

THIRD EDITION

SECURITY ENGINEERING

.....

**A GUIDE TO
BUILDING DEPENDABLE
DISTRIBUTED SYSTEMS**

ROSS ANDERSON

.....

WILEY

Table of Contents

[Cover](#)

[Title Page](#)

[Preface to the Third Edition](#)

[Preface to the Second Edition](#)

[Preface to the First Edition](#)

[For my daughter, and other lawyers...](#)

[Foreword](#)

[PART I](#)

[CHAPTER 1: What Is Security Engineering?](#)

[1.1 Introduction](#)

[1.2 A framework](#)

[1.3 Example 1 - a bank](#)

[1.4 Example 2 - a military base](#)

[1.5 Example 3 - a hospital](#)

[1.6 Example 4 - the home](#)

[1.7 Definitions](#)

[1.8 Summary](#)

[Note](#)

[CHAPTER 2: Who Is the Opponent?](#)

[2.1 Introduction](#)

[2.2 Spies](#)

[2.3 Crooks](#)

[2.4 Geeks](#)

[2.5 The swamp](#)

[2.6 Summary](#)

[Research problems](#)

[Further reading](#)

[Notes](#)

[CHAPTER 3: Psychology and Usability](#)

[3.1 Introduction](#)

[3.2 Insights from psychology research](#)

[3.3 Deception in practice](#)

[3.4 Passwords](#)

[3.5 CAPTCHAs](#)

[3.6 Summary](#)

[Research problems](#)

[Further reading](#)

[Notes](#)

[CHAPTER 4: Protocols](#)

[4.1 Introduction](#)

[4.2 Password eavesdropping risks](#)

[4.3 Who goes there? - simple authentication](#)

[4.4 Manipulating the message](#)

[4.5 Changing the environment](#)

[4.6 Chosen protocol attacks](#)

[4.7 Managing encryption keys](#)

[4.8 Design assurance](#)

[4.9 Summary](#)

[Research problems](#)

[Further reading](#)

[Notes](#)

[CHAPTER 5: Cryptography](#)

[5.1 Introduction](#)

[5.2 Historical background](#)

[5.3 Security models](#)

[5.4 Symmetric crypto algorithms](#)

[5.5 Modes of operation](#)

[5.6 Hash functions](#)

[5.7 Asymmetric crypto primitives](#)

[5.8 Summary](#)

[Research problems](#)

[Further reading](#)

[Notes](#)

[CHAPTER 6: Access Control](#)

[6.1 Introduction](#)

[6.2 Operating system access controls](#)

[6.3 Hardware protection](#)

[6.4 What goes wrong](#)

[6.5 Summary](#)

[Research problems](#)

[Further reading](#)

[Notes](#)

[CHAPTER 7: Distributed Systems](#)

[7.1 Introduction](#)

[7.2 Concurrency](#)

[7.3 Fault tolerance and failure recovery](#)

[7.4 Naming](#)

[7.5 Summary](#)

[Research problems](#)

[Further reading](#)

[Notes](#)

CHAPTER 8: Economics

8.1 Introduction

8.2 Classical economics

8.3 Information economics

8.4 Game theory

8.5 Auction theory

8.6 The economics of security and dependability

8.7 Summary

Research problems

Further reading

Notes

PART II

CHAPTER 9: Multilevel Security

9.1 Introduction

9.2 What is a security_policy_model?

9.3 Multilevel security_policy

9.4 Historical examples of MLS systems

9.5 MAC: from MLS to IFC and integrity

9.6 What goes wrong

9.7 Summary

Research problems

Further reading

Notes

CHAPTER 10: Boundaries

10.1 Introduction

10.2 Compartmentation and the lattice model

10.3 Privacy for tigers

[10.4 Health record privacy](#)

[10.5 Summary](#)

[Research problems](#)

[Further reading](#)

[Notes](#)

[CHAPTER 11: Inference Control](#)

[11.1 Introduction](#)

[11.2 The early history of inference control](#)

[11.3 Differential privacy](#)

[11.4 Mind the gap?](#)

[11.5 Summary](#)

[Research problems](#)

[Further reading](#)

[Notes](#)

[CHAPTER 12: Banking and Bookkeeping](#)

[12.1 Introduction](#)

[12.2 Bookkeeping systems](#)

[12.3 Interbank payment systems](#)

[12.4 Automatic teller machines](#)

[12.5 Credit cards](#)

[12.6 EMV payment cards](#)

[12.7 Online banking](#)

[12.8 Nonbank payments](#)

[12.9 Summary](#)

[Research problems](#)

[Further reading](#)

[Notes](#)

[CHAPTER 13: Locks and Alarms](#)

[13.1 Introduction](#)

[13.2 Threats and barriers](#)

[13.3 Alarms](#)

[13.4 Summary](#)

[Research problems](#)

[Further reading](#)

[Notes](#)

[CHAPTER 14: Monitoring and Metering](#)

[14.1 Introduction](#)

[14.2 Prepayment tokens](#)

[14.3 Taxi meters, tachographs and truck speed limiters](#)

[14.4 Curfew tags: GPS as policeman](#)

[14.5 Postage meters](#)

[14.6 Summary](#)

[Research problems](#)

[Further reading](#)

[Notes](#)

[CHAPTER 15: Nuclear Command and Control](#)

[15.1 Introduction](#)

[15.2 The evolution of command and control](#)

[15.3 Unconditionally secure authentication](#)

[15.4 Shared control schemes](#)

[15.5 Tamper resistance and PALs](#)

[15.6 Treaty verification](#)

[15.7 What goes wrong](#)

[15.8 Secrecy or openness?](#)

[15.9 Summary](#)

[Research problems](#)

[Further reading](#)

[Notes](#)

[CHAPTER 16: Security Printing and Seals](#)

[16.1 Introduction](#)

[16.2 History](#)

[16.3 Security printing](#)

[16.4 Packaging and seals](#)

[16.5 Systemic vulnerabilities](#)

[16.6 Evaluation methodology](#)

[16.7 Summary](#)

[Research problems](#)

[Further reading](#)

[CHAPTER 17: Biometrics](#)

[17.1 Introduction](#)

[17.2 Handwritten signatures](#)

[17.3 Face recognition](#)

[17.4 Fingerprints](#)

[17.5 Iris codes](#)

[17.6 Voice recognition and morphing](#)

[17.7 Other systems](#)

[17.8 What goes wrong](#)

[17.9 Summary](#)

[Research problems](#)

[Further reading](#)

[Notes](#)

[CHAPTER 18: Tamper Resistance](#)

[18.1 Introduction](#)

[18.2 History](#)

[18.3 Hardware security modules](#)

[18.4 Evaluation](#)

[18.5 Smartcards and other security chips](#)

[18.6 The residual risk](#)

[18.7 So what should one protect?](#)

[18.8 Summary](#)

[Research problems](#)

[Further reading](#)

[Notes](#)

[CHAPTER 19: Side Channels](#)

[19.1 Introduction](#)

[19.2 Emission security](#)

[19.3 Passive attacks](#)

[19.4 Attacks between and within computers](#)

[19.5 Environmental side channels](#)

[19.6 Social side channels](#)

[19.7 Summary](#)

[Research problems](#)

[Further reading](#)

[CHAPTER 20: Advanced Cryptographic Engineering](#)

[20.1 Introduction](#)

[20.2 Full-disk encryption](#)

[20.3 Signal](#)

[20.4 Tor](#)

[20.5 HSMs](#)

[20.6 Enclaves](#)

[20.7 Blockchains](#)

[20.8 Crypto dreams that failed](#)

[20.9 Summary](#)

[Research problems](#)

[Further reading](#)

[Notes](#)

[CHAPTER 21: Network Attack and Defence](#)

[21.1 Introduction](#)

[21.2 Network protocols and service denial](#)

[21.3 The malware menagerie – Trojans, worms and RATs](#)

[21.4 Defense against network attack](#)

[21.5 Cryptography: the ragged boundary](#)

[21.6 CAs and PKI](#)

[21.7 Topology](#)

[21.8 Summary](#)

[Research problems](#)

[Further reading](#)

[Notes](#)

[CHAPTER 22: Phones](#)

[22.1 Introduction](#)

[22.2 Attacks on phone networks](#)

[22.3 Going mobile](#)

[22.4 Platform security](#)

[22.5 Summary](#)

[Research problems](#)

[Further reading](#)

[Notes](#)

CHAPTER 23: Electronic and Information Warfare

23.1 Introduction

23.2 Basics

23.3 Communications systems

23.4 Surveillance and target acquisition

23.5 IFF systems

23.6 Improvised explosive devices

23.7 Directed energy weapons

23.8 Information warfare

23.9 Summary

Research problems

Further reading

Note

CHAPTER 24: Copyright and DRM

24.1 Introduction

24.2 Copyright

24.3 DRM on general-purpose computers

24.4 Information hiding

24.5 Policy

24.6 Accessory control

24.7 Summary

Research problems

Further reading

Notes

CHAPTER 25: New Directions?

25.1 Introduction

25.2 Autonomous and remotely-piloted vehicles

25.3 AI / ML

[25.4 PETS and operational security](#)

[25.5 Elections](#)

[25.6 Summary](#)

[Research problems](#)

[Further reading](#)

[Notes](#)

[PART III](#)

[CHAPTER 26: Surveillance or Privacy?](#)

[26.1 Introduction](#)

[26.2 Surveillance](#)

[26.3 Terrorism](#)

[26.4 Censorship](#)

[26.5 Forensics and rules of evidence](#)

[26.6 Privacy and data protection](#)

[26.7 Freedom of information](#)

[26.8 Summary](#)

[Research problems](#)

[Further reading](#)

[Notes](#)

[CHAPTER 27: Secure Systems Development](#)

[27.1 Introduction](#)

[27.2 Risk management](#)

[27.3 Lessons from safety-critical systems](#)

[27.4 Prioritising protection goals](#)

[27.5 Methodology](#)

[27.6 Managing the team](#)

[27.7 Summary](#)

[Research problems](#)

[Further reading](#)

[Notes](#)

[CHAPTER 28: Assurance and Sustainability](#)

[28.1 Introduction](#)

[28.2 Evaluation](#)

[28.3 Metrics and dynamics of dependability](#)

[28.4 The entanglement of safety and security](#)

[28.5 Sustainability](#)

[28.6 Summary](#)

[Research problems](#)

[Further reading](#)

[Notes](#)

[CHAPTER 29: Beyond “Computer Says No”](#)

[Bibliography](#)

[Index](#)

[End User License Agreement](#)

List of Illustrations

Chapter 1

[Figure 1.1: – Security Engineering Analysis Framework](#)

Chapter 2

[Figure 2.1: Muscular – the slide](#)

Chapter 4

[Figure 4.1: Password generator use](#)

[Figure 4.2: The MIG-in-the middle attack](#)

[Figure 4.3: The Mafia-in-the-middle attack](#)

Chapter 5

[Figure 5.1: Monoalphabetic substitution cipher](#)

[Figure 5.9: The random oracle](#)

[Figure 5.10: A simple 16-bit SP-network block cipher](#)

[Figure 5.11: The AES linear transformation, illustrated by its effect on byt...](#)

[Figure 5.12: The Feistel cipher structure](#)

[Figure 5.13: The DES round function](#)

[Figure 5.14: The Linux penguin, in clear and ECB encrypted \(from Wikipedia, ...](#)

[Figure 5.15: Cipher Block Chaining \(CBC\) mode](#)

[Figure 5.16: Feedforward mode \(hash function\)](#)

Chapter 6

[Figure 6.5: Stack smashing attack](#)

Chapter 8

[Figure 8.1: The market for apartments](#)

Chapter 9

[Figure 9.1: typical corporate policy language](#)

[Figure 9.3: The NRL pump](#)

[Figure 9.4: Insecure composition of secure systems with feedback](#)

[Figure 9.5: The cascade problem](#)

Chapter 10

[Figure 10.3: A lattice of security labels](#)

Chapter 11

[Figure 11.3: Table lattice for a database with three attributes](#)

Chapter 12

[Figure 12.1: Clay envelope and its content of tokens representing 7 jars of ...](#)

[Figure 12.2: Architecture of SWIFT](#)

[Figure 12.3: IBM method for generating bank card PINs](#)

[Figure 12.4: Card fraud in the UK from 2004 to 2018](#)

[Figure 12.5: A rigid wire is inserted through a hole in the Ingenico's conce...](#)

Chapter 13

[Figure 13.1: A cutaway pin-tumbler lock \(courtesy of Marc Weber Tobias\)](#)

[Figure 13.2: Key for a sidebar lock](#)

[Figure 13.3: Sidebar bump key](#)

Chapter 14

[Figure 14.1: A prepayment electricity meter \(courtesy of Schlumberger\)](#)

[Figure 14.2: A tachograph chart](#)

[Figure 14.3: A tachograph with an interruptor controlled by the driver using...](#)

[Figure 14.4: One of the new formats for US postal meters \(courtesy of Symbol...](#)

Chapter 15

[Figure 15.1: Shared control using geometry](#)

Chapter 16

[Figure 16.1: Scanning electron micrograph of paper \(courtesy Ingenia Technol...](#)

[Figure 16.2: A wristband seal from our local swimming pool](#)

Chapter 17

[Figure 17.1: The prints in the McKie case](#)

[Figure 17.2: an iris with iris code \(courtesy John Daugman\)](#)

Chapter 18

[Figure 18.1: The IBM 4758 cryptoprocessor \(courtesy of Steve Weingart\)](#)

[Figure 18.2: The 4758 partially opened showing \(from top left downward\) the ...](#)

[Figure 18.3: Our probing station](#)

[Figure 18.4: The data bus of an ST16 smartcard prepared for probing by excav...](#)

[Figure 18.5: The protective mesh of an ST16 smartcard with a FIB cross for p...](#)

[Figure 18.6: SX28 microcontroller with 'glue logic' \(courtesy of Sergei Skor...](#)

Chapter 19

[Figure 19.1: RF signal from a Toshiba laptop reconstructed several rooms awa...](#)

[Figure 19.2: Normal text](#)

[Figure 19.3: Text low-pass filtered](#)

[Figure 19.4: Screen, normal text](#)

[Figure 19.5: Screen, filtered text](#)

[Figure 19.6: Page of normal text](#)

[Figure 19.7: Page of filtered text](#)

[Figure 19.8: Hz AM signal](#)

[Figure 19.9: 1200 Hz AM signal](#)

[Figure 19.10: Plot of the current measured during 256 single attempts to gue...](#)

Chapter 21

[Figure 21.2: complex firewalls for an MLS network](#)

Chapter 22

[Figure 22.1: GSM authentication system components](#)

Chapter 23

[Figure 23.1: Spreading in DSSS \(courtesy of Roche and Dugelay\).](#)

[Figure 23.2: Unspreading in DSSS \(courtesy of Roche and Dugelay\).](#)

Chapter 24

[Figure 24.1: Cut-and-rotate scrambling](#)

[Figure 24.2: Scrambled video frame](#)

[Figure 24.3: Processed video frame](#)

[Figure 24.4: The multiplexer generator](#)

[Figure 24.5: Binary revocation tree](#)

[Figure 24.6: The Mosaic attack \(courtesy Jet Photographic, \[www.jetphotograph...\]\(http://www.jetphotograph.com\)](#)

Chapter 27

[Figure 27.2: Hazard elimination in motor reversing circuit](#)

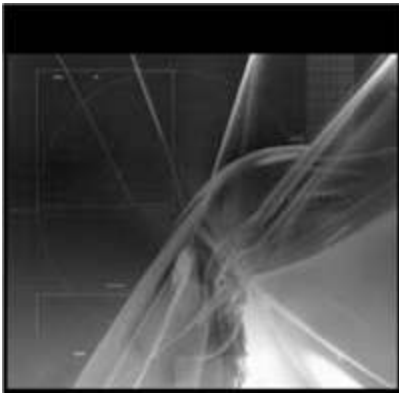
[Figure 27.3: A threat tree](#)

[Figure 27.4: The waterfall model](#)

[Figure 27.5: The spiral model](#)

Chapter 28

[Figure 28.1: Two infusion pumps that are apparently of the same model \(photo...](#)



Security Engineering

A Guide to Building Dependable Distributed Systems

Third Edition

Ross Anderson

WILEY

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