Lecture 13: the theorems of Fermat, Euler and Wilson

In this lecture, we shall bring together the number theory and group theory to prove some theorems which are of the utmost importance in cryptography. This will bring to an end our algebraic detour. The most important theorem is Fermat's Little Theorem which is the basis of the RSA cryptosystem. I have included Wilson's theorem because it leads to an interesting characterization of prime numbers.

Fermat's little theorem

This theorem is so-called to distinguish it from the more famous *Fermat's Last Theorem.*

Theorem 0.1 (Fermat)**.** *Let p be a prime and let a be any integer not divisible by p. Then*

$$a^{p-1} \equiv 1 \pmod{p}.$$

Proof. Let p be a prime and let a be any integer such that p does not divide a. Then $\gcd(a, p) = 1$ and so $[a]$ is invertible in $\mathbf{Z}_p$. The order of the group $\mathbf{Z}_p$ is $p - 1$. Thus by the corollary to Lagrange's theorem we have that

$$[a]^{p-1} = [1].$$

It follows that $a^{p-1} \equiv 1 \pmod{p}$ as claimed. □

There is a variation of the above theorem that is also useful.

Corollary 0.2. *Let p be a prime and let a be any integer. Then $a^p \equiv a \pmod{p}$. In other words, if p is a prime then $p \mid a^p - a$ for any integer a.*

Proof. If p does not divide a then by Fermat's Little Theorem we have that $a^{p-1} \equiv 1 \pmod{p}$. Multiply both sides by a to get $a^p \equiv a$. If p does divide a then $a \equiv 0$ and the equation $a^p \equiv a \pmod{p}$ is trivially true. □

Euler's theorem

This can be viewed as a direct generalization of Fermat's little theorem.

Theorem 0.3 (Euler)**.** *Let $\gcd(a, n) = 1$. Then*

$$a^{\phi(n)} \equiv 1 \pmod{n}.$$

Proof. We are given that $\gcd(a, n) = 1$ and so $[a]$ is invertible in $\mathbf{Z}_n$. The order of the group $\mathbf{Z}_n$ is $\phi(n)$. Thus by the corollary to Lagrange's theorem we have that

$$[a]^{\phi(n)} = [1].$$

It follows that $a^{\phi(n)} \equiv 1 \pmod{n}$ as claimed. □

If $n = p$ a prime in the above theorem then $\phi(p) = p - 1$ and we are back to Fermat's Little Theorem.

Wilson's theorem

Another way of phrasing the following theorem is to say that for every prime p we have that

$$p \mid (p-1)! + 1.$$

Theorem 0.4 (Wilson)**.** *Let p be a prime number. Then*

$$(p-1)! \equiv -1 \pmod{p}.$$

Proof. In a group, each element has a unique inverse. Sometimes, an element may be its own inverse: the identity element is such an element but there may be others. We shall begin by finding the elements that are their own inverse in the group $\mathbf{U}_p$. Such an element has the property that $1 \leq x \leq p-1$ and $[x]^2 = [1]$. Thus $x^2 \equiv 1 \pmod{p}$. This means that $p \mid (x+1)(x-1)$. Thus either $x \equiv 1$ or $x \equiv -1$ and so $x \equiv 1$ or $x \equiv p-1$. Thus there are only two such elements which are self inverse. Multiply all the elements of the group $\mathbb{U}_p$ together. Each element that has an inverse different from itself will multiply that inverse to get the identity. We are therefore left with $p-1$ being paired. Thus $(p-1)! \equiv p-1 \pmod{p}$. □

We shall not use this result in what follows but it does have one consequence that is interesting. It is a characterization of when a number is prime that does not require us to look for factors of that number. Sadly, it is not a practical characterization because it involves calculating factorials.

Theorem 0.5. *Let $n \geq 2$ be any natural number. Then n is prime if and only if $n \mid (n-1)! + 1$.*

Proof. If n is a prime then the claim follows by Wilson's theorem. We shall prove that if n is not prime — that is composite — then n cannot divide $(n-1)! + 1$. We deal with the case $n = 4$ separately. We calculate $3! + 1 = 7$ and clearly 4 doesn't divide 7. Let n be any composite number not equal to 4. We may write $n = pq$ where $p, q \neq 1$. If $p \neq q$ then both occur as factors in $(n-1)!$ and so $n \mid (n-1)!$. If $p = q$ then $n = p^2$. If $p > 2$, which we are assuming, then $n > 2p$, and so both p and $2p$ occur as factors in $(n-1)!$. Thus once again $n \mid (n-1)!$. □

Here is an example of the above result. Take $p = 13$. Then $12! = 479001600$. And so $12! + 1 = 479001601$. This number is exactly divisible by 13 and so we deduce that 13 is prime. This is clearly not a practical test but it is very surprising that we can determine whether a number is prime or not by looking at a number that it divides into rather than by trying to factorize it.

The method of repeated squaring

Fermat's Little Theorem requires us to look at large powers of numbers modulo a number. This can be accomplished very quickly using the method of repeated squaring. This method requires you to be able to compute the binary representation of any natural number. I have reminded you how this is done below.

I shall describe the method by means of an example. Let's suppose that we wish to calculate $38^{75} \pmod{103}$. This looks like a lot of work but we can actually speed up this calulcation considerably as follows.

First of all we compute the binary representation of the power we are interested in; here this is the number 75. In binary this number is 1001011. This tells us that

$$75 = 2^6 + 2^3 + 2^1 + 2^0 = 64 + 8 + 2 + 1.$$

Thus

$$38^{75} = 38^{64+8+2+1} = 38^{64} \cdot 38^8 \cdot 38^2 \cdot 38^1.$$

We now set up the following table:

$*38^2$	2
38^4	4
$*38^8$	16
38^{16}	50
38^{32}	28
$*38^{64}$	63

The righthand column is always the reduction modulo 103 of the first column. Each row is the square of the preceding row, apart from the first, of course. I have put an asterisk in some of the rows. These enable us to compute $38^{75} \pmod{103}$ as follows:

$$\begin{aligned} 38^{75} &\equiv 38^{64} \cdot 38^8 \cdot 38^2 \cdot 38 \\ &\equiv 63 \cdot 16 \cdot 2 \cdot 38 \\ &\equiv 81 \cdot 76 \\ &\equiv 79 \end{aligned}$$

The Fermat test for primes

Let n be an odd integer that we think might be prime. If it were prime then Fermat's Little Theorem implies that

$$2^{n-1} \equiv 1 \pmod{n}.$$

It follows that if this does not hold then in fact n is composite. Suppose on the other hand it does hold — is n necessarily prime? The answer is no in general. A composite number that satisfies the above congruence is called a *base 2 pseudoprime*. It turns out, however, that base-2 pseudoprimes are very thin on the ground. It can be shown that a sufficiently large random numbers n that satisies the above congruence is highly likely to be prime. This topic is one that I would like to say more about but sadly this year we don't have the time.

Revision of number bases

The simplest way of writing a number down is to use a mark like |, called a *tally*, for each thing being counted. So

$$||||||||||$$

means 10 things. This system has advantages and disadvantages. The advantage is that you don't have to go on a training course to learn it. The disadvantage is that even quite small numbers need a lot of space like

$$||||||||||||||||||||||||||||||||||||||.$$

It's also hard to tell whether

$$|||||||||||||||||||||||||||||||||||||||$$

is the same number or not. (It's not.)

It's inevitable that people will introduce abbreviations to make the system easier to use. Perhaps it was in this way that the next development occurred. Both the ancient Egyptians and Romans used similar systems but I'll describe the Roman system because it involves letters rather than pictures. First, you have a list of basic symbols:

number	1	5	10	50	100	500	1000
symbol	I	V	X	L	C	D	M

There are more symbols for bigger numbers but we won't worry about them. Numbers are then written according to the *additive principle*. Thus MMVIIII is 2009. Incidently, I understand that the custom of also using a *subtractive principle* so that, for example, IX means 9 rather than using VIIII, is a more modern innovation.

This system is clearly a great improvement on the tally-system. Even quite big numbers are written compactly and it is easy to compare numbers. On the other hand, there is a bit more to learn. The other disadvantage is that we need separate symbols for different powers of 10 and their multiples by 5. This was probably not too inconvenient in the ancient world where it is likely that the numbers needed on a day-to-day basis were never going to be that big.

The system used throughout the world today, called the *Hindu-Arabic number system*, seems to have been in place by the ninth century in India but it was hundreds of years in development and the result of ideas from many different cultures; the invention of zero on its own is one of the great steps in human intellectual development. The genius of the system is that it requires only 10 symbols

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9$$

and every natural number can be written using a sequence of these symbols. The trick to making the system work is that we use the *position* on the page of a symbol to tell us what number it means. Thus 2009 means

10^3	10^2	10^1	10^0
2	0	0	9

In other words

$$2 \times 10^3 + 0 \times 10^2 + 0 \times 10^1 + 9 \times 10^0.$$

Notice the important role played by the symbol 0 which makes it clear which column a symbol belongs in otherwise we couldn't tell 29 from 209 from 2009. The disadvantage of this system is that you *do* have to go on a course to learn it because it is a highly sophisticated way of writing numbers. On the other hand, it has the enormous advantage that any number can be written down in a compact way.

Once the basic system had been accepted it could be adapted to deal not only with positive whole numbers but also negative whole numbers, using the symbol $-$, and also fractions with the introduction of the decimal point. By the end of the sixteenth century the full decimal system was in place.

We shall now look in more detail at the way in which numbers can be written down using a positional notation. In order not to be biased, we shall not just work in base 10 but show how any base can be used. Base 10 probably arose for biological reasons since we have ten fingers.

Let's see how to represent numbers in *base* b where $b \geq 2$. If $d \leq 10$ then we represent numbers by sequences of symbols taken from the set

$$\mathbb{Z}_d = \{0, 1, 2, 3, \dots d - 1\}$$

but if $d > 10$ then we need new symbols for 10, 11, 12 and so forth. It's convenient to use A,B,C, ... For example, if we want to write numbers in base 12 we use the set of symbols

$$\{0, 1, \ldots, 9, A, B\}$$

whereas if we work in base 16 we use the set of symbols

$$\{0, 1, \ldots, 9, A, B, C, D, E, F\}.$$

If x is a sequence of symbols then we write x_d to make it clear that we are to interpret this sequence as a number in base d. Thus BAD_{16} is a number in base 16.

The symbols in a sequence x_d, reading from right to left, tell us the contribution each power of d such as d^0, d^1, d^2, etc makes to the number the sequence represents. Here are some examples.

Examples 0.6. Converting from base d to base 10.

(i): $11A9_{12}$ is a number in base 12. This represents the following number in base 10:

$$\mathbf{1} \times 12^3 + \mathbf{1} \times 12^2 + \mathbf{A} \times 12^1 + \mathbf{9} \times 12^0,$$

which is just the number

$$12^3 + 12^2 + 10 \times 12 + 9 = 2001.$$

(ii): BAD_{16} represents a number in base 16. This represents the following number in base 10:

$$B \times 16^2 + A \times 16^1 + D \times 16^0,$$

which is just the number

$$11 \times 16^2 + 10 \times 16 + 13 = 2989.$$

(iii): 5556_7 represents a number in base 7. This represents the following number in base 10:

$$5 \times 7^3 + 5 \times 7^2 + 5 \times 7^1 + 6 \times 7^0 = 2001.$$

These examples show how easy it is to convert from base d to base 10.

There are two ways to convert from base 10 to base d.

The first runs in outline as follows. Let n be the number in base 10 that we wish to write in base d. Look for the largest power m of d such that $ad^m \leq n$ where $a < d$. Then repeat for $n - ad^m$. Continuing in this way, we write n as a sum of multiples of powers of d and so we can write n in base d.

The second makes use of the remainder theorem. The idea behind this method is as follows. Let

$$n = a_m \dots a_1 a_0$$

in base d. Then a_0 is the remainder when n is divided by d, and the quotient is $n' = a_m \dots a_1$. Thus we can generate the digits of n in base d from *right to left* by repeatedly finding the next quotient and next remainder by dividing the current quotient by d; the process starts with our input number as first quotient.

Examples 0.7.

(i): Write 2001 in base 2.

	quotient	remainder
2	2001	
2	1000	1
2	500	0
2	250	0
2	125	0
2	62	1
2	31	0
2	15	1
2	7	1
2	3	1
2	1	1
	0	1

Thus 2001 in base 2 is (reading from bottom to top):

11111010001.

(ii): Write 2001 in base 12.

	quotient	remainder
12	2001	
12	166	9
12	13	$10 = A$
12	1	1
	0	1

Thus 2001 in base 12 is:

$11A9$.

(iii): Write 2001 in base 7.

	quotient	remainder
7	2001	
7	285	6
7	40	5
7	5	5
	0	5

Thus 2001 in base 7 is:

$$5556.$$

When converting from one base to another it is always wise to **check** your calculations by converting back.