Chapter 62 (Long Version)

After he had showered and shared a meal with the others, Danny Tenacce was about to return to the room to check his work when he spied Salito heading toward him.

"It's my shift. You're stuck with me again."

Danny entered the room and motioned her in.

"What's wrong?" she said in a low voice.

"Nothing. I just want to know what you think about Father McCleary?"

"Well, he's big, strong, smart, incredibly even-tempered. A little too perfect, if you ask me."

"Exactly! Do you think he's really a priest?"

"Suspicious, eh? You'd make a good cop. I was wonderin' that myself."

"Well?"

"I've had a couple o' thoughts. I always consider the worst scenario first."

"What's that?"

"That he and DeAngelo are Knights posing as Brothers, waitin' for us to find and decipher the key so they can steal it and decipher the book. Or that they're both Brothers and The Brothers really are the bad guys. I talked ta Joe about it."

"What did he say?"

"I said maybe DeAngelo is the one that tipped the Knights as to the location o' the safe house. He said that he never told DeAngelo where the house was. I said maybe he put The Hulk on your tail. After all, there's no question the man's got skills."

Danny laughed and nodded. "And what did he say to that?"

"He said, no way, that he and DeAngelo went way back and if DeAngelo was a rat, then his life was over anyway. I thought, strong words from a strong man with good instincts. I believe him."

"He convinced me similarly. That's why I didn't balk when he said he thought the book was safe with Father Frank. But The Hulk, as you aptly called him, could still be a double agent working for The Knights. Or some other entity."

"Joe considered that. His argument against it was that DeAngelo had learned ta keep his guard up from his military experience, and from his years as a priest, had acquired a deep understanding of human character. He didn't think he could be duped."

"And if he was?"

"Then when McCleary tries to take the book, Joe plans ta take 'im out."

"That might be easier said than done."

"I agree, but you know Joe. He said another thing that made me think he's right, though.

"What's that?"

"The Knights don't seem too interested in keepin' us around ta decipher the book. I mean, they were tryin' ta have us arrested, and before that, at Mary's Bookstore, they frankly tried ta kill us. More likely they just wanna destroy it. Either, contrary to legend, they know

what's in the book and don't want its contents known or they don't wanna take any chances that the book says what they don't want it ta say. Either way, if McCleary was a Knight, he woulda made a play for the book on Staten Island. And if Father DeAngelo was a Knight. Well …"

"He's got a point."

"I've got another idea."

"Which is?"

"That McCleary works for the United States government. DeAngelo said that he was with the Delta Force. Maybe he still works for the Delta Force, or the CIA, or the government in some capacity, and is just posin' as a priest. Think what would happen if, God forbid, the book reveals that Jesus was just part of a scheme ta deceive the public into thinkin' he was divine; deceive 'em so they would believe in an afterlife where there was reward for good behavior and punishment for evil; that the whole thing was a ruse ta keep people's behavior in check. If the ruse were exposed, social chaos would ensue."

"Anarchy."

"Huh?"

"Anarchy would be a more appropriate term. Chaos has a very specific meaning in science and mathematics. It refers to a deterministic system in which the state of a system at some time is sensitive to the system's initial conditions; to quote Lorenz, 'a system in which the present determines the future, but the approximate present does not approximately determine the future.'"

"Genesis says it's the waste and void before God said 'Let there be light.' I mean it as defined in common usage: disorder. Does the word really matter?"

"It might."

"Ok. Anarchy then. No way the government wants ta deal with that. I think McCleary may be here ta snatch the book if it says what I hope and pray it doesn't."

"You think there'll be anarchy if the truth is exposed?"

"Absolutely."

"I disagree. I think people are rational beings who would recognize that behaving in a civilized manner is in their best interest, whether or not there's eternal bliss or damnation hanging over their heads."

"With all due respect, doctor, I contend that your judgement is impaired by your exposure ta only a limited segment of society. As cops, your father and I see the whole spectrum of humanity, especially the unsavory element with which you have little experience, and I assure you, if Christianity was shown ta be a hoax, lawless pandemonium would result."

"I must admit, I find myself wondering—after this debate and our previous discussion about consciousness—whether the woman who passes herself off as a jockette/cheerleader from Brooklyn isn't really a professor from Harvard. No Standerford—that school's a little more conservative. At any rate, I think we'll have to settle both of these issues later. I hear the rest of them coming."

"Yes we will," Salito replied.

The door opened. Tenacce made his customary intrusive entrance with McCleary in tow. "Did I interrupt somethin'?"

"No, we were just talking."

"About you, actually."

"Oh," said Tenacce, his detective's suspicion aroused. "Any progress?"

"Not a bit but I haven't checked the computer in a while."

Danny Tenacce sprang hopefully from his chair and rattled the mouse to bring the digital screen to life. When he saw the numbers being added at warp speed to the bottom of the window he slammed his hand down on the desk.

"I understand your frustration," said McCleary.

"How could you?"

"Because I've worked on this problem myself. For seven years. With the same results."

"You've worked on this problem?"

"Why are you surprised?"

"Because we just found the key to the key."

"It's 2048 bits long, if that will help."

"I know that but how do you know that?"

"Word of mouth."

"Ah, yes. The Brotherhood. So then you know that solution of this problem is tantamount to proving P = NP."

"It would have far-reaching consequences, to be sure."

"I was considering the analogy more as an illustration of the problem's difficulty but you're right."

"You've already developed an algorithm with far-reaching consequences in your decryption of the last papyrus. What do you plan to do with it?"

"I'm uncertain."

Tenacce scowled. "Sounds interesting. Maybe you two could translate inta English."

Danny was surprised that his father had been listening. "We're talking about the difficulty of decoding the key."

He remained at the desk, a place at which he had been for the better part of the last three days, save the few hours when his brain and lagging lids would not allow him to be there—sat there, regathering his pencil and pad, about to readdress the problem that had befuddled him—when he noticed Salito and his father staring.

"And," Salito said finally.

"And what?"

"Why is decoding the key so difficult?"

"It's a complicated issue," said Danny, doing his best to deflect the inquiry.

"Like everything else in this case," Salito responded.

"Let's have it," demanded Tenacce.

"It's gonna take some time." Danny looked at his father.

"We got plenty o' that," his father said.

"Better make yourselves comfortable then. I need a break anyway." Danny rose from his seat and moved a whiteboard to the center of the room before his audience. He uncapped a marker and began.

"In computer science, the complexity of solving a problem is classified into categories. This complexity depends on the time it takes to solve the problem, which, in turn, depends on the number of steps it takes to accomplish the task. P, one of those categories, stands for polynomial time. This refers to the fact that the mathematical expression for the number of steps needed to solve the problem is a polynomial. It means that, in the worst case scenario, given you have n things to test, it would take no more than n^k steps to solve the problem—solve the problem, that is, on what's called a deterministic Turing machine, a theoretical machine like a computer that can execute only one task at a time. Let me explain.

"Say you want to compare each of 1000 cards with each other to see which card has the highest number printed on it. Then n = 1000. You would compare each card to the other 999. Each comparison is a step. That's (n-1) = 999 steps per card, times n = 1000 cards; $(n - 1)n = n^2 - n = 1000^2 - 1000 = 1{,}000{,}000 - 1000 = 999{,}000$ steps. A computer can do that kind of calculation in less than a second. Notice that as n becomes large, the n^2 term gives you most of the value of the expression; n is much smaller. In the case of the cards, the n^2 term gives you 1,000,000 and the n term only contributes minus 1000, small compared to a million, so you ignore the n term. That's why they say the expression for the number of steps in polynomial time is n^k. In the above example, n = 1000 and k = 2. The characteristic of P-type problems is that they are easy for a computer to solve and also easy to verify. In the problem of the cards, I showed that you can solve the problem in polynomial time. You can also verify, in polynomial time, whether or not the card you picked as having the highest number actually has the highest number.

"Next consider the problem of trying to figure out what numbers you have to multiply together to get another number. First, let me give you a couple of definitions. The numbers that we're going to be dealing with in this problem are natural numbers: positive whole numbers like you're used to dealing with: 1, 2, 3, 4 …. Not fractions, decimals, square roots, and so forth. Second, the numbers you multiply together to get another number are called factors. A prime number is one in which the only factors you can multiply together to get it are 1 and the number itself. A composite number is a number that has factors other than 1 and itself. So 29 is a prime number because the only numbers you can multiply to get it are 1 and 29. On the other hand, 6 is a composite number because you can multiple 2 x 3 to get 6, as well as 1 x 6. Euclid proved about 2300 years ago that every composite number can be expressed as multiple prime numbers, multiplied together. The string of prime factors multiplied together to make an integer are unique. For example, the number 12 has prime factors 2 x 2 x 3. No other number has this combination of prime factors.

"Now for the specific problem I want to examine. I want to talk about multiplying two prime numbers together to get a composite number. Let's say that I give you the number 203. To determine it's prime factors, you'd start with 2 and divide it into 203. It doesn't go evenly so you

go to the next prime integer 3. It doesn't divide 203 evenly either. Neither does the next prime integer, 5. How about 7. 7 into 203 goes 29 times. When you get two prime factors that, when multiplied together, give you the number, you're done. So the prime factorization of 203 is 7 times 29.

"That's easy. Now suppose I ask you to factor 8,162,821."

"Good luck," blurted Tenacce. "Ya'd have ta start at 3. Ya don't need ta bother with 2 'cause ya know it won't divide an odd number without somethin' left over. Ya'd take 3 and divide that and every odd number inta your number—correction, every odd prime number …"

"All prime numbers are odd other than 2," Danny pointed out.

Tenacce thought about it for a few seconds. "Right, every prime number until ya find somethin' that goes in even. If you're lucky, ya find the answer before ya get ta 8,162,820."

"That's about right. Actually, you can shorten the process a little more. Say you're going to factor 49. You start with 2 and you try to divide it into 49. It doesn't work. Same with 3, and 5. You don't need to try 4 or 6 because they're not prime. 7 x 7 works. That's all you need to do. Going any further is redundant. In the worst case, the furthest you'll ever have to go is to an integer that, when multiplied by itself, equals the number. Such a number is called a square root. If you want to find prime factors for 119, the furthest you would possibly need to go would be to 10 because the square root of 119 is 10.9. Although, in this case, you wouldn't even need to go that far because 7 x 17 = 119. The square root of 8,162,821 is 2,857 plus a decimal. So you might need to try dividing up to 2,857 numbers into 8,162,821 to figure out if it has prime factors. Of course, the only numbers you would need to divide into 8,162,821 would be prime numbers, so if you knew which numbers less than 2,857 are prime, you would only have to try those numbers and the number of steps to solve the problem would be considerably less than 2,857.

"The prime factors that happen to give 8,162,821 are 3011 and 2711. It would take a lot more time to do the 2711 steps to figure this out than to *verify* that 3011 and 2711 are the two prime factors that are the correct ones. All you would have to do to verify that 3011 and 2711 are the answer is multiply them together. That wouldn't take much time at all. A computer can still handle checking 2711 numbers easily but what if the number you're trying to factor is 617 digits long, like the 2048 bit key that we—er, I—have got to decode to find out where the key to the book is?

"What's this bit thing that ya keep talking' about?"

"A bit is the basic information unit with which a computer works. A computer is basically a long string of logic gates hooked together. Each gate can give an answer of yes or no, represented numerically by a 1 or a 0, respectively. Numbers are also represented by 0's or 1's. They're called binary or base 2 numbers. The usual numbers we're used to are decimal or base ten numbers. Basically, that means that each digit in the number can be one of ten possibilities, 0-9. With binary numbers, there are only 2 possibilities for each number: 0 or 1. You can represent the same number in either binary or decimal form."

Danny Tenacce erased what he had and scrawled a new diagram on the whiteboard:

Decimal

100	10	1
10^2	10^1	10^0
	2	7

Binary

32	16	8	4	2	1
2^5	2^4	2^3	2^2	2^1	2^0
	1	1	0	1	1

"Take the decimal number 27. You remember exponents. That's the number of times you have to multiply the base number (in this case, 10) together. So, in the diagram, $10^2 = 10x10$, $= 100$, $10^1 = 10, 10^0 = 1$ (any number to the zero power is 1). 27 in decimal form is 2 groups of tens and 7 groups of ones so you put a 2 in the tens column and 7 in the ones column. There are no hundreds or higher digits so you leave them blank.

"You separate binary numbers into columns in a similar fashion. $2^5 = 32, 2^4 = 16, 2^3 = 8$ $2^2 = 4, 2^1 = 2, 2^0 = 1$. To express the decimal number 27 in binary form, start by taking the column with the largest value that can be divided into 27, in this case, 16. Divide 16 into 27. It goes once so you have 1 group of 16 in 27. Therefore, put a 1 in the sixteens column. When you divide 16 into 27, you get a remainder of 11. Go to the next column, 8. 8 goes into 11 once so put a 1 in the eights column. You're left with 3. Go to the next column, 4. 4 doesn't go into 3 so put a 0 in the fours column. You still have 3 left over. Go to the twos column. 2 goes into 3 once so put a 1 in the twos column. You've got 1 left over. Go to the ones column. 1 goes into 1 once. Put a 1 into the ones column. You've got no more numbers left so you're done. So 27 in binary form is 11011. Each column is a bit so 27 is a 5 bit number. It's stored in a computer with five little logical units. Notice that it takes more digits to express a given number in binary form than in decimal form: two digits in decimal form and five digits in binary form in the case of 27. A 2048 bit key like the one that guards directions to the key is a 2048 digit binary number. The decimal equivalent is 617 digits long.

"To factor a number this big (call it n), as I described previously, you might need to carry out as many as the square root of n calculations. The square root sign is $\sqrt{\ }$. Taking the square root of a number is the same as the number raised to half its exponent. So

$\sqrt{10^{617}} = 10^{617 \cdot \frac{1}{2}} = 10^{308.5}$. Let's round off and make it 10^{308}.

So you might need to carry out as many as 10^{308} calculations, i.e. divide as many as 10^{308} numbers into the n to find the answer. Each of these divisions probably takes—I don't know—

maybe 1000 (i.e. 10^3) computer steps or FLOPs (floating point operations) as they're called. That's probably a conservative estimate, but as you'll see, it really won't matter. A decent computer can carry out about a teraflop each second. That's 10^{12} FLOPs, or computer steps, per second. So a decent computer will be able to carry out 10^9 divisions per second. I got that by doing the following:

$$\frac{10^{12}\cdot\frac{\text{steps}}{\text{second}}}{10^{3}\cdot\frac{\text{steps}}{\text{division}}}=10^{9}\cdot\frac{\text{steps}}{\text{second}}\cdot\frac{\text{division}}{\text{steps}}=10^{9}\frac{\text{divisions}}{\text{second}}$$

To figure out how many seconds it'll take to carry out 10^{308} divisions, you divide that by the number of divisions per second your computer can carry out:

$$\frac{10^{308}\text{divisions}}{10^{9}\frac{\text{divisions}}{\text{second}}}=\frac{10^{308}}{10^{9}}\cdot\frac{\text{divisions}}{\frac{\text{divisions}}{\text{second}}}=10^{299}\text{ seconds}$$

There are 60 seconds per minute, 60 minutes per hour, 24 hours per day and 365 days per year, so there are 31,536,000 seconds per year:

$$60\frac{\text{seconds}}{\text{minute}}\cdot 60\frac{\text{minutes}}{\text{hour}}\cdot 24\frac{\text{hours}}{\text{day}}\cdot 365\frac{\text{days}}{\text{year}}=31{,}536{,}000\frac{\text{seconds}}{\text{year}}=3.1536\times10^{7}\frac{\text{seconds}}{\text{year}}$$

To figure out how long it would take—in years—to carry out all 10^{308} division steps necessary to find the secret key, you divide the length of time to do this—in seconds—by the number of seconds in a year:

$$\frac{1\times10^{299}\text{seconds}}{3.1536\times10^{7}\frac{\text{seconds}}{\text{year}}}=0.31709792\times10^{292}\text{years}=3.1709792\times10^{291}\text{years}$$

Even if you had 100 computers, it would still take 3.1709792×10^{289} years. The universe is about 13.7 billion years old. A billion is 10^9.

"I think you're starting to get the picture. Factoring large numbers with two large prime factors is an extremely hard problem to solve but if you have the two factors, multiplying them together on a computer to verify that you have the correct answer is easy.

"A problem that is hard to solve but easy to verify is called an NP problem. NP stands for nondeterministic polynomial. A nondeterministic Turing machine is a theoretical machine that can carry out multiple possible actions simultaneously. A deterministic Turing machine, as I described a little while ago, can execute only one action at a time and takes a long time to solve NP problems. The time that a deterministic Turing machine would require to solve the prime factorization problem I described is called exponential time because it can be expressed mathematically by the general formula k^n. For factoring prime numbers, k would be the base of the number to be factored and n is the square root of the number; in our example $k^n = 10^{308}$ steps. As you can see, that number gets very big, very fast as n gets big. On the other hand, the finding of the card with the highest number in a deck of 1000 that I described before could be solved in polynomial time. The number of steps to the solve the problem is given by a polynomial expression, specifically, $n^k = 1000^2 = 10^{3\cdot2} = 10^6$. Obviously, 10^6 is a lot less than 10^{308}.

"So it would take a deterministic Turing machine exponential time to solve an NP problem like finding prime factors of a large number. In contrast, a nondeterministic Turing machine could carry out the same NP problem in much less time; specifically in polynomial time. Thus the name. NP - nondeterministic polynomial.

"As I've said, modern computers work like deterministic Turing machines. No NP problem has ever been solved by modern computers in polynomial time. Computer scientists aren't sure if that's because it's impossible or because no one has been smart enough to do it yet. If it is possible, that would mean that P equals NP. If it's impossible, then it would mean that P does not equal NP. The Clay Mathematics Institute in Oxford, England has seven unsolved problems for which it will pay anyone who solves them $1,000,000 per problem. A formal proof that P equals NP, or that P does not equal NP, is among those problems. It would be important to know since, if P equals NP, programmers will work harder to solve NP problems, many of which have important practical implications. If it can be proven that P does not equal NP, then computer scientists will stop wasting their time working on NP problems.

"Decrypting the key would essentially provide an example that P=NP which would be tantamount to proving P = NP, a task that many have tried but at which no one has yet succeeded. That gives you some idea what we're up against."

"We have reason to be optimistic, though" said McCleary.

"Why is that?"

"At least it's an RSA key you're dealing with. If it were a 2048 bit symmetric key … or even 256."

Salito shifted uncomfortably in her seat. "Then the far reachin' consequences that you mentioned are that, usin' this algorithm, current encryption methods could be cracked. Computer security would be out the window."

"Those would be the consequences," confirmed McCleary. "I can assure you, there are many people who would love to gain possession of the factoring algorithm Dr. Tenacce has already designed. I wouldn't be surprised if a couple of agents from the US government pay you a visit when the smoke clears."

"If the smoke clears," Danny Tenacce said. "And I'm sure you'll be the one sending them."

"Better than the Russian mob. So what are you going to do with it?"

"I haven't decided yet."

"What does factoring a number have ta do with encryption anyway?" asked Salito.

"Everything," Danny replied. "At least where RSA encryption is concerned."

"Can you elaborate."

"Do you want the long version or the short version?"

"We'll take the long version," Salito volunteered quickly.

"Are you sure? The math gets considerably worse."

"We endured P versus NP, didn't we?" Tenacce said with resignation.

"Ok, then. Say Mary wants to send a confidential message to Danny and keep it from John, who's always trying to intercept it. Danny sends an open padlock to Mary, a padlock a copy of which John or anyone else who wants one can have. Mary puts her message in a box and locks it with the padlock. The nature of the box and lock are such that no one can cut off the lock, cut a hole in the box or otherwise access the message except by using the key to open it, a key that only Danny has. Mary sends the locked message back to Danny, in plain view of John, who, without the key, can't open it. When Danny receives the box, he easily opens the box and reads the message. This is an example of a one-way function. The padlock is easy to lock but hard to unlock. Unless, of course, you have the key.

"In RSA encryption, the functions of the physical lock and key are accomplished with mathematics. The padlock function is performed by a thing called the public key while the function of the key that Danny uses to unlock the padlock is referred to as the private key.

"Our situation is a little different in that Mary made up Danny's public key, encrypted the message with the public key and sent the encrypted message to Danny. Now all Danny has to do is use the secret key to unlock the message. The problem is, Danny doesn't have the secret key. He has to guess it. By factoring a 2048 bit number."

"Kinda like an NP problem," said Tenacce.

"Exactly like an NP problem," chimed McCleary.

"That's the basic concept. To practically implement this strategy, computers use a one-way mathematical function called a trap door function—the mathematical analogue of the padlock—easy to encrypt but difficult to decrypt, unless you have the key. This function makes use of modular arithmetic, a technique to which you've already been exposed. Peterson used it to construct his message in blood. It's crucial to the process so I need to refresh your memories.

"In principle, it's simple. It's just clock arithmetic. Let me give you an example. Take the equation $8 = 20 \bmod 12$. That means if you divide 20 by 12 you get a remainder of 8. An alternative way to think about it is to picture a clock like the ones you tell time with. It has 12 numbers on its face. Ignore the minute markers. Each number signifying an hour is a tick. If you start at 12 and go 20 ticks around the clock you end up on 8. The number of ticks on the clock is called the modulus. In this case, it's 12 but it could be 3 or 8 or 4892. Anything. Let's take a

clock with 8 numbers—or ticks—on its face. Let's use this to figure out the meaning of this equation: $2 = 10 \bmod 8$. The clock in this case has 8 numbers—or ticks—on its face. To solve this equation, you'd start at the top, at 8 (which is the same as 0, by the way), go 10 ticks around the clock and end up at 2, which is the answer. Alternatively, we could take 10 and divide it by the modulus 8. We'd end up with a remainder of 2, which is, again, the correct answer.

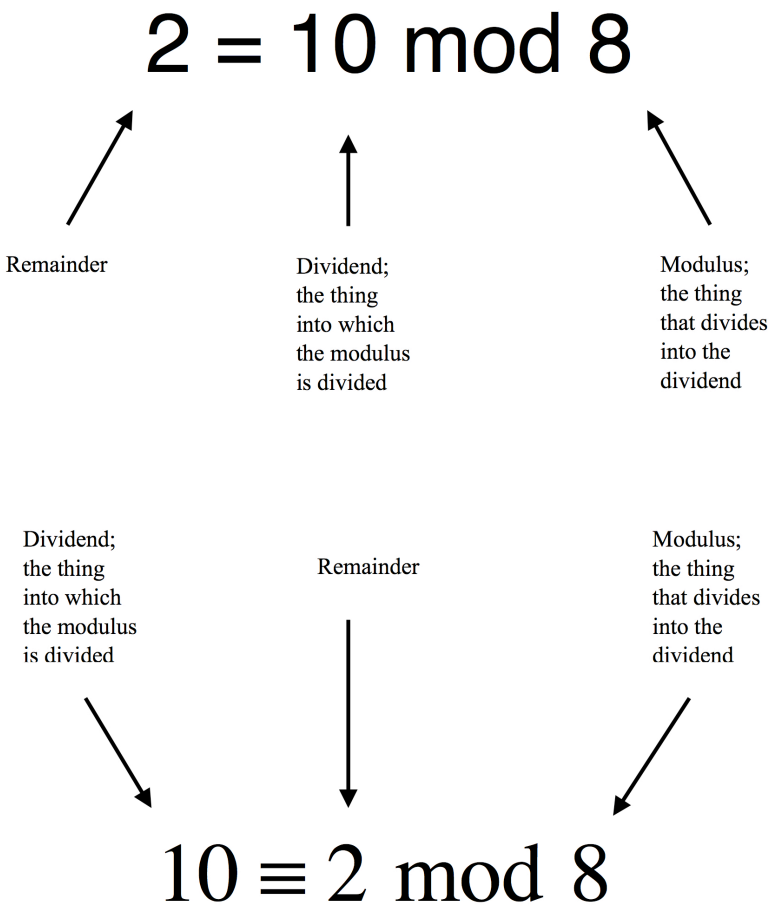
"One other thing we need to address regarding modular arithmetic is the concept of congruence. The symbol for congruence in modular arithmetic is '$\equiv$.' This refers to a group of entities that are equivalent. For example, 2, 10 and 18 are all congruent to $2 \bmod 8$. That means use a clock with 8 hour marks, or ticks. Start at the top of the clock, at 8 (or 0, to which it is equivalent). If you go around the clock 2 ticks, 10 ticks or 18 ticks, in each case, you wind up in the same place: at 2. The mathematical expressions for these relationships would

$$2 \equiv 2 \bmod 8 \quad \text{or equivalently} \quad 2 = 2 \bmod 8$$

$$10 \equiv 2 \bmod 8 \quad \text{or equivalently} \quad 2 = 10 \bmod 8$$

$$18 \equiv 2 \bmod 8 \quad \text{or equivalently} \quad 2 = 18 \bmod 8$$

"Another way to think about this is as follows: divide 8 into 2 and you get 0 with a remainder of 2; divide 8 into 10 and you get 1 with a remainder of 2; divide 8 into 18 and you get 2 with a remainder of 2. The remainder is the thing of interest here. If you divide the modulus into the number on either side of the equation, you wind up with the same remainder. From this, we have two ways of saying the same thing:



"Also note that if you subtract the two numbers other than the modulus in the above equations, the modulus will divide that difference evenly. The answer will be an integer and there will be no remainder. (10 - 2)/8 = 1, (2 -10)/8 = -1, (18 - 2)/8 = 2, and so on. This idea is expressed in equations as $8 \mid 18-2;\ m \mid a_1 - a_2$, etc.

"So that's a crash course in modular arithmetic. Now on to the specific equation on which modern cryptography, the so-called RSA algorithm, is based. That equation is called Euler's Totipotent theorem. This is it:

$$m^{\Phi(n)} \equiv 1 \bmod n$$

where,

n tells us how many numbers are on the face of the clock that we're going to use to do modular arithmetic; said differently, n is the number we're going to divide into $m^{\Phi(n)}$ to get a remainder of 1.

$\Phi(n)$, Euler's totient function, is a function that counts the number of relatively prime positive integers less than or equal to n. By positive integers, I mean whole numbers like 1, 2, 3, 4, etc. Relatively prime means that the integer and n share no common factors other than 1. $\Phi(n)$ is a number. In this case we're using it as an exponent. In the equation above, it means to multiply m together with itself $\Phi(n)$ times.

Let me give you some examples that demonstrate the concept of relatively prime.

$\Phi(1)=1$

$\Phi(2)=1$

$\Phi(3)=2$ (i.e. 1,2)

$\Phi(4)=2$ (i.e. 1,3)

$\Phi(5)=4$ (i.e. 1,2,3,4)

$\Phi(6)=2$ (i.e. 1,5)

$\Phi(7)=6$ (i.e. 1,2,3,4,5,6)

$\Phi(8)=4$ (i.e. 1,3,5,7)

$\Phi(9)=6$ (i.e. 1,2,4,5,7,8)

$\Phi(10)=4$ (i.e. 1,3,7,9)

Let's examine $\Phi(9)$ in more detail so you really understand what this means. In order to do this, we have to start by making a list of numbers less than 9 that are relatively prime to 9 (i.e, that share no factor with 9 other than 1):

1 is relatively prime to everything because 1 is the only factor of 1, so include 1.
No number other than 1 divides both 2 and 9 evenly, so include 2 in our list.
3 divides evenly into 3 and 9; that is, 3 and 9 share a factor of 3 so exclude 3.

No number other than 1 divides both 4 and 9 evenly so include 4 in our list.
No number other than 1 divides both 5 and 9 evenly so include 5 in our list.
3 goes into 6 twice and 9 three times; 6 and 9 share a factor of 3 so exclude 6.
No number other than 1 divides 7 and 9 evenly so include 7 in our list.
No number other than 1 divides 8 and 9 evenly so include 8 in our list.
So here's our list: 1, 2, 4, 5, 7, 8. Count the numbers in our list: 6 - that's the value of $\Phi(9)$.

"Notice something. For any prime number n, $\Phi(n)$ is $n-1$. This makes sense since the definition of a prime number is that the only factors it has are 1 and itself. You can't include the number itself in $\Phi(n)$ because any number goes into itself once. For example, for n = 7, 7 goes into 7 once, so when calculating $\Phi(n)$, you have to exclude 7. But you would include all the other numbers from 1 to 6 in enumerating $\Phi(n)$.

"Ok, back to Euler's totient theorem. That's the equation we started with, $m^{\Phi(n)} \equiv 1 \bmod n$. First of all, it only works if m and n are relatively prime to each other. Now let me prove it to you. I'll be abstract and use letters to start, then put in some numbers to make it clearer.

"Let's find a set of numbers that consists of all the positive integers that are relatively prime to n, like we would do if we're trying to figure out Euler's totient function for that number. Notice that when you do this, all of the members of the set, called the set's elements, have to be different from each other (i.e. each number is unique.) Call that set A. A would consist of r_1, r_2, r_3, $r_4 \ldots r_{\Phi(n)}$, where r are numbers relatively prime to n. (In the example of $n=9$, the first element of the set, r_1 is 1 because, as we've already said, 1 is relatively prime to all numbers. The other elements are $r_2=2$, $r_3=3$, $r_4=5$, $r_5=7$, $r_6=8$. $\Phi(n)$ is 6, so $r_{\Phi(n)}=r_6$ — or the $\Phi(n)^{\text{th}}$ (the 6th) relatively prime number in the series (in this case, that relatively prime number is 8.) When you define a set in math, you put the elements of the set in curly brackets. So $A=\{r_1, r_2, r_3, r_4 \ \ldots \ r_{\Phi(n)}\}$. Now we'll define a second set, B, by multiplying each element in A by

the number, m. So $B=\left\{m\cdot r_1, m\cdot r_2, m\cdot r_3, m\cdot r_4 \ \ldots\ m\cdot r_{\Phi(n)}\right\}$. We're just multiplying each element of A by the same number, so like A, all the elements of B have to be unique. Now let's take the $\bmod n$ of sets A and B. That means take $\bmod n$ of each element in A to make a new set, A', and take $\bmod n$ of each element in B to make a new set, B'. It turns out, if we do this, A' and B' will be the same. To see this, let's put in some numbers.

"Let $m=5$, $n=8$. Notice that 5 and 8 are relatively prime. (As I said at the outset, they have to be or this thing won't work.) A different way of putting it is that the greatest common divisor of 5 and 8 is 1. Or, said yet another way, the greatest number that divides both 5 and 8 evenly is 1. Anyway,

$$\begin{aligned} A &= \left\{1, 3, 5, 7\right\} \\ B &= \left\{5\cdot 1, 5\cdot 3, 5\cdot 5, 5\cdot 7\right\} \\ &= \left\{5, 15, 25, 35\right\} \end{aligned}$$

"So for each element, we're going to take $\bmod n$. Mathematically speaking, we can write this as $r_i \bmod n$; i, in this case, being 1, 2, 3 or 4.

For A'	For B'
1 mod 8 = 1	5 mod 8 = 5
3 mod 8 = 3	15 mod 8 = 7
5 mod 8 = 5	25 mod 8 = 1
7 mod 8 = 7	35 mod 8 = 3

"So $A=A'=\left\{1, 3, 5, 7, \right\}$ and $B'=\left\{5, 7, 1, 3, \right\}$; From this, you can see that A' and B' have the same elements.

"Since the elements of the sets are the same, if the elements of each set are multiplied together with each other, the resulting products should be equal:

$$\left(1\cdot 3\cdot 5\cdot 7\right)=\left(5\cdot 7\cdot 1\cdot 3\right)$$

"But remember where these numbers came from:

$$\left[\left(1 \bmod 8\right)\cdot\left(3 \bmod 8\right)\cdot\left(5 \bmod 8\right)\cdot\left(7 \bmod 8\right)\right]=\left[\left(5 \bmod 8\right)\cdot\left(15 \bmod 8\right)\cdot\left(25 \bmod 8\right)\cdot\left(35 \bmod 8\right)\right]$$

"It's easy to prove that $\left[\left(a \bmod n\right)\cdot\left(b \bmod n\right)\right]\bmod n=\left(a\cdot b\right)\bmod n$:

Let a, b, q_1, q_2, q_3, r_1, r_2 and r_3 be integers

By definition

$r_1 = a \bmod n$ means $a = q_1 n + r_1$

$r_2 = b \bmod n$ means $b = q_2 n + r_2$

$r_3 = r_1 \cdot r_2 \bmod n$ means $r_1 \cdot r_2 = q_3 n + r_3$

with $0 \le r_1, r_2, r_3 < n$

We've seen that

$r_1 = a \bmod n$ and $r_2 = b \bmod n$

Therefore,

$r_1 \cdot r_2 = \left(a \bmod n\right)\cdot\left(b \bmod n\right)$

And since

$r_3 = r_1 \cdot r_2 \bmod n$

then

$r_3 = \left[\left(a \bmod n\right)\cdot\left(b \bmod n\right)\right] \bmod n$

From what I've already told you

$$\begin{aligned} a\cdot b &= \left(q_1 n + r_1\right)\left(q_2 n + r_2\right) \\ &= q_1 q_2 n^2 + \left(q_1 r_2 + q_2 r_1\right)n + r_1 r_2 \\ &= q_1 q_2 n^2 + \left(q_1 r_2 + q_2 r_1\right)n + q_3 n + r_3 \\ &= \left(q_1 q_2 n + q_1 r_2 + q_2 r_1 + q_3\right)n + r_3 \end{aligned}$$

The equation $a\cdot b = \left(q_1 q_2 n + q_1 r_2 + q_2 r_1 + q_3\right)n + r_3$ should look familiar

It means that if you divide $a\cdot b$ by n you get a remainder of r_3

Which means $r_3 = \left(a\cdot b\right) \bmod n$

But $r_3 = \left[\left(a \bmod n\right)\cdot\left(b \bmod n\right)\right] \bmod n$

Therefore, $\left[\left(a \bmod n\right)\cdot\left(b \bmod n\right)\right] \bmod n = \left(a\cdot b\right) \bmod n$

Which is what we were trying to prove. Let's check it out by trying some numbers:

$\left[\left(10 \bmod 8\right)\cdot\left(15 \bmod 8\right)\bmod 8\right] = \left(10\cdot 15\right)\bmod 8$

$\left(2\cdot 7\right)\bmod 8 = 150 \bmod 8$

$$14 \bmod 8 = 6 = 150 \bmod 8 .$$

"That is, 14/8 = 1 with remainder 6; 150/8 = 18 with the same remainder: 6.

"By similar arguments to those I just employed, you can generalize this result and show that any number of terms can be multiplied together on each side. Like this:

$$\left[(a \bmod n)\cdot(b \bmod n)\cdot(c \bmod n)\cdot(d \bmod n)\ldots\right]\bmod n = (a\cdot b\cdot c\cdot d\cdot\ldots)\bmod n$$

Then Danny Tenacce pointed the marker he had been using at his father. "So what do you think we ought to do next, Professor?"

"Well, I think the thing ta do is do the circle thing ta both sides o' that equation ya showed us about an hour ago. At least that's what us professors call it—doin' the circle thing. I know you detectives call it somethin' else. Takin' the modulus or some such layman's term."

"You mean this equation?" He went back to the board, did some erasing and re-wrote the following equation.

$$\left[(1 \bmod 8)\cdot(3 \bmod 8)\cdot(5 \bmod 8)\cdot(7 \bmod 8)\right] = \left[(5 \bmod 8)\cdot(15 \bmod 8)\cdot(25 \bmod 8)\cdot(35 \bmod 8)\right]$$

"Yeah, that's the one."

"The circle thing. Okay, I'll give it a try. I'll take mod8 of both sides."

$$\left[(1\bmod 8)\cdot(3\bmod 8)\cdot(5\bmod 8)\cdot(7\bmod 8)\right]\bmod 8 = \left[(5\bmod 8)\cdot(15\bmod 8)\cdot(25\bmod 8)\cdot(35\bmod 8)\right]\bmod 8$$

$$\left[(1\bmod 8)\cdot(3\bmod 8)\cdot(5\bmod 8)\cdot(7\bmod 8)\right]\bmod 8 = \left[(5\cdot 1\bmod 8)\cdot(5\cdot 3\bmod 8)\cdot(5\cdot 5\bmod 8)\cdot(5\cdot 7\bmod 8)\right]\bmod 8$$

$$(1\bmod 8\cdot 3\bmod 8\cdot 5\bmod 8\cdot 7\bmod 8) = \left[(5\cdot 1)\cdot(5\cdot 3)\cdot(5\cdot 5)\cdot(5\cdot 7)\right]\bmod 8$$

"This gives

$$(1\cdot 3\cdot 5\cdot 7) = (5\cdot 1\cdot 5\cdot 3\cdot 5\cdot 5\cdot 5\cdot 7)\bmod 8$$

$$(1\cdot 3\cdot 5\cdot 7) = (5\cdot 5\cdot 5\cdot 5\cdot 1\cdot 3\cdot 5\cdot 7)\bmod 8$$

$$(1\cdot 3\cdot 5\cdot 7) = (5\cdot 5\cdot 5\cdot 5)(1\cdot 3\cdot 5\cdot 7)\bmod 8$$

$$(1\cdot 3\cdot 5\cdot 7) = 5^4(1\cdot 3\cdot 5\cdot 7)\bmod 8$$

$105 = 105(5^4)\bmod 8$; Divide both sides by 105

$1 = 5^4 \bmod 8$

"This means, if you divide 5^4 by 8, you get a remainder of 1. Hmm. That looks a lot like the Euler's Totipotent theorem equation. I guess the circle thing may have worked.

"Generalizing this using the variable names (that is, the letters) we used before, this becomes:

$$1 = m^4 \bmod n$$

"But notice that the exponent 4 is the same as $\Phi(n)$. It has to be because we multiplied m times each element in the set A to get set B and the number of elements in set A *is* $\Phi(n)$. So,

$$1 = m^{\Phi(n)} \bmod n$$

"Real mathematicians would cringe if they were listening to this, but you get the idea. There are a couple of other things I need to tell you about before I can show you how the above equation is used for encryption.

"First, $1^k = 1$ You can put in any number you want for k. $1^3 = 1 \cdot 1 \cdot 1$. $1^5 = 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1$. No matter how many times you multiply one with itself, you still get 1. Let's go back to this equation for a minute: $1 = m^{\Phi(n)} \bmod n$. Both sides of this equation are equal to 1, so if we raise both sides of this equation to the k^{th} power, we don't change the value of either side. When you raise a number that is already raised to an exponent, by another exponent, you multiply the exponents together. For example, $\left(2^2\right)^3 = 4^3 = 4 \cdot 4 \cdot 4 = 64$. Equivalently, $\left(2^2\right)^3 = 2^{2 \cdot 3} = 2^6 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 64$. So $1 = 1^k = \left(m^{\Phi(n)}\right)^k \bmod n$; therefore, $1 = m^{k \cdot \Phi(n)} \bmod n$.

"Next multiply both sides of this equation by m. We get $m \cdot 1 = m \cdot m^{k \cdot \Phi(n)} \bmod n$ which means $m = m \cdot m^{k \cdot \Phi(n)} \bmod n$. Any number (or variable) raised to the 1rst power is just that number or variable. So $m = m^1$. When you multiply a number raised to one exponent by the same number raised to another exponent, you get the number raised to the two exponents added together. Example: $2^2 \cdot 2^3 = 4 \cdot 8 = 32 = 2^{(2+3)} = 2^5$. Therefore, $m = m^1 \cdot m^{k \cdot \Phi(n)} \bmod n$ is equivalent to $m = m^{k \cdot \Phi(n)+1} \bmod n$.

"The number n can be broken down into the product of two prime numbers, P_1 and P_2:

$$n = P_1 \cdot P_2$$

"Take it on faith that $\Phi(n) = \Phi(P_1) \cdot \Phi(P_2)$."

Tenacce snickered. "You're asking' *us* ta have faith?"

"You want me to prove it?"

"Ya proved everything else," Salito pointed out.

"Ok then. The best thing to do is start with an example. Note before starting that for this to work, P_1 and P_2 need to be relatively prime (that is, their greatest common denominator is 1, or equivalently—as I've told you several times now, but let me say it one more time to reinforce it—the only common factor they share is 1). Consider $n = P_1 \cdot P_2 = 21, P_1 = 3, P_2 = 7$.
List the relatively prime numbers that you need to calculate $\Phi(P_1 \cdot P_2)$ and put them into, set S_1:

$$S_1 = \{1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20\}$$

Call the elements in S_1, a_i where $i = 1,2 \ldots \Phi(P_1 \cdot P_2)$.

"Next, list the relatively prime numbers that you need to calculate

$$\Phi(P_1) = \Phi(3) \Rightarrow 1, 2$$
$$\Phi(P_1) = \Phi(7) \Rightarrow 1, 2, 3, 4, 5, 6$$

Put those numbers into a set in which you pair numbers from $\Phi(P_1)$ and $\Phi(P_2)$. Call it S_2:

$$S_2 = \{(1,1), (1,2), (1,3), (1,4), (1,5), (1,6), (2,1), (2,2), (2,3), (2,4), (2,5), (2,6)\}$$

Call the first pair of the elements of S_2, b_j where $j = 1 \ldots \Phi(P_1)$. Call the second pair of the elements of S_2, c_k where $k = 1 \ldots \Phi(P_2)$.

Now we need to count elements in each set. By definition, the number of elements in S_1 is $\Phi(P_1 \cdot P_2)$, in this case, 12. To find the number of elements in S_2, we have to consider how we made the set. Namely, we took each element from $\Phi(P_1)$ and paired it with each element of $\Phi(P_2)$:

		From $\Phi(P_2)$				
From $\Phi(P_1)$	**1**	**2**	**3**	**4**	**5**	**6**
1	1,1	1,2	1,3	1,4	1,5	1,6
2	2,1	2,2	2,3	2,4	2,5	2,6

You can see from this table that the number of elements in the set is just the number of rows times the number of columns: 2 x 6. If the table were a 3 by 5 table, then the number of elements

would be 3 x 5 = 15. In every case, the number of rows is $\Phi(P_1)$ and the number of columns is $\Phi(P_2)$. Therefore, in general, the number of elements in such a set is $\Phi(P_1)\cdot\Phi(P_2)$.

"The next thing we have to do is prove that there is a one-to-one correspondence between elements of sets S_1 and S_2. To do this, we need to associate elements $a_1 \mathrm{mod} P_1 P_2$ from S_1 with paired elements $a_1 \mathrm{mod} P_1, a_1 \mathrm{mod} P_2$ from S_2. It looks like this:

$1 \rightarrow 1\mathrm{mod}3,1\mathrm{mod}7 \rightarrow 1,1$
$2 \rightarrow 2\mathrm{mod}3,2\mathrm{mod}7 \rightarrow 2,2$
$4 \rightarrow 4\mathrm{mod}3,4\mathrm{mod}7 \rightarrow 1,4$
$5 \rightarrow 5\mathrm{mod}3,5\mathrm{mod}7 \rightarrow 2,5$
$8 \rightarrow 8\mathrm{mod}3,8\mathrm{mod}7 \rightarrow 2,1$
$10 \rightarrow 10\mathrm{mod}3,10\mathrm{mod}7 \rightarrow 1,3$
$11 \rightarrow 11\mathrm{mod}3,11\mathrm{mod}7 \rightarrow 2,4$
$13 \rightarrow 13\mathrm{mod}3,13\mathrm{mod}7 \rightarrow 1,6$
$16 \rightarrow 16\mathrm{mod}3,16\mathrm{mod}7 \rightarrow 1,2$
$17 \rightarrow 17\mathrm{mod}3,17\mathrm{mod}7 \rightarrow 2,3$
$19 \rightarrow 19\mathrm{mod}3,19\mathrm{mod}7 \rightarrow 1,5$
$20 \rightarrow 20\mathrm{mod}3,20\mathrm{mod}7 \rightarrow 2,6$

"You can see from the table that, in this case, there is, in fact, a one-to-one correspondence between elements of sets S_1 and S_2. However, to generalize this result, we have to show:

1. Different elements in S_1 are associated with different pairs in S_2
2. Each pair in S_2 is associated with an element in S_1

"To accomplish #1, suppose a_1 and a_2 are different elements of S_1 but are both mapped to the same element in S_2. If this were the case, then, for example:

$$a_1 \Rightarrow a_2 \mathrm{mod} P_1,\ a_2 \mathrm{mod} P_2$$
and
$$a_2 \Rightarrow a_2 \mathrm{mod} P,\ a_2 \mathrm{mod} P_2$$

Let's take the case of $a_1 \Rightarrow a_2 \mathrm{mod} P_1,\ a_2 \mathrm{mod} P_2$. This means

$$a_1 \equiv a_2 \bmod P_1 \text{ and } a_1 \equiv a_2 \bmod P_2$$

Remember,

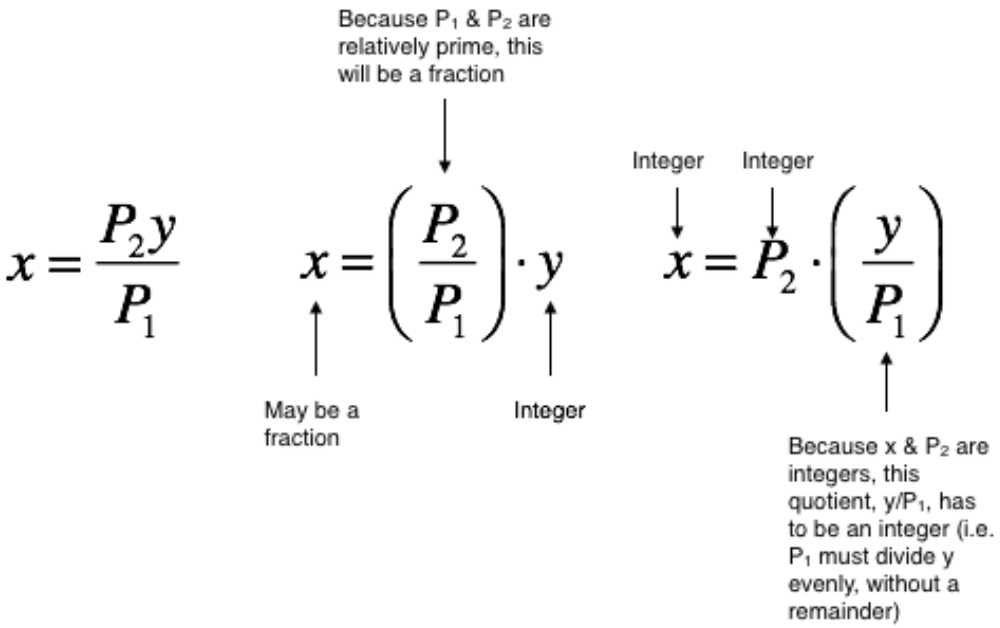
$$a_1 \equiv a_2 \bmod P_1 \text{ means } a_1 = P_1 \cdot x + a_2$$
and
$$a_1 \equiv a_2 \bmod P_2 \text{ means } a_1 = P_2 \cdot y + a_2$$
where
x and y are integers (i.e. not fractions)

Rearranging the right-sided equations:

$$a_1 - a_2 = P_1 x \text{ and } a_1 - a_2 = P_2 y$$
so
$$P_1 x = P_2 y$$
and
$$x = \frac{P_2 y}{P_1}$$

That means that P_1 divides y evenly (i.e., without a remainder). Mathematically, this is expressed as follows: $P_1 \mid y$. Why is this true? Well, P_1, P_2, x and y are all integers. P_1 and P_2 are relatively prime. That means that they share no common factors except 1. If you divide P_1 into P_2, you get a fraction. If you multiply a fraction by an integer, like y, you may make their product, x, a fraction. But we've already said that x is an integer. In order to make x an integer, P_1 would have to divide evenly into y to get an integer. Then when you multiply that integer, y/P_1, with another integer, P_2, their product, x, is sure to be an integer.

"This diagram may help you visualize it better:



So if P_1 divides y evenly, that means that $y = P_1 \cdot q$ where q is a positive integer. We've seen previously that $a_1 = P_2 \cdot y + a_2$. Substituting $y = P_1 \cdot q$, we get $a_1 = (P_1 \cdot P_2)q + a_2$ and $a_1 - a_2 = (P_1 \cdot P_2)q$. But $a_1 - a_2 = (P_1 \cdot P_2)q$ is another way of saying $a_1 \equiv a_2 \bmod P_1 \cdot P_2$, just as $a_1 \equiv a_2 \bmod P_1$ means $a_1 = P_1 \cdot x + a_2$ and $a_1 \equiv a_2 \bmod P_2$ means $a_1 = P_2 \cdot y + a_2$.

"So after all this, we end up with $a_1 \equiv a_2 \bmod P_1 \cdot P_2$. For this equation to be true, a_1 must equal a_2. But this contradicts our original premise that a_1 and a_2 are different. Why must $a_1 = a_2$ in the equation $a_1 \equiv a_2 \bmod P_1 \cdot P_2$? Because all of the elements of set S_1 are relatively prime to, and less than, $P_1 \cdot P_2$. When you take $\bmod P_1 \cdot P_2$ of a_1, you divide $P_1 \cdot P_2$ into a_1. Since every element of set S_1 is less than $P_1 \cdot P_2$, $P_1 \cdot P_2$ goes into every element, a_i, zero times. The only way to make the equation $a_1 \equiv a_2 \bmod P_1 \cdot P_2$ true is if the remainder, a_2, equals a_1. Another way of saying it is $a_1 = P_1 \cdot P_2 \cdot q + a_2$; $q = 0$ so $a_1 = P_1 \cdot P_2 \cdot 0 + a_2 \Rightarrow a_1 = 0 + a_2 \Rightarrow a_1 = a_2$.

"The above argument shows that it's not possible for any two elements of S_1 to map to the same element in S_2 . Therefore, it must be that only one element from S_1 can map to a given element of S_2. So that proves #1.

"Proving #2 is a little trickier," said Danny. He smiled when Tenacce expressed the expected sigh.

"What we need to show," Danny continued, "is that each paired element of S_2, (b,c), maps to a unique element of S_1, a. Mathematically, this can be expressed as follows:

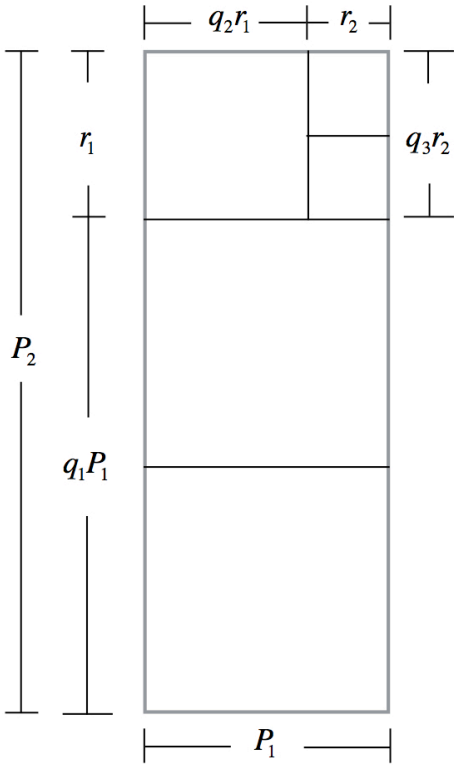
$$a = b \bmod P_1 \text{ and } a = c \bmod P_2$$

"This is essentially the Chinese Remainder Theorem. So now I need to prove that theorem.

"The equation $a = b \bmod P_1$ means that if you multiply P_1 by some integer (call that integer y) then add a remainder of b to it, you get $a : a = P_1 y + b$. Likewise, $a = c \bmod P_2$ means $a = P_2 z + c$ where z is any integer. Since the right side of both of these equations are equal to a:

$$P_1 y + b = P_2 z + c \text{ and}$$
$$P_1 y - P_2 z = c - b \text{ where } c \text{ and } b \text{ are just numbers}$$

"This is called Bezout's theorem and can be proved by reversing the Extended Euclidean Algorithm which finds the greatest common divisor for two integers (call them P_1 and P_2). This is how it works:



(1) $P_2 = q_1 P_1 + r_1$
(2) $P_1 = q_2 r_1 + r_2$
(3) $r_1 = q_3 r_2 + r_3$

(4) $r_3 = r_1 - q_3 r_2$
(5) $r_2 = P_1 - q_2 r_1$
(6) $r_1 = P_2 - q_1 P_1$

(7) $r_2 = P_1 - q_2 r_1$
(8) $r_2 = P_1 - q_2 (P_2 - q_1 P_1)$
(9) $r_2 = (1 + q_1 q_2) P_1 - q_2 P_2$

"Consider a rectangle with sides P_1 and P_2. The area of the rectangle is $P_1 \cdot P_2$. Fill the rectangle in the upward vertical direction with $P_1 \times P_1$ square boxes until the remaining height of the rectangle, r_1, is less than P_2. This is depicted mathematically by equation (1) where the length of the height of the original rectangle, P_2, is equal to the number of squares, q_1, times the length of a side of the square, P_1, plus a remainder, r_1. (You can see where this comes from by looking at the left-hand side of the rectangle in the diagram.). That leaves a rectangle within the top of the original rectangle which is smaller than the original rectangle and has dimensions $r_1 \times P_1$.

"Fill that smaller upper rectangle horizontally to the right with q_2 $r_1 \times r_1$ squares until an even smaller rectangle of dimensions $r_1 \times r_2$ remains within the upper right-hand portion of the original rectangle. Mathematically (check out the upper and lower sides of the original rectangle) this is represented by equation (2) which states that the horizontal length of the original rectangle, P_1, equals the number of squares used to fill the smaller upper rectangle, q_2, times the length of each of these squares, r_1, plus a second remainder, r_2.

"Next fill the even smaller $r_1 \times r_2$ rectangle in the right upper corner in the downward direction with q_3 squares of dimension $r_2 \times r_2$ until an even smaller still rectangle remains at the bottom of dimensions $r_2 \times r_3$. In this example, $r_3 = 0$ so there is no additional tiny rectangle within the bottom of the $r_1 \times r_2$ rectangle. Obviously, this process could go on for many more further steps, depending on how big the numbers you start with are., However, once a remainder of zero is reached, the procedure is terminated and the last nonzero remainder *is* the greatest common divisor. In this case, it's r_2."

"How do ya know that r_2 is this greatest common divisor thing?" intercede Salito quickly.

"You don't. Yet. But you will. Because I'm going to prove it right now."

Danny spoke as he scratched equations onto the whiteboard. "I need to start by showing that the last remainder that the algorithm produces is, in fact, a divisor of a and b. To do this, let's start with a general statement of the Extended Euclid's Algorithm:

$$P_1 = q_1 P_2 + r_1 \qquad \text{where } 0 < r_1 < P_2$$

$$P_2 = q_2 r_1 + r_2 \qquad \text{where } 0 < r_2 < r_1$$

$r_1 = q_3 r_2 + r_3$ where $0 < r_3 < r_2$

$\vdots$

$r_i = q_{i+2} r_{i+1} + r_{i+2}$ where $0 < r_{i+2} < r_{i+1}$

$\vdots$

The final equation that we have to solve, to get what we said was the greatest common divisor, looks like this (notice that this equation has no remainder):

$$r_{k-1} = q_{k+1} r_k$$

So we're saying that r_k is the greatest common divisor. We need to show that it is a divisor at all. Rearranging the last equation, we get:

$$q_{k+1} = \frac{r_{k-1}}{r_k}$$

q_{k+1} is an integer. That means r_k divides r_{k-1} evenly which means that r_k is a divisor of r_{k-1}. Mathematically, this is written as $r_k \mid r_{k+1}$. This means that $r_{k+1} = cr_k$ where c is an integer. Now consider the equation just previous to the last equation. It can be written like so:

$r_{k-2} = q_k r_{k-1} + r_k$	substituting cr_k for r_{k-1}
$r_{k-2} = q_k c r_k + r_k$	factoring out r_k
$r_{k-2} = r_k (q_k c + 1)$	dividing thru by r_k
$\frac{r_{k-2}}{r_k} = (q_k c + 1)$	q_k, c and 1 are integers; therefore
$\frac{r_{k-2}}{r_k}$	is an integer

That means that r_k is a divisor of r_{k+2}.

"We can continue this process, working backward through the Euclidean algorithm until we get to the first two equations, where we will find that $r_k \mid P_1$ and $r_k \mid P_2$. Thus, we have shown that r_k is a common divisor of P_1 and P_2. Now we have to show that it is the *greatest* common divisor of P_1 and P_2.

"The proof for this is going to look similar to the previous one, but it is different in that we will start by assuming that P_1 and P_2 have a common factor d, and then show that $d \mid r_k$.

"Consider an arbitrary common factor, d, of P_1 and P_2. If d is a common factor, we can rewrite P_1 and P_2 as follows:

$P_1 = dP_1'$ and $P_2 = dP_2'$ where $d, P_1', \text{and } P_2'$ are all positive integers

"Now, consider the first equation from Euclid's algorithm:

$P_1 = q_1 P_2 + r_1$ substitute P_1 and P_2 and solve for r_1

$r_1 = dP_1' - q_1 dP_2'$

$r_1 = d\left(P_1' - q_1 P_2'\right)$

$\frac{r_1}{d} = P_1' - q_1 P_2'$

q_1, P_1' and P_2' are all integers. Therefore, $d \mid r_1$.

"Now, consider the second equation, and repeat the steps we did on the first, this time solving for r_2.

$P_2 = q_2 r_1 + r_2$

Let $r_1 = dr_1'$ where r_1' is an integer; then

$P_2 = q_2 dr_1' + r_2$

$P_2 = dP_2' - q_2 dr_1'$

$r_2 = d\left(P_2' - q_2 r_1'\right)$

$\frac{r_2}{d} = P_2' - q_2 r_1'$

q_2, r_1 and P_2' are all integers. Therefore, $d \mid r_2$.

"As you can see, we can continue this process through each of the equations until we get to the second to last one, where we will have:

$$r_{k-2} = q_k r_{k-1} + r_k$$
$$r_k = dr'_{k-2} - q_k dr'_{k-1} = d\left(r'_{k-2} - q_k r'_{k-1}\right); \text{ thus}$$
$$d \mid r_k$$

"But this says that any arbitrary common factor of P_1 and P_2 that we originally picked divides into r_k, the value that Euclid's algorithm produced. We've already shown that r_k is a common factor of P_1 and P_2. If all of these arbitrary common factors (which we've called d) divide r_k, then all of these arbitrary common factors must be less than or equal to r_k. Why? Because if d were larger than r_k, then $\frac{r_k}{d}$ would be a fraction. But that can't be so because we've already proved that $d \mid r_k$ (i.e., $\frac{r_k}{d}$ produces an integer). So r_k must be the largest possible common factor (i.e., it must be the greatest common divisor).

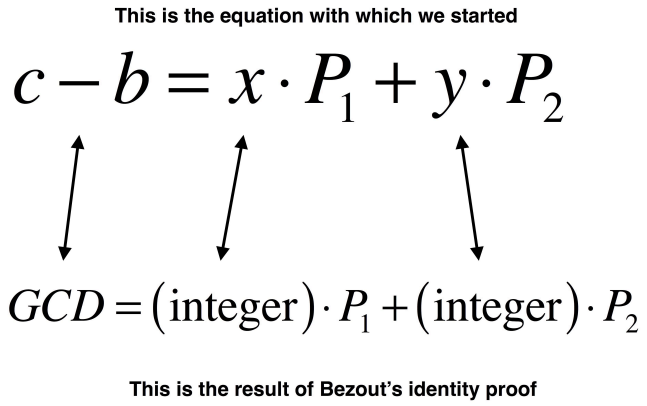
"So that's proof that the Extended Euclidean Algorithm produces the greatest common divisor of P_1 and P_2. But as I said previously, to prove Bezout's identity, we need to start with the greatest common divisor, r_2 in our example, and work backwards to the original numbers (here P_1 and P_2).

"To do that, we take equations 1-3 and rearrange them, as shown in equations 4-6, to find remainders r_1, r_2 and r_3. What we ultimately want to end up with is an equation like this:

$$\text{Greatest Common Divisor} = (\text{integer}) \cdot P_1 + (\text{integer}) \cdot P_2$$

"We've already said that r_2 is the greatest common divisor so we start with equation 7. Substitute the value of r_1 (from equation 6) into equation 7 to get equation 8. Rearrange equation 8 and we get equation 9. And there we have it; equation 9 is of exactly the form we want: r_2 is the greatest common divisor, $1 + q_1 q_2$ is an integer and so is q_2.

"Here, this diagram may help you to see it:



"So our equation $c - b = x \cdot P_1 + y \cdot P_2$ is a statement of Bezout's Identity which we have shown to be true. That equation was derived from $a = b \bmod P_1$ and $a = c \bmod P_2$ which is a statement of the fact that, for each pair 'b,c' in S_2, a corresponding element 'a' in set S_1 *exists*. What remains to be proven is that those elements, 'a', to which 'b,c' pairs map, are *unique* (i.e., there's only one 'a' for each 'b,c').

"That proof is similar to one we've already seen. It goes like this: if $a = n$ and $a = n'$ both satisfy

$$a = b \bmod \mathrm{P}_1 \text{ and } a = c \bmod \mathrm{P}_2$$

then $n \equiv n' \bmod \mathrm{P}_1$ and $n \equiv n' \bmod \mathrm{P}_2$. If that's true, then $P_1 \left| (n - n') \right.$ and $P_2 \left| (n - n') \right.$. And because P_1 and P_2 are relatively prime, $P_1 \cdot P_2 \left| (n - n') \right.$. That means $n \equiv n' \bmod P_1 \cdot P_2$, which means that n and n' are the same modulo $P_1 \cdot P_2$ (i.e., n and n' correspond to the same element of S_1 which means that each pair 'b,c' in set S_2 maps to only one element of S_1).

"So we've finally seen why $\Phi(n) = \Phi(P_1) \cdot \Phi(P_2)$. Now what I need to do is show you how this fact can be used in RSA encryption.

"Recall that $\Phi(P)$ of any prime number is $P - 1$ (i.e., one less than the number). So,

$\Phi(P_1) = P_1 - 1$ and $\Phi(P_2) = P_2 - 1$; then,

$$\Phi(n) = \Phi(P_1) \cdot \Phi(P_2) = (P_1 - 1)(P_2 - 1)$$

Next, define the product $e \cdot d$ such that $e \cdot d = k \cdot \Phi(n) + 1$

Remember the equation $m = m^{k \cdot \Phi(n)+1} \bmod n$? Substitute $e \cdot d$ for $k \cdot \Phi(n) + 1$. We get

$$m = m^{e \cdot d} \bmod n$$

From $e \cdot d = k \cdot \Phi(n) + 1$, we find that $d = \dfrac{k \cdot \Phi(n) + 1}{e} = \dfrac{k \cdot (P_1 - 1) \cdot (P_2 - 1) + 1}{e}$

"We picked P_1 and P_2. That's how we came up with n. We also chose k. So we have all the information needed to determine d. d, it turns out, is the secret key needed to decode a message sent to us. Let me show you how it all works.

"If we were using computers, in the modern era, and Danny wanted to receive a message from Mary, he would send his public key, which consists of the numbers e and n, to Mary. John and anyone else who wants it has access to that key. Since computers only understands numbers, Mary's message consists of a string of numbers. Each number stands for a character like you can type on a keyboard. There is a standardized system of translation from characters to numbers that computers use, called ASCII, which stands for American Standard Code for Information Interchange. Let's say Mary's message is just the letter J. The ASCII code for J is 74. 74 would be the value of m. Mary uses the values of m, e and n to generate an encrypted message, let's call it c, according to the following equation:

$$c = m^e \bmod n$$

"Mary sends the encrypted message, c, to Danny. The actual message is a long string of characters which translates into a number with a long string of digits. John can see the encrypted message but can't decrypt it."

"Why can't he decrypt it?" Salito was animated as she spoke. "John knows c, e and n. He oughtta be able to figure out m, shouldn't he?"

"No, because we're using modular arithmetic. Remember,

$$c = m^e \bmod n \text{ means } m^e = nq + c \text{ where } q \text{ is some integer}$$

therefore,

$$m = \sqrt[e]{n \cdot q + c}$$

As you said, John knows c, e and n—they're just numbers that anyone can see. However, he has no idea what m and q are. What you have, then, is one equation with two unknowns. There is no unique solution to such an equation; m and q, literally, could be anything. I suppose John could try putting in values for m and q by trial and error, then check to see if the message encoded by the string of numbers that is m, makes sense. However, because P_1 and P_2 are so large, the range of values that q could be would astronomical. The time it would take to test values of q and find the correct answer would be so long as to make this method impractical, on a par with trying to factor n and guess the private key.

Danny, however, has the secret key, d, which he can use to recover the message, m, by using the equation

$$m = m^{e \cdot d} \bmod n$$

"John knows e and could decrypt the message if he could factor n since the factors of n are P_1 and P_2, and P_1 and P_2 determine d, but we've already seen how long factoring n could take if P_1 and P_2 are large."

"So where are we in all this?" asked Salito.

"The papyrus contains a bunch of encrypted text and two numbers. I assume those numbers are n and e. Of course, that's assuming we're dealing with RSA encryption. There are other algorithms out there."

"That's a lot of assumin'. You know what that gets ya," said Tenacce.

"I think those are pretty good assumptions," said McCleary.

"Does tradition tell you that?" Danny Tenacce asked with some bite.

"Yes."

Salito regarded the equations on the whiteboard then at the one who had written them. "So it looks like we're in about the same position as John."

"That's about right."

"I'll pray for you," said McCleary.

"You said that already. It hasn't helped much," said Danny Tenacce.

"Consider the words of the poem again, then."

But Danny did not answer. Instead, he capped his marker, returned to the desk and began work on a new algorithm.