Public Key Cryptography

Chapter 3/Part2

PRESENTED BY
DR. ALI J. ABBOUD
3.1 Objectives

• Prime Numbers.
• Fermat’s and Euler’s Theorems.
• Testing for Primarily.
• Discrete Logarithm
• Diffie-Hellman Key Exchange Algorithm.
• Security of Diffie-Hellman Algorithm.
• Key Exchange Protocols.
• Man-in-the-Middle Attacks.
3.2 Prime Numbers

Any integer \( a > 1 \) can be factored in a unique way as

\[
a = p_1^{a_1} \times p_2^{a_2} \times \cdots \times p_l^{a_l}
\]

where \( p_1 < p_2 < \cdots < p_l \) are prime numbers and where each \( a_i \) is a positive integer. This is known as the fundamental theorem of arithmetic; a proof can be found in any text on number theory.

<table>
<thead>
<tr>
<th>91</th>
<th>7 × 13</th>
</tr>
</thead>
<tbody>
<tr>
<td>3600</td>
<td>(2^4 \times 3^2 \times 5^2)</td>
</tr>
<tr>
<td>11011</td>
<td>7 × 11^2 × 13</td>
</tr>
</tbody>
</table>

It is useful for what follows to express this another way. If \( P \) is the set of all prime numbers, then any positive integer \( a \) can be written uniquely in the following form:

\[
a = \prod_{p \in P} p^{a_p} \quad \text{where each } a_p \geq 0
\]

The right-hand side is the product over all possible prime numbers \( p \); for any particular value of \( a \), most of the exponents \( a_p \) will be 0.

The value of any given positive integer can be specified by simply listing all the nonzero exponents in the foregoing formulation.

The integer 12 is represented by \( \{a_2 = 2, a_3 = 1\} \).
The integer 18 is represented by \( \{a_2 = 1, a_3 = 2\} \).
The integer 91 is represented by \( \{a_7 = 1, a_{13} = 1\} \).
3.2 Prime Numbers

Multiplication of two numbers is equivalent to adding the corresponding exponents. Given $a = \prod_{p \in P} p^{a_p}$, $b = \prod_{p \in P} p^{b_p}$. Define $k = ab$. We know that the integer $k$ can be expressed as the product of powers of primes: $k = \prod_{p \in P} p^{k_p}$. It follows that $k_p = a_p + b_p$ for all $p \in P$.

$k = 12 \times 18 = (2^2 \times 3) \times (2 \times 3^2) = 216$
$k_2 = 2 + 1 = 3; k_3 = 1 + 2 = 3$
$216 = 2^3 \times 3^3 = 8 \times 27$

What does it mean, in terms of the prime factors of $a$ and $b$, to say that $a$ divides $b$? Any integer of the form $p^n$ can be divided only by an integer that is of a lesser or equal power of the same prime number, $p^j$ with $j \leq n$. Thus, we can say the following.
3.2 Prime Numbers

Given

\[ a = \prod_{p \in P} p^{a_p}, \quad b = \prod_{p \in P} p^{b_p} \]

If \(a|b\), then \(a_p \leq b_p\) for all \(p\).

\(a = 12; \quad b = 36; \quad 12|36\)
\(12 = 2^2 \times 3; \quad 36 = 2^2 \times 3^2\)
\(a_2 = 2 = b_2\)
\(a_3 = 1 \leq 2 = b_3\)

Thus, the inequality \(a_p \leq b_p\) is satisfied for all prime numbers.

It is easy to determine the greatest common divisor of two positive integers if we express each integer as the product of primes.

\[ 300 = 2^2 \times 3^1 \times 5^2 \]
\[ 18 = 2^1 \times 3^2 \]
\[ \gcd(18, 300) = 2^1 \times 3^1 \times 5^0 = 6 \]

The following relationship always holds:

\[
\text{If } k = \gcd(a, b), \text{ then } k_p = \min(a_p, b_p) \text{ for all } p.
\]

Determining the prime factors of a large number is no easy task, so the preceding relationship does not directly lead to a practical method of calculating the greatest common divisor.
3.3 FERMAT’S AND EULER’S THEOREMS

Two theorems that play important roles in public-key cryptography are Fermat’s theorem and Euler’s theorem.

**Fermat’s Theorem**

Fermat’s theorem states the following: If \( p \) is prime and \( a \) is a positive integer not divisible by \( p \), then

\[
a^{p-1} = 1 \pmod{p}
\]

An alternative form of Fermat’s theorem is also useful: If \( p \) is prime and \( a \) is a positive integer, then

\[
a^p = a \pmod{p}
\]

<table>
<thead>
<tr>
<th>( p = 5 ), ( a = 3 )</th>
<th>( a^p = 3^5 = 243 = 3 \pmod{5} = a \pmod{p} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p = 5 ), ( a = 10 )</td>
<td>( a^p = 10^5 = 100000 = 10 \pmod{5} = 0 \pmod{5} = a \pmod{p} )</td>
</tr>
</tbody>
</table>
3.3 FERMAT’S AND EULER’S THEOREMS

Euler’s Totient Function

Before presenting Euler’s theorem, we need to introduce an important quantity in number theory, referred to as Euler’s totient function, written $\phi(n)$, and defined as the number of positive integers less than $n$ and relatively prime to $n$. By convention, $\phi(1) = 1$.

Determine $\phi(37)$ and $\phi(35)$.

Because 37 is prime, all of the positive integers from 1 through 36 are relatively prime to 37. Thus $\phi(37) = 36$.

To determine $\phi(35)$, we list all of the positive integers less than 35 that are relatively prime to it:

$$1, 2, 3, 4, 6, 8, 9, 11, 12, 13, 16, 17, 18$$
$$19, 22, 23, 24, 26, 27, 29, 31, 32, 33, 34$$

There are 24 numbers on the list, so $\phi(35) = 24$. 
3.3 FERMAT’S AND EULER’S THEOREMS

Table lists the first 30 values of $\phi(n)$. The value $\phi(1)$ is without meaning but is defined to have the value 1.

It should be clear that, for a prime number $p$,

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

Now suppose that we have two prime numbers $p$ and $q$ with $p \neq q$. Then we can show that, for $n = pq$,

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

To see that $\phi(n) = \phi(p) \times \phi(q)$, consider that the set of positive integers less than $n$ is the set $\{1, \ldots, (pq - 1)\}$. The integers in this set that are not relatively prime to $n$ are the set $\{p, 2p, \ldots, (q - 1)p\}$ and the set $\{q, 2q, \ldots, (p - 1)q\}$. Accordingly,

$$\phi(n) = (pq - 1) - [(q - 1) + (p - 1)]$$
$$= pq - (p + q) + 1$$
$$= (p - 1) \times (q - 1)$$
$$= \phi(p) \times \phi(q)$$
### 3.3 FERMAT’S AND EULER’S THEOREMS

#### Table: Some Values of Euler’s Totient Function $\phi(n)$

<table>
<thead>
<tr>
<th>$n$</th>
<th>$\phi(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>13</td>
<td>12</td>
</tr>
<tr>
<td>14</td>
<td>6</td>
</tr>
<tr>
<td>15</td>
<td>8</td>
</tr>
<tr>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>17</td>
<td>16</td>
</tr>
<tr>
<td>18</td>
<td>6</td>
</tr>
<tr>
<td>19</td>
<td>18</td>
</tr>
<tr>
<td>20</td>
<td>8</td>
</tr>
<tr>
<td>21</td>
<td>12</td>
</tr>
<tr>
<td>22</td>
<td>10</td>
</tr>
<tr>
<td>23</td>
<td>22</td>
</tr>
<tr>
<td>24</td>
<td>8</td>
</tr>
<tr>
<td>25</td>
<td>20</td>
</tr>
<tr>
<td>26</td>
<td>12</td>
</tr>
<tr>
<td>27</td>
<td>18</td>
</tr>
<tr>
<td>28</td>
<td>12</td>
</tr>
<tr>
<td>29</td>
<td>28</td>
</tr>
<tr>
<td>30</td>
<td>8</td>
</tr>
</tbody>
</table>

$\phi(21) = \phi(3) \times \phi(7) = (3 - 1) \times (7 - 1) = 2 \times 6 = 12$

where the 12 integers are $\{1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20\}$. 

3.3 FERMAT’S AND EULER’S THEOREMS

Euler’s Theorem

Euler’s theorem states that for every $a$ and $n$ that are relatively prime:

$$a^\phi(n) \equiv 1 \pmod{n}$$

As is the case for Fermat’s theorem, an alternative form of the theorem is also useful:

$$a^{\phi(n)+1} \equiv a \pmod{n}$$

TESTING FOR PRIMALITY

For many cryptographic algorithms, it is necessary to select one or more very large prime numbers at random. Thus, we are faced with the task of determining whether a given large number is prime. There is no simple yet efficient means of accomplishing this task.

In this section, we present one attractive and popular algorithm. You may be surprised to learn that this algorithm yields a number that is not necessarily a prime. However, the algorithm can yield a number that is almost certainly a prime. This will be explained presently. We also make reference to a deterministic algorithm for finding primes. The section closes with a discussion concerning the distribution of primes.
3.4 Testing For Primality

Miller-Rabin Algorithm

DETAILS OF THE ALGORITHM These considerations lead to the conclusion that, if \( n \) is prime, then either the first element in the list of residues, or remainders, \((a^q,a^{2q},...,a^{2^{k-1}q},a^{2^kq})\) modulo \( n \) equals 1; or some element in the list equals \((n - 1)\); otherwise \( n \) is composite (i.e., not a prime). On the other hand, if the condition is met, that does not necessarily mean that \( n \) is prime. For example, if \( n = 2047 = 23 \times 89 \), then \( n - 1 = 2 \times 1023 \). We compute \( 2^{1023} \mod 2047 = 1 \), so that 2047 meets the condition but is not prime.

We can use the preceding property to devise a test for primality. The procedure TEST takes a candidate integer \( n \) as input and returns the result composite if \( n \) is definitely not a prime, and the result inconclusive if \( n \) may or may not be a prime.

TEST \((n)\)
1. Find integers \( k, q \), with \( k > 0 \), \( q \) odd, so that \((n - 1 = 2^kq)\);
2. Select a random integer \( a, 1 < a < n - 1 \);
3. if \( a^q \mod n = 1 \) then return("inconclusive");
4. for \( j = 0 \) to \( k - 1 \) do
5. if \( a^{2^jq} \mod n = n - 1 \) then return("inconclusive");
6. return("composite");
3.4 Testing For Primality

Let us apply the test to the prime number $n = 29$. We have $(n - 1) = 28 = 2^2(7) = 2^k q$. First, let us try $a = 10$. We compute $10^7 \mod 29 = 17$, which is neither 1 nor 28, so we continue the test. The next calculation finds that $(10^7)^2 \mod 29 = 28$, and the test returns **inconclusive** (i.e., 29 may be prime). Let’s try again with $a = 2$. We have the following calculations: $2^7 \mod 29 = 12; 2^{14} \mod 29 = 28$; and the test again returns **inconclusive**. If we perform the test for all integers $a$ in the range 1 through 28, we get the same **inconclusive** result, which is compatible with $n$ being a prime number.

Now let us apply the test to the composite number $n = 13 \times 17 = 221$. Then $(n - 1) = 220 = 2^2(55) = 2^k q$. Let us try $a = 5$. Then we have $5^{55} \mod 221 = 112$, which is neither 1 nor $220(5^{55})^2 \mod 221 = 168$. Because we have used all values of $j$ (i.e., $j = 0$ and $j = 1$) in line 4 of the TEST algorithm, the test returns **composite**, indicating that 221 is definitely a composite number. But suppose we had selected $a = 21$. Then we have $21^{55} \mod 221 = 200; (21^{55})^2 \mod 221 = 220$; and the test returns **inconclusive**, indicating that 221 may be prime. In fact, of the 218 integers from 2 through 219, four of these will return an inconclusive result, namely 21, 47, 174, and 200.
3.5 Discrete Logarithm

\[ a^{\phi(n)} \equiv 1 \pmod{n} \]

where \( \phi(n) \), Euler’s totient function, is the number of positive integers less than \( n \) and relatively prime to \( n \). Now consider the more general expression:

\[ a^m \equiv 1 \pmod{n} \]

If \( a \) and \( n \) are relatively prime, then there is at least one integer \( m \) that satisfies Equation above, namely, \( M = \phi(n) \). The least positive exponent \( m \) for which Equation above holds is referred to in several ways:

- The order of \( a \) (mod \( n \))
- The exponent to which \( a \) belongs (mod \( n \))
- The length of the period generated by \( a \)
3.5 Discrete Logarithm

To see this last point, consider the powers of 7, modulo 19:

\[
\begin{align*}
7^1 & \equiv 7 \pmod{19} \\
7^2 &= 49 = 2 \times 19 + 11 \equiv 11 \pmod{19} \\
7^3 &= 343 = 18 \times 19 + 1 \equiv 1 \pmod{19} \\
7^4 &= 2401 = 126 \times 19 + 7 \equiv 7 \pmod{19} \\
7^5 &= 16807 = 884 \times 19 + 11 \equiv 11 \pmod{19}
\end{align*}
\]

There is no point in continuing because the sequence is repeating. This can be proven by noting that \(7^3 \equiv 1 \pmod{19}\), and therefore, \(7^{3+j} = 7^3 \cdot 7^j \equiv 7^j \pmod{19}\), and hence, any two powers of 7 whose exponents differ by 3 (or a multiple of 3) are congruent to each other \(\pmod{19}\). In other words, the sequence is periodic, and the length of the period is the smallest positive exponent \(m\) such that \(7^m \equiv 1 \pmod{19}\).

Table 8.3 shows all the powers of \(a\), modulo 19 for all positive \(a < 19\). The length of the sequence for each base value is indicated by shading. Note the following:

1. All sequences end in 1. This is consistent with the reasoning of the preceding few paragraphs.
2. The length of a sequence divides \(\phi(19) = 18\). That is, an integral number of sequences occur in each row of the table.
3. Some of the sequences are of length 18. In this case, it is said that the base integer \(a\) generates (via powers) the set of nonzero integers modulo 19. Each such integer is called a primitive root of the modulus 19.
### 3.5 Discrete Logarithm

**Table 8.3 Powers of Integers, Modulo 19**

<table>
<thead>
<tr>
<th>a</th>
<th>a^2</th>
<th>a^3</th>
<th>a^4</th>
<th>a^5</th>
<th>a^6</th>
<th>a^7</th>
<th>a^8</th>
<th>a^9</th>
<th>a^10</th>
<th>a^11</th>
<th>a^12</th>
<th>a^13</th>
<th>a^14</th>
<th>a^15</th>
<th>a^16</th>
<th>a^17</th>
<th>a^18</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>13</td>
<td>7</td>
<td>14</td>
<td>9</td>
<td>18</td>
<td>17</td>
<td>15</td>
<td>11</td>
<td>3</td>
<td>6</td>
<td>12</td>
<td>5</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>8</td>
<td>5</td>
<td>15</td>
<td>7</td>
<td>2</td>
<td>6</td>
<td>18</td>
<td>16</td>
<td>10</td>
<td>11</td>
<td>14</td>
<td>4</td>
<td>12</td>
<td>17</td>
<td>13</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>7</td>
<td>9</td>
<td>17</td>
<td>11</td>
<td>11</td>
<td>6</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>16</td>
<td>7</td>
<td>9</td>
<td>17</td>
<td>11</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>11</td>
<td>17</td>
<td>9</td>
<td>7</td>
<td>16</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>11</td>
<td>17</td>
<td>9</td>
<td>7</td>
<td>16</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>17</td>
<td>7</td>
<td>4</td>
<td>5</td>
<td>11</td>
<td>9</td>
<td>16</td>
<td>1</td>
<td>6</td>
<td>17</td>
<td>7</td>
<td>4</td>
<td>5</td>
<td>11</td>
<td>9</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>11</td>
<td>1</td>
<td>11</td>
<td>1</td>
<td>7</td>
<td>11</td>
<td>1</td>
<td>7</td>
<td>11</td>
<td>1</td>
<td>7</td>
<td>11</td>
<td>1</td>
<td>7</td>
<td>11</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>18</td>
<td>11</td>
<td>12</td>
<td>1</td>
<td>8</td>
<td>7</td>
<td>18</td>
<td>11</td>
<td>12</td>
<td>1</td>
<td>8</td>
<td>7</td>
<td>18</td>
<td>11</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>7</td>
<td>6</td>
<td>16</td>
<td>11</td>
<td>4</td>
<td>17</td>
<td>1</td>
<td>9</td>
<td>5</td>
<td>7</td>
<td>6</td>
<td>16</td>
<td>11</td>
<td>4</td>
<td>17</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>12</td>
<td>6</td>
<td>3</td>
<td>11</td>
<td>15</td>
<td>17</td>
<td>18</td>
<td>9</td>
<td>14</td>
<td>17</td>
<td>18</td>
<td>13</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>7</td>
<td>1</td>
<td>11</td>
<td>7</td>
<td>1</td>
<td>11</td>
<td>7</td>
<td>1</td>
<td>11</td>
<td>7</td>
<td>1</td>
<td>11</td>
<td>7</td>
<td>1</td>
<td>11</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>11</td>
<td>18</td>
<td>7</td>
<td>8</td>
<td>1</td>
<td>12</td>
<td>11</td>
<td>18</td>
<td>7</td>
<td>8</td>
<td>1</td>
<td>12</td>
<td>11</td>
<td>18</td>
<td>7</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>17</td>
<td>12</td>
<td>4</td>
<td>14</td>
<td>11</td>
<td>10</td>
<td>16</td>
<td>18</td>
<td>6</td>
<td>2</td>
<td>7</td>
<td>15</td>
<td>5</td>
<td>8</td>
<td>9</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>6</td>
<td>8</td>
<td>17</td>
<td>10</td>
<td>7</td>
<td>3</td>
<td>4</td>
<td>18</td>
<td>5</td>
<td>13</td>
<td>11</td>
<td>2</td>
<td>9</td>
<td>12</td>
<td>16</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>16</td>
<td>12</td>
<td>9</td>
<td>2</td>
<td>11</td>
<td>13</td>
<td>5</td>
<td>18</td>
<td>4</td>
<td>3</td>
<td>7</td>
<td>10</td>
<td>17</td>
<td>8</td>
<td>6</td>
<td>14</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>9</td>
<td>11</td>
<td>5</td>
<td>4</td>
<td>7</td>
<td>17</td>
<td>6</td>
<td>1</td>
<td>16</td>
<td>9</td>
<td>11</td>
<td>5</td>
<td>4</td>
<td>7</td>
<td>17</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>4</td>
<td>11</td>
<td>16</td>
<td>6</td>
<td>7</td>
<td>5</td>
<td>9</td>
<td>1</td>
<td>17</td>
<td>4</td>
<td>11</td>
<td>16</td>
<td>6</td>
<td>7</td>
<td>5</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>18</td>
<td>1</td>
<td>18</td>
<td>1</td>
<td>18</td>
<td>1</td>
<td>18</td>
<td>1</td>
<td>18</td>
<td>1</td>
<td>18</td>
<td>1</td>
<td>18</td>
<td>1</td>
<td>18</td>
<td>1</td>
</tr>
</tbody>
</table>
3.5 Discrete Logarithm

More generally, we can say that the highest possible exponent to which a number can belong (mod \( n \)) is \( \phi(n) \). If a number is of this order, it is referred to as a **primitive root** of \( n \). The importance of this notion is that if \( a \) is a primitive root of \( n \), then its powers

\[
a, a^2, \ldots, a^{\phi(n)}
\]

are distinct (mod \( n \)) and are all relatively prime to \( n \). In particular, for a prime number \( p \), if \( a \) is a primitive root of \( p \), then

\[
a, a^2, \ldots, a^{p-1}
\]

are distinct (mod \( p \)). For the prime number 19, its primitive roots are 2, 3, 10, 13, 14, and 15.

Not all integers have primitive roots. In fact, the only integers with primitive roots are those of the form \( 2, 4, p^\alpha, \) and \( 2p^\alpha \), where \( p \) is any odd prime and \( \alpha \) is a positive integer.
3.5 Discrete Logarithm

Logarithms for Modular Arithmetic

With ordinary positive real numbers, the logarithm function is the inverse of exponentiation. An analogous function exists for modular arithmetic.

Let us briefly review the properties of ordinary logarithms. The logarithm of a number is defined to be the power to which some positive base (except 1) must be raised in order to equal the number. That is, for base $x$ and for a value $y$,

$$y = x^{\log_x(y)}$$

The properties of logarithms include

$$\log_x(1) = 0$$
$$\log_x(x) = 1$$
$$\log_x(yz) = \log_x(y) + \log_x(z)$$
$$\log_x(y^r) = r \times \log_x(y)$$

Consider a primitive root $a$ for some prime number $p$ (the argument can be developed for nonprimes as well). Then we know that the powers of $a$ from 1 through $(p - 1)$ produce each integer from 1 through $(p - 1)$ exactly once. We also know that any integer $b$ satisfies

$$b = r \pmod{p}$$

for some $r$, where $0 \leq r \leq (p - 1)$

by the definition of modular arithmetic. It follows that for any integer $b$ and a primitive root $a$ of prime number $p$, we can find a unique exponent $i$ such that

$$b = a^i \pmod{p} \quad \text{where} \quad 0 \leq i \leq (p - 1)$$
This exponent $i$ is referred to as the **discrete logarithm** of the number $b$ for the base $a \pmod{p}$. We denote this value as $\text{dlog}_{a,p}(b)$.

Note the following:

\[
\text{dlog}_{a,p}(1) = 0 \quad \text{because} \quad a^0 \pmod{p} = 1 \pmod{p} = 1
\]

\[
\text{dlog}_{a,p}(a) = 1 \quad \text{because} \quad a^1 \pmod{p} = a
\]

Here is an example using a nonprime modulus, $n = 9$. Here $\phi(n) = 6$ and $a = 2$ is a primitive root. We compute the various powers of $a$ and find

\[
\begin{align*}
2^0 &= 1 \\
2^1 &= 2 \\
2^2 &= 4 \\
2^3 &= 8 \\
2^4 &= 7 \pmod{9} \\
2^5 &= 5 \pmod{9} \\
2^6 &= 1 \pmod{9}
\end{align*}
\]

This gives us the following table of the numbers with given discrete logarithms $(\pmod{9})$ for the root $a = 2$:

<table>
<thead>
<tr>
<th>Logarithm</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>7</td>
<td>5</td>
</tr>
</tbody>
</table>

To make it easy to obtain the discrete logarithms of a given number, we rearrange the table:

<table>
<thead>
<tr>
<th>Number</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>5</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logarithm</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>
3.5 Discrete Logarithm

Any positive integer \( z \) can be expressed in the form \( z = q + k\phi(n) \), with \( 0 \leq q < \phi(n) \). Therefore, by Euler’s theorem,

\[
a^z = a^q \pmod{n} \quad \text{if } z = q \pmod{\phi(n)}
\]

Applying this to the foregoing equality, we have

\[
d\log_{a,p}(xy) = [d\log_{a,p}(x) + d\log_{a,p}(y)](\pmod{\phi(p)})
\]

and generalizing,

\[
d\log_{a,p}(y^r) = [r \times d\log_{a,p}(y)](\pmod{\phi(p)})
\]

This demonstrates the analogy between true logarithms and discrete logarithms.

Keep in mind that unique discrete logarithms mod \( m \) to some base \( a \) exist only if \( a \) is a primitive root of \( m \).

Table 8.4, which is directly derived from Table 8.3, shows the sets of discrete logarithms that can be defined for modulus 19.

<table>
<thead>
<tr>
<th>( a )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \log_{2,19}(a) )</td>
<td>18</td>
<td>1</td>
<td>13</td>
<td>2</td>
<td>16</td>
<td>14</td>
<td>6</td>
<td>3</td>
<td>8</td>
<td>17</td>
<td>12</td>
<td>15</td>
<td>5</td>
<td>7</td>
<td>11</td>
<td>4</td>
<td>10</td>
<td>9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( a )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \log_{3,19}(a) )</td>
<td>18</td>
<td>7</td>
<td>1</td>
<td>14</td>
<td>4</td>
<td>8</td>
<td>6</td>
<td>3</td>
<td>2</td>
<td>11</td>
<td>12</td>
<td>15</td>
<td>17</td>
<td>13</td>
<td>5</td>
<td>10</td>
<td>16</td>
<td>9</td>
</tr>
</tbody>
</table>
### 3.5 Discrete Logarithm

|      | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| \(a\) | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| \(log_{10,19}(a)\) | 18 | 17 | 5  | 16 | 2  | 4  | 12 | 15 | 10 | 1  | 6  | 3  | 13 | 11 | 7  | 14 | 8  | 9  |

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>(log_{13,19}(a))</td>
<td>18</td>
<td>11</td>
<td>17</td>
<td>4</td>
<td>14</td>
<td>10</td>
<td>12</td>
<td>15</td>
<td>16</td>
<td>7</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>13</td>
<td>8</td>
<td>2</td>
<td>9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>(log_{14,19}(a))</td>
<td>18</td>
<td>13</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td>2</td>
<td>6</td>
<td>3</td>
<td>14</td>
<td>5</td>
<td>12</td>
<td>15</td>
<td>11</td>
<td>1</td>
<td>17</td>
<td>16</td>
<td>4</td>
<td>9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>(log_{15,19}(a))</td>
<td>18</td>
<td>5</td>
<td>11</td>
<td>10</td>
<td>8</td>
<td>16</td>
<td>12</td>
<td>15</td>
<td>4</td>
<td>13</td>
<td>6</td>
<td>3</td>
<td>7</td>
<td>17</td>
<td>1</td>
<td>2</td>
<td>14</td>
<td>9</td>
</tr>
</tbody>
</table>
3.6 Diffie-Hellman Key Exchange Algorithm

The first published public-key algorithm appeared in the seminal paper by Diffie and Hellman that defined public-key cryptography and is generally referred to as Diffie-Hellman key exchange. A number of commercial products employ this key exchange technique.

The purpose of the algorithm is to enable two users to securely exchange a key that can then be used for subsequent encryption of messages. The algorithm itself is limited to the exchange of secret values.

The Diffie-Hellman algorithm depends for its effectiveness on the difficulty of computing discrete logarithms. Briefly, we can define the discrete logarithm in the following way. Recall from Chapter 8 that a primitive root of a prime number \( p \) as one whose powers modulo \( p \) generate all the integers from 1 to \( p - 1 \). That is, if \( a \) is a primitive root of the prime number \( p \), then the numbers

\[
a \mod p, \ a^2 \mod p, \ldots, \ a^{p-1} \mod p
\]

are distinct and consist of the integers from 1 through \( p - 1 \) in some permutation.

For any integer \( b \) and a primitive root \( a \) of prime number \( p \), we can find a unique exponent \( i \) such that

\[
b = a^i \pmod{p} \quad \text{where} \ 0 \leq i \leq (p - 1)
\]

The exponent \( i \) is referred to as the **discrete logarithm** of \( b \) for the base \( a \mod p \). We express this value as \( \text{dlog}_{a,p}(b) \).
3.6 Diffie-Hellman Key Exchange

Figure Below summarizes the Diffie-Hellman key exchange algorithm. For this scheme, there are two publicly known numbers: a prime number $q$ and an integer $\alpha$ that is a primitive root of $q$. Suppose the users A and B wish to exchange a key. User A selects a random integer $X_A < q$ and computes $Y_A = \alpha^{X_A} \mod q$. Similarly, user B independently selects a random integer $X_B < q$ and computes $Y_B = \alpha^{X_B} \mod q$. Each side keeps the $X$ value private and makes the $Y$ value available publicly to the other side. User A computes the key as $K = (Y_B)^{X_A} \mod q$ and user B computes the key as $K = (Y_A)^{X_B} \mod q$. These two calculations produce identical results:

$$K = (Y_B)^{X_A} \mod q$$
$$= (\alpha^{X_B} \mod q)^{X_A} \mod q$$
$$= (\alpha^{X_B})^{X_A} \mod q$$
$$= \alpha^{X_B X_A} \mod q$$
$$= (\alpha^{X_A})^{X_B} \mod q$$
$$= (\alpha^{X_A \mod q})^{X_B} \mod q$$
$$= (Y_A)^{X_B} \mod q$$

by the rules of modular arithmetic

The result is that the two sides have exchanged a secret value. Furthermore, because $X_A$ and $X_B$ are private, an adversary only has the following ingredients to work with: $q$, $\alpha$, $Y_A$, and $Y_B$. Thus, the adversary is forced to take a discrete logarithm to determine the key. For example, to determine the private key of user B, an adversary must compute

$$X_B = \text{dlog}_{\alpha,q}(Y_B)$$

The adversary can then calculate the key $K$ in the same manner as user B calculates it.
### 3.6 Diffie-Hellman Key Exchange

#### Global Public Elements

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$q$</td>
<td>prime number</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>$\alpha &lt; q$ and $\alpha$ a primitive root of $q$</td>
</tr>
</tbody>
</table>

#### User A Key Generation

- Select private $X_A$
  - $X_A < q$
- Calculate public $Y_A$
  - $Y_A = \alpha^{X_A} \mod q$

#### User B Key Generation

- Select private $X_B$
  - $X_B < q$
- Calculate public $Y_B$
  - $Y_B = \alpha^{X_B} \mod q$

#### Calculation of Secret Key by User A

$$K = (Y_B)^{X_A} \mod q$$

#### Calculation of Secret Key by User B

$$K = (Y_A)^{X_B} \mod q$$
3.7 Security of Diffie-Hellman Key Exchange

The security of the Diffie-Hellman key exchange lies in the fact that, while it is relatively easy to calculate exponentials modulo a prime, it is very difficult to calculate discrete logarithms. For large primes, the latter task is considered infeasible.

Here is an example. Key exchange is based on the use of the prime number \( q = 353 \) and a primitive root of 353, in this case \( \alpha = 3 \). A and B select secret keys \( X_A = 97 \) and \( X_B = 233 \), respectively. Each computes its public key:

- A computes \( Y_A = 3^{97} \mod 353 = 40 \).
- B computes \( Y_B = 3^{233} \mod 353 = 248 \).

After they exchange public keys, each can compute the common secret key:

- A computes \( K = (Y_B)^{X_A} \mod 353 = 248^{97} \mod 353 = 160 \).
- B computes \( K = (Y_A)^{X_B} \mod 353 = 40^{233} \mod 353 = 160 \).

We assume an attacker would have available the following information:

\[
q = 353; \alpha = 3; Y_A = 40; Y_B = 248
\]

In this simple example, it would be possible by brute force to determine the secret key 160. In particular, an attacker E can determine the common key by discovering a solution to the equation \( 3^a \mod 353 = 40 \) or the equation \( 3^b \mod 353 = 248 \). The brute-force approach is to calculate powers of 3 modulo 353, stopping when the result equals either 40 or 248. The desired answer is reached with the exponent value of 97, which provides \( 3^{97} \mod 353 = 40 \).

With larger numbers, the problem becomes impractical.
3.8 Key Exchange Protocols

Figure Below shows a simple protocol that makes use of the Diffie-Hellman calculation. Suppose that user A wishes to set up a connection with user B and use a secret key to encrypt messages on that connection. User A can generate a one-time private key $X_A$, calculate $Y_A$, and send that to user B. User B responds by generating a private value $X_B$, calculating $Y_B$, and sending $Y_B$ to user A. Both users can now calculate the key. The necessary public values $q$ and $\alpha$ would need to be known ahead of time. Alternatively, user A could pick values for $q$ and $\alpha$ and include those in the first message.

As an example of another use of the Diffie-Hellman algorithm, suppose that a group of users (e.g., all users on a LAN) each generate a long-lasting private value $X_i$ (for user $i$) and calculate a public value $Y_i$. These public values, together with global public values for $q$ and $\alpha$, are stored in some central directory. At any time, user $j$ can access user $i$’s public value, calculate a secret key, and use that to send an encrypted message to user A. If the central directory is trusted, then this form of communication provides both confidentiality and a degree of authentication. Because only $i$ and $j$ can determine the key, no other user can read the message (confidentiality). Recipient $i$ knows that only user $j$ could have created a message using this key (authentication). However, the technique does not protect against replay attacks.
3.8 Key Exchange Protocols

User A

- Generate random $X_A < q$
- Calculate $Y_A = \alpha^{X_A} \mod q$
- Calculate $K = (Y_B)^{X_A} \mod q$

User B

- Generate random $X_B < q$
- Calculate $Y_B = \alpha^{X_B} \mod q$
- Calculate $K = (Y_A)^{X_B} \mod q$
3.8 Man-in-the-Middle Attacks

The protocol depicted in last Figure is insecure against a man-in-the-middle attack. Suppose Alice and Bob wish to exchange keys, and Darth is the adversary. The attack proceeds as follows.

1. Darth prepares for the attack by generating two random private keys $X_{D1}$ and $X_{D2}$ and then computing the corresponding public keys $Y_{D1}$ and $Y_{D2}$.
2. Alice transmits $Y_A$ to Bob.
3. Darth intercepts $Y_A$ and transmits $Y_{D1}$ to Bob. Darth also calculates $K2 = (Y_A)^{X_{D2}} \mod q$.
4. Bob receives $Y_{D1}$ and calculates $K1 = (Y_{D1})^{X_B} \mod q$.
5. Bob transmits $Y_B$ to Alice.
6. Darth intercepts $Y_B$ and transmits $Y_{D2}$ to Alice. Darth calculates $K1 = (Y_B)^{X_{D1}} \mod q$.
7. Alice receives $Y_{D2}$ and calculates $K2 = (Y_{D2})^{X_A} \mod q$.

At this point, Bob and Alice think that they share a secret key, but instead Bob and Darth share secret key $K1$ and Alice and Darth share secret key $K2$. All future communication between Bob and Alice is compromised in the following way.
3.8 Man-in-the-Middle Attacks

1. Alice sends an encrypted message \( M: E(K2, M) \).
2. Darth intercepts the encrypted message and decrypts it to recover \( M \).
3. Darth sends Bob \( E(K1, M) \) or \( E(K1, M') \), where \( M' \) is any message. In the first case, Darth simply wants to eavesdrop on the communication without altering it. In the second case, Darth wants to modify the message going to Bob.

The key exchange protocol is vulnerable to such an attack because it does not authenticate the participants. This vulnerability can be overcome with the use of digital signatures and public-key certificates.
End of Chapter 3/Part2