Cipher Program	152
Sample Run of the Transposition File Cipher Program	155
Reading From Files	155
Writing To Files	156
How the Program Works	157
The <code>os.path.exists()</code> Function	158
The <code>startswith()</code> and <code>endswith()</code> String Methods	159
The <code>title()</code> String Method	160
The <code>time</code> Module and <code>time.time()</code> Function	161
Back to the Code	162
Practice Exercises, Chapter 11, Set A	163
Summary	163
Chapter 12 - Detecting English Programmatically	164
How Can a Computer Understand English?	165
Practice Exercises, Chapter 12, Section A	167

The Detect English Module	167
Source Code for the Detect English Module.....	167
How the Program Works.....	168
Dictionaries and the Dictionary Data Type	169
Adding or Changing Items in a Dictionary	170
Practice Exercises, Chapter 12, Set B	171
Using the len () Function with Dictionaries.....	171
Using the in Operator with Dictionaries.....	171
Using for Loops with Dictionaries	172
Practice Exercises, Chapter 12, Set C	172
The Difference Between Dictionaries and Lists.....	172
Finding Items is Faster with Dictionaries Than Lists	172
The split () Method	173
The None Value	174
Back to the Code.....	175
“Divide by Zero” Errors.....	176
The float (), int (), and str () Functions and Integer Division	177
Practice Exercises, Chapter 12, Set D	178
Back to the Code.....	178
The append() List Method.....	179
Default Arguments.....	180
Calculating Percentage.....	181
Practice Exercises, Chapter 12, Set E	182
Summary	182
Chapter 13 - Hacking the Transposition Cipher	184
Source Code of the Transposition Cipher Hacker Program	184
Sample Run of the Transposition Breaker Program	186
How the Program Works.....	187
Multi-line Strings with Triple Quotes	187
Back to the Code.....	188
The strip () String Method	190
Practice Exercises, Chapter 13, Set A	192
Summary	192
Chapter 14 - Modular Arithmetic with the Multiplicative and Affine Ciphers	193

Oh No Math!	194
Math Oh Yeah!	194
Modular Arithmetic (aka Clock Arithmetic).....	194
The % Mod Operator	196
Practice Exercises, Chapter 14, Set A	196
GCD: Greatest Common Divisor (aka Greatest Common Factor)	196
Visualize Factors and GCD with Cuisenaire Rods.....	197
Practice Exercises, Chapter 14, Set B	199
Multiple Assignment.....	199
Swapping Values with the Multiple Assignment Trick.....	200
Euclid's Algorithm for Finding the GCD of Two Numbers.....	200
"Relatively Prime"	202
Practice Exercises, Chapter 14, Set C	202
The Multiplicative Cipher.....	202
Practice Exercises, Chapter 14, Set D	204
Multiplicative Cipher + Caesar Cipher = The Affine Cipher	204
The First Affine Key Problem.....	204
Decrypting with the Affine Cipher.....	205
Finding Modular Inverses	206
The // Integer Division Operator	207
Source Code of the cryptomath Module.....	207
Practice Exercises, Chapter 14, Set E	208
Summary.....	208
Chapter 15 - The Affine Cipher	210
Source Code of the Affine Cipher Program	211
Sample Run of the Affine Cipher Program	213
Practice Exercises, Chapter 15, Set A	213
How the Program Works.....	213
Splitting One Key into Two Keys	215
The Tuple Data Type	215
Input Validation on the Keys	216
The Affine Cipher Encryption Function	217
The Affine Cipher Decryption Function	218
Generating Random Keys	219

The Second Affine Key Problem: How Many Keys Can the Affine Cipher Have?	220
Summary	222
Chapter 16 - Hacking the Affine Cipher	223
Source Code of the Affine Cipher Hacker Program	223
Sample Run of the Affine Cipher Hacker Program	225
How the Program Works	225
The Affine Cipher Hacking Function	227
The ** Exponent Operator	227
The continue Statement	228
Practice Exercises, Chapter 16, Set A	231
Summary	231
Chapter 17 - The Simple Substitution Cipher	232
The Simple Substitution Cipher with Paper and Pencil	233
Practice Exercises, Chapter 17, Set A	233
Source Code of the Simple Substitution Cipher	234
Sample Run of the Simple Substitution Cipher Program	236
How the Program Works	236
The Program's main() Function	237
The sort() List Method	238
Wrapper Functions	239
The Program's translateMessage() Function	240
The isupper() and islower() String Methods	242
Practice Exercises, Chapter 17, Set B	244
Generating a Random Key	244
Encrypting Spaces and Punctuation	244
Practice Exercises, Chapter 17, Set C	246
Summary	246
Chapter 18 - Hacking the Simple Substitution Cipher	247
Computing Word Patterns	248
Getting a List of Candidates for a Cipherword	249
Practice Exercises, Chapter 18, Set A	250
Source Code of the Word Pattern Module	250
Sample Run of the Word Pattern Module	252
How the Program Works	253

The pprint.pprint() and pprint.pformat() Functions	253
Building Strings in Python with Lists	254
Calculating the Word Pattern	255
The Word Pattern Program's main() Function	256
Hacking the Simple Substitution Cipher	258
Source Code of the Simple Substitution Hacking Program	259
Hacking the Simple Substitution Cipher (in Theory)	262
Explore the Hacking Functions with the Interactive Shell	263
How the Program Works	268
Import All the Things	268
A Brief Intro to Regular Expressions and the sub() Regex Method	269
The Hacking Program's main() Function	270
Partially Hacking the Cipher	270
Blank Cipherletter Mappings	272
Adding Letters to a Cipherletter Mapping	272
Intersecting Two Letter Mappings	274
Removing Solved Letters from the Letter Mapping	275
Hacking the Simple Substitution Cipher	277
Creating a Key from a Letter Mapping	280
Couldn't We Just Encrypt the Spaces Too?	282
Summary	282
Chapter 19 - The Vigenère Cipher	283
Le Chiffre Indéchiffrable	284
Multiple "Keys" in the Vigenère Key	284
Source Code of Vigenère Cipher Program	287
Sample Run of the Vigenère Cipher Program	290
How the Program Works	290
Summary	294
Chapter 20 - Frequency Analysis	295
The Code for Matching Letter Frequencies	300
How the Program Works	302
The Most Common Letters, "ETAOIN"	303
The Program's getLettersCount() Function	303
The Program's getItemAtIndexZero() Function	304

The Program's <code>getFrequencyOrder()</code> Function.....	304
The <code>sort()</code> Method's <code>key</code> and <code>reverse</code> Keyword Arguments	306
Passing Functions as Values	307
Converting Dictionaries to Lists with the <code>keys()</code> , <code>values()</code> , <code>items()</code> Dictionary Methods	309
Sorting the Items from a Dictionary.....	310
The Program's <code>englishFreqMatchScore()</code> Function	311
Summary.....	312
Chapter 21 - Hacking the Vigenère Cipher	313
The Dictionary Attack.....	314
Source Code for a Vigenère Dictionary Attack Program	314
Sample Run of the Vigenère Dictionary Hacker Program	315
The <code>readlines()</code> File Object Method.....	316
The Babbage Attack & Kasiski Examination.....	316
Kasiski Examination, Step 1 – Find Repeat Sequences' Spacings.....	316
Kasiski Examination, Step 2 – Get Factors of Spacings	317
Get Every Nth Letters from a String	318
Frequency Analysis.....	318
Brute-Force through the Possible Keys.....	320
Source Code for the Vigenère Hacking Program.....	321
Sample Run of the Vigenère Hacking Program	327
How the Program Works.....	329
Finding Repeated Sequences	330
Calculating Factors	332
Removing Duplicates with the <code>set()</code> Function	333
The Kasiski Examination Algorithm.....	336
The <code>extend()</code> List Method.....	336
The <code>end</code> Keyword Argument for <code>print()</code>	342
The <code>itertools.product()</code> Function.....	343
The <code>break</code> Statement	347
Practice Exercises, Chapter 21, Set A	348
Modifying the Constants of the Hacking Program	349
Summary.....	350
Chapter 22 - The One-Time Pad Cipher	351
The Unbreakable One-Time Pad Cipher.....	352

Why the One-Time Pad is Unbreakable.....	352
Beware Pseudorandomness.....	353
Beware the Two-Time Pad	353
The Two-Time Pad is the Vigenère Cipher.....	354
Practice Exercises, Chapter 22, Set A	355
Summary.....	355
Chapter 23 - Finding Prime Numbers.....	356
Prime Numbers	357
Composite Numbers.....	358
Source Code for The Prime Sieve Module.....	358
How the Program Works.....	359
How to Calculate if a Number is Prime	360
The Sieve of Eratosthenes.....	361
The primeSieve() Function.....	363
Detecting Prime Numbers.....	364
Source Code for the Rabin-Miller Module.....	365
Sample Run of the Rabin Miller Module	367
How the Program Works.....	367
The Rabin-Miller Primality Algorithm	367
The New and Improved isPrime() Function	368
Summary.....	370
Chapter 24 - Public Key Cryptography and the RSA Cipher.....	372
Public Key Cryptography.....	373
The Dangers of “Textbook” RSA	375
A Note About Authentication	375
The Man-In-The-Middle Attack	376
Generating Public and Private Keys.....	377
Source Code for the RSA Key Generation Program	377
Sample Run of the RSA Key Generation Program	379
How the Key Generation Program Works	380
The Program’s generateKey() Function.....	381
RSA Key File Format	383
Hybrid Cryptosystems	384
Source Code for the RSA Cipher Program	385

Sample Run of the RSA Cipher Program.....	389
Practice Exercises, Chapter 24, Set A	390
Digital Signatures	391
How the RSA Cipher Program Works	392
ASCII: Using Numbers to Represent Characters	394
The chr () and ord () Functions	394
Practice Exercises, Chapter 24, Set B	395
Blocks	395
Converting Strings to Blocks with getBlocksFromText ()	398
The encode () String Method and the Bytes Data Type	399
The bytes () Function and decode () Bytes Method	399
Practice Exercises, Chapter 24, Set C	400
Back to the Code.....	400
The min () and max () Functions	401
The insert () List Method.....	404
The Mathematics of RSA Encrypting and Decrypting.....	405
The pow () Function	405
Reading in the Public & Private Keys from their Key Files.....	407
The Full RSA Encryption Process	407
The Full RSA Decryption Process	410
Practice Exercises, Chapter 24, Set D	411
Why Can't We Hack the RSA Cipher.....	412
Summary.....	414
About the Author	416