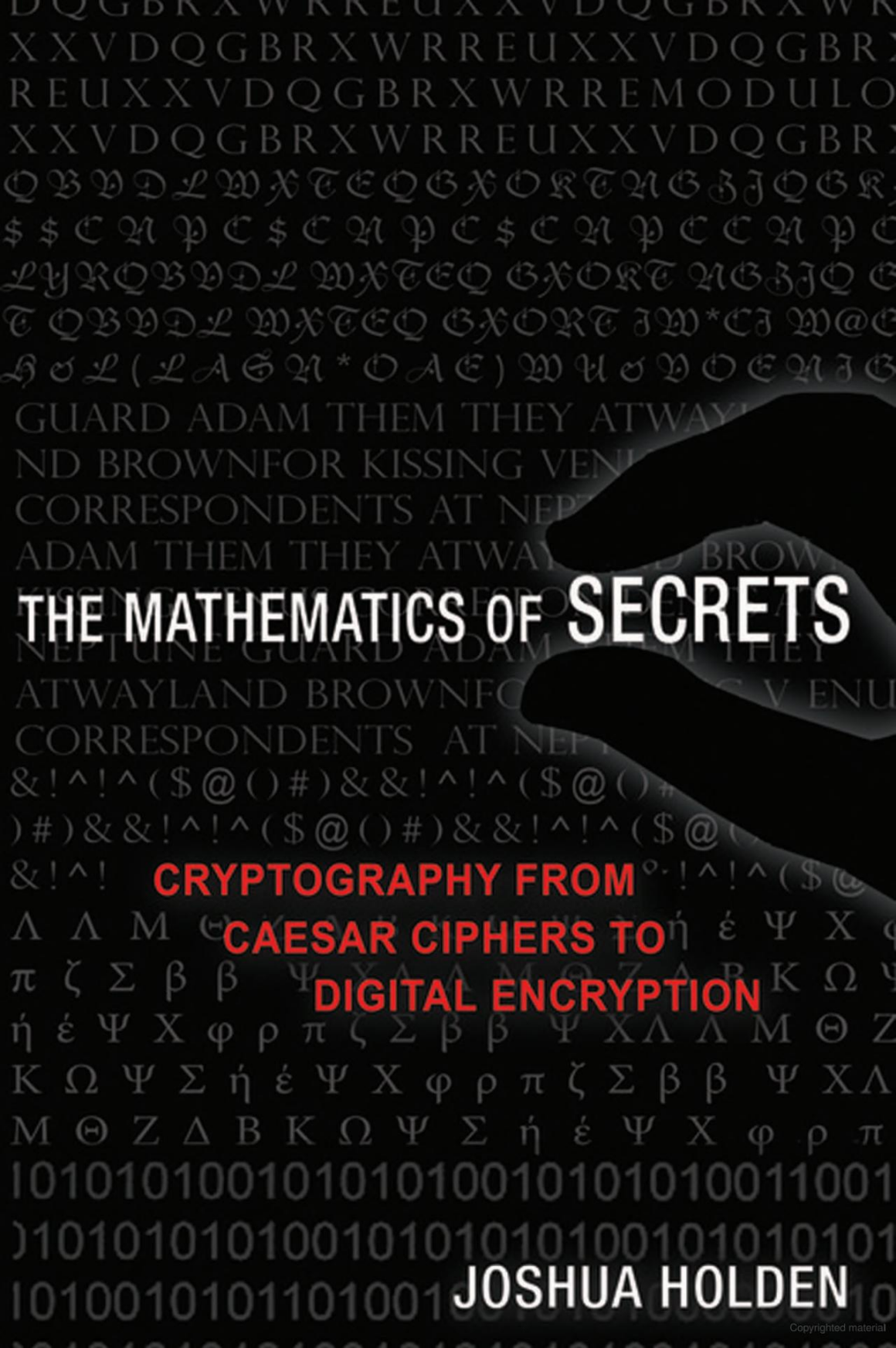


THE MATHEMATICS OF SECRETS

CRYPTOGRAPHY FROM CAESAR CIPHERS TO DIGITAL ENCRYPTION

JOSHUA HOLDEN

PRINCETON UNIVERSITY PRESS
PRINCETON AND OXFORD

Published by Princeton University Press, 41 William Street,
Princeton, New Jersey 08540
In the United Kingdom: Princeton University Press, 6 Oxford
Street, Woodstock, Oxfordshire OX20 1TR

press.princeton.edu

Cover image courtesy of Shutterstock; design by Lorraine Betz Doneker

First Paperback printing, 2019
Paper ISBN 9780691183312

The Library of Congress has cataloged the cloth edition as follows:

Names: Holden, Joshua, 1970– author.
Title: The mathematics of secrets : cryptography from Caesar
ciphers to digital encryption / Joshua Holden.
Description: Princeton : Princeton University Press, [2017] |
Includes bibliographical references and index.
Identifiers: LCCN 2016014840 | ISBN 9780691141756
(hardcover : alk. paper)
Subjects: LCSH: Cryptography—Mathematics. | Ciphers. |
Computer security.
Classification: LCC Z103 .H664 2017 | DDC 005.8/2—dc23 LC
record available at https://lccn.loc.gov/2016014840

British Library Cataloging-in-Publication Data is available

This book has been composed in Linux Libertine
Printed on acid-free paper. ∞

Printed in the United States of America

■ ■ ■ CONTENTS ■ ■ ■

■ ■ ■ PREFACE ■ ■ ■

This book is about the mathematics behind the modern science of sending secret messages, or cryptography. Modern cryptography *is* a science, and like all modern science, it relies on mathematics. Without the mathematics, you can only go so far in understanding cryptography. I want you to be able to go farther, not only because I think you should know about cryptography, but also because I think the particular kinds of mathematics the cryptographers use are really pretty, and I want to introduce you to them.

In *A Brief History of Time*, Stephen Hawking says that someone told him that each equation he included in the book would halve the sales. I hope that's not true of this book, because there are lots of equations. But I don't think the math is necessarily that hard. I once taught a class on cryptography in which I said that the prerequisite was high school algebra. Probably I should have said that the prerequisite was high school algebra and a willingness to think really hard about it. There's no trigonometry here, no calculus, no differential equations. There are some ideas that don't usually come up in an algebra course, and I'll try to walk you through them. If you want to really understand these ideas, you can do it without any previous college-level math—but you might have to think hard. (The math in some of the sidebars is a little harder, but you can skip those and still understand the rest of the book just fine.)

Mathematics isn't all there is to cryptography. Unlike most sciences, cryptography is about intelligent adversaries who are actively fighting over whether secrets will be revealed. Ian Cassels, who was both a prominent mathematician at Cambridge and a former British cryptanalyst from World War II, had a good perspective on this. He said that "cryptography is a mixture of mathematics and muddle, and without the muddle the mathematics can be used against you." In this book I've removed some of the muddle in order to focus on the mathematics. Some

professional cryptographers may take issue with that, because I am not really showing you the most secure systems that I could. In response, I can say only that this book for those interested in learning about a particular part of cryptography, namely, the mathematical foundations. There are many additional books in *Suggestions for Further Reading* and the Bibliography that you should read if you want to become a well-rounded professional.

Here is where I have drawn my personal line: I have tried not to say anything false in this book in the name of simplification, but I have left things out. I have left out some details of how to use the systems most securely, and I have left out some systems that I don't feel contribute to the mathematical story I want to tell. When possible, I have tried to present systems that have actually been used to protect real secrets. However, I have included some that were made up by me or another academic type when I feel that they best illustrate a point.

Computer technology has changed both the types of data with which cryptographers work and the techniques that are feasible. Some of the systems for protecting data that I discuss are either no longer applicable or no longer secure in today's world, even if they were in the past. Likewise, some of the techniques I discuss for breaking these systems are no longer effective in the forms presented here. Despite this, I feel that all the topics in this book illustrate issues that are still important and relevant to modern cryptography. I have tried to indicate how the principles are still used today, even when the actual systems are not. "Looking Forward" at the end of each chapter gives you a preview of how the chapter you just finished relates to the chapters yet to come or to future developments that I think are possible or likely.

A lot of the chapters follow the historical development of their topic, because that development is often a logical progression through the ideas I'm describing. History is also a good way to tell a story, so I like to use it when it fits. There's lots more about the history of cryptography out there, so if you would like to know more, definitely check out *Suggestions for Further Reading.*

I tell my students that I became a math professor because I like math and I like to talk. This book is me talking to you about a particular application of mathematics that I really like. My hope is that by the end of the book, you will really like it too.

■ ■ ■ ACKNOWLEDGMENTS ■ ■ ■

I wish I could individually thank everyone with whom I have ever had a good conversation about math or cryptography, but obviously I can't. I do want to single out some of the people who have particularly helped with my teaching of cryptography: by letting me sit in on their classes, by encouraging me, by teaching with me, or by sharing relevant materials. In roughly chronological order, these include David Hayes, Susan Landau (from whom I learned the "cosmic ray" principle, among many other cryptographic things), Richard Hain, Stephen Greenfield, Gary Sherman (from whom I learned the "shoes and socks" principle), and David Mutchler. I apologize if I've left anyone out.

Thank you to all the attendees of the Algorithmic Number Theory Symposia, particularly Carl Pomerance, Jon Sorenson, Hugh Williams, and all the members of Hugh's "posse" at (or formerly at) the University of Calgary. I'd also like to thank Brian Winkel, Craig Bauer, and the present and past members of the Editorial Board of *Cryptologia*. Without the friendship and encouragement of all of you, I'm sure my cryptography research would never have gotten off of the ground. And thanks go to all my research students at Rose-Hulman and at the Rose-Hulman Summer Research Experience for Undergraduates, who gave me the best reason to keep my research going.

This book has been in progress for a long time and many people have reviewed various drafts of it over the years. Many of you I don't know personally, and I don't even know some of your names, but thank you to all of you. Two people I particularly would like to thank are Jean Donaldson and Jon Sorenson. Jean volunteered to read a very early draft despite my being unable to offer any personal or professional incentive whatsoever. Not being a professional mathematician or cryptographer, she was the perfect audience and everything she said was immensely useful. Jon Sorenson likewise read an early draft and made encouraging

and helpful comments. In addition to being a reviewer, Jon has been a colleague and a friend for many years and has helped my career in numerous ways. Paul Nahin, David Kahn, and John MacCormick are also among those who gave me encouraging and helpful reviews.

The staff at Rose-Hulman's Logan Library have been invaluable through this process. Amy Harshbarger has come up with articles and technical reports through Interlibrary Loan that I thought would never be found. And Jan Jerrell let me keep library books far beyond the limits of a reasonable circulation policy. I thank them both, and everyone else at the library, profusely. Speaking of the library, Heather Chenette and Michelle Marincel Payne helped organize the "Shut Up and Write" group that met there and got me through the final revisions.

I could not have done this without the support and tolerance of my wife, Lana, our housemate, Richard, and the cats, who "tolerated" the occasional late dinner. You've put up with a lot through this process. I really appreciate it.

Finally, thank you to everyone at Princeton University Press, especially my editor, Vickie Kearn. Vickie first approached me about writing a cryptography book 12 years ago, and in all that time she never lost faith that it would happen some day. I can't believe it's finally finished. Thanks so much.

For the paperback edition: Heartfelt thanks to all of the readers who found typos and errors in the first edition, including Richard Bean, Chris Christensen, John Fuqua, Tom Jerardi, Steve Greenfield, Karst Koymans, Liu Mingxing, and David Miller. My apologies to anyone I've forgotten. Any errors I've still failed to fix are my own fault, not theirs. Thanks also to Sid Stamm and Nadine Shillingford Wondem for great advice on computer security issues.

THE MATHEMATICS OF SECRETS

1

Introduction to Ciphers and Substitution

1.1 ALICE AND BOB AND CARL AND JULIUS: TERMINOLOGY AND CAESAR CIPHER

People have been trying to hide the content of written messages almost as long as writing itself has existed and have developed a multitude of different methods of doing it. And almost as soon as people started trying to hide their messages, scholars started trying to classify and describe these methods. Unfortunately, this means that I've got to hit you straight up with a bunch of terminology. Even worse, a lot of words that are used interchangeably in ordinary conversation have specific meanings to experts in the field. It's not really hard to get the hang of what's what, though.

As our first example, people who study secret messages often use the terms **code** and **cipher** to mean two different things. David Kahn, author of perhaps the definitive account of the history of cryptography, said it about as well as anyone could: "A code consists of thousands of words, phrases, letters, and syllables with the codewords or codenumbers . . . that replace these plaintext elements . . . . In ciphers, on the other hand, the basic unit is the letter, sometimes the letter-pair . . . , very rarely larger groups of letters . . . ." A third method of sending secret messages, **steganography**, consists of concealing the very existence of the message, for example, through the use of invisible ink. In this book we will concentrate on ciphers as they are generally the most interesting mathematically, although examples of the other methods may come up from time to time.

A few more terms will be helpful before we get started. The study of how to send secret messages by codes and ciphers is called **cryptography**, whereas the study of how to read such secret messages without

permission is called **cryptanalysis**, or **codebreaking**. Together, the two fields make up the field of **cryptology**. (Sometimes cryptography is also used for the two fields combined, but we will try to keep the terms separate.)

It's become customary when talking about cryptology to talk about Alice, who wants to send a message to Bob. We're going to start with Julius, though. That's **Julius Caesar**, who in addition to being *dictator perpetuus* of Rome was also a military genius, a writer, and . . . a cryptographer.

Caesar probably wasn't the original inventor of what we now call the **Caesar cipher**, but he certainly made it popular. The Roman historian Suetonius describes the cipher:

> There are also letters of his [Caesar's] to Cicero, as well as to his intimates on private affairs, and in the latter, if he 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, whenever Alice wants to send a message, she first writes out the **plaintext**, or the text of the message in ordinary language. She is going to **encipher** this message, or put it into secret form using a cipher, and the result will be the **ciphertext** of the message. To put it into code would be to **encode** it, and the term **encrypt** can be used for either. For every a in the plaintext, Alice substitutes D in the ciphertext, for every b, she substitutes E, and so on. Each letter is shifted 3 letters down the alphabet. That's perfectly straightforward. The interesting part happens when Alice gets to the end of the alphabet and runs out of letters. The letter w becomes Z, so where does the letter x go? It wraps around, to A! The letter y becomes B and z becomes C. For example, the message "and you too, Brutus" becomes

plaintext:	a	n	d	y	o	u	t	o	o	b	r	u	t	u	s
ciphertext:	D	Q	G	B	R	X	W	R	R	E	U	X	W	X	V

This would be the message Alice sends to Bob.

You have actually used this "wraparound" idea in daily life since you were a child. What's 3 hours after 1:00? It's 4:00. Three hours after 2:00 is 5:00. What's 3 hours after 10:00? It's 1:00. You wrapped around.

It was around 1800 CE when **Carl Friedrich Gauss** codified this wraparound idea formally. It's now called **modular arithmetic**, and the wraparound number is called the **modulus**. A mathematician would write our wraparound clock example like this:

$$10 + 3 \equiv 1 \pmod{12}$$

and read it as "ten plus three is one modulo twelve."

But what about Caesar cipher? We can represent it using modular arithmetic if we are willing to change our letters into numbers. Instead of a think of the number 1, instead of b think of the number 2, and so on. This changing of letters to numbers is not really considered part of the secret cipher. It's a pretty obvious idea to those of us in the digital age, and Alice shouldn't really expect to keep it a secret. Only the operations that we do on the numbers are considered secret.

Now our modulus is 26 and our Caesar cipher looks like this.

plaintext	number	plus 3	ciphertext
a	1	4	D
b	2	5	E
⋮	⋮	⋮	⋮
x	24	1	A
y	25	2	B
z	26	3	C

Remember that the "plus 3" wraps around at 26.

To **decipher** the message, or take it from ciphertext to plaintext, Bob shifts three letters in the opposite direction, left. This time, he has to wrap around when he goes past a, or in terms of numbers, when he goes past 1. 0 wraps to 26, -1 wraps to 25, and so on. In the form we used earlier, that looks like the following.

ciphertext	number	minus 3	plaintext
A	1	24	x
B	2	25	y
C	3	26	z
⋮	⋮	⋮	⋮
Y	25	22	v
Z	26	23	w

1.2 THE KEY TO THE MATTER: GENERALIZING THE CAESAR CIPHER

From Caesar's point of view, he had a pretty secure cipher. After all, most of the people who might intercept one of his messages couldn't even read, much less analyze a cipher. However, from a modern cryptologic point of view it has a major drawback—once you have figured out that someone is using Caesar cipher, you know everything about the system. There's no **key**, or extra piece of information, that lets you vary the cipher. This is considered to be a very bad thing.

Stop to think about that a moment. What's the big deal? Your cipher is either secret or it isn't, right? That was the view in Caesar's time and for many centuries afterward. But in 1883, **Auguste Kerckhoffs** published a revolutionary essay, in which he stated, "The system must not require secrecy and can be stolen by the enemy without causing trouble." Amazing! How can having your system stolen not cause trouble?

Kerckhoffs went on to point out that it is just too easy for Eve the Eavesdropper to find out what system Alice and Bob are using. In Kerckhoffs' time, like Caesar's, cryptography was used mostly by militaries and governments, so Kerckhoffs was thinking about the information that an enemy might get through bribing or capturing a member of Alice or Bob's staff. These are still valid concerns in many situations today, and we can add to them the possibilities of Eve tapping phone lines, installing spyware on computers, and plain lucky guessing.

On the other hand, if Alice and Bob have a system that requires a key to encipher and decipher, then things aren't so bad. If Eve finds out what general system is being used, she still can't easily read any messages. Attempting to read a message without the key and/or determining the key used for a message is called **cryptanalyzing** the message or cipher or, more colloquially, **breaking** it. And even if she manages to

find out Alice and Bob's key, all is not lost. If Alice and Bob are smart, they are changing the key regularly. Since the basic system is the same, this isn't too hard, and then even if Eve gets the key to some of the messages, she won't be able to read all of them.

So we need to find a way to make small changes to Caesar cipher, depending on the value of some key. A logical place to start would be to ask why Alice is shifting her plaintext 3 places and not some other number? There is no particular reason; perhaps Caesar was just fond of the number 3. His successor, Augustus, used a similar system but shifted each letter only 1 place to the right. The "rot13" ("rot" stands for rotate) cipher shifts each letter forward by 13 places, wrapping around when you get to the end. This cipher is often used on the Internet to hide the punchlines of jokes or things that some people might find offensive. The general idea of shifting by k letters (or adding k modulo 26) is called a **shift cipher**, or **additive cipher**, with a key of k. For example, consider a shift cipher with a key of 21. Then Caesar's message would be:

plaintext:	a	n	d	y	o	u	t	o	o	b	r	u	t	u	s
numbers:	1	14	4	25	15	21	20	15	15	2	18	21	20	21	19
plus 21:	22	9	25	20	10	16	15	10	10	23	13	16	15	16	14
ciphertext:	V	I	Y	T	J	P	O	J	J	W	M	P	O	P	N

How many different keys are there? Shifting by 0 letters is probably not a good idea, but you could do it. Shifting by 26 letters is the same as shifting by 0 letters—or, in other words, 26 is the same as 0 modulo 26. Shifting by 27 letters is the same as shifting by 1 letter, and so on. So there are 26 ways of shifting that actually give you different results, or 26 keys. Note that this includes 0, the "stupid key," which doesn't do anything to the message. The technical term for when a cipher doesn't do anything is the **trivial cipher**. Suppose Alice sends Bob a message using a shift cipher and Eve intercepts it. Even if Eve has somehow learned that Alice and Bob are using a shift cipher, she still has to try 26 different keys in order to decipher the message. That's not a large number, but it's better than Caesar cipher.

Can we add some more keys? What about shifting our letters left instead of right? Unfortunately, that doesn't help. Suppose we shift our plaintext 1 letter to the left and wrap around the other direction.

plaintext:	a	n	d	y	o	u	t	o	o	b	r	u	t	u	s
numbers:	1	14	4	25	15	21	20	15	15	2	18	21	20	21	19
minus 1:	0	13	3	24	14	20	19	14	14	1	17	20	19	20	18
ciphertext:	Z	M	C	X	N	T	S	N	N	A	Q	T	S	T	R

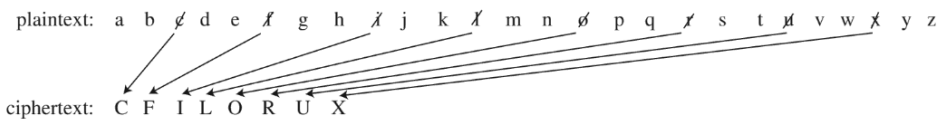
Note that since 0 is the same as 26 modulo 26, we can assign them both to the ciphertext letter "Z" interchangeably. If you think about it, you'll see that shifting 1 letter to the left is the same as shifting 25 letters to the right. Or in terms of modular arithmetic, you can think of left shifts as negative, so we are saying -1 is the same as 25 modulo 26. So left shifts don't help.

1.3 MULTIPLICATIVE CIPHERS

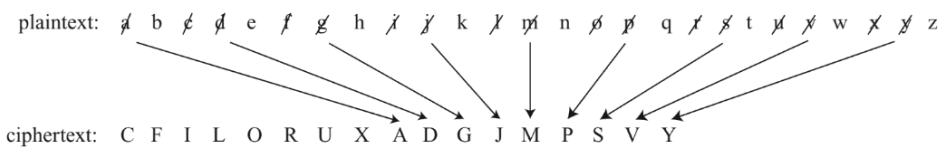
Let's look at a different type of cipher for some inspiration. This is called the **decimation method** of constructing a cipher. We need to pick a key, say 3. We start by writing out our plaintext alphabet.

plaintext: 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

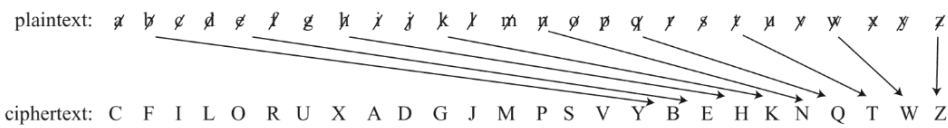
Then we count off every third letter, crossing each out (or "decimating" it) and writing each such letter as our ciphertext alphabet.



When you get to the end, "wrap around" to the beginning. In this case, cross out the "a" and keep going.



Finally, wrap around to the "b" and finish up:



So our final translation of plaintext to ciphertext is

plaintext:	a	b	c	d	e	f	g	h	i	j	k	l	m
ciphertext:	C	F	I	L	O	R	U	X	A	D	G	J	M
plaintext:	n	o	p	q	r	s	t	u	v	w	x	y	z
ciphertext:	P	S	V	Y	B	E	H	K	N	Q	T	W	Z

Okay, now let's try to look at this like a mathematician. How can we describe the decimation method in terms of modular arithmetic? Well, we should translate our letters into numbers, of course.

plaintext:	a	b	c	d	e	f	g	h	i	j	···	y	z
numbers:	1	2	3	4	5	6	7	8	9	10	···	25	26
some operation?:	3	6	9	12	15	18	21	24	1	4	···	23	26
ciphertext:	C	F	I	L	O	R	U	X	A	D	···	W	Z

Very interesting! For the first eight letters, all we have to do is multiply the number corresponding to the plaintext by 3 (the key) and we get the ciphertext. For the letter i this doesn't quite work, because 9 times 3 is 27—but 27 is the same as 1 modulo 26, which corresponds correctly to our ciphertext letter A.

Apparently there was nothing much special about the addition part of our additive cipher. Instead of adding 3 to each plaintext number, we can multiply by 3 instead, wrapping around when we get to 26. This makes sense from the "clock arithmetic" point of view also: Start at midnight. Three times 3 hours later is 9:00. Three times 4 hours later is 12:00. Three times 5 hours later is 3:00, and so on. Our new **multiplicative cipher** with key 3 looks like this:

plaintext	number	times 3	ciphertext
a	1	3	C
b	2	6	F
⋮	⋮	⋮	⋮
y	25	23	W
z	26	26	Z

If we want to encipher the message "be fruitful and multiply," it would look like this:

plaintext:	b	e	f	r	u	i	t	f	u	l	a	n	d
numbers:	2	5	6	18	21	9	20	6	21	12	1	14	4
times 3:	6	15	18	2	11	1	8	18	11	10	3	16	12
ciphertext:	F	O	R	B	K	A	H	R	K	J	C	P	L

plaintext:	m	u	l	t	i	p	l	y
numbers:	13	21	12	20	9	16	12	25
times 3:	13	11	10	8	1	22	10	23
ciphertext:	M	K	J	H	A	V	J	W

Incidentally, it's often useful to have a faster way of dealing with the wraparound than subtracting 26 over and over again. Luckily, you already know one—it's division with remainder, just like you learned in grade school. Only now, once we have seen how many 26s go into the number, we are going to throw all the 26s away and just keep the remainder. For example, to encipher the last letter of the preceding example, I multiplied 25 by 3 to get 75. Then I divided 75 by 26:

$$\begin{array}{r} 2 \\ 26\overline{)\ 75} \\ -52 \\ \hline 23 \end{array}$$

The quotient is 2, which I can throw away, and the remainder is 23, which is the number I need for my ciphertext. Another way of thinking about it is that the division with remainder shows that $75 = 2 \times 26 + 23$; that is, 75 is twice 26 with 23 left over. But 26 is the same as 0 modulo 26, so 75 is the same as $2 \times 0 + 23 = 23$ modulo 26.

How many keys does the multiplicative cipher have? At first glance, you might expect 26 again, including one stupid key. But hold on a moment—multiplying by 26 modulo 26 is the same as multiplying by 0. And multiplying by 0 is *bad*. Not just stupid, but bad. A multiplicative cipher with a key of 0 looks like this:

plaintext	number	times 0	ciphertext
a	1	0	Z
b	2	0	Z
⋮	⋮	⋮	⋮
y	25	0	Z
z	26	0	Z

So if we encrypt a message with this cipher, it comes out as

plaintext:	a	r	e	a	l	l	y	b	a	d	k	e	y
numbers:	1	18	5	1	12	12	25	2	1	4	11	5	25
times 0:	0	0	0	0	0	0	0	0	0	0	0	0	0
ciphertext:	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z

There's no way on earth to decrypt that! So we can't use that key.

Are there any other keys we can't use? Think about multiplying by 2—we know that any number multiplied by 2 is even. A multiplicative cipher with a key of 2 looks like this:

plaintext	number	times 2	ciphertext
a	1	2	B
b	2	4	D
⋮	⋮	⋮	⋮
l	12	24	X
m	13	26	Z
n	14	2	B
o	15	4	D
⋮	⋮	⋮	⋮
y	25	24	X
z	26	26	Z

That's better than multiplying by 0, but it still presents a problem when deciphering: a ciphertext B could be plaintext a or plaintext n; similarly, there are two plaintext letters for every other ciphertext letter. The same thing will happen with every other even key, so that makes 13 bad keys so far, and 13 left. There's one more bad key—take a moment to try and find it. So there are actually only 12 good keys for a multiplicative cipher, including multiplication by 1, the stupid key.

We've talked about enciphering a message with a multiplicative cipher but not really about deciphering it. Remember that to decipher a message, you need to do the opposite from enciphering it. To decrypt a Caesar cipher, you shift 3 letters left instead of shifting right. To decrypt a shift cipher, you shift k letters left. What about a multiplicative cipher? Well, you could just write out the whole table and use it backward, and in practice you probably would most of the time. But for very short messages, you might not want to write out the whole table. How can you reverse a multiplication?

The everyday answer is to divide. The opposite of multiplying by 3 is dividing by 3. That works fine for some of the letters in our multiplicative cipher with key 3. Ciphertext C becomes 3, which divided by 3 becomes 1, which is plaintext a. Ciphertext F becomes 6, which divided by 3 is 2, which is b. But what about A? It becomes 1, which divided by 3 is $\frac{1}{3}$, which isn't a letter. The solution is in the wraparound. The number 1 is the same as 27 modulo 26, so we could also say A becomes 27, which divided by 3 is 9, which is i. Likewise B could be not just 2 but also 28 and 54, and 54 divided by 3 is 18, so B corresponds to r.

ciphertext	number	divided by 3	plaintext
B	2	$\frac{2}{3}$	(not a letter)
B	28	$9\frac{1}{3}$	(not a letter)
B	54	18	r

This sort of trial and error works but is not much more efficient than writing out the table. For example, suppose your key is 15 instead of 3 for a moment. What plaintext letter does ciphertext B correspond to? Modulo 26, B could be any of the numbers 2, 28, 54, 80, 106, 132, 158, 184, 210, ...

ciphertext	number	divided by 15	plaintext
B	2	$\frac{2}{15}$	(not a letter)
B	28	$1\frac{13}{15}$	(not a letter)
B	54	$3\frac{9}{15}$	(not a letter)
B	80	$5\frac{5}{15}$	(not a letter)
B	106	$7\frac{1}{15}$	(not a letter)
B	132	$8\frac{12}{15}$	(not a letter)
B	158	$10\frac{8}{15}$	(not a letter)
B	184	$12\frac{4}{15}$	(not a letter)
B	210	14	n

It takes 9 tries before you find a value that's divisible by 15, and there's nothing to assure you that it won't be even worse for other letters. What would be really useful is a whole number that works modulo 26 like $\frac{1}{3}$ does for ordinary numbers. We could call this number $\overline{3}$. Then multiplying by $\overline{3}$ modulo 26 would be the same as multiplying by $\frac{1}{3}$ modulo 26, which is the same as dividing by 3 modulo 26.

Why might we think that $\overline{3}$ exists? If we look back at our example multiplicative cipher with key 3 from earlier, its deciphering table would look like this:

ciphertext	number	divided by 3 modulo 26	plaintext
A	1	9	i
B	2	18	r
C	3	1	a
D	4	10	j
⋮	⋮	⋮	⋮
Y	25	17	q
Z	26	26	z

It appears that perhaps dividing by 3 modulo 26 is the same as multiplying by 9 modulo 26. If this is true, then to decipher another letter, say E, we could calculate as follows:

ciphertext	number	times $\overline{3}$ = times 9	plaintext
E	5	19	s

Once I know what $\overline{3}$ is, then I can calculate this without using trial and error or searching through the encryption table.

If k is the key to a multiplicative cipher, can we be sure $\overline{k}$ exists? If so, how do we find it? Answering these questions will take us on a little detour, which, strangely enough, starts back at our "bad keys" for our multiplicative cipher.

We discovered that these bad keys are 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, and one more, which I will now reveal is 13. (You should check that this is, in fact, bad.) What these numbers have in common is that they are all multiples of 2, 13, or both. And $2 \times 13 = 26$, which is not a coincidence. If we were working with Julius Caesar's 21-letter alphabet (i.e., modulo 21), then the bad keys would be multiples of 3 or 7 (or both), since $21 = 3 \times 7$. Romanian has 28 letters and $28 = 2 \times 2 \times 7$, so the bad keys would be multiples of 2 or 7 (or both). In Danish, Norwegian, and Swedish, which have 29 letters, 29 would be the only bad key.

What we have done with these letters (26, 21, 28, 29) is to break them up into their smallest irreducible components, the **prime numbers**. This process, which is called **factoring**, can always be done in one and only one way. This was known at least as long ago as the fourth century BCE, when **Euclid** put it in his *Elements*. What we want to know is whether our key and our modulus have a **common divisor**, that is, a number that divides them both. The number 1 always divides both numbers, but that's considered trivial and doesn't count for this purpose. Euclid's *Elements* also tells us how to find a common divisor very efficiently by finding the **greatest common divisor**, or GCD, which is just what it sounds like. The method for calculating the GCD is known as the **Euclidean algorithm**, although we don't really know whether Euclid invented it or borrowed it from someone else. An **algorithm** is a well-defined method for doing something which always produces a specific correct answer for each input, such as a computer program.

Here's an example of the Euclidean algorithm in action, calculating the GCD of 756 and 210.

ciphertext	number	times 9	plaintext
A	1	9	i
⋮	⋮	⋮	⋮
E	5	19	s
⋮	⋮	⋮	⋮

Incidentally, the technical term for $\overline{3}$ is the **multiplicative inverse** of 3 modulo 26. The general idea of **inverses** is terribly important in many branches of mathematics. We've now seen **additive inverses**—that is, negatives—and multiplicative inverses, and we will see other examples in the future. A good thing to notice about inverses in modular arithmetic is that, unlike in ordinary arithmetic, there isn't usually any qualitative difference between a number and its inverse. That is, in ordinary arithmetic, 2 is a positive number and -2 is a negative number, but modulo 26, $-2 \equiv 24$. So 2 and 24 are arithmetic inverses, but neither is particularly "negative." Likewise, in ordinary arithmetic, 3 is a whole number and $\frac{1}{3}$ is a fraction, but modulo 26, 3 and 9 are multiplicative inverses, despite neither being "fractional." This is characteristic of situations where there are only finitely many numbers that are considered distinct. Another way of looking at it is that there is no real distinction between forward and backward in these situations. Likewise, there is no mathematical difference between an arbitrary encryption and an arbitrary decryption for ciphers that use these operations—once you have figured out the inverse, you can "go forward to go backward." This will be sufficiently important in later sections that you might want to think about it a bit before going on.

1.4 AFFINE CIPHERS

Now we have a shift cipher with 26 good keys, 1 of which is stupid, and a multiplicative cipher with 12 good keys, 1 of which is stupid. Both of these are pretty easy for Eve to attack with a **brute-force attack**, meaning that she just tries every possible key until she gets the right one. Even if Alice and Bob can choose either type of cipher, that still leaves Eve only 38 choices to try. But what if Alice and Bob could use more than one cipher at the same time?

This has the potential to get complicated enough so that we'll introduce a little more mathematical notation. We'll use P to stand for any number between 1 and 26 that represents a plaintext letter and C to stand for a number that represents a ciphertext letter. We'll still use k to stand for a key. Encrypting using a shift cipher with a key of k can be written as

$$C \equiv P + k \quad \text{modulo } 26,$$

and using a multiplicative cipher with a key of k can be written as

$$C \equiv kP \quad \text{modulo } 26.$$

Similarly, decrypting in the shift cipher case looks like

$$P \equiv C - k \quad \text{modulo } 26,$$

and, in the multiplicative cipher case, looks like

$$P \equiv \overline{k}C \quad \text{modulo } 26.$$

What if Alice tries to encrypt using two different shift cipher keys, say k and m?* Is that twice as secure? It would look like

$$C \equiv P + k + m \quad \text{modulo } 26.$$

Unfortunately for Alice and Bob, from Eve's point of view this looks exactly the same as encrypting once using the key $k + m$, so Eve will break the cipher just as easily if she tries a brute-force attack. The same thing will happen if Alice uses two different multiplicative cipher keys. But what if she uses one of each? Suppose Alice first multiplies the plaintext by k and then adds m to get the ciphertext:

$$C \equiv kP + m \quad \text{modulo } 26.$$

Bob will decrypt by first subtracting m and then multiplying by $\overline{k}$:

$$P \equiv \overline{k}(C - m) \quad \text{modulo } 26.$$

Notice that Bob has to not only reverse the operations, but also reverse their order! If this seems unintuitive, think about getting dressed and undressed. To get dressed, you have to put on your socks first, and then

*Cryptographers sometimes use m to stand for a second cipher key because it comes after k and the letter l looks too much like the number 1.

your shoes. To get undressed, you have to remove them both, but in the opposite order. Otherwise bad things happen.

This combination gives us a new kind of cipher, which is technically called an **affine cipher**, although I sometimes prefer to just call it a $kP + m$ cipher. There are 12 choices for k and 26 choices for m, so there are $12 \times 26 = 312$ different keys for this cipher. This is getting to be enough to make Eve's brute-force attack a little difficult, although it is still not very hard if she has access to a computer.

The idea of combining two ciphers to make a **product cipher** is a fairly obvious one and goes back quite a long time in history. The idea that one can combine any decimation method (i.e., multiplicative cipher; see Section 1.3) with any shift cipher (i.e., additive cipher, see Section 1.2) goes back at least as far as the 1930s. It's worth mentioning one much older cipher that is a particular form of a $kP + m$ cipher. This is called the **atbash** cipher, and it's at least as old as the Biblical Book of Jeremiah. Like the decimation method, it starts by writing out the plaintext alphabet. Below it, the ciphertext alphabet is the same alphabet written backward. We'll use the modern English alphabet instead of the Hebrew alphabet:

plaintext:	a	b	c	d	e	f	g	h	i	j	k	l	m
ciphertext:	Z	Y	X	W	V	U	T	S	R	Q	P	O	N
plaintext:	n	o	p	q	r	s	t	u	v	w	x	y	z
ciphertext:	M	L	K	J	I	H	G	F	E	D	C	B	A

So why is this a form of a $kP + m$ cipher? When we translate the numbers into letters, we get

plaintext:	a	b	c	d	e	f	g	h	i	j	⋯	y	z
numbers:	1	2	3	4	5	6	7	8	9	10	⋯	25	26
some operation?:	26	25	24	23	22	21	20	19	18	17	⋯	2	1
ciphertext:	Z	Y	X	W	V	U	T	S	R	Q	⋯	B	A

We see that the ciphertext obeys the rule

$$C \equiv 27 - P \mod 26.$$

Of course we can also write that as

$$C \equiv (-1)P + 27 \quad \text{modulo } 26,$$

and modulo 26 that's the same as

$$C \equiv 25P + 1 \quad \text{modulo } 26.$$

So this is a $kP + m$ cipher with the key $k = 25$, $m = 1$.

1.5 ATTACK AT DAWN: CRYPTANALYSIS OF SIMPLE SUBSTITUTION CIPHERS

If we continue along this path of making our operations modulo 26 more and more complicated, we could eventually figure out a way to specify where every single plaintext letter goes individually. So a can go to any of the 26 ciphertext letters. Then we could send b to any ciphertext letter different from the ciphertext for a, so there are 25 choices. There are 24 ciphertext letters still unused for c, then 23 for d, and so on, until we have only one letter left for z. A cipher of this kind is called a **monoalphabetic monographic substitution cipher**, **monographic** meaning that it makes substitutions one letter at a time and **monoalphabetic** meaning that the substitution rule is the same for every letter in the message. That's a pretty unwieldy name and it's a pretty common cipher, so to save time I'm just going to call it a **simple substitution cipher**. All told there are $26 \times 25 \times 24 \times \cdots \times 3 \times 2 \times 1 =$ 403,291,461,126,605,635,584,000,000 ways to make this kind of cipher, which includes all three of the ciphers we have discussed as well as the cryptogram puzzles that one finds in many daily papers. That's way too many keys to attack by brute force. Unfortunately for Alice and Bob, Eve has a much better attack available to her.

A very effective way of breaking simple substitution ciphers is called **letter frequency analysis**. This technique goes back at least as far as the ninth-century Arab scholar **Abu Yusuf Yaqub ibn Ishaq al-Sabbah al-Kindi**. The idea is simply that some letters in English, Arabic, or any other human language are used more often than others. For example, in a typical English text, the letter e will occur about 13% of the time, far more than any other. If Eve has a piece of ciphertext where a

letter, say R, occurs about 13% of the time and more often than any other letter, then there's a good chance that R ($C = 18$) represents e ($P = 5$). If the cipher is an additive cipher, then Eve knows that

$$5 + k \equiv 18 \quad \text{modulo } 26,$$

so there is a very good chance that the key is $k = 13$.

If Eve has another type of cipher, such as an affine cipher, this might not be enough information. In this case, she might need to guess another letter, such as t, which occurs about 8% of the time, or a, which occurs about 7% of the time. For example, if Eve guesses that R represents e and F represents a, then she knows that

$$5k + m \equiv 18 \quad \text{modulo } 26,$$
$$1k + m \equiv 6 \quad \text{modulo } 26.$$

Now Eve has two equations in two unknowns. Subtracting them gives

$$4k \equiv 12 \quad \text{modulo } 26.$$

If the number 4 had an inverse modulo 26, then Eve could multiply each side by that inverse to cancel out the 4 and find k. Unfortunately, the GCD of 4 and 26 is 2, so 4 doesn't have an inverse. This means that our equation has either no solutions or more than one solution. If there are no solutions, it means in this case that Eve probably made a bad guess from the letter frequencies and she should try again. But in this case it turns out that there are two solutions, $k = 3$ and $k = 16$, and in either case m must be $6 - 1k$ modulo 26. So the possibilities are $k = 3$ and $m = 3$ or $k = 16$ and $m = 16$. Eve can then try to decrypt using each combination and see if she gets readable text. Since a, t, and several other letters have similar frequencies, it's possible that neither one is correct, in which case Eve has to go back to the beginning and try to guess e and a again. It might take a few guesses, but in the end Eve should be able to determine the correct key a lot faster than using brute force.

The one big caveat to this technique is that you need to have enough ciphertext to work with. The frequencies I have mentioned are only averages, and short messages may very well have radically different letter frequencies. Just imagine trying to decrypt the message "Zola is

Similarly, Bob can multiply the top equation by k_3 and the bottom equation by k_1, which gives him

$$6 \times 5 \equiv (6 \times 3)P_1 + (6 \times 5)P_2 \quad \text{modulo } 26,$$
$$3 \times 2 \equiv (3 \times 6)P_1 + (3 \times 1)P_2 \quad \text{modulo } 26.$$

This time he takes the bottom minus the top to get

$$3 \times 2 - 6 \times 5 \equiv (3 \times 1 - 6 \times 5)P_2 \quad \text{modulo } 26.$$

Notice that in both cases there is a -27 on the right-hand side, which is $k_1k_4 - k_2k_3$. This number is called the **determinant** of the system. If the greatest common divisor of the determinant and 26 is 1, then the determinant has a multiplicative inverse, and Bob can multiply each side of his equations by that inverse to find P_1 and P_2. This is very similar to the case of ordinary arithmetic, where two equations in two unknowns can always be solved as long as the determinant of the system is not equal to zero.

In our example, the determinant is -27, as we said, which is the same as 25 modulo 26. If Bob runs through the Euclidean algorithm, he will find that

$$\overline{25} \equiv 25 \quad \text{modulo } 26,$$

so he gets

$$P_1 \equiv ((1 \times 5) - (5 \times 2)) \times 25 \quad \text{modulo } 26,$$
$$P_2 \equiv ((3 \times 2) - (6 \times 5)) \times 25 \quad \text{modulo } 26,$$

which finally reduces to

$$P_1 \equiv 5 \quad \text{modulo } 26, \qquad P_2 \equiv 24 \quad \text{modulo } 26,$$

or ex.

In general, if $k_1k_4 - k_2k_3$ has an inverse, then the solution to

$$C_1 \equiv k_1P_1 + k_2P_2 \quad \text{modulo } 26,$$
$$C_2 \equiv k_3P_1 + k_4P_2 \quad \text{modulo } 26$$

is

$$P_1 \equiv \overline{(k_1k_4 - k_2k_3)}(k_4C_1 - k_2C_2) \quad \text{modulo } 26,$$
$$P_2 \equiv \overline{(k_1k_4 - k_2k_3)}(-k_3C_1 + k_1C_2) \quad \text{modulo } 26.$$

The general form of this method for solving a system of several equations in the same number of unknowns is usually known as **Cramer's rule**, named for **Gabriel Cramer**. Cramer was an eighteenth-century Swiss mathematician who did much work studying systems of equations and the curves they describe. The same rule seems to have been published slightly earlier by **Colin Maclaurin** in Scotland. Cramer's rule is not the fastest way of solving large systems of equations, but it's certainly good enough for the block sizes one is likely to use in a Hill cipher.

Notice that if we give new names to the numbers

$$m_1 = \overline{(k_1k_4 - k_2k_3)}(k_4),$$
$$m_2 = \overline{(k_1k_4 - k_2k_3)}(-k_2),$$
$$m_3 = \overline{(k_1k_4 - k_2k_3)}(-k_3), \qquad \text{and}$$
$$m_4 = \overline{(k_1k_4 - k_2k_3)}(k_1),$$

then we can write

$$P_1 \equiv m_1C_1 + m_2C_2 \quad \text{modulo } 26,$$
$$P_2 \equiv m_3C_1 + m_4C_2 \quad \text{modulo } 26.$$

We can think of this system of equations as an inverse of the original system, and we can think of m_1, m_2, m_3, m_4 as a sort of "inverse key" for the original encryption key k_1, k_2, k_3, k_4. In our example this key would be 25×1, 25×-5, 25×-6, 25×3, or 25, 5, 6, 23 modulo 26. Once Bob has worked this out, the process of decryption works exactly the same as encryption. This is another example of the idea of going forward to go backward that we talked about in Section 1.3.

It's a little involved to work out exactly how many good keys (i.e., keys where the determinant has an inverse) there are for a Hill cipher, but it's about 160,000 for a block size of 2 and about 1,600,000,000,000 for a block size of 3, so a brute-force attack is getting to be rather difficult. Also note that Bob needs to be aware that there may be nulls at the end of his message. This ought to be clear when he reads it.

In 1931, Hill followed up his original cipher with several extensions. The most important one is now generally known as the **affine Hill cipher**, because it combines the original Hill cipher with an addition step, just like we combined the multiplicative and additive ciphers to get the affine cipher. If we let the block size be 2 again, the new formulas are

$$C_1 \equiv k_1P_1 + k_2P_2 + m_1 \quad \text{modulo } 26,$$
$$C_2 \equiv k_3P_1 + k_4P_2 + m_2 \quad \text{modulo } 26,$$

where the key now consists of six numbers, k_1, k_2, k_3, k_4, m_1, and m_2, all between 1 and 26. Once again, this is a good key as long as the greatest common divisor of the determinant $k_1k_4 - k_2k_3$ and 26 is 1. (The new key numbers m_1 and m_2 can be anything.) To decrypt, Bob just needs to subtract m_1 from C_1 and m_2 from C_2 and then solve the system as before.

A letter frequency analysis no longer works on a polygraphic cipher, because, as you can see from the example, the same letter in the plaintext doesn't always go to the same letter in the ciphertext. Therefore, the whole idea of guessing which letter is e fails. On the other hand, we also saw that the same plaintext block always goes to the same ciphertext block, and in the case of block size 2 or 3, it is possible to exploit this. For example, the most common digraph, or 2-letter block, is "th," which occurs, according to one study, approximately 2.5% of the time. The most common trigraph (3-letter block) is "the," which occurs, by the same study, just under 1% of the time. Eve could use facts like these to do a digraph or trigraph frequency analysis and perhaps break a digraphic or trigraphic substitution cipher. However, for larger block sizes this quickly gets very difficult, as there are a lot of possible blocks and not a lot of difference between the frequencies of the various blocks. Even in 1929, Hill designed a machine that used a set of gears to mechanically encipher texts using block size 6 and was thus essentially unbreakable using frequency analysis. Unfortunately for Hill, his machine never caught on.

The Hill ciphers were never used much—they were too unwieldy to use by hand, and cryptography via mechanical devices went in the direction of polyalphabetic substitution ciphers instead. Hill's idea of using systems of equations has regained substantial importance with

the advent of digital computers in cryptography, but from a modern point of view, these ciphers used by themselves have the problem that they are badly vulnerable to a type of attack that is rather different from the ones we have talked about so far.

1.7 KNOWN-PLAINTEXT ATTACKS

So far, all of the cryptanalytic attacks we have discussed are **ciphertext-only attacks**, where all that Eve knows is the ciphertext message she has intercepted passing between Alice and Bob. But suppose that somehow Eve has gotten hold of both the plaintext and ciphertext of some message (or part of a message) that Alice has sent. Then she can try a **known-plaintext attack**, where she knows both the plaintext and the ciphertext and the goal is to get the key. Once she has the key, she can find out the content of not just the message she has, but other messages or parts of messages sent with the same key.

In the case of block size 2 and the original Hill cipher, suppose Eve recovers four letters of plaintext, P_1, P_2, P_3, and P_4, and the matching letters of ciphertext, C_1, C_2, C_3, and C_4. Then she knows

$$\begin{aligned} C_1 &\equiv k_1P_1 + k_2P_2 \quad \text{modulo } 26,\\ C_2 &\equiv k_3P_1 + k_4P_2 \quad \text{modulo } 26,\\ C_3 &\equiv k_1P_3 + k_2P_4 \quad \text{modulo } 26,\\ C_4 &\equiv k_3P_3 + k_4P_4 \quad \text{modulo } 26. \end{aligned}$$

From Eve's point of view, only the key numbers are unknowns, so she has four equations in four unknowns, and she can solve the system to recover the key.

In the example from earlier, if Eve managed to recover the first and last blocks of plaintext, she will know:

$$\begin{aligned} 9 &\equiv k_1 10 + k_2 1 \quad \text{modulo } 26,\\ 9 &\equiv k_3 10 + k_4 1 \quad \text{modulo } 26,\\ 5 &\equiv k_1 5 + k_2 24 \quad \text{modulo } 26,\\ 2 &\equiv k_3 5 + k_4 24 \quad \text{modulo } 26. \end{aligned}$$

This is really two sets of equations,

$$9 \equiv k_1 10 + k_2 1 \quad \text{modulo } 26,$$
$$5 \equiv k_1 5 + k_2 24 \quad \text{modulo } 26$$

and

$$9 \equiv k_3 10 + k_4 1 \quad \text{modulo } 26,$$
$$2 \equiv k_3 5 + k_4 24 \quad \text{modulo } 26.$$

Eve could solve each set with Cramer's Rule in the same way that Bob solved his equations in the previous section. For the first set, the rule gives

$$k_1 \equiv \overline{(10 \times 24 - 1 \times 5)}(24 \times 9 - 22 \times 5) \quad \text{modulo } 26,$$
$$k_2 \equiv \overline{(10 \times 24 - 1 \times 5)}(-5 \times 9 + 10 \times 5) \quad \text{modulo } 26.$$

If you finish the calculations, you will see

$$k_1 \equiv 3 \quad \text{modulo } 26, \qquad k_2 \equiv 5 \quad \text{modulo } 26.$$

Similarly, the second set gives Eve

$$k_3 \equiv \overline{(10 \times 24 - 1 \times 5)}(24 \times 9 - 1 \times 2) \quad \text{modulo } 26,$$
$$k_4 \equiv \overline{(10 \times 24 - 1 \times 5)}(-5 \times 9 + 10 \times 2) \quad \text{modulo } 26.$$

which gives her the last two key numbers:

$$k_3 \equiv 6 \quad \text{modulo } 26, \qquad k_4 \equiv 1 \quad \text{modulo } 26.$$

In general, Eve will need to recover only as many blocks of plaintext as there are letters in a block. So it's almost as easy to break the Hill cipher using a known-plaintext attack as it is to decipher a message. This is considered unacceptable, so the Hill cipher is never used in its original form. The idea of using a system of equations for polygraphic encryption, however, forms a piece of many modern ciphers.

1.8 LOOKING FORWARD

I warned you in the preface to this book that some of the ciphers I discuss in this book are considered obsolete in today's world, and that includes all the ciphers in this chapter and the next two, more or less. For one

2

Polyalphabetic Substitution Ciphers

2.1 HOMOPHONIC CIPHERS

Polygraphic ciphers, which work on more than one letter at a time, are one way to make ciphers that resist a straightforward letter frequency analysis. As we have seen, they can be difficult or impossible to do by hand, even with 3-letter blocks, and somewhat cumbersome even with machines. A polyalphabetic cipher, on the other hand, still works on 1 letter at a time like a monoalphabetic cipher, but it changes the substitution rule from letter to letter. This can be as simple as giving Alice, the encipherer, more than one ciphertext option for some or all plaintext letters, which she can choose from at whim. This is called a **homophonic** cipher—in linguistics, homophones are 2 letters or groups of letters that are spelled differently but pronounced the same. In cryptography, homophones are letters or groups of letters that are written differently in the ciphertext but deciphered the same.

As with many other aspects of cryptography, the ideas behind homophonic ciphers seem to have been first explored by the Arabs. The first known cipher that explicitly uses homophones as a central technique, however, appeared in Italy, having been prepared in 1401 by a cipher secretary of the Duchy of Mantua. This cipher appears to simply be a version of the atbash cipher, with the addition of 12 extra symbols: 3 each for the letters a, e, o, and u, which were high-frequency letters in fifteenth-century Italian. A representation of this idea with modern English letters and typographical symbols might look like this:

plaintext:	a	b	c	d	e	f	g	h	i	j	k	l	m
ciphertext:	Z	Y	X	W	V	U	T	S	R	Q	P	O	N
	!				@								
	%				&								
	)				-								
plaintext:	n	o	p	q	r	s	t	u	v	w	x	y	z
ciphertext:	M	L	K	J	I	H	G	F	E	D	C	B	A
		#						$					
		*						(					
		=						+					

One suspects that this didn't improve the security of such a simple cipher by very much, but the idea is sound: if the ciphertext letters corresponding to high-frequency plaintext letters are randomly divided up between multiple options, a straightforward letter frequency analysis becomes rather difficult. When used properly, the cipher shown here will produce a ciphertext in which no letter comes even near to the 13% frequency that one expects for the ciphertext letter corresponding to e. Instead, there will be four different symbols (V, @, &, and -), which each occur with just over 3% frequency. Lots of other letters also occur with 4% frequency, so this doesn't help the cryptanalyst much. This works only if Alice really picks one of the four symbols at random. A common pitfall is for a sloppy encipherer to primarily use only one of the options (say V, which might be more convenient on a keyboard) and only occasionally use the others—this will pretty much destroy the usefulness of the homophones.

It is not clear how much was known in Europe at this point in time about letter frequency analysis. The fact that the Mantuan cipher gives homophones only for vowels, which are high frequency, leads one to suspect that they knew something about the subject. We don't know for sure because unlike in the Arab world, where cryptography was mostly an academic pursuit, in Renaissance Europe it was a deadly serious part of diplomacy and its secrets were kept well guarded. It would not be until 1466 or 1467 that a description of frequency analysis would appear in print in Europe, by Leon Battista Alberti, whom we shall meet shortly. And due, perhaps, to the stereotypical conservatism of diplomats, the

first ciphers with homophones for consonants as well as vowels did not appear until the mid-1500s.

2.2 COINCIDENCE OR CONSPIRACY?

So far we have been assuming Kerckhoffs' principle without too much reflection when we take the role of Eve. Often, however, Eve doesn't even need to steal the system in order to make some good guesses about how it works. For instance, how might Eve guess that a homophonic system is being used? True, it will generally have more than 26 characters. But perhaps the message is in a language other than English, or perhaps not all the possible ciphertext letters actually appear in the message. Can we tell what is going on?

Making a table of the frequency of each letter in the ciphertext is a good first step. Suppose Eve has intercepted this ciphertext:

QBVDL	WXTEQ	GXOKT	NGZJQ	GKXST	RQLYR
XJYGJ	NALRX	OTQLS	LRKJQ	FJYGJ	NGXLK
QLYUZ	GJSXQ	GXSLQ	XNQXL	VXKOJ	DVJNN
BTKJZ	BKPXU	LYUNZ	XLQXU	JYQGX	NTYQG
XKXQJ	KXULK	QJNQN	LQBYL	OLKKX	SJYQG
XNGLU	XRSBN	XOFUL	YDSXU	GJNSX	DNVTY
RGXUG	JNLEE	SXLYU	ESLYY	XUQGX	NSLTD
GQXKB	AVBKX	JYYBR	XYQNQ	GXKXZ	LNYBS
LRPBA	VLQXK	JLSOB	FNGLE	EXYXU	LSBYD
XWXKF	SJQQS	XZGJS	XQGXF	RLVXQ	BMXXK
OTQKX	VLJYX	UQBZG	JQXZL	NG	

Alice has removed the spaces from her plaintext and divided it up into 5-letter groups in order to make things harder for Eve by obscuring any short, common words. Eve starts by counting how often each letter appears and what percentage each letter takes up of the 322 letters total (see Table 2.1).

There are only 23 distinct characters in the ciphertext, which could mean that Eve is dealing with a language with less than 26 letters, or that Alice used some sort of polygraphic system which doesn't need all of the characters, or just that some letters in the plaintext don't appear.

TABLE 2.1.
Letter frequencies observed in our ciphertext

Letter	Number of Occurrences	Percent Frequency
A	3	.9
B	14	4.3
D	6	1.9
E	6	1.9
F	5	1.6
G	23	7.1
J	22	6.8
K	19	5.9
L	30	9.3
M	1	.3
N	20	6.2
O	7	2.2
P	2	.6
Q	30	9.3
R	9	2.8
S	17	5.3
T	9	2.8
U	13	4.0
V	8	2.5
W	2	.6
X	47	14.6
Y	21	6.5
Z	8	2.5

How does Eve's table compare with the expected frequencies in English text? See Table 2.2.

It seems reasonably plausible that what we have is a simple substitution cipher that just doesn't happen to have some of the lowest-frequency letters in its plaintext. If homophones were being used, we would expect to see more low-frequency letters and fewer (if any) high-frequency ones. It would be nice if we could make this observation more quantitative, though.

The tool for that is called the **index of coincidence**, and it was invented by **William Friedman**, easily one of the most important figures in early twentieth-century cryptology. Friedman never set out to be a cryptologist. He studied genetics in college and graduate school and

Table 2.2.
Letter frequencies in English text compared with our ciphertext

Letter	Percent Frequency in English Text	Letter	Percent Frequency in Our Ciphertext
e	12.7	X	14.6
t	9.1	L	9.3
a	8.2	Q	9.3
o	7.5	G	7.1
i	7.0	J	6.8
n	6.7	Y	6.5
s	6.3	N	6.2
h	6.1	K	5.9
r	6.0	S	5.3
d	4.3	B	4.3
l	4.0	U	4.0
c	2.8	R	2.8
u	2.8	T	2.8
m	2.4	V	2.5
w	2.4	Z	2.5
f	2.2	O	2.2
g	2.0	D	1.9
y	2.0	E	1.9
p	1.9	F	1.6
b	1.5	A	.9
v	1.0	P	.6
k	.8	W	.6
j	.2	M	.3
x	.2		
q	.1		
z	.1		

was invited to join the Department of Genetics at the Riverbank Laboratories, an organization founded and run by an eccentric Illinois millionaire. Friedman got involved in cryptology when he was asked to help with photography for a group attempting to find hidden ciphers in the works of Shakespeare. Although he eventually concluded that no such ciphers were present, he found both his future wife and his future career in the Riverbank cryptology group. Friedman left Riverbank to join the US Army during World War I and eventually moved to the National Security Agency when it was formed after World War II. His wife, Elizebeth, had her own distinguished career in the meantime,