Fermat's little theorem. RSA.

Computing large numbers modulo n

Fermat's little theorem

The Chinese reminder theorem

RSA

(a) In modulo arithmetic, you can always reduce a large number to its remainder

$$a \equiv a \textbf{ rem } n \pmod{n}.$$

(b) Addition, subtraction, and multiplication preserve congruence:

if $a \equiv b \pmod{n}$ and $c \equiv d \pmod{n}$, then

$$\begin{aligned} a + c &\equiv b + d \pmod{n}, \\ a - c &\equiv b - d \pmod{n}, \quad \text{and} \\ ac &\equiv bd \pmod{n}. \end{aligned}$$

(c) Instead of a congruence relation, you can work with the equality of the remainders

$$a \equiv b \pmod{n} \quad \text{if and only if} \quad a \textbf{ rem } n = b \textbf{ rem } n$$

Example

Compute $(25^8 + 13^2 + 15)$ **rem** 21

$$25^2 \equiv 625 \equiv 16 \pmod{21}$$
$$25^4 \equiv (25^2)^2 \equiv 16^2 \equiv 256 \equiv 4 \pmod{21}$$
$$25^8 \equiv (25^4)^2 \equiv 4^2 \equiv 16 \pmod{21}$$

$$13^2 \equiv 169 \equiv 1 \pmod{21}$$

$$25^8 + 13^2 + 15 \equiv 16 + 1 + 15 \equiv 32 \equiv 11 \pmod{21}$$

So, finally,

$$(25^8 + 13^2 + 15) \textbf{ rem } 21 = 11.$$

Lemma 1

Fermat's little theorem

The Chinese reminder theorem

RSA

Lemma. Let p be a prime. If k is not a multiple of p, then if

$$ak \equiv bk \pmod{p}$$

then

$$a \equiv b \pmod{p}$$

Proof.

$$p \mid (ak - bk)$$

$$p \mid k(a - b)$$

By the lemma we proved in the previous lecture, $p \mid k$ or $p \mid (a-b)$. Since $p \nmid k$, we conclude that $p \mid (a - b)$, or equivalently,

$$a \equiv b \pmod{p}.$$

□

Observation

Fermat's little theorem

The Chinese reminder theorem

RSA

x	5	6	7	8	9	10	11	12	13	14	15	16
x **rem** 7	5	6	*0*	*1*	*2*	*3*	*4*	*5*	*6*	0	1	2

When computing remainders of the division by p for a sequence of consecutive integers, we get all numbers from 0 to $p-1$.

Can we come up with other ways for producing these numbers?

Observation

For exmaple, we take every second integer: 2, 4, 6, 8, ...

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14
x **rem** 7		2		4		6		1		3		5		0

We generate the same numbers, even though they are permuted.

Can we try another sequence: 3, 6, 9, 12, 15, ... right?

Yes, we can, but we can prove a more general result.

Lemma 2

Fermat's little theorem

The Chinese reminder theorem

RSA

Lemma. Let p be a prime. If k is not a multiple of p, the sequence

$$k \textbf{ rem } p, \quad 2k \textbf{ rem } p, \quad 3k \textbf{ rem } p, \quad \ldots \quad (p-1)k \textbf{ rem } p$$

is a permuation of

$$1,\ 2,\ 3,\ \ldots\ p-1.$$

Proof. All remainders are in the interval between 1 and $p-1$.
Let's prove by contradiction that all of them are different.
Assume that exist two distinct integers $1 \leq i \neq j \leq p-1$ such that

$$ik \textbf{ rem } p = jk \textbf{ rem } p$$

$$ik \equiv jk \pmod{p}$$

By Lemma 1, $i \equiv j \pmod{p}$, and so $i = j$, because both are integers between 1 and $p-1$. Therefore, all remainders are different, and this proves the lemma. □

Examples

Fermat's little theorem

The Chinese reminder theorem

RSA

$k = 2$ and $p = 5$:

$$2 \textbf{ rem } 5 = 2$$
$$4 \textbf{ rem } 5 = 4$$
$$6 \textbf{ rem } 5 = 1$$
$$8 \textbf{ rem } 5 = 3$$

$k = 12$ and $p = 7$:

$$12 \textbf{ rem } 7 = 5$$
$$24 \textbf{ rem } 7 = 3$$
$$36 \textbf{ rem } 7 = 1$$
$$48 \textbf{ rem } 7 = 6$$
$$60 \textbf{ rem } 7 = 4$$
$$72 \textbf{ rem } 7 = 2$$

Fermat's little theorem

Fermat's little theorem

The Chinese reminder theorem

RSA

Theorem. Let p be a prime. If k is not a multiple of p (i.e. $p \nmid k$) then

$$k^{p-1} \equiv 1 \pmod{p}$$

Proof.

$$1 \cdot 2 \cdot 3 \cdots (p-1) = (\underbrace{k \bmod p}_{\equiv k \pmod{p}})(\underbrace{2k \bmod p}_{\equiv 2k \pmod{p}}) \cdot (\underbrace{3k \bmod p}_{\cdots}) \cdots ((p-1)k \bmod p)$$

$$\begin{aligned} 1 \cdot 2 \cdot 3 \cdots (p-1) &\equiv k \cdot 2k \cdot 3k \cdots (p-1)k \pmod{p} \\ &\equiv 1 \cdot 2 \cdot 3 \cdots (p-1) \cdot k^{p-1} \pmod{p}. \end{aligned}$$

$$1 \equiv k^{p-1} \pmod{p}. \qquad \text{(by Lemma 1)}$$

□

Computing a multiplicative inverse

Fermat's little theorem

The Chinese reminder theorem

RSA

Theorem. Let p be a prime. If k is not a multiple of p then

$$k^{p-1} \equiv 1 \pmod{p}$$

Using this theorem, we can find an inverse, x,

$$kx \equiv 1 \pmod{p}.$$

How can we do that?
If k is not a multiple of p and p is a prime than k^{p-2} is an inverse!

$$k \cdot k^{p-2} = k^{p-1} \equiv 1 \pmod{p}$$

Because we need an inverse modulo p, the remainder (k^{p-2} **rem** p) is an inverse too.

Notice the difference with the Extended Euclid's algorithm that can be used for computing an inverse even when p is not a prime.

Computing a multiplicative inverse

Fermat's little theorem

The Chinese reminder theorem

RSA

Compute an inverse of 12 modulo 137, that is x such that

$$12x \equiv 1 \pmod{137}$$

137 is a prime, so we apply the theorem,

$$12 \cdot 12^{137-2} \equiv 1 \pmod{137}.$$

The inverse is

$$x = 12^{137-2} = 12^{135}.$$

Or, in fact, just the remainder 12^{135} **rem** 137.

Computing a multiplicative inverse

Fermat's little theorem

The Chinese reminder theorem

RSA

Compute $x = 12^{135}$ **rem** 137.

$$12^2 \equiv 144 \equiv 7 \pmod{137}$$
$$12^4 \equiv (12^2)^2 \equiv 7^2 \equiv 49 \pmod{137}$$
$$12^8 \equiv (12^4)^2 \equiv 49^2 \equiv 2401 \equiv 72 \pmod{137}$$
$$12^{16} \equiv (12^8)^2 \equiv 72^2 \equiv 5184 \equiv 115 \pmod{137}$$
$$12^{32} \equiv (12^{16})^2 \equiv 115^2 \equiv 13225 \equiv 73 \pmod{137}$$
$$12^{64} \equiv (12^{32})^2 \equiv 73^2 \equiv 5329 \equiv 123 \pmod{137}$$
$$12^{128} \equiv (12^{64})^2 \equiv 123^2 \equiv 15129 \equiv 59 \pmod{137}$$

$$12^{135} \equiv 12^{128} \cdot 12^4 \cdot 12^2 \cdot 12 \equiv 59 \cdot 49 \cdot 7 \cdot 12 \equiv 242844 \equiv \mathit{80} \pmod{137}$$

Let's check:

$$12 \cdot 80 \equiv 960 \equiv 1 \pmod{137}$$

Thus 80 is a multiplicative inverse, indeed.

The Chinese reminder theorem

Fermat's little theorem

The Chinese reminder theorem

RSA

Theorem (Chinese reminder). Let n_1 , n_2 , ..., n_k be pairwise relatively prime positive integers greater than one and a_1, a_2 , ..., a_n arbitrary integers. Then the system

$$\begin{aligned} x &\equiv a_1 \pmod{n_1}, \\ x &\equiv a_2 \pmod{n_2}, \\ &\dots \\ x &\equiv a_k \pmod{n_k} \end{aligned}$$

has a unique solution modulo $n = n_1 \cdot n_2 \cdots n_k$.

(That is, there is a solution x with $0 \leq x < n$, and all other solutions are congruent modulo n to this solution.)

RSA cryptosystem

Fermat's little theorem

The Chinese reminder theorem

RSA

RSA is a public key cryptosystem introduced by Ronald Rivest, Adi Shamir, and Leonard Adleman.

Constructing a publuc key:

1. You pick two large primes p and q. (How large? In practice, if their decimal representations are more than 200 digits long, they are large enough).
2. Let $n = pq$.
3. You pick an integer e such that $\gcd\big(e, (p-1)(q-1)\big) = 1$.

The public key is a pair (n, e).

RSA encryption

Fermat's little theorem

The Chinese reminder theorem

RSA

You are given the *public key* (n, e), such that $n = pq$, and $\gcd\big(e, (p-1)(q-1)\big) = 1$. Where p and q are unknown primes.

1. You break the message (a long string of digits) into blocks so that each is less than n

$$\text{Message} = \underbrace{17519273}_{M_1<n}\,\underbrace{40137520}_{M_2<n}\,\underbrace{43028230}_{M_3<n}\,\underbrace{14459489}_{M_4<n}$$

2. Each block is encrypted separately:

$$C_i = M_i^e \ \textbf{rem}\ n$$

3. The encrypted blocks can be transmitted to the receiver.

RSA decryption

Fermat's little theorem

The Chinese reminder theorem

RSA

The *decryption key* is an integer d such that

$$ed \equiv 1 \pmod{(p-1)(q-1)}$$

So, d is the multiplicative inverse of e modulo $(p-1)(q-1)$.
If p and q are unknown, it's hard to compute d.

To decrypt each encrypted block C_i, you compute $C_i^d \textbf{ rem } n$.

Turns out that the original message

$$M_i = C_i^d \textbf{ rem } n.$$

Let's show that this is the case.

RSA decryption

Fermat's little theorem

The Chinese reminder theorem

RSA

The decryption key d is such that $ed \equiv 1 \pmod{(p-1)(q-1)}$

$$ed = 1 + k(p-1)(q-1)$$

$$\begin{aligned} C_i &\equiv M_i^e \pmod{pq} \\ C_i^d &\equiv (M_i^e)^d \equiv M_i^{1+k(p-1)(q-1)} \pmod{pq} \\ &\equiv M_i \cdot M_i^{k(p-1)(q-1)} \pmod{pq} \end{aligned}$$

If $a \equiv b \pmod{pq}$, there exists t s.t. $a - b = tpq$, and so

$$a \equiv b \pmod{q} \quad \text{and} \quad a \equiv b \pmod{p}.$$

Therefore,

$$\begin{aligned} C_i^d &\equiv M_i \cdot M_i^{k(p-1)(q-1)} \pmod{p} \\ C_i^d &\equiv M_i \cdot M_i^{k(p-1)(q-1)} \pmod{q} \end{aligned}$$

RSA decryption

Fermat's little theorem

The Chinese reminder theorem

RSA

Take the first:

$$C_i^d \equiv M_i \cdot (M_i^{p-1})^{k(q-1)} \pmod{p}$$

If $p \mid M_i$, then $M_i \equiv 0 \pmod{p}$. This means $C_i^d \equiv 0 \pmod{p}$. Hence,

$$C_i^d \equiv 0 \equiv M_i \pmod{p}$$

If $p \nmid M_i$, then $M_i^{p-1} \equiv 1 \pmod{p}$ by Fermat's little theorem, hence

$$C_i^d \equiv M_i \cdot 1^{k(q-1)} \equiv M_i \pmod{p}$$

Therefore in any case,

$$C_i^d \equiv M_i \pmod{p}$$

Similarly,

$$C_i^d \equiv M_i \pmod{q}$$

RSA decryption

Fermat's little theorem

The Chinese reminder theorem

RSA

We have shown that

$$C_i^d \equiv M_i \pmod{p}$$
$$C_i^d \equiv M_i \pmod{q}$$

Recall that we want to prove that

$$C_i^d \equiv M_i \pmod{pq}$$

If this is the case, then we can decrypt C_i by taking the remainder C_i^d **rem** pq.

RSA decryption

Fermat's little theorem

The Chinese reminder theorem

RSA

We have shown that

$$C_i^d \equiv M_i \pmod{p}$$
$$C_i^d \equiv M_i \pmod{q}$$

Recall that we want to prove that

$$C_i^d \equiv M_i \pmod{pq}$$

If this is the case, then we can decrypt C_i by taking the remainder C_i^d **rem** pq.

$C_i^d - M_i$ is a multiple of p and a multiple of q, which are primes. But because the prime factorization is unique, the difference $C_i^d - M_i$ should be also a multiple of pq. Thus

$$C_i^d \equiv M_i \pmod{pq}$$

This is why RSA decryption works!