CPSC 467b: Cryptography and Computer Security

Michael J. Fischer

Lecture 9
February 14, 2013

Integer Division (cont.)
   Relatively prime numbers, $\mathbf{Z}_n^*$, and $\phi(n)$

Computing in $\mathbf{Z}_n$
   Modular multiplication
   Modular inverses
   Extended Euclidean algorithm

Generating RSA Encryption and Decryption Exponents

Euler's Theorem

Generating RSA Modulus
   Finding primes by guess and check
   Density of primes

Integer Division (cont.)

Outline	Integer Division (cont.)	Computing in $\mathbf{Z}_n$	RSA exponents	Euler	RSA modulus
	●○○○	○○○○○○○○○○○			○○○○○○

Relatively prime numbers, $\mathbf{Z}_n^*$, and $\phi(n)$

Relatively prime numbers

Two integers a and b are *relatively prime* if they have no common prime factors.

Equivalently, a and b are *relatively prime* if $\gcd(a, b) = 1$.

Let $\mathbf{Z}_n^*$ be the set of integers in $\mathbf{Z}_n$ that are relatively prime to n, so

$$\mathbf{Z}_n^* = \{a \in \mathbf{Z}_n \mid \gcd(a, n) = 1\}.$$

Outline	Integer Division (cont.)	Computing in $\mathbf{Z}_n$	RSA exponents	Euler	RSA modulus
	○●○○	○○○○○○○○○○○			○○○○○○

Relatively prime numbers, $\mathbf{Z}_n^*$, and $\phi(n)$

Euler's totient function $\phi(n)$

$\phi(n)$ is the cardinality (number of elements) of $\mathbf{Z}_n^*$, i.e.,

$$\phi(n) = |\mathbf{Z}_n^*|.$$

Properties of $\phi(n)$:

1. If p is prime, then

$$\phi(p) = p - 1.$$

2. More generally, if p is prime and $k \geq 1$, then

$$\phi(p^k) = p^k - p^{k-1} = (p-1)p^{k-1}.$$

3. If $\gcd(m, n) = 1$, then

$$\phi(mn) = \phi(m)\phi(n).$$

Outline	Integer Division (cont.)	Computing in $\mathbf{Z}_n$	RSA exponents	Euler	RSA modulus
	○○●○	○○○○○○○○○○○			○○○○○○

Relatively prime numbers, $\mathbf{Z}_n^*$, and $\phi(n)$

Example: $\phi(26)$

Can compute $\phi(n)$ for all $n \geq 1$ given the factorization of n.

$$
\begin{aligned}
\phi(126) &= \phi(2) \cdot \phi(3^2) \cdot \phi(7) \\
&= (2-1) \cdot (3-1)(3^{2-1}) \cdot (7-1) \\
&= 1 \cdot 2 \cdot 3 \cdot 6 = 36.
\end{aligned}
$$

The 36 elements of $\mathbf{Z}_{126}^*$ are:

1, 5, 11, 13, 17, 19, 23, 25, 29, 31, 37, 41, 43, 47, 53, 55, 59, 61, 65, 67, 71, 73, 79, 83, 85, 89, 95, 97, 101, 103, 107, 109, 113, 115, 121, 125.

Outline	Integer Division (cont.)	Computing in $\mathbf{Z}_n$	RSA exponents	Euler	RSA modulus
	0000	00000000000			000000

Relatively prime numbers, $\mathbf{Z}_n^*$, and $\phi(n)$

A formula for $\phi(n)$

Here is an explicit formula for $\phi(n)$.

Theorem
Write n in factored form, so $n = p_1^{e_1} \cdots p_k^{e_k}$, where $p_1, \ldots, p_k$ are distinct primes and $e_1, \ldots, e_k$ are positive integers.[1] *Then*

$$\phi(n) = (p_1 - 1) \cdot p_1^{e_1 - 1} \cdots (p_k - 1) \cdot p_k^{e_k - 1}.$$

For the product of distinct primes p and q,

$$\phi(pq) = (p - 1)(q - 1).$$

---

[1]By the fundamental theorem of arithmetic, every integer can be written uniquely in this way up to the ordering of the factors.

Computing in $\mathbf{Z}_n$

Outline	Integer Division (cont.)	Computing in $\mathbf{Z}_n$	RSA exponents	Euler	RSA modulus
	0000	●000000000			000000

Modular multiplication

Multiplication modulo n

Theorem
$\mathbf{Z}_n^*$ is closed under multiplication modulo n.

This says, if a and b are both in $\mathbf{Z}_n^*$, then $(ab \bmod n)$ is also in $\mathbf{Z}_n^*$.

Proof.
If neither a nor b share a prime factor with n, then neither does their product ab. $\qquad\square$

Outline	Integer Division (cont.)	Computing in $\mathbf{Z}_n$	RSA exponents	Euler	RSA modulus
	○○○○	○●○○○○○○○○○			○○○○○○

Modular multiplication

Example: Multiplication in $\mathbf{Z}_{26}^*$

Let $n = 26 = 2 \cdot 13$. Then

$$\mathbf{Z}_{26}^* = \{1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25\}$$
$$\phi(26) = |\mathbf{Z}_{26}^*| = 12.$$

Multiplication examples:

$$5 \times 7 \bmod 26 = 35 \bmod 26 = 9.$$

$$3 \times 25 \bmod 26 = 75 \bmod 26 = 23.$$

$$9 \times 3 \bmod 26 = 27 \bmod 26 = 1.$$

We say that 3 is the *multiplicative inverse* of 9 in $\mathbf{Z}_{26}^*$.

Outline	Integer Division (cont.)	Computing in $\mathbf{Z}_n$	RSA exponents	Euler	RSA modulus
	oooo	oo●oooooooo			oooooo

Modular inverses

Example: Inverses the elements in $\mathbf{Z}_{26}^*$.

x	1	3	5	7	9	11	15	17	19	21	23	25
x^{-1}	1	9	21	15	3	19	7	23	11	5	17	25
$\equiv_n$	1	9	-5	-11	3	-7	7	-3	11	5	-9	-1

Bottom row gives equivalent integers in range $[-12, \ldots, 13]$.

Note that $(26 - x)^{-1} = -x^{-1}$.

Hence, last row reads same back to front except for change of sign.

Once the inverses for the first six numbers are known, the rest of the table is easily filled in.

Outline	Integer Division (cont.)	Computing in $\mathbf{Z}_n$	RSA exponents	Euler	RSA modulus
	○○○○	○○○●○○○○○○○○			○○○○○○

Modular inverses

Finding modular inverses

Let $u \in \mathbf{Z}_n^*$. We wish to find u^{-1} modulo n.

By definition, u^{-1} is the element $v \in \mathbf{Z}_n^*$ (if it exists) such that

$$uv \equiv 1 \pmod{n}.$$

This equation holds iff $n \mid (uv - 1)$ iff $uv - 1 = qn$ for some integer q (positive or negative).

We can rewrite this equation as

$$uv - nq = 1. \tag{1}$$

u and n are given and v and q are unknowns. If we succeed in finding a solution over the integers, then v is the desired inverse u^{-1}.

Outline	Integer Division (cont.)	Computing in $\mathbf{Z}_n$	RSA exponents	Euler	RSA modulus
	○○○○	○○○○●○○○○○○			○○○○○○

Modular inverses

Diophantine equations

A *Diophantine equation* is a linear equation in two unknowns over the integers.

$$ax + by = c \tag{2}$$

Here, a, b, c are given integers. A solution consists of integer values for the unknowns x and y that make (2) true.

We see that equation 1 fits the general form for a Diophantine equation, where

$$\begin{aligned} a &= u \\ b &= -n \\ c &= 1 \end{aligned} \tag{3}$$

Outline	Integer Division (cont.)	Computing in $\mathbf{Z}_n$	RSA exponents	Euler	RSA modulus
	oooo	oooooo●ooooo			oooooo

Modular inverses

Existence of solution

Theorem
The Diophantine equation

$$ax + by = c$$

has a solution over $\mathbf{Z}$ *(the integers) iff* $\gcd(a, b) \mid c$.

It can be solved by a process akin to the Euclidean algorithm,
which we call the *Extended Euclidean algorithm*.

Outline	Integer Division (cont.)	Computing in $\mathbf{Z}_n$	RSA exponents	Euler	RSA modulus
	oooo	ooooooo●oooo			oooooo

Extended Euclidean algorithm

Extended Euclidean algorithm

The algorithm generates a sequence of triples of numbers
$T_i = (r_i, u_i, v_i)$, each satisfying the invariant

$$r_i = au_i + bv_i \geq 0. \qquad (4)$$

$$T_1 = \begin{cases} (a, 1, 0) & \text{if } a \geq 0 \\ (-a, -1, 0) & \text{if } a < 0 \end{cases}$$

$$T_2 = \begin{cases} (b, 0, 1) & \text{if } b \geq 0 \\ (-b, 0, -1) & \text{if } b < 0 \end{cases}$$

T_{i+2} is obtained by subtracting a multiple of T_{i+1} from from T_i so
that $r_{i+2} < r_{i+1}$. This is similar to the way the Euclidean algorithm
obtains ($a \bmod b$) from a and b.

In detail, let $q_{i+1} = \lfloor r_i/r_{i+1} \rfloor$. Then $T_{i+2} = T_i - q_{i+1}T_{i+1}$, so
$r_{i+2} = r_i - q_{i+1}r_{i+1} = r_i \bmod r_{i+1}$

$u_{i+2} = u_i - q_{i+1}u_{i+1}$

Outline	Integer Division (cont.)	Computing in $\mathbf{Z}_n$	RSA exponents	Euler	RSA modulus
	OOOO	OOOOOOOO●OOO			OOOOOO

Extended Euclidean algorithm

Finding all solutions

Returning to the original equation,

$$ax + by = c \qquad (2)$$

if $c = \gcd(a, b)$, then $x = u_{t-1}$ and $y = v_{t-1}$ is a solution.

If $c = k \cdot \gcd(a, b)$ is a multiple of $\gcd(a, b)$, then $x = ku_{t-1}$ and $y = kv_{t-1}$ is a solution.

Otherwise, $\gcd(a, b)$ does not divide c, and one can show that (2) has no solution.

Outline	Integer Division (cont.)	Computing in $\mathbf{Z}_n$	RSA exponents	Euler	RSA modulus
	oooo	ooooooooo●oo			oooooo

Extended Euclidean algorithm

Example of extended Euclidean algorithm

Suppose one wants to solve the equation

$$31x - 45y = 3 \qquad (6)$$

Here, $a = 31$ and $b = -45$. We begin with the triples

$$
\begin{aligned}
T_1 &= (31, 1, 0) \\
T_2 &= (45, 0, -1)
\end{aligned}
$$

Outline	Integer Division (cont.)	Computing in $\mathbf{Z}_n$	RSA exponents	Euler	RSA modulus
	oooo	ooooooooo00●o			oooooo

Extended Euclidean algorithm

Computing the triples

The computation is shown in the following table:

i	r_i	u_i	v_i	q_i
1	31	1	0	
2	45	0	-1	0
3	31	1	0	1
4	14	-1	-1	2
5	3	3	2	4
6	2	-13	-9	1
7	1	16	11	2
8	0	-45	-31	

Outline	Integer Division (cont.)	Computing in $\mathbf{Z}_n$	RSA exponents	Euler	RSA modulus
	oooo	oooooooooo●			oooooo

Extended Euclidean algorithm

Extracting the solution

From $T_7 = (1, 16, 11)$, we obtain the solution $x = 16$ and $y = 11$
to the equation

$$1 = 31x - 45y$$

We can check this by substituting for x and y:

$$31 \cdot 16 + (-45) \cdot 11 = 496 - 495 = 1.$$

The solution to

$$31x - 45y = 3 \tag{6}$$

is then $x = 3 \cdot 16 = 48$ and $y = 3 \cdot 11 = 33$.

Generating RSA Encryption and Decryption Exponents

Recall RSA exponent requirement

Recall that the RSA encryption and decryption exponents must be chosen so that

$$ed \equiv 1 \pmod{\phi(n)}, \tag{7}$$

that is, d is e^{-1} in $\mathbf{Z}_{\phi(n)}^*$.

How does Alice choose e and d to satisfy (7)?

- Choose a random integer $e \in \mathbf{Z}_{\phi(n)}^*$.
- Solve (7) for d.

We know now how to solve (7), but how does Alice sample at random from $\mathbf{Z}_{\phi(n)}^*$?

Sampling from $\mathbf{Z}_n^*$

If $\mathbf{Z}_{\phi(n)}^*$ is large enough, Alice can just choose random elements from $\mathbf{Z}_{\phi(n)}$ until she encounters one that also lies in $\mathbf{Z}_{\phi(n)}^*$.

A candidate element e lies in $\mathbf{Z}_{\phi(n)}^*$ iff $\gcd(e, \phi(n)) = 1$, which can be computed efficiently using the Euclidean algorithm.[2]

---

[2] $\phi(n)$ itself is easily computed for an RSA modulus $n = pq$ since $\phi(n) = (p-1)(q-1)$ and Alice knows p and q.

How large is large enough?

If $\phi(\phi(n))$ (the size of $\mathbf{Z}_{\phi(n)}^*$) is much smaller than $\phi(n)$ (the size of $\mathbf{Z}_{\phi(n)}$), Alice might have to search for a long time before finding a suitable candidate for e.

In general, $\mathbf{Z}_m^*$ can be considerably smaller than m.
Example:

$$m = |\mathbf{Z}_m| = 2 \cdot 3 \cdot 5 \cdot 7 = 210$$
$$\phi(m) = |\mathbf{Z}_m^*| = 1 \cdot 2 \cdot 4 \cdot 6 = 48.$$

In this case, the probability that a randomly-chosen element of $\mathbf{Z}_m$ falls in $\mathbf{Z}_m^*$ is only $48/210 = 8/35 = 0.228\ldots$.

A lower bound on $\phi(m)/m$

The following theorem provides a crude lower bound on how small $\mathbf{Z}_m^*$ can be relative to the size of $\mathbf{Z}_m$.

Theorem
For all $m \geq 2$,

$$\frac{|\mathbf{Z}_m^*|}{|\mathbf{Z}_m|} \geq \frac{1}{1 + \lfloor \log_2 m \rfloor}.$$

Proof.

Write $m = \prod_{i=1}^{t} p_i^{e_i}$, where p_i is the ith prime that divides m and $e_i \geq 1$. Then $\phi(m) = \prod_{i=1}^{t}(p_i - 1)p_i^{e_i-1}$, so

$$\frac{|\mathbf{Z}_m^*|}{|\mathbf{Z}_m|} = \frac{\phi(m)}{m} = \frac{\prod_{i=1}^{t}(p_i-1)p_i^{e_i-1}}{\prod_{i=1}^{t} p_i^{e_i}} = \prod_{i=1}^{t}\left(\frac{p_i - 1}{p_i}\right). \quad (8)$$

To estimate the size of $\prod_{i=1}^{t}(p_i - 1)/p_i$, note that

Expected difficulty of choosing RSA exponent e

For n a 1024-bit integer, $\phi(n) < n < 2^{1024}$.

Hence, $\log_2(\phi(n)) < 1024$, so $\lfloor \log_2(\phi(n)) \rfloor \leq 1023$.

By the theorem, the fraction of elements in $\mathbf{Z}_{\phi(n)}$ that also lie in $\mathbf{Z}^*_{\phi(n)}$ is at least

$$\frac{1}{1 + \lfloor \log_2 \phi(n) \rfloor} \geq \frac{1}{1024}.$$

Therefore, the expected number of random trials before Alice finds a number in $\mathbf{Z}^*_{\phi(n)}$ is provably at most 1024 and is likely much smaller.

Euler's Theorem

Repeated multiplication in $\mathbf{Z}_n^*$

If any element $x \in \mathbf{Z}_n^*$ is repeatedly multiplied by itself, the result is eventually 1. [3]

Example, for $x = 5 \in \mathbf{Z}_{26}^*$: 5, 25, 21, 1, 5, 25, 21, 1, . . .

Let x^k denote the result of multiplying x by itself k times.
The *order of x*, written $\text{ord}(x)$, is the smallest integer $k \geq 1$ for which $x^k = 1$.

Theorem
$\text{ord}(x) | \phi(n)$. *(Recall, $\phi(n)$ is the size of $\mathbf{Z}_n^*$).*

---

[3]The first repeated element must be x. If not, then some $y \neq x$ is the first to repeat. The element immediately preceding each occurrence of y is yx^{-1}. But then yx^{-1} is the first to repeat, a contradiction. Hence, $x = x^{k+1}$ for some $k \geq 1$, so $x^k = x^{k+1}x^{-1} = xx^{-1} = 1$.

Euler's and Fermat's theorem

Theorem (Euler's theorem)
$x^{\phi(n)} \equiv 1 \ (mod \ n)$ for all $x \in \mathbf{Z}_n^*$.

Proof.
Since $\text{ord}(x) \,|\, \phi(n)$, we have

$$x^{\phi(n)} \equiv (x^{\text{ord}(x)})^{\phi(n)/\text{ord}(x)} \equiv 1^{\phi(n)/\text{ord}(x)} \equiv 1 \ (\text{mod } n).$$

□

As a special case, we have

Theorem (Fermat's theorem)
$x^{(p-1)} \equiv 1 \ (mod \ p)$ for all x, $1 \le x \le p-1$, where p is prime.

An important corollary

Corollary
Let $r \equiv s \ (mod \ \phi(n))$. Then $a^r \equiv a^s \ (mod \ n)$ for all $a \in \mathbf{Z}_n^*$.

Proof.
If $r \equiv s \ (mod \ \phi(n))$, then $r = s + u\phi(n)$ for some integer u. Then using Euler's theorem, we have

$$a^r \equiv a^{s+u\phi(n)} \equiv a^s \cdot (a^u)^{\phi(n)} \equiv a^s \cdot 1 \equiv a^s \ (mod \ n),$$

as desired. $\qquad\square$

Application to RSA

Recall the RSA encryption and decryption functions

$$\begin{aligned} E_e(m) &= m^e \text{ mod } n \\ D_d(c) &= c^d \text{ mod } n \end{aligned}$$

where $n = pq$ is the product of two distinct large primes p and q.

This corollary gives a sufficient condition on e and d to ensure that the resulting cryptosystem works. That is, we require that

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

Then $D_d(E_e(m)) \equiv m^{ed} \equiv m^1 \equiv m \pmod{n}$ for all messages $m \in \mathbf{Z}_n^*$.

Messages not in $\mathbf{Z}_n^*$

What about the case of messages $m \in \mathbf{Z}_n - \mathbf{Z}_n^*$?

There are several answers to this question.

1. Alice doesn't really want to send such messages if she can avoid it.
2. If Alice sends random messages, her probability of choosing a message not in $\mathbf{Z}_n^*$ is very small — only about $2/\sqrt{n}$.
3. RSA does in fact work for all $m \in \mathbf{Z}_n$, even though Euler's theorem fails for $m \notin \mathbf{Z}_n^*$.

Why Alice might want to avoid sending messages not in $\mathbf{Z}_n^*$

If $m \in \mathbf{Z}_n - \mathbf{Z}_n^*$, either $p \,|\, m$ or $q \,|\, m$ (but not both because $m < pq$).

If Alice ever sends such a message and Eve is astute enough to compute $\gcd(m, n)$ (which she can easily do), then Eve will succeed in breaking the cryptosystem.

Why?

Why a random message is likely to be in $\mathbf{Z}_n^*$

The number of messages in $\mathbf{Z}_n - \mathbf{Z}_n^*$ is only

$$n - \phi(n) = pq - (p-1)(q-1) = p + q - 1$$

out of a total of $n = pq$ messages altogether.

If p and q are both 512 bits long, then the probability of choosing a bad message is only about $2 \cdot 2^{512}/2^{1024} = 1/2^{511}$.

Such a low-probability event will likely never occur during the lifetime of the universe.

RSA works anyway

For $m \in \mathbf{Z}_n - \mathbf{Z}_n^*$, RSA works anyway, but for different reasons.

For example, if $m = 0$, it is clear that $(0^e)^d \equiv 0 \pmod{n}$, yet Euler's theorem fails since $0^{\phi(n)} \not\equiv 1 \pmod{n}$.

We omit the proof of this curiosity.

Generating RSA Modulus

Outline	Integer Division (cont.)	Computing in $\mathbf{Z}_n$	RSA exponents	Euler	RSA modulus
	0000	00000000000			●00000

Random primes

Recall RSA modulus

Recall the RSA modulus, $n = pq$. The numbers p and q should be random distinct primes of about the same length.

The method for finding p and q is similar to the "guess-and-check" method used to find random numbers in $\mathbf{Z}_m^*$.

Namely, keep generating random numbers p of the right length until a prime is found. Then keep generating random numbers q of the right length until a prime different from p is found.

| Outline | Integer Division (cont.) | Computing in $\mathbf{Z}_n$ | RSA exponents | Euler | RSA modulus |
| | oooo | oooooooooo | | | o●oooo |

Random primes

Generating random primes of a given length

To generate a k-bit prime:

- ▶ Generate $k - 1$ random bits.
- ▶ Put a "1" at the front.
- ▶ Regard the result as binary number, and test if it is prime.

We defer the question of how to test if the number is prime and look now at the expected number of trials before this procedure will terminate.

Outline	Integer Division (cont.)	Computing in $\mathbf{Z}_n$	RSA exponents	Euler	RSA modulus
	oooo	ooooooooooo			ooo●ooo

Density of primes

Expected number of trials to find a prime

The above procedure samples uniformly from the set
$B_k = \mathbf{Z}_{2^k} - \mathbf{Z}_{2^{k-1}}$ of binary numbers of length exactly k.

Let p_k be the fraction of elements in B_k that are prime. Then the
expected number of trials to find a prime is $1/p_k$.

While p_k is difficult to determine exactly, the celebrated *Prime
Number Theorem* allows us to get a good estimate on that
number.

Outline	Integer Division (cont.)	Computing in $\mathbf{Z}_n$	RSA exponents	Euler	RSA modulus
	0000	00000000000			000●00

Density of primes

Prime number function

Let $\pi(n)$ be the number of numbers $\leq n$ that are prime.

For example, $\pi(10) = 4$ since there are four primes ≤ 10, namely, 2, 3, 5, 7.

Outline	Integer Division (cont.)	Computing in $\mathbf{Z}_n$	RSA exponents	Euler	RSA modulus
	oooo	ooooooooooo			oooooo●o

Density of primes

Prime number theorem

Theorem

$\pi(n) \approx n/(\ln n)$, where $\ln n$ is the natural logarithm $\log_e n$.

Notes:

▶ We ignore the critical issue of how good an approximation this is. The interested reader is referred to a good mathematical text on number theory.

▶ Here $e = 2.71828\ldots$ is the base of the natural logarithm, not to be confused with the RSA encryption exponent, which, by an unfortunate choice of notation, we also denote by e.

Outline	Integer Division (cont.)	Computing in $\mathbf{Z}_n$	RSA exponents	Euler	RSA modulus
	oooo	ooooooooooo			oooooo●

Density of primes

Likelihood of randomly finding a prime

The chance that a randomly picked number in $\mathbf{Z}_n$ is prime is

$$\frac{\pi(n-1)}{n} \approx \frac{n-1}{n \cdot \ln(n-1)} \approx \frac{1}{\ln n}.$$

Since $B_k = \mathbf{Z}_{2^k} - \mathbf{Z}_{2^{k-1}}$, we have

$$
\begin{aligned}
p_k &= \frac{\pi(2^k - 1) - \pi(2^{k-1} - 1)}{2^{k-1}} \\
&= \frac{2\pi(2^k - 1)}{2^k} - \frac{\pi(2^{k-1} - 1)}{2^{k-1}} \\
&\approx \frac{2}{\ln 2^k} - \frac{1}{\ln 2^{k-1}} \approx \frac{1}{\ln 2^k} = \frac{1}{k \ln 2}.
\end{aligned}
$$

Hence, the expected number of trials before success is $\approx k \ln 2$.
For $k = 512$, this works out to $512 \times 0.693\ldots \approx 355$.