

Practical Cryptography in Python

**Learning Correct Cryptography
by Example**

**Seth James Nielson
Christopher K. Monson**

Apress®

Practical Cryptography in Python: Learning Correct Cryptography by Example

Seth James Nielson
Austin, TX, USA

Christopher K. Monson
Hampstead, MD, USA

ISBN-13 (pbk): 978-1-4842-4899-7

ISBN-13 (electronic): 978-1-4842-4900-0

<https://doi.org/10.1007/978-1-4842-4900-0>

Copyright © 2019 by Seth James Nielson, Christopher K. Monson

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr

Acquisitions Editor: Susan McDermott

Development Editor: Laura Berendson

Coordinating Editor: Rita Fernando

Cover designed by eStudioCalamar

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit <http://www.apress.com/rights-permissions>.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484248997. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

For Saige, who hopes to be a Computer Scientist like Daddy.

—Seth

*To Mom and Dad, who valued good writing
and never settled for less than my best.*

—Chris

Table of Contents

About the Authors..... xi

About the Technical Reviewer xiii

Introductionxv

Chapter 1: Cryptography: More Than Secrecy 1

 Setting Up Your Python Environment 1

 Caesar’s Shifty Cipher..... 3

 A Gentle Introduction to Cryptography 12

 Uses of Cryptography..... 13

 What Could Go Wrong? 14

 YANAC: You Are Not A Cryptographer 15

 “Jump Off This Cliff”—The Internet 16

 The cryptodoneright.org Project 17

 Enough Talk, Let’s Sum Up..... 18

 Onward..... 19

Chapter 2: Hashing 21

 Hash Liberally with hashlib..... 21

 Making a Hash of Education 25

 Preimage Resistance..... 27

 Second-Preimage and Collision Resistance 33

 Digestible Hash 36

 Pass Hashwords...Er...Hash Passwords 39

 Pick Perfect Parameters..... 44

TABLE OF CONTENTS

Cracking Weak Passwords	45
Proof of Work	48
Time to Rehash	52
Chapter 3: Symmetric Encryption: Two Sides, One Key	53
Let's Scramble!	53
What Is Encryption, Really?	57
AES: A Symmetric Block Cipher	58
ECB Is Not for Me	60
Wanted: Spontaneous Independence	70
Not That Blockchain	71
Cross the Streams	86
Key and IV Management	91
Exploiting Malleability	96
Gaze into the Padding	99
Weak Keys, Bad Management	107
Other Encryption Algorithms	109
finalize()	109
Chapter 4: Asymmetric Encryption: Public/Private Keys	111
A Tale of Two Keys	111
Getting Keyed Up	112
RSA Done Wrong: Part One	114
Stuffing the Outbox	122
What Makes Asymmetric Encryption Different?	126
Pass the Padding	128
Deterministic Outputs	129
Chosen Ciphertext Attack	131
Common Modulus Attack	135
The Proof Is in the Padding	138
Exploiting RSA Encryption with PKCS #1 v1.5 Padding	142

Step 1: Blinding	148
Step 2: Searching for PKCS-Conforming Messages	150
Step 3: Narrowing the Set of Solutions	156
Step 4: Computing the Solution	158
Additional Notes About RSA	160
Key Management.....	161
Algorithm Parameters.....	162
Quantum Cryptography.....	162
Really Short Addendum.....	163
Chapter 5: Message Integrity, Signatures, and Certificates	165
An Overly Simplistic Message Authentication Code (MAC)	165
MAC, HMAC, and CBC-MAC	168
HMAC.....	169
CBC-MAC	174
Encrypting and MACing	181
Digital Signatures: Authentication and Integrity.....	183
Elliptic Curves: An Alternative to RSA	193
Certificates: Proving Ownership of Public Keys	195
Certificates and Trust	208
Revocation and Private Key Protection	210
Replay Attacks	210
Summarize-Then-MAC.....	212
Chapter 6: Combining Asymmetric and Symmetric Algorithms.....	213
Exchange AES Keys with RSA	213
Asymmetric and Symmetric: Like Chocolate and Peanut Butter.....	217
Measuring RSA's Relative Performance.....	218
Diffie-Hellman and Key Agreement.....	227
Diffie-Hellman and Forward Secrecy	233
Challenge-Response Protocols	240
Common Problems.....	242

TABLE OF CONTENTS

An Unfortunate Example of Asymmetric and Symmetric Harmony	244
That's a Wrap	248
Chapter 7: More Symmetric Crypto: Authenticated Encryption and Kerberos	249
AES-GCM	249
AES-GCM Details and Nuances	254
Other AEAD Algorithms	258
Working the Network	260
An Introduction to Kerberos	268
Additional Data	291
Chapter 8: TLS Communications.....	293
Intercepting Traffic	293
Digital Identities: X.509 Certificates	299
X.509 Fields	299
Certificate Signing Requests	302
Creating Keys, CSRs, and Certificates in Python	315
An Overview of TLS 1.2 and 1.3	320
The Introductory “Hellos”	322
Client Authentication	326
Deriving Session Keys	327
Switching to the New Cipher	330
Deriving Keys and Bulk Data Transfer	331
TLS 1.3.....	337
Certificate Verification and Trusting Trust	339
Certificate Revocation	340
Untrustworthy Roots, Pinning, and Certificate Transparency	341
Known Attacks Against TLS.....	344
POODLE.....	344
FREAK and Logjam	345
Sweet32	346
ROBOT.....	347

CRIME, TIME, and BREACH.....	347
Heartbleed	348
Using OpenSSL with Python for TLS	348
The End of the Beginning.....	359
Bibliography	361
Index.....	363

About the Authors



Seth James Nielson is the Founder and Chief Scientist of Crimson Vista, Inc., a boutique computer security research and consulting company. He is also an adjunct professor at Johns Hopkins University where he teaches network security and has also served as the Director of Advanced Research Projects in the Information Security Institute. As part of his Hopkins work, he co-founded the <https://cryptodoneright.org> knowledge base, through a generous grant from Cisco.



Christopher K. Monson has a PhD in machine learning and has spent over a decade at Google in various engineering, machine learning, and leadership roles. He has broad experience writing and teaching programming courses in multiple languages and has worked in document password recovery, malware detection, and large-scale secure computing. He is serving as the Chief Technology Officer at Data Machines Corp. and teaches Cloud Computing Security at the Johns Hopkins University Information Security Institute.

About the Technical Reviewer



Mike Ounsworth is a Software Security Architect at Entrust Datacard. He holds an undergraduate degree in computer science with concentrations in mathematics and physics and an MSc in computer science in robotics and artificial intelligence. Professionally, his day job is mainly application security architecture and penetration testing, with some research side projects in cryptography and post-quantum cryptography. Outside of work, he mentors teams competing in the high-school-age FIRST Robotics Competition.

Introduction

The interconnected world of the current era has drastically changed everything, including banking, entertainment, and even statecraft. Despite differences in users, purposes, and security profiles, these digital applications have at least one thing in common: they all require properly applied cryptography to work correctly.

Informally, cryptography is the mathematics of secrets. We need secret codes to make messages unreadable to unauthorized eyes, to make messages unchangeable, and to know who sent the message. Practical cryptography is the design and use of these codes in real systems.

This book is primarily for computer programmers with little or no previous background with cryptography. Although mathematics makes brief appearances in the book, the overall approach is to teach introductory cryptography concepts by example.

Our journey begins with some introductory components, including hashing algorithms, symmetric encryption, and asymmetric encryption. Next, we go beyond encryption and into the realm of digital certificates, signatures, and message authentication codes. The final chapters show how these various elements come together in interesting and useful combinations, such as Kerberos and TLS.

Another important part of cryptography by example is cryptography by bad example! In this book we will break things on purpose to help the reader appreciate what motivates accepted best practices. Exercises and examples include walk-throughs of real vulnerabilities that have afflicted the Internet. The bad examples will help the reader gain a greater intuition of what goes wrong in cryptography and why.