Fermat’s little theorem and the Euler phi-function

Every one knows that the square of an odd number is odd and the square of an even number is even. This is the case \( p = 2 \) of the following theorem.

**Theorem 1.** (Fermat’s little theorem) *For a prime \( p \) and any integer \( a \), we have \( a^p \equiv a \pmod{p} \).*

**Proof:** We will prove this for positive integers \( a \) by induction. The reader may easily complete the proof. But we will use the binomial theorem and a fact about the binomial coefficients that we will not explain completely (until perhaps later in the course).

Instances of the binomial theorem are
\[
(x + y)^3 = x^3 + 3x^2y + 3xy^2 + y^3
\]
and
\[
(x + y)^7 = x^7 + 7x^6y + 21x^5y^2 + 35x^4y^3 + 35x^3y^4 + 21x^2y^5 + 7xy^6 + y^7.
\]
Note that \( (x + y)^7 \equiv x^7 + y^7 \pmod{7} \) for integers \( x, y \). In general, for primes \( p \), the coefficient of \( x^k y^{p-k} \) in the expansion of \( (x + y)^p \) is divisible by \( p \) for \( 0 < k < p \). This can be easily seen from the formula for binomial coefficients, but we omit the details. So we have
\[
(x + y)^p \equiv x^p + y^p \pmod{p}.
\]

Now it is simple to prove Theorem 1 by induction. Fix a prime \( p \). When \( a = 1 \), of course \( a^p \equiv a \pmod{p} \). If we know \( a^p \equiv a \pmod{p} \) for some integer \( a \), then, using \((*)\) for the first congruence,
\[
(a + 1)^p \equiv a^p + 1^p \equiv a + 1.
\]

\( \square \)

**Corollary.** *If an integer \( a \) is not divisible by a prime \( p \), then \( a^{p-1} \equiv 1 \pmod{p} \).*

**Proof:** If \( \gcd(a, p) = 1 \), we can use the cancellation law on \( a^p \equiv a \pmod{p} \) to cancel one power of \( p \).

We use \( \phi(n) \) to denote the number of integers \( k \) in the range \( 0 \leq k < n \) that are relatively prime to \( n \). Thus \( \phi(7) = 6 \) and \( \phi(15) = 8 \), for example. Also note \( \phi(1) = 1 \) because 0 is relatively prime to 1. The function \( \phi \) is called the *Euler phi-function* or the *Euler totient-function*. For primes \( p \), \( \phi(p) = p - 1 \) and for distinct primes \( p \) and \( q \), \( \phi(pq) = (p - 1)(q - 1) \). (Later in the course we will give a general formula for \( \phi(n) \).)

We now give an important extension of Theorem 1.
**Theorem 2.** (Euler) For any integer $n$ and any positive integer $a$ relatively prime to $n$, we have $a^{\phi(n)} \equiv 1 \pmod{n}$.

*Proof:* Postponed until later in the course. \(\square\)

The RSA Public Key Cryptosystem

We are concerned with ‘secret codes’ (cryptosystems) today. We speak of a *public key cryptosystem* when the encoding procedure is made public. Of course, the decoding procedure is kept secret by the person or persons who will receive the encoded messages. You may wonder why someone who sees an encoded message will not be able to figure out the original message since the encoding procedure is known to everyone.

The RSA (Rivest-Shamir-Adelman, 1977) public key cryptosystem is based on the fact that it is relatively easy to compute $y = x^a \pmod{m}$ when we are given $x$, $a$, and $m$, even when these are 1000-digit integers, but it is usually computationally infeasible to find $x$ when $y$, $a$, and $m$ are known. (When I use ‘$\equiv$’ rather than ‘$\equiv$’, I mean the left-hand side is the remainder, i.e. between 0 and the modulus.)

To implement RSA, we first require a large integer (maybe 300 digits, maybe 1000) $m$ and the value of $\phi(m)$. Then we choose a large integer $a$ relatively prime to $\phi(m)$. The integers $a$ and $m$ are made public. The value of $\phi(m)$ is kept secret by the recipient of the encoded messages. The ‘messages’ are to be integers $0 < x < m$, that are relatively prime to $m$. Such a message $x$ is to be encoded as $y = x^a \pmod{m}$. The encoded message is sent to the intended recipient, but it may be seen by the public.

The recipient must compute (once) an integer $b$ so that $ab \equiv 1 \pmod{\phi(m)}$; this is possible because $\text{GCD}(a, \phi(m)) = 1$. Then, the original message, the solution $x$ of $y = x^a \pmod{m}$, is simply $x = y^b \pmod{m}$. Proof: We chose $b$ so that $ab = 1 + t\phi(m)$ for some integer $t$. Then, using Theorem 2 near the end,

$$y^b \equiv (x^a)^b \equiv x^{ab} \equiv x^{1+t\phi(m)} \equiv x^1 \cdot (x^{\phi(m)})^t \equiv x \cdot (1)^t \equiv x \pmod{m}.$$  

Example: Take $m = 98765432109876543211$ and $a = 1111111111$. These are made public. Anyone may encode a message $x$ as

$$y = x^{1111111111} \pmod{98765432109876543211}.$$  

Suppose

$$y = 5394433917768191799.$$  

What is $x$. Can we find it in less that 1000000 years? In this case yes—because you can factor $m$ (with e.g. Mathematica) and use the factorization to find the secret $\phi(m)$. But no one knows how to reliable compute $\phi$ for 200 or 1000 digit integers.

There are still many details to be be discussed about implementation of RSA. More about this in the next lecture.

You may read many things about RSA and other public key cryptosystems, and the state of factorization algorithms, on the web.