Lecture Notes for CS-578 Cryptography (SS2006) Prof. Michael Backes

6. Basic Number Theory Facts

Lecture 9 Saarland University

In this chapter we introduce some basic facts about $\mathbb{Z}_p$ and $\mathbb{Z}_p^*$ for primes p. In particular, we study the existence and computational complexity of finding inverses, of exponentiation, of solving linear and quadratic equations and of extracting square roots. Later in the course we will move from groups $\mathbb{Z}_p$ for primes p to groups $\mathbb{Z}_n$ where n is a composite number (usually consisting of the product of two primes). We do not give any proofs here since they can be found in essentially every textbook on the topic; moreover, none of these proofs will matter in the subsequent lectures.

6.1 Basic Number Theory Facts – Arithmetic Modulo Primes

In the following, we are dealing with large primes p, e.g., primes that are 1024 bits long.

6.1.1 $\mathbb{Z}_p$ and Some Basic Facts

For a prime p let $\mathbb{Z}_p := \{0, 1, 2, \dots, p-1\}$. Elements of $\mathbb{Z}_p$ can be added modulo p and multiplied modulo p as usual. For adding two elements $a, b \in \mathbb{Z}_p$, we often write $a + b \bmod p$ instead of $a + b$ to make clear in which group the addition is performed, similarly for multiplication.

It turns out that all elements $g \in \mathbb{Z}_p$ except for $g = 0$ are *invertible (with respect to multiplication)*, i.e., for every such g there exists some element h such that $g \cdot h = 1 \bmod p$. We write g^{-1} to denote the inverse of g. We denote by $\mathbb{Z}_p^* = \{1, 2, \dots, p-1\}$ the set of invertible elements in $\mathbb{Z}_p$.

Let us now mention *Fermat's little theorem.*

Theorem 6.1 (Fermat) *For any $g \in \mathbb{Z}_p^*$, we have $g^{p-1} = 1 \bmod p$.* □

Here and in the following, we complement definitions and algorithms with simple examples to foster basic understanding.

Example 6.1 *As an example for illustrating Fermat's little theorem, we have* $3^4 \bmod 5 = 81 \bmod 5 = 1 \bmod 5$.

Fermat's little theorem reasons about exponentiation in $\mathbb{Z}_p^*$. Getting to the computational side, it is easy to see that addition and multiplication modulo p can be done in polynomial time (with respect to the bitlength of p). In contrast to $\mathbb{Z}$ however, it turns out that even exponentiation in $\mathbb{Z}_p^*$ can be achieved in polynomial time in the bitlength of p, using the so-called *repeated squaring technique.*

Definition 6.1 (Repeated Squaring Technique) *Let $g \in \mathbb{Z}_p^*$ and $a \in \{0, \dots, p-2\}$. Then $g^a \bmod p$ can be computed as follows. Let the bit representation of a be given by $a = \sum_{i=0}^{k-1} a_i 2^i$. Then we have*

$$
\begin{aligned}
g^{\sum_{i=0}^{k-1} a_i 2^i} &= (g)^{a_0} \cdot (g^2)^{\sum_{i=1}^{k-1} a_i 2^i} \\
&= (g)^{a_0} \cdot (g^2)^{a_1} \cdot (g^4)^{\sum_{i=2}^{k-1} a_i 2^i} = \dots \\
&= (g)^{a_0} \cdot (g^2)^{a_1} \cdot (g^4)^{a_2} \cdots (g^{2^{k-1}})^{a_{k-1}}
\end{aligned}
$$

It is easy to see that the computing time is bounded by $O(k^3)$ since it takes $O(k^2)$ time to square in $\mathbb{Z}_p$, and we need to do k repeated squarings to successively compute $(g^{2^i} \bmod p)$ for all $1 \leq i \leq k$. $\diamond$

Example 6.2 *Let $g = 3$, $a = 5$, and $p = 7$. Thus $a_0 = a_2 = 1$ and $a_1 = 0$. We first compute $g = 3$, $g^2 = 3^2 = 2 \bmod 7$, $g^4 = 2^2 = 4 \bmod 7$. Thus we get $g^a = g^{a_0} \cdot (g^2)^{a_1} \cdot (g^4)^{a_2} = g \cdot g^4 = 3 \cdot 4 = 5 \bmod 7$.*

Fermat's little theorem furthermore entails a very simple procedure for computing the inverse of an element $g \in \mathbb{Z}_p^*$: one simply sets $g^{-1} := g^{p-2} \bmod p$. One easily verifies that $g \cdot g^{p-2} = g^{p-1} = 1 \bmod p$, according to Fermat's little theorem.

Example 6.3 *The inverse of 3 modulo 5 is given by $3^{5-2} = 27 = 2 \bmod 5$, and indeed $2 \cdot 3 = 1 \bmod 5$.*

Note further that we have $2^{-1} \bmod p = \frac{p+1}{2}$ for all p, since $2 \cdot \frac{p+1}{2} = p + 1 = 1 \bmod p$.

Moreover, joining the repeated squaring technique with this way of computing inverses entails an efficient algorithm for solving linear equations of the form $a \cdot x = b \mod p$: one simply computes $x = b \cdot a^{-1} = b \cdot a^{p-2} \bmod p$. One again easily verifies that $a \cdot (b \cdot a^{p-2}) = b \cdot a^{p-1} = b \cdot 1 = b \bmod p$, where we again exploited Fermat's little theorem.

What however remains unclear at this point is how to solve quadratic equations or equations of even higher order. We will discuss this in the next subsection in detail for quadratic equations while referring to the literature for equations of higher order.

6.1.2 $\mathbb{Z}_p^*$ and Some Basic Facts

We start with the following important theorem.

Theorem 6.2 *For every prime p, $\mathbb{Z}_p^*$ constitutes a* cyclic group, *i.e., there exists some $g \in \mathbb{Z}_p^*$ such that $\mathbb{Z}_p^* = \{1 = g^0, g, g^2, g^3, \dots, g^{p-2}\}$. An element g with this property is called a* generator *of $\mathbb{Z}_p^*$.* □

In other words, if g is a generator of $\mathbb{Z}_p^*$, then for every $h \in \mathbb{Z}_p^*$ there exists some $i \in \{0, \dots, p-2\}$ such that $h = g^i \bmod p$.

Example 6.4 *The element 3 constitutes a generator of $\mathbb{Z}_7^*$ since $\{1, 3, 3^2, 3^3, 3^4, 3^5, 3^6\} = \{1, 3, 2, 6, 4, 5\}(\text{mod} 7) = \mathbb{Z}_7^*$.*

However, not every element of $\mathbb{Z}_p^*$ is a generator. The number 2, for example, is not a generator of $\mathbb{Z}_7^*$, since we we have $\{1, 2, 2^2, 2^3, 2^4, 2^5, 2^6\} = \{1, 2, 4\}(\text{mod} 7) \neq \mathbb{Z}_7^*$.

Although not every element is a generator, every element generates at least a *subgroup* $\langle g \rangle$ of $\mathbb{Z}_p^*$, defined as $\langle g \rangle := \{h \in \mathbb{Z}_p^* \mid \exists i \in \{0, \dots, p-2\} \colon h = g^i\}$. For generators g of $\mathbb{Z}_p^*$, we thus have $\langle g \rangle = \mathbb{Z}_p^*$.

The *order* of $g \in \mathbb{Z}_p^*$ is defined to be the size of the group it generates. Equivalently, it is the smallest positive integer i such that $g^i = 1 \bmod p$. We denote the order of $g \in \mathbb{Z}_p^*$ by $\mathsf{ord}_p(g)$. If we use the aforementioned examples, we obtain $\mathsf{ord}_7(3) = 6$ and $\mathsf{ord}_7(2) = 3$. If the factorization of $p-1$ is known then there is a simple and efficient algorithm to determine $\mathsf{ord}_p(g)$ for any $g \in \mathbb{Z}_p^*$ (We will see this later).

Let us conclude this subsection with the famous theorem of Lagrange.

Theorem 6.3 (Lagrange) *For all $g \in \mathbb{Z}_p^*$ we have that $\mathsf{ord}_p(g)$ divides $p-1$.* □

6.1.3 Quadratic Residues and Quadratic Non-Residues

The *square root* of an element $g \in \mathbb{Z}_p^*$ is an element $h \in \mathbb{Z}_p^*$ such that $h^2 = g \bmod p$. We as usual write $\sqrt{g}$ instead of h. For instance, we have that $\sqrt{2} = 3 \bmod 7$ since $3^2 = 2 \bmod 7$. Square root however do not necessarily exist, e.g., 3 does not have a square root modulo 7.

An element $g \in \mathbb{Z}_p^*$ is called a *Quadratic Residue* (or *QR* for short) if it has a square root in $\mathbb{Z}_p^*$. Otherwise it is called a *Quadratic Non-Residue* (or *QNR* for short).

How many square roots does an element $g \in \mathbb{Z}_p^*$ have? Consider the equation $x^2 = y^2 \bmod p$. This is equivalent to $0 = x^2 - y^2 = (x-y)(x+y) \bmod p$. Since $\mathbb{Z}_p$ is an "integral domain" we know that $x = y \bmod p$ or $x = -y \bmod p$. Hence, elements in $\mathbb{Z}_p^*$ have either zero square roots or two square roots. If h is the square root of g modulo p then $-h$ is also a square root of g modulo p.

The following theorem of Euler tells us how to decide if an element is a quadratic residue or not.

Theorem 6.4 (Euler) *An element $g \in \mathbb{Z}_p^*$ is a QR if and only if $g^{(p-1)/2} = 1 \bmod p$.* □

Example 6.5 *We have that 2 is a QR in $\mathbb{Z}_7^*$ since $2^{(7-1)/2} = 1 \bmod 7$; on the other hand 3 is a QNR in $\mathbb{Z}_7^*$ since $3^{(7-1)/2} = -1 \bmod 7$.*

A special case we are often interested in are the roots of the multiplicative unit: For every $g \in \mathbb{Z}_p^*$, we have that $h = g^{(p-1)/2}$ is a square root of 1 since $h^2 = g^{p-1} = 1 \bmod p$ according to Fermat's little theorem. One the other hand, the element 1 can have at most two square roots as we have seen, and 1 and -1 are of course square roots of 1 modulo every prime p. Consequently, we know that $g^{(p-1)/2} \bmod p \in \{1, -1\}$ for every $g \in \mathbb{Z}_p^*$.

Next we define the Legendre symbol.

Definition 6.2 (Legendre Symbol) *For $g \in \mathbb{Z}_p$, we define*

$$\left(\frac{g}{p}\right) = \begin{cases} 1 & \textit{if } g \textit{ is a QR in } \mathbb{Z}_p \\ -1 & \textit{if } g \textit{ is not a QR in } \mathbb{Z}_p \\ 0 & \textit{if } g = 0 \bmod p \end{cases}$$

◇

Euler's theorem implies that $\left(\frac{g}{p}\right) = g^{(p-1)/2} \bmod p$. Thus the Legendre symbol can be efficiently computed, e.g., using the repeated squaring technique.

We conclude with some easy additional facts:

1. Let g be a generator of $\mathbb{Z}_p^*$ and let $h = g^r$ for some integer r. Then h is a QR in $\mathbb{Z}_p^*$ if and only if r is even. This implies that the Legendre symbol reveals the parity of r. More precisely, $\left(\frac{g^r}{p}\right) = \left(\frac{g}{p}\right)^r$.

2. Since $x = g^r$ is a QR if and only if r is even, it follows that exactly half the elements of $\mathbb{Z}_p$ are QR's.

3. When $p = 3 \bmod 4$ computing square roots of a QR $g \in \mathbb{Z}_p^*$ is easy. One simply computes $h_1 = g^{p+1}/4 \bmod p$ and $h_2 = -g^{p+1}/4 \bmod p$. This gives the correct result since $h_1^2 = h_2^2 = g^{p+1/2} = g \cdot g^{p-1/2} = g \cdot \left(\frac{g}{p}\right) = g \cdot 1 = g \bmod p$.

4. When $p = 1 \bmod 4$ computing square roots of a QR $g \in \mathbb{Z}_p^*$ is possible but somewhat more complicated (one needs a randomized algorithm).

5. There exists a simple algorithm for solving quadratic equations in $\mathbb{Z}_p$: We know that if a solution to $ax^2 + bx + c = 0 \bmod p$ exists then it is given by

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} (\mathrm{mod} p).$$

   Hence, the equation has a solution in $\mathbb{Z}_p$ if and only if $A := b^2 - 4ac$ is a QR in $\mathbb{Z}_p^*$ (or if $A = 0$). Using our algorithm for taking square roots in $\mathbb{Z}_p^*$ we can find $\sqrt{A} \bmod p$ and recover x_1 and x_2.

6. What about higher-order equations in $\mathbb{Z}_p$? We only need to know that there exists an efficient randomized algorithm that solves any equation of degree d in time polynomial in d and the length of p.

6.1.4 Efficient Computing in $\mathbb{Z}_p$ — Upper Time Bounds

We briefly summarize here in which time the respective operations can be computed.

1. Adding two elements $x, y \in \mathbb{Z}_p$ can be done in linear time in the length of p.

2. Multiplying two elements $x, y \in \mathbb{Z}_p$ can be done in quadratic time in the length of p. If p is n bits long, more clever (and practical) algorithms work in time $O(n^{1.7})$ (rather than $O(n^2)$).

3. Inverting an element $x \in \mathbb{Z}_p^*$ can be done in quadratic time in the length of p.

4. Using the repeated squaring technique, $x^r \bmod p$ can be computed in time $(\log_2 r)O(n^2)$ where p is n bits long. Note that the algorithm takes linear time in the length of r.

6.1.5 Summary of Easy and Hard Problems

Let p be a 1024 bit prime. The following problems constitute easy problems in $\mathbb{Z}_p$:

1. Adding and multiplying elements.

2. Generating a random element in $\mathbb{Z}_p$ or a random generator of $\mathbb{Z}_p^*$.

3. Computing $g^r \bmod p$ even if r is very large.

4. Inverting an element and solving linear systems.

5. Testing if an element is a QR and computing its square root if it is a QR.

6. Solving polynomial equations of degree d can be done in polynomial time in d.

The following problems are believed to be hard (intractable) in $\mathbb{Z}_p$:

1. Let g be a generator of $\mathbb{Z}_p^*$. Given $x \in \mathbb{Z}_p^*$, find r such that $x = g^r \bmod p$. This is known as the *discrete logarithm problem.*
2. Let g be a generator of $\mathbb{Z}_p^*$. Given $i, j \in \mathbb{Z}_p^*$ where $i = g^a$ and $j = g^b$. Find $h = g^{ab}$. This is known as the *Diffie-Hellman problem.*
3. Finding roots of sparse polynomials of high degree. For example finding a root of: $x^{2^{1000}} + 23 \cdot x^{2^{167}} + 9 \cdot x^{2^{62}} + x^3 + 7 = 0 \bmod p$.