

《Python密码学编程》

图书基本信息

书名：《Python密码学编程》

13位ISBN编号：9787115424292

出版时间：2016-8-1

作者：【美】Al Sweigart

页数：324

译者：李永伦

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu111.com

《Python密码学编程》

内容概要

本书主要介绍了加密算法，同时从Python编程的角度来引导读者将加密算法更好地实现。书中不仅讲述了详细的算法理论，还附以详细的代码示例帮助读者更好地学习算法，并最终实现加密算法。除此之外，书中还提供了相应的源码下载资源，能够让读者获取到最完整真实的代码示例，帮助读者更好地进行后续的开发和学习。

《Python密码学编程》

作者简介

Al Sweigart是加利福尼亚州旧金山的一名软件开发者。他很喜欢骑自行车、当志愿者、泡咖啡吧以及开发有用的软件。他编写了《Python游戏编程快速上手》《Python和Pygame游戏开发指南》《Python密码学编程》《Python编程快速上手——让繁琐工作自动化》等图书，深受读者欢迎。他生于德克萨斯的休斯顿。他在德克萨斯大学Austin分校读完了计算机科学学位。

书籍目录

第1章 制作纸质加密工具	1
1.1 密码学是什么	1
1.2 代码与加密法	2
1.3 制作纸质加密轮盘	2
1.4 虚拟加密轮盘	4
1.5 如何使用加密轮盘加密	4
1.6 如何使用加密轮盘解密	5
1.7 另一个加密法工具：St. Cyr滑条	6
1.8 A组练习	6
1.9 不用纸质工具做加密	7
1.10 B组练习	9
1.11 双重强度加密	9
1.12 通过计算机编程进行加密	9
第2章 Pygame基础知识	11
2.1 下载和安装Python	11
2.1.1 Windows安装步骤	11
2.1.2 OS X安装步骤	12
2.1.3 Ubuntu和Linux安装步骤	12
2.2 下载pyperclip.py	12
2.3 启动IDLE	12
2.4 特色程序	13
2.5 行号和空格	14
2.6 本书的文本换行	14
2.7 在线跟踪程序	15
2.8 使用在线比较工具检查输入的 代码	15
2.9 复制粘贴文本	15
2.10 更多信息链接	15
2.11 编程和密码学	16
第3章 Interactive Shell	20
3.1 一些简单的数学知识	20
3.2 整数和浮点数	21
3.3 表达式	21
3.4 运算符顺序	22
3.5 计算表达式	22
3.6 错误是可以接受的	22
3.7 A组练习	23
3.8 每个值都有一个数据类型	23
3.9 通过赋值语句把值存到变量里	23
3.10 重写变量	24
3.11 使用多个变量	25
3.12 变量名	26
3.13 驼峰式大小写	26
3.14 B组练习	26
3.15 总结——那我们什么时候开始 破译	26
第4章 字符串和写程序	28

- 4.1 字符串 28
- 4.2 使用+运算符的字符串连接 29
- 4.3 使用*运算符的字符串复制 30
- 4.4 使用print()函数输出值 30
- 4.5 转义字符 31
- 4.6 引号和双引号 32
- 4.7 A组练习 32
- 4.8 索引操作 33
- 4.9 负索引 33
- 4.10 分片操作 34
- 4.11 空分片索引 35
- 4.12 B组练习 35
- 4.13 在IDLE的文件编辑器里写程序 35
- 4.14 Hello World! 36
- 4.15 Hello World的源代码 36
- 4.16 保存你的程序 37
- 4.17 运行你的程序 37
- 4.18 打开你保存的程序 38
- 4.19 “Hello World”程序如何工作 38
- 4.20 注释 38
- 4.21 函数 39
- 4.22 print()函数 39
- 4.23 input()函数 39
- 4.24 结束程序 40
- 4.25 C组练习 40
- 4.26 总结 40
- 第5章 反转加密 41
 - 5.1 反转加密 41
 - 5.2 反转加密程序的源代码 41
 - 5.3 运行反转加密程序 42
 - 5.4 用在线比较工具检查你的源代码 42
 - 5.5 这个程序如何工作 43
 - 5.6 len()函数 43
 - 5.7 while循环简介 44
 - 5.8 布尔数据类型 44
 - 5.9 比较运算符 45
 - 5.10 条件 46
 - 5.11 代码块 47
 - 5.12 while循环语句 47
 - 5.13 “增长”一个字符串 48
 - 5.14 一步一步跟踪程序 50
 - 5.15 在我们的程序里使用input() 52
 - 5.16 A组练习 52
 - 5.17 总结 52
- 第6章 凯撒加密法 53
 - 6.1 实现程序 53
 - 6.2 凯撒加密程序的源代码 53
 - 6.3 运行凯撒加密程序 54
 - 6.4 使用在线比较工具检查你的

源代码	55
6.5 A组练习	55
6.6 这个程序如何工作	55
6.7 使用import语句导入模块	55
6.8 常量	56
6.9 upper()和lower()字符串方法	57
6.10 for循环语句	58
6.11 相当于for循环的while循环	59
6.12 B组练习	59
6.13 if语句	59
6.14 else语句	60
6.15 elif语句	60
6.16 in和not in运算符	61
6.17 find()字符串方法	62
6.18 C组练习	62
6.19 回到代码	62
6.20 显示和复制加密/解密之后的字符串	64
6.21 加密非字母字符	65
6.22 总结	66
第7章 暴力破译凯撒加密法	67
7.1 破译加密	67
7.2 暴力破译	67
7.3 凯撒加密法破译程序的源代码	68
7.4 运行凯撒加密法破译程序	68
7.5 这个程序如何工作	69
7.6 range()函数	69
7.7 回到代码	70
7.8 字符串格式化	72
7.9 A组练习	72
7.10 总结	72
第8章 使用换位加密法加密	73
8.1 换位加密法	73
8.2 A组练习	74
8.3 换位加密法加密程序	74
8.4 换位加密法加密程序的源代码	75
8.5 运行换位加密法加密程序	76
8.6 这个程序如何工作	76
8.7 使用def语句创建你自己的函数	76
8.8 程序的main()函数	77
8.9 形参	78
8.10 对形参的修改只存在于函数之内	79
8.11 全局作用域和本地作用域里的变量	79
8.12 global语句	79
8.13 B组练习	81
8.14 列表数据类型	81
8.15 使用list()函数把区间对象转换	

- 成列表 82
- 8.16 重新赋值列表里的项 83
- 8.17 重新赋值字符串里的字符 83
- 8.18 列表的列表 83
- 8.19 C组练习 84
- 8.20 在列表上使用len()和in运算符 84
- 8.21 使用+和*运算符的列表连接和复制 85
- 8.22 D组练习 85
- 8.23 换位加密算法 85
- 8.24 增强赋值运算符 88
- 8.25 回到代码 88
- 8.26 join()字符串方法 90
- 8.27 返回值和return语句 91
- 8.28 E组练习 91
- 8.29 回到代码 91
- 8.30 特殊的__name__变量 92
- 8.31 密钥的大小和消息的长度 93
- 8.32 总结 93
- 第9章 使用换位加密法解密 94
- 9.1 在纸上使用换位加密法解密 94
- 9.2 练习A组 96
- 9.3 换位加密法解密程序 96
- 9.4 换位加密法解密程序的源代码 96
- 9.5 这个程序如何工作 97
- 9.6 math.ceil()、math.floor()和round()函数 98
- 9.7 and和or布尔运算符 101
- 9.8 B组练习 102
- 9.9 真值表 102
- 9.10 and和or运算符可以简化代码 103
- 9.11 布尔运算符的运算顺序 103
- 9.12 回到代码 103
- 9.13 C组练习 105
- 9.14 总结 105
- 第10章 写一个程序测试我们的程序 106
- 10.1 换位加密法测试程序的源代码 106
- 10.2 运行换位加密法测试程序 107
- 10.3 这个程序如何工作 108
- 10.4 伪随机数和random.seed()函数 108
- 10.5 random.randint()函数 109
- 10.6 引用 110
- 10.7 copy.deepcopy()函数 112
- 10.8 A组练习 112
- 10.9 random.shuffle()函数 112
- 10.10 随机打乱一个字符串 113
- 10.11 回到代码 114
- 10.12 sys.exit()函数 114
- 10.13 测试我们的测试程序 115

- 10.14 总结 116
- 第11章 加密和解密文件 117
 - 11.1 纯文本文件 117
 - 11.2 换位加密法文件加密程序的源代码 118
 - 11.3 运行换位加密法文件加密程序 120
 - 11.4 读取文件 120
 - 11.4.1 open()函数和文件对象 120
 - 11.4.2 read()文件对象方法 120
 - 11.4.3 close()文件对象方法 121
 - 11.5 写入文件 121
 - 11.5.1 write()文件对象方法 122
 - 11.6 这个程序如何工作 122
 - 11.7 os.path.exists()函数 123
 - 11.8 startswith()和endswith()字符串方法 123
 - 11.9 title()字符串方法 124
 - 11.10 time模块和time.time()函数 125
 - 11.11 回到代码 126
 - 11.12 A组练习 126
 - 11.13 总结 127
- 第12章 通过编程检测英文 128
 - 12.1 计算机如何理解英文 128
 - 12.2 A组练习 130
 - 12.3 检测英文模块 130
 - 12.4 检测英文模块的源代码 130
 - 12.5 这个程序如何工作 131
 - 12.6 词典和词典数据类型 132
 - 12.7 添加或修改词典里的项 132
 - 12.8 B组练习 133
 - 12.9 在词典上使用len()函数 133
 - 12.10 在词典上使用in运算符 133
 - 12.11 在词典上使用for循环 134
 - 12.12 C组练习 134
 - 12.13 词典与列表之间的区别 134
 - 12.14 在词典上查找项比在列表上更快 135
 - 12.15 split()方法 135
 - 12.16 None值 136
 - 12.17 回到代码 136
 - 12.18 “除以零”错误 138
 - 12.19 float()、int()和str()函数以及整数除法 138
 - 12.20 D组练习 139
 - 12.21 回到代码 139
 - 12.22 append()列表方法 139
 - 12.23 默认参数值 140
 - 12.24 计算比例 141
 - 12.25 E组练习 142
 - 12.26 总结 143

- 第13章 破译换位加密法 144
 - 13.1 换位加密法破译程序的源代码 144
 - 13.2 运行换位加密法破译程序 145
 - 13.3 这个程序如何工作 146
 - 13.4 使用三引号的多行字符串 146
 - 13.5 回到代码 147
 - 13.6 strip()字符串方法 148
 - 13.7 A组练习 150
 - 13.8 总结 150
- 第14章 取模运算与乘数加密法和仿射加密法 151
 - 14.1 噢，不，数学！ 151
 - 14.2 数学，噢耶！ 151
 - 14.3 取模运算（又名时钟运算） 151
 - 14.4 取模运算符% 152
 - 14.5 A组练习 153
 - 14.6 GCD：最大公约数（又名最大公因数） 153
 - 14.7 使用古氏积木（Cuisenaire rods）可视化因数和GCD 154
 - 14.8 B组练习 155
 - 14.9 多重赋值 155
 - 14.10 通过多重赋值交换值 156
 - 14.11 找出两个数字的GCD的欧几里得算法 156
 - 14.12 “互质” 157
 - 14.13 C组练习 157
 - 14.14 乘数加密法 157
 - 14.15 D组练习 159
 - 14.16 乘数加密法 + 凯撒加密法 = 仿射加密法 159
 - 14.17 仿射密钥的第一个问题 159
 - 14.18 使用仿射加密法解密 160
 - 14.19 找出模逆 161
 - 14.20 //整数除法运算符 161
 - 14.21 cryptomath模块的源代码 162
 - 14.22 E组练习 163
 - 14.23 总结 163
- 第15章 仿射加密法 164
 - 15.1 仿射加密法程序的源代码 164
 - 15.2 运行仿射加密法程序 166
 - 15.3 A组练习 166
 - 15.4 这个程序如何工作 166
 - 15.5 把一个密钥分成两个密钥 167
 - 15.6 元祖数据类型 168
 - 15.7 密钥的输入验证 168
 - 15.8 仿射加密法加密函数 169
 - 15.9 仿射加密法解密函数 170
 - 15.10 生成随机密钥 171
 - 15.11 仿射密钥的第二个问题：仿射加密法可以有多少个密钥 172

- 15.12 总结 173
- 第16章 破译仿射加密法 174
 - 16.1 仿射加密法破译程序的源代码 174
 - 16.2 运行仿射加密法破译程序 175
 - 16.3 这个程序如何工作 176
 - 16.4 仿射加密法破译函数 177
 - 16.5 **指数运算符 177
 - 16.6 continue语句 178
 - 16.7 A组练习 180
 - 16.8 总结 180
- 第17章 简单替代加密法 181
 - 17.1 使用纸笔实现简单替代加密法 181
 - 17.2 A组练习 182
 - 17.3 简单替代加密法的源代码 182
 - 17.4 运行简单替代加密法程序 183
 - 17.5 这个程序如何工作 184
 - 17.6 程序的main()函数 184
 - 17.7 sort()列表方法 185
 - 17.8 包装器函数 186
 - 17.9 程序的translateMessage()函数 187
 - 17.10 isupper()和islower()字符串方法 189
 - 17.11 B组练习 190
 - 17.12 生成随机密钥 190
 - 17.13 加密空格和标点符号 191
 - 17.14 C组练习 191
 - 17.15 总结 192
- 第18章 破译简单替代加密法 193
 - 18.1 计算单词模式 193
 - 18.2 获取密词的候选单词列表 194
 - 18.3 A组练习 195
 - 18.4 单词模式模块的源代码 195
 - 18.5 运行单词模式模块 196
 - 18.6 这个程序如何工作 197
 - 18.7 pprint.pprint()和pprint.pformat()函数 197
 - 18.8 在Python里使用列表创建字符串 198
 - 18.9 计算单词模式 199
 - 18.10 单词模式程序的main()函数 200
 - 18.11 破译简单替代加密法 202
 - 18.12 简单替代破译程序的源代码 202
 - 18.13 破译简单替代加密法(理论) 205
 - 18.14 使用Interactive Shell探索破译函数 205
 - 18.15 这个程序如何工作 209
 - 18.16 导入所有东西 209

- 18.17 正则表达式和sub()正则方法
简介 210
- 18.18 破译程序的main()函数 211
- 18.19 部分破译加密法 211
- 18.20 空密字映射 212
- 18.21 把字母添加到密字映射 213
- 18.22 计算两个字母映射的交集 214
- 18.23 从密字映射移除已经破译的字母 215
- 18.24 破译简单替代加密法 217
- 18.25 从密字映射创建密钥 219
- 18.26 我们不能把空格也加密吗 220
- 18.27 总结 220
- 第19章 维吉尼亚加密法 221
 - 19.1 不可破译的加密法 221
 - 19.2 维吉尼亚密钥里的多个“密钥” 221
 - 19.3 维吉尼亚加密法程序的源代码 224
 - 19.4 运行维吉尼亚加密法程序 226
 - 19.5 这个程序如何工作 227
 - 19.6 总结 230
- 第20章 频率分析 231
 - 20.1 字母频率和ETAOIN 231
 - 20.1.1 匹配字母频率 232
 - 20.1.2 计算频率匹配分值的例子 233
 - 20.1.3 另一个计算频率匹配分值的例子 233
 - 20.1.4 破译每个子密钥 234
 - 20.2 匹配字母频率的代码 234
 - 20.3 这个程序如何工作 236
 - 20.4 最常见的字母“ETAOIN” 237
 - 20.5 这个程序的getLettersCount()函数 237
 - 20.6 这个程序的getItemAtIndexZero()函数 238
 - 20.7 这个程序的getFrequencyOrder()函数 238
 - 20.8 sort()方法的关键字和reverse关键字参数 239
 - 20.9 把函数作为值传递 240
 - 20.10 通过keys()、values()和items()词典方法把词典转换成列表 241
 - 20.11 对词典的项进行排序 242
 - 20.12 这个程序的englishFreqMatchScore()函数 243
 - 20.13 总结 244
- 第21章 破译维吉尼亚加密法 245
 - 21.1 词典攻击 245
 - 21.2 维吉尼亚词典攻击程序的源代码 245
 - 21.3 运行维吉尼亚词典破译程序 246
 - 21.4 readlines()文件对象方法 247
 - 21.5 巴贝奇攻击和卡斯基斯基试验 247
 - 21.6 卡斯基斯基试验的第1步——找出重复序列的间距 247

- 21.7 卡斯基试验的第2步——
获取间距的因数 248
- 21.8 从字符串获取每隔N个字母 249
- 21.9 频率分析 249
- 21.10 暴力破译可能密钥 251
- 21.11 维吉尼亚破译程序的源代码 251
- 21.12 运行维吉尼亚破译程序 256
- 21.13 这个程序如何工作 258
- 21.14 找出重复序列 259
- 21.15 计算因数 260
- 21.16 通过set()函数来移除重复值 261
- 21.17 卡斯基测试算法 263
- 21.18 extend()列表方法 264
- 21.19 print()的end关键字参数 268
- 21.20 itertools.product()函数 269
- 21.21 break语句 272
- 21.22 A组练习 273
- 21.23 修改破译程序的常量 273
- 21.24 总结 274
- 第22章 一次密码本加密法 275
 - 22.1 牢不可破的一次密码本加密法 275
 - 22.2 为什么一次密码本加密法是
牢不可破的 275
 - 22.3 小心伪随机 276
 - 22.4 小心二次密码本加密法 277
 - 22.5 二次密码本加密法就是维吉尼亚加密法 277
 - 22.6 A组练习 278
 - 22.7 总结 278
- 第23章 寻找质数 279
 - 23.1 质数 279
 - 23.2 合数 280
 - 23.3 质数筛选模块的源代码 280
 - 23.4 这个程序如何工作 281
 - 23.5 如何判断一个数字是不是质数 282
 - 23.6 埃拉托色尼筛选法 283
 - 23.7 primeSieve()函数 284
 - 23.8 检测质数 285
 - 23.9 拉宾米勒模块的源代码 285
 - 23.10 运行拉宾米勒模块 287
 - 23.11 这个程序如何工作 287
 - 23.12 拉宾米勒算法 287
 - 23.13 新的经过改进的isPrime()函数 288
 - 23.14 总结 289
- 第24章 公钥密码学和RSA加密法 291
 - 24.1 公钥密码学 291
 - 24.2 “课本”RSA的危险 292
 - 24.3 身份验证的问题 292
 - 24.4 中间人攻击 293
 - 24.5 生成公钥和私钥 293

24.6	RSA密钥生成程序的源代码	294
24.7	运行RSA密钥生成程序	295
24.8	这个密钥生成程序如何工作	296
24.9	这个程序的generateKey()函数	297
24.10	RSA密钥文件格式	299
24.11	混合加密机制	300
24.12	RSA加密法程序的源代码	300
24.13	运行RSA加密法程序	303
24.14	A组练习	304
24.15	数字签名	304
24.16	RSA加密法程序如何工作	306
24.17	ASCII：使用数字来表示字符	307
24.18	chr()和ord()函数	308
24.19	B组练习	308
24.20	区块	308
24.21	使用getBlocksFromText()把字符串转成区块	311
24.22	encode()字符串方法和字节数据类型	311
24.23	bytes()函数和decode()bytes方法	312
24.24	C组练习	312
24.25	回到代码	313
24.26	min()和max()函数	313
24.27	insert()列表方法	315
24.28	RSA加密和解密的数学运算	316
24.29	pow()函数	317
24.30	从密钥文件读取公钥和私钥	318
24.31	完整的RSA加密流程	318
24.32	完整的RSA解密流程	320
24.33	D组练习	321
24.34	我们为什么不能破译RSA加密法	321
24.35	总结	323

《Python密码学编程》

精彩短评

- 1、循序渐进，适合没编程和密码学基础的新手。
- 2、非常基础，如果想深入了解，不要读这本。这本只适合没写过代码的小白。
- 3、基于Python3，作为python初学者和ctf爱好者非常喜欢！因为教程里的例子都是切实有用的（我就不说rsa分解考过多少次了），所以编程的时候动力十足，也比较好理解。之前已经粗略看完了廖雪峰的教程，可以自行看题实现功能，写完后再看作者的程序，琢磨思路的亮点，同时还能巩固基础知识（密码学和python都是）。总而言之收获非常大~
- 4、简单易懂，但是讲的不够深入。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu111.com