



---

# **Security Engineering**

**A Guide to Building Dependable  
Distributed Systems  
Third Edition**

Ross Anderson

**WILEY**

Copyright © 2020 by Ross Anderson  
Published by John Wiley & Sons, Inc., Indianapolis, Indiana  
Published simultaneously in Canada and the United Kingdom

ISBN: 978-1-119-64278-7  
ISBN: 978-1-119-64283-1 (ebk)  
ISBN: 978-1-119-64281-7 (ebk)

Manufactured in the United States of America

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at [www.wiley.com/go/permissions](http://www.wiley.com/go/permissions).

**Limit of Liability/Disclaimer of Warranty:** The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or website may provide or recommendations it may make. Further, readers should be aware that Internet websites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at [booksupport.wiley.com](http://booksupport.wiley.com). For more information about Wiley products, visit [www.wiley.com](http://www.wiley.com).

**Library of Congress Control Number:** 2020948679

**Trademarks:** Wiley and the Wiley logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

*For Shireen, Bavani, Nav, Ivan, Lily-Rani, Veddle and Bella*





## About the Author

I've worked with systems for over forty years. I graduated in mathematics and natural science from Cambridge in the 1970s, and got a qualification in computer engineering; my first proper job was in avionics; and after getting interested in cryptology and computer security, I worked in the banking industry in the 1980s. I then started working for companies who designed equipment for banks, and then on related applications such as prepayment electricity meters.

I moved to academia in 1992 but continued to consult to industry on security technology. During the 1990s, the number of applications that used cryptology rose rapidly: burglar alarms, car door locks, road toll tags and satellite TV systems all made their appearance. The first legal disputes about these systems came along, and I was lucky enough to be an expert witness in some of the important cases. The research team I lead had the good fortune to be in the right place at the right time when technologies such as peer-to-peer systems, tamper-resistance and digital watermarking became hot topics.

After I'd taught security and cryptology to students for a few years, it became clear to me that the existing textbooks were too narrow and theoretical: the security textbooks focused on the access control mechanisms in operating systems, while the cryptology books developed the theory behind cryptographic algorithms and protocols. These topics are interesting, and important. But they're only part of the story. Most working engineers are not overly concerned with crypto or operating system internals, but with getting good tools and learning how to use them effectively. The inappropriate use of protection mechanisms is one of the main causes of security failure. I was encouraged by the positive reception of a number of articles I wrote on security engineering (starting with 'Why Cryptosystems Fail' in 1993).

Finally, in 1999, I got round to rewriting my class lecture notes and a number of real-world case studies into a book for a general technical audience.

The first edition of the book, which appeared in 2001, helped me consolidate my thinking on the economics of information security, as I found that when I pulled my experiences about some field together into a narrative, the backbone of the story was often the incentives that the various players had faced. As the first edition of this book established itself as the standard textbook in the field, I worked on establishing security economics as a discipline. In 2002, we started the Workshop on the Economics of Information Security to bring researchers and practitioners together.

By the time the second edition came out in 2008, it was clear we'd not paid enough attention to the psychology of security either. Although we'd worked on security usability from the 1990s, there's much more to it than that. We need to understand everything from the arts of deception to how people's perception of risk is manipulated. So in 2008 we started the Workshop on Security and Human Behaviour to get security engineers talking to psychologists, anthropologists, philosophers and even magicians.

A sabbatical in 2011, which I spent partly at Google and partly at Carnegie Mellon University, persuaded me to broaden our research group to hire psychologists and criminologists. Eventually in 2015 we set up the Cambridge Cybercrime Centre to collect lots of data on the bad things that happen online and make them available to over a hundred researchers worldwide. This hasn't stopped us doing research on technical security; in fact it's helped us pick more relevant technical research topics.

A medic needs to understand a whole series of subjects including anatomy, physiology, biochemistry, pharmacy and psychology, and then temper this knowledge with experience of working on hundreds of cases with experienced colleagues. So also a security engineer needs to understand technical subjects like crypto, access controls, protocols and side channels; but this knowledge also needs to be honed by studying real cases. My goal in my academic career has been to pull all this together. The result you now hold in your hands.

I have learned a lot in the process; writing down what you think you know is a good way of finding out what you don't. I have also had a lot of fun. I hope you have as much fun reading it!

Ross Anderson  
Cambridge, November 2020



# Acknowledgements

A great many people have helped in various ways with the third edition of this book. I put the chapters online for comment as I wrote them, and I owe thanks to the many people who read them and pointed out assorted errors and obscurities. They are: Mansoor Ahmed, Sam Ainsworth, Peter Allan, Amit Seal Ami, James Andrews, Tom Auger, Asokan, Maria Bada, Daniel Bates, Craig Bauer, Pilgrim Beart, Gerd Beuster, Johann Bezuidenhout, Fred Bone, Matt Brockman, Nick Bohm, Fred Bone, Phil Booth, Lorenzo Cavallaro, David Chaiken, Yi Ting Chua, Valerio Cini, Ben Collier, Hugo Connery, Lachlan Cooper, Franck Courbon, Christopher Cowan, Ot van Daalen, Ezra Darshan, Roman Dickmann, Saar Drimer, Charles Duffy, Marlena Erdos, Andy Farnell, Bob Fenichel, David Fernée, Alexis FitzGerald, Jean-Alain Fournier, Jordan Frank, Steve Friedl, Jerry Gamache, Alex Gantman, Ben Gardiner, Jon Geater, Stuart Gentry, Cam Gerlach, John Gilmore, Jan Goette, Ralph Gross, Cyril Guerin, Pedram Hayati, Chengying He, Matt Hermannson, Alex Hicks, Ross Hinds, Timothy Howell, Nick Humphrey, James Humphry, Duncan Hurwood, Gary Irvine, Erik Itland, Christian Jeschke, Gary Johnson, Doug Jones, Henrik Karlzen, Joud Khoury, Jon Kilian, Timm Korte, Ronny Kuckuck, Mart Kung, Jay Lala, Jack Lang, Susan Landau, Peter Landrock, Carl Landwehr, Peter Lansley, Jeff Leese, Jochen Leidner, Tom de Leon, Andrew Lewis, David Lewis, Steve Lipner, Jim Lippard, Liz Louis, Simon Luyten, Christian Mainka, Dhruv Malik, Ivan Marsa-Maestra, Phil Maud, Patrick McCorry, TJ McIntyre, Marco Mesturino, Luke Mewburn, Spencer Moss, Steven Murdoch, Arvind Narayanan, Lakshmi Narayanan, Kristi Nikolla, Greg Norcie, Stanislav Ochotnický, Andy Ozment, Deborah Peel, Stephen Perlmutter, Tony Plank, William Porquet, David Pottage, Mark Quevedo, Roderick Rees, Larry Reeves, Philipp Reisinger, Mark Richards, Niklas Rosencrantz, Andy Sayler, Philipp

Schaumann, Christian Schneider, Ben Scott, Jean-Pierre Seifert, Mark Shawyer, Adam Shostack, Ilia Shumailov, Barbara Simons, Sam Smith, Saija Sorsa, Michael Specter, Chris Tarnowski, Don Taylor, Andrew Thaeler, Kurt Thomas, Anthony Vance, Jonas Vautherin, Alex Vetterl, Jeffrey Walton, Andrew Watson, Debora Weber-Wulff, Nienke Weiland, David White, Blake Wiggs, Robin Wilton, Ron Woerner, Bruno Wolff, Stuart Wray, Jeff Yan, Tom Yates, Andrew Yeomans, Haarooun Yousaf, Tim Zander and Yiren Zhao. I am also grateful to my editors at Wiley, Tom Dinse, Jim Minatel and Pete Gaughan, and to my copyeditors Judy Flynn and Kim Wimpsett, who have all helped make the process run smoothly.

The people who contributed in various ways to the first and second editions included the late Anne Anderson, Adam Atkinson, Jean Bacon, Robin Ball, Andreas Bender, Alastair Beresford, Johann Bezuidenhout, Maximilian Blochberger, David Boddie, Kristof Boeynaems, Nick Bohm, Mike Bond, Richard Bondi, Robert Brady, Martin Brain, John Brazier, Ian Brown, Mike Brown, Nick Bohm, Richard Bondi, the late Caspar Bowden, Duncan Campbell, Piotr Carlson, Peter Chambers, Valerio Cini, Richard Clayton, Frank Clish, Jolyon Clulow, Richard Cox, Dan Cvrcek, George Danezis, James Davenport, Peter Dean, John Daugman, Whit Diffie, Roger Dingledine, Nick Drage, Austin Donnelly, Ben Dougall, Saar Drimer, Orr Dunkelman, Steve Early, Dan Eble, Mike Ellims, Jeremy Epstein, Rasit Eskicioğlu, Robert Fenichel, Fleur Fisher, Shawn Fitzgerald, Darren Foong, Shailendra Fuloria, Dan Geer, Gary Geldart, Paul Gillingwater, John Gilmore, Brian Gladman, Virgil Gligor, Bruce Godfrey, John Gordon, Gary Graunke, Rich Graveman, Wendy Grossman, Dan Hagon, Feng Hao, Tony Harminc, Pieter Hartel, David Häsäther, Bill Hey, Fay Hider, Konstantin Hyppönen, Ian Jackson, Neil Jenkins, Simon Jenkins, Roger Johnston, Oliver Jorns, Nikolaos Karapanos, the late Paul Karger, Ian Kelly, Grant Kelly, Alistair Kelman, Ronald De Keulenaer, Hyoungh Joong Kim, Patrick Koeberl, Oliver Kömmerling, Simon Kramer, Markus Kuhn, Peter Landrock, Susan Landau, Jack Lang, Jong-Hyeon Lee, the late Owen Lewis, Stephen Lewis, Paul Leyland, Jim Lippard, Willie List, Dan Lough, John McHugh, the late David MacKay, Garry McKay, Udi Manber, John Martin, Nick Mathewson, Tyler Moore, the late Bob Morris, Ira Moskowitz, Steven Murdoch, Shishir Nagaraja, Roger Nebel, the late Roger Needham, Stephan Neuhaus, Andrew Odlyzko, Mark Oeltjenbruns, Joe Osborne, Andy Ozment, Alexandros Papadopoulos, Roy Paterson, Chris Pepper, Oscar Pereira, Fabien Petitcolas, Raphael Phan, Mike Roe, Mark Rotenberg, Avi Rubin, Jerry Saltzer, Marv Schaefer, Denise Schmandt-Besserat, Gus Simmons, Sam Simpson, Sergei Skorobogatov, Matthew Slyman, Rick Smith, Sijbrand Spannenburg, the late Karen Spärck Jones, Mark Staples, Frank Stajano, Philipp Steinmetz, Nik Sultana, Don Taylor, Martin Taylor, Peter Taylor, Daniel Thomas, Paul Thomas,



Vlasios Tsiatsis, Marc Tobias, Hal Varian, Nick Volenec, Daniel Wagner-Hall, Randall Walker, Robert Watson, Keith Willis, Simon Wiseman, Stuart Wray, Jeff Yan and the late Stefek Zaba. I also owe a lot to my first publisher, Carol Long.

Through the whole process I have been supported by my family, and especially by my long-suffering wife Shireen. Each edition of the book meant over a year when I was constantly distracted. Huge thanks to all for putting up with me!





# Contents at a Glance

Preface to the Third Edition	xxxvii
Preface to the Second Edition	xli
Preface to the First Edition	xlili
For my daughter, and other lawyers ...	xlvi
Foreword	xlix
Part I	
Chapter 1    What Is Security Engineering?	3
Chapter 2    Who Is the Opponent?	17
Chapter 3    Psychology and Usability	63
Chapter 4    Protocols	119
Chapter 5    Cryptography	145
Chapter 6    Access Control	207
Chapter 7    Distributed Systems	243
Chapter 8    Economics	275

**Part II**

<b>Chapter 9</b>	<b>Multilevel Security</b>	<b>315</b>
<b>Chapter 10</b>	<b>Boundaries</b>	<b>341</b>
<b>Chapter 11</b>	<b>Inference Control</b>	<b>375</b>
<b>Chapter 12</b>	<b>Banking and Bookkeeping</b>	<b>405</b>
<b>Chapter 13</b>	<b>Locks and Alarms</b>	<b>471</b>
<b>Chapter 14</b>	<b>Monitoring and Metering</b>	<b>497</b>
<b>Chapter 15</b>	<b>Nuclear Command and Control</b>	<b>529</b>
<b>Chapter 16</b>	<b>Security Printing and Seals</b>	<b>549</b>
<b>Chapter 17</b>	<b>Biometrics</b>	<b>571</b>
<b>Chapter 18</b>	<b>Tamper Resistance</b>	<b>599</b>
<b>Chapter 19</b>	<b>Side Channels</b>	<b>639</b>
<b>Chapter 20</b>	<b>Advanced Cryptographic Engineering</b>	<b>667</b>
<b>Chapter 21</b>	<b>Network Attack and Defence</b>	<b>699</b>
<b>Chapter 22</b>	<b>Phones</b>	<b>737</b>
<b>Chapter 23</b>	<b>Electronic and Information Warfare</b>	<b>777</b>
<b>Chapter 24</b>	<b>Copyright and DRM</b>	<b>815</b>
<b>Chapter 25</b>	<b>New Directions?</b>	<b>865</b>

**Part III**

<b>Chapter 26</b>	<b>Surveillance or Privacy?</b>	<b>909</b>
<b>Chapter 27</b>	<b>Secure Systems Development</b>	<b>965</b>
<b>Chapter 28</b>	<b>Assurance and Sustainability</b>	<b>1015</b>
<b>Chapter 29</b>	<b>Beyond “Computer Says No”</b>	<b>1059</b>

<b>Bibliography</b>	<b>1061</b>
---------------------	-------------

<b>Index</b>	<b>1143</b>
--------------	-------------



# Contents

<b>Preface to the Third Edition</b>	<b>xxxvii</b>
<b>Preface to the Second Edition</b>	<b>xli</b>
<b>Preface to the First Edition</b>	<b>xlili</b>
<b>For my daughter, and other lawyers ...</b>	<b>xlvii</b>
<b>Foreword</b>	<b>xlix</b>
<b>Part I</b>	
<b>Chapter 1    What Is Security Engineering?</b>	<b>3</b>
1.1    Introduction	3
1.2    A framework	4
1.3    Example 1 – a bank	6
1.4    Example 2 – a military base	7
1.5    Example 3 – a hospital	8
1.6    Example 4 – the home	10
1.7    Definitions	11
1.8    Summary	16
<b>Chapter 2    Who Is the Opponent?</b>	<b>17</b>
2.1    Introduction	17
2.2    Spies	19
2.2.1    The Five Eyes	19
2.2.1.1    Prism	19
2.2.1.2    Tempora	20
2.2.1.3    Muscular	21
2.2.1.4    Special collection	22

	2.2.1.5	Bullrun and Edgehill	22
	2.2.1.6	Xkeyscore	23
	2.2.1.7	Longhaul	24
	2.2.1.8	Quantum	25
	2.2.1.9	CNE	25
	2.2.1.10	The analyst's viewpoint	27
	2.2.1.11	Offensive operations	28
	2.2.1.12	Attack scaling	29
	2.2.2	China	30
	2.2.3	Russia	35
	2.2.4	The rest	38
	2.2.5	Attribution	40
2.3	Crooks		41
	2.3.1	Criminal infrastructure	42
	2.3.1.1	Botnet herders	42
	2.3.1.2	Malware devs	44
	2.3.1.3	Spam senders	45
	2.3.1.4	Bulk account compromise	45
	2.3.1.5	Targeted attackers	46
	2.3.1.6	Cashout gangs	46
	2.3.1.7	Ransomware	47
	2.3.2	Attacks on banking and payment systems	47
	2.3.3	Sectoral cybercrime ecosystems	49
	2.3.4	Internal attacks	49
	2.3.5	CEO crimes	49
	2.3.6	Whistleblowers	50
2.4	Geeks		52
2.5	The swamp		53
	2.5.1	Hacktivism and hate campaigns	54
	2.5.2	Child sex abuse material	55
	2.5.3	School and workplace bullying	57
	2.5.4	Intimate relationship abuse	57
2.6	Summary		59
	Research problems		60
	Further reading		61
<b>Chapter 3</b>	<b>Psychology and Usability</b>		<b>63</b>
	3.1	Introduction	63
	3.2	Insights from psychology research	64
	3.2.1	Cognitive psychology	65
	3.2.2	Gender, diversity and interpersonal variation	68

3.2.3	Social psychology	70
3.2.3.1	Authority and its abuse	71
3.2.3.2	The bystander effect	72
3.2.4	The social-brain theory of deception	73
3.2.5	Heuristics, biases and behavioural economics	76
3.2.5.1	Prospect theory and risk misperception	77
3.2.5.2	Present bias and hyperbolic discounting	78
3.2.5.3	Defaults and nudges	79
3.2.5.4	The default to intentionality	79
3.2.5.5	The affect heuristic	80
3.2.5.6	Cognitive dissonance	81
3.2.5.7	The risk thermostat	81
3.3	Deception in practice	81
3.3.1	The salesman and the scamster	82
3.3.2	Social engineering	84
3.3.3	Phishing	86
3.3.4	Opsec	88
3.3.5	Deception research	89
3.4	Passwords	90
3.4.1	Password recovery	92
3.4.2	Password choice	94
3.4.3	Difficulties with reliable password entry	94
3.4.4	Difficulties with remembering the password	95
3.4.4.1	Naïve choice	96
3.4.4.2	User abilities and training	96
3.4.4.3	Design errors	98
3.4.4.4	Operational failures	100
3.4.4.5	Social-engineering attacks	101
3.4.4.6	Customer education	102
3.4.4.7	Phishing warnings	103
3.4.5	System issues	104
3.4.6	Can you deny service?	105
3.4.7	Protecting oneself or others?	105
3.4.8	Attacks on password entry	106
3.4.8.1	Interface design	106
3.4.8.2	Trusted path, and bogus terminals	107
3.4.8.3	Technical defeats of password retry counters	107
3.4.9	Attacks on password storage	108
3.4.9.1	One-way encryption	109
3.4.9.2	Password cracking	109
3.4.9.3	Remote password checking	109

3.4.10	Absolute limits	110
3.4.11	Using a password manager	111
3.4.12	Will we ever get rid of passwords?	113
3.5	CAPTCHAs	115
3.6	Summary	116
	Research problems	117
	Further reading	118
<b>Chapter 4</b>	<b>Protocols</b>	<b>119</b>
4.1	Introduction	119
4.2	Password eavesdropping risks	120
4.3	Who goes there? – simple authentication	122
4.3.1	Challenge and response	124
4.3.2	Two-factor authentication	128
4.3.3	The MITM-in-the-middle attack	129
4.3.4	Reflection attacks	132
4.4	Manipulating the message	133
4.5	Changing the environment	134
4.6	Chosen protocol attacks	135
4.7	Managing encryption keys	136
4.7.1	The resurrecting duckling	137
4.7.2	Remote key management	137
4.7.3	The Needham-Schroeder protocol	138
4.7.4	Kerberos	139
4.7.5	Practical key management	141
4.8	Design assurance	141
4.9	Summary	143
	Research problems	143
	Further reading	144
<b>Chapter 5</b>	<b>Cryptography</b>	<b>145</b>
5.1	Introduction	145
5.2	Historical background	146
5.2.1	An early stream cipher – the Vigenère	147
5.2.2	The one-time pad	148
5.2.3	An early block cipher – Playfair	150
5.2.4	Hash functions	152
5.2.5	Asymmetric primitives	154
5.3	Security models	155
5.3.1	Random functions – hash functions	157
5.3.1.1	Properties	157
5.3.1.2	The birthday theorem	158
5.3.2	Random generators – stream ciphers	159
5.3.3	Random permutations – block ciphers	161



5.3.4	Public key encryption and trapdoor one-way permutations	163
5.3.5	Digital signatures	164
5.4	Symmetric crypto algorithms	165
5.4.1	SP-networks	165
5.4.1.1	Block size	166
5.4.1.2	Number of rounds	166
5.4.1.3	Choice of S-boxes	167
5.4.1.4	Linear cryptanalysis	167
5.4.1.5	Differential cryptanalysis	168
5.4.2	The Advanced Encryption Standard (AES)	169
5.4.3	Feistel ciphers	171
5.4.3.1	The Luby-Rackoff result	173
5.4.3.2	DES	173
5.5	Modes of operation	175
5.5.1	How not to use a block cipher	176
5.5.2	Cipher block chaining	177
5.5.3	Counter encryption	178
5.5.4	Legacy stream cipher modes	178
5.5.5	Message authentication code	179
5.5.6	Galois counter mode	180
5.5.7	XTS	180
5.6	Hash functions	181
5.6.1	Common hash functions	181
5.6.2	Hash function applications – HMAC, commitments and updating	183
5.7	Asymmetric crypto primitives	185
5.7.1	Cryptography based on factoring	185
5.7.2	Cryptography based on discrete logarithms	188
5.7.2.1	One-way commutative encryption	189
5.7.2.2	Diffie-Hellman key establishment	190
5.7.2.3	ElGamal digital signature and DSA	192
5.7.3	Elliptic curve cryptography	193
5.7.4	Certification authorities	194
5.7.5	TLS	195
5.7.5.1	TLS uses	196
5.7.5.2	TLS security	196
5.7.5.3	TLS 1.3	197
5.7.6	Other public-key protocols	197
5.7.6.1	Code signing	197
5.7.6.2	PGP/GPG	198
5.7.6.3	QUIC	199
5.7.7	Special-purpose primitives	199

5.7.8	How strong are asymmetric cryptographic primitives?	200
5.7.9	What else goes wrong	202
5.8	Summary	203
	Research problems	204
	Further reading	204
<b>Chapter 6</b>	<b>Access Control</b>	<b>207</b>
6.1	Introduction	207
6.2	Operating system access controls	209
6.2.1	Groups and roles	210
6.2.2	Access control lists	211
6.2.3	Unix operating system security	212
6.2.4	Capabilities	214
6.2.5	DAC and MAC	215
6.2.6	Apple's macOS	217
6.2.7	iOS	217
6.2.8	Android	218
6.2.9	Windows	219
6.2.10	Middleware	222
	6.2.10.1 Database access controls	222
	6.2.10.2 Browsers	223
6.2.11	Sandboxing	224
6.2.12	Virtualisation	225
6.3	Hardware protection	227
6.3.1	Intel processors	228
6.3.2	Arm processors	230
6.4	What goes wrong	231
6.4.1	Smashing the stack	232
6.4.2	Other technical attacks	234
6.4.3	User interface failures	236
6.4.4	Remedies	237
6.4.5	Environmental creep	238
6.5	Summary	239
	Research problems	240
	Further reading	240
<b>Chapter 7</b>	<b>Distributed Systems</b>	<b>243</b>
7.1	Introduction	243
7.2	Concurrency	244
7.2.1	Using old data versus paying to propagate state	245
7.2.2	Locking to prevent inconsistent updates	246
7.2.3	The order of updates	247
7.2.4	Deadlock	248

7.2.5	Non-convergent state	249
7.2.6	Secure time	250
7.3	Fault tolerance and failure recovery	251
7.3.1	Failure models	252
7.3.1.1	Byzantine failure	252
7.3.1.2	Interaction with fault tolerance	253
7.3.2	What is resilience for?	254
7.3.3	At what level is the redundancy?	255
7.3.4	Service-denial attacks	257
7.4	Naming	259
7.4.1	The Needham naming principles	260
7.4.2	What else goes wrong	263
7.4.2.1	Naming and identity	264
7.4.2.2	Cultural assumptions	265
7.4.2.3	Semantic content of names	267
7.4.2.4	Uniqueness of names	268
7.4.2.5	Stability of names and addresses	269
7.4.2.6	Restrictions on the use of names	269
7.4.3	Types of name	270
7.5	Summary	271
	Research problems	272
	Further reading	273
<b>Chapter 8</b>	<b>Economics</b>	<b>275</b>
8.1	Introduction	275
8.2	Classical economics	276
8.2.1	Monopoly	278
8.3	Information economics	281
8.3.1	Why information markets are different	281
8.3.2	The value of lock-in	282
8.3.3	Asymmetric information	284
8.3.4	Public goods	285
8.4	Game theory	286
8.4.1	The prisoners' dilemma	287
8.4.2	Repeated and evolutionary games	288
8.5	Auction theory	291
8.6	The economics of security and dependability	293
8.6.1	Why is Windows so insecure?	294
8.6.2	Managing the patching cycle	296
8.6.3	Structural models of attack and defence	298
8.6.4	The economics of lock-in, tying and DRM	300
8.6.5	Antitrust law and competition policy	302
8.6.6	Perversely motivated guards	304

8.6.7	Economics of privacy	305
8.6.8	Organisations and human behaviour	307
8.6.9	Economics of cybercrime	308
8.7	Summary	310
	Research problems	311
	Further reading	311

## Part II

<b>Chapter 9</b>	<b>Multilevel Security</b>	<b>315</b>
9.1	Introduction	315
9.2	What is a security policy model?	316
9.3	Multilevel security policy	318
9.3.1	The Anderson report	319
9.3.2	The Bell-LaPadula model	320
9.3.3	The standard criticisms of Bell-LaPadula	321
9.3.4	The evolution of MLS policies	323
9.3.5	The Biba model	325
9.4	Historical examples of MLS systems	326
9.4.1	SCOMP	326
9.4.2	Data diodes	327
9.5	MAC: from MLS to IFC and integrity	329
9.5.1	Windows	329
9.5.2	SELinux	330
9.5.3	Embedded systems	330
9.6	What goes wrong	331
9.6.1	Composability	331
9.6.2	The cascade problem	332
9.6.3	Covert channels	333
9.6.4	The threat from malware	333
9.6.5	Polyinstantiation	334
9.6.6	Practical problems with MLS	335
9.7	Summary	337
	Research problems	338
	Further reading	339
<b>Chapter 10</b>	<b>Boundaries</b>	<b>341</b>
10.1	Introduction	341
10.2	Compartmentation and the lattice model	344
10.3	Privacy for tigers	346
10.4	Health record privacy	349
10.4.1	The threat model	351
10.4.2	The BMA security policy	353
10.4.3	First practical steps	356

10.4.4	What actually goes wrong	357
10.4.4.1	Emergency care	358
10.4.4.2	Resilience	359
10.4.4.3	Secondary uses	359
10.4.5	Confidentiality – the future	362
10.4.6	Ethics	365
10.4.7	Social care and education	367
10.4.8	The Chinese Wall	369
10.5	Summary	371
	Research problems	372
	Further reading	373
<b>Chapter 11</b>	<b>Inference Control</b>	<b>375</b>
11.1	Introduction	375
11.2	The early history of inference control	377
11.2.1	The basic theory of inference control	378
11.2.1.1	Query set size control	378
11.2.1.2	Trackers	379
11.2.1.3	Cell suppression	379
11.2.1.4	Other statistical disclosure control mechanisms	380
11.2.1.5	More sophisticated query controls	381
11.2.1.6	Randomization	382
11.2.2	Limits of classical statistical security	383
11.2.3	Active attacks	384
11.2.4	Inference control in rich medical data	385
11.2.5	The third wave: preferences and search	388
11.2.6	The fourth wave: location and social	389
11.3	Differential privacy	392
11.4	Mind the gap?	394
11.4.1	Tactical anonymity and its problems	395
11.4.2	Incentives	398
11.4.3	Alternatives	399
11.4.4	The dark side	400
11.5	Summary	401
	Research problems	402
	Further reading	402
<b>Chapter 12</b>	<b>Banking and Bookkeeping</b>	<b>405</b>
12.1	Introduction	405
12.2	Bookkeeping systems	406
12.2.1	Double-entry bookkeeping	408
12.2.2	Bookkeeping in banks	408
12.2.3	The Clark-Wilson security policy model	410

12.2.4	Designing internal controls	411
12.2.5	Insider frauds	415
12.2.6	Executive frauds	416
12.2.6.1	The post office case	418
12.2.6.2	Other failures	419
12.2.6.3	Ecological validity	420
12.2.6.4	Control tuning and corporate governance	421
12.2.7	Finding the weak spots	422
12.3	Interbank payment systems	424
12.3.1	A telegraphic history of E-commerce	424
12.3.2	SWIFT	425
12.3.3	What goes wrong	427
12.4	Automatic teller machines	430
12.4.1	ATM basics	430
12.4.2	What goes wrong	433
12.4.3	Incentives and injustices	437
12.5	Credit cards	438
12.5.1	Credit card fraud	439
12.5.2	Online card fraud	440
12.5.3	3DS	443
12.5.4	Fraud engines	444
12.6	EMV payment cards	445
12.6.1	Chip cards	445
12.6.1.1	Static data authentication	446
12.6.1.2	ICVVs, DDA and CDA	450
12.6.1.3	The No-PIN attack	451
12.6.2	The preplay attack	452
12.6.3	Contactless	454
12.7	Online banking	457
12.7.1	Phishing	457
12.7.2	CAP	458
12.7.3	Banking malware	459
12.7.4	Phones as second factors	459
12.7.5	Liability	461
12.7.6	Authorised push payment fraud	462
12.8	Nonbank payments	463
12.8.1	M-Pesa	463
12.8.2	Other phone payment systems	464
12.8.3	Sofort, and open banking	465
12.9	Summary	466
	Research problems	466
	Further reading	468

<b>Chapter 13</b>	<b>Locks and Alarms</b>	<b>471</b>
13.1	Introduction	471
13.2	Threats and barriers	472
13.2.1	Threat model	473
13.2.2	Deterrence	474
13.2.3	Walls and barriers	476
13.2.4	Mechanical locks	478
13.2.5	Electronic locks	482
13.3	Alarms	484
13.3.1	How not to protect a painting	485
13.3.2	Sensor defeats	486
13.3.3	Feature interactions	488
13.3.4	Attacks on communications	489
13.3.5	Lessons learned	493
13.4	Summary	494
	Research problems	495
	Further reading	495
 <b>Chapter 14</b>	 <b>Monitoring and Metering</b>	 <b>497</b>
14.1	Introduction	497
14.2	Prepayment tokens	498
14.2.1	Utility metering	499
14.2.2	How the STS system works	501
14.2.3	What goes wrong	502
14.2.4	Smart meters and smart grids	504
14.2.5	Ticketing fraud	508
14.3	Taxi meters, tachographs and truck speed limiters	509
14.3.1	The tachograph	509
14.3.2	What goes wrong	511
	14.3.2.1 How most tachograph manipulation is done	511
	14.3.2.2 Tampering with the supply	512
	14.3.2.3 Tampering with the instrument	512
	14.3.2.4 High-tech attacks	513
14.3.3	Digital tachographs	514
	14.3.3.1 System-level problems	515
	14.3.3.2 Other problems	516
14.3.4	Sensor defeats and third-generation devices	518
14.3.5	The fourth generation – smart tachographs	518
14.4	Curfew tags: GPS as policeman	519
14.5	Postage meters	522

14.6	Summary	526
	Research problems	527
	Further reading	527
<b>Chapter 15</b>	<b>Nuclear Command and Control</b>	<b>529</b>
15.1	Introduction	529
15.2	The evolution of command and control	532
15.2.1	The Kennedy memorandum	532
15.2.2	Authorization, environment, intent	534
15.3	Unconditionally secure authentication	534
15.4	Shared control schemes	536
15.5	Tamper resistance and PALs	538
15.6	Treaty verification	540
15.7	What goes wrong	541
15.7.1	Nuclear accidents	541
15.7.2	Interaction with cyberwar	542
15.7.3	Technical failures	543
15.8	Secrecy or openness?	544
15.9	Summary	545
	Research problems	546
	Further reading	546
<b>Chapter 16</b>	<b>Security Printing and Seals</b>	<b>549</b>
16.1	Introduction	549
16.2	History	550
16.3	Security printing	551
16.3.1	Threat model	552
16.3.2	Security printing techniques	553
16.4	Packaging and seals	557
16.4.1	Substrate properties	558
16.4.2	The problems of glue	558
16.4.3	PIN mailers	559
16.5	Systemic vulnerabilities	560
16.5.1	Peculiarities of the threat model	562
16.5.2	Anti-gundecking measures	563
16.5.3	The effect of random failure	564
16.5.4	Materials control	564
16.5.5	Not protecting the right things	565
16.5.6	The cost and nature of inspection	566
16.6	Evaluation methodology	567
16.7	Summary	569
	Research problems	569
	Further reading	570



<b>Chapter 17</b>	<b>Biometrics</b>	<b>571</b>
	17.1 Introduction	571
	17.2 Handwritten signatures	572
	17.3 Face recognition	575
	17.4 Fingerprints	579
	17.4.1 Verifying positive or negative identity claims	581
	17.4.2 Crime scene forensics	584
	17.5 Iris codes	588
	17.6 Voice recognition and morphing	590
	17.7 Other systems	591
	17.8 What goes wrong	593
	17.9 Summary	596
	Research problems	597
	Further reading	597
<b>Chapter 18</b>	<b>Tamper Resistance</b>	<b>599</b>
	18.1 Introduction	599
	18.2 History	601
	18.3 Hardware security modules	601
	18.4 Evaluation	607
	18.5 Smartcards and other security chips	609
	18.5.1 History	609
	18.5.2 Architecture	610
	18.5.3 Security evolution	611
	18.5.4 Random number generators and PUFs	621
	18.5.5 Larger chips	624
	18.5.6 The state of the art	628
	18.6 The residual risk	630
	18.6.1 The trusted interface problem	630
	18.6.2 Conflicts	631
	18.6.3 The lemons market, risk dumping and evaluation games	632
	18.6.4 Security-by-obscurity	632
	18.6.5 Changing environments	633
	18.7 So what should one protect?	634
	18.8 Summary	636
	Research problems	636
	Further reading	636
<b>Chapter 19</b>	<b>Side Channels</b>	<b>639</b>
	19.1 Introduction	639
	19.2 Emission security	640
	19.2.1 History	641
	19.2.2 Technical surveillance and countermeasures	642

19.3	Passive attacks	645
19.3.1	Leakage through power and signal cables	645
19.3.2	Leakage through RF signals	645
19.3.3	What goes wrong	649
19.4	Attacks between and within computers	650
19.4.1	Timing analysis	651
19.4.2	Power analysis	652
19.4.3	Glitching and differential fault analysis	655
19.4.4	Rowhammer, CLKscrew and Plundervolt	656
19.4.5	Meltdown, Spectre and other enclave side channels	657
19.5	Environmental side channels	659
19.5.1	Acoustic side channels	659
19.5.2	Optical side channels	661
19.5.3	Other side-channels	661
19.6	Social side channels	663
19.7	Summary	663
	Research problems	664
	Further reading	664
<b>Chapter 20</b>	<b>Advanced Cryptographic Engineering</b>	<b>667</b>
20.1	Introduction	667
20.2	Full-disk encryption	668
20.3	Signal	670
20.4	Tor	674
20.5	HSMs	677
20.5.1	The xor-to-null-key attack	677
20.5.2	Attacks using backwards compatibility and time-memory tradeoffs	678
20.5.3	Differential protocol attacks	679
20.5.4	The EMV attack	681
20.5.5	Hacking the HSMs in CAs and clouds	681
20.5.6	Managing HSM risks	681
20.6	Enclaves	682
20.7	Blockchains	685
20.7.1	Wallets	688
20.7.2	Miners	689
20.7.3	Smart contracts	689
20.7.4	Off-chain payment mechanisms	691
20.7.5	Exchanges, cryptocrime and regulation	692
20.7.6	Permissioned blockchains	695
20.8	Crypto dreams that failed	695
20.9	Summary	696
	Research problems	698
	Further reading	698

<b>Chapter 21</b>	<b>Network Attack and Defence</b>	<b>699</b>
21.1	Introduction	699
21.2	Network protocols and service denial	701
21.2.1	BGP security	701
21.2.2	DNS security	703
21.2.3	UDP, TCP, SYN floods and SYN reflection	704
21.2.4	Other amplifiers	705
21.2.5	Other denial-of-service attacks	706
21.2.6	Email – from spies to spammers	706
21.3	The malware menagerie – Trojans, worms and RATs	708
21.3.1	Early history of malware	709
21.3.2	The Internet worm	710
21.3.3	Further malware evolution	711
21.3.4	How malware works	713
21.3.5	Countermeasures	714
21.4	Defense against network attack	715
21.4.1	Filtering: firewalls, censorware and wiretaps	717
21.4.1.1	Packet filtering	718
21.4.1.2	Circuit gateways	718
21.4.1.3	Application proxies	719
21.4.1.4	Ingress versus egress filtering	719
21.4.1.5	Architecture	720
21.4.2	Intrusion detection	722
21.4.2.1	Types of intrusion detection	722
21.4.2.2	General limitations of intrusion detection	724
21.4.2.3	Specific problems detecting network attacks	724
21.5	Cryptography: the ragged boundary	725
21.5.1	SSH	726
21.5.2	Wireless networking at the periphery	727
21.5.2.1	WiFi	727
21.5.2.2	Bluetooth	728
21.5.2.3	HomePlug	729
21.5.2.4	VPNs	729
21.6	CAs and PKI	730
21.7	Topology	733
21.8	Summary	734
	Research problems	734
	Further reading	735
<b>Chapter 22</b>	<b>Phones</b>	<b>737</b>
22.1	Introduction	737
22.2	Attacks on phone networks	738

22.2.1	Attacks on phone-call metering	739
22.2.2	Attacks on signaling	742
22.2.3	Attacks on switching and configuration	743
22.2.4	Insecure end systems	745
22.2.5	Feature interaction	746
22.2.6	VOIP	747
22.2.7	Frauds by phone companies	748
22.2.8	Security economics of telecomms	749
22.3	Going mobile	750
22.3.1	GSM	751
22.3.2	3G	755
22.3.3	4G	757
22.3.4	5G and beyond	758
22.3.5	General MNO failings	760
22.4	Platform security	761
22.4.1	The Android app ecosystem	763
22.4.1.1	App markets and developers	764
22.4.1.2	Bad Android implementations	764
22.4.1.3	Permissions	766
22.4.1.4	Android malware	767
22.4.1.5	Ads and third-party services	768
22.4.1.6	Pre-installed apps	770
22.4.2	Apple's app ecosystem	770
22.4.3	Cross-cutting issues	774
22.5	Summary	775
	Research problems	776
	Further reading	776
<b>Chapter 23</b>	<b>Electronic and Information Warfare</b>	<b>777</b>
23.1	Introduction	777
23.2	Basics	778
23.3	Communications systems	779
23.3.1	Signals intelligence techniques	781
23.3.2	Attacks on communications	784
23.3.3	Protection techniques	785
23.3.3.1	Frequency hopping	786
23.3.3.2	DSSS	787
23.3.3.3	Burst communications	788
23.3.3.4	Combining covertness and jam resistance	789
23.3.4	Interaction between civil and military uses	790
23.4	Surveillance and target acquisition	791
23.4.1	Types of radar	792

23.4.2	Jamming techniques	793
23.4.3	Advanced radars and countermeasures	795
23.4.4	Other sensors and multisensor issues	796
23.5	IFF systems	797
23.6	Improvised explosive devices	800
23.7	Directed energy weapons	802
23.8	Information warfare	803
23.8.1	Attacks on control systems	805
23.8.2	Attacks on other infrastructure	808
23.8.3	Attacks on elections and political stability	809
23.8.4	Doctrine	811
23.9	Summary	812
	Research problems	813
	Further reading	813
<b>Chapter 24</b>	<b>Copyright and DRM</b>	<b>815</b>
24.1	Introduction	815
24.2	Copyright	817
24.2.1	Software	817
24.2.2	Free software, free culture?	823
24.2.3	Books and music	827
24.2.4	Video and pay-TV	828
24.2.4.1	Typical system architecture	829
24.2.4.2	Video scrambling techniques	830
24.2.4.3	Attacks on hybrid scrambling systems	832
24.2.4.4	DVB	836
24.2.5	DVD	837
24.3	DRM on general-purpose computers	838
24.3.1	Windows media rights management	839
24.3.2	FairPlay, HTML5 and other DRM systems	840
24.3.3	Software obfuscation	841
24.3.4	Gaming, cheating, and DRM	843
24.3.5	Peer-to-peer systems	845
24.3.6	Managing hardware design rights	847
24.4	Information hiding	848
24.4.1	Watermarks and copy generation management	849
24.4.2	General information hiding techniques	849
24.4.3	Attacks on copyright marking schemes	851
24.5	Policy	854
24.5.1	The IP lobby	857
24.5.2	Who benefits?	859
24.6	Accessory control	860

24.7	Summary	862
	Research problems	862
	Further reading	863
<b>Chapter 25</b>	<b>New Directions?</b>	<b>865</b>
25.1	Introduction	865
25.2	Autonomous and remotely-piloted vehicles	866
25.2.1	Drones	866
25.2.2	Self-driving cars	867
25.2.3	The levels and limits of automation	869
25.2.4	How to hack a self-driving car	872
25.3	AI / ML	874
25.3.1	ML and security	875
25.3.2	Attacks on ML systems	876
25.3.3	ML and society	879
25.4	PETS and operational security	882
25.4.1	Anonymous messaging devices	885
25.4.2	Social support	887
25.4.3	Living off the land	890
25.4.4	Putting it all together	891
25.4.5	The name's Bond. James Bond	893
25.5	Elections	895
25.5.1	The history of voting machines	896
25.5.2	Hanging chads	896
25.5.3	Optical scan	898
25.5.4	Software independence	899
25.5.5	Why electronic elections are hard	900
25.6	Summary	904
	Research problems	904
	Further reading	905
 <b>Part III</b>		
<b>Chapter 26</b>	<b>Surveillance or Privacy?</b>	<b>909</b>
26.1	Introduction	909
26.2	Surveillance	912
26.2.1	The history of government wiretapping	912
26.2.2	Call data records (CDRs)	916
26.2.3	Search terms and location data	919
26.2.4	Algorithmic processing	920
26.2.5	ISPs and CSPs	921
26.2.6	The Five Eyes' system of systems	922
26.2.7	The crypto wars	925
26.2.7.1	The back story to crypto policy	926
26.2.7.2	DES and crypto research	927

26.2.7.3	Crypto War 1 – the Clipper chip	928
26.2.7.4	Crypto War 2 – going spotty	931
26.2.8	Export control	934
26.3	Terrorism	936
26.3.1	Causes of political violence	936
26.3.2	The psychology of political violence	937
26.3.3	The role of institutions	938
26.3.4	The democratic response	940
26.4	Censorship	941
26.4.1	Censorship by authoritarian regimes	942
26.4.2	Filtering, hate speech and radicalisation	944
26.5	Forensics and rules of evidence	948
26.5.1	Forensics	948
26.5.2	Admissibility of evidence	950
26.5.3	What goes wrong	951
26.6	Privacy and data protection	953
26.6.1	European data protection	953
26.6.2	Privacy regulation in the USA	956
26.6.3	Fragmentation?	958
26.7	Freedom of information	960
26.8	Summary	961
	Research problems	962
	Further reading	962
<b>Chapter 27</b>	<b>Secure Systems Development</b>	<b>965</b>
27.1	Introduction	965
27.2	Risk management	966
27.3	Lessons from safety-critical systems	969
27.3.1	Safety engineering methodologies	970
27.3.2	Hazard analysis	971
27.3.3	Fault trees and threat trees	971
27.3.4	Failure modes and effects analysis	972
27.3.5	Threat modelling	973
27.3.6	Quantifying risks	975
27.4	Prioritising protection goals	978
27.5	Methodology	980
27.5.1	Top-down design	981
27.5.2	Iterative design: from spiral to agile	983
27.5.3	The secure development lifecycle	985
27.5.4	Gated development	987
27.5.5	Software as a Service	988
27.5.6	From DevOps to DevSecOps	991
	27.5.6.1 The Azure ecosystem	991

27.5.6.2	The Google ecosystem	992
27.5.6.3	Creating a learning system	994
27.5.7	The vulnerability cycle	995
27.5.7.1	The CVE system	997
27.5.7.2	Coordinated disclosure	998
27.5.7.3	Security incident and event management	999
27.5.8	Organizational mismanagement of risk	1000
27.6	Managing the team	1004
27.6.1	Elite engineers	1004
27.6.2	Diversity	1005
27.6.3	Nurturing skills and attitudes	1007
27.6.4	Emergent properties	1008
27.6.5	Evolving your workflow	1008
27.6.6	And finally ...	1010
27.7	Summary	1010
	Research problems	1011
	Further reading	1012
<b>Chapter 28</b>	<b>Assurance and Sustainability</b>	<b>1015</b>
28.1	Introduction	1015
28.2	Evaluation	1018
28.2.1	Alarms and locks	1019
28.2.2	Safety evaluation regimes	1019
28.2.3	Medical device safety	1020
28.2.4	Aviation safety	1023
28.2.5	The Orange book	1025
28.2.6	FIPS 140 and HSMs	1026
28.2.7	The common criteria	1026
28.2.7.1	The gory details	1027
28.2.7.2	What goes wrong with the Common Criteria	1029
28.2.7.3	Collaborative protection profiles	1031
28.2.8	The 'Principle of Maximum Complacency'	1032
28.2.9	Next steps	1034
28.3	Metrics and dynamics of dependability	1036
28.3.1	Reliability growth models	1036
28.3.2	Hostile review	1039
28.3.3	Free and open-source software	1040
28.3.4	Process assurance	1042
28.4	The entanglement of safety and security	1044
28.4.1	The electronic safety and security of cars	1046
28.4.2	Modernising safety and security regulation	1049
28.4.3	The Cybersecurity Act 2019	1050



28.5 Sustainability	1051
28.5.1 The Sales of goods directive	1052
28.5.2 New research directions	1053
28.6 Summary	1056
Research problems	1057
Further reading	1058
<b>Chapter 29 Beyond “Computer Says No”</b>	<b>1059</b>
<b>Bibliography</b>	<b>1061</b>
<b>Index</b>	<b>1143</b>



# Preface to the Third Edition

The first edition of *Security Engineering* was published in 2001 and the second in 2008. Since then there have been huge changes.

The most obvious is that the smartphone has displaced the PC and laptop. Most of the world's population now walk around with a computer that's also a phone, a camera and a satnav; and the apps that run on these magic devices have displaced many of the things we were building ten years ago. Taxi rides are now charged by ride-hailing apps rather than by taxi meters. Banking has largely gone online, with phones starting to displace credit cards. Energy saving is no longer about your meter talking to your heating system but about both talking to your phone. Social networking has taken over many people's lives, driving everything from advertising to politics.

A related but less visible change is the move to large server farms. Sensitive data have moved from servers in schools, doctors' offices and law firms to cloud service providers. Many people no longer do their writing on word processing software on their laptop but on Google Docs or Office365 (I'm writing this book on Overleaf). This has consequences. Security breaches can happen at a scale no-one would have imagined twenty years ago. Compromises of tens of millions of passwords, or credit cards, have become almost routine. And in 2013, we discovered that fifteen years' worth of UK hospital medical records had been sold to 1200 organisations worldwide without the consent of the patients (who were still identifiable via their postcodes and dates of birth).

A real game-changer of the last decade was the Snowden revelations, also in 2013, when over 50,000 Top Secret documents about the NSA's signals intelligence activities were leaked to the press. The scale and intrusiveness of government surveillance surprised even cynical security engineers. It followed on from Stuxnet, where America attacked Iran's nuclear weapons program using malware, and was followed by NotPetya, where a Russian

cyberweapon, deployed against the Ukraine, inflicted hundreds of millions of dollars' worth of collateral damage on firms elsewhere. This brings us to the third big change, which is a much better understanding of nation-state security threats. In addition to understanding the capabilities and priorities of western intelligence agencies, we have a reasonably good idea of what the Chinese, the Russians and even the Syrians get up to.

And where the money is, the crooks follow too. The last decade has also seen the emergence of a cyber-crime ecosystem, with malware writers providing the tools to subvert millions of machines, many of which are used as criminal infrastructure while others are subverted in various ways into defrauding their users. We have a team at Cambridge that studies this, and so do dozens of other research groups worldwide. The rise of cybercrime is changing policing, and other state activity too: cryptocurrencies are not just making it easier to write ransomware, but undermining financial regulation. And then there are non-financial threats from cyber-bullying up through hate speech to election manipulation and videos of rape and murder.

So online harms now engage all sorts of people from teachers and the police to banks and the military. It is ever more important to measure the costs of these harms, and the effectiveness of the measures we deploy to mitigate them.

Some of the changes would have really surprised someone who read my book ten years ago and then spent a decade in solitary confinement. For example, the multilevel security industry is moribund, despite being the beneficiary of billions of dollars of US government funding over forty years; the Pentagon's entire information security philosophy – of mandating architectures to stop information flowing downward from Top Secret to Secret to Confidential to Unclassified – has been abandoned as unworkable. While architecture still matters, the emphasis has shifted to ecosystems. Given that bugs are ubiquitous and exploits inevitable, we had better be good at detecting exploits, fixing bugs and recovering from attacks. The game is no longer trusted systems but coordinated disclosure, DevSecOps and resilience.

What might the future hold? A likely game-changer is that as we put software into safety-critical systems like cars and medical devices, and connect them to the Internet, safety and security engineering are converging. This is leading to real strains; while security engineers fix bugs quickly, safety engineers like to test systems rigorously against standards that change slowly if at all. A wicked problem is how we will patch durable goods. At present, you might get security patches for your phone for three years and your laptop for five; you're expected to buy a new one after that. But cars last for fifteen years on average and if we're suddenly asked to scrap them after five the environmental costs won't be acceptable. So tell me, if you're writing navigation software today in 2020 for a car that will launch in 2023, how will you ensure that you can keep on shipping security patches in 2033, 2043 and 2053? What tools will you choose today?

Finally, there has been a sea change in the political environment. After decades in which political leaders considered technology policy to be for men in anoraks, and generally took the line of least resistance, the reports of Russian interference in the Brexit referendum and the Trump election got their attention. The prospect of losing your job can concentrate the mind wonderfully. The close attention of lawmakers is changing the game, first with tighter general rules such as Europe's General Data Protection Regulation; and second as products that are already regulated for safety, from cars and railway signals to children's toys acquire software and online connectivity, which has led to rules in Europe about how long software has to be maintained.

The questions the security engineer has to ask today are just the same as a decade ago: what are we seeking to prevent, and will the proposed mechanisms actually work? However, the canvas on which we work is now much broader. Almost all human life is there.

Ross Anderson  
Cambridge, October 2020





# Preface to the Second Edition

The first edition of *Security Engineering* was published in May 2001. Since then the world has changed.

System security was one of Microsoft's lowest priorities then; it's now one of the highest. The volume of malware continues to increase along with the nuisance that it causes. Although a lot of effort has gone into defence – we have seen Windows NT replaced by XP and then Vista, and occasional service packs replaced by monthly security patches – the effort put into attacks has increased far more. People who write viruses no longer do so for fun, but for profit; the last few years have seen the emergence of a criminal economy that supports diverse specialists. Spammers, virus writers, phishermen, money launderers and spies trade busily with each other.

Cryptography has also moved on. The Advanced Encryption Standard is being embedded into more and more products, and we have some interesting developments on the public-key side of things too. But just as our algorithm problems get solved, so we face a host of implementation issues. Side channels, poorly designed APIs and protocol failures continue to break systems. Applied cryptography is harder than ever to do well.

Pervasive computing also opens up new challenges. As computers and communications become embedded invisibly everywhere, so problems that used to only afflict 'proper computers' crop up in all sorts of other devices too. What does it mean for a thermometer to be secure, or an air-conditioner?

The great diversity of intelligent devices brings with it a great diversity of interests and actors. Security is not just about keeping the bad guys out, but increasingly concerned with tussles for power and control. DRM pits the content and platform industries against consumers, and against each other; accessory control is used to tie printers to their vendors' cartridges, but leads to antitrust lawsuits and government intervention. Security also interacts with

safety in applications from cars through utilities to electronic healthcare. The security engineer needs to understand not just crypto and operating systems, but economics and human factors as well.

And the ubiquity of digital devices means that ‘computer security’ is no longer just a problem for a few systems specialists. Almost all white-collar crime (and much crime of the serious violent sort) now involves computers or mobile phones, so a detective needs to understand computer forensics just as she needs to know how to drive. More and more lawyers, accountants, managers and other people with no formal engineering training are going to have to understand system security in order to do their jobs well.

The rapid growth of online services, from Google and Facebook to massively multiplayer games, has also changed the world. Bugs in online applications can be fixed rapidly once they’re noticed, but the applications get ever more complex and their side-effects harder to predict. We may have a reasonably good idea what it means for an operating system or even a banking service to be secure, but we can’t make any such claims for online lifestyles that evolve all the time. We’re entering a novel world of evolving socio-technical systems, and that raises profound questions about how the evolution is driven and who is in control.

The largest changes, however, may be those driven by the tragic events of September 2001 and by our reaction to them. These have altered perceptions and priorities in many ways, and changed the shape of the security industry. Terrorism is not just about risk, but about the perception of risk, and about the manipulation of perception. This adds psychology and politics to the mix. Security engineers also have a duty to contribute to the political debate. Where inappropriate reactions to terrorist crimes have led to major waste of resources and unforced policy errors, we have to keep on educating people to ask a few simple questions: what are we seeking to prevent, and will the proposed mechanisms actually work?

Ross Anderson  
Cambridge, January 2008



# Preface to the First Edition

For generations, people have defined and protected their property and their privacy using locks, fences, signatures, seals, account books, and meters. These have been supported by a host of social constructs ranging from international treaties through national laws to manners and customs.

This is changing, and quickly. Most records are now electronic, from bank accounts to registers of real property; and transactions are increasingly electronic, as shopping moves to the Internet. Just as important, but less obvious, are the many everyday systems that have been quietly automated. Burglar alarms no longer wake up the neighborhood, but send silent messages to the police; students no longer fill their dormitory washers and dryers with coins, but credit them using a smartcard they recharge at the college bookstore; locks are no longer simple mechanical affairs, but are operated by electronic remote controls or swipe cards; and instead of renting videocassettes, millions of people get their movies from satellite or cable channels. Even the humble banknote is no longer just ink on paper, but may contain digital watermarks that enable many forgeries to be detected by machine.

How good is all this new security technology? Unfortunately, the honest answer is 'nowhere near as good as it should be.' New systems are often rapidly broken, and the same elementary mistakes are repeated in one application after another. It often takes four or five attempts to get a security design right, and that is far too many.

The media regularly report security breaches on the Internet; banks fight their customers over 'phantom withdrawals' from cash machines; VISA reports huge increases in the number of disputed Internet credit card transactions; satellite TV companies hound pirates who copy their smartcards; and law enforcement agencies try to stake out territory in cyberspace with laws controlling the use of encryption. Worse still, features interact. A mobile phone



that calls the last number again if one of the keys is pressed by accident may be just a minor nuisance – until someone invents a machine that dispenses a can of soft drink every time its phone number is called. When all of a sudden you find 50 cans of Coke on your phone bill, who is responsible, the phone company, the handset manufacturer, or the vending machine operator? Once almost every electronic device that affects your life is connected to the Internet – which Microsoft expects to happen by 2010 – what does ‘Internet security’ mean to you, and how do you cope with it?

As well as the systems that fail, many systems just don’t work well enough. Medical record systems don’t let doctors share personal health information as they would like, but still don’t protect it against inquisitive private eyes. Zillion-dollar military systems prevent anyone without a “top secret” clearance from getting at intelligence data, but are often designed so that almost everyone needs this clearance to do any work. Passenger ticket systems are designed to prevent customers cheating, but when trustbusters break up the railroad, they cannot stop the new rail companies cheating each other. Many of these failures could have been foreseen if designers had just a little bit more knowledge of what had been tried, and had failed, elsewhere.

Security engineering is the new discipline that is starting to emerge out of all this chaos.

Although most of the underlying technologies (cryptology, software reliability, tamper resistance, security printing, auditing, etc.) are relatively well understood, the knowledge and experience of how to apply them effectively is much scarcer. And since the move from mechanical to digital mechanisms is happening everywhere at once, there just has not been time for the lessons learned to percolate through the engineering community. Time and again, we see the same old square wheels being reinvented.

The industries that have managed the transition most capably are often those that have been able to borrow an appropriate technology from another discipline. Examples include the reuse of technology designed for military identify-friend-or-foe equipment in bank cash machines and even prepayment gas meters. So even if a security designer has serious expertise in some particular speciality – whether as a mathematician working with ciphers or a chemist developing banknote inks – it is still prudent to have an overview of the whole subject. The essence of good security engineering is understanding the potential threats to a system, then applying an appropriate mix of protective measures – both technological and organizational – to control them. Knowing what has worked, and more importantly what has failed, in other applications is a great help in developing judgment. It can also save a lot of money.

The purpose of this book is to give a solid introduction to security engineering, as we understand it at the beginning of the twenty-first century. My goal is that it works at four different levels:

1. as a textbook that you can read from one end to the other over a few days as an introduction to the subject. The book is to be used mainly by the working IT professional who needs to learn about the subject, but it can also be used in a one-semester course in a university;
2. as a reference book to which you can come for an overview of the workings of some particular type of system (such as cash machines, taxi meters, radar jammers, anonymous medical record databases or whatever);
3. as an introduction to the underlying technologies, such as crypto, access control, inference control, tamper resistance, and seals. Space prevents me from going into great depth; but I provide a basic road map for each subject, plus a reading list for the curious (and a list of open research problems for the prospective graduate student);
4. as an original scientific contribution in which I have tried to draw out the common principles that underlie security engineering, and the lessons that people building one kind of system should have learned from others. In the many years I have been working in security, I keep coming across these. For example, a simple attack on stream ciphers wasn't known to the people who designed a common anti-aircraft fire control radar so it was easy to jam; while a trick well known to the radar community wasn't understood by banknote printers and people who design copyright marking schemes, which led to a quite general attack on most digital watermarks.

I have tried to keep this book resolutely mid-Atlantic. A security engineering book has to be, as many of the fundamental technologies are American, while many of the interesting applications are European. (This isn't surprising given the better funding of US universities and research labs, and the greater diversity of nations and markets in Europe.) What's more, many of the successful European innovations – from the smartcard to the GSM mobile phone to the pay-per-view TV service – have crossed the Atlantic and now thrive in the Americas. Both the science, and the case studies, are necessary.

This book grew out of the security engineering courses I teach at Cambridge University, but I have rewritten my notes to make them self-contained and added at least as much material again. It should be useful to the established professional security manager or consultant as a first-line reference; to the computer science professor doing research in cryptology; to the working police detective trying to figure out the latest computer scam; and to policy wonks struggling with the conflicts involved in regulating cryptography and

anonymity. Above all, it is aimed at Dilbert. My main audience is the working programmer or engineer who is trying to design real systems that will keep on working despite the best efforts of customers, managers, and everybody else.

This book is divided into three parts.

- The first looks at basic concepts, starting with the central concept of a security protocol, and going on to the human-computer interface, access controls, cryptology and distributed system issues. It does not assume any particular technical background other than basic computer literacy. It is based on an 'Introduction to Security' course which we teach to second year undergraduates.
- The second part looks in much more detail at a number of important applications such as military communications, medical record systems, cash machines, mobile phones and pay-TV. These are used to introduce more of the advanced technologies and concepts. It also considers information security from the viewpoint of a number of different interest groups such as companies, consumers, criminals, the police and spies. This material is drawn from my senior course on security, from research work, and from experience consulting.
- The third part looks at the organizational and policy issues: how computer security interacts with law, with evidence, and with corporate politics; how we can gain confidence that a system will perform as intended; and how the whole business of security engineering can best be managed.

I believe that building systems which continue to perform robustly in the face of malice is one of the most important, interesting, and difficult tasks facing engineers in the twenty-first century.

Ross Anderson  
Cambridge, January 2001



## For my daughter, and other lawyers ...

The tricks taught in this book are intended only to enable you to build better systems. They are not in any way given as a means of helping you to break into systems or do anything else illegal. So where possible I have tried to give case histories at a level of detail that illustrates the underlying principles without giving a ‘hacker’s cookbook’.

Governments fought to restrict knowledge of cryptography until the turn of the century, and there may still be people who believe that the knowledge contained in this book should not be published.

Their fears were answered in the first book in English that discussed cryptology, a 1641 treatise on optical and acoustic telegraphy written by Oliver Cromwell’s cryptographer and son-in-law John Wilkins [2025]. He traced scientific censorship back to the Egyptian priests who forbade the use of alphabetic writing on the grounds that it would spread literacy among the common people and thus foster dissent. As he said:

*‘It will not follow that everything must be suppressed which may be abused ... If all those useful inventions that are liable to abuse should therefore be concealed there is not any Art of Science which may be lawfully professed.’*

The question was raised again in the nineteenth century, when some well-meaning people wanted to ban books on locksmithing. In 1853, a contemporary writer replied [1899]:

*‘Many well-meaning persons suppose that the discussion respecting the means for baffling the supposed safety of locks offers a premium for dishonesty, by showing others how to be dishonest. This is a fallacy. Rogues are very keen in their profession, and already know much more than we can teach them respecting their several kinds of roguery. Rogues knew a good deal about lockpicking long before*

*locksmiths discussed it among themselves ... if there be harm, it will be much more than counterbalanced by good.'*

Thirty years later, in the first book on cryptographic engineering, Auguste Kerckhoffs explained that you must always assume that the other side knows the system, so security must reside in the choice of a key.

His wisdom has been borne out by long experience since. The relative benefits of 'Open' versus 'Closed' security systems have also been studied by researchers applying the tools of dependability analysis and security economics. We discuss their findings in this book.

In short, while some bad guys will benefit from a book such as this, they mostly know it already – and the good guys benefit much more.

Ross Anderson  
Cambridge, November 2020



## Foreword

In a paper he wrote with Roger Needham, Ross Anderson coined the phrase ‘programming Satan’s computer’ to describe the problems faced by computer-security engineers. It’s the sort of evocative image I’ve come to expect from Ross, and a phrase I’ve used ever since.

Programming a computer is straightforward: keep hammering away at the problem until the computer does what it’s supposed to do. Large application programs and operating systems are a lot more complicated, but the methodology is basically the same. Writing a reliable computer program is much harder, because the program needs to work even in the face of random errors and mistakes: Murphy’s computer, if you will. Significant research has gone into reliable software design, and there are many mission-critical software applications that are designed to withstand Murphy’s Law.

Writing a secure computer program is another matter entirely. Security involves making sure things work, not in the presence of random faults, but in the face of an intelligent and malicious adversary trying to ensure that things fail in the worst possible way at the worst possible time ... again and again. It truly is programming Satan’s computer.

Security engineering is different from any other kind of programming. It’s a point I made over and over again: in my own book, *Secrets and Lies*, in my monthly newsletter *Crypto-Gram*, and in my other writings. And it’s a point Ross makes in every chapter of this book. This is why, if you’re doing any security engineering ... if you’re even thinking of doing any security engineering, you need to read this book. It’s the first, and only, end-to-end modern security design and engineering book ever written.

And it comes just in time. You can divide the history of the Internet into three waves. The first wave centered around mainframes and terminals. Computers

were expensive and rare. The second wave, from about 1992 until now, centered around personal computers, browsers, and large application programs. And the third, starting now, will see the connection of all sorts of devices that are currently in proprietary networks, standalone, and non-computerized. By 2003, there will be more mobile phones connected to the Internet than computers. Within a few years we'll see many of the world's refrigerators, heart monitors, bus and train ticket dispensers, burglar alarms, and electricity meters talking IP. Personal computers will be a minority player on the Internet.

Security engineering, especially in this third wave, requires you to think differently. You need to figure out not how something works, but how something can be made to not work. You have to imagine an intelligent and malicious adversary inside your system (remember Satan's computer), constantly trying new ways to subvert it. You have to consider all the ways your system can fail, most of them having nothing to do with the design itself. You have to look at everything backwards, upside down, and sideways. You have to think like an alien.

As the late great science fiction editor John W. Campbell, said: "An alien thinks as well as a human, but not like a human." Computer security is a lot like that. Ross is one of those rare people who can think like an alien, and then explain that thinking to humans. Have fun reading.

Bruce Schneier  
January 2001