Chapter 6. Number Theory

6.1 Introduction

Problems such as factorization or primality of large integers, root extraction, solution to simultaneous equations modulo different moduli, etc., are among the frequently used ingredients in modern cryptography. They are also fascinating topics in the theory of numbers. In this chapter we study some basic facts and algorithms in number theory, which have important relevance to modern cryptography.

6.1.1 Chapter Outline

§6.2 introduces the basic notions and operations of congruences and residue classes. §6.3 introduces Euler's phi function. §6.4 shows a unified view of the theorems of Fermat, Euler and Lagrange. §6.5 introduces the notion of quadratic residues. §6.6 introduces algorithms for computing square roots modulo an integer. Finally, §6.7 introduces the Blum integers.

6.2 Congruences and Residue Classes

In §4.3.2.5 we have defined congruence system modulo a positive integer $n > 1$ and studied a few properties of such systems. Here we shall study a few more facts of the congruence systems.

. Theorem 6.1

For integer $n > 1$, *the relation of congruence* (mod n) *is* reflexive, symmetric *and* transitive. *That is, for every* $a, b, c \in \mathbb{Z}$,

i. $a \equiv a \pmod{n}$;

ii. *If* $a \equiv b \pmod{n}$, *then* $b \equiv a \pmod{n}$;

iii. *If* $a \equiv b \pmod{n}$ *and* $b \equiv c \pmod{n}$, *then* $a \equiv c \pmod{n}$. □

A relation having the three properties in Theorem 6.1 is called an equivalence relation. It is well known that an equivalence relation over a set partitions the set into equivalence classes. Let us denote by "$\equiv_n$" the equivalence relation of congruence modulo n. This relation is defined over the set $\mathbb{Z}$, and therefore it partitions $\mathbb{Z}$ into exactly n equivalence classes, each class contains integers which are congruent to an integer modulo n. Let us denote these n classes by

$$\overline{0}, \overline{1}, \ldots, \overline{n-1},$$

where

Equation 6.2.1

$$\overline{a} = \{ x \in \mathbb{Z} \mid x \pmod{n} \equiv a \}.$$

We call each of them a residue class modulo n. Clearly, we can view

Equation 6.2.2

$$\mathbb{Z}_n = \{ \overline{0}, \overline{1}, \ldots, \overline{n-1} \}.$$

On the other hand, if we consider $\mathbb{Z}$ as a (trivial) subset of $\mathbb{Z}$, then coset $n\mathbb{Z}$ (Definition 5.7 in §5.2.1) is the set all integers which are multiples of n, i.e.,

Equation 6.2.3

$$n\mathbb{Z} = \{0, \pm n, \pm 2n, \dots\}.$$

Now consider quotient group (Definition 5.8 in §5.2.1) with addition as the group operation:

Equation 6.2.4

$$\mathbb{Z}/n\mathbb{Z} = \{\, x + n\mathbb{Z} \mid x \in \mathbb{Z} \,\}.$$

If we unfold (6.2.4) using $n\mathbb{Z}$ in (6.2.3), we have

Equation 6.2.5

$$\begin{aligned}
\mathbb{Z}/n\mathbb{Z} &= \{\, x + n\mathbb{Z} \mid x \in \mathbb{Z} \,\} \\
&= \{\, 0 + \{0, \pm n, \pm 2n, \dots\}, \\
&\quad\; 1 + \{0, \pm n, \pm 2n, \dots\}, \\
&\quad\; 2 + \{0, \pm n, \pm 2n, \dots\}, \\
&\quad\; \dots, \\
&\quad\; (n-1) + \{0, \pm n, \pm 2n, \dots\} \,\} \\
&= \{\, \{0, \pm n, \pm 2n, \dots\}, \\
&\quad\; \{1, \pm n + 1, \pm 2n + 1, \dots\}, \\
&\quad\; \{2, \pm n + 2, \pm 2n + 2, \dots\}, \\
&\quad\; \dots, \\
&\quad\; \{(n-1), \pm n + (n-1), \pm 2n + (n-1), \dots\} \,\}.
\end{aligned}$$

There are only *n* distinct elements in the structure (6.2.5). No more case is possible. For example

$$n + \{0, \pm n, \pm 2n, \dots\} = \{0, \pm n, \pm 2n, \dots\},$$

and

$$(n+1) + \{0, \pm n, \pm 2n, \dots\} = \{1, \pm n + 1, \pm 2n + 1, \dots\},$$

and so on. Comparing (6.2.2) and (6.2.5) with noticing the definition of $\bar{\mathbf{a}}$ in (6.2.1), we now know exactly that for $n > 1$:

$$\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z}.$$

$\mathbb{Z}/n\mathbb{Z}$ is the standard notation (in fact, the definition) for the residue classes modulo *n*, although for presentation convenience, in this book we will always use the short notation $\mathbb{Z}_n$ in place of $\mathbb{Z}/n\mathbb{Z}$.

. Theorem 6.2

For any a, b $\in \mathbb{Z}$, *define addition and multiplication between the residue classes* $\bar{a}$ *and* $\bar{b}$ *by*

$$\bar{a}+\bar{b}=\overline{a+b}, \qquad \bar{a}\cdot\bar{b}=\overline{ab}.$$

Then for any n > 1, *the mapping f:* $\mathbb{Z} \mapsto \mathbb{Z}_n$ *defined by* "(mod *n*)"*is a homomorphism from* $\mathbb{Z}$ *onto* $\mathbb{Z}_n$. □

6.2.1 Congruent Properties for Arithmetic in $\mathbb{Z}_n$

The homomorphism from $\mathbb{Z}$ onto $\mathbb{Z}_n$ means that arithmetic in $\mathbb{Z}_n$ (arithmetic modulo *n*) inheres the properties of arithmetic in $\mathbb{Z}$, as shown in the following theorem.

. Theorem 6.3

For integer n > 1, *if a* ≡ *b* (mod *n*)*and c* ≡ *d* (mod *n*),*then a* ± *c* ≡ *b* ± *d* (mod *n*)*and ac* ≡ *bd* (mod*n*).

Although the statements in this theorem hold trivially as an immediate result of the homomorphic relationship between $\mathbb{Z}$ and $\mathbb{Z}_n$, we provide a proof which is based purely on using the properties of arithmetic in $\mathbb{Z}_n$.

Proof If *n*|*a* – *b* and *n*|*c* – *d* then *n*|(*a* ± *c*) – (*b* ± *d*).

Also*n*|(*a* – *b*)(*c* – *d*) = (*ac* – *bd*) – *b*(*c* – *d*)(*c* – *d*) – *d*(*a* – *b*). So *n*|(*ac* – *bd*). □

The properties of the arithmetic in $\mathbb{Z}_n$ shown in Theorem 6.3 are called congruent properties, meaning performing the same calculation on both sides of an equation derives a new equation. However, Theorem 6.3 has left out division. Division in $\mathbb{Z}$ has the congruent property as follows:

Equation 6.2.6

$$\forall d \neq 0 : ad = bd \text{ implies } a = b.$$

The counterpart congruent property for division in $\mathbb{Z}_n$ will take a formula which is slightly different from (6.2.6). Before we find out what this formula is, let us provide an explanation on (6.2.6) in $\mathbb{Z}$. We may imagine that $\mathbb{Z}$ is the case of $\mathbb{Z}_n$ for $n = \infty$, and that ∞ is divisible by any integer and the resultant quotient is still ∞. Thus, we may further imagine that the first equation in (6.2.6) holds in terms of modulo ∞ while the second equation holds in terms of modulo ∞/d. Since $\infty/d = \infty$, the two equations in (6.2.6) take the same formula. This congruent property for division in $\mathbb{Z}$ is inhered into $\mathbb{Z}_n$ in the following formula.

. Theorem 6.4

For integer $n > 1$ *and* $d \neq 0$, *if* $ab \equiv bd \pmod{n}$ *then* $a \equiv b \pmod{\frac{n}{\gcd(d,n)}}$.

Proof Denote $k = \gcd(d, n)$. Then $n|(ad - bd)$ implies $(n/k)|(d/k)(a - b)$. Since $\gcd(d/k, n/k) = 1$, we know $(n/k)|(k/k)(a - b)$ implies $(n/k)|(a - b)$. □

To this end we know that the arithmetic in $\mathbb{Z}_n$ fully preserves the congruent properties of the arithmetic in $\mathbb{Z}$. Consequently, we have

. Corollary 6.1

If $f(x)$ *is a polynomial over* $\mathbb{Z}$, *and* $a \equiv b \pmod{n}$ *for* $n > 1$, *then* $f(a) \equiv f(b) \pmod{n}$. □

6.2.2 Solving Linear Congruence in $\mathbb{Z}_n$

In Theorem 4.2 (in §4.3.2.5) we have defined the multiplicative inverse modulo n and shown that for an integer a to have the multiplicative inverse modulo n, i.e., a unique number $x < n$ satisfying $ax \equiv 1 \pmod{n}$, it is necessary and sufficient for a to satisfy $\gcd(a, n) = 1$. The following theorem provides the condition for general case of solving linear congruence equation.

. Theorem 6.5

For integer $n > 1$, *a necessary and sufficient condition that the congruence*

Equation 6.2.7

$$ax \equiv b \pmod{n},$$

be solvable is that $\gcd(a, n)|b$.

Proof By Definition 4.4 (in §4.3.2.5), the congruence (6.2.7) is the linear equation

Equation 6.2.8

$$ax + kn = b,$$

for some integer *k*.

($\Longrightarrow$) Let (6.2.8) hold. Since gcd(*a*, *n*) divides the left-hand side, it must divide the right-hand side.

($\Longleftarrow$) For *a* and *n*, using Extended Euclid Algorithm (Alg 4.2) we can compute

$$a\lambda + \mu n = \gcd(a, n).$$

Since*b*/ gcd(*a*, *n*) is an integer, multiplying this integer to both sides, we obtain (6.2.8) or (6.2.7), where $x = \frac{\lambda b}{\gcd(a,n)} \pmod{n}$ is one solution.□

It is easy to check that given solution *x* for (6.2.7),

$$x + \frac{ni}{\gcd(a, n)} \pmod{n} \quad \text{for } i = 0, 1, 2, \ldots, \gcd(a, n) - 1$$

are gcd(*a*, *n*) different solutions less than *n*. Clearly, gcd(*a*, *n*) = 1 is the condition for the congruence (6.2.8) to have a unique solution less than *n*.

Example 6.1. Congruence

$$2x \equiv 5 \pmod{10}$$

is unsolvable since gcd(2, 10) = 2 $\nmid$ 5. In fact, the left-hand side, 2*x*, must be an even number, while the right-hand side, 10*k* + 5, can only be an odd number, and so trying to solve this congruence is an attempt to equalize an even number to an odd number, which is of course impossible.

On the other hand, congruence

$$6x \equiv 18 \pmod{36}$$

is solvable because gcd(6, 36)|18. The six solutions are 3, 9, 15, 21, 27, and 33.□

. Theorem 6.6

For integer n > 1, *if* gcd(*a, n) = 1, then ai* + *b* $\not\equiv$ *aj* + *b* (mod *n*)*for all b, i, j such that* 0 $\leq$ *i* < *j* <*n*.

Proof Suppose on the contrary $ai + b \equiv aj + b \pmod{n}$. Then by Theorem 6.4 we have $i \equiv j \pmod{n}$, a contradiction to $0 \leq i < j < n$. □

This property implies that for a, n satisfying $\gcd(a, n) = 1$, $ai + b \pmod{n}$ $(i = 0, 1, \ldots, n-1)$ is a complete residue system modulo n, that is, the expression $ai + b \pmod{n}$ ranges through $\mathbb{Z}_n$ for i ranging through $\mathbb{Z}_n$.

6.2.3 The Chinese Remainder Theorem

We have studied the condition for solving a single linear congruence in the form of (6.2.7). Often we will meet the problem of solving a system of simultaneous linear congruences with different moduli:

Equation 6.2.9

$$\begin{aligned} a_1 x &\equiv b_1 \pmod{n_1} \\ a_2 x &\equiv b_2 \pmod{n_2} \\ &\vdots \\ a_r x &\equiv b_r \pmod{n_r} \end{aligned}$$

where $a_i, b_i \in \mathbb{Z}$ with $a_i \neq 0$ for $i = 1, 2, \ldots, r$.

For this system of congruences to be solvable it is clearly necessary for each congruence to be solvable. So for $i = 1, 2, \ldots, r$ and denoting

$$d_i = \gcd(a_i, n_i),$$

by Theorem 6.5, it is necessary $d_i \mid b_i$. With this being the case, the congruent properties for multiplication (Theorem 6.3) and for division (Theorem 6.4) allow us to transform the system (6.2.9) into the following linear congruence system which is equivalent to but simpler than the system (6.2.9):

Equation 6.2.10

$$
\begin{aligned}
x &\equiv c_1 \pmod{m_1} \\
x &\equiv c_2 \pmod{m_2} \\
&\vdots \\
x &\equiv c_r \pmod{m_r}
\end{aligned}
$$

where for $i = 1, 2, \ldots, r$:

$$m_i = n_i/d_i$$

and

$$c_i = (b_i/d_i)(a_i/d_i)^{-1} \pmod{m_i}.$$

Notice that $(a_i/d_i)^{-1} \pmod{m_i}$ exists since $\gcd(a_i/d_i, m_i) = 1$ (review Theorem 4.2 in §4.3.2.5).

In linear algebra, the system (6.2.10) can be represented by the following vectorspace version:

Equation 6.2.11

$$A\vec{X} = \vec{C}$$

where

Equation 6.2.12

$$A = \begin{pmatrix} \overline{1}_{m_1} & & & & \\ & \overline{1}_{m_2} & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \overline{1}_{m_r} \end{pmatrix},$$

Equation 6.2.13

$$\vec{X} = \begin{pmatrix} x \\ x \\ \cdot \\ \cdot \\ \cdot \\ x \end{pmatrix},$$

Equation 6.2.14

$$\vec{C} = \begin{pmatrix} c_1 \\ c_2 \\ \cdot \\ \cdot \\ \cdot \\ c_r \end{pmatrix}.$$

Notice that because the *i*-th equation (for *i* = 1, 2, …, *r*) in the congruence system (6.2.10) holds modulom_i, in the diagonal part of the the matrix *A*, $\overline{1}_{m_i}$ denotes the residue class 1 modulo m_i, that is,

Equation 6.2.15

$$\overline{1}_{m_i} = k_i m_i + 1$$

for some integer k_i (*i* = 1, 2, …, *r*). The blank part of the matrix *A* represents 0 modulo respective modulus (i.e., zeros in the *i* row are means zeros modulo m_i).

Thus, given any *r*-dimension vector $\vec{C}$ the problem of solving the system (6.2.10), or its vector-space version (6.2.11), boils down to that of identifying the diagonal matrix *A*, or in other words, finding the residue class 1 modulo m_i as required in (6.2.15) for *i* = 1, 2, …, *r*. We know from a fact in linear algebra that if the matrix *A* exists, then because none of the elements in its diagonal line is zero, the matrix has the full rank *r* and consequently, there *exists* a *unique* solution.

When the moduli in (6.2.10) are pairwise relatively prime to each other, it is not difficult to find a system of residue classes 1. This is according to the useful Chinese Remainder Theorem (CRT).

. Theorem 6.7 Chinese Remainder Theorem

For the linear congruence system (6.2.10), if gcd(m_i, m_j) = 1 *for* 1 $\leq i < j \leq r$,*then there exists* $\overline{1}_{m_i}$ *satisfying*

Equation 6.2.16

$$\overline{1}_{m_i} \equiv 0 \pmod{m_j}.$$

Consequently, there exists $x \in \mathbb{Z}_m$ *as the unique solution to the system* (6.2.10) *where* $M = m_1 m_2 \ldots m_r$.

Proof We prove first the existence and then the uniqueness of the solution.

Existence For each $i = 1, 2, \ldots, r$, $\gcd(m_i, M/m_i) = 1$. By Theorem 4.2 (§4.3.2.5), there exists $y_i \in y_i \in \mathbb{Z}_{m_i}$ satisfying

Equation 6.2.17

$$(M/m_i)y_i \equiv 1 \pmod{m_i}.$$

Moreover, for $j \neq i$, because $m_j | (M/m_i)$, we have

Equation 6.2.18

$$(M/m_i)y_i \equiv 0 \pmod{m_j}.$$

So $(M/m_i)y_i$ is exactly the number that we are looking for to play the role of $\overline{1}_{m_i}$. Let

Equation 6.2.19

$$x \leftarrow \sum_{i=1}^{r} \overline{1}_{m_i} c_i \pmod{M}.$$

Then x is a solution to the system (6.2.10) and is a residue class modulo M.

Uniqueness View the linear system defined by (6.2.11), (6.2.12), (6.2.13) and (6.2.14) such that the elements of the matrix A and those of the vector $\vec{C}$ are all in $\mathbb{Z}$ (i.e., they are all integers). Notice that in $\mathbb{Z}$

Equation 6.2.20

$$\det(A) = \overline{1}_{m_1} \overline{1}_{m_2} \cdots \overline{1}_{m_r} \neq 0.$$

This means that the r columns (vectors) of the matrix A form a basis for the r-dimension vector space $\underbrace{\mathbb{Z} \times \mathbb{Z} \times \cdots \times \mathbb{Z}}_{r}$ (this basis is similar to a so-called "natural basis" in linear algebra where the only non-zero element in any basis-vector is 1). Therefore, for any vector $\vec{C} \in \underbrace{\mathbb{Z} \times \mathbb{Z} \times \cdots \times \mathbb{Z}}_{r}$, $\vec{X} \in \underbrace{\mathbb{Z} \times \mathbb{Z} \times \cdots \times \mathbb{Z}}_{r}$. We have seen in the existence part of the proof that the unique elements of $\vec{X}$ are given by (6.2.19). □

The proof of Theorem 6.7 is constructive, that is, we have constructed an algorithm for finding the solution to the system (6.2.10). This algorithm is now specified in Alg 6.1.

Algorithm 6.1: Chinese Remainder

INPUT	integer tuple $(m_1, m_2, \ldots, m_r)$, pairwise relatively prime;
	integer tuple $(c_1 \pmod{m_1}, c_2 \pmod{m_2}, \ldots, c_r \pmod{m_r})$.
OUTPUT	integer $x < M = m_1 m_2 \ldots m_r$ satisfying the system (6.2.10).

1. $M \longleftarrow m_1 m_2 \ldots m_r$;
2. for (i from 1 to r) do
   a. $y_i \longleftarrow (M/m_i)^{-1} \pmod{m_i}$; (* by Extended Euclid Algorithm *)
   b. $\overline{1}_{m_i} \longleftarrow y_i M/m_i$;
3. return($\sum_{i=1}^{r} \overline{1}_{m_i} c_i \pmod{M}$).

In Alg 6.1, the only time-consuming part is in step 2(a) where a multiplicative inversion of a large number is computed. This can be done by applying the Extended Euclid Algorithm (Alg 4.2). Considering $m_i < M$ for $i = 1, 2, \ldots, r$, the time complexity of Alg 6.1 is $O_B(r(\log M)^2)$.

It is also easy to see the following results from Theorem 6.7:

i. every $x \in \mathbb{Z}_m$ yields a vector $\vec{C} \in \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_r}$; from (6.2.19) we can see

that the elements in $\vec{C}$ are computed by (for i = 1, 2, ..., r)

$$c_i \leftarrow x \pmod{m_i};$$

ii. in particular, 0 and 1 in $\mathbb{Z}_m$ yield $\vec{0}$ and $\vec{1}$ in $\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_r}$, respectively;

iii. for x, x' yielding $\begin{pmatrix} c_1 \\ c_2 \\ \cdot \\ \cdot \\ \cdot \\ c_r \end{pmatrix}, \begin{pmatrix} c'_1 \\ c'_2 \\ \cdot \\ \cdot \\ \cdot \\ c'_r \end{pmatrix}$, respectively, $x \cdot x'$ yields

$$\begin{pmatrix} c_1 \cdot c'_1 \pmod{m_1} \\ c_2 \cdot c'_2 \pmod{m_2} \\ \cdot \\ \cdot \\ \cdot \\ c_r \cdot c'_r \pmod{m_r} \end{pmatrix}.$$

Thus, we have also proven the following theorem (following Definition 5.16):

. Theorem 6.8

If gcd(m_i, m_j) = 1 *for* 1 $\leq i < j \leq r$, *then for* $M = m_1 m_2 \ldots m_r$, $\mathbb{Z}_m$ *is isomorphic to* $\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_r}$, *and the isomorphism*

$$f : \mathbb{Z}_M \mapsto \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_r}$$

is

$$f(x) = (x \pmod{m_1},\ x \pmod{m_2},\ \ldots,\ x \pmod{m_r}).$$

Theorem 6.8 is very useful in the study of cryptographic systems or protocols which use groups modulo composite integers. In many places in the rest of this book we will need to make use of the isomorphism between $\mathbb{Z}_n^*$ and $\mathbb{Z}_p^* \times \mathbb{Z}_q^*$ where $n = pq$ with p, q prime numbers. For example, we will make use of a property that the non-cyclic group $\mathbb{Z}_n^*$ is generated by two generators of the cyclic groups $\mathbb{Z}_p^*$ and $\mathbb{Z}_q^*$, respectively.

Let us now look at an application of the Chinese Remainder Theorem: a calculation is made easy by applying the isomorphic relationship.

Example 6.2.

At this stage we do not yet know how to compute square root modulo an integer (we will study the techniques in §6.6). However in some cases a square number in some space (such as in $\mathbb{Z}$) is evident and so square rooting in that space is easy without need of using modulo arithmetic. Let us apply Theorem 6.8 to compute one of the square roots of 29 in $\mathbb{Z}_{35}$.

Limited to our knowledge for the moment, it is not evident to us that 29 is a square number in $\mathbb{Z}_{35}$ and so for the time being we do not know how to root it directly. However, if we apply Theorem 6.8 and map 29 to the isomorphic space $\mathbb{Z}_5$ x $\mathbb{Z}_7$, we have

$$29 \pmod 5 \mapsto 4, \quad 29 \pmod 7 \mapsto 1,$$

that is, the image is (4, 1). Both 4 and 1 are evident square numbers with 2 being a square root of 4 and 1 being a square root of 1. By isomorphism, we know one of the square roots of 29 in $\mathbb{Z}_{35}$ corresponds to (2, 1) in $\mathbb{Z}_5$ x $\mathbb{Z}_7$. Applying the Chinese Remainder Algorithm (Alg 6.1), we obtain

$$\overline{1}_5 = 21, \quad \overline{1}_7 = 15,$$

and

$$\sqrt{29} \equiv 21 \cdot 2 + 15 \cdot 1 \equiv 22 \pmod{35}.$$

Indeed, $22^2 = 484 \equiv 29 \pmod{35}$. □

As a matter of fact, 29 has four distinct square roots in $\mathbb{Z}_{35}^*$. For an exercise, the reader may find the other three square roots of 29 (Exercise 6.4).

6.3 Euler's Phi Function

In §5.2.3 we have defined Euler's phi function in Definition 5.11. Now let us study some useful properties of it.

. Lemma 6.1

Let $\phi(n)$ be Euler's phi function defined in Definition 5.11. Then

i. $\phi(1) = 1$.

ii. *If p is prime then $\phi(p) = p - 1$.*

iii. *Euler's phi function is multiplicative. That is, if* $\gcd(m, n) = 1$, *then* $\phi(mn) = \phi(m)\phi(n)$.

iv. $If\ n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$ *is the prime factorization of n, then*

$$\phi(n) = n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)\cdots\left(1 - \frac{1}{p_k}\right).$$

Proof (i) and (ii) are trivial from Definition 5.11.

iii) Since $\phi(1) = 1$, the equation $\phi(mn) = \phi(m)\phi(n)$ holds when either $m = 1$ or $n = 1$. So suppose $m > 1$ and $n > 1$. For $\gcd(m, n) = 1$, consider the array

Equation 6.3.1

$$\begin{array}{ccccc} 0 & 1 & 2 & \cdots & m-1 \\ m & m+1 & m+2 & \cdots & m+(m-1) \\ . & . & . & \cdots & . \\ . & . & . & \cdots & . \\ (n-1)m & (n-1)m+1 & (n-1)m+2 & \cdots & (n-1)m+(m-1) \end{array}$$

On the one hand, (6.3.1) consists of mn consecutive integers, so it is all the numbers modulo mn and therefore contains $\phi(mn)$ elements prime to mn.

On the other hand, observe (6.3.1). The first row is all the numbers modulo m, and all the elements in any column are congruent modulo m. So there are $\phi(m)$ columns consisting entirely of integers prime to m. Let

$$b, m+b, 2m+b, \ldots, (n-1)m+b$$

be any such column of n elements. With $\gcd(m, n) = 1$, by Theorem 6.6, such a column is a

complete residue system modulo n. So in each such column there are $\phi(n)$ elements prime to n. To this end we know that in (6.3.1) there are $\phi(m)\phi(n)$ elements prime to both m and n. Further notice that any element prime to both m and to n if and only if it is prime to mn.

Combining the results of the above two paragraphs, we have derived $\phi(mn) = \phi(m)\phi(n)$.

iv) For any prime p, in 1, 2, …, p^e, the elements which are not prime to p^e are the multiples of p, i.e.,p, $2p$, …, $p^{e-1}p$. Clearly, there are exactly p^{e-1} such numbers. So

$$\phi(p^e) = p^e - p^{e-1} = p^e\left(1 - \frac{1}{p}\right).$$

This holds for each prime power $p^e|n$ with $p^{e+1} \nmid n$. Noticing that different such prime powers of n are relatively prime to each other, the targeted result follows from (iii). □

In §4.5 we considered a problem named SQUARE-FREENESS: answering whether a given odd composite integer n is square free. Three we used $\phi(n)$ to serve an auxiliary input to show that SQUARE-FREENESS is in $\mathcal{NP}$. Now from Property (iv) of Lemma 6.1 we know that for any prime$p > 1$, if $p^2|n$ then $p|\phi(n)$. This is why we used $\gcd(n,\phi(n)) = 1$ as a witness for n being square free. The reader may consider the case $\gcd(n,\phi(n)) > 1$ (be careful of the case, e.g., $n = pq$ with $p|\phi(q)$, see Exercise 6.5).

Euler's phi function has the following elegant property.

. Theorem 6.9

$$\textit{For integer } n > 0, \sum_{d|n} \phi(d) = n.$$

Proof Let $S_d = \{ x \mid 1 \le x \le n, \gcd(x, n) = d\}$. It is clear that set $S = \{1, 2, \ldots, n\}$ is partitioned into disjoint subsets S_d for each $d|n$. Hence

$$\bigcup_{d|n} S_d = S.$$

Notice that for each $d|n$ $\#S_d = \phi(n/d)$, therefore

$$\sum_{d|n} \phi(n/d) = n.$$

However, for any $d|n$, we have $(n/d)|n$, therefore

$$\sum_{d|n} \phi(n/d) = \sum_{(n/d)|n} \phi(n/d) = \sum_{d|n} \phi(d).$$

□

Example 6.3.

For $n = 12$, the possible values of $d|12$ are 1, 2, 3, 4, 6, and 12. We have $\phi(1) + \phi(2) + \phi(3) + \phi(4) + \phi(6) + \phi(12) = 1 + 1 + 2 + 2 + 2 + 4 = 12$. □

6.4 The Theorems of Fermat, Euler and Lagrange

We have introduced Fermat's Little Theorem in Chapter 4 (congruence (4.4.8)) and have since used it for a few times but without having proved it. Now we prove Fermat's Little Theorem by showing that it is a special case of another famous theorem in number theory: Euler's Theorem.

. Theorem 6.10 Fermat's Little Theorem

If p is prime and $p \nmid a$, then $a^{p-1} \equiv 1 \pmod{p}$.

Since $\phi(p) = p - 1$ for p being prime, Fermat's Little Theorem is a special case of the following theorem.

. Theorem 6.11 Euler's Theorem

If $\gcd(a, n) = 1$ *then* $a^{\phi(n)} \equiv 1 \pmod{n}$.

Proof For $\gcd(a, n) = 1$, we know $a \pmod{n} \in \mathbb{Z}_n^*$. Also $\#\mathbb{Z}_n^*$. By Corollary 5.2, we have $\mathrm{ord}_n(a) \mid \#\mathbb{Z}_n^*$ which implies $a^{\phi(n)} \equiv 1 \pmod{n}$. □

SinceCorollary 5.2 used in the proof of Theorem 6.11 is a direct application of Lagrange's Theorem (Theorem 5.1), we therefore say that Fermat's Little Theorem and Euler's Theorem are special cases of the beautiful Theorem of Lagrange.

InChapter 4 we have seen the important role of Fermat's Little Theorem in probabilistic primality test, which is useful for the generation of key material for many public-key cryptographic systems and protocols. Euler's Theorem will have an important application for the RSA cryptosystem which will be introduced in §8.5

6.5 Quadratic Residues

Quadratic residues play important roles in number theory. For example, integer factorization algorithms invariantly involve using quadratic residues. They also have frequent uses in encryption and interesting cryptographic protocols.

Definition 6.1: Quadratic Residue*Let integer n* > 1. *For a* $\in \mathbb{Z}_n^*$, *a is called a quadratic residue modulo n if* $x^2 \equiv a \pmod{n}$ *for some* $x \in \mathbb{Z}_n$; *otherwise, a iscalled a quadratic non-residue modulo n. The set of quadratic residues modulo n is denoted by* QR_n,*and the set of quadratic non-residues modulo n is denoted by* QNR_n.

Example 6.4.

Let us compute QR_{11}, the set of all quadratic residues modulo 11. $QR_{11} = \{ 1^2, 2^2, 3^2, 4^2, 5^2, 6^2, 7^2, 8^2, 9^2, 10^2 \} \pmod{11} = \{ 1, 3, 4, 5, 9 \}$. □

In this example, we have computed QR_{11} by exhaustively squaring elements in $\mathbb{Z}_{11}^*$. However, this is not necessary. In fact, the reader may check

$$QR_{11} = \{ 1^2, 2^2, 3^2, 4^2, 5^2 \} \pmod{11},$$

i.e., exhaustively squaring elements up to half the magnitude of the modulus suffices. The following theorem claims so for any prime modulus.

. Theorem 6.12

Let p be a prime number. Then

i. $QR_p = \{ x^2 \pmod{p} \mid 0 < x \leq (p-1)/2 \}$;

ii. *There are precisely* $(p-1)/2$ *quadratic residues and* $(p-1)/2$ *quadratic non-residues modulo p, that is,* $\mathbb{Z}_p^*$ *is partitioned into two equal-size subsets* QR_p *and* QNR_p.

Proof (i) Clearly, set $S = \{ x^2 \pmod{p} \mid 0 < x \leq (p-1)/2 \} \subseteq QR_p$. To show $QR_p = S$ we only need to prove $QR_p \subseteq S$.

Let any $a \in QR_p$. Then $x^2 \equiv a \pmod{p}$ for some $x < p$. If $x \leq (p-1)/2$ then $a \in S$. Suppose $x > (p-1)/2$. Then $y = p - x \leq (p-1)/2$ and $y^2 \equiv (p-x)^2 \equiv p^2 - 2px + x^2 \equiv x^2 \equiv a \pmod{p}$. So $QR_p \subseteq S$.

ii) To show $\#QR_p = (p-1)/2$ it suffices to show that for $0 < x < y \leq (p-1)/2, x^2 \not\equiv y^2 \pmod{p}$. Suppose on the contrary, $x^2 - y^2 \equiv (x+y)(x-y) \equiv 0 \pmod{p}$. Then $p \mid x+y$ or $p \mid x-y$. Only the latter case is possible since $x + y < p$. Hence $x = y$, a contradiction.

Then #QNR$_p$ = (p–1)/2 since $\mathrm{QNR}_p = \mathbb{Z}_p^* \setminus \mathrm{QR}_p$ and $\#\mathbb{Z}_p^* = p - 1$. □

In the proof of Theorem 6.12(i) we have actually shown the following:

. Corollary 6.2

Let p be a prime number. Then for any a ∈ QR$_p$,*there are exactly two square roots of a modulo p. Denoting by x one to them, then the other is –x (= p – x).* □

6.5.1 Quadratic Residuosity

Often we need to decide if a number is a quadratic residue element modulo a given modulus. This is the so-called quadratic residuosity problem.

. Theorem 6.13 Euler's Criterion

Let p be a prime number. Then for any $x \in \mathbb{Z}_p^*$, *x* ∈ QR$_p$*if and only if*

Equation 6.5.1

$$x^{(p-1)/2} \equiv 1 \pmod{p}.$$

Proof (⟹) For $x \in \mathrm{QR}_p$, there exists $y \in \mathbb{Z}_p^*$ such that $y^2 \equiv x \pmod{p}$. So $x^{(p-1)/2} \equiv y^{p-1} \equiv 1 \pmod{p}$ follows from Fermat's Theorem (Theorem 6.10).

(⟸) Let $x^{(p-1)/2} \equiv 1 \pmod{p}$. Then x is a root of polynomial $y^{(p-1)/2} - 1 \equiv 0 \pmod{p}$. Notice that $\mathbb{Z}_p$ is a field, by Theorem 5.9(iii) (in §5.4.3) every element in the field is a root of the polynomial $y^p - y \equiv 0 \pmod{p}$. In other words, every non-zero element of the field, i.e., every element in the group $\mathbb{Z}_p^*$ is a root of

$$y^{p-1} - 1 \equiv (y^{(p-1)/2} - 1)(y^{(p-1)/2} + 1) \equiv 0 \pmod{p}.$$

These roots are all distinct since this degree-(p – 1) polynomial can have at most p – 1 roots. Consequently, the (p – 1)/2 roots of polynomial $y^{(p-1)/2} - 1 \equiv 0 \pmod{p}$ must all be distinct. We have shown in Theorem 6.12 that QR$_p$ contains exactly (p – 1)/2 elements, and they all satisfy $y^{(p-1)/2} - 1 \equiv 0 \pmod{p}$. Any other element in $\mathbb{Z}_p^*$ must satisfy $y^{(p-1)/2} + 1 \equiv 0 \pmod{p}$. Therefore$x$ ∈ QR$_p$. □

In the proof of Theorem 6.13 we have shown that if the criterion is not met for $x \in \mathbb{Z}_p$, then

Equation 6.5.2

$$x^{(p-1)/2} \equiv -1 \pmod{p}.$$

Euler's Criterion provides a criterion to test whether or not an element in $\mathbb{Z}_p^*$ is a quadratic residue: if congruence (6.5.1) is satisfied, then $x \in QR_p$; otherwise (6.5.2) is satisfied and $x \in QNR_p$.

Let n be a composite natural number with its prime factorization as

Equation 6.5.3

$$n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}.$$

Then by Theorem 6.8, $\mathbb{Z}_n$ is isomorphic to $\mathbb{Z}_{p_1^{e_1}} \times \mathbb{Z}_{p_2^{e_2}} \times \cdots \times \mathbb{Z}_{p_k^{e_k}}$. Since isomorphism preserves arithmetic, we have:

. Theorem 6.14

Let n be a composite integer with complete factorization in (6.5.3). Then $x \in QR_n$ *if and only if* $x \pmod{p_i^{e_i}} \in \mathrm{QR}_{p_i^{e_i}}$ *and hence if and only if* $x \pmod{p_i} \in QR_{p_i}$ *for prime* p_i *with* $i = 1, 2, \ldots, k$. □

Therefore, if the factorization of n is known, given $x \in \mathbb{Z}_n^*$, the quadratic residuosity of x modulo n can be decided by deciding the residuosity of $x \pmod{p}$ for each prime $p|n$. The latter task can be done by testing Euler's criterion.

However, if the factorization of n is unknown, deciding quardratic residuosity modulo n is a non-trivial task.

Definition 6.2:Quadratic Residuosity (QR) Problem

INPUT	*n: a composite number*; $x \in \mathbb{Z}_n^*$.
OUTPUT	YES*if* $x \in QR_n$.

The QRP is a well-known hard problem in number theory and is one of the main four algorithmic problems discussed by Gauss in his "Disquisitiones Arithmeticae" [119]. An efficient solution for it would imply an efficient solution to some other open problems in number theory. In Chapter 14 we will study a well-known public-key cryptosystem named the Goldwasser-Micali cryptosystem; that cryptosystem has its security based on the difficult for deciding the QRP.

Combining Theorem 6.12 and Theorem 6.14 we can obtain:

. Theorem 6.15

Let n be a composite integer with k > 1 distinct prime factors. Then exactly $\frac{1}{2^k}$ *fraction of elements in* $\mathbb{Z}_n^*$ *are quadratic residues modulo n.* □

Thus, for a composite number *n*, an efficient algorithm for deciding quadratic residuosity modulo *n* will provide an efficient statistic test on the proportion of quadratic residues in $\mathbb{Z}_n^*$, and hence by Theorem 6.15, provide an efficient algorithm for answering the question whether *n* has two or three distinct prime factors. This is because, by Theorem 6.15, in the former case (*n* has two distinct prime factors), exactly a quarter of elements in $\mathbb{Z}_n^*$ are quadratic residues, and in the latter case, exactly one-eighth of them are. Consequently, ensembles $E_{2\text{–Prime}}$ and $E_{3\text{–Prime}}$ (see §4.7) can be distinguished.

To date, for a composite *n* of unknown factorization, no algorithm is known to be able to decide quadratic residuosity modulo *n* in time polynomial in the size of *n*.

6.5.2 Legendre-Jacobi Symbols

Testing quadratic residuosity modulo a prime using Euler's criterion (6.5.1) involves evaluating modulo exponentiation which is quite computation intensive. However, quadratic residuosity can be tested by a much faster algorithm. Such an algorithm is based on the notion of Legendre-Jacobi symbol.

Definition 6.3: Legendre-Jacobi Symbol *For each prime number p and for any* $x \in \mathbb{Z}_p^*$ *let*

$$\left(\frac{x}{p}\right) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } x \in \mathrm{QR}_p \\ -1 & \text{if } x \in \mathrm{QNR}_p. \end{cases}$$

$\left(\frac{x}{p}\right)$ *is called Legendre symbol of x modulo p.*

Let $n = p_1 p_2 \ldots p_k$ *be the prime factorization of n (some of these prime factors may repeat). Then*

$$\left(\frac{x}{n}\right) \stackrel{\text{def}}{=} \left(\frac{x}{p_1}\right)\left(\frac{x}{p_2}\right)\cdots\left(\frac{x}{p_k}\right)$$

is called Jacobi symbol of x modulo n.

In the rest of this book $\left(\frac{a}{b}\right)$ will always be referred to as Jacobi symbol whether or not *b* is prime.

For p being prime, comparing (6.5.1), (6.5.2) with Definition 6.3, we know

Equation 6.5.4

$$\left(\frac{x}{p}\right) = x^{(p-1)/2} \pmod{p}.$$

Moreover, Jacobi symbol has the following properties.

. Theorem 6.16

Jacobi symbol has the following properties:

i. $\left(\frac{1}{n}\right) = 1$;

ii. $\left(\frac{xy}{n}\right) = \left(\frac{x}{n}\right)\left(\frac{y}{n}\right)$;

iii. $\left(\frac{x}{mn}\right) = \left(\frac{x}{m}\right)\left(\frac{x}{n}\right)$;

iv. *if* $x \equiv y \pmod{n}$ *then* $\left(\frac{x}{n}\right) = \left(\frac{y}{n}\right)$; (*below* m, n *are odd numbers*)

v. $\left(\frac{-1}{n}\right) = (-1)^{(n-1)/2}$;

vi. $\left(\frac{2}{n}\right) = (-1)^{(n^2-1)/8}$;

vii. *if* $\gcd(m, n) = 1$ *and* $m, n > 2$ *then* $\left(\frac{m}{n}\right)\left(\frac{n}{m}\right) = (-1)^{(m-1)(n-1)/4}$.

In Theorem 6.16, (i–iv) are immediate from the definition of Jacobi symbol. A proof for (v–vii) uses no special technique either. However, due to the lengthiness and lack of immediate relevance to the topic of this book, we shall not include a proof but refer the reader to the standard textbooks for number theory (e.g., [170,176]).

Theorem 6.16(vii) is known as the Gauss' Law of Quadratic Reciprocity. Thanks to this law, it is not hard to see that the evaluation of $\left(\frac{x}{n}\right)$ for gcd $(x, n) = 1$ has a fashion and hence the same

computational complexity of computing the greatest common divisor.

. Remark 6.1

When we evaluate Jacobi symbol by applying Theorem 6.16, the evaluation of the right-hand sides of (v–vii) must not be done via exponentiations. Since ord(–1) = 2 *(in multiplication), all we need is the parity of these exponents. In Alg 6.2 we realize the evaluation by testing whether* 2*divides these exponents*. □

Alg 6.2 provides a recursive specification of the properties of Jacobi symbol listed in Theorem 6.2.

Algorithm 6.2: Legendre/Jacobi Symbol

INPUT odd integer $n > 2$, integer $x \in \mathbb{Z}_n^*$.

OUTPUT $\left(\frac{x}{n}\right)$.

Jacobi(x, n)

1. if ($x == 1$) return (1);
2. if ($2|x$)
   a. if ($2|(n^2-1)/8$ return (Jacobi($x/2$,n));
   b. return(–Jacobi($x/2$,n));

   (* now x is odd *)
3. if ($2|$ $(x-1)(n-1)/4$) return(Jacobi(n mod x, x));
4. return(–Jacobi(n mod x, x)).

InAlg 6.2, each recursive call of the function Jacobi(,) will cause either the first input value being divided by 2, or the second input value being reduced modulo the first. Therefore there can be at most $\log_2 n$ calls and the first input value is reduced to 1, reaching the terminating condition. So rigorously expressed, because each modulo operation costs $O_B((\log n)^2)$ time, Alg 6.2 computes $\left(\frac{x}{n}\right)$ can be computed in $O_B((\log n)^3)$ time.

However we should notice that, in order to present the algorithm with ease of understanding, we have again chosen to sacrifice efficiency!

Instead of bounding each modulo operation with $O_B((\log n)^2)$, via a careful realization, *total* modulo operations in steps 3, 4 can be bounded by $O_B((\log n)^2)$. This situation is exactly the same as that for computing greatest common divisor with a carefully designed algorithm: to exploit the fact expressed in (4.3.12). Consequently, for $x \in \mathbb{Z}_n^*$, $\left(\frac{x}{n}\right)$ can be computed in $O_B((\log n)^2)$ time. A careful realization of the counterpart for Alg 6.2 can be found in Chapter 1 of [79].

Compared with the complexity of evaluating Euler's criterion (5.4.5), which is $O_B((\log p)^3)$ due to modulo exponentiation, testing quadratic residuosity modulo prime p using Alg 6.2 is $\log p$ times faster.

Example 6.5.

Let us show that $384 \in \mathrm{QNR}_{443}$.

Going through Alg 6.2 step by step, we have

$$\begin{aligned}
\mathrm{Jacobi}(384, 443) &= -\mathrm{Jacobi}(192, 443) \\
&= \mathrm{Jacobi}(96, 443) \\
&= -\mathrm{Jacobi}(48, 443) \\
&= \mathrm{Jacobi}(24, 443) \\
&= -\mathrm{Jacobi}(12, 443) \\
&= \mathrm{Jacobi}(6, 443) \\
&= -\mathrm{Jacobi}(3, 443) \\
&= \mathrm{Jacobi}(2, 3) \\
&= -\mathrm{Jacobi}(1, 3) \\
&= -1.
\end{aligned}$$

Therefore $384 \in \mathrm{QNR}_{443}$. □

Finally, we should notice that evaluation of Jacobi symbol $\left(\frac{x}{n}\right)$ using Alg 6.2 does not need to know the factorization of n. This is a very important property which has a wide application in public-key cryptography, e.g., in Goldwasser-Micali cryptosystem (§14.3.3) and in Blum's coin-flipping protocol (Chapter 19).

6.6 Square Roots Modulo Integer

In Example 6.2 we have had an experience of "computing a square root modulo an integer." However the "algorithm" used there should not qualify as an algorithm because we were lucky to have managed to map, using the isomorphism in Theorem 6.8, a seemingly difficult task to two trivially easy ones: computing square roots of 1 and 4, which happen to be square numbers in $\mathbb{Z}$ and the "rooting algorithm" is known even to primary school pupils. In general, the isomorphism in Theorem 6.8 will not be so kind to us: for overwhelming cases the image should not be a square number in $\mathbb{Z}$.

Now we introduce algorithmic methods for computing square roots of a quadratic residue element modulo a positive integer. We start by considering prime modulus. By Corollary 6.2, the two roots of a quadratic residue complements to one another modulo the prime modulus; so it suffices for us to consider computing one square root of a quadratic residue element.

For most of the odd prime numbers, the task is very easy. These cases include primes p such that $p \equiv 3, 5, 7 \pmod 8$.

6.6.1 Computing Square Roots Modulo Prime

Case $p \equiv 3, 7 \pmod 8$

In this case, $p + 1$ is divisible by 4. For $a \in \mathrm{QR}_p$, let

$$x \stackrel{\text{def}}{=} a^{(p+1)/4} \pmod p.$$

Then because $a^{(p-1)/2} \equiv 1 \pmod p$, we have

$$x^2 \equiv a^{(p+1)/2} \equiv a^{(p-1)/2}\, a \equiv a \pmod p$$

So indeed, x is a square root of a modulo p.

Case $p \equiv 5 \pmod 8$

In this case, $p + 3$ is divisible by 8; also because $(p - 1)/2$ is even, -1 meets Euler's criterion as a quadratic residue. For $a \in \mathrm{QR}_p$, let

Equation 6.6.1

$$x \stackrel{\text{def}}{=} a^{(p+3)/8} \pmod p.$$

From $a^{(p-1)/2} \equiv 1 \pmod p$ we know $a^{(p-1)/4} \equiv \pm 1 \pmod p$; this is because in field $\mathbb{Z}_p^*$ 1 has only two square roots: 1 and -1. Consequently

$$x^2 \equiv a^{(p+3)/4} \equiv a^{(p-1)/4}\, a \equiv \pm a \pmod{p}.$$

That is, we have found that *x* computed in (6.6.1) is a square root of either *a* or –*a*. If the sign is + we are done. If the sign is –, then we have

$$-x^2 \equiv (\sqrt{-1}\, x)^2 \equiv a \pmod{p}.$$

Therefore

Equation 6.6.2

$$x \stackrel{\text{def}}{=} \sqrt{-1}\, a^{(p+3)/8} \pmod{p}$$

will be the solution. So the task boils down to computing $\sqrt{-1} \pmod{p}$. Let *b* be any quadratic non-residue mod *p*. Then by Euler's criterion

$$(b^{(p-1)/4})^2 \equiv b^{(p-1)/2} \equiv -1 \pmod{p},$$

so $b^{(p-1)/4} \pmod{p}$ can be used in place of $\sqrt{-1}$. By the way, since

$$p^2 - 1 = (p+1)(p-1) = (8k+6)(8k+4) = 8(4k'+3)(2k''+1),$$

and the right-hand side is 8 times an odd number; so by Theorem 6.16(vi) $2 \in \mathrm{QNR}_p$. That is, for this case of *p* we can use $2^{(p-1)/4}$ in place of $\sqrt{-1}$. Then, one may check that (6.6.2) becomes

Equation 6.6.3

$$2^{(p-1)/4}\, a^{(p+3)/8} \equiv (4a)^{(p+3)/8}/2 \pmod{p}.$$

We can save one modulo exponentiation by using the right-hand-side of (6.6.3).

Algorithm 6.3: Square Root Modulo $p \equiv 3, 5, 7 \pmod 8$

INPUT	prime p satisfying $p \equiv 3, 5, 7 \pmod 8$; integer $a \in QR_p$.
OUTPUT	a square root of a modulo p.

1. if ($p \equiv 3, 7 \pmod 8$) return($a^{(p+1)/4} \pmod p$);

   (* below $p \equiv 5 \pmod 8$ *)

2. if ($a^{(p-1)/4} \equiv 1 \pmod p$) return($a^{(p+3)/8} \pmod p$);

3. return($(4a)(^{(p+3)/8}/2)$.

The time complexity of Alg 6.3 is $O_B((\log p)^3)$.

Computing Square Roots Modulo Prime in General Case

The method described here is due to Shanks (see §1.5.1 of [79]).

For general case of prime p, we can write

$$p - 1 = 2^e q$$

with q odd and $e \geq 1$. By Theorem 5.2 (in §5.2.3), cyclic group $\mathbb{Z}_p^*$ has a unique cyclic subgroup G of order 2^e. Clearly, quadratic residues in G have orders as powers of 2 since they divide 2^{e-1}. For $a \in QR_p$, since

$$a^{(p-1)/2} \equiv (a^q)^{2^{e-1}} \equiv 1 \pmod p,$$

so $a^q \pmod p$ is in G and is of course a quadratic residue. So there exists an even integer k with $0 \geq k > 2^e$ such that

Equation 6.6.4

$$a^q g^k \equiv 1 \pmod p,$$

where g is a generator of G. Suppose that we have found the generator g and the even integer k.

Then setting

$$x \stackrel{\text{def}}{=} a^{(q+1)/2} g^{k/2},$$

it is easy to check that $x^2 \equiv a \pmod{p}$.

Thus, the task boils down to two sub-tasks: (i) finding a generator g of group G, and (ii) finding the least non-negative even integer k, such that (6.6.4) is satisfied.

Sub-task (i) is rather easy. For any $f \notin \text{QNR}_p$, because q is odd, $f^q \notin \text{QNR}_p$ and $\text{ord}_p(f^q) = 2^e$; hence f^q is a generator of G. Finding f is rather easy: picking a random element $f \in \mathbb{Z}_p^*$ and testing $\left(\frac{f}{p}\right) = -1$ (using Alg 6.2). Since half the elements in $\mathbb{Z}_p^*$ are quadratic non-residues, the probability of finding a correct f in one go is one-half.

Sub-task (ii) is not too difficult either. The search of k from (6.6.4) is fast by utilizing the fact that non-unity quadratic-residue elements in G have orders as powers of 2. Thus, letting initially

Equation 6.6.5

$$b \stackrel{\text{def}}{=} a^q \equiv a^q g^{2^e} \pmod{p},$$

then $b \notin G$. We can search the least integer m for $0 \leq m < e$ such that

Equation 6.6.6

$$b^{2^m} \equiv 1 \pmod{p}$$

and then modify b into

Equation 6.6.7

$$b \leftarrow b g^{2^{e-m}} \equiv a^q g^{2^{e-m}} \pmod{p}.$$

Notice that b, after the modification in (6.6.7), has its order been reduced from that in (6.6.5) while remaining a quadratic residue in G and so the reduced order should remain being a power of 2. Therefore, the reduction must be in terms of a power of 2, and consequently, repeating (6.6.6) and (6.6.7), m in (6.6.6) will strictly decrease. Upon $m = 0$, (6.6.6) shows $b = 1$, and thereby (6.6.7) becomes (6.6.4) and so k can be found by accumulating 2^m in each loop of repetition. The search will terminate in at most e loops.

It is now straightforward to put our descriptions into Alg 6.4.

Since $e < \log_2 p$, the time complexity of Alg 6.4 is $O_B((\log p)^4)$.

. Remark 6.2

For the purpose of better exposition, we have presented Alg 6.4 by following our explanation on the working principle of Shanks' algorithm; in particular, we have followed precisely the explanation on Sub-task (ii) for searching the even exponent k. In so doing, our presentation of Shanks' algorithm sacrifices a little bit of efficiency: explicitly finding k, while is unnecessary since $g^{k/2}$ can be obtained as a byproduct in step 3, costs an additional modulo exponentiation in step 4. For the optimized version of Shanks' algorithm, see Algorithm 1.5.1 in [79]. □

Finally we should point out that Alg 6.4 contains Alg 6.3 as three special cases.

Algorithm 6.4: Square Root Modulo Prime

INPUT	prime p; integer $a \in QR_p$.
OUTPUT	a square root of a modulo p.

1. (*initialize*)

   set $p - 1 = 2^e q$ with q odd; $b \leftarrow a^q \pmod p$; $r \leftarrow e$; $k \leftarrow 0$;
2. (* sub-task (i), using Alg 6.2 *)

   find $f \in QNR_p$; $g \leftarrow f^q \pmod p$;
3. (* sub-task (ii), searching even exponent k *)

   while ($b \neq 1$) do

   3.1 find the least non-negative integer m such that $b^{2^m} \equiv 1 \pmod p$;

   3.2 $b \leftarrow bg^{2^{r-m}} \pmod p$; $k \leftarrow k + 2^{r-m}$; $r \leftarrow m$;
4. return($a^{(q+1)/2} g^{k/2} \pmod p$).

6.6.2 Computing Square Roots Modulo Composite

Thanks to Theorem 6.8, we know that, for $n = pq$ with p, q primes $\mathbb{Z}_n^*$ is isomorphic to $\mathbb{Z}_p^* \times \mathbb{Z}_q^*$. Since isomorphism preserves the arithmetic, relation

$$x^2 \equiv y \pmod n$$

holds if and only if it holds modulo both p and q. Therefore, if the factorization of n is given, square rooting modulo n can computed using Alg 6.5.

Clearly, the time complexity of Alg 6.5 is $O_B((\log n)^4)$.

By Corollary 6.2, $y \pmod p$ has two distinct square roots, which we denote by x_p and $p - x_p$, respectively. So does $y \pmod q$, which we denote by x_q and $q - x_q$, respectively. By the isomorphic relationship between $\mathbb{Z}_n^*$ and $\mathbb{Z}_p^* \times \mathbb{Z}_q^*$ (Theorem 6.8), we know that $y \in \mathrm{QR}_n$ has exactly four square roots in $\mathbb{Z}_n^*$. By Alg 6.5, these four roots are

Equation 6.6.8

$$\left.\begin{array}{rcl} x_1 \equiv & \overline{1}_p\, x_p & + \quad \overline{1}_q\, x_q \\ x_2 \equiv & \overline{1}_p\, x_p & + \quad \overline{1}_q\, (q - x_q) \\ x_3 \equiv & \overline{1}_p\, (p - x_p) & + \quad \overline{1}_q\, x_q \\ x_4 \equiv & \overline{1}_p\, (p - x_p) & + \quad \overline{1}_q\, (q - x_q) \end{array}\right\} \pmod n$$

Thus, if we apply (6.6.8) in Step 2 of Alg 6.5, we can compute all four square roots of the element input to the algorithm.

Algorithm 6.5: Square Root Modulo Composite

INPUT	primes p, q with $n = pq$; integer $y \in \mathrm{QR}_n$.
OUTPUT	a square root of y modulo n.

1. $x_p \leftarrow \sqrt{y \pmod p}$;
   $x_p \leftarrow \sqrt{y \pmod p}$; (* applying Algorithms 6.3 or 6.4 *)
2. return($\overline{1}_p x_p + \overline{1}_q x_q \pmod n$). (* applying Alg 6.1 *)

For an exercise, we ask: if $n = pqr$ with p, q, r distinct prime numbers, how many square roots for each $y \in \mathrm{QR}_n$?

We now know that if the factorization of n is known, then computing square roots of any given element in QR_n can be done efficiently. Now, what can we say about square rooting modulo n without knowing the factorization of n? The third part of the following theorem answers this question.

. Theorem 6.17

Let $n = pq$ with p, q being distinct odd primes and let $y \in QR_n$. Then the four square roots of y constructed in (6.6.8) have the following properties:

i. *they are distinct from one another*;

ii. $x_1 + x_4 = x_2 + x_3 = n$;

iii. $\gcd(x_1 + x_2, n) = \gcd(x_3 + x_4, n) = q$, $\gcd(x_1 + x_3, n) = \gcd(x_2 + x_4, n) = p$.

Proof

i. Noticing the meaning of $\overline{1}_p$ $_p$ and $\overline{1}_q$ $_q$ defined by (6.2.15) and (6.2.16), we have, e.g., $x_1 \pmod q = x_q$ and $x_2 \pmod q = q - x_q$. Remember, x_q and $q - x_q$ are two distinct square roots of $y \pmod q$. So $x_1 \not\equiv x_2 \pmod q$ implies $x_1 \not\equiv x_2 \pmod n$, i.e., x_1 and x_2 are distinct. Other cases can be shown analogously.

ii. From (6.6.8) we have

$$x_1 + x_4 = x_2 + x_3 = \overline{1}_p p + \overline{1}_q q.$$

The right-hand side value is congruent to 0 modulo p and modulo q. From these roots' membership in $\mathbb{Z}_n^*$ we have $0 < x_1 + x_4 = x_2 + x_3 < 2n$. Clearly, n is the only value in the interval $(0, 2n)$ and is congruent to 0 modulo p and q. So $x_1 = n - x_4$ and $x_2 = n - x_3$.

iii. We only study the case $x_1 + x_2$; other cases are analogous. Observing (6.6.8) we have

$$x_1 + x_2 = 2 \cdot \overline{1}_p x_p + \overline{1}_q q.$$

Therefore $x_1 + x_2 \pmod p \equiv 2x_p \not\equiv 0$ and $x_1 + x_2 \equiv 0 \pmod q$. Namely, $x_1 + x_2$ is a non-zero multiple of q, but not a multiple of p. This implies $\gcd(x_1 + x_2, n) = q$. □

Suppose there exists an efficient algorithm A, which, on input (y, n) for $y \in QR_n$, outputs x such that $x^2 \equiv y \pmod n$. Then we can run $A(x^2, n)$ to obtain a square root of x^2 which we denote by x'. By Theorem 6.17(iii), the probability for $1 < \gcd(x + x', n) < n$ is exactly one half (the probability space being the four square roots of y). That is, the algorithm A is an efficient algorithm for factoring n.

Combining Alg 6.5 and Theorem 6.5(iii), we have

. Corollary 6.3

Let $n = pq$ with p and q being distinct odd primes. Then factoring n is computationally equivalent to computing square root modulo n. □

Also from Theorem 6.17(ii) and the fact that n is odd, we have

. Corollary 6.4

Let $n = pq$ with p and q being distinct odd primes. Then for any $y \in QR_n$, two square roots of y are less than $n/2$, and the other two roots are larger than $n/2$. □

6.7 Blum Integers

Blum integers have wide applications in public-key cryptography.

Definition 6.4: Blum Integer*A composite integer n is called a Blum integer if n = pq where p and q are distinct prime numbers satisfying p ≡ q ≡* 3 (mod 4).

A Blum integer has many interesting properties. The following are some of them which are very useful in public-key cryptography and cryptographic protocols.

. Theorem 6.18

Let n be a Blum integer. Then the following properties hold for n:

i. $\left(\frac{-1}{p}\right) = \left(\frac{-1}{q}\right) = -1$ *(hence* $\left(\frac{-1}{n}\right) = 1$*)*;

ii. *For* $y \in \mathbb{Z}_n^*$*, if* $\left(\frac{y}{n}\right) = 1$ *then either* $y \in QR_n$ *or* $-y = n - y \in QR_n$;

iii. *Any* $y \in QR_n$ *has four square roots u, –u, v, –v and they satisfy (w.l.o.g.)*

a. $\left(\frac{u}{p}\right) = 1, \left(\frac{u}{q}\right) = 1,\ i.e.,\ u \in QR_n$;

b. $\left(\frac{-u}{p}\right) = -1, \left(\frac{-u}{q}\right) = -1$;

c. $\left(\frac{v}{p}\right) = -1, \left(\frac{v}{q}\right) = 1$;

d. $\left(\frac{-v}{p}\right) = 1, \left(\frac{-v}{q}\right) = -1$;

iv. *Function* $f(x) = x^2 \pmod n$ *is a permutation over* QR_n;

v. *For any* $y \in QR_n$*, exactly one square root of y with Jacobi symbol 1 is less than n/2*;

vi. $\mathbb{Z}_n^*$ *is partitioned into four equivalence classes: one multiplicative group* QR_n*, and three cosets* $(-1)QR_n, \xi QR_n, (-\xi)QR_n$*; here* x *is a square root of 1 with Jacobi symbol* –1.

Proof

i. Notice that $p \equiv 3 \pmod 4$ implies $\frac{p-1}{2} = 2k+1$. Then by Euler's Criterion (6.5.1), we have

$$\left(\frac{-1}{p}\right) = (-1)^{\frac{p-1}{2}} = (-1)^{2k+1} = -1.$$

Analogously, $\left(\frac{-1}{q}\right) = -1$.

ii. $\left(\frac{y}{n}\right) = 1$ implies either $\left(\frac{y}{p}\right) = \left(\frac{y}{q}\right) = 1$ or $\left(\frac{y}{p}\right) = \left(\frac{y}{q}\right) = -1$. For the first case, $y \in QR_n$ due to the definition of Legendre symbol (Definition 6.3) and Theorem 6.14.

For the second case, (i) implies $\left(\frac{-y}{p}\right) = \left(\frac{-y}{q}\right) = 1$. Hence $-y \in QR_n$.

iii. First of all, by Theorem 6.17(ii), we can indeed denote the four distinct square roots of x by u, $-u(=n-u)$, v and $-v$.

Next, from $u^2 \equiv v^2 \pmod n$, we have $(u+v)(u-v) \equiv 0 \pmod p$, that is, $u \equiv \pm v \pmod p$. Similarly, $u \equiv \pm v \pmod q$. However, by Theorem 6.17(i), $u \not\equiv \pm v \pmod n$, so only the following two cases are possible:

$$u \equiv v \pmod p \text{ and } u \equiv -v \pmod q,$$

or

$$u \equiv -v \pmod p \text{ and } u \equiv v \pmod q.$$

These two cases plus (i) imply $\left(\frac{u}{n}\right) = -\left(\frac{v}{n}\right)$.

Thus, if $\left(\frac{u}{n}\right) = 1$ then $\left(\frac{v}{n}\right) = -1$ and if $\left(\frac{u}{n}\right) = -1$ then $\left(\frac{v}{n}\right) = 1$. Without loss of generality, the four distinct Legendre-symbol characterizations in (a)-(d) follow the multiplicative property of Legendre-Jacobi symbol and (i).

iv. For any $y \in QR_n$, by (iii) there exists a unique $x \in QR_n$ satisfying $f(x) = y$. Thus, $f(x)$ is a 1-1 and onto mapping, i.e., a permutation, over QR_n.

v. By (iii), the square root with Jacobi symbol 1 is either u or $n-u$. Only one of them

can be less than $n/2$ since n is odd. (So, exactly one square root with Jacobi symbol -1 is less than $n/2$; the other two roots are larger than $n/2$ and have the opposite Jacobi symbols.)

vi. It is trivial to check that QR_n forms a group under multiplication modulo n with 1 as the identity. Now by (iii), the four distinct square roots of 1 have the four distinct Legendre-symbol characterizations in (a), (b), (c), and (d), respectively. Therefore the four sets QR_n, $(-1)QR_n$,ξQR_n, $(-\xi)QR_n$ are pair wise disjoint. These four sets make up $\mathbb{Z}_n^*$ because by Theorem 6.15, $\#\mathrm{QR}_n = \frac{\#\mathbb{Z}_n^*}{4}$.

6.8 Chapter Summary

In this chapter we have conducted a study in the following topics of elementary number theory:

- Linear congruences
- Chinese Remainder Theorem (with algorithm)
- Lagrange's, Euler's and Fermat's theorems
- Quadratic residues and Legendre-Jacobi symbols (with algorithm)
- Square roots modulo integers and the relation to factorization (with algorithm for root extraction)
- Blum integers and their properties

In addition to introducing the basic knowledge and facts, we have also studied several important algorithms (Chinese Remainder, Jacobi symbol, square-rooting), with their working principles explained and their time complexity behaviors analyzed. In so doing, we considered that these algorithms not only have theoretic importance, but also have practical importance: these algorithms are frequently used in cryptography and cryptographic protocols.

In the rest of this book we will frequently apply the knowledge, facts, skills and algorithms which we have learned in this chapter.

Exercises

6.1 Let m, n be positive integers satisfying $m|n$. Show that operation "(mod m)" partitions $\mathbb{Z}_n$ into n/m equivalence classes, each has m elements.

6.2 Under the same condition of the preceding problem, show $\mathbb{Z}_n/mZ_n = \mathbb{Z}_m$.

6.3 Use the Chinese Remainder Algorithm (Alg 6.1) to construct an element in $\mathbb{Z}_{35}$ which maps to (2, 3) $\in \mathbb{Z}_5 \times \mathbb{Z}_7$ under the isomorphism in Theorem 6.1. Prove that this element has the maximum order.

6.4 Use the method in Example 6.2 to find the other three square roots of 29 in $\mathbb{Z}_{35}^*$. Find analogously the four square roots of 1 in $\mathbb{Z}_{35}^*$.

Hint: 29 (mod 5) = 4 which has square roots 2 and 3 (= –2 (mod 5)), and 29 (mod 7) = 1 which has square roots 1 and 6 (= –1 (mod 7)); the four square roots of 29 modulo 35 are isomorphic to (2, 1), (2, 6), (3, 1) and (3, 6) in $\mathbb{Z}_5 \times \mathbb{Z}_7$.

6.5 Construct an odd composite number n such that n is square free, i.e., there exists no prime p such that $p^2|N$, however $\gcd(n,\phi(n)) > 1$.

6.6 Let $m|n$. Prove that for any $x \in \mathbb{Z}_n^*$, $\text{ord}_m(x)|\text{ord}_n(x)$.

6.7 Let $n = pq$ with p, q being distinct primes. Since $p - 1|\phi(n)$, there exists elements in $\mathbb{Z}_n^*$ of order dividing $p - 1$. (Similarly, there are elements of order dividing $q - 1$.) Prove that for any $g \in \mathbb{Z}_n^*$, if $\text{ord}_n(g)|p - 1$ and $\text{ord}_n(g) \nmid q - 1$, then $\gcd(g - 1,n) = q$. (Similarly, any $h \in \mathbb{Z}_n^*$ of $\text{ord}_n(h)|q - 1$ and $\text{ord}_n(h) \mid p - 1$, $\gcd(h - 1, n) = p$.)

6.8 Let $n = pq$ with p, q being distinct primes. Show that for any $g \in \mathbb{Z}_n^*$, it holds $g^{p+q} \equiv g^{n+1} \pmod n$. For $|p| \approx |q|$, show that an upper bound for factoring n is $n^{1/4}$.

Hint: find $p + q$ from $g^{n+1} \pmod n$ using Pollard's λ-algorithm; then factor n using $p + q$ and pq.

6.9 Let p be a prime. Show that a generator of the group $\mathbb{Z}_p^*$ must be a quadratic non-residue. Analogously, let n be an odd composite; show that elements in $\mathbb{Z}_n^*$ of the maximum order must be quadratic non-residues.

6.10 Testing quadratic residuosity modulo p using Euler's criterion is $\log p$ times slower than doing so via evaluation of Legendre symbol. Why?

6.11 Factor 35 using the square roots computed in Exercise 6.4

6.12 Show that QR_n is a subgroup of $J_n(1)$ and the latter is a subgroup of $\mathbb{Z}_n^*$.

6.13 Let $n = pq$ with p and q being distinct primes. Under what condition $-1 \in QR_n$?

Under what condition $\left(\frac{-1}{n}\right) = -1$?

6.14 Let n be a Blum integer. Construct the inversion of the function $f(x) = x^2 \pmod{n}$ over QR_n.

Hint: apply the Chinese Remainder Theorem (Alg 6.1) to Case 1 of Alg 6.3.

6.15 Let $n = pq$ be a Blum integer satisfying $\gcd(p - 1, q - 1) = 2$. Show that group $J_n(1)$ is cyclic.

Hint: apply Chinese Remainder Theorem to construct an element using a generator of $\mathbb{Z}_p^*$ and one of $\mathbb{Z}_q^*$. Prove that this element is in $J_n(1)$ and is of order $\#J_n(1)$.