Blown to Bits

Your Life, Liberty, and Happiness After the Digital Explosion

Hal Abelson
Ken Ledeen
Harry Lewis

Addison-Wesley

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Cape Town • Sydney • Tokyo • Singapore • Mexico City

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U.S. Corporate and Government Sales
(800) 382-3419
corpsales@pearsontechgroup.com

For sales outside the United States, please contact:

International Sales
international@pearson.com

Visit us on the Web: www.informit.com/aw

Library of Congress Cataloging-in-Publication Data:

Abelson, Harold.
Blown to bits : your life, liberty, and happiness after the digital explosion / Hal Abelson, Ken Ledeen, Harry Lewis.
p. cm.
ISBN 0-13-713559-9 (hardback : alk. paper) 1. Computers and civilization. 2. Information technology—Technological innovations. 3. Digital media. I. Ledeen, Ken, 1946- II. Lewis, Harry R. III. Title.
QA76.9.C66A245 2008
303.48'33—dc22

2008005910

For information regarding permissions, write to:

Pearson Education, Inc.
Rights and Contracts Department
501 Boylston Street, Suite 900
Boston, MA 02116
Fax (617) 671 3447

ISBN-13: 978-0-13-713559-2
ISBN-10: 0-13-713559-9
Text printed in the United States on recycled paper at RR Donnelley in Crawfordsville, Indiana.
Third printing December 2008

Editor in Chief
Mark Taub

Acquisitions Editor
Greg Doench

Development Editor
Michael Thurston

Managing Editor
Gina Kanouse

Senior Project Editor
Kristy Hart

Copy Editor
Water Crest Publishing, Inc.

Indexer
Erika Millen

Proofreader
Williams Woods Publishing Services

Publishing Coordinator
Michelle Housley

Interior Designer and Composition
Nonie Ratcliff

Cover Designer
Chuti Prasertsith

CHAPTER 5

Secret Bits

How Codes Became Unbreakable

Encryption in the Hands of Terrorists, and Everyone Else

September 13, 2001. Fires were still smoldering in the wreckage of the World Trade Center when Judd Gregg of New Hampshire rose to tell the Senate what had to happen. He recalled the warnings issued by the FBI years before the country had been attacked: the FBI's most serious problem was "the encryption capability of the people who have an intention to hurt America." "It used to be," the senator went on, "that we had the capability to break most codes because of our sophistication." No more. "The technology has outstripped the code breakers," he warned. Even civil libertarian cryptographer Phil Zimmermann, whose encryption software appeared on the Internet in 1991 for use by human rights workers world-wide, agreed that the terrorists were probably encoding their messages. "I just assumed," he said, "somebody planning something so diabolical would want to hide their activities using encryption."

Encryption is the art of encoding messages so they can't be understood by eavesdroppers or adversaries into whose hands the messages might fall. De-scrambling an encrypted message requires knowing the sequence of symbols—the "key"—that was used to encrypt it. An encrypted message may be visible to the world, but without the key, it may as well be hidden in a locked box. Without the key—exactly the right key—the contents of the box, or the message, remains secret.

What was needed, Senator Gregg asserted, was "the cooperation of the community that is building the software, producing the software, and building the equipment that creates the encoding technology"—cooperation, that is, enforced by legislation. The makers of encryption software would have to enable the government to bypass the locks and retrieve the decrypted messages. And what about encryption programs written abroad, which could be shared around the world in the blink of an eye, as Zimmermann's had been? The U.S. should use "the market of the United States as leverage" in getting foreign manufacturers to follow U.S. requirements for "back doors" that could be used by the U.S. government.

By September 27, Gregg's legislation was beginning to take shape. The keys used to encrypt messages would be held in escrow by the government under tight security. There would be a "quasi-judicial entity," appointed by the Supreme Court, which would decide when law enforcement had made its case for release of the keys. Civil libertarians squawked, and doubts were raised as to whether the key escrow idea could actually work. No matter, opined the Senator in late September. "Nothing's ever perfect. If you don't try, you're never going to accomplish it. If you do try, you've at least got some opportunity for accomplishing it."

Abruptly, three weeks later, Senator Gregg dropped his legislative plan. "We are not working on an encryption bill and have no intention to," said the Senator's spokesman on October 17.

On October 24, 2001, Congress passed the USA PATRIOT Act, which gave the FBI sweeping new powers to combat terrorism. But the PATRIOT Act does not mention encryption. U.S. authorities have made no serious attempt to legislate control over cryptographic software since Gregg's proposal.

Why Not Regulate Encryption?

Throughout the 1990s, the FBI had made control of encryption its top legislative priority. Senator Gregg's proposal was a milder form of a bill, drafted by the FBI and reported out favorably by the House Select Committee on Intelligence in 1997, which would have mandated a five-year prison sentence for selling encryption products unless they enabled immediate decryption by authorized officials.

How could regulatory measures that law enforcement deemed critical in 1997 for fighting terrorism drop off the legislative agenda four years later, in the aftermath of the worst terrorist attack ever suffered by the United States of America?

No technological breakthrough in cryptography in the fall of 2001 had legislative significance. There also weren't any relevant diplomatic breakthroughs.

No other circumstances conspired to make the use of encryption by terrorists and criminals an unimportant problem. It was just that something else about encryption had become accepted as more important: the explosion of commercial transactions over the Internet. Congress suddenly realized that it had to allow banks and their customers to use encryption tools, as well as airlines and their customers, and eBay and Amazon and their customers. Anyone using the Internet for commerce needed the protection that encryption provided. Very suddenly, there were millions of such people, so many that the entire U.S. and world economy depended on public confidence in the security of electronic transactions.

The tension between enabling secure conduct of electronic commerce and preventing secret communication among outlaws had been in the air for a decade. Senator Gregg was but the last of the voices calling for restrictions on encryption. The National Research Council had issued a report of nearly 700 pages in 1996 that weighed the alternatives. The report concluded that on balance, efforts to control encryption would be ineffective, and that their costs would exceed any imaginable benefit. The intelligence and defense establishment was not persuaded. FBI Director Louis Freeh testified before Congress in 1997 that "Law enforcement is in unanimous agreement that the widespread use of robust non-key recovery [i.e., non-escrowed] encryption ultimately will devastate our ability to fight crime and prevent terrorism."

Yet only four years later, even in the face of the September 11th attack, the needs of commerce admitted no alternative to widespread dissemination of encryption software to every business in the country, as well as to every home computer from which a commercial transaction might take place. In 1997, average citizens, including elected officials, might never have bought anything online. Congress members' families might not have been regular computer users. By 2001, all that had changed—the digital explosion was happening. Computers had become consumer appliances, Internet connections were common in American homes—and awareness of electronic fraud had become widespread. Consumers did not want their credit card numbers, birthdates, and Social Security numbers exposed on the Internet.

Why is encryption so important to Internet communications that Congress was willing to risk terrorists using encryption, so that American businesses and consumers could use it too? After all, information security is not a new need. People communicating by postal mail, for example, have reasonable assurances of privacy without any use of encryption.

The answer lies in the Internet's open architecture. Bits move through the Internet not in a continuous stream, but in discrete blocks, called *packets*. A packet consists of about 1500 bytes, no more (see the Appendix). Data packets are not like envelopes sent through postal mail, with an address on the outside and contents hidden. They are like postcards, with everything exposed for anyone to see. As the packets move through the Internet, they are steered on their way by computers called *routers,* which are located at the switching points. Every data packet gets handled at every router: stored, examined, checked, analyzed, and sent on its way. Even if all the fibers and wires could be secured, wireless networks would allow bits to be grabbed out of the air without detection.

If you send your credit card number to a store in an ordinary email, you might as well stand in Times Square and shout it at the top of your lungs. By 2001, a lot of credit card numbers were traveling as bits though glass fibers and through the air, and it was impossible to prevent snoopers from looking at them.

The way to make Internet communications secure—to make sure that no one but the intended recipient knows what is in a message—is for the sender to encrypt the information so that only the recipient can decrypt it. If that can be accomplished, then eavesdroppers along the route from sender to receiver can examine the packets all they want. All they will find is an undecipherable scramble of bits.

In a world awakening to Internet commerce, encryption could no longer be thought of as it had been from ancient times until the turn of the third millennium: as armor used by generals and diplomats to protect information critical to national security. Even in the early 1990s, the State Department demanded that an encryption researcher register as an international arms dealer. Now suddenly, encryption was less like a weapon and more like the armored cars used to transport cash on city streets, except that these armored cars were needed by everyone. Encryption was no longer a munition; it was money.

The commoditization of a critical military tool was more than a technology shift. It sparked, and continues to spark, a rethinking of fundamental notions of privacy and of the balance between security and freedom in a democratic society.

"The question," posed MIT's Ron Rivest, one of the world's leading cryptographers, during one of the many debates over encryption policy that occurred during the 1990s, "is whether people should be able to conduct private conversations, immune from government surveillance, even when that surveillance is fully authorized by a Court order." In the post-2001 atmosphere that produced the PATRIOT Act, it's far from certain that Congress would have responded to Rivest's question with a resounding "Yes." But by 2001, commercial realities had overtaken the debates.

To fit the needs of electronic commerce, encryption software had to be widely available. It had to work perfectly and quickly, with no chance of anyone cracking the codes. And there was more: Although encryption had been used for more than four millennia, no method known until the late twentieth century would have worked well enough for Internet commerce. But in 1976, two young mathematicians, operating outside the intelligence community that was the center of cryptography research, published a paper that made a reality out of a seemingly absurd scenario: Two parties work out a secret key that enables them to exchange messages securely—even if they have never met and all their messages to each other are in the open, for anyone to hear. With the invention of *public-key cryptography*, it became possible for every man, woman, and child to transmit credit card numbers to Amazon more securely than any general had been able to communicate military orders fifty years earlier, orders on which the fate of nations depended.

Historical Cryptography

Cryptography—"secret writing"—has been around almost as long as writing itself. Ciphers have been found in Egyptian hieroglyphics from as early as 2000 B.C. A *cipher* is a method for transforming a message into an obscured form, together with a way of undoing the transformation to recover the message. Suetonius, the biographer of the Caesars, describes Julius Caesar's use of a cipher in his letters to the orator Cicero, with whom he was planning and plotting in the dying days of the Roman Republic: "... if he [Caesar] had anything confidential to say, he wrote it in cipher, that is, by so changing the order of the letters of the alphabet, that not a word could be made out. If anyone wishes to decipher these, and get at their meaning, he must substitute the fourth letter of the alphabet, namely D, for A, and so with the others." In other words, Caesar used a letter-by-letter translation to encrypt his messages:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
DEFGHIJKLMNOPQRSTUVWXYZABC
```

To encrypt a message with Caesar's method, replace each letter in the top row by the corresponding letter in the bottom row. For example, the opening of Caesar's Commentaries "Gallia est omnis divisa in partes tres" would be encrypted as:

Plaintext: `GALLIA EST OMNIS DIVISA IN PARTES TRES`

Ciphertext: `JDOOLD HVW RPQLV GLYLVD LQ SDUWHV WUHV`

The original message is called the *plaintext* and the encoded message is called the *ciphertext.* Messages are decrypted by doing the reverse substitutions.

This method is called the *Caesar shift* or the *Caesar cipher.* The encryption/decryption rule is easy to remember: "Shift the alphabet three places." Of course, the same idea would work if the alphabet were shifted more than three places, or fewer. The Caesar cipher is really a family of ciphers, with 25 possible variations, one for each different amount of shifting.

Caesar ciphers are very simple, and an enemy who knew that Caesar was simply shifting the plaintext could easily try all the 25 possible shifts of the alphabet to decrypt the message. But Caesar's method is a representative of a larger class of ciphers, called *substitution ciphers*, in which one symbol is substituted for another according to a uniform rule (the same letter is always translated the same way).

There are a great many more substitution ciphers than just shifts. For example, we could scramble the letters according to the rule

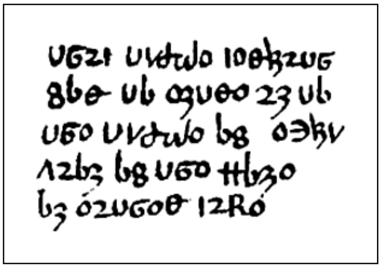
```
ABCDEFGHIJKLMNOPQRSTUVWXYZ

XAPZRDWIBMQEOFTYCGSHULJVKN
```

so that A becomes X, B becomes A, C becomes P, and so on. There is a similar substitution for every way of reordering the letters of the alphabet. The number of different reorderings is
$26 \times 25 \times 24 \times \cdots \times 3 \times 2$

which is about 4×10^{26} different methods—ten thousand times the number of stars in the universe! It would be impossible to try them all. General substitution ciphers must be secure—or so it might seem.

Breaking Substitution Ciphers

In about 1392, an English author—once thought to be the great English poet Geoffrey Chaucer, although that is now disputed—wrote a manual for use of an astronomical instrument. Parts of this manual, which was entitled *The Equatorie of the Planetis*, were written in a substitution cipher (see Figure 5.1). This puzzle is not as hard as it looks, even though there is very little ciphertext with which to work. We know it is written in English—Middle English, actually—but let's see how far we can get thinking of it as encrypted English.



Folio 30v of Peterson MS 75.1, *The Equatorie of Planetis*, a 14th century manuscript held at University of Cambridge.

FIGURE 5.1 Ciphertext in *The Equatorie of Planetis* (1392).

Although this looks like gibberish, it contains some patterns that may be clues. For example, certain symbols occur more frequently than others. There are twelve ɵs and ten υs, and no other symbol occurs as frequently as these. In ordinary English texts, the two most frequently occurring letters are E and T, so a fair guess is that these two symbols correspond to these two letters. Figure 5.2 shows what happens if we assume that ɵ = E and υ = T. The pattern υGɵ appears twice and apparently represents a three-letter word beginning with T and ending with E. It could be TIE or TOE, but THE seems more likely, so a reasonable assumption is that G = H. If that is true, what is the four-letter word at the beginning of the text, which begins with TH? Not THAT, because it ends with a new symbol, nor THEN, because the third letter is also new. Perhaps THIS. And there is a two-letter word beginning with T that appears twice in the second line—that must be TO. Filling in the equivalencies for H, I, S, and O yields Figure 5.3.

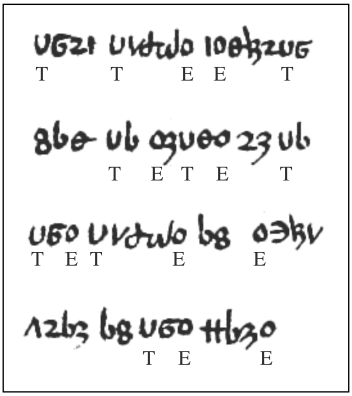


FIGURE 5.2 *Equatorie* ciphertext, with the two most common symbols assumed to stand for E and T.

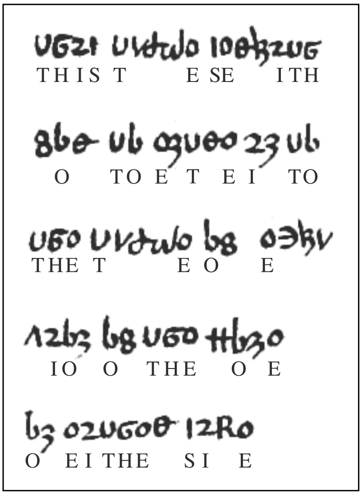

THIS T E SE ITH
O TO E T E I TO
THE T E O E
IO O THE O E
O EI THE SI E

FIGURE 5.3 *Equatorie* ciphertext, with more conjectural decodings.

At this point, the guessing gets easier—probably the last two words are EITHER SIDE—and the last few symbols can be inferred with a knowledge of Middle English and some idea of what the text is about. The complete plaintext is: *This table servith for to entre in to the table of equacion of the mone on either side* (see Figure 5.4).

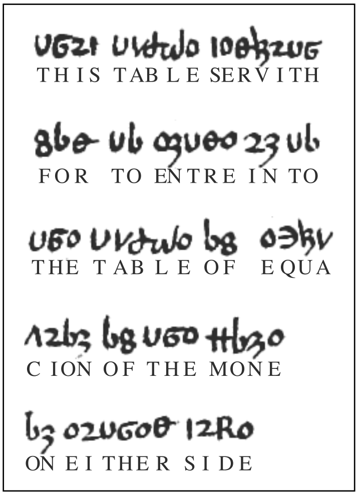

THIS TABLE SERVITH
FOR TO ENTRE IN TO
THE TABLE OF EQUA
C ION OF THE MONE
ON EITHER SIDE

FIGURE 5.4 *Equatorie* ciphertext, fully decoded.

The technique used to crack the code is *frequency analysis*: If the cipher is a simple substitution of symbols for letters, then crucial information about which symbols represent which letters can be gathered from how often the various symbols appear in the ciphertext. This idea was first described by the Arabic philosopher and mathematician Al-Kindi, who lived in Baghdad in the ninth century.

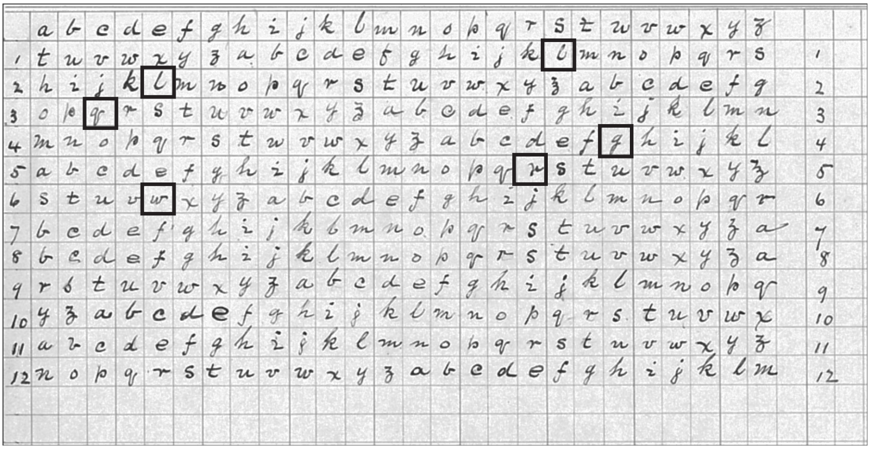
By the Renaissance, this kind of informed guesswork had been reduced to a fine art that was well known to European governments. In a famous example of the insecurity of substitution ciphers, Mary Queen of Scots was beheaded in 1587 due to her misplaced reliance on a substitution cipher to conceal her correspondence with plotters against Queen Elizabeth I. She was not the last to have put too much confidence in an encryption scheme that looked hard to crack, but wasn't. Substitution ciphers were in common use as late as the 1800s, even though they had been insecure for a millennium by that time! Edgar Allen Poe's mystery story *The Gold Bug* (1843) and A. Conan Doyle's Sherlock Holmes mystery *Adventure of the Dancing Men* (1903) both turn on the decryption of substitution ciphers.

Secret Keys and One-Time Pads

In cryptography, every advance in code-breaking yields an innovation in code-making. Seeing how easily the *Equatorie* code was broken, what could we do to make it more secure, or *stronger*, as cryptographers would say? We might use more than one symbol to represent the same plaintext letter. A method named for the sixteenth-century French diplomat Blaise de Vigenère uses multiple Caesar ciphers. For example, we can pick twelve Caesar ciphers and use the first cipher for encrypting the 1st, 13th, and 25th letters of the plaintext; the second cipher for encrypting the 2nd, 14th, and 26th plaintext letters; and so on. Figure 5.5 shows such a Vigenère cipher. A plaintext message beginning SECURE... would be encrypted to produce the ciphertext *llqgrw...*, as indicated by the boxed characters in the figure—S is encrypted using the first row, E is encrypted using the second row, and so on. After we use the bottom row of the table, we start again at the top row, and repeat the process over and over.

We can use the cipher of Figure 5.5 without having to send our correspondent the entire table. Scanning down the first column spells out *thomasbbryan*, which is the key for the message. To communicate using Vigenère encryption, the correspondents must first agree on a key. They then use the key to construct a substitution table for encrypting and decrypting messages.

When SECURE was encrypted as *llqgrw*, the two occurrences of E at the second and sixth positions in the plaintext were represented by different ciphertext letters, and the two occurrences of the ciphertext letter *l* represented different plaintext letters. This illustrates how the Vigenère cipher confounds simple frequency analysis, which was the main tool of cryptanalysts at the time. Although the idea may seem simple, the discovery of the Vigenère cipher is regarded as a fundamental advance in cryptography, and the method was considered to be unbreakable for hundreds of years.



	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	
1	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	1
2	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	2
3	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	3
4	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	4
5	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	5
6	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	6
7	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	7
8	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	8
9	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	9
10	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	10
11	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	11
12	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	12

Harvard University Archives.

FIGURE 5.5 A Vigenère cipher. The key, *thomasbbryan*, runs down the second column. Each row represents a Caesar cipher in which the shift amount is determined by a letter of the key. (Thomas B. Bryan was an attorney who used this code for communicating with a client, Gordon McKay, in 1894.)

CRYPTOGRAPHY AND HISTORY

Cryptography (code-making) and cryptanalysis (code-breaking) have been at the heart of many momentous events in human history. The intertwined stories of diplomacy, war, and coding technology are told beautifully in two books: *The Code-Breakers*, revised edition, by David Kahn (Scribner's, 1996) and *The Code Book* by Simon Singh (Anchor paperback, 2000).

Cryptographers use stock figures for describing encryption scenarios: Alice wants to send a message to Bob, and Eve is an adversary who may be eavesdropping.

Suppose Alice wants to send Bob a message (see Figure 5.6). The lock-and-key metaphor goes this way: Alice puts the message in a box and locks the box, using a key that only she and Bob possess. (Imagine that the lock on Alice's box is the kind that needs the key to lock it as well as to open it.) If Eve intercepts the box in transit, she has no way to figure out what key to use to open it. When Bob receives the box, he uses his copy of the key to open it. As long as the key is kept secret, it doesn't matter that others can see that there is a box with something in it, and even what kind of lock is on the box. In the same way, even if an encrypted message comes with an announcement that it is encrypted using a Vigenère cipher, it will not be easy to decrypt, except by someone who has the key.

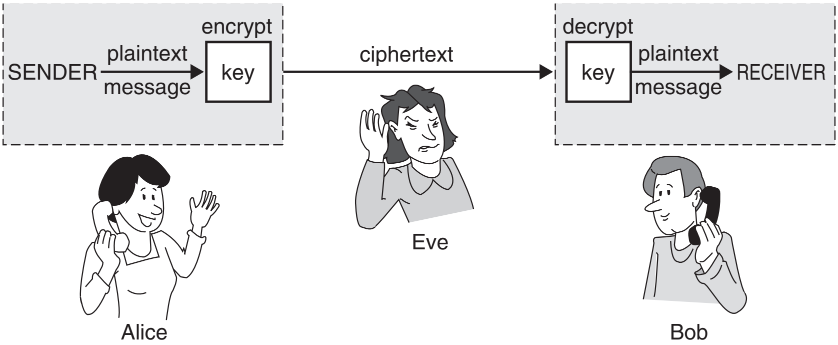


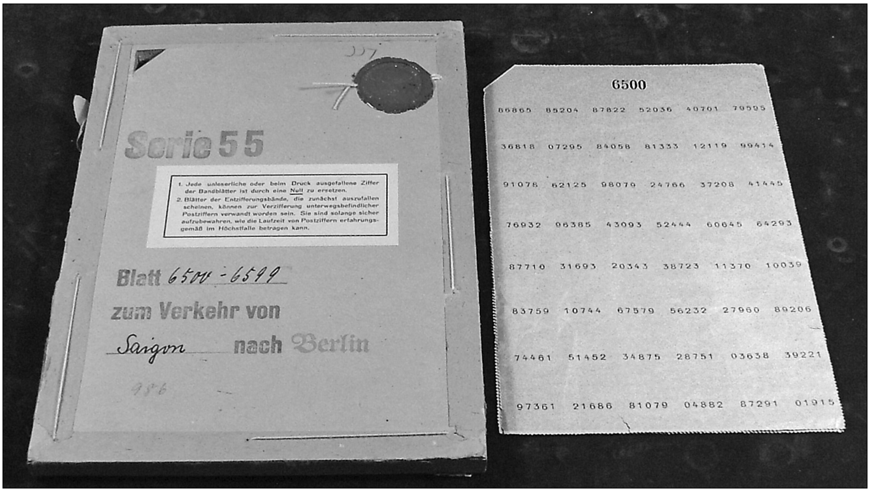
FIGURE 5.6 Standard cryptographic scenario. Alice wants to send a message to Bob. She encrypts it using a secret key. Bob decrypts it using his copy of the key. Eve is an eavesdropper. She intercepts the coded message in transit, and tries to decrypt it.

Or at least that's the idea. The Vigenère cipher was actually broken in the mid 1800s by the English mathematician Charles Babbage, who is now recognized as a founding figure in the field of computing. Babbage recognized that if someone could guess or otherwise deduce the length of the key, and hence the length of the cycle on which the Vigenère cipher was repeated, the problem was reduced to breaking several simple substitutions. He then used a brilliant extension of frequency analysis to discover the length of the key. Babbage never published his technique, perhaps at the request of British Intelligence. A Prussian Army officer, William Kasiski, independently figured out how to break the Vigenère code and published the method in 1863. The Vigenère cipher has been insecure ever since.

The sure way to beat this attack is to use a key that is as long as the plaintext, so that there are no repetitions. If we wanted to encrypt a message of length 100, we might use 100 Caesar ciphers in an arrangement like that of Figure 5.5, extended to 100 rows. Every table row would be used only once. A code like this is known as a *Vernam cipher*, after its World War I-era inventor, AT&T telegraph engineer Gilbert Vernam, and is more commonly referred to as a *one-time pad.*

The term "one-time pad" is based on a particular physical implementation of the cipher. Let's again imagine that Alice wants to get a message to Bob. Alice and Bob have identical pads of paper. Each page of the pad has a key written on it. Alice uses the top page to encrypt a message. When Bob receives it, he uses the top page of his pad to decrypt the message. Both Alice and Bob tear off and destroy the top page of the pad when they have used it. It is essential that the pages not be re-used, as doing so could create patterns like those exploited in cracking the Vigenère cipher.

One-time pads were used during the Second World War and the Cold War in the form of booklets filled with digits (see Figure 5.7). Governments still use one-time pads today for sensitive communications, with large amounts of keying material carefully generated and distributed on CDs or DVDs.



1. Jede unleserliche oder beim Druck ausgefallene Ziffer der Bandblätter ist durch eine Null zu ersetzen.
2. Blätter der Entzifferungsbände, die zunächst auszufallen scheinen, können zur Verzifferung unterwegsbefindlicher Postziffern verwandt worden sein. Sie sind solange sicher aufzubewahren, wie die Laufzeit von Postziffern erfahrungsgemäß im Höchstfalle betragen kann.

86865	85204	87822	52036	40701	79595
36818	07295	84058	81333	12119	99414
91078	62125	98079	24766	37208	41445
76932	96385	43093	52444	60645	64293
87710	31693	20343	38723	11370	10039
83759	10744	67579	56232	27960	89206
74461	51452	34875	28751	03638	39221
97361	21686	81079	04882	87291	01915

National Security Agency.

FIGURE 5.7 German one-time pad used for communication between Berlin and Saigon during the 1940s. Encrypted messages identified the page to be used in decryption. The cover warns, "Sheets of this encryption book that seem to be unused could contain codes for messages that are still on their way. They should be kept safe for the longest time a message might need for delivery."

A one-time pad, if used correctly, cannot be broken by cryptanalysis. There are simply no patterns to be found in the ciphertext. There is a deep relation between information theory and cryptography, which Shannon explored in 1949. (In fact, it was probably his wartime research on this sensitive subject that gave birth to his brilliant discoveries about communication in general.) Shannon proved mathematically what is obvious intuitively: The one-time pad is, in principle, as good as it gets in cryptography. It is absolutely unbreakable—in theory.

But as Yogi Berra said, "In theory, there is no difference between theory and practice. In practice, there is." Good one-time pads are hard to produce. If the pad contains repetitions or other patterns, Shannon's proof that one-time pads are uncrackable no longer holds. More seriously, transmitting a pad between the parties without loss or interception is likely to be just as difficult as communicating the plaintext of the message itself without detection. Typically, the parties would share a pad ahead of time and hope to conceal it in their travels. Big pads are harder to conceal than small pads, however, so the temptation arises to re-use pages—the kiss of death for security.

The Soviet KGB fell victim to exactly this temptation, which led to the partial or complete decryption of over 3000 diplomatic and espionage messages by U.S. and British intelligence during the years 1942–1946. The National Security Agency's VENONA project, publicly revealed only in 1995, was responsible for exposing major KGB agents such as Klaus Fuchs and Kim Philby. The Soviet messages were doubly encrypted, using a one-time pad on top of other techniques; this made the code-breaking project enormously difficult. It was successful only because, as World War II wore on and material conditions deteriorated, the Soviets re-used the pads.

Because one-time pads are impractical, almost all encryption uses relatively short keys. Some methods are more secure than others, however. Computer programs that break Vigenère encryption are readily available on the Internet, and no professional would use a Vigenère cipher today. Today's sophisticated ciphers are the distant descendents of the old substitution methods. Rather than substituting message texts letter for letter, computers divide the ASCII-encoded plaintext message into blocks. They then transform the bits in the block according to some method that depends on a key. The key itself is a sequence of bits on which Alice and Bob must agree and keep secret from Eve. Unlike the Vigenère cipher, there are no known shortcuts for breaking these ciphers (or at least none known publicly). The best method to decrypt a ciphertext without knowing the secret key seems to be brute-force exhaustive search, trying all possible keys.

The amount of computation required to break a cipher by exhaustive search grows exponentially in the size of the key. Increasing the key length by one bit doubles the amount of work required to break the cipher, but only slightly increases the work required to encrypt and decrypt. This is what makes these ciphers so useful: Computers may keep getting faster—even at an exponential rate—but the work required to break the cipher can also be made to grow exponentially by picking longer and longer keys.

Lessons for the Internet Age

Let's pause for a moment to consider some of the lessons of cryptographic history—morals that were well-understood by the early twentieth century. In the late twentieth century, cryptography changed drastically because of modern computer technology and new cryptographic algorithms, but these lessons are still true today. They are too often forgotten.

Breakthroughs Happen, but News Travels Slowly

Mary Stuart was beheaded when her letters plotting against Elizabeth were deciphered by frequency analysis, which Al-Kindi had described nine centuries earlier. Older methods have also remained in use to the present day, even for high-stakes communications. Suetonius explained the Caesar cipher in the first century A.D. Yet two millennia later, the Sicilian Mafia was still using the code. Bernardo Provenzano was a notorious Mafia boss who managed to stay on the run from Italian police for 43 years. But in 2002, some *pizzini*—ciphertexts typed on small pieces of paper—were found in the possession of one of his associates. The messages included correspondence between Bernardo and his son Angelo, written in a Caesar cipher—with a shift of three, exactly as Suetonius had described it. Bernardo switched to a more secure code, but the dominos started to topple. He was finally traced to a farmhouse and arrested in April 2006.

Even scientists are not immune from such follies. Although Babbage and Kasiski had broken the Vigenère cipher in the mid-nineteenth century, *Scientific American* 50 years later described the Vigenère method as "impossible of translation."

Encoded messages tend to look indecipherable. The incautious, whether naïve or sophisticated, are lulled into a false sense of security when they look at apparently unintelligible jumbles of numbers and letters. Cryptography is a science, and the experts know a lot about code-breaking.

Confidence Is Good, but Certainty Would Be Better

There are no guarantees that even the best contemporary ciphers won't be broken, or haven't been broken already. Some of the ciphers have the potential to be validated by mathematical proofs, but actually providing those proofs will require deep mathematical breakthroughs. If anyone knows how to break modern codes, it is probably someone in the National Security

Agency or a comparable agency of a foreign government, and those folks don't tend to say much publicly.

In the absence of a formal proof of security, all one can do is to rely on what has been dubbed the Fundamental Tenet of Cryptography: *If lots of smart people have failed to solve a problem, then it probably won't be solved (soon).*

Of course, that is not a very useful principle in practice—by definition, breakthroughs are unlikely to happen "soon." But they do happen, and when they do, indigestion among cryptographers is widespread. In August 2004, at an annual cryptography conference, researchers announced that they had been able to break a popular algorithm (MD5) for computing cryptographic operations called *message digests*, which are fundamental security elements in almost all web servers, password programs, and office products. Cryptographers recommended switching to a stronger algorithm (SHA-1) but within a year, weaknesses were uncovered in this method as well.

A provably secure encryption algorithm is one of the holy grails of computer science. Every weakness exposed in proposed algorithms yields new ideas about how to make them stronger. We aren't there yet, but progress is being made.

A provably secure encryption algorithm is one of the holy grails of computer science.

Having a Good System Doesn't Mean People Will Use It

Before we explain that unbreakable encryption may finally be possible, we need to caution that even mathematical certainty would not suffice to create perfect security, if people don't change their behavior.

Vigenère published his encryption method in 1586. But foreign-office cipher secretaries commonly avoided the Vigenère cipher because it was cumbersome to use. They stayed with simple substitution ciphers—even though it was well-known that these ciphers were readily broken—and they hoped for the best. By the eighteenth century, most European governments had skilled "Black Chambers" through which all mail to and from foreign embassies was routed for decryption. Finally, the embassies switched to Vigenère ciphers, which themselves continued to be used after information about how to crack them had become widely known.

And so it is today. Technological inventions, no matter how solid in theory, will not be used for everyday purposes if they are inconvenient or expensive. The risks of weak systems are often rationalized in attempts to avoid the trouble of switching to more secure alternatives.

In 1999, an encryption standard known as WEP (Wired Equivalent Privacy) was introduced for home and office wireless connections. In 2001, however, WEP was found to have serious flaws that made it easy to eavesdrop on wireless networks, a fact that became widely known in the security community. Despite this, wireless equipment companies continued to sell WEP products, while industry pundits comforted people that "WEP is better than nothing." A new standard (WPA—Wi-Fi Protected Access) was finally introduced in 2002, but it wasn't until September 2003 that products were required to use the new standard in order to be certified. Hackers were able to steal more than 45 million credit and debit card records from TJX, the parent company of several major retail store chains, because the company was still using WEP encryption as late as 2005. That was long after WEP's insecurities were known and WPA was available as a replacement. The cost of that security breach has reached the hundreds of millions of dollars.

Similarly, many of today's "smart card" systems that use RFID (Radio Frequency Identification) tags are insecure. In January 2005, computer scientists from Johns Hopkins University and RSA Data Security announced that they had cracked an RFID-based automobile anti-theft and electronic payment system built into millions of automobile key tags. They demonstrated this by making multiple gasoline purchases at an Exxon/Mobile station. A spokesman for Texas Instruments, which developed the system, countered that the methods the team used were "wildly beyond the reach of most researchers," saying "I don't see any reason to change this approach."

When encryption was a military monopoly, it was possible in principle for a commander to order everyone to start using a new code if he suspected that the enemy had cracked the old one. The risks of insecure encryption today arise from three forces acting in consort: the high speed at which news of insecurities travels among experts, the slow speed at which the inexpert recognize their vulnerabilities, and the massive scale at which cryptographic software is deployed. When a university researcher discovers a tiny hole in an algorithm, computers everywhere become vulnerable, and there is no central authority to give the command for software upgrades everywhere.

The Enemy Knows Your System

The last lesson from history may seem counterintuitive. It is that a cryptographic method, especially one designed for widespread use, should be regarded as more reliable if it is widely known and seems not to have been broken, rather than if the method itself has been kept secret.

The Flemish linguist Auguste Kerckhoffs articulated this principle in an 1883 essay on military cryptography. As he explained it,

> The system must not require secrecy, and it could fall into the hands of the enemy without causing trouble.... Here I mean by system, not the key itself, but the material part of the system: tables, dictionaries, or whatever mechanical apparatus is needed to apply it. Indeed, it's not necessary to create imaginary phantoms or to suspect the integrity of employees or subordinates, in order to understand that, if a system requiring secrecy were to find itself in the hands of too many individuals, it could be compromised upon each engagement in which any of them take part.

In other words, if a cryptographic method is put in widespread use, it is unrealistic to expect that the method can remain secret for long. Thus, it should be designed so that it will remain secure, even if everything but a small amount of information (the key) becomes exposed.

Claude Shannon restated Kerckhoffs's Principle in his paper on systems for secret communication: "... we shall assume that *the enemy knows the system being used*." He went on to write:

> The assumption is actually the one ordinarily used in cryptographic studies. It is pessimistic and hence safe, but in the long run realistic, since one must expect his system to be found out eventually.

Kerckhoffs's Principle is frequently violated in modern Internet security practice. Internet start-up companies routinely make bold announcements about new breakthrough proprietary encryption methods, which they refuse to subject to public scrutiny, explaining that the method must be kept secret in order to protect its security. Cryptographers generally regard such "security through obscurity" claims with extreme skepticism.

Even well-established organizations run afoul of Kerckhoffs's Principle. The Content Scrambling System (CSS) used on DVDs (Digital Versatile Disks) was developed by a consortium of motion picture studios and consumer electronics companies in 1996. It encrypts DVD contents in order to limit unauthorized copying. The method was kept secret to prevent the manufacture of unlicensed DVD players. The encryption algorithm, which consequently was never widely analyzed by experts, turned out to be weak and was cracked within three years after it was announced. Today, CSS decryption programs, together with numerous unauthorized "ripped" DVD contents, circulate widely on the Internet (see Chapter 6, "Balance Toppled" for a more detailed discussion of copy protection).

Kerckhoffs's Principle has been institutionalized in the form of encryption standards. The *Data Encryption Standard* (DES) was adopted as a national standard in the 1970s and is widely used in the worlds of business and finance. It has pretty much survived all attempts at cracking, although the inexorable progress of Moore's Law has made exhaustive searching through all possible keys more feasible in recent years. A newer standard, Advanced Encryption Standard (AES), was adopted in 2002 after a thorough and public review. It is precisely because these encryption methods are so widely known that confidence in them can be high. They have been subjected to both professional analysis and amateur experimentation, and no serious deficiencies have been discovered.

These lessons are as true today as they ever were. And yet, something else, something fundamental about cryptography, is different today. In the late twentieth century, cryptographic methods stopped being state secrets and became consumer goods.

Secrecy Changes Forever

For four thousand years, cryptography was about making sure Eve could not read Alice's message to Bob if Eve intercepted the message *en route*. Nothing could be done if the key itself was somehow discovered. Keeping the key secret was therefore of inestimable importance, and was a very uncertain business.

If Alice and Bob worked out the key when they met, how could Bob keep the key secret during the dangers of travel? Protecting keys was a military and diplomatic priority of supreme importance. Pilots and soldiers were instructed that, even in the face of certain death from enemy attack, their first responsibility was to destroy their codebooks. Discovery of the codes could cost thousands of lives. The secrecy of the codes was everything.

And if Alice and Bob never met, then how could they agree on a key without *already* having a secure method for transmitting the key? That seemed like a fundamental limitation: Secure communication was practical only for people who could arrange to meet beforehand, or who had access to a prior method of secure communication (such as military couriers) for carrying the key between them. If Internet communications had to proceed on this assumption, electronic commerce never could have gotten off the ground. Bit packets racing through the network are completely unprotected from eavesdropping.

And then, in the 1970s, everything changed. Whitfield Diffie was a 32-year-old mathematical free spirit who had been obsessed with cryptography since his years as an MIT undergraduate. 31-year-old Martin Hellman was a hard-nosed graduate of the Bronx High School of Science and an Assistant Professor at Stanford. Diffie had traveled the length of the country in search of collaborators on the mathematics of secret communication. This was not an easy field to enter, since most serious work in this area was being done behind the firmly locked doors of the National Security Agency. Ralph Merkle, a 24-year-old computer science graduate student, was exploring a new approach to secure communication. In the most important discovery in the entire history of cryptography, Diffie and Hellman found a practical realization of Merkle's ideas, which they presented in a paper entitled "New Directions in Cryptography." This is what the paper described:

> *A way for Alice and Bob, without any prior arrangement, to agree on a secret key, known only to the two of them, by using messages between them that are not secret at all.*

In other words, as long as Alice and Bob can communicate with each other, they can establish a secret key. It does not matter if Eve or anyone else can hear everything they say. Alice and Bob can come to a consensus on a secret key, and there is no way for Eve to use what she overhears to figure out what that secret key is. This is true even if Alice and Bob have never met before and have never made any prior agreements.

It was revealed in 1997 that the same public-key techniques had been developed within the British secret Government Communication Headquarters (GCHQ) two years before Diffie and Hellman's work, by James Ellis, Clifford Cocks, and Malcolm Williamson.

The impact of this discovery cannot be overstated. The art of secret communication was a government monopoly, and had been since the dawn of writing—governments had the largest interests in secrets, and the smartest scientists worked for governments. But there was another reason why governments had done all the serious cryptography. Only governments had the wherewithal to assure the production, protection, and distribution of the keys on which secret communication depended. If the secret keys could be produced by public communication, everyone could use cryptography. They just had to know how; they did not need armies or brave couriers to transmit and protect the keys.

Diffie, Hellman, and Merkle dubbed their discovery "public-key cryptography." Although its significance was not recognized at the time, it is the invention that made electronic commerce possible. If Alice is you and Bob is Amazon, there is no possibility of a meeting—how could you physically go to Amazon to procure a key? Does Amazon even *have* a physical location? If Alice is to send her credit card number to Amazon securely, the encryption has to be worked out on the spot, or rather, on the two separate spots separated by the Internet. Diffie-Hellman-Merkle, and a suite of related methods that followed, made secure Internet transactions possible. If you have ever ordered anything from an online store, you have been a cryptographer without realizing it. Your computer and the store's computer played the roles of Alice and Bob.

It seems wildly counterintuitive that Alice and Bob could agree on a secret key over a public communication channel. It was not so much that the scientific community had tried and failed to do what Diffie, Hellman, and Merkle did. It never occurred to them to try, because it seemed so obvious that Alice had to give Bob the keys somehow.

Even the great Shannon missed this possibility. In his 1949 paper that brought all known cryptographic methods under a unified framework, he did not realize that there might be an alternative. "The key must be transmitted by non-interceptable means from transmitting to receiving points," he wrote.

Alice and Bob can get the same secret key, even though all their messages are intercepted.

Not true. Alice and Bob can get the same secret key, even though all their messages are intercepted.

The basic picture of how Alice communicates her secret to Bob remains as shown in Figure 5.6. Alice sends Bob a coded message, and Bob uses a secret key to decrypt it. Eve may intercept the ciphertext *en route*.

The goal is for Alice to do the encryption in such a way that it is *impossible* for Eve to decrypt the message in any way other than a brute-force search through all possible keys. If the decryption problem is "hard" in this sense, then the phenomenon of exponential growth becomes the friend of Alice and Bob. For example, suppose they are using ordinary decimal numerals as keys, and their keys are ten digits long. If they suspect that Eve's computers are getting powerful enough to search through all possible keys, they can switch to 20-digit keys. The amount of time Eve would require goes up by a factor of 10^{10} = 10,000,000,000. Even if Eve's computers were powerful enough to crack any 10-digit key in a second, it would then take her more than 300 years to crack a 20-digit key!

Exhaustive search is always *one* way for Eve to discover the key. But if Alice encrypts her message using a substitution or Vigenère cipher, the encrypted message will have patterns that enable Eve to find the key far more quickly. The trick is to find a means of encrypting the message so that the ciphertext reveals no patterns from which the key could be inferred.

The Key Agreement Protocol

The crucial invention was the concept of a *one-way computation*—a computation with two important properties: It can be done quickly, but it can't be undone quickly. To be more precise, the computation quickly combines two numbers x and y to produce a third number, which we'll call $x * y$. If you know the value of $x * y$, there is no quick way to figure out what value of y was used to produce it, even if you also know the value of x. That is, if you know the values of x and the result z, the only way to find a value of y so that $z = x * y$ is trial and error search. Such an exhaustive search would take time that grows exponentially with the number of digits of z—practically impossible, for numbers of a few hundred digits. Diffie and Hellman's one-way computation also has an important third property: $(x * y) * z$ always produces the same result as $(x * z) * y$.

The key agreement protocol starts from a base of public knowledge: how to do the computation $x * y$, and also the value of a particular large number g. (See the Endnotes for the details.) All this information is available to the entire world. Knowing it, here is how Alice and Bob proceed.

1. Alice and Bob each choose a random number. We'll call Alice's number a and Bob's number b. We'll refer to a and b as Alice and Bob's *secret keys*. Alice and Bob keep their secret keys secret. *No one except Alice knows the value of* a, *and no one except Bob knows the value of* b.

2. Alice calculates $g * a$ and Bob calculates $g * b$. (Not hard to do.) The results are called their *public keys* A and B, respectively.

3. Alice sends Bob the value of A and Bob sends Alice the value of B. It doesn't matter if Eve overhears these communications; A and B are not secret numbers.

4. When she has received Bob's public key B, Alice computes $B * a$, using her secret key a as well as Bob's public key B. Likewise, when Bob receives A from Alice, he computes $A * b$.

Even though Alice and Bob have done different computations, they have ended up with the same value. Bob computes $A * b$, that is, $(g * a) * b$ (see Step 2—A is $g * a$). Alice computes $B * a$, that is, $(g * b) * a$. Because of the

Are We Sure No One Can Crack the Code?

No one has proved mathematically that the public-key encryption algorithms are unbreakable, in spite of determined efforts by top mathematicians and computer scientists to provide absolute proof of their security. So our confidence in them rests on the Fundamental Tenet: *No one has broken them so far.* If anyone knows a fast method, it's probably the National Security Agency, which operates in an environment of extreme secrecy. Maybe the NSA knows how and isn't telling. Or maybe some inventive loner has cracked the code but prefers profit to celebrity, and is quietly socking away huge profits from decoding messages about financial transactions. Our bet is that no one knows how and no one will.

third property of the one-way computation, that number is $(g * a) * b$ once again—the same value, arrived at in a different way!

This shared value, call it K, is the key Alice and Bob will use for encrypting and decrypting their subsequent messages, using whatever standard method of encryption they choose.

Now here's the crucial point. Suppose Eve has been listening to Alice and Bob's communications. Can she do anything with all the information she has? She has overheard A and B, and she knows g because it is an industry standard. She knows all the algorithms and protocols that Alice and Bob are using; Eve has read Diffie and Hellman's paper too! But to compute the key K, Eve would have to know one of the secret keys, either a or b. She doesn't—only Alice knows a and only Bob knows b. On numbers of a few hundred digits, no one knows how to find a or b from g, A, and B without searching through impossibly many trial values.

Alice and Bob can carry out their computations with personal computers or simple special-purpose hardware. But even the most powerful computers aren't remotely fast enough to let Eve break the system, at least not by any method known.

Exploiting this difference in computational effort was Diffie, Hellman, and Merkle's breakthrough. They showed how to create shared secret keys, without requiring secure channels.

Public Keys for Private Messages

Suppose Alice wants to have a way for anyone in the world to send her encrypted messages that only she can decrypt. She can do this with a small variation of the key-agreement protocol. All the computations are the same as in the key agreement protocol, except they take place in a slightly different order.

Alice picks a secret key a and computes the corresponding public key A. She publishes A in a directory.

If Bob (or anyone) now wants to send Alice an encrypted message, he gets Alice's public key from the directory. Next, he picks his own secret key b and computes B as before. He also uses Alice's public key A from the directory to compute an encryption key K just as with the key-agreement protocol: $K = A * b$. Bob uses K as a key to encrypt a message to Alice, and he sends Alice the ciphertext, along with B. Because he uses K only once, K is like a one-time pad.

When Alice receives Bob's encrypted message, she takes the B that came with message, together with her secret key a, just as in the key agreement protocol, and computes the same $K = B * a$. Alice now uses K as the key for decrypting the message. Eve can't decrypt it, because she doesn't know the secret keys.

With public-key encryption, anyone can send encrypted mail to anyone over an insecure, publicly exposed communication path.

This might seem like just a simple variant of key agreement, but it results in a major conceptual change in how we think about secure communication. With public-key encryption, *anyone* can send encrypted mail to *anyone* over an insecure, publicly exposed communication path. The only thing on which they need to agree is to use the Diffie-Hellman-Merkle method—and knowing that is of no use to an adversary trying to decipher an intercepted message.

Digital Signatures

In addition to secret communication, a second breakthrough achievement of public-key cryptography is preventing forgeries and impersonations in electronic transactions.

Suppose Alice wants to create a public announcement. How can people who see the announcement be sure that it really comes from Alice—that it's not a forgery? What's required is a method for marking Alice's public message in such a way that anyone can easily verify that the mark is Alice's and no one can forge it. Such a mark is called a *digital signature*.

To build on the drama we have used already, we'll continue to talk about Alice sending a message to Bob, with Eve trying to do something evil while the message is in transit. In this case, however, we are not concerned with the secrecy of Alice's message—only with assuring Bob that what he receives is really what Alice sent. In other words, the message may not be secret—perhaps it is an important public announcement. Bob needs to be confident that the signature he sees on the message is Alice's and that the message could not have been tampered with before he received it.

Digital signature protocols use public keys and secret keys, but in a different way. The protocol consists of two computations: one Alice uses to process her message to create the signature, and one Bob uses to verify the signature. Alice uses her secret key and the message itself to create the signature. Anyone can then use Alice's public key to verify the signature. The point is that everyone can know the public key and thus verify the signature, but only the person who knows the secret key could have produced the signature. This is the reverse of the scenario of the previous section, where anyone can encrypt a message, but only the person with the secret key can decrypt it.

A digital signature scheme requires a computational method that makes signing easy if you have the secret key and verifying easy if you have the public key—and yet makes it computationally infeasible to produce a verifiable signature if you don't know the secret key. Moreover, the signature depends on the message as well as on the secret key of the person signing it. Thus, the digital signature protocol attests to the *integrity* of the message—that it was not tampered with in transit—as well as to its *authenticity*—that the person who sent it really is Alice.

In typical real systems, used to sign unencrypted email, for example, Alice doesn't encrypt the message itself. Instead, to speed up the signature computation, she first computes a compressed version of the message, called a *message digest*, which is much shorter than the message itself. It requires less computation to produce the signature for the digest than for the full message. How message digests are computed is public knowledge. When Bob receives Alice's signed message, he computes the digest of the message and verifies that it is identical to what he gets by decrypting the attached signature using Alice's public key.

The digesting process needs to produce a kind of fingerprint—something small that is nonetheless virtually unique to the original. This compression process must avoid a risk associated with using digests. If Eve could produce a different message with the same digest, then she could attach Alice's signature to Eve's message. Bob would not realize that someone had tampered with the message before he received it. When he went through the verification process, he would compute the digest of Eve's message, compare it to the result of decrypting the signature that Alice attached to Alice's message, and find them identical. This risk is the source of the insecurity of the message digest function MD5 mentioned earlier in this chapter, which is making the cryptographic community wary about the use of message digests.

RSA

Diffie and Hellman introduced the concept of digital signatures in their 1976 paper. They suggested an approach to designing signatures, but they did not present a concrete method. The problem of devising a practical digital signature scheme was left as a challenge to the computer science community.

The challenge was met in 1977 by Ron Rivest, Adi Shamir, and Len Adleman of the MIT Laboratory for Computer Science. Not only was the RSA (Rivest-Shamir-Adleman) algorithm a practical digital signature scheme, but it could also be used for confidential messaging. With RSA, each person generates a pair of keys—a public key and a secret key. We'll again call Alice's public key A and her secret key a. The public and private keys are inverses: If you transform a value with a, then transforming the result with A recovers the original value. If you transform a value with A, then transforming the result with a recovers the original value.

Here's how RSA key pairs are used. People publish their public keys and keep their secret keys to themselves. If Bob wants to send Alice a message, he picks a standard algorithm such as DES and a key K, and transforms K using Alice's public key A. Alice transforms the result using her secret key a to recover K. As with all public-key encryption, only Alice knows her secret key, so only Alice can recover K and decrypt the message.

To produce a digital signature, Alice transforms the message using her secret key a and uses the result as the signature to be sent along with the message. Anyone can then check the signature by transforming it with Alice's public key A to verify that this matches the original message. Because only Alice knows her secret key, only Alice could have produced something that, when transformed with her public key, will reproduce the original message.

It seems to be infeasible in the RSA cryptosystem—as in the Diffie-Hellman-Merkle system—to compute a secret key corresponding to a public key. RSA uses a different one-way computation than the one used by the Diffie-Hellman-Merkle system. RSA is secure only if it takes much longer to factor an n-digit number than to multiply two $n/2$-digit numbers. RSA's reliance on the difficulty of factoring has engendered enormous interest in finding fast ways to factor numbers. Until the 1970s, this was a mathematical pastime of theoretical interest only. One can multiply numbers in time comparable to *the number of digits*, while factoring a number requires effort comparable to *the value of the number itself*, as far as anyone knows. A breakthrough in factoring would render RSA useless and would undermine many of the current standards for Internet security.

A breakthrough in factoring would render RSA useless and would undermine many of the current standards for Internet security.

Certificates and Certification Authorities

There's a problem with the public-key methods we've described so far. How can Bob know that the "Alice" he's communicating with really is Alice? Anyone could be at the other end of the key-agreement communication pretending to be Alice. Or, for secure messaging, after Alice places her public key in the directory, Eve might tamper with the directory, substituting her own key in place of Alice's. Then, anyone who tries to use the key to create secret messages intended for Alice, will actually be creating messages that Eve, not Alice, can read. If "Bob" is you and "Alice" is the mayor ordering an evacuation of the city, some impostor could be trying to create a panic. If "Bob" is your computer and "Alice" is your bank's, "Eve" could be trying to steal your money!

This is where digital signatures can help. Alice goes to a trusted authority, to which she presents her public key together with proof of her identity. The authority digitally signs Alice's key—producing a signed key called a *certificate*. Now, instead of just presenting her key when she wants to communicate, Alice presents the certificate. Anyone who wants to use the key to communicate with Alice first checks the authority's signature to see that the key is legitimate.

People check a certificate by checking the trusted authority's signature. How do they know that the signature on the certificate really is the trusted authority's signature, and not some fraud that Eve set up for the purpose of issuing fake certificates? The authority's signature is itself guaranteed by another certificate, signed by another authority, and so on, until we reach an authority whose certificate is

Commercial Certificates

VeriSign, which is currently the major commercial certification authority, issues three classes of personal certificates. Class 1 is for assuring that a browser is associated with a particular email address and makes no claims about anyone's real identity. Class 2 provides a modest level of identity checking. Organizations issuing them should require an application with information that can be checked against employee records or credit records. Class 3 certificates require applying in person for verification of identity.

well-known. In this way, Alice's public key is vouched for, not only by a certificate and a single signature, but by a chain of certificates, each one with a signature guaranteed by the next certificate.

Organizations that issue certificates are called *certification authorities.* Certification authorities can be set up for limited use (for example, a corporation might serve as a certification authority that issues certificates for use on its corporate network). There are also companies that make a business of selling certificates for public use. The trust you should put in a certificate depends on two things: your assessment of the reliability of the signature on the certificate and *also* your assessment of the certification authority's policy in being willing to sign things.

Cryptography for Everyone

In real life, none of us is aware that we are carrying out one-way computations while we are browsing the Web. But every time we order a book from Amazon, check our bank or credit card balance, or pay for a purchase using PayPal, that is exactly what happens. The tell-tale sign that an encrypted web transaction is taking place is that the URL of the web site begins with "`https`" (the "s" is for "secure") instead of "`http`." The consumer's computer and the computer of the store or the bank negotiate the encryption, using public key cryptography—unbeknownst to the human beings involved in the transaction. The store attests to its identity by presenting a certificate signed by a Certification authority that the consumer's computer has been preconfigured to recognize. New keys are generated for each new transaction. Keys are cheap. Secret messages are everywhere on the Internet. We are all cryptographers now.

We are all cryptographers now.

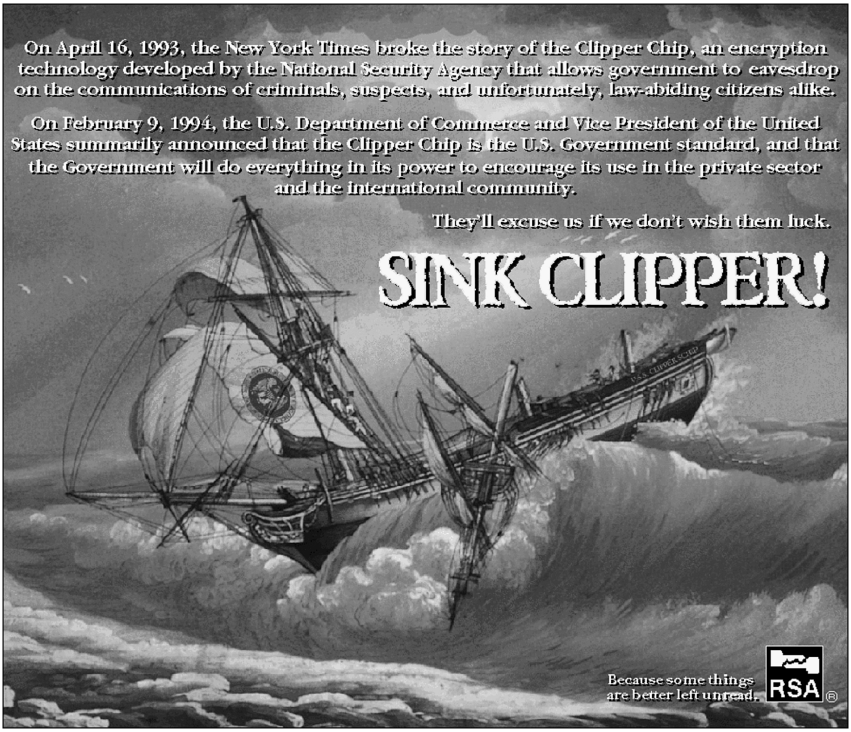
At first, public-key encryption was treated as a mathematical curiosity. Len Adleman, one of the inventors of RSA, thought that the RSA paper would be "the least interesting paper I would ever be on." Even the National Security Agency, as late as 1977, was not overly concerned about the spread of these methods. They simply did not appreciate how the personal computer revolution, just a few years away, would enable anyone with a home PC to exchange encrypted messages that even NSA could not decipher.

But as the 1980s progressed, and Internet use increased, the potential of ubiquitous cryptography began to become apparent. Intelligence agencies became increasingly concerned, and law enforcement feared that encrypted communications could put an end to government wiretapping, one of its most powerful tools. On the commercial side, industry was beginning to appreciate that customers would want private communication, especially in an era of electronic commerce. In the late 1980s and early 1990s, the Bush and the Clinton administrations were floating proposals to control the spread of cryptographic systems.

In 1994, the Clinton administration unveiled a plan for an "Escrowed Encryption Standard" that would be used on telephones that provided encrypted communications. The technology, dubbed "Clipper," was an encryption chip developed by the NSA that included a *back door*—an extra key held by the government, which would let law enforcement and intelligence agencies decrypt the phone communications. According to the proposal, the government would purchase only Clipper phones for secure communication. Anyone wanting to do business with the government over a secure telephone would also have to use a Clipper phone. Industry reception was cold, however (see Figure 5.8), and the plan was dropped. But in a sequence of modified proposals beginning in 1995, the White House attempted to convince industry to create encryption products that had similar back doors. The carrot here, and the stick, was export control law. Under U.S. law, cryptographic products could not be exported without a license, and violating export controls could result in severe criminal penalties. The administration proposed that encryption software would receive export licenses only if it contained back doors.

The ensuing, often heated negotiations, sometimes referred to as the "crypto wars," played out over the remainder of the 1990s. Law enforcement and national security argued the need for encryption controls. On the other side of the debate were the technology companies, who did not want government regulation, and civil liberties groups, who warned against the potential for growing communication surveillance. In essence, policymakers could not come to grips with the transformation of a major military technology into an everyday personal tool.

We met Phil Zimmermann at the beginning of this chapter, and his career now becomes a central part of the story. Zimmermann was a journeyman programmer and civil libertarian who had been interested in cryptography since his youth. He had read a *Scientific American* column about RSA encryption in 1977, but did not have access to the kinds of computers that would be needed to implement arithmetic on huge integers, as the RSA algorithms demanded. But computers will get powerful enough if you wait. As the 1980s progressed, it became possible to implement RSA on home computers. Zimmermann set about to produce encryption software for the people, to counter the threat of increased government surveillance. As he later testified before Congress:



On April 16, 1993, the New York Times broke the story of the Clipper Chip, an encryption technology developed by the National Security Agency that allows government to eavesdrop on the communications of criminals, suspects, and unfortunately, law-abiding citizens alike.

On February 9, 1994, the U.S. Department of Commerce and Vice President of the United States summarily announced that the Clipper Chip is the U.S. Government standard, and that the Government will do everything in its power to encourage its use in the private sector and the international community.

They'll excuse us if we don't wish them luck.

Reprinted with permission of RSA Security, Inc.

FIGURE 5.8 Part of the "crypto wars," the furious industry reaction against the Clinton Administration's "Clipper chip" proposal.

The power of computers had shifted the balance towards ease of surveillance. In the past, if the government wanted to violate the privacy of ordinary citizens, it had to expend a certain amount of effort to intercept and steam open and read paper mail, or listen to and possibly transcribe spoken telephone conversations. This is analogous to catching fish with a hook and a line, one fish at a time. Fortunately for freedom and democracy, this kind of labor-intensive monitoring is not practical on a large scale. Today, electronic mail is gradually replacing conventional paper mail, and is soon to be the norm for everyone, not the novelty it is today. Unlike paper mail, e-mail messages are just too easy to intercept and scan for interesting keywords. This can be done easily, routinely, automatically, and undetectable on a grand scale. This is analogous to driftnet fishing—making a quantitative and qualitative Orwellian difference to the health of democracy.

Cryptography was the answer. If governments were to have unlimited surveillance powers over electronic communications, people everywhere needed easy-to-use, cheap, uncrackable cryptography so they could communicate without governments being able to understand them.

Zimmermann faced obstacles that would have stopped less-zealous souls. RSA was a patented invention. MIT had licensed it exclusively to the RSA Data Security Company, which produced commercial encryption software for corporations, and RSA Data Security had no interest in granting Zimmermann the license he would need to distribute his RSA code freely, as he wished to do.

And there was government policy, which was, of course, exactly the problem to which Zimmermann felt his encryption software was the solution. On January 24, 1991, Senator Joseph Biden, a co-sponsor of antiterrorist legislation Senate Bill 266, inserted some new language into the bill:

> It is the sense of Congress that providers of electronic communications services and manufacturers of electronic communications service equipment shall ensure that communications systems permit the government to obtain the plaintext contents of voice, data, and other communications when appropriate authorized by law.

This language received a furious reaction from civil liberties groups and wound up not surviving, but Zimmermann decided to take matters into his own hands.

By June of 1991, Zimmermann had completed a working version of his software. He named it PGP for "Pretty Good Privacy," after Ralph's mythical Pretty Good Groceries that sponsored Garrison Keillor's *Prairie Home Companion*. The software mysteriously appeared on several U.S. computers, available for anyone in the world to download. Soon copies were everywhere—not just in the U.S., but all over the world. In Zimmermann's own words: "This technology belongs to everybody." The genie was out of the bottle and was not going back in.

Zimmermann paid a price for his libertarian gesture. First, RSA Data Security was confident that this technology belonged to *it*, not to "everybody." The company was enraged that its patented technology was being given away. Second, the government was furious. It instituted a criminal investigation for violation of the export control laws, although it was not clear what laws, if any, Zimmermann had violated. Eventually MIT brokered an agreement that let Zimmermann use the RSA patent, and devised a way to put PGP on the Internet for use in the U.S., and in conformance with export controls.

By the end of the decade, the progress of electronic commerce had overtaken the key escrow debate, and the government had ended its criminal investigation without an indictment. Zimmermann built a business around PGP (see `www.pgp.com`), while still allowing free downloads for individuals. His web site contains testimonials from human rights groups in Eastern Europe and Guatemala attesting to the liberating force of secret communication among individuals and agencies working against oppressive regimes. Zimmermann had won.

Sort of.

Encryption Regulation Abroad

Some countries have adjusted to multiple uses of the same encryption algorithms, for commercial, military, and conspiratorial purposes. For example, the Chinese government strictly regulates the sale of encryption products, "to protect information safety, to safeguard the legal interests of citizens and organizations, and to ensure the safety and interests of the nation." In 2007, the United Kingdom enacted laws requiring the disclosure of encryption keys to government authorities investigating criminal or terror investigations, on penalty of up to five years in prison.

Cryptography Unsettled

Today, every banking and credit card transaction over the Web is encrypted. There is widespread concern about information security, identity theft, and degradation of personal privacy. PGP and other high-quality email encryption programs are widely available—many for free.

But very little email is encrypted today. Human rights groups use encrypted email. People with something to hide probably encrypt their email. But most of us don't bother encrypting our email. In fact, millions of people use Gmail, willingly trading their privacy for the benefits of free, reliable service. Google's computers scan every email, and supply advertisements related to the subject matter. Google might turn over email to the government in response to a court order, without challenging the demand. Why are we so unconcerned about email privacy?

Why are we so unconcerned about email privacy?

First, there is still little awareness of how easily our email can be captured as the packets flow through the Internet. The password requests needed to get our email out of the mail server may provide the

Spying on Citizens

Historically, spying on citizens required a warrant (since citizens have an expectation of privacy), but spying on foreigners did not. A series of executive orders and laws intended to combat terrorism allow the government to inspect bits that are on their way into or out of the country. (Perhaps even a phone call to an airline, if it is answered by a call center in India.) Also excluded from judicial oversight is any "surveillance directed at a person reasonably believed to be located outside of the United States," whether that person is a U.S. citizen or not. Such developments may stimulate encryption of electronic communications, and hence in the end prove to be counterproductive. That in turn might renew efforts to criminalize encryption of email and telephone communications in the U.S.

illusion of security, but they do nothing to protect the messages themselves from being sniffed as they float through fibers, wires, and the air. The world's biggest eavesdropping enterprise is very poorly known. It is the international ECHELON system, which automatically monitors data communications to and from satellites that relay Internet traffic. ECHELON is a cooperative project of the U.S. and several of its allies, and is the descendant of communications intelligence systems from the time of the Second World War. But it is up-to-date technologically. If your email messages use words that turn up in ECHELON's dictionary, they may get a close look.

Second, there is little concern because most ordinary citizens feel they have little to hide, so why would anyone bother looking? They are not considering the vastly increased capacity for automatic monitoring that governments now possess—the driftnet monitoring of which Zimmermann warned.

Finally, encrypted email is not built into the Internet infrastructure in the way encrypted web browsing is. You have to use nonstandard software, and the people you communicate with have to use some compatible software. In commercial settings, companies may not want to make encryption easy for office workers. They have an interest—and in many cases, regulatory requirements—to watch out for criminal activities. And they may not want to suggest that email is being kept private if they are unable to make that guarantee, out of fear of liability if unsecured email falls into the wrong hands.

It is not just email and credit card numbers that might be encrypted. Instant Messaging and VoIP telephone conversations are just packets flowing through the Internet that can be encrypted like anything else. Some Internet phone software (such as Skype) encrypts conversations, and there are several other products under development—including one led by Zimmermann himself—to create easy-to-use encryption software for Internet telephone conversations. But for the most part, digital communications are open, and Eve the evil eavesdropper, or anyone else, can listen in.

✷

Overall, the public seems unconcerned about privacy of communication today, and the privacy fervor that permeated the crypto wars a decade ago is nowhere to be seen. In a very real sense, the dystopian predictions of both sides of that debate are being realized: On the one hand, encryption technology is readily available around the world, and people can hide the contents of their messages, just as law enforcement feared—there is widespread speculation about Al Qaeda's use of PGP, for example. At the same time, the spread of the Internet has been accompanied by an increase in surveillance, just as the opponents of encryption regulation feared.

So although outright prohibitions on encryption are now impossible, the social and systems aspects of encryption remain in an unstable equilibrium. Will some information privacy catastrophe spark a massive re-education of the Internet-using public, or massive regulatory changes to corporate practice? Will some major supplier of email services and software, responding to consumers wary of information theft and government surveillance, make encrypted email the default option?

The bottom-line question is this: As encryption becomes as ordinary a tool for personal messages as it already is for commercial transactions, will the benefits to personal privacy, free expression, and human liberty outweigh the costs to law enforcement and national intelligence, whose capacity to eavesdrop and wiretap will be at an end?

Whatever the future of encrypted communication, encryption technology has another use. Perfect copies and instant communication have blown the legal notion of "intellectual property" into billions of bits of teenage movie and music downloads. Encryption is the tool used to lock movies so only certain people can see them and to lock songs so only certain people can hear them—to put a hard shell around this part of the digital explosion. The changed meaning of copyright is the next stop on our tour of the exploded landscape.