# 6

# Ciphers Involving Exponentiation

## 6.1 ENCRYPTING USING EXPONENTIATION

We'd like our next cipher to be a simple mathematical cipher resistant to both ciphertext-only and known-plaintext attacks, as explained in Section 1.7. For the first, we'll make it a polygraphic cipher, although the way we construct the blocks is just a little bit different from what we did in Section 1.6. Once again, we'll take a block size of 2 in our example and divide up the plaintext into 2-letter blocks.

po we rt ot he pe op le

This time, we'll convert each 2-letter block into a number by just jamming the numbers from the 2 letters together, putting in 0s where appropriate.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| plaintext: | po | we | rt | ot | he | pe | op | le |
| numbers: | 16, 15 | 23, 5 | 18, 20 | 15, 20 | 8, 5 | 16, 5 | 15, 16 | 12, 5 |
| "jammed together": | 1615 | 2305 | 1820 | 1520 | 805 | 1605 | 1516 | 1205 |

We will also need to pick a modulus for the cipher. A modulus of 26 is no longer going to do it, since our blocks can be as large as 2626. It will be convenient to pick a modulus that is a prime number, although later in this chapter, we will see that we can get around that. For the moment, 2819 will be a good choice, since it is prime and larger than 2626.

We've tried addition, multiplication, and various combinations of them. A mathematician's next idea might be to try exponentiation, or raising a number to a power. Remember that raising a number to a power means multiplying it by itself repeatedly. For example, $2^3 = 2 \times 2 \times 2 = 8$. In particular, we will use

$$C \equiv P^e \quad \text{modulo } 2819.$$

The key for this cipher is traditionally called $e$, for **encryption exponent**. Note that $e$ has nothing to do here with the number 2.71828..., which is the base of the natural logarithm. The encryption exponent is a number between 1 and 2818, with some restrictions, which we will explore in more detail shortly. For the moment, let's take $e = 769$.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| plaintext: | po | we | rt | ot | he | pe | op | le |
| numbers: | 16, 15 | 23, 5 | 18, 20 | 15, 20 | 8, 5 | 16, 5 | 15, 16 | 12, 5 |
| together: | 1615 | 2305 | 1820 | 1520 | 805 | 1605 | 1516 | 1205 |
| to the 769th power: | 1592 | 783 | 2264 | 924 | 211 | 44 | 1220 | 1548 |

What we are doing here is raising 1615 to the 769th power, wrapping around every time we get to 2819, which means really a lot of multiplications and wraparounds. You need a computer, or at least a very good calculator, to have any hope of doing this. We can't change all these blocks back into letters, but that's okay. Alice can just send Bob the numbers.

How is Bob going to decrypt this? Just as the opposite of addition is subtraction and the opposite of multiplication is division, the opposite of taking a power is taking a root. For example, if $8 = 2^3$, then $2 = \sqrt[3]{8}$, and if $C = P^e$, then $P = \sqrt[e]{C}$. But if you thought doing division and making sure you get a whole number was problematic, taking roots is even worse. For instance, in our example the first ciphertext block was 1592, and the 769th root of 1592 is approximately 1.0096, which is pretty useless for our purposes.

## 6.2 Fermat's Little Theorem

In order to help Bob, we're going to have to go a little bit deeper into number theory than we have so far. Up until now, we've basically been using one big mathematical idea, namely, modular arithmetic, as formalized by Gauss. Now we need a second big idea, which is generally credited to **Pierre de Fermat**. Fermat was a seventeenth-century Frenchman who was a lawyer by profession and a mathematician by avocation. Possibly because of this, he had a bit of a mathematical chip on his shoulder. He had a habit of writing letters to his colleagues in

which he announced that he had proven something. Instead of giving the proof, he challenged the recipient to come up with the proof himself. He also claimed he had proved some things that turned out to be false, and at least one, now known as Fermat's last theorem, that turned out to be true but probably a lot harder to prove than Fermat thought.

The mathematical fact, or **theorem**, that we need here is definitely true, and Fermat may very well have come up with a proof, although as usual he didn't write it down. It's now called **Fermat's little theorem**, even though it has big implications. We don't know how Fermat discovered it, but here's how you might have discovered it using the ideas we've already explored.

Suppose you are working with a multiplicative cipher with a very small alphabet that has a prime number of letters. The 13-letter Hawaiian alphabet would work. With a key of 3, the table for this alphabet looks like this:

| plaintext | number | times 3 | ciphertext |
|---|---|---|---|
| a | 1 | 3 | I |
| e | 2 | 6 | H |
| i | 3 | 9 | M |
| o | 4 | 12 | W |
| u | 5 | 2 | E |
| h | 6 | 5 | U |
| k | 7 | 8 | L |
| l | 8 | 11 | P |
| m | 9 | 1 | A |
| n | 10 | 4 | O |
| p | 11 | 7 | K |
| w | 12 | 10 | N |
| ' | 13 | 13 | ' |

The important thing here is that since 13 is prime, 3 is a good key, and so is every other number from 1 to 12. Thus the column of numbers on the left-hand side is the same as the column of the numbers on the right-hand side, except in a different order. If you were playing around with this, you might have tried adding each column. You would get the same answer modulo 13, since they are the same numbers modulo 13:

$$1 + 2 + 3 + \cdots + 13 \equiv (1 \times 3) + (2 \times 3) + (3 \times 3) + \cdots + (13 \times 3) \quad \text{modulo } 13.$$

Collect like terms on the right:

$$1 + 2 + 3 + \cdots + 13 \equiv (1 + 2 + 3 + \cdots + 13) \times 3 \quad \text{modulo } 13,$$

or

$$91 \equiv 91 \times 3 \quad \text{modulo } 13,$$

or

$$\equiv 0 \times 3 \quad \text{modulo } 13.$$

That wasn't that interesting. Instead of adding up each column, you could try multiplying it instead. Then you would get

$$1 \times 2 \times 3 \times \cdots \times 13 \equiv (1 \times 3) \times (2 \times 3) \times (3 \times 3) \times \cdots \times (13 \times 3) \quad \text{modulo } 13,$$

$$1 \times 2 \times 3 \times \cdots \times 0 \equiv (1 \times 3) \times (2 \times 3) \times (3 \times 3) \times \cdots \times (0 \times 3) \quad \text{modulo } 13,$$

$$0 \equiv 0 \quad \text{modulo } 13.$$

That's even less interesting, but clearly the problem is the 13 at the end of each column. You could try just leaving that out.

$$1 \times 2 \times 3 \times \cdots \times 12 \equiv (1 \times 3) \times (2 \times 3) \times (3 \times 3) \times \cdots \times (12 \times 3) \quad \text{modulo } 13.$$

Now you could pull out all the 3s on the right, which came from the key.

$$1 \times 2 \times 3 \times \cdots \times 12 \equiv (1 \times 2 \times 3 \times \cdots \times 12) \times 3^{12} \quad \text{modulo } 13.$$

Cancel $1 \times 2 \times 3 \times \cdots \times 12$:

$$1 \equiv 3^{12} \quad \text{modulo } 13.$$

And that, I hope you agree, is interesting.

Notice that the choices of 13 and 3 weren't important. Any prime modulus $p$ and any good key number $k$ will do. So Fermat's little theorem tells us the following.

**Theorem** (Fermat's Little Theorem) *For any prime $p$ and any $k$ between 1 and $p-1$,*

$$k^{p-1} \equiv 1 \quad \text{modulo } p.$$

## 6.3 DECRYPTING USING EXPONENTIATION

Now would probably be a good time to drop back and try to remember our goal. We wanted to undo the equation

$$C \equiv P^e \quad \text{modulo } 2819.$$

Remember from Section 1.3 that in modular situations we should be able to go forward to go backward. So it should be reasonable to look for a number $\overline{e}$ such that

$$C^{\overline{e}} \equiv P \quad \text{modulo } 2819.$$

Since $C \equiv P^e$ modulo 2819, this is the same as saying

$$(P^e)^{\overline{e}} \equiv P \quad \text{modulo } 2819,$$

or, using the laws of exponents,

$$P^{e\overline{e}} \equiv P \quad \text{modulo } 2819.$$

If we look at Fermat's little theorem closely here, we see that it says

$$P^{2818} \equiv 1 \quad \text{modulo } 2819,$$

but we could also write it as

$$P^{2818} \equiv P^0 \quad \text{modulo } 2819.$$

We are working modulo 2819, which means 2819 is the same as 0 if we are looking at the whole equation. But if we are looking at the *exponent*, then Fermat's little theorem says 2818 is the same as 0. In general, if we are looking at an equation modulo a prime $p$, then we can treat the exponent as if we were working modulo $p-1$. Therefore, the number $\overline{e}$ that we are looking for should be the inverse of $e$ modulo 2818. For

future reference, it is important to note that exponents work quite this way only for primes. We will see the equivalent for other numbers in Section 6.6.

So we'll use the Euclidean algorithm on $e$ (which was 769) and 2818 like we did in Section 1.3. I'll put in a little less detail than I did there, but feel free to fill in the gaps.

$$\begin{aligned}
2818 &= 769 \times 3 + 511 & 511 &= 2818 - (769 \times 3)\\
769 &= 511 \times 1 + 258 & 258 &= 769 - (511 \times 1)\\
& & &= (769 \times 4) - (2818 \times 1)\\
511 &= 258 \times 1 + 253 & 253 &= 511 - (258 \times 1)\\
& & &= (2818 \times 2) - (769 \times 7)\\
258 &= 253 \times 1 + 5 & 5 &= 258 - (253 \times 1)\\
& & &= (769 \times 11) - (2818 \times 3)\\
253 &= 5 \times 50 + 3 & 3 &= 253 - (5 \times 50)\\
& & &= (2818 \times 152) - (769 \times 557)\\
5 &= 3 \times 1 + 2 & 2 &= 5 - (3 \times 1)\\
& & &= (769 \times 568) - (2818 \times 155)\\
3 &= 2 \times 1 + 1 & 1 &= 3 - (2 \times 1)\\
& & &= (2818 \times 307) - (769 \times 1125)
\end{aligned}$$

so

$$1 = (2818 \times 307) + (769 \times -1125)$$

and

$$1 \equiv 769 \times -1125 \mod 2818 \equiv 769 \times 1693 \mod 2818.$$

This tells us that the inverse of 769 modulo 2818 is 1693, so we get, for the first plaintext block,

$$P \equiv C^{1693} \equiv 1592^{1693} \equiv 1615 \mod 2819.$$

Aha! The number 1615 corresponds to the plaintext "po." Bob's complete decryption goes as follows:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ciphertext: | 1592 | 783 | 2264 | 924 | 211 | 44 | 1220 | 1548 |
| to the 1693rd power: | 1615 | 2305 | 1820 | 1520 | 805 | 1605 | 1516 | 1205 |
| split apart: | 16, 15 | 23, 5 | 18, 20 | 15, 20 | 8, 5 | 16, 5 | 15, 16 | 12, 5 |
| plaintext: | po | we | rt | ot | he | pe | op | le |

The number $\overline{e}$ that Bob needs to decrypt is traditionally called $d$, for **decryption exponent**. So, to summarize, Alice and Bob need to pick a prime $p$ larger than the largest possible plaintext number. They also need a key $e$ such that the GCD of $e$ and $p-1$ is 1, so that $e$ has an inverse modulo $p-1$. Then Bob needs to calculate the number $d$ that is the inverse of $e$ modulo $p-1$. Alice encrypts using the formula

$$C \equiv P^e \quad \text{modulo } p$$

and Bob decrypts using the formula

$$P \equiv C^d \quad \text{modulo } p.$$

This cipher is called the **Pohlig-Hellman exponentiation cipher**. It was invented by **Stephen Pohlig** and **Martin Hellman** in 1976 while they were working on the first public-key cryptography systems, which we shall explore in Chapter 7.

## 6.4 THE DISCRETE LOGARITHM PROBLEM

Now we can encrypt and decrypt using the Pohlig-Hellman cipher. What about Eve's methods of attack? The way to measure resistance to brute-force attacks is to see how many keys there are. The good keys are the numbers between 1 and $p-1$ that don't share any factors with $p-1$. If $p = 2819$, then $p - 1 = 2818 = 2 \times 1409$, and 1409 is prime. So $e$ can be any number between 1 and 2818 that doesn't have a factor of 2 or 1409, which means any odd number except 1409. There are 1408 such numbers, so there are 1408 good keys. That's not a huge number, but all we have to do to get more is choose a larger modulus, which also lets us use a larger block size. So brute-force attacks aren't a big problem, and ciphertext-only frequency attacks can be defeated by using a large block size.

What about known-plaintext attacks? For us to consider a cipher to be resistant to known-plaintext attacks, it needs to be clearly harder for Eve to recover the key than it is for Alice to encrypt or Bob to decrypt. If it weren't for the modular arithmetic, recovering the key would be easy. In order to find the exponent of an exponential expression when you know the base, you take a logarithm. If $C = P^e$, then $e = \log_P C$. In this case, Eve would see that the plaintext is 1615 and the ciphertext is 1592. So she knows $1615^e = 1592$ and $e = \log_{1615} 1592$. However, $\log_{1615} 1592$ is approximately 0.9981, and once again the modular arithmetic has messed things up. The problem of finding a whole number $e$ such that $C \equiv P^e$ modulo $p$ is called the **discrete logarithm problem**, and this is what Eve needs to solve.

It's not clear that solving the discrete logarithm problem is in fact harder than encryption or decryption—if Eve has some examples of $P$ and $C$, her first step is to guess $p$, which she can do fairly easily by looking at the largest ciphertext number in the message. Then she can multiply $P$ by itself repeatedly modulo $p$ until she gets $C$, keeping track of how many times it takes, and that will be $e$.

That seems remarkably like what Alice does to encrypt, right? The issue is that multiplying $P$ by itself $e$ times is actually *not* the best way for Alice to encrypt. Here's a better way.

Consider $e = 769$. I reminded you in Section 4.1 that 769 really means $7 \times 10 \times 10 + 6 \times 10 + 9$. So

$$P^{769} = P^{7\times10\times10+6\times10+9} = \left(\left(P^{10}\right)^{10}\right)^7 \left(P^{10}\right)^6 P^9.$$

If you count this out, you'll see that Alice needs only 46 multiplications, not 768. On the other hand, Eve will need all 768, since she doesn't know $e$ beforehand, so she can't split it up this way. As of 2016, people have been working hard for more than 35 years to find a fast way to solve the discrete logarithm problem, and so far Eve is not even close to being able to keep up with Alice and Bob. On the other hand, no one has been able to prove that she can't, either. Like several other problems we shall see in the next few chapters, the discrete logarithm problem is one that we think is hard, but no one knows for sure. We will talk more about this problem in Section 7.2.

### 6.5 COMPOSITE MODULI

You might think it's kind of annoying to have to use a prime number as the modulus in the Pohlig-Hellman cipher. Round numbers are easier to work with, so maybe you'd rather use 3000 as the modulus when the block size is 2. Alternatively, maybe the extra numbers in between the largest block and the modulus were bothering you and you'd rather use a modulus of exactly 2626. These are **composite numbers**, because they are made up of more than one prime multiplied together.

Encryption using exponentiation is no problem with a composite modulus. For example, if Alice wants to send Bob a message using a modulus of 2626 and the same key, $e = 769$, as before, she converts the plaintext to numbers and raises them to the 769th power as before.

| | | | | | |
|---|---|---|---|---|---|
| plaintext: | de | co | mp | os | in |
| numbers: | 4, 5 | 3, 15 | 13, 16 | 15, 19 | 9, 14 |
| together: | 405 | 315 | 1316 | 1519 | 914 |
| to the 769th power: | 405 | 1667 | 1992 | 817 | 1148 |

| | | | | | |
|---|---|---|---|---|---|
| plaintext: | gc | om | po | se | rs |
| numbers: | 7, 3 | 15, 13 | 16, 15 | 19, 5 | 18, 19 |
| together: | 703 | 1513 | 1615 | 1905 | 1819 |
| to the 769th power: | 1405 | 603 | 1615 | 137 | 1819 |

Decryption, once again, is the problem, and this time Fermat's little theorem is not going to come to our rescue. We can see the problem if we try to go through an example similar to the one in Section 6.2. Instead of the 13-letter Hawaiian alphabet, we will use the 15-letter Maori alphabet. Note that 13 is prime, but $15 = 3 \times 5$ is composite. Since 15 is not prime, not every number between 1 and 14 will be a good key. The number 2 will be, though, since the GCD of 15 and 2 is 1.

| plaintext | number | times 2 | ciphertext |
|---|---|---|---|
| a | 1 | 2 | E |
| e | 2 | 4 | I |
| h | 3 | 6 | M |
| i | 4 | 8 | O |
| k | 5 | 10 | R |
| m | 6 | 12 | U |
| n | 7 | 14 | NG |
| o | 8 | 1 | A |
| p | 9 | 3 | H |
| r | 10 | 5 | K |
| t | 11 | 7 | N |
| u | 12 | 9 | P |
| w | 13 | 11 | T |
| ng | 14 | 13 | W |
| wh | 15 | 15 | WH |

For the prime case, we multiplied all the numbers in the left column together and all the numbers in right column together, leaving out the number at the end of each column because it reduces to zero. If we do that here, we get

$$1 \times 2 \times 3 \times \cdots \times 14 \equiv (1 \times 2) \times (2 \times 2) \times (3 \times 2) \times \cdots \times (14 \times 2) \quad \text{modulo } 15,$$

$$1 \times 2 \times 3 \times \cdots \times 14 \equiv (1 \times 2 \times 3 \times \cdots \times 14) \times 2^{14} \quad \text{modulo } 15.$$

Now we want to cancel out $1 \times 2 \times 3 \times \cdots \times 14$ from each side, but unfortunately not all of those numbers have multiplicative inverses. Only the ones that have a GCD of 1 with 15 have inverses, and those are the only ones we can cancel out.

This is just like the problem with the bad keys. Since $15 = 3 \times 5$, we need to start over, leaving out the numbers that are multiples of 3, or 5, or both.

| plaintext | number | times 2 | ciphertext |
|---|---|---|---|
| a | 1 | 2 | E |
| e | 2 | 4 | I |
| i | 4 | 8 | O |
| n | 7 | 14 | NG |
| o | 8 | 1 | A |
| t | 11 | 7 | N |
| w | 13 | 11 | T |
| ng | 14 | 13 | W |

The numbers on the left-hand side are still the same as the numbers on the right-hand side, but in a different order. This kind of makes sense, since if a number on the left was a multiple of 3 or 5, we would expect 2 times it to be one also. So we crossed out the same numbers from each side.

If we try multiplying the columns again, we get

$$\begin{aligned} &1 \times 2 \times 4 \times 7 \times 8 \times 11 \times 13 \times 14 \\ &\quad \equiv (1 \times 2) \times (2 \times 2) \times (4 \times 2) \times \cdots \times (14 \times 2) \qquad \text{modulo } 15 \\ &1 \times 2 \times 4 \times 7 \times 8 \times 11 \times 13 \times 14 \\ &\quad \equiv (1 \times 2 \times 4 \times 7 \times 8 \times 11 \times 13 \times 14) \times 2^8 \qquad \text{modulo } 15. \end{aligned}$$

And now we *can* cancel $1 \times 2 \times 4 \times 7 \times 8 \times 11 \times 13 \times 14$, for instance, by multiplying by the inverse of each of them, so finally we get

$$1 \equiv 2^8 \qquad \text{modulo } 15.$$

Once again, the choice of 2 isn't important; any good key will do. But the choice of 15 clearly does make a difference—the 15 in the modulus produced an 8 in the exponent, and if we figure out how that happened, we'll be well on our way to figuring out how Bob can decrypt his message.

## 6.6 THE EULER PHI FUNCTION

Let's take a closer look at where the 8 came from in the last example. We listed all of the numbers from 1 to 15,

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,

and we got rid of all those that did *not* have a GCD of 1 with 15:

1, 2, ~~3~~, 4, ~~5~~, ~~6~~, 7, 8, ~~9~~, ~~10~~, 11, ~~12~~, 13, 14, ~~15~~.

This leaves 8 numbers behind. In other words, 8 is the number of whole numbers less than or equal to 15 that have a GCD of 1 with 15.

In general, we can define $\phi(n)$ (that's the Greek letter phi) to be the number of positive whole numbers less than or equal to $n$ that have a GCD of 1 with $n$. For example, we have

| $n$ | $\phi(n)$ | $n$ | $\phi(n)$ |
|---|---|---|---|
| 1 | 1 | 11 | 10 |
| 2 | 1 | 12 | 4 |
| 3 | 2 | 13 | 12 |
| 4 | 2 | 14 | 6 |
| 5 | 4 | 15 | 8 |
| 6 | 2 | 16 | 8 |
| 7 | 6 | 17 | 16 |
| 8 | 4 | 18 | 6 |
| 9 | 6 | 19 | 18 |
| 10 | 4 | 20 | 8 |

We already know what $\phi(n)$ should be if $n$ is prime, since every whole number will be counted except the number itself. Other than that, the function seems pretty mysterious.

The person who figured out the pattern was the great genius mathematician of the eighteenth century, in the way Gauss was for the nineteenth and Fermat was in the seventeenth. His name was **Leonhard Euler**, and while he was born in Switzerland, he did most of his work at prestigious scientific academies in Russia and Prussia. In 1736 he was the first one to publish a proof of Fermat's little theorem, and he later published several more. In one of these papers, in 1763, he introduced the function we now write as $\phi(n)$ and call the **Euler phi function**. And, he uses this function to prove what we now call the **Euler-Fermat theorem**.

**Theorem** (The Euler-Fermat Theorem) *For any positive whole number n and any k between* 1 *and n such that the GCD of n and k is* 1,

$$k^{\phi(n)} \equiv 1 \quad \text{modulo } n.$$

If $n$ is a prime number, then $\phi(n)$ will be $n - 1$, and we have Fermat's little theorem again. And if $n$ is 15, then $\phi(n)$ is 8 and we have our example. Now we know what the Euler phi function is and we have some idea what it's good for. But if we have to calculate $\phi(n)$ by checking a GCD for every number between 1 and $n$, that's going to be a very slow process.

Luckily, there's an easier way. Let's go back to our example and watch a little more closely as we cross out the "bad keys." The divisors of 15 are 1, 3, 5, and 15, so we know we have to cross out the numbers that are multiples of 3:

| | | |
|---|---|---|
| 1 | 2 | ~~3~~ |
| 4 | 5 | ~~6~~ |
| 7 | 8 | ~~9~~ |
| 10 | 11 | ~~12~~ |
| 13 | 14 | ~~15~~ |

Since we are crossing out every third number, there are $15/3 = 5$ crossed-out numbers. We also have to cross out multiples of 5:

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | ~~5~~ |
| 6 | 7 | 8 | 9 | ~~10~~ |
| 11 | 12 | 13 | 14 | ~~15~~ |

This time we have crossed out every fifth number, and there are $15/5 = 3$ crossed-out numbers. We don't have to cross out multiples of 15 because any multiple of 15 is a multiple of 3 (and of 5), so it's already been crossed out.

So how many numbers are not crossed out? It should be $15 - 3 - 5 = 7$, but when we did it before, there were 8. Do you see why? It's because we crossed out 15, which is a multiple of both 3 and 5, twice. So we have to add it back in, giving us $15 - 3 - 5 + 1 = 8$ numbers not crossed out. In general, we have this formula if $p$ and $q$ are two different prime numbers:

$$\phi(pq) = pq - p - q + 1.$$

With a little bit of algebra, we can rearrange that into the more common form:

$$\phi(pq) = (p-1)(q-1).$$

Now what about Bob and our cipher? In that case we had $n = 2626 = 2 \times 13 \times 101$, and if you work through all the crossings out, you will see that

$$\frac{2626}{2} + \frac{2626}{13} + \frac{2626}{101} = 13 \times 101 + 2 \times 101 + 2 \times 13$$

numbers get crossed out, but

$$\frac{2626}{2 \times 13} + \frac{2626}{2 \times 101} + \frac{2626}{13 \times 101} = 101 + 13 + 2$$

got crossed out twice and have to be added back. However 1 number, namely, 2626, has now been crossed out 3 times and added back in 3 times, so it has to come out again. In other words:

$$\begin{aligned}\phi(2626) &= 2626 - 2 \times 13 - 2 \times 101 - 13 \times 101 \\ &\quad + 2 + 13 + 101 - 1 = 1200.\end{aligned}$$

In general, if $p$, $q$, and $r$ are three different prime numbers,

$$\begin{aligned}\phi(pqr) &= pqr - pq - pr - qr + p + q + r - 1 \\ &= (p-1)(q-1)(r-1).\end{aligned}$$

And you can probably see the pattern for any product of different primes.

## 6.7 DECRYPTION WITH COMPOSITE MODULI

Now we should be able to figure out how to decrypt a message encrypted using the Pohlig-Hellman cipher and a composite modulus. Once we know $\phi(n)$, the Euler-Fermat theorem tells us that

$$P^{\phi(n)} \equiv 1 \equiv P^0 \quad \text{modulo } n.$$

This means that if we are looking at an equation modulo $n$, then we can treat the exponent as if we were working modulo $\phi(n)$. This is the equivalent of what we did with Fermat's little theorem earlier. In the case of $n = 2626$, we have

$$P^{1200} \equiv P^0 \quad \text{modulo } 2626.$$

If the encryption exponent is $e = 769$, the decryption exponent will be the inverse of $e$ modulo 1200—assuming there is one. Remember that for $e$ to have an inverse modulo 1200, the GCD of $e$ and 1200 needs to be 1. Otherwise, $e$ is a bad key, and Alice shouldn't have picked it in the first place.

So Bob's first step in decrypting the message is to use the Euclidean algorithm to find the inverse of $e = 769$ modulo 1200.

$$\begin{aligned}
1200 &= 769 \times 1 + 431 & 431 &= 1200 - (769 \times 1)\\
769 &= 431 \times 1 + 338 & 338 &= 769 - (431 \times 1)\\
&& &= (769 \times 2) - (1200 \times 1)\\
431 &= 338 \times 1 + 93 & 93 &= 431 - (338 \times 1)\\
&& &= (1200 \times 2) - (769 \times 3)\\
338 &= 93 \times 3 + 59 & 59 &= 338 - (93 \times 3)\\
&& &= (769 \times 11) - (1200 \times 7)\\
93 &= 59 \times 1 + 34 & 34 &= 93 - (59 \times 1)\\
&& &= (1200 \times 9) - (769 \times 14)\\
59 &= 34 \times 1 + 25 & 25 &= 59 - (34 \times 1)\\
&& &= (769 \times 25) - (1200 \times 16)\\
34 &= 25 \times 1 + 9 & 9 &= 34 - (25 \times 1)\\
&& &= (1200 \times 25) - (769 \times 39)\\
25 &= 9 \times 2 + 7 & 7 &= 25 - (9 \times 2)\\
&& &= (769 \times 103) - (1200 \times 66)\\
9 &= 7 \times 1 + 2 & 2 &= 9 - (7 \times 1)\\
&& &= (1200 \times 91) - (769 \times 142)\\
7 &= 2 \times 3 + 1 & 1 &= 7 - (2 \times 3)\\
&& &= (769 \times 529) - (1200 \times 339)
\end{aligned}$$

so

$$1 = (769 \times 529) + (1200 \times -339)$$

and

$$1 \equiv 769 \times 529 \quad \text{modulo } 1200.$$

The decryption exponent is $d = 529$, and the decryption goes as follows:

| | | | | | |
|---|---|---|---|---|---|
| ciphertext: | 405 | 1667 | 1992 | 817 | 1148 |
| to the 529th power: | 405 | 315 | 1316 | 1519 | 914 |
| split apart: | 4, 5 | 3, 15 | 13, 16 | 15, 19 | 9, 14 |
| plaintext: | de | co | mp | os | in |

| | | | | | |
|---|---|---|---|---|---|
| ciphertext: | 1405 | 603 | 1615 | 137 | 1819 |
| to the 529th power: | 703 | 1513 | 1615 | 1905 | 1819 |
| split apart: | 7, 3 | 15, 13 | 16, 15 | 19, 5 | 18, 19 |
| plaintext: | gc | om | po | se | rs |

Actually, I have cheated a bit. The Euler-Fermat theorem guarantees only that the exponents behave like we want if the GCD of $P$ and $n$ is 1. This isn't true for some of our plaintext blocks, such as 1316; in fact the GCD of 1316 and 2626 is 2. It turns out that if $n$ is a product of *different* primes, then decryption *does* always work properly, but I'm not going to try to justify that in this book. If you want to see the proof, I've put some references in the endnotes.

---

**■ ■ ■ SIDEBAR 6.1. FEE-FI-FO-FUM ■ ■ ■**

If $n$ is a product of primes that appear multiple times, then we can still find a formula for $\phi(n)$, even though we won't be able to easily use the Pohlig-Hellman cipher. Suppose that $n = 12 = 2^2 \times 3$. The divisors of 12 are 1, 2, 3, 4, 6, and 12. When we are crossing out bad keys, we need to cross out multiples of 2 and multiples of 3, and this will also eliminate multiples of 4, 6, and 12. First we cross out all the multiples of 2:

| | |
|---|---|
| 1 | ~~2~~ |
| 3 | ~~4~~ |
| 5 | ~~6~~ |
| 7 | ~~8~~ |
| 9 | ~~10~~ |
| 11 | ~~12~~ |

There are $12/2 = 6$ of those. And then we cross out all the multiples of 3:

| | | |
|---|---|---|
| 1 | 2 | ~~3~~ |
| 4 | 5 | ~~6~~ |
| 7 | 8 | ~~9~~ |
| 10 | 11 | ~~12~~ |

There are $12/3 = 4$ of those. But both 12 and 6 have been crossed out twice, since they are both divisible by 2 and by 3, so we have to add them back. Thus $\phi(n) = 12 - 6 - 4 + 2 = 4$. In general, we have this formula if $p$ and $q$ are different prime numbers:

$$\phi(p^a q^b) = p^a q^b - \frac{p^a q^b}{p} - \frac{p^a q^b}{q} + \frac{p^a q^b}{pq}.$$

And we can rearrange that into the more common form:

$$\phi(p^a q^b) = \left(p^a - \frac{p^a}{p}\right)\left(q^b - \frac{q^b}{q}\right) = \left(p^a - p^{a-1}\right)\left(q^b - q^{b-1}\right).$$

If $n = p^a q^b r^c$ is a product containing three different primes, then

$$\phi(p^a q^b r^c) = \left(p^a - p^{a-1}\right)\left(q^b - q^{b-1}\right)\left(r^c - r^{c-1}\right),$$

and so on.

For instance, if $n = 3000 = 2^3 \times 3 \times 5^3$, then

$$\phi(3000) = (2^3 - 2^2) \times (3 - 1) \times (5^3 - 5^2) = 800.$$

Alice can encrypt a message with $e = 769$ and $n = 3000$:

| | | | | | | |
|---|---|---|---|---|---|---|
| plaintext: | sy | st | em | er | ro | rx |
| numbers: | 19, 25 | 19, 20 | 5, 13 | 5, 18 | 18, 15 | 18, 24 |
| together: | 1925 | 1920 | 513 | 518 | 1815 | 1824 |
| to the 769th power: | 125 | 0 | 2073 | 368 | 375 | 2424 |

If Bob uses the Euclidean algorithm, he will find that the inverse of 769 modulo 800 is 129, so he attempts to decrypt using $d = 129$:

| | | | | | | |
|---|---|---|---|---|---|---|
| ciphertext: | 125 | 0 | 2073 | 368 | 375 | 2424 |
| to the 129th power: | 125 | 0 | 513 | 2768 | 375 | 1824 |
| split apart: | 1, 25 | 0, 0 | 5, 13 | 27, 68 | 3, 75 | 18, 24 |
| plaintext?: | ay | ?? | em | ?? | c? | rx |

Remember that the Euler-Fermat theorem does not guarantee that decryption will work properly unless the GCD of $P$ and $n$ is 1. Two of the blocks come through all right: 513, which has a GCD of 1 with 3000, and 1824, which has a GCD of $24 = 2^3 \times 3$ with 3000 but works anyway. However, most of the blocks come through with incorrect letters or numbers that do not correspond to letters at all. You might hope that reducing the individual 2-digit numbers modulo 26 would help, but it doesn't. If the system was working correctly, Bob would get the same numbers that Alice started with. The general formula for $\phi(n)$ is useful for other situations but not really for the Pohlig-Hellman cipher.

---

## 6.8 LOOKING FORWARD

So, you may ask, are exponentiation ciphers the state of the art in modern ciphers? As it happens, they aren't actually used that much. Ciphers such as AES appear to have just as good resistance to attacks and work much faster, even with the trick that we have noted for speeding up exponentiation. Instead, we shall see in Chapters 7 and 8 that the ideas used in this cipher, and especially the hardness of the discrete logarithm problem, turn out to be very important to a very exciting idea known as public-key cryptography.

When Pohlig and Hellman were developing their cipher, they briefly considered using composite moduli but rejected it on the grounds that the convenience wasn't worth the complication. They missed a bet, because exponentiation with composite moduli is a key ingredient in the very important system that we will see in Section 7.4.

On the other hand, Pohlig and Hellman also figured out how to use their cipher with the sort of finite field arithmetic we saw in Section 4.5. This eventually turned out to be another important idea because finite-field arithmetic modulo 2 is a convenient way for computers to manipulate bits, as we also saw in that section.