

COMS10003 Lecture 9.

Julian Gough 2014-11-24

Preface

These are outline notes for lecture 9; they are based on *Number Theory with Computer Applications* by Ramanujachary Kumanduri and Cristina Romero. This is an excellent book, but the material can be found in many number theory and discrete mathematics books. A manicule (☞ or ☞) is used to indicate that a proof, derivation or piece of material has been omitted from the lecture but will be covered in the workshop.

Fermat's theorem and Euler's theorem.

Today we are going to concentrate on two important theorems in number theory, Fermat's Little Theorem and Euler's theorem.

Fermat's Little Theorem. Let p be a prime. Then $a^p \equiv a \pmod{p}$. In particular, if $p \nmid a$ then $a^{p-1} \equiv 1 \pmod{p}$.

The proof of Fermat's Little Theorem isn't hard, but doing it would involve a detour into either the binomial theorem or complete residue systems, so we will skip it. Note the point of the 'in particular', if $p|a$ then both sides are zero modulo p , if $p \nmid a$ then $(a, p) = 1$ so a has an inverse and multiplying both sides by the inverse give $a^{p-1} \equiv 1 \pmod{p}$, it also means $a^{-1} \equiv a^{p-2} \pmod{p}$.

Let's look at a couple of examples. Let $p = 13$ and $a = 6$:

$$6^{12} = 2176782336 \tag{1}$$

and

$$2176782336 - 1 = 2176782335 = 167444795 \cdot 13 \tag{2}$$

so $6^{12} \equiv 1 \pmod{13}$ as the theorem would suggest.

Fermat's Little Theorem can be used to simplify quite fierce looking modular calculations. Consider for example the problem of showing 13 divides $2^{50} + 3^{50}$. Since 13 is prime we can use Fermat's Little Theorem so we know $2^{12} \equiv 1 \pmod{13}$ and $3^{12} \equiv 1 \pmod{13}$ to reduce the original expression to $2^2 + 3^2$ but $2^2 + 3^2 = 13$ which is zero modulo 13.

Euler's theorem is a generalization of Fermat's Little Theorem that applies to modular arithmetic for m where m need not be a prime; it says

Euler's Theorem. If a and m are integers such that $(a, m) = 1$ then

$$a^{\phi(m)} \equiv 1 \pmod{m} \tag{3}$$

Here $\phi(m)$ is the *Euler totient function* which counts the number of numbers co-prime with m and less than it, we met it before in the workshop. For a prime p the totient is $\phi(p) = p - 1$ because, if p is prime $(p, a) = 1$ for all $a < p$. Hence, if m is prime Euler's theorem reduces to Fermat's Little Theorem.

Lets do an example, $\phi(12) = 4$, because the numbers relatively prime to twelve are one, five, seven and eleven. Alternatively, from the workshop $12 = 2^2 \cdot 3$ so

$$\phi(12) = 12 \left(1 - \frac{1}{2}\right) \left(1 - \frac{1}{3}\right) = 4 \quad (4)$$

This uses the formula

$$a = \prod p_i^{r_i} \quad (5)$$

then

$$\phi(a) = a \prod \left(1 - \frac{1}{p_i}\right) \quad (6)$$

Now $\phi(12) = 4$ means that, for example

$$7^4 \equiv 1 \pmod{12} \quad (7)$$

In fact $7^4 = 2401 = 12 \cdot 200 + 1$.

Cryptography

Cryptography is the science of converting a message into another form so that it can't be read by someone for whom it isn't intended. We saw some simple ciphers in workshop, here we will look at some more complicated ciphers which exploit the power of number theory. These ciphers will work with numbers rather than strings, so the letters or characters must first be converted to a number. This is easy, a simple scheme would leave out the spaces and other punctuation marks and number the letters from zero to 25, more sophisticated schemes would use ASCII or a specific encoding that would use letter frequency to keep the code text as short as possible. The cipher will also work with numbers of a specific length, so the original text is broken up into suitable chunks. If the chunks aren't all the same size a few random letters can always be added at the end.

Pohlig-Hellman exponentiation cipher

This cipher was invented in 1976 and has many similarities with RSA, the public key scheme we will look at; however, it isn't a public key scheme, both parties, traditionally called Alice and Bob, must both have the secret key, an odd prime p and a number e which is co-prime with $p - 1$. Now, let m be the message and m_i a block of the message so that $0 < m_i < p$. As mentioned above, let p be a prime and e be an integer $0 < e < p$ with $(e, p - 1) = 1$ then the encoded message is

$$c(m_i) = m_i^e \pmod{p} \quad (8)$$

Now if $d \equiv e^{-1} \pmod{p-1}$, we know e has an inverse because $(e, p-1) = 1$, then $ed \equiv 1 \pmod{p-1}$ Hence $ed = (p-1)k + 1$ for some k . Now, by Fermat's Little Theorem

$$[c(m)]^d = m^{ed} \equiv (m^{p-1})^k m \equiv m \pmod{p} \quad (9)$$

Hence, to encode a message block m_i it is raised to the e th power modulo p , to decode, it is raised to the d th power.

Lets try an example, say the message is 'rosebud', and the secret keys are $p = 10007$ and $e = 5$, which works because $(10006, 5) = 1$. Now, using the simple conversion approach to turn 'rosebud' into a number. So, $r \rightarrow 17$, $o \rightarrow 14$ and so on to give

$$m = 17141804012003 \quad (10)$$

Breaking this up into chunks less than 10000 gives

$$\begin{aligned} m_1 &= 1714 \\ m_2 &= 1804 \\ m_3 &= 0120 \\ m_4 &= 0305 \end{aligned} \quad (11)$$

where it turns out we are splitting m into four digit chunks, that is two letter, chunks and there is an odd number of letters, so we added a random letter, 'f', at the end. Now

$$\begin{aligned} m_1^5 &\equiv 1122 \pmod{10007} \\ m_2^5 &\equiv 6853 \pmod{10007} \\ m_3^5 &\equiv 3947 \pmod{10007} \\ m_4^5 &\equiv 8606 \pmod{10007} \end{aligned} \quad (12)$$

so the cipher text is

$$c(17141804012003) = 1122685339478606 \quad (13)$$

and Alice can send this to Bob.

Now when Bob receives this, he knows $p = 10007$ and $e = 5$, he can also work out 5^{-1} modulo 10006 using Euclid

$$10006 = 5 \cdot 2001 + 1 \quad (14)$$

so $1 = 10006 - 5 \cdot 2001$ and $5^{-1} \equiv -2001 \equiv 8005 \pmod{10006}$ Now

$$\begin{aligned} 1122^{8005} &\equiv 1714 \pmod{10007} \\ 6853^{8005} &\equiv 1804 \pmod{10007} \\ 3947^{8005} &\equiv 0120 \pmod{10007} \\ 8606^{8005} &\equiv 0305 \pmod{10007} \end{aligned} \quad (15)$$

The website includes the short program that was used to do this calculation.

Public key encryption and RSA

The Pohlig-Hellman exponentiation cipher is a secret key cipher; if Alice wants Bob to send her an encrypted message she must first securely send Bob a key. Not only that, but if she also wants Chuck to send her an encrypted message, she needs to send him a different key, if she uses the same key then Chuck could eavesdrop on Bob's message to Alice. This creates a whole morass of key passing and management, this might have been possible when cryptography was the domain of spies and dark deeds, but not now that it is a routine part of our everyday interactions.

Public key cryptography solves this, it allows Alice to share a key, a *public key* with Bob that allows Bob to encrypt a message but does not allow the message to be decrypted, that requires another key, the *private key*. Thus, anyone can see the public key, this is only useful for encrypting, but only the person receiving the messages needs the private key, the key required for decrypting. This seems impossible, that anyone can encrypt but only one person can decrypt, but it works by replacing Fermat's Little Theorem in the Pohlig-Hellman scheme with Euler's theorem.

In the Pohlig-Hellman scheme $c(m) \equiv m^e \pmod{p}$ and then

$$m_i \equiv [c(m_i)]^d \equiv m_i^{ed} \equiv m_i^{(p-1)k} m_i \pmod{p} \quad (16)$$

where Fermat's Little Theorem is used to give

$$m_i^{p-1} \equiv 1 \pmod{p} \quad (17)$$

and $d \equiv e^{-1} \pmod{p-1}$.

The idea is to use Euler's theorem instead, so p is replaced by a number that is not a prime, in fact, it is replaced by $n = pq$ where p and q are primes; because n has a simple form we know $\phi(n)$, it is

$$\phi(n) = (p-1)(q-1) \quad (18)$$

When I say we know $\phi(n)$ I mean we know it because we know p and q , but finding $\phi(n)$ is as hard as factorizing n . If p and q are large this is hard, so if we know p and q we know $\phi(n)$, if we don't, then finding $\phi(n)$ is computationally a big challenge. Now n is the public key along with some e chosen so that

$$(\phi(n), e) = 1 \quad (19)$$

These are used to encrypt the message using exponentiation

$$c(m_i) = m_i^e \pmod{n} \quad (20)$$

Now, to decrypt we proceed much as before but using $\phi(n)$ rather than $p-1$. Hence $d \equiv e^{-1} \pmod{\phi(n)}$ so $de = \phi(n)k + 1$ for some integer k and

$$[c(m_i)]^d = m_i^{ed} \equiv \left(m_i^{\phi(n)}\right)^k m_i \equiv m_i \pmod{n} \quad (21)$$

where we have used Euler's theorem to set $m_i^{\phi(n)} \equiv 1 \pmod{n}$.

Thus, say Alice wants Bob to sent her an encrypted message. She publicly announces n and e , Bob forms $c(m_i) \equiv m_i^e \pmod{n}$, however, even if Eve the eavesdropper intercepts the message, she can't find m_i since she doesn't know $\phi(n)$. However, Alice formed n by first choosing p and q so she can calculate $\phi(n)$ and $d = e^{-1}$, this allows her to find $m_i \equiv [c(m_i)]^d \pmod{n}$.

Let's do an example. consider the public key

$$n = 10001 = 73 \cdot 137 \quad (22)$$

Obviously this is an artificial example, 10001 isn't that hard to factorize, these days n is usually 2048 bits long; anything shorter is considered insecure against a potential eavesdropper factorizing n . We know $\phi(n) = 72 \cdot 136 = 9792$ and it turns out $(9792, 5) = 1$ so we can take $e = 5$ as before. Hence the public key is $n = 10001$ and $e = 5$. Now, lets say the message is 'rosebud' as before. We have

$$\begin{aligned} m_1 &= 1714 \\ m_2 &= 1804 \\ m_3 &= 0120 \\ m_4 &= 0305 \end{aligned} \quad (23)$$

and

$$\begin{aligned} m_1^5 &\equiv 7427 \pmod{10001} \\ m_2^5 &\equiv 2001 \pmod{10001} \\ m_3^5 &\equiv 1929 \pmod{10001} \\ m_4^5 &\equiv 0672 \pmod{10001} \end{aligned} \quad (24)$$

and the encrypted message is

$$c(m) = 7427200119290672 \quad (25)$$

Now, $\phi(10001) = 9792$ so let's apply the Euclid algorithm to $1 = (5, 9792)$

$$\begin{aligned} 9792 &= 5 \cdot 1958 + 2 \\ 5 &= 2 \cdot 2 + 1 \end{aligned} \quad (26)$$

so $1 = 5 - 2 \cdot 2 = 5 - 2(9792 - 1958 \cdot 5) = 3917 \cdot 5 - 2 \cdot 9792$ or

$$d \equiv 5^{-1} \equiv 3917 \pmod{9792} \quad (27)$$

and, using the program again

$$\begin{aligned} 7427^{3917} &\equiv 1714 \pmod{10001} \\ 2001^{3917} &\equiv 1804 \pmod{10001} \\ 1929^{3917} &\equiv 0120 \pmod{10001} \end{aligned}$$

$$0672^{3917} \equiv 0305 \pmod{10001} \quad (28)$$

Of course, we haven't fully examined the RSA scheme, we haven't checked that knowing n and e and being able to encrypt as many messages as is computationally feasible doesn't allow us to spot a pattern which might reveal m_i , we haven't checked that there isn't a way of finding m_i from our knowledge of m_i^e without factorizing n , nor have we quantified how hard factorizing n is and we haven't checked that finding $\phi(n)$ is really as hard as factorizing n . This is all part of the study of cryptography and goes beyond the scope of this course.